KNOWLEDGE CONSTRUCTION OF 3D GEOMETRY IN VIRTUAL REALITY MICROWORLDS

Andy Ju-Chih Yeh
葉儒智

BEd (National PingTung University of Education, Taiwan)
MEd (National University of Tainan, Taiwan)

A thesis submitted in fulfilment of the requirements for degree of
Doctor of Philosophy

Centre for Learning Innovation
Faculty of Education
Queensland University of Technology

August, 2007
For my parents

黄英才 and 黄林玉
My beloved wife

Ariel Jui-Hua Tseng

and

My lovely sons

Nicholas Kuang-Hsiang Yeh

Oliver Kuang-Hsien Yeh
KEYWORDS

ABSTRACT

The recent development of virtual reality (VR) technology carries powerful potential that can be utilised to facilitate the learning of 3D geometry. Therefore, a new approach for teaching and learning of 3D geometry that utilises a virtual reality learning environment (VRLE) is proposed in this research study. This research study aimed to: (a) design and evaluate a VRLE to facilitate the learning of 3D geometry concepts and processes by upper primary school students, and (b) generate theoretical and design principles that will have application both within and beyond the immediate research study.

The research methodology employed was design experiments or design-based research. Informed by this methodology, the research design consisted of iterative cycles of developing/revising a conceptual framework, designing/prototyping a VRLE, enacting/evaluating the VRLE, and reflecting/redesigning the research. An initial conceptual framework was generated through extensive literature review to inform the design and evaluation of a VRLE. Based on the conceptual framework, a prototype VRLE named VRMath was then designed and implemented.

The enactment and evaluation of VRMath consisted of two iterations. Iteration 1 (six hours/sessions with two students of Year 5 and 6) was conducted using the prototype VRMath (Yeh & Nason, 2004). Based on the findings from Iteration 1, nine learning activities were developed and research protocols (e.g., observation and interview) were revised for Iteration 2. Iteration 2 involved six primary school students (Year 4-5) for eight weeks (two hours/sessions per week). Findings from Iteration 2 confirmed and identified some usability issues of VRMath system and many new ways of thinking and doing 3D geometry when students interacted with VRMath. These have implications on the design of VRMath and the teaching and learning of 3D geometry within the VRMath environment.

Justifications about the conceptual framework and students’ learning within VRMath were made after the two iterations of enactment and evaluation. The learning activities and VRMath were also revised and redesigned for the preparation of future iterations. After a full cycle of the design-experiments, this research study concluded with a proto-theory (semiotic framework) for the design of and learning
within VRLEs, and visions for using VRLEs in mathematic and technology education.
# KEYWORDS

... ................................................................. i

# ABSTRACT

... ............................................................... iii

# TABLE OF CONTENTS

... ............................................................... v

# LIST OF FIGURES

... ............................................................... xi

# LIST OF TABLES

... ............................................................... xv

# STATEMENT OF ORIGINAL AUTHORSHIP

... ............................................................... xvii

# ACKNOWLEDGEMENTS

... ............................................................... xix

# CHAPTER 1

**SETTING THE SCENE** ........................................ 1

1.1 Overview .................................................. 1

1.2 Aims .......................................................... 1

1.3 Context ...................................................... 1

1.4 Significance ................................................ 4

1.5 Research design ........................................... 6

1.5.1 Methodology ........................................... 6

1.5.2 Stages .................................................... 9

1.6 Chapter outlines .......................................... 11

# CHAPTER 2

**DEVELOPMENT OF CONCEPTUAL FRAMEWORK** ............... 15

2.1 Overview ................................................... 15

2.2 Learning in mathematics .................................. 15

2.2.1 What is mathematics? .................................. 16

2.2.1.1 Influence of beliefs and philosophies about the nature of mathematics on teaching practices ........................................ 16

2.2.1.2 Mathematics as meaning-making .................. 19

2.2.1.3 Discussion ........................................... 26

2.2.2 What is learning? ........................................ 28

2.2.2.1 Constructivism as a learning theory ............... 28

2.2.2.2 Constructionism as a learning theory .............. 31

2.2.2.3 Discussion ........................................... 33

2.2.3 Synthesis ................................................ 33

2.3 3D geometry ................................................ 36

2.3.1 Components of 3D geometry .......................... 36

2.3.1.1 External material world ............................ 37

2.3.1.2 Internal spatial abilities ........................... 39

2.3.1.3 Communication of 3D geometry concepts ......... 42

2.3.2 Development and understanding of 3D geometry .... 44

2.3.2.1 Piaget's stages of development .................. 45

2.3.2.2 van Hiele's levels of geometrical understanding . 46

2.3.3 Teaching and learning of 3D geometry ............... 49

2.3.3.1 Teaching and learning approach .................. 50
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3.3.2 Computers and 3D geometry</td>
<td>53</td>
</tr>
<tr>
<td>2.3.4 Synthesis</td>
<td>66</td>
</tr>
<tr>
<td>2.4 Technology</td>
<td>69</td>
</tr>
<tr>
<td>2.4.1 The nature of technology use in mathematics education</td>
<td>70</td>
</tr>
<tr>
<td>2.4.2 The role of computer</td>
<td>72</td>
</tr>
<tr>
<td>2.4.3 Microworld</td>
<td>74</td>
</tr>
<tr>
<td>2.4.4 Computer programming language</td>
<td>76</td>
</tr>
<tr>
<td>2.4.4.1 The domain of computer science</td>
<td>76</td>
</tr>
<tr>
<td>2.4.4.2 The educational context</td>
<td>78</td>
</tr>
<tr>
<td>2.4.4.3 The case of Logo</td>
<td>81</td>
</tr>
<tr>
<td>2.4.5 Virtual Reality</td>
<td>84</td>
</tr>
<tr>
<td>2.4.5.1 Scope</td>
<td>85</td>
</tr>
<tr>
<td>2.4.5.2 General environment</td>
<td>86</td>
</tr>
<tr>
<td>2.4.5.3 Avatar</td>
<td>87</td>
</tr>
<tr>
<td>2.4.5.4 Objects</td>
<td>88</td>
</tr>
<tr>
<td>2.4.5.5 Behaviours</td>
<td>89</td>
</tr>
<tr>
<td>2.4.5.6 Navigation modes</td>
<td>90</td>
</tr>
<tr>
<td>2.4.6 Design of technology-rich learning environments</td>
<td>91</td>
</tr>
<tr>
<td>2.4.6.1 Assumptions and principles of design</td>
<td>92</td>
</tr>
<tr>
<td>2.4.6.2 Process of design</td>
<td>94</td>
</tr>
<tr>
<td>2.4.6.3 Evaluation of design</td>
<td>95</td>
</tr>
<tr>
<td>2.4.7 Synthesis</td>
<td>98</td>
</tr>
<tr>
<td>2.5 Conceptual framework for the research study</td>
<td>100</td>
</tr>
<tr>
<td>2.5.1 Specifications to inform the design of the VRLE</td>
<td>101</td>
</tr>
<tr>
<td>2.5.2 Perspectives to inform the evaluation of student learning of 3D</td>
<td>103</td>
</tr>
<tr>
<td>geometry within the VRLE</td>
<td></td>
</tr>
</tbody>
</table>

**CHAPTER 3**  
**DESIGN OF PROTOTYPE VRLE** .......................................................... 105

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Overview</td>
<td>105</td>
</tr>
<tr>
<td>3.2 Technical considerations</td>
<td>105</td>
</tr>
<tr>
<td>3.3 Phase 1: Identification of components</td>
<td>107</td>
</tr>
<tr>
<td>3.4 Phase 2: Design of topological component (VR interface)</td>
<td>108</td>
</tr>
<tr>
<td>3.4.1 3D navigation</td>
<td>109</td>
</tr>
<tr>
<td>3.4.2 VR tool bar</td>
<td>111</td>
</tr>
<tr>
<td>3.5 Phase 3: Design of typological component (programming interface)</td>
<td>114</td>
</tr>
<tr>
<td>3.5.1 Extension of Logo language in VRMath (commands)</td>
<td>114</td>
</tr>
<tr>
<td>3.5.1.1 Move commands</td>
<td>115</td>
</tr>
<tr>
<td>3.5.1.2 Turn commands</td>
<td>116</td>
</tr>
<tr>
<td>3.5.1.3 Create commands</td>
<td>117</td>
</tr>
<tr>
<td>3.5.1.4 Material commands</td>
<td>118</td>
</tr>
<tr>
<td>3.5.1.5 Setting commands</td>
<td>119</td>
</tr>
<tr>
<td>3.5.2 Graphic User Interface (GUI) of programming interface</td>
<td>120</td>
</tr>
<tr>
<td>3.5.2.1 Project menu</td>
<td>121</td>
</tr>
<tr>
<td>3.5.2.2 Edit menu</td>
<td>122</td>
</tr>
<tr>
<td>3.5.2.3 Tool menu</td>
<td>123</td>
</tr>
<tr>
<td>3.5.2.4 Environment menu</td>
<td>127</td>
</tr>
<tr>
<td>3.6 Phase 4: Design of social-actional component (hypermedia and forum</td>
<td>128</td>
</tr>
<tr>
<td>interface)</td>
<td></td>
</tr>
<tr>
<td>3.6.1 Hypermedia documentation</td>
<td>128</td>
</tr>
</tbody>
</table>
Appendix 4: Pre interview questions for Iteration 2 ........................................ 355
Appendix 5: Post interview questions in Iteration 2 ...................................... 356
Appendix 6: Navigation toolbar usability test ..................................................... 357
Appendix 7: Public projects in VRMath created by the researcher ...................... 358
Appendix 8: Users’ projects in VRMath .............................................................. 362
Appendix 9: Final group projects in VRMath ...................................................... 371
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>VR4</td>
<td>3</td>
</tr>
<tr>
<td>1.2</td>
<td>5DT Data Glove 5</td>
<td>3</td>
</tr>
<tr>
<td>1.3</td>
<td>The complex features of design experiments</td>
<td>8</td>
</tr>
<tr>
<td>1.4</td>
<td>Stages of the research design</td>
<td>10</td>
</tr>
<tr>
<td>1.5</td>
<td>Stages of the design experiments with corresponding chapters</td>
<td>11</td>
</tr>
<tr>
<td>2.1</td>
<td>Peirce’s triad integrated with Saussure’s tradition of semiotics</td>
<td>20</td>
</tr>
<tr>
<td>2.2</td>
<td>Types of inference</td>
<td>22</td>
</tr>
<tr>
<td>2.3</td>
<td>Topological versus Typological Semiosis</td>
<td>24</td>
</tr>
<tr>
<td>2.4</td>
<td>Initial conceptual framework</td>
<td>34</td>
</tr>
<tr>
<td>2.5</td>
<td>Chambered Nautilus shell and golden ratio</td>
<td>37</td>
</tr>
<tr>
<td>2.6</td>
<td>Fern leaf and L-System fractal program</td>
<td>38</td>
</tr>
<tr>
<td>2.7</td>
<td>Diagnosis in teaching and learning activities</td>
<td>52</td>
</tr>
<tr>
<td>2.8</td>
<td>Pythagoras tree and cubic shape in Elica Logo</td>
<td>55</td>
</tr>
<tr>
<td>2.9</td>
<td>Spherical solids and polyhedron in MSWLogo</td>
<td>56</td>
</tr>
<tr>
<td>2.10</td>
<td>Conic section in Geometer’s SketchPad</td>
<td>58</td>
</tr>
<tr>
<td>2.11</td>
<td>3D cube and projected rectangular box in Cabri-Géomètre</td>
<td>59</td>
</tr>
<tr>
<td>2.12</td>
<td>A cylinder in CosmoWorlds</td>
<td>60</td>
</tr>
<tr>
<td>2.13</td>
<td>Spatial visualisation software using virtual reality</td>
<td>61</td>
</tr>
<tr>
<td>2.14</td>
<td>Examples of objects constructed using the MAT3D</td>
<td>61</td>
</tr>
<tr>
<td>2.15</td>
<td>A simple spiral move in AquaMOOSE 3D</td>
<td>63</td>
</tr>
<tr>
<td>2.16</td>
<td>Examples of multimedia for teaching and learning 3D geometry</td>
<td>64</td>
</tr>
<tr>
<td>2.17</td>
<td>Hypermedia learning environment for teaching descriptive geometry</td>
<td>65</td>
</tr>
<tr>
<td>2.18</td>
<td>Math Forum web discussion in geometry</td>
<td>66</td>
</tr>
<tr>
<td>2.19</td>
<td>Semiotic triad of 3D geometry</td>
<td>67</td>
</tr>
<tr>
<td>2.20</td>
<td>Coordinate system in VRML space</td>
<td>86</td>
</tr>
<tr>
<td>2.21</td>
<td>Avatars in a multi-user virtual world</td>
<td>87</td>
</tr>
<tr>
<td>2.22</td>
<td>SpaceMouse and SpaceBall</td>
<td>90</td>
</tr>
<tr>
<td>3.1</td>
<td>Prototype of VRMath</td>
<td>108</td>
</tr>
<tr>
<td>3.2</td>
<td>3D rotation in VRMath</td>
<td>116</td>
</tr>
<tr>
<td>3.3</td>
<td>A frame of cube and a face created in VRMath</td>
<td>117</td>
</tr>
<tr>
<td>3.4</td>
<td>Primitive objects of VRMath</td>
<td>118</td>
</tr>
<tr>
<td>3.5</td>
<td>A flat sphere in VRMath</td>
<td>120</td>
</tr>
<tr>
<td>3.6</td>
<td>The layout of programming interface GUI</td>
<td>121</td>
</tr>
<tr>
<td>3.7</td>
<td>Project menu</td>
<td>122</td>
</tr>
<tr>
<td>3.8</td>
<td>Edit menu</td>
<td>122</td>
</tr>
<tr>
<td>3.9</td>
<td>Procedure Editor</td>
<td>123</td>
</tr>
<tr>
<td>3.10</td>
<td>Procedure list dialog</td>
<td>123</td>
</tr>
<tr>
<td>3.11</td>
<td>Tool menu</td>
<td>123</td>
</tr>
<tr>
<td>3.12</td>
<td>Background Chooser</td>
<td>124</td>
</tr>
<tr>
<td>3.13</td>
<td>Pen Color Editor</td>
<td>125</td>
</tr>
<tr>
<td>3.14</td>
<td>Font Chooser</td>
<td>125</td>
</tr>
<tr>
<td>3.15</td>
<td>Material Editor</td>
<td>126</td>
</tr>
<tr>
<td>3.16</td>
<td>Quick Command</td>
<td>127</td>
</tr>
<tr>
<td>3.17</td>
<td>Options</td>
<td>127</td>
</tr>
<tr>
<td>3.18</td>
<td>Environment menu</td>
<td>128</td>
</tr>
<tr>
<td>3.19</td>
<td>VRMath Forum</td>
<td>131</td>
</tr>
</tbody>
</table>
Figure 4.1 Fixed frame of reference cubes ......................................................... 145
Figure 6.1 Forum icon ..................................................................................... 168
Figure 6.2 Forum register link ......................................................................... 168
Figure 6.3 Information required for Forum registration .................................... 168
Figure 6.4 Viewing user profile in forum .......................................................... 170
Figure 6.5 Initial Avatars of participants ......................................................... 170
Figure 6.6 Web posting form for sending private message .............................. 171
Figure 6.7 The secret of the turtle’s eyes ....................................................... 175
Figure 6.8 The legend of the turtle ................................................................. 176
Figure 6.9 Beneath the turtle and heading up ............................................... 180
Figure 6.10 Dragging to navigate in Walk mode ............................................ 182
Figure 6.11 A poll in VRMath Forum about 3D navigation ............................. 184
Figure 6.12 Quick Command UI explanation ................................................ 189
Figure 6.13 Objects created by repeat command .......................................... 191
Figure 6.14 A poll for turtle dance activity .................................................... 196
Figure 7.1 Five primitive objects in VRMath ................................................. 201
Figure 7.2 A poll for Material Editor .............................................................. 201
Figure 7.3 Planet Saturn ................................................................................ 202
Figure 7.4 Scaled primitive objects in VRMath ............................................. 203
Figure 7.5 A tower made by scaled primitive objects ................................. 204
Figure 7.6 A candle made by scaled primitive objects ................................. 204
Figure 7.7 A long box created by 12 cubes .................................................... 209
Figure 7.8 A staircase created by 12 cubes .................................................... 209
Figure 7.9 A walkable staircase created by 12 cubes ..................................... 210
Figure 7.10 A spiral structure of up-side-down cones ................................. 212
Figure 7.11 A spiral staircase from knowledge building discourse .................. 212
Figure 7.12 A spiral DNA structure ............................................................... 213
Figure 7.13 A poll about why number 12 is so popular .................................. 214
Figure 7.14 Avatar View experiment about 3D rotation ................................ 219
Figure 7.15 Turtle experiment about 3D rotation ......................................... 220
Figure 7.16 Create 3D objects experiment about 3D rotation (1) ................. 220
Figure 7.17 Create 3D objects experiment about 3D rotation (2) ................. 221
Figure 7.18 Reasoning from moving for turning ........................................... 222
Figure 7.19 A cone shape formed by 3D rotations ....................................... 224
Figure 7.20 Bonbon's triangle ....................................................................... 227
Figure 7.21 Bonbon's attempt for a square .................................................. 228
Figure 7.22 Grae's attempt for a square .......................................................... 228
Figure 7.23 Polygon scaffold ......................................................................... 229
Figure 7.24 Overlap and gap in septagons (1) ............................................. 231
Figure 7.25 Overlap and gap in septagons (2) ............................................. 232
Figure 7.26 Drawing of a 2D square in 3D virtual space .............................. 237
Figure 7.27 Rosco's cube by Quick Command ............................................ 239
Figure 7.28 Grae's cube by Quick Command .............................................. 240
Figure 7.29 Bonbon's cube by Quick Command .......................................... 240
Figure 7.30 Drawing of a 2D square in 3D virtual space ................................ 241
Figure 7.31 R2D2's cube by Quick Command ............................................. 243
Figure 7.32 Victor's pyramid by Quick Command ......................................... 244
Figure 7.33 The effects of different frames of reference on cubes ................. 245
Figure 7.34 Alekat20's six cubes ................................................................. 246
Figure 7.35 Repeated cubes of Victor's procedure ....................................... 246
Figure 8.1 50 random balls in space ................................................................. 261
Figure 8.2 50 random balls connected with lines in space ............................ 261
Figure 8.3 A weird virtual world by Bonbon .................................................. 262
Figure 8.4 A window on a wall ...................................................................... 264
Figure 8.5 Stonehenge for Temple project ...................................................... 267
Figure 8.6 R2D2's Temple drawing................................................................. 267
Figure 8.7 Grae, Bonbon, and Rosco's house plans ....................................... 269
Figure 8.8 Three proportional beds ............................................................... 270
Figure 8.9 Bonbon's household items .............................................................. 271
Figure 8.10 Rosco's house body ..................................................................... 272
Figure 8.11 Rosco's detailed plan of the house body ....................................... 272
Figure 8.12 Grae's fence experiment 1 ............................................................ 273
Figure 8.13 Grae's fence experiment 2 ............................................................. 274
Figure 8.14 Grae's fence experiment 3 ............................................................. 274
Figure 8.15 Grae's pool and trees .................................................................... 275
Figure 8.16 R2D2, Victor and Alekat20's Temple plans .................................. 275
Figure 8.17 R2D2's Temple stage .................................................................. 276
Figure 8.18 R2D2's Temple fountain .............................................................. 276
Figure 8.19 Victor's Temple ground without/with bridges .............................. 277
Figure 8.20 Victor's simple tree ....................................................................... 278
Figure 8.21 Alekat20's temple base on Temple ground .................................... 279
Figure 8.22 Alekat20's final temple base ......................................................... 279
Figure 8.23 Final group projects .................................................................... 281
Figure 8.24 VAM Temple's procedural structure ......................................... 283
Figure 8.25 Objects overlapping in VR space ............................................... 286
Figure 9.1 Rosco's explanation about wall_2 procedure .................................. 292
Figure 9.2 Rosco's explanation about roof procedure ..................................... 292
Figure 9.3 Suggestions for icons in Navigation Toolbar ............................... 297
Figure 9.4 Improved Quick Command interface .......................................... 297
Figure 10.1 Revised research design .............................................................. 314
LIST OF TABLES

Table 2.1 Beliefs and Philosophies about Mathematics and Teaching Practices .... 18
Table 2.2 Primitive Objects from 1D to 3D ................................................................. 38
Table 2.3 The Developmental Stages of Formal Operations and Spatial Ability ...... 45
Table 2.4 Tutor, Tool and Tutee – the Three Modes of Using Computing in Education .......................................................................................................................... 72
Table 2.5 Genres of Children’s Software ........................................................................... 73
Table 4.1 Generalisation of the Polygon ............................................................................. 143
Table 4.2 Procedures of a Cube .......................................................................................... 144
Table 4.3 Learning Activities and Perspectives for Evaluation ........................................ 151
Table 5.1 Timeline for the Enactment of VRMath ................................................................ 157
Table 5.2 Participants’ Drawing of 3D Shapes in Pre Interview ........................................ 161
Table 5.3 Summary of Pre-Interview about Participants .................................................... 163
Table 7.1 Generalisation of the Polygon (1) ..................................................................... 230
Table 7.2 Generalisation of the Polygon (2) ..................................................................... 234
Table 7.3 Procedures of a Cube by School C Participants ................................................ 242
Table 7.4 Procedures of a Cube by School K Participants ................................................ 244
Table 9.1 Comparison of Participants’ Drawing of 3D Shapes in Pre and Post Interview .......................................................................................................................... 294
Table 9.2 Navigation Toolbar Icon Test .............................................................................. 300
Table 9.3 Statistics of Forum Posts in Iteration 2 (1) .......................................................... 302
Table 9.4 Statistics of Forum Posts in Iteration 2 (2) .......................................................... 303
Table 9.5 Statistics of Private Messages in Iteration 2 ....................................................... 305
Table 9.6 Behavious of Maintaining Profiles ..................................................................... 307
STATEMENT OF ORIGINAL AUTHORSHIP

The work contained in this thesis has not been previously submitted to meet requirements for an award at this or any other higher education institution. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made.

Signature: _______________________________________

Date: ___________________________________________
ACKNOWLEDGEMENTS

First of all, I want to express my deepest gratitude to Associate Professor Rod Nason, my principal supervisor, for his unconditional and continuous support over the five years of my PhD study. Rod is a wise and perspective teacher and researcher. His expertise in using ICTs in mathematics education guided the direction of this research study. Rod has a great sense of humour. I have enjoyed much laughter during the stressful times of my study. Without his guidance and encouragement, I would not possibly have completed this thesis.

I am indebted to my associate supervisor Professor Tom Cooper. Tom always gives me positive feedback, which helps me build up confidence. His insightful advice is influential to the structure of this thesis. I am grateful to the internal assessors Professor Kar-Tin Lee, Associate Professor Stephen Ritchie, Dr. Gillian Kidman, and Professor Carmel Diezmann, and the Committee Panel Associate Professor Suzanne Carrington, Professor Lyn English, and Associate Professor Jim Watters for their constructive comments on my thesis. I also want to extend my thanks to Dr. Bill Atweh, Dr. Stephen Norton, and many friends for their friendships and supports during my study.

I wish to thank Professor Erica McWilliam for her financial support in 2005, which prevented an interruption to my PhD study. Thanks also to Mr. Shaun Nykvist for his technical support and resources for setting up the VRMath system on the server and the access to the Mini Lab. My appreciation also goes to the participating schools and students in the VRMath Project. They are the key persons to make the data collection possible.

Last but not the least, I want to thank my beloved wife and sons, for their love, care, and tenderness, are the very reason and impetus for me to complete this PhD study.
CHAPTER 1
SETTING THE SCENE

1.1 OVERVIEW

This thesis reports on a research study investigating the use of Virtual Reality (VR) technology in mathematics education to facilitate the knowledge construction by primary school students of 3-dimensional (3D) geometry concepts and processes. This chapter provides the aims, context, significance, and the overall research design of the study. It concludes with an outline of each of the chapters.

1.2 AIMS

The primary aims of this study were to: (a) design and evaluate a virtual reality learning environment (VRLE) to facilitate the learning by upper primary school students of 3D geometry concepts and processes, and (b) generate theoretical and design principles that will have application both within and beyond the immediate research study.

1.3 CONTEXT

In recent years, much disquiet has been expressed about the limitations of traditional approaches to the teaching and learning of geometry (Andrews, 1999; Battista & Clements, 1990; Mason, 1998; Papert, 1993; Rahim, 2002; Song, Han, & Lee, 2000). Alternative ways of teaching and learning geometry thus have been developed. Some of the alternative approaches have been based on van Hiele’s (1999) cognitive stage development theory whilst many others have utilised information and communication technologies (ICT). Included in the latter are Logo (Clements & Sarama, 1997; Hoyles & Noss, 1992; Papert, 1993; Resnick, 1991), Cabri-Géomètre (C. Laborde, 2000; C. Laborde & Laborde, 1995a, 1995b; J.-M. Laborde, 1996) and Geometer’s Sketchpad (Jackiw, 1991, 1995).

Unfortunately, most of the ICT tools such as Logo, Cabri-Géomètre, and Geometer’s Sketchpad, which have been utilised to reform the teaching and learning of geometry, operate within 2-dimensional (2D) environments. Therefore, it can be argued that the number and types of investigations of 3D shape, position and
orientation concepts and processes that can be investigated in these environments are rather limited. This may be in marked contrast to the number and types of investigations that can be carried out in VR environments. Because of the enhanced capabilities of VR environments for 3D exploration, it has been suggested that VR learning environments (VRLE) are worthy of investigation as tools for facilitating the teaching and learning of 3D geometry (Ainge, 1996; Allport, Sines, Schreiner, & Das, 1999; Barab et al., 2000; Chan, Lui, Ng, & Chui, 1999; Kwon, Kim, & Kim, 2002; Pasqualotti & Freitas, 2002; Song et al., 2000; Teixeira, Silva, & Silva, 1999).

The term “Virtual Reality” was first coined by Jaron Lanier in the 1980s, but its actual development can be traced back to the 1940s. VR technology uses the computer’s powerful computing ability to generate real time three-dimensional space to mimic the real world. VR comprises all kinds of multimedia but more importantly, it is three-dimensional in nature. Therefore, it is an ideal medium to express and reflect our physical environment. It can create virtual environments and simulate impossible or dangerous scenes without the limitation of space and time. VR has everything that the real world has and has not. Many educators, researchers and research institutions such as Virtual Reality and Education Laboratory (VREL) at the East Carolina University, the University of Michigan Virtual Reality Laboratory (VRL) and Human Interface Technology Laboratory (HITL) at the University of Washington in USA, have seen its potential to facilitate teaching and learning. It has at times been criticised as being over hyped. VR is definitely not a panacea, but undoubtedly has potential in education.

High-end VR requires special, expensive sensory equipments such as a Head Mounted Display (HMD) and a Data Glove (Figures 1.1 & 1.2). Users immersed in the computer generated virtual world can have visual, tactile, acoustic, multi-sensory feedback and get the most realistic and intuitive experiences. This immersion kind of VR is often regarded as the most powerful and valuable educational application that dominates VR research (Bricken, 1991; W. Bricken, 1990; Byrne, 1996; Dede, Salzman, Loftin, & Ash, 1999; Dede, Salzman, Loftin, & Sprague, 1999; Pantelidis, 1996; Pimentel & Teixeira, 1995; Psotka, 1995; Winn, 1993; Winn et al., 1999). However, at this moment, immersive VR still is laboratory based, and is not affordable to general users.
Desktop VR has made Virtual Reality available and affordable in recent years. It utilises general personal computers to generate a 3D space viewable on the 2D screen. Users can get real-time visual and audio experience to an endless space from the flat screen. VR differs from the other computer software or multimedia as it computes, renders and allows users to freely manipulate in a 3D manner. In other words, learning in VR environments enables one to naturally see, think, talk and do things three-dimensionally. In such an environment, we can envisage that it can offer these following advantages to learning:

1. Inclusiveness: VR offers a rich environment with a wide range of media, which can be cross-disciplinary and easily accessible for all (Auld & Pantelidis, 1999; Bowman, Hodges, Allison, & Wineman, 1999; Bricken, 1991; W. Bricken, 1990; Dede, 1992; Heudin, 1999; Pantelidis, 1996).

2. Real world like experience: VR not only provides a realistic 3D visualisation but also allows users to freely navigate and manipulate in the 3D space. Therefore, users can get real world like participation and experience (Bricken, 1991; Byrne, 1996; Kwon et al., 2002; Larijani, 1994; Thurman & Mattoon, 1994; Wexelblat, 1993; Winn & Jackson, 1999).

3. Impossible scene simulation: In addition to the real world like experience, VR is able to simulate dangerous and/or impossible scenes that break the physical constraints of time and space (Byrne, 1993; Dede, 1995; Hedberg & Alexander, 1994; Osberg, 1993; Pantelidis, 1996; Winn & Jackson, 1999; Yeh, 1999).
4. Extension of the microworlds paradigm to 3D spaces: VR can simulate a 3D coherent domain of complex and dynamic objects and activities, which is unprecedented and different from the existing 2D forms of microworlds (Allport et al., 1999; Barnett, Keating, Barab, & Hay, 2000; Dede, Salzman, Loftin, & Sprague, 1999; Pantelidis, 1996; Pimentel & Teixeira, 1995; Salzman, Dede, Loftin, & Chen, 1999; Winn & Jackson, 1999).

5. Networked collaborative learning: Some desktop VR technologies such as Virtual Reality Modelling Language (VRML), ActiveWorlds and Java3D permit its use on the Internet. A multi-user virtual community built upon these technologies can promote good quality networked collaborative learning and distance learning (dos Santos & Fraga, 2002; Jackson, 1998; Riva, 1999; Roussos et al., 1999; Winn & Jackson, 1999; Yeh, 1999).

6. High motivation and engagement: Learners are often highly motivated by new and fancy computer technology. VR with its vivid simulation can also engage learners in convivial learning activities (Barab et al., 2000; Bricken & Byrne, 1992; Byrne, Holland, Moffit, Hodas, & Furness, 1994; Furness, Winn, & Yu, 1997; Macpherson & Keppell, 1998; McLellan, 1993; Middleton, 1992; Pantelidis, 1996; Yeh, 1999).

7. Affordability: Desktop VR is now employed on normal PC. It may not be as powerful as High-end immersive VR, but its affordance can benefit most schools (Cronin, 1997; Davis, Huxor, & Lansdown, 1996; Kwon, 2001; Larijani, 1994; Song et al., 2000; Yeh, 1999).

Despite the many advantages that VR can offer, there are also a number of questions underlying this technology. Questions such as the elements of VR, the design of VRLE and the learning and teaching theories underpinning this technology have been identified and investigated during the course of this study.

1.4 SIGNIFICANCE

Most current curriculum materials and computer-based systems address the investigation of 3D geometry by requiring students to operate within two-dimensional representations of 3D geometry. This is not a natural way of investigating human 3D spatial ability. The limitations of these representations tend
to hinder the development of many of the 3D spatial abilities identified in the research literature (see Section 2.2.1.2). Furthermore, because of the predominance of 2D forms of representation within most learning technologies, it is not possible to investigate natural 3D thinking within these contexts.

The VRLE for the exploration of 3D geometry being designed and evaluated in this study will enable young children to investigate 3D geometry within the context of 3D representations that closely reflects the 3D world in which they operate. It was hypothesised that this will facilitate the development of many spatial abilities denied to young children by current curriculum materials and computer-based systems. It was also hypothesised that observing the children’s interactions with the VRLE may lead to the identification of 3D spatial abilities hitherto unidentified in the research literature.

Therefore, the outcomes from this research study have the potential to fill gaps within research literature, and have both practical and theoretical significance for the investigation of 3D spatial abilities and the teaching and learning of 3D geometry and the application of virtual reality systems in education.

The outcomes of this study have practical significance for the teaching and learning of geometry. It provides new ways for children to think and do 3D geometry within their classrooms and also provides teachers with new and better ways to introduce the investigation of 3D geometry concepts and processes to their students.

The outcomes also have practical significance for the ways computers are used in mathematics classrooms. It creates a new 3D technological empowered teaching and learning paradigm, and guides the future use of technology in education.

With respect to theoretical significance, this research study has generated theoretical models to inform future research and development of 3D VRLEs in the field of teaching and learning geometry. It also advances theoretical knowledge about the ways young students can investigate concepts and processes about 3D geometry. This has implications for both mathematics education and virtual reality technology.
1.5 RESEARCH DESIGN

1.5.1 Methodology

The methodology adopted for this study was design experiments. Many seminal researchers in the field of computer-mediated education such as Bereiter (2002), Brown (1992), Bruckman (2002), Collins (1993), Hawkins and Collins (1992), Hawkins (1992), Hsi (1998), and Kelly (2003) have indicated the inappropriateness of traditional research approaches for conducting research into classrooms uses of Information and Communication Technologies (ICTs). These researchers have identified a number of reasons for why traditional approaches to conducting research are not adequate for investigating computer-mediated learning in classrooms, and have started to advocate a new paradigm of research namely “design experiments”. Hsi (1998) summarised the reasons for why design experiments are much needed in research into the educational applications of ICTs:

1. Previous research models have failed: Because the goals of education are more encompassing now, the former criteria for measuring learning success on standardised tests or variables control have become inappropriate and weak for advancing our understanding of how to design for change. The practice of testing software in laboratory contexts has also failed to generate findings that can be readily transported into classroom practice. New models of research that engage teachers as co-investigators or those involve strong teacher-researcher collaborations are more promising because the needs of learners and constraints of a social context are voiced early in the design process.

2. Learners are a moving target: Learners come from a diversity of backgrounds with different experiences, prior knowledge, and beliefs. When learners are taken into new learning situations, their practices respond to the new tools and result in desired or unexpected evolutions. Learning technology and activities must be flexible to meet the evolving learning demands of students.

3. Technology is a moving target: Computer technology is developing at a rapid pace. New features of software such as interaction styles and interfaces will have profound impacts and provide opportunities to
scaffold and facilitate learning. Learning technology should also evolve to use adaptable features of new technology.

4. Real settings are messy: Unlike psychology laboratories, the real settings of classrooms are dynamic and messy. There are always some contingencies and/or variables that we cannot control. Thus, we need methodological criteria to address the dynamic settings of classrooms.

5. New methods, shared criteria, and assessment: Design experiments not only include the strengths of controlled experiments but also involve rapid prototyping, trial and refinement of hypotheses in the study of dynamic variables and social interactions. Thus, they can produce rigorous scientific findings, which contain and reflect the rich nature of unanticipated consequences and the ecological validity of research practices.

Because of all of these reasons, this researcher believed that design experiments or design-based research, which focuses not only on the improvement of instructional design for bringing about new forms of learning but also on development of theory to inform current and future research was the most appropriate research methodology for achieving the aims of this research study.

Brown (1992) presented a model to provide an overview of design experiments (see Figure 1.3). She argued that a learning environment is a complex and systemic whole so that it is impossible to change one aspect of the system without creating perturbations in others, and difficult to just study any one aspect independently from the whole operating system. The learning environment itself, its underpinning theory and its practical application are simultaneously changing (co-evolving) within their causal relationships. The inputs of the learning environment are enlarged to include any levels of participants (as in a community) and any related issues such as classroom ethos, curriculum and technology that are responsible for the changes of learning environment. And as a result, the outputs of the learning environment not only assess the goals of the learning environment, but also take the accountability of all inputs. Hence, Figure 1.3 shows a completely dynamic nature of engineering a learning environment.
The Design-Based Research Collective\(^1\) (2003) proposed five characteristics to be exhibited in good design-based research. They stated that:

First, the central goals of designing learning environments and developing theories or “prototheories” of learning are intertwined. Second, development and research take place through continuous cycles of design, enactment, analysis, and redesign. Third, research on designs must lead to sharable theories that help communicate relevant implications to practitioners and other educational designers. Fourth, research must account for how designs function in authentic settings. It must not only document success or failure but also focus on interactions that refine our understanding of the learning issues involved. Fifth, the development of such accounts relies on methods that can document and connect processes of enactment to outcomes of interest.

(The Design-Based Research Collective, 2003, p. 5)

In a similar vein, Bereiter (2002) identified four characteristics for successful design research: (a) design research is carried out by or in close collaboration with designers; (b) design research is inherently interventionist; design researchers are trying to make something happen, and thus crossing the boundary between observer and actor; (c) design research requires a community of practice in which people both believe in what they are doing and pay close attention to negative results; and (d) design research is guided by some vision of as-yet-unrealised possibilities and is

---

\(^1\) The Design-Based Research Collective is a group of faculty and researchers founded to examine, improve, and practice design-based research methods in education. The group’s members all blend research on learning and the design of educational interventions. More information can be found online at http://www.designbasedresearch.org/
characterized by emergent goals, which arise and evolve in the course of cycles of design and research.

1.5.2 Stages

Design experiments iterate over a relatively long period of time (e.g., 2-3 years) through cycles of design and research. Each cycle of the design experiment consists of four stages:

1. Development of conceptual framework,
2. Design of prototype VRLE,
3. Enactment and evaluation of VRLE, and
4. Reflection and redesign of the VRLE.

This thesis reports on the first cycle of the design experiment.

The outcome of every iteration cycle is theory building about the future design of VRLEs and the learning within VRLEs, which can inform: (a) future modifications to the conceptual frameworks, (b) refinements of the interface, functionalities, and activities of the VRLE, and (c) redesign of enactment settings and evaluation protocols. Figure 1.4 illustrates the stages of the design experiments utilised in this study.

Stage 1: Development of conceptual framework

The development of the conceptual framework to inform the design of the VRLE was the outcome of a review of research literature about mathematics and the learning of mathematics, semiotics and 3D geometry, and on issues pertaining to using technology (in particular Virtual Reality) in the teaching and learning of 3D geometry. Two facets of the overall conceptual framework including a specification to inform the design of the VRLE and perspectives to inform the evaluation of students’ learning of 3D geometry within the VRLE were generated.

Stage 2: Design of prototype VRLE

A prototype VRLE was implemented in this stage. The design of the prototype VRLE was informed by the conceptual framework developed in previous stage. Rationale for the interface design of VRLE was addressed to conform to the specification.
Stage 3: Enactment and evaluation of VRLE

At this stage, the prototype VRLE was ready for field testing. A series of orientation and learning activities were designed for the enactment and evaluation of the VRLE. Participants were chosen to interact with the VRLE. Data were collected and analysed to evaluate the usability of the VRLE and students’ learning of 3D geometry within the VRLE.

Stage 4: Reflection and redesign of VRLE

The findings from the evaluation of the VRLE in Stage 3 were reflected on in order to ascertain where modifications needed to be made to the underlying conceptual framework that informed the research and development of the present version of the learning environment. Thus, a modified set of research and design principles was generated from this reflection. In addition to informing the process of revising and improving the VRLE developed in this research study, the modified set
of research and design principles will advance the mathematics education and computer-supported learning communities’ corpus of knowledge about:

1. How to design more effective 3D geometry VRLEs;

2. How to design more effective 3D geometry learning activities within VRLEs; and

3. How young students can investigate concepts and processes about 3D geometry within VRLEs.

Following this, the VRLE and its accompanying learning activities were redesigned to take cognisance of modifications that have been made to the underlying conceptual framework.

1.6 CHAPTER OUTLINES

There are ten chapters for this thesis. Chapter 1 provides the aims, context, significance, and the overall research design of the study. From Chapter 2 to Chapter 10 is a full cycle of the design experiments of this research study. Figure 1.5 maps each stage of the design experiments with corresponding chapters.
In Chapter 2, the development of the conceptual framework to inform the design of the prototype VRLE is described. This chapter begins with an extensive literature review about mathematics and the learning of mathematics, semiotics and 3D geometry, and on issues pertaining the use of technology (in particular Virtual Reality and Logo programming language) in the teaching and learning of 3D geometry. The chapter concludes with a description of the conceptual framework generated from the literature review.

Chapter 3 reports on the Stage 2 of the study: Design of prototype VRLE. The process of designing and implementing the prototype VRLE is presented. This chapter elaborates on how the design was informed from the conceptual framework in Chapter 2, and presents a detailed description of the prototype VRLE.

The enactment and evaluation phase of this design-based research consists of two iterations. Chapter 4 reports on the first iteration of Stage 3 of the design experiments: the enactment and evaluation of the VRLE. Because of the novelty of the VRLE, Iteration 1 focuses on the design of learning activities that could utilise the unique 3D virtual space provided by the VRLE for learning 3D geometry. Five learning activities and an initial evaluation of both the design of and the learning within the VRLE are presented. Based on the results from Iteration 1, implications to inform the design of and the learning within the VRLE are drawn. This chapter then concludes with nine refined learning activities for Iteration 2.

Iteration 2 is a more extensive enactment and evaluation of the design experiments, which consists of pre-interview, enactment of nine learning activities, post-interview, usability inspection, and artefact analysis. Due to extensive nature of this iteration, Iteration 2 is reported on in Chapters 5-9.

Chapter 5 reports on the design and method, and the pre-interview of Iteration 2. The design and method section presents the participants, data collection and analysis, and procedures of enactment and evaluation settings. The pre-interview section investigates the background of the participants, and records the initial state of the participants for later evaluation and justification within the nine learning activities.

The nine learning activities were classified into three categories: Introductory, Specific Concepts, and Application types of learning activities.
Chapter 6 reports on the introductory type of learning activities, which consists of the first three learning activities aiming to familiarise the participants with the VRMath system.

Chapter 7 reports on the Specific concepts type of learning activity. This type of learning activity focuses on developing some specific mathematical and geometrical concepts. Learning Activities 4-8 are of this type.

Chapter 8 reports on the Application type of learning activity. This type of learning activity investigates how the participants utilise their geometrical understanding and the facilities provided by the VRMath system to collaboratively construct a virtual reality microworld. Learning Activity 9 is the only learning activity in this category.

Chapter 9 reports on the post-interview, usability inspection, and artefact analysis of Iteration 2. The post-interview investigates the participants' learning after the nine learning activities and compares with the initial state identified in the pre-interview. The usability inspection was conducted throughout the nine learning activity. However, this chapter reports a specific usability inspection on the design of the interface icons. The artefact analysis focuses on the participants' writings in VRMath Forum for justification of the evaluation.

Lastly, Chapter 10 presents the last stage of this design research: reflection and redesign of the VRLE. The reflection based on the enactment and evaluation has implications for the modification of the conceptual framework, the refinement of the VRLE, and the enactment settings and evaluation protocols. Implications for future iterations of the design experiments and for other research in the applications of VRLE in mathematics education are presented in this chapter. This chapter concludes with a refined conceptual framework to inform the future design of VRLEs, and implications and visions for future applications of the VRLEs in mathematics and technology education.

It is suggested that readers should bear in mind about the different type of learning activities reported in Chapters 6-8. If the reader does not desire to focus on the detail of each type of learning activity but instead focus on the overall outcomes of this study, then it is suggested that the reader can read the summary at the end of each of these three chapters before proceeding on to Chapters 9 and 10.
CHAPTER 2
DEVELOPMENT OF CONCEPTUAL FRAMEWORK

2.1 OVERVIEW

This chapter reports on Stage 1 of the study. In this stage, a conceptual framework was constructed to inform the design and evaluation of a VRLE, which aimed to facilitate the knowledge construction of 3D geometry concepts and processes.

To construct the conceptual framework, the literature review covers three threads of thought, which involve: (a) an understanding of the learner (e.g., cognitive development), (b) an understanding of domain knowledge (e.g., mathematical and geometrical knowledge), and (c) an understanding of computational ideas and paradigms (e.g., VR technology and microworlds) (Resnick, 1996). These three threads are interrelated and overlap to some extent.

Firstly, this chapter reviews some fundamental as well as contemporary issues about the learning in mathematics that served as the foundations of this research study. Issues such as the nature of mathematics, mathematics as meaning making activities, and learning theories are reviewed. Secondly, the content of 3D geometry including components, meanings and understandings, abilities and processes are reviewed. A review of the research literature on computer and 3D geometry is then presented. Lastly, a literature review focusing on technology in mathematics education, including philosophical issues, metaphors, interface design, instructional design issues, and a review on VR technology is presented. The outcomes of the literature review are then synthesised to form a conceptual framework to inform the design and evaluation of the VRLE.

2.2 LEARNING IN MATHEMATICS

This section explores two fundamental questions for this research study, which are: (a) What is mathematics? and (b) What is learning? Undoubtedly, the human cognition about mathematics and learning keeps evolving. Particularly, some views about what mathematics and learning are have changed dramatically in recent
years. The following review was set out to capture the current thinking about mathematics and learning, and draw them together to approach the learning in mathematics.

2.2.1 What is mathematics?

Mathematics as an expression of the human mind reflects the active will, the contemplative reason, and the desire for aesthetic perfection. Its basic elements are logic and intuition, analysis and construction, generality and individuality. Though different traditions may emphasize different aspects, it is only the interplay of these antithetic forces and the struggle for their synthesis that constitute the life, usefulness, and supreme value of mathematical science.

(Courant & Robbins, 1947, p. xv)

2.2.1.1 Influence of beliefs and philosophies about the nature of mathematics on teaching practices

To deal with the nature of mathematics is not the aim of this research, but it is fundamental to all research in mathematics education. As Vergnaud (1997) stated, the nature of mathematics is a controversial, high-level epistemological question, but one cannot simply ignore it. How a person perceives the nature of mathematics will affect his mathematical thinking and learning and vice versa.

Ernest (1999) identified two antithetical views about the nature of mathematics, namely absolutist and fallibilist. According to Ernest (1996), absolutist philosophies of mathematics are not concerned with the description of mathematics or mathematical knowledge. Instead, they are concerned with the epistemological project of providing rigorous systems to warrant mathematical knowledge absolutely. He went on to say,

The outcome is therefore a philosophically sanctioned image of mathematics as rigid, fixed, logical, absolute, inhuman, cold, objective, pure, abstract, remote and ultra-rational.

(Ernest, 1996, para. 5)

In contrast to this, fallibilist philosophies view mathematics as the outcome of social processes. Mathematical knowledge thus is understood to be fallible and eternally open to revision, both in terms of its proofs and its concepts. Consequently this view embraces as legitimate philosophical concerns the practices of mathematicians, its
history and application, the place of mathematics in human culture, including issues of values and education – in short – it fully admits the human face and basis of mathematics.

Much research based on Ernest’s dichotomy has found a relationship between teachers’ beliefs and philosophies about the nature of mathematics and their teaching practice (Becker & Pence, 1996; Beswick & Dole, 2001; Cobb, Yackel, & Wood, 1992; Ernest, 1996; Handal, Bobis, & Grimison, 2001; Howard, Perry, & Keong, 2000; Kagan, 1992; Lim, 1999; Perry, Howard, & Conroy, 1996; Perry, Howard, & Tracey, 1999; Shield, 1999; A. Thompson, 1992; Tracey, Perry, & Howard, 1998). These studies have found that teachers with absolutist views about mathematics utilise different curriculum, pedagogy, assessment and uses of technology from those adopted by teachers with fallibilist viewpoints about mathematics.

Handal, Bobis, and Grimison (2001), for example, found that mathematics teachers with absolutist beliefs and philosophies about mathematics favoured instructional practices such as drill and practice exercises, teacher’s exposition (lecture, knowledge transmission), gave paper-pencil tests to students, mainly used textbook resources and avoided teaching through real-life problems. They also found that mathematics teachers with fallibilist beliefs and philosophies about mathematics tended to teach in the ways of whole-class discussion of the real-life situation of the problem, used pair and small group work, fieldwork, open-ended students’ investigational projects, and computer programs. These teachers also required students to generate written reports from their open-ended mathematical investigations. In addition to this, they tended to make up their own problems and made use of a variety of resources.

Norton (1999) found that teachers’ beliefs about the nature of mathematics were linked to their computer usage in mathematics teaching and learning. Teachers with absolutist beliefs about mathematics tended to use computers in a calculational way or reject their use, while teachers associated with fallibilist beliefs favoured the conceptual use of computers in teaching.

---

2 In their study, Handal et al. (2001) classified absolutist viewpoints such as there is a “best way” to do a maths problem, maths is a set of rules and formulas, maths consists of unrelated topics, trial and error should not be allowed in solving a maths problem were classified as behaviourist orientations.
3 Handal et al. (2001) classified viewpoints such as there are many ways to solve a maths problem, the study of maths should be integrated with other subjects, students can create and do mathematics by themselves as constructivist orientations.
Ernest (1996) indicated that although there is no logical necessity for transmission-style pedagogy to be associated with an absolutist, objectivist epistemology and philosophy of mathematics, such associations often are the case. McGalliard (1983, cited in Thompson, 1992) reported that based on their dualistic conceptions of mathematics, teachers with absolutist-like viewpoints about mathematics acted in “authoritative” ways regarding the content of the lessons, emphasised a “right vs. wrong” stance, the use of rules without explanations or justifications, and the memorisation of answers and the taking notes in class. In a similar vein, Kesler (1985, cited in Thompson, 1992) reported in his doctoral study that two teachers with dualistic/absolutistic conceptions of mathematics taught in a manner consistent with their conceptions, whilst another two teachers with multiplistic conceptions taught in a range from “Strict authoritarian to an inquiry mode of instruction.”

Table 2.1 presents a synthesis of the findings from the review of the research that has investigated the relationship between teachers’ beliefs and philosophies about mathematics and teaching practices.

<table>
<thead>
<tr>
<th></th>
<th>Absolutist</th>
<th>Fallibilist</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Curriculum</strong></td>
<td>Mathematics as a collection of truths (Ernest, 1996)</td>
<td>Humanistic and connected values of school mathematics</td>
</tr>
<tr>
<td></td>
<td>Authoritarian or separated values of school mathematics (A. Thompson, 1992)</td>
<td>Link to complex real world application (Ernest, 1996)</td>
</tr>
<tr>
<td></td>
<td>Limited on simplified real life experience or textbook only (Handal et al., 2001)</td>
<td></td>
</tr>
<tr>
<td><strong>Pedagogy</strong></td>
<td>Focus upon the transmission of mathematical truth (Ernest, 1996)</td>
<td>Problem posing and solving</td>
</tr>
<tr>
<td></td>
<td>Teacher-focused models of teaching (A. Thompson, 1992)</td>
<td>Learner-focused models of teaching</td>
</tr>
<tr>
<td></td>
<td>Thing-centred or less social values add on</td>
<td>Person-centred and humanistic approaches to teaching</td>
</tr>
<tr>
<td></td>
<td>More didactic styles of teaching (Handal et al., 2001)</td>
<td>Active, creative, investigative and cultural activities.</td>
</tr>
<tr>
<td></td>
<td>Less emphases on group work</td>
<td>Promote collaborative learning (Handal et al., 2001)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Inquiry mode of instruction (A.</td>
</tr>
</tbody>
</table>
Lim (1999) indicated that absolutist views about mathematics were associated with mostly negative images of mathematics, which resulted in mathematics being perceived as “difficult, cold, abstract, and in many cultures, largely masculine” and "fixed, immutable, external, intractable and uncreative” or "a timed-test.” These negative views about mathematics, Lim claimed, are one of the major contributing factors to: (a) Low performance in mathematics and adult numeracy, (b) Low enrolments in mathematics and science in Higher Education, and (c) Shortage of mathematics and science teachers in schools.

As was noted in Section 1.2, virtual reality learning environments (VRLEs) can provide learners with highly motivational and engaging learning environments that allow them to freely navigate and manipulate in 3D space and engage in authentic mathematical problem solving. Therefore, it can be argued that any curriculum, pedagogy, assessment and technology use of VRLEs based on absolutist beliefs and philosophies about mathematics would seem not to be utilising the power and potential of VRLEs; instead of exploring new and potentially more powerful ways of learning, VRLEs based on absolutist viewpoints about mathematics would be merely amplifying many existing ways of teaching and learning 3D geometry. In contrast to this, any curriculum, pedagogy, assessment and technology use of VRLEs based on fallibilist viewpoints open up opportunities to harness the potential of VRLEs to go beyond amplification (Pea, 1985) and explore new and more powerful and enjoyable ways of teaching and learning 3D geometry.

2.2.1.2 Mathematics as meaning-making

So far, the discussion about mathematics has been investigated from the reference points of philosophy in general, and epistemology and ontology in particular. Mathematics, however can be investigated from a different reference
point, namely that of mathematics being perceived as an integral part of a single unified system for meaning-making.

This aspect about the nature of mathematics has its genesis in the field of semiotics (Ernest, 1997; Lemke, 2001). Semiotics is generally regarded as the study of signs and sign-using behaviour. It has its roots in two origins: the European tradition founded by F. de Saussure and the American tradition based on the works of C. S. Peirce and C. W. Morris (Andersen, 1990). Saussure’s tradition of semiotics was based around the constructs of signifier and signified. Pierce developed a triad of constructs consisting of representamen (sign), object and interpretant, and argued that all cognition is irreducibly triad, of the nature of a sign, fallible, and thoroughly immersed in a continuing process of interpretation (Halton, 1992). The two traditions can be synthesised into the triad below as shown in Figure 2.1.

![Figure 2.1 Peirce’s triad integrated with Saussure’s tradition of semiotics](image_url)

This triad presents the structural components of meaningful knowledge constructions (Osberg, 1997).

Modern semioticians argue that semiotics views the world and everything as a language (Gorny, 1994; Oliveira & Baranauskas, 2000) where language does not only refer to the spoken or written verbal languages, but also to all other types of language: dance, gestures, fashion, rituals of primitive tribes, music, sculptures, visual pictorial imageries, dreams, and so forth. Semiotics investigates all possible languages as a phenomenon of producing meaning. Its scope is vast, ranging from the study of the communication in non-human communities to the social study of ideologies: the study of moral, political, economic, religious and military codes (Oliveira & Baranauskas, 2000).

Cunningham (1992b), based on Peirce’s works, introduced four essential concepts of semiotics into educational practices:
1. The sign

2. Semiosis

3. Inference

4. Reflexivity

Each of these four essential constructs is discussed below in turn.

*The sign:*

A sign is defined as “something which stands to somebody for something in some respect or capacity.” In mathematical educational contexts, sign could be seen as the important issue of “representation.” Cunningham elaborated the semiotics triad (Figure 2.1) that a sign mediates between the object and its interpretant. However, a sign is not the object itself. He pointed out that a sign is only an incomplete representation of the object. A sign can only represent certain aspects of the object and in addition, it has aspects that are not relevant to the object. According to semiotics, signs were classified into three categories: icon, index and symbol. An icon stands for an object by resembling or imitating it, in particular, in a visual way such as a map visually resembles some characteristics of the territory. An index refers to the sign which is the effect produced by the object. An example is that a fire produces smoke, where the smoke becomes the index of the fire. The last category “symbol” refers to objects by virtue of a law, rule or convention. In this case, language could be a prototype of symbols.

In Cunningham’s (1992b) opinion, sign theory offers a significant advantage into the analysis of mind, which is to consider systems of signs as acting like a code for some system of objects. The systems of codes are used to structure our experience. Codes are not equivalent to the objects they represent, but it is possible to specify rules for those equivalences in that the rules are called syntax, which allows the manipulation of signs in a potentially indefinite number of ways. The implications, as indicated by Cunningham, are that we don’t finalise a single correct way of manipulating syntax and don’t limit our conception of the structure of signs to any particular type of linguistic model or only to linguistic models.
Semiosis:

Semiotics is the science of signs and the structure of signs within which meaning emerge. The process by means of which these sign structures are built up is called semiosis (Cunningham, 1984). Cunningham pointed out that two benefits could be brought into education if we turn the focus of education onto semiosis. The first is that a focus on the semiosis or construction process will stress the multiple ways in which we can know things, the variety of codes that we can use to represent experience. The second is that because a semiosis approach encourages multiple perspectives and representational types, we may be able to change the mindset of teachers and students about what is supposed to go on in school such as overcoming and/or going beyond the dualistic view of knowledge as either right or wrong.

Inference:

There are three types of inference in semiotics. They are induction, deduction and abduction, which are in contrast to traditional types of inference that have only the first two of the three. Cunningham (1992b) presented a figure (adopted from Deely, 1986) that shows the interrelationship among these three modes of inference (Figure 2.2).

Figure 2.2 Types of inference
(Figure adopted from Cunningham, 1992b, p.185)

In the figure above, Cunningham explained that semiosis is a process of applying signs to understand some phenomenon (induction), reasoning from sign to sign (deduction), and/or inventing signs to make sense of some new experience (abduction). In inductive and deductive reasoning, it is possible that we construct our cognitive structures to mirror the environmental structures, and then tend to equate them with “reality” and seldom question their inevitability. Abductive reasoning or thinking, on the other hand, constantly provides us a way to question, invent and alter our beliefs (cognitive structures) about reality when we confront some experience not accounted for by our existing beliefs. Hence, abductive
thinking reveals that inquiry is to be an ongoing process of life and challenges the assumption held by absolutists that the objective knowledge of few privileged people is unchallengeable. Abductive inference is valuable in education and should be encouraged and promoted during teaching and learning. However, it is often ignored in most educational research (Cunningham, 1992b).

Reflexivity:

Cunningham (1992b) stated that reflexivity at its core is the reflection about semiosis. He saw reflexivity as a reflection on our reflections, thinking about our thinking process, knowing how we know. In that explanation, reflexivity could be seen as a form of metacognition (Osberg, 1997). However, it has a much broader meaning than metacognition; reflexivity also reflects others’ thinking about thinking. Cunningham (1992b) gave an example of reflexivity in school curricula. He stated that a course in history would emphasize not simply history, but how historians know, the sign structures or systems of belief that historians use in their discipline.

To the benefit of employing reflexivity in education, Cunningham pointed out that:

An important component of reflexivity is the development of an informed scepticism, a healthy distrust of things at their face value and an openness to explore new interpretations, new sets of beliefs. ….. In essence, it is an attitude to actively seek out opportunities to look at things in new ways, to make the familiar strange and the strange familiar, to consider and evaluate perspectives other than your own, to admit that the world that seems to make so much sense is not so certain after all.

(Cunningham, 1992b, p.189)

Thus, Cunningham (1992b) suggested making reflexivity a primary educational goal by giving a reasoning circle: Reflexivity is reflection about semiosis. Semiotics is the study of semiosis. And educational goal is to turn our students into semioticians.

In addition to Cunningham’s educational semiotics, another seminal thinker whose work in semiotics has much relevance for the study of mathematics as meaning-making is Lemke (1998, 2001). Lemke (2001) stated that a semiotic perspective helps us understand how natural language, mathematics and visual representations form a single unified system for meaning making in which mathematics extends the typological resources of natural language to enable it to
connect to more topological meanings made with visual representations. He argued that mathematics is an on-going process of “semiosis” or meaning making of mathematical symbols (signs), for those signs form special relationships. Based on Peirce’s semiotics triad, Lemke (2001) interpreted that in mathematics, there is a material signifier or “representamen” (R) that we encounter on page, the “object” (X) or signified which could be a real object, a concept, a quantity, an abstraction or another sign, and the “interpretant” (I) which connects the R to X. Lemke pointed out was there has to be some system of interpretance (SI) in which the R gets interpreted as X. He restated Peirce’s idea that we have a sign when something (R) stands for something else (X) for somebody in some context (SI). Lemke identified two types of semiotics in mathematics, which he termed typological and topological semiotics.

Typological semiotics represents meanings by types or categories such as spoken words, written words, mathematical symbols and chemical species. They are discrete, point-like and distinctive signs. In contrast to this, topological semiotics makes meaning by degree such as size, shape, position, colour spectrum, visual intensity, pitch, loudness and quantitative representation in mathematics. There is no continuum between typological signs (e.g., word “GOOD” and “GOAD”, variable “X” and “Y”, atomic species “Carbon” and “Nitrogen”), but there is a continuum in topological signs (e.g., the acoustical spectrum of sounds, between any two real numbers). The differences between Topological and Typological semiotics identified by Lemke are illustrated in Figure 2.3.

<table>
<thead>
<tr>
<th>Topological Semiosis</th>
<th>Typological Semiosis</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Meaning by degree</td>
<td>• Meaning by kind</td>
</tr>
<tr>
<td>• Quantitative difference</td>
<td>• Qualitative distinction</td>
</tr>
<tr>
<td>• Gradients</td>
<td>• Categories</td>
</tr>
<tr>
<td>• Continuous variation</td>
<td>• Discrete variants</td>
</tr>
</tbody>
</table>

\[
\text{quant-X} \quad \text{quant-R} \\
\text{SI(X,R)}=0
\]

\[
\text{type-X} \quad \text{type-R} \\
\text{SI(R} \rightarrow \text{X)}
\]

*Figure 2.3 Topological versus Typological Semiosis*  
(Figure adopted from Lemke, 2001)
Lemke (2001) stated that in general, mathematical expressions are constructed by typological systems of signs, but the values of mathematical expressions can in general vary by degree within the topology of the real number. He pointed out that students have a great deal of trouble typically in understanding functional notation (typology) and its meaning in terms of quantitative co-variation (topology). However, they can be greatly aided by employing the topological strategies such as graphs and other visual representations. For example, when asked to order the size or ratio of a given set of fractions 13/19, 11/17, 4/6, 9/13, there is no simple way to tell from these typological representations except by performing the divisions. But their relationships can be easily understood if a graph or diagram of those ratios is visually presented.

In addition to his semiotic perspective of mathematical meaning-making, Lemke also took a view that mathematics is a system of related social practices, a system of ways of doing things. He argued that mathematics is not in the “how-to” handbook or textbook, but is embedded in writing about physics, engineering, accounting, surveying, and so forth; Meaning-making is always a material process as well as a social semiotic practice. Therefore, he presented a social semiotics account of mathematics in that:

Social semiotics seeks to explicate how we make meaning with all the resources at our disposal: linguistic, pictorial, gestural, musical, choreographic, and most generally actional.

(Lemke, 1998)

Social semiotics tries to look at meaning-making as an aspect of a whole activity, i.e. some on-going process of inter-related doings in which meaning-making plays a key role. ....... People make mathematical meanings as an integral part of activities such as building and operating a power plant, sending an expedition to Mars, creating global climate models, or trying to find more efficient compression algorithms for digital video. .......

(Lemke, 2001)

And in such a notion of social semiotics, he proposed that we should integrate multiple semiotic resources in classroom learning with both actional-typological and actional-topological resources. Moreover, Lemke (2001) offered three semiotic strategies on the teaching of mathematics:
1. We should at all stages make repeatedly explicit for students just how mathematical expressions and mathematized visual representations can be partially but never completely translated into natural language statements and questions, and into one another’s forms, and how natural language can be partially translated into mathematical symbolisms, both algebraic and geometric;

2. We should embed the uses of mathematics in the application contexts that either gave rise to the mathematics historically, or are among the most typical or interesting uses of the mathematics today; and

3. Where possible, we should expose students to real, out-of-school settings and practices, in which real people are using mathematics for practical (and theoretical) purposes as part of normal activities in a actual institutional contexts, including occasionally pure research mathematics; students should observe the practices, simulate them in simplified fashion, gain a sense of their real contexts and complexity, and handle the important artifacts associated with the practices (such as seeing real technical and mathematical research papers, journals and books, operating computers, etc.)

(Lemke, 2001)

2.2.1.3 Discussion

In this section, two aspects about the nature of mathematics were identified and their impacts on the learning and teaching of mathematics were discussed. The first aspect took a philosophical viewpoint about the nature of mathematics and focused on the ontological and epistemological aspects of mathematics. This analysis revolved around a discussion about the fallibilist and absolutist dichotomy in mathematics. The second aspect of the nature of mathematics reviewed took a semiotic viewpoint about mathematics; it thus reviewed the research literature that looked at mathematics as a meaning-making endeavour.

The review of the literature that focused on the absolutist-fallibilist dichotomy indicated that the possession of absolutist viewpoints about the nature of mathematics most often results in the generation of mathematics curriculum, pedagogy, assessment and uses of technology that does not utilise the power and potential of VRLE’s to enable learners to explore new and potentially more powerful ways of learning and knowing mathematics. In contrast, the review of the research literature indicated that mathematics curriculum, pedagogy, assessment and uses of technology based on fallibilist viewpoints were more likely to utilise the power and
potential of VRLE’s to enable learners to explore new and potentially more powerful ways of learning and knowing mathematics. Furthermore, they also were more likely to produce much more positive attitudes and beliefs about learning mathematics. In addition, fallibilist viewpoints about the nature of mathematics and constructivists practices were found to be analogous.

Further implications of this within the contexts of this research study are that if an absolutist philosophy about the nature of 3D geometry was to underlie the VRLE being developed, then what the children learnt about the nature and discourse of the subject matter would be inconsistent with the fallibilist viewpoints influencing the philosophies and practices of most participants in the mathematics education community of practice. Also, it is highly unlikely that the children would have positive learning experiences if the VRLE was based on absolutist viewpoints. Therefore, in this study, the VRLE being designed will attempt to facilitate the construction of a viewpoint about the nature and discourse of three-dimensional geometry that has more in common with a fallibilist viewpoint than an absolutist viewpoint.

A semiotics perspective in which mathematics is perceived as meaning-making endeavour covers important issues such as natural languages and visual representations in the learning and teaching of mathematics. Through this perspective, we perceive that mathematics is situated or located in a variety of sign systems in which we are doing things, creating ideas, solving problems to make connections among signs and make meanings of our living community and world. Furthermore, aside from identifying this nature of mathematics, a semiotic analysis of mathematics that focuses on sign structures and their emissions of meanings, inferences in the process of semiosis, ways of inquiring (deductive, inductive and abductive), and reflexivity upon our beliefs, all provide powerful insights into the nature of mathematics and also about how mathematics can be taught, learnt and communicated. And notably, this semiotic perspective is also analogous to fallibilist viewpoints about the nature of mathematics in that both of them clearly imply that mathematical knowledge is subject to revision and is challengeable. The implications of semiotics review for the present research study are that multiple semiotic resources such as Lemke’s (2001) framework of typological and topological semiotics, and Peirce’s categories of signs (icon, index, and symbol) should be
integrated into VRLE in order for learners to perceive and construct mathematical meaning within it. And the VRLE shall encourage learners with a healthy inquiring and creative inventing of multiple ideas and ways of doing things in which the abductive reasoning and reflexivity are applied throughout the process of semiosis.

2.2.2 What is learning?

There are many theories (a review of the literature revealed at least 50 major theories) that describe learning (see Kearsley, 2001). However, a review of the research literature along the evolution of learning theories (i.e., from behaviourism to cognitivism to constructivism) indicates that constructivism is the category of learning theory currently dominating the field of mathematics education. In addition, during the 1990s a notion similar to constructivism about learning emerged and named constructionism, which has a great impact on this research study. The following sections will focus on the review of constructivism and constructionism to approach the nature of learning.

2.2.2.1 Constructivism as a learning theory

Constructivism had its genesis in the research conducted by Piaget in the 1920s (Kearsley, 2001). It gained major impetus in the USA and the rest of the English-speaking world during the 1960s mainly due to the work done by Bruner (1960) and Flavell (1963). Duffy and Jonassen (1992b) summarised the major components of these constructivist theories, highlighting the work of:

1. Bednar et al. (1992) who argued that learning outcomes should focus on the process of knowledge construction and the development of reflexive awareness of that process, learning goals should be determined from authentic tasks with more specific objectives resulting from the process of solving the real-world task, the processes of learning should be modelled and coached for students with unscripted teacher responses, and learners should be able to construct multiple perspectives on an issue, that is, see an issue from different viewpoints;

2. Cunningham (1992a) who argued that the goal of instruction is not to assure that individuals know particular things but rather to show them how
to construct plausible interpretations of those things, using the tools that have been provided or developed in collaboration with them;

3. Perkins (1992) who emphasised the “active learner” component of constructivism, which means not just that the learner is an active processor of information but more importantly that the learning elaborates upon and interprets the information;

4. Spiro et al. (1992a) who emphasised that context is an integral part of meaning, thus learning should be working with a variety of contexts or examples; and

5. The Cognition and Technology Group at Vanderbilt (CTGV) (1992b) who emphasised the importance of situating learning in a macro-context in which the learner can engage in sustained exploration.

One criticism of “constructivist” theories that has emerged in recent years is constructivist theories imply that teachers should never tell students anything directly but, instead, should always allow them to construct knowledge for themselves (Bransford, National Research Council (U.S.). Committee on Developments in the Science of Learning, & National Research Council (U.S.). Committee on Learning Research and Educational Practice, 2000). This claim has been refuted by many proponents of constructivism such as Bransford et al. (2000) and Driver et al. (1994). Bransford et al. argued that there are times, usually after people have first grappled with issues on their own, that “teaching on telling” can work extremely well. However, teachers still need to pay attention to students’ interpretations and provide guidance when necessary. Driver et al. also noted that teachers have many roles to play within constructivist leaning environments such as guide, co-problem solver, and facilitator. However, in their opinion, the major role was that of a more knowledgeable peer or guide inducting her learners into a community of practice.

To talk about instructional strategies from constructivist perspectives may seem odd because most major principles of instructional design have been strongly rooted and derived from Skinnerian psychology (behaviourism) and Gagné’s conditions of learning (cognitivism) (Dick, 1992). According to constructivist theory, knowledge is constructed by active learners in authentic and meaningful contexts. Learning thus is learner centred, experiential, contextualised, complex and
constructed through discourse and negotiation among peers, which differs much from the assumptions in behaviourist and cognitivist that learning can be divided into tasks and ordered in a hierarchy. Therefore, the constructivist strategies for instructions tend to focus on different constructs such as:

1. Learner control: An “active learner” perspective specifies that learners control the learning pace but not in linear or any prescriptive modes. Information can be given directly, however, learning should be able to “go beyond the information given” (Perkins, 1992).

2. Generative learning environments: These are based on the theoretical framework that emphasises the importance of anchoring or situating instruction in meaningful, problem-solving contexts that allow one to simulate in the classroom some of the advantages of apprenticeship learning. In mathematics education, it echoes the National Council of Teachers of Mathematics’s (NTCM) suggestion for changes in classroom activities include more emphasis on complex, open-ended problem solving, communication, and reasoning; more connections from mathematics to other subjects and to the world outside the classroom; and more uses of powerful computer-based tools (Cognition and Technology Group at Vanderbilt, 1992b).

3. Multiple knowledge representations, collaborative learning and meaning negotiated from multiple perspectives: Constructivists argue that one must have available a diverse repertoire of ways of constructing situation-specific understandings (Spiro, Feltovich, Jacobson, & Coulson, 1992b). And these constructed understandings do not mean that any interpretation is as good as any other, but rather that there is a social negotiation of meaning (Bednar et al., 1992; Cognition and Technology Group at Vanderbilt, 1992a; Duffy & Jonassen, 1992b). Therefore, strategies of learning collaboratively in a group to evaluate and negotiate various understandings are employed by constructivists.

4. Goal-free evaluation of learning: Because constructivists see the importance of the learning process rather than the learning outcome, constructivists propose that a constructivist learning environments should be more goal free, assess knowledge construction in real-world contexts,
and require authentic learning tasks that represent multiple perspective and viewpoints (Guba & Lincoln, 1989; Jonassen, 1992). Assessment methods must take into account the zone of proximal development and target both the level of actual development and the level of potential development (Vygotsky & Cole, 1978).

5. Scaffolding: With appropriate adult help, children can often perform tasks that they are incapable of completing on their own. With this in mind, scaffolding--where the adult continually adjusts the level of his or her help in response to the child's level of performance--is an effective form of teaching. Scaffolding not only produces immediate results, but also instils the skills necessary for independent problem solving in the future (Vygotsky & Cole, 1978).

2.2.2.2 Constructionism as a learning theory

During the past 15 years, these constructivist notions have been further developed by Papert and his Epistemology and Learning Research Group in Massachusetts Institute of Technology (MIT). Their ideas have been encapsulated by the term, constructionism. Papert (1991) gave a simple catchy version of the idea of constructionism – “learning-by-making” that encapsulates the whole philosophy of constructionism. Kafai and Resnick (1996) elaborated that:

Constructionism is both a theory of learning and a strategy for education. It builds on the “constructivist” theories of Jean Piaget, asserting that knowledge is not simply transmitted from teacher to student, but actively constructed by the mind of the learner. Children don’t get ideas; they create ideas. Moreover, constructionism suggests that learners are particularly likely to make new ideas when they are actively engaged in making some type of external artifact – be it a robot, a poem, a sand castle, or a computer program – which they can reflect upon and share with others. Thus, constructionism involves two intertwined types of construction: the construction of knowledge in the context of building personally meaning artefacts.

(Kafai & Resnick, 1996, p. 1)

Constructionism resonates with the most important aspects of constructivism about learning, and extends and emphasises several perspectives such as:
1. Most constructivist theories describe knowledge acquisition in purely cognitive terms while constructionism sees an important role for affect. In this view, constructionism argues that learners are most likely to become intellectually engaged when they are working on personally meaningful activities and projects. It sees that forming new relationships with knowledge is as important as forming new representations of knowledge.

2. Constructionism also emphasises diversity. It recognises that learners can make connections with knowledge in many different ways. Constructionist learning environments encourage multiple learning styles and multiple representations of knowledge.

3. Constructionism suggests a strong connection between design and learning. It asserts that activities involving making, building, or programming – in short, designing – provide a rich context for learning.

4. In constructionism, learning is not only a matter of “individual thinker” but also a process in social communities. Community members act as collaborators, coaches, audience and co-constructors of knowledge. Constructionism promotes knowledge-building communities and computer support for collaborative learning, which can be in a variety of communities such as classroom, urban and virtual communities.

5. Constructionism argues that most current curriculum, including mathematics is still pencil and paper-based activities. However, as new computational tools and media become available, we must rethink the content of the curricula. For example, the world is full of dynamic and complicated systems such as biological, technological and social systems. It is impossible to analyse these systems by traditional pencil and paper tools but now can be studied with new computational tools such as computers. (Kafai & Resnick, 1996).

In addition to “learning by making,” constructionists also put a strong emphasis on the use of technology in learning, in particular, the computer programming and networks. Computer programming and network established a new learning paradigm and enabled constructionist’s practices. For instance, learners can design and construct a simplified or complex system by computer programming, and participate
in a virtual community to share and help each other through a computer network. This new learning paradigm can reflect social, linguistic, communicational, and biological and many other settings in the real world. Constructionism thus creates many powerful ideas in learning, such as the “Microworlds” metaphor (see Section 2.4.3 for detailed review), new ways of thinking, doing, and making that can favour and engage learners into deep learning.

2.2.2.3 Discussion

Constructivist and in particular constructionist theories about learning and their associated strategies for instructions reviewed in this section have implications for this research study. The VRLE being developed in this research study is incorporating many of the powerful ideas derived from constructivism and constructionism to facilitate the learning of 3D geometry. The powerful ideas being integrated in VRLE include:

1. An open and generative VRLE in which learners have full control of using multiple resources (knowledge representation, media, tools) for expressing and creating ideas from multiple perspectives (3D viewpoints, ways of doing or solutions).

2. An authentic and convivial VRLE in which learners are engaged in personal meaningful and enjoyable construction activities.

3. A collaborative and scaffolding VRLE in which learners are supported and helped by advanced peers and structured links (hypertexts), and also those who interact within on-line virtual community.

4. A Constructionist Microworlds-paradigmatic VRLE in which learning of 3D geometry emerges from designing and programming of a coherent domain of objects and activities of complex systems.

2.2.3 Synthesis

Section 2.2 has endeavoured to uncover the meaning of the term: “learning in mathematics.” The review has focused on the following two themes:

1. What is mathematics?

2. What is learning?
From this review of the literature, an initial conceptual framework (presented in Figure 2.4) to inform the research and design activities of this research study has been developed.

There are three interlinking components within this initial conceptual framework:

1. Fallibilist perspective about the nature of mathematics;

2. Semiotics perspective about mathematics as meaning-making activity; and


These three components contribute to the meaning of “learning in mathematics” from philosophical, semiotic, and psychological viewpoints respectively.

These three components share common ideas about the nature, source, developmental process and methods of knowing of mathematics. For example, when focusing on the nature and source of mathematics, there seems to be a consensus that mathematics is a product of human construction within the meaning-making activities of the world, and mathematical knowledge is challengeable and must be constantly qualified when presented in different systems of signs or situated in diverse social cultural contexts. Similarly, there seems to be a consensus in the
developmental processes and ways of learning mathematics that multiple semiotic resources such as topological (e.g., visual graph, sound, colour, size, position, time, spatial continuum), typological (e.g., written word, spoken language, symbol, chemical species), and social-actional (e.g., group playing, building, cooperating, discussing, negotiating, creating, designing) resources should be integrated into learning activities and at learners’ disposal. Thus learners can have multiple perspectives and solutions to reflect upon and then get better understandings or inventions (abductive creations) of mathematics. These common ideas reveal that the three components of the initial conceptual framework are consistent.

From this initial conceptual framework, it has been possible to generate the following initial set of principles to inform the design of the VRLE from a synthesis of the literature review about learning in mathematics from the three components identified in Figure 2.4 above:

1. The authoritative or absolutist views of mathematics should be avoided. The views of mathematics as revisable, and where there are multiple solutions to be found in solving mathematical problems should be encouraged in VRLE.

2. For satisfying the preceding principle, the VRLE should be an open and generative learning environment. Multiple semiotic resources (e.g., topological, typological and social-actional) should be integrated into VRLE to enable learners to have multiple interpretations and perspectives for mathematical meaning-making and going beyond the information giving.

3. When employing multiple semiotic resources in VRLE, the sign structures and their meaning emission capacities should be analysed and considered in terms of icon, index, symbol etc. This can reduce the unwanted interferences of signs and scaffold learners to easily use or immerse into VRLE.

4. While employing an abundance of resources, VRLE should also provide resources for learners to express ideas and communicate and collaborate with other learners (e.g., a programming language, knowledge forum).
The principles listed above are derived from the initial conceptual framework. In the following sections of this chapter, the research literature about 3D geometry, technology in education, and VR technology in educational contexts are reviewed. Based on the review of the research literature in these areas of research, modifications are then made to the initial conceptual framework.

2.3 3D GEOMETRY

In most mathematics curriculum, syllabus and support documents, geometry is categorised into chapters such as three-dimensional shapes, two-dimensional shapes, symmetry and tessellations, position and orientation, geometry on a sphere, deductive geometry, and analytical geometry etc. (see Anderson, 1994; Cooper, 1986; Queensland. Dept. of Education, 1987). These categorisations are mostly based on Euclidean geometry. However, geometry can be perceived as a study of the space (see for example the Space Strand in Queensland mathematics syllabus, Queensland Studies Authority, 2004) we are living in rather than only a hierarchical subset of Euclidean concepts and processes. Therefore, the literature on 3D geometry being reviewed here will not address the issues of teaching and learning of 3D geometry from a Euclidean viewpoint. Instead, the review of the literature addresses these issues from viewpoint of an object living and carrying out activities within a 3D space. From this viewpoint, explorations within 3D geometry are concerned with moving, positioning, orientating, constructing, building, and communicating etc. in space (e.g., in real space or in a virtual space of a VRLE). This will naturally and/or virtually include most topics in the school geometry while at the same time going beyond the Euclidean geometry to address topics derived from projective and topological geometry.

The following review focuses on the components, development and understanding, and teaching and learning of 3D geometry.

2.3.1 Components of 3D geometry

This section is concerned with the aspects of 3D geometry that we possibly encounter in daily life. Taking inspiration from semiotics, thinking within the domain of 3D geometry can be categorised into three components: external material
world (objects), internal spatial abilities (interpretants), and the communications (signs) that mediate between them.

The external material world component of 3D geometry concerns the real world objects and their properties and behaviours. For instance, a field of grass, buildings in the city, rocks of mountains, a rotating planet, and the growth of trees etc., are included in the component. The internal spatial abilities component of 3D geometry addresses the human capacity of perceiving and knowing about the external material world. Abilities such as visualisation, orientation, and eye-motor coordination etc. are included within this domain. The communication component of 3D geometry involves largely the spoken and written languages that are currently agreed upon by worldwide communities, and also non-language or topological communications such as visual and body kinaesthetic ways of communication.

2.3.1.1 External material world

The natural objects in our material world are mostly ill structured and complicated. However, many of them seem to obey some subtle rules that can be expressed mathematically (McCombs, 2000). For examples, a chambered Nautilus shell is consistent with Fibonacci’s golden ratio 1.618033… (Figure 2.5), and fern leaves can be represented in L-System’s fractal programs (Prusinkiewicz, Lindenmayer, & Hanan, 1990) (Figure 2.6).

Figure 2.5 Chambered Nautilus shell and golden ratio
(Figures derived from: http://www.math.unc.edu/Faculty/mccombs/math18/fib.html)
Figure 2.6 Fern leaf and L-System fractal program
(Figures derived from: http://www.home.aone.net.au/byzantium/ferns/fractal.html)

On the other hand, the basic human creations of objects (shapes) are relatively simplified and based much on Euclidean geometry. These basic objects can be categorised in terms of a space conception “dimension.” The table below lists a collection of primitive objects ranging from 1D to 3D.

Table 2.2

<table>
<thead>
<tr>
<th>1 dimensional object</th>
<th>2 dimensional object</th>
<th>3 dimensional object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point</td>
<td>Line</td>
<td>Sphere (ball)</td>
</tr>
<tr>
<td></td>
<td>▪ Curve, arc</td>
<td>Box, cube</td>
</tr>
<tr>
<td></td>
<td>▪ Triangle, rectangle, square, circle, ellipse, rhombus (diamond), trapezium, parallelogram, polygons (pentagon, hexagon, heptagon, octagon, nonagon, decagon, hendecagon, dodecagon...)</td>
<td>Cylinder (can)</td>
</tr>
<tr>
<td>Face</td>
<td></td>
<td>Cone</td>
</tr>
<tr>
<td></td>
<td>▪ Triangular, rectangular (box), square, pentagonal, hexagonal...</td>
<td>Prism</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▪ Triangular (tetrahedron), square (pyramid), pentagonal, hexagonal...</td>
</tr>
</tbody>
</table>

The 1D and 2D primitives are building blocks of 3D primitives, and moreover, they are all building blocks of almost any complex real world object.

Both natural and artificial primitive objects have their properties and behaviours. Certain properties such as length, width, depth, surface, edge, angle, and vertex etc. have been termed formally in mathematics community. Other properties of concern are objects’ colours, transparency, and texture. A complex colouring model combined with environmental lights in the real world may comprise diffuse, emissive and specular colours, shininess, ambient intensity, and transparency.
Colours and texture can provide visual discriminations of objects and create different type of material sense (e.g., glass, metal, and tile). Transparency can provide a look through effect that may be helpful for the projective and perspective geometrical perceptions.

In regard to behaviours of objects, they are seen as being of great importance to the learning of 3D geometry because they reflect the dynamic nature of the real world. Behaviours can be a rolling ball, a spinning planet, an animation of growing tree, the morph of an object (topology of object), or simply the gradient change of objects’ properties such as colours. In short, behaviours can be manifested by changes of position, orientation and properties of objects, and also the transformation of objects. For the design of behaviours, the concept of “time” can be used to control the speed of the behaviours.

In addition to those physical objects, there are some environmental objects that are important to the perceptions of physical objects such as background (e.g., sky, ground), gravity, light and sound etc. Environmental objects don’t occupy any space; however, they provide the environmental cues that would be valuable for the semiosis of 3D geometry. For example, light and sound produce stereo perceptions of objects so that the depth or “near and far” can be perceived.

There are more objects, properties and behaviours in the real world than those discussed above. However, it is considered that those discussed above have enough potential to provide the building blocks for learners to construct much knowledge of 3D geometry.

2.3.1.2 Internal spatial abilities

With the 3-dimensional, complex and dynamic nature of the material world around us, how do we cope with it? What inherent potentials and abilities do human beings have for living in the space?

From an early age, babies lying in a cradle have the ability to wave their hands in the air trying to grab the toys hanging above them. Later on, they develop abilities that allow them to freely move or jump into various positions, catch or reach objects in different distances, and discriminate objects from diverse orientations or directions. This human ability to operate in 3D space has interested psychologists
since the 1920s (McGee, 1979), and has been labelled by McGee with the term “spatial ability.”

McGee’s (1979) review of the literature described the emergence of two generally accepted components of spatial ability:

1. Spatial visualisation – the ability to mentally rotate, manipulate, and twist two-and three-dimensional stimulus objects.

2. Spatial orientation – the comprehension of the arrangement of elements within a visual stimulus pattern; the aptitude for remaining unconfused by the changing orientations in which a figure may be presented; the ability to determine spatial relation with respect to one’s body.

Frostig and Horne (1964) generated a more detailed analysis of spatial ability. Based on Frostig and Horne’s work, Hoffer (1977) identified seven spatial perception abilities, which were relevant to academic development and especially the learning of geometry:

1. Eye-motor coordination: the ability to coordinate vision with movement of the body.

2. Figure ground perception: the visual act of identifying a specific figure (the focus) in a picture (the ground).

3. Perceptual constancy: the ability to recognize that an object has invariant properties such as size and shape in spite of the variability of its impression as seen from a different viewpoint.

4. Position in space perception: the ability to determine the relationship of one object to another object and to the observer.

5. Perception of spatial relationships: the ability to see two or more objects in relation to oneself or in relation to each other and is closely related to position-in-space perception for some tasks.

6. Visual discrimination: the ability to distinguish similarities and differences between objects.
7. Visual memory: the ability to recall accurately an object no longer in view
and then to relate its characteristics to other objects either in view or not in
view.

After 1980, the focus of internal spatial abilities seemed to shift in
terminologies but the content remained analogous. Gutiérrez (1996) explored spatial
abilities in the literature in search of a framework for visualisation in 3-dimensional
geometry, and found many similar constructs such as:

1. “Mental image” used by neuroscientist Kosslyn (1980) to depict that a
quasi-picture created in the mind from memory without any physical
support;

2. “Mental imagery” as the occurrence of mental activity corresponding to
the perception of an object, but when the object is not presented to the
scene again (Lean & Clements, 1981);

3. “Visual imagery” as mental imagery which occurs as a picture in the
mind’s eye (Lean & Clements, 1981); and

4. “Spatial ability” as the ability to formulate mental images and to
manipulate these images in the mind (Lean & Clements, 1981).

Bishop (1983) identified two abilities in visualisation: VP (visual processing of
information) and IFI (interpretation of figural information). VP includes the
translation of abstract relationships and non-figural data into visual terms, the
manipulation and extrapolation of visual imagery, and the transformation of one
visual image into another. IFI involves knowledge of the visual conventions and
spatial vocabulary used in geometrical work, graphs, charts, and diagrams of all
types, and the reading and interpreting of visual images, either mental or physical, to
get from them any relevant information that could help to solve a problem.
Yakimanskaya, Wilson, and Davis (1991) described two levels of activity in “spatial
thinking,” the creation of images and their manipulation or use, as two closely
interrelated processes.

Gutiérrez (1996) considered those abilities reviewed above overlapped in
terms of general abilities useful in many ordinary life or professional activities, or as
specific to mathematized contexts. He then further formulated a framework
specializing for the visualisation in 3D geometry that integrated the concepts and terminologies discussed above. The framework consists of four elements:

1. Mental images: any kind of cognitive representation of a mathematical concept or property by means of visual or spatial elements.

2. External representation: any kind of verbal or graphical representation of concepts of properties including pictures drawings, diagrams, etc. that helps to create or transform mental images and to do visual reasoning.

3. Process of visualisation: mental or physical action where mental images are involved. It consists of “visual interpretation of information” to create mental images and “interpretation of mental images” to generate information.

4. Abilities of visualisation: depending on the characteristics of the mathematics problem to be solved and the images created, students should be able to choose among several visual abilities such as figure-ground perception, perceptual constancy, mental rotation, perception of spatial relationships, and visual discrimination etc.

Despite Gutiérrez’s attempt at synthesisisation, it seems that there is still no agreement about the use of the terminologies in this field. However, it can be seen that the internal spatial ability interacts closely with other components of 3D geometry: to mentally create, manipulate, comprehend the images of external material world, and to interpret the mental images through communications. These notions are reviewed and discussed in the next section.

2.3.1.3 Communication of 3D geometry concepts

It was argued that geometry could be a starting point for all kinds of mathematical learning (Goldenberg, Cuoco, & Mark, 1998), and a vehicle for developing other mathematical concepts (Bruni & Seidenstein, 1990). Therefore, the communication of 3D geometry concepts also involves the communication from the other main strands of mathematics such as number, measurement, algebra, and space/geometry. The communication being presented here mainly focuses on the space/geometry strand.
This communication component of 3D geometry acts as signs and mediates between the external geometrical world and internal mental world. Languages including the written and spoken, formal and informal are the most common means of communication. However, there are other means of communication to aid the process of communication. These may include visual drawings, diagrams, pictures, computer images, VR spaces and body movements etc.

Anderson (1994) identified some important aspects of language used in space/geometry strand covering position, plane shapes, 3D shapes, coordinates, line, symmetry and angles, spanning from Years 1 to 7. These aspects of language are summarised in Appendix 1. The set of vocabulary presented in Appendix 1 is very useful for representing the geometrical objects and for describing their attributes and relationships.

When describing the spatial relationships, the construct of a “reference frame” is an important underlying concept. Sachter (1991) explained that the reference frame is a construct, which underlies projective and Euclidean space and is an important factor in the performance of various spatial and developmental tasks. A “reference frame” is a systematic representation of spatial relations among objects, which provides a set of coordinates for expressing transformations of such relations. There have been some classifications about the “frame of reference”. For example, Olson and Bialystok (1983) identified four general classes of referents: ego, observer, object, and environment. Some just simply classified as egocentric, exocentric, and/or the combination of the two (McCormick, Wickens, Banks, & Yeh, 1998; Salzman, Dede, & Loftin, 1999). In this research study, three main reference frames: egocentric, fixed object, and coordinate systems (Darken, 1996) have been adopted. The egocentric frame of reference can only locate objects in the environment relative to the body. The fixed object reference frame locates objects relatively according to a particular object such as a tree or landmarks. The coordinate system of reference (e.g., the Cartesian coordinate system) is more abstract and locates objects to an absolute position not relevant to body or any fixed objects. Geometrical language can be associated with each frame of reference. For example, relative descriptors such as forward, back, turn left or right, above, below, beside, front and behind, and north, east, west, south etc. can be used in egocentric
and fixed reference frames, while absolute terms such as $x$, $y$, and $z$ axis etc. can be used in coordinate system of reference.

Yakimanskaya et al. (1991) found that a variety of frames of reference are used to solve graphic problems, and constant transition from one frame of reference to another enriches and influences each other. However, they pointed out that the predominant use of some one particular frame of reference (most often the human body) often impedes successful problem solving particularly in descriptive geometry. They suggested that it is necessary to use several frames of reference simultaneously. Sachter (1991) also pointed out that it is important to know how, why, and when a particular reference frame is chosen in problem solving.

Language (both spoken and written) serves as a rich resource to link between external geometrical objects and internal geometrical concepts personally and interpersonally. However, they are typological (Lemke, 2001) and can only partly interpret and represent meanings between external and internal geometry. Topological communications such as visualisations and kinaesthetic body movements on the other hand provide supplements toward a more complete communication.

Computer technology is one of the best ways to visualise and communicate geometrical ideas, especially the recent advances in VR technology that have great potential for facilitating the learning of 3D geometry. The use of body movements as a means to construct geometrical concepts is another powerful medium that aids the communication of 3D geometry. For example, Logo has set up a paradigm that uses a turtle to go forward, back, and turn left and right and so forth to express geometrical ideas.

### 2.3.2 Development and understanding of 3D geometry

The word “development” means a progression from a simpler or lower to a more advanced, mature, or complex form or stage. Therefore, the developmental model is often hierarchical with levels or stages. This section reviews two developmental models of 3D geometry generated by Piaget and van Hiele.
2.3.2.1 Piaget’s stages of development

Darken (1996) presented a table (see Table 2.3) elaborating the relationship between spatial ability, frames of reference, and Piaget’s stages of cognitive development by relating to the frame of reference.

Table 2.3
The Developmental Stages of Formal Operations and Spatial Ability

<table>
<thead>
<tr>
<th>Operational Development</th>
<th>Frame of Reference</th>
<th>Spatial Relations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensorimotor (0-2 years)</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Preoperational (2-6 years)</td>
<td>Egocentric</td>
<td>Proximity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Separation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Open/Close</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Between</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Order</td>
</tr>
<tr>
<td>Concrete Operational (7-9 years)</td>
<td>Fixed</td>
<td>Enclosure</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Continuity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Geometry</td>
</tr>
<tr>
<td>Formal Operations (11 years after)</td>
<td>Coordinate</td>
<td>Proportional Scale</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reduction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Distance Estimates</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Coordinates</td>
</tr>
</tbody>
</table>

This table indicates that the development in the use of reference frames from concrete to abstract, and from subjective to objective viewpoints. Piaget and Inhelder (1956) characterize three major types of relations or properties of space: topological, projective, and Euclidean space. They conjectured that the development of spatial relations progresses from topological to projective to Euclidean space.

Although it can be seen that some topological geometry develops in children before projective and then Euclidean geometry, however, it is not suggested to be the sequence for teaching and learning geometry. Research found that some topological notions such as invariance and network appear late and some projective and Euclidean notions appear early. Cooper (1986) indicated that the findings from this body of research suggest that the focus on change or transformation of the three geometrical spaces is a powerful method of developing geometrical ideas.
2.3.2.2 van Hiele’s levels of geometrical understanding

In late 1950’s, Pierre and Dina van Hiele introduced their developmental model of geometrical thinking by observing the difficulties that their students had in learning geometry (Bosnick, 2001; Mason, 1998). van Hiele (1986) identified five levels of geometrical understandings, which are sequential and hierarchical:

Level 1 (Visualisation): Students recognize figures by appearance alone, often by comparing them to a known prototype. The properties of a figure are not perceived. At this level, students make decisions based on perception, not reasoning.

Level 2 (Analysis): Students see figures as collections of properties. They can recognize and name properties of geometrical figures, but they do not see relationships between these properties. When describing an object, a student operating at this level might list all the properties the student knows, but not discern which properties are necessary and which are sufficient to describe the object.

Level 3 (Abstraction): Students perceive relationships between properties and between figures. At this level, students can create meaningful definitions and give informal arguments to justify their reasoning. Logical implications and class inclusions, such as squares being a type of rectangle, are understood. The role and significance of formal deduction, however, is not understood.

Level 4 (Deduction): Students can construct proofs, understand the role of axioms and definitions, and know the meaning of necessary and sufficient conditions. At this level, students should be able to construct proofs such as those typically found in a high school geometry class.

Level 5 (Rigor): Students at this level understand the formal aspects of deduction, such as establishing and comparing mathematical systems. Students at this level can understand the use of indirect proof and proof by contra positive, and can understand non-Euclidean systems.

Clements and Battista (1992) also proposed the existence of a Level 0, which they called pre-recognition. Students at this level notice only a subset of the visual characteristics of a shape, resulting in an inability to distinguish between figures. For
example, they may distinguish between triangles and quadrilaterals, but may not be able to distinguish between a rhombus and a parallelogram.

According to van Hiele, each student’s progress through each level of thought involves five phases of learning. A student may need to cycle through some of the five phases more than once with a particular topic. The phases are described below.

1. Information: Through discussion, the teacher identifies what students already know about a topic and the students become oriented to the new topic.

2. Guided orientation: Students explore the objects of instruction in carefully structured tasks such as folding, measuring, or constructing. The teacher ensures that students explore specific concepts.

3. Explicitation: Students describe what they have learned about the topic in their own words. The teacher introduces relevant mathematical terms.

4. Free Orientation: Students apply the relationships they are learning to solve problems and investigate more open-ended tasks.

5. Integration: Students summarize and integrate what they have learned, developing a new network of objects and relations.

van Hiele asserted that the five levels of geometrical thought do not correspond with students’ age, and as students develop the cognitive skills necessary to master one level, they progress to the next.

van Hiele’s notion that students cannot skip levels has been both confirmed and questioned in the research literature (see Bennie, 2000). Research in the United States and other countries has found that some mathematically talented students appear to skip levels, perhaps because they develop logical reasoning skills in ways other than through geometry (Mason, 1995, 1998).

van Hiele also asserted that there is no gradual change, but a sudden leap from one level to the next (see Pegg & Davey, 1998). This assertion was disputed by Burger and Shaughnessy (1986), Fuys, Geddes, and Tischler (1988), and Gutiérrez, Jaime, and Fortuny (1991). They found that students may move back and forth between levels, and they concluded that the levels were not discrete but continuous.
Other criticisms made by De Villiers and Njisane (1987) indicated that the use of hierarchical classification might not be necessary for formal deductive thinking. They have also suggested that the van Hiele theory needs refinement with regards to the levels at which deduction can occur, and propose that simpler intuitive deductive reasoning might be possible at levels lower than the ordering level. De Villiers and Njisane (ibid.) noted that there is some confusion within the writings about the van Hiele theory as to where class inclusion is supposed to occur. De Villiers has also suggested that the van Hiele uses a limited notion to proof – learners who cannot see the meaning in terms of logical systematisation do not see the meaning of proof. He suggested that if other meanings of proof were used, then proof could possibly be done at lower van Hiele levels than van Hiele has suggested (De Villiers, 1987).

Despite these criticisms, the van Hiele levels are still being applied and evaluated in relation to 3D geometry, spatial ability and computer technology. Saads and Davis (1997a; 1997b) investigated van Hiele levels and spatial abilities. They tried to relate use of language by students at different van Hiele levels to their spatial perception abilities as classified by Del Grande’s taxonomy (see Section 2.3.1.2). They found that there was a general correlation between van Hiele levels and visual perception. They claimed that this finding generally confirmed the hierarchical nature of the van Hiele levels. However, because the Del Grande’s classification of spatial abilities is not hierarchical, they found that the students’ use of language reflects a higher van Hiele level in discussion than in written tests. Exceptions to this finding such as image formation ability does not correlate to students’ van Hiele level were also noted. Therefore, the relationship between the van Hiele levels and spatial abilities still waits to be revealed.

The relationship between the use of computer technology and van Hiele levels has also been documented in the literature (Battista & Clements, 1991; Choi-Koh, 1999; Clements & Battista, 1994; Dreyfus & Thompson, 1985; Everett & Muligan, 2000; Olive, 1991; Vincent, 1998). Geometrical software such as Logo, Cabri-géomètre and Geometer’s Sketch-Pad etc. were used in these studies. Generally, the findings were that computer technology could build connections between van Hiele levels; computer graphics could be an important factor to the visual level as well as to all levels of geometrical thinking; and computer technology
enabled a more rapid progression through the van Hiele levels. These research studies, however, have put the van Hiele levels as an axiom or a fixed mindset in the first place. Some researchers in the application of computer technology in mathematics education have questioned this assumption. According to Resnick (1996), many of the “new paradigms for computing, new paradigms for thinking” have not been utilised in the research focusing on the relationship of computer technology to the sequential developmental model of the van Hiele levels of geometrical thinking.

2.3.3 Teaching and learning of 3D geometry

Many implications on teaching and learning geometry have been derived from van Hiele’s theory of geometrical thinking (Mason, 1998), such as:

1. Language plays an important role in learning; each level of thought has its own language and its own interpretation of the same term. Discussing and verbalizing concepts are important aspects of the Information, Explicitation, and Integration phases of learning. Students clarify and reorganize their ideas through talking about them.

2. Effective learning takes place when students actively experience the objects of study in appropriate contexts, and when they engage in discussion and reflection.

3. Using lecture and memorization as the main methods of instruction will not lead to effective learning.

4. Teachers can assess their students’ levels of thought and provide instruction at those levels.

5. Teachers should provide experiences organized according to the phases of learning to develop each successive level of understanding.

van Hiele’s theory has had great influence on curriculum and evaluation standards. In USA, the Curriculum and Evaluation Standards are consistent with the methodology advocated by the van Hiele model, especially the phases of learning (see Mason, 1998; National Council of Teachers of Mathematics, 1989). Similarly, in Queensland mathematics source book 9-10 book 2, it states:
These postulated levels of development in understanding have had a major influence on the choice and sequencing of activities in the deductive geometry topic. It is recommended that teachers observe student progress and choose learning experiences with these levels firmly in mind. In particular, it should be recognised that, during the compulsory years of education, very few students will be at a level of geometrical thinking necessary to grasp the intricacies of deductive proof that have been a feature of mathematics programs in the past.

(Queensland. Dept. of Education, 1993, p.87)

Bruni and Seidenstein (1990) suggested that learning of geometry starts from three-dimensional shapes such as block building. They argued that it is natural to begin investigations of geometrical shapes by exploring three-dimensional objects in a child’s environment; children need to have direct experiences with three-dimensional objects so they can see that two-dimensional shapes are components of three-dimensional shapes. Similarly, Cooper (1986) proposed a framework for teaching geometry in which he argued that teaching activities should be based on three levels of development:

1. Experiential level: children learn through their own interaction with their environment.

2. Informal level: certain shapes and concepts are singled out for investigation at an active, non-theoretical level.

3. Formal level: a systematic study of specific shapes is undertaken with important properties identified and labelled.

He noted that the three levels described above are similar to the first three van Hiele levels.

A consensus emerging from the literature is that the teaching and learning of geometry should follow closely to a generic developmental model, which is from empirical to analytical, informal to formal, 3D to 2D, inductive to deductive and concrete to abstract. This has been the root of most teaching and learning approaches of 3D geometry proposed by seminal thinkers in this field.

2.3.3.1 Teaching and learning approach

There have been some approaches for the teaching and learning of geometry proposed in the research literature. For example, Cooper (1986) proposed four
alternative and complementary approaches: environmental, subconcept, thematic, and transformational approaches. He proposed that the use of the approaches depends on the needs of the class, and the geometry instructions can be sequenced by:

1. Spiralling the curriculum: Cooper argued that geometrical concepts and processes are not developed completely in any one year. They must be returned to many times so that other geometrical understandings (e.g., symmetry) can give students a deeper and more complex meaning. He gave an example on the various approaches used within this spiral that:

   The environmental approach could be used in the early years to first bring rectangle to children’s notice. Children could observe and learn to recognise it from the instances in the world (such as doors, table tops, windows, walls, bricks) where it appears. Then later, once line and angle have emerged, the subconcept approach could be used to more deeply study rectangle and its properties. A thematic approach could then be used in a middle primary situation to integrate symmetry and tessellation with rectangle and to look at properties of diagonals. With a subconcept approach three-dimensional solids that are constructed with rectangular faces, could be more formally studied, thus completing the circle back to the environment. These deeper understandings of rectangle could then be drawn into a transformational style look at how the rectangular shape is involved in change in the environment – to similarity, congruence and artistic patterns.

   (Cooper, 1986, p. 14-15)

   He also noted that not only the approaches will be revisited within this spiral; concepts and processes will also be reprised. For example, an intuitive understanding can be developed for line symmetry in the early years; later on, this line symmetry can be used to categorise shapes.

2. Diagnosis: Two aspects of diagnosis are involved. The first is the identification of the approach that suits children and teacher’s objectives. The second is the choice of topic according to the context of geometrical concepts and processes. The diagnosis should be applied throughout teaching and learning activities, which could be further illustrated in the following diagram (adopted from Ashlock, Johnson, Wilson, & Jones, 1983).
Cooper argued that there must be a balance of activities that develop the ideas, consolidate the ideas and apply the ideas. He made six points:

- Consolidation and development should ensure that the new ideas are integrated into the children’s existing knowledge to form conceptual schemas;
- The new ideas are not retained as isolated pigeonhole pieces of information but interconnected into the existing knowledge;
- Development should have exploration as its starting point;
- Development should be active not imitative teaching;
- Activities should be giving in both ways: from model to language and from language to model; and
- Children should experience many embodiments (e.g., using different materials or showing different types) of a concept.

In Baturo and Cooper (1993), these ideas were extended to suggest that the following teaching perspectives should be employed during the teaching of shape and structure, transformation and symmetry, and location and arrangement:

1. Perspective A: The child’s world. In this perspective, teachers introduce spatial concepts and processes through 3D shapes because these are the shapes that children meet in their everyday life. When children explore their worlds, they are necessarily concerned with where they are, where
they wish to go, and where they have been. That can be the starting point for learning the space and geometry, and can lead to activities such as giving and following directions, walking shapes.

2. Perspective B: Subconcepts. This perspective provides opportunities to use a sequenced approach, which focuses on spatial activities in a holistic manner. For example, one intriguing sequence for working out shapes is from position and direction to turn and angle, from turn and angle to straight line and path, then from path to shape.

3. Perspective C: Reversing and Openness. This perspective provides a flexible reverse thinking on normal teaching approaches. For example, teachers often provide an example of a shape and ask children to give its properties. In reversing and openness perspective, we can provide the children with the properties and ask them to construct a shape (or more shapes because of the openness of properties).

4. Perspective D: Dynamic vs. Static. The dynamic perspective here refers to the transformation of geometrical thinking, in which activities such as networks, similarity, congruence, tessellations and dissections can be developed. A particularly important notion in this dynamic perspective is that of invariance – what things are left unchanged by the transformations.

In the discussion above, a key idea emerged is to use the various teaching and learning approaches and perspectives flexibly according to the contexts (e.g., geometrical concepts and processes, children’s development of geometrical understandings, environment and resources). However, this cannot eliminate some fundamental limitations such as the manipulation and visualisation of time, space, and dynamic and complex objects in real environments that impede further development and advancement of children’s construction of knowledge of 3D geometry. The next section focuses on a review of computer technology – the ICT or technology approach that would possibly address these limitations.

2.3.3.2 Computers and 3D geometry

Papert (1980; 1993) suggested that computers have the potential to bridge the gap between formal knowledge and intuitive understanding. Hoyles (1996) elaborated on Papert’s idea and identified more specific points regarding computers’
power for facilitating the learning of geometry. She indicated that computers could bridge the gap between:

1. inductive and deductive thinking;
2. the analytic and the visual; and
3. the drawing and the figure.

The gap-bridging power of computers has been extensively reported in the research literature. Many of the studies used specific geometry software that engaged students directly in geometrical activities, whilst some used multimedia and/or hypermedia to anchor the geometrical understanding and aid the communication of geometrical knowledge. It is possible to identify and discuss some categories of computer utilisation in the learning and teaching of 3D geometry.

**Turtle geometry**

Turtle geometry has its origins in Logo (Papert, 1980). Turtle geometry established a new paradigm for geometry that differed from traditional school geometry. Abelson and diSessa (1981) identified and elaborated on the three fundamental types of differences between turtle geometry and Cartesian coordinate geometry:

1. Intrinsic versus Extrinsic: An intrinsic property of geometrical figures is one which depends only on the figure in question (turtle geometry), not on the figure’s relation to a frame of reference (extrinsic coordinate geometry). For example, in turtle geometry, a circle can be expressed as go FORWARD a little bit, turn RIGHT a little bit, and REPEAT this over and over. While in the Cartesian coordinate representation, a circle is $x^2 + y^2 = r^2$.

2. Local versus Global: In the circle example above, turtle geometry locally deals with geometry a little piece at a time, which is in contrast to $x^2 + y^2 = r^2$ that relies on a large-scale, global coordinate system to define its properties.

3. Procedures versus Equations: Turtle geometry characteristically describes geometrical objects in terms of procedures rather than in terms of equations. This procedural mechanism (such as iteration) is valuable and
hard to capture in the traditional algebraic formalism. Moreover, the procedural descriptions used in turtle geometry are readily modified in many ways. This makes turtle geometry a fruitful arena for mathematical exploration.

Logo is probably the most researched software in use in education. Hoyles (1996) noted that: (a) The Logo environment enables students to directly experience their constructions, (b) Logo has the potential to link visual and symbolic representations in a relatively natural way, (c) Logo is an extensible language to help a variety of approaches and styles of activity, (d) the syntax of Logo programs is mathematically expressive, and (e) in Logo, the mathematics is hidden whilst remaining crucial as a tool to achieve a student’s goal. Similarly, Clements et al. (1998) implied that Logo can: (a) encourage construction of the abstract from the visual, (b) maintain close ties between representations, (c) facilitate examination and modification of code and thus facilitate the explorations of mathematical ideas, (d) encourage procedural thinking, and (e) provide freedom within constraints.

Turtle geometry has been highly regarded as an incubator for knowledge construction of geometry concepts and processes. However, most of its applications have focused on 2D and Euclidean geometry. With respect to 3D geometry, little seems to have been done.

Below, Figure 2.8 shows a 3D graphics of a turtle drawing a Pythagoras tree on a plane and a complex cubic shape in Elica Logo, and Figure 2.9 shows six spherical solids and a polyhedron in MSWLogo (for information about various Logo versions, please refer to Section 2.4.4.3).
Elica Logo employs a 3D graphical window in which the turtle and the perspective viewpoint animate in motion while drawing. This real-time 3D animation allows users to perceive a spatial sense and “rotate the eye,” which is important in children’s spatial learning and development (Sachter, 1991). However, the 3D graphics is neither interactive nor easy to do; the viewpoints of the 3D graphics must be controlled in programs through high level, complex mathematical computation of vectors. Furthermore, Elica has included the third dimension z in its Logo language, but again it only based on formal geometrical concepts such as point and vector.

In contrast to this, MSWLogo not only includes the z dimension, but also employs natural actions into the turtle movement semantics. In MSWLogo, four more turning commands were added: `uppitch`, `downpitch`, `leftroll`, and `rightroll`. This allows users to easily construct geometrical objects in 3D perspective. Nevertheless, MSWLogo still lacks real-time interactive 3D graphics.

Most current implementations of Logo therefore have inadequate means for dealing with 3D geometry. There are other applications in Logo with respect to 3D geometry, but most of them take a formal, algebraic and computational approach based on Euclidean deductive geometry (e.g., to compute the 2D points’ x and y in translation of 3D points’ x, y and z) to draw 3D geometrical figures. Moreover, these turtle graphics are not interactive; the visualisation of geometrical objects can only be manipulated by means of text-based programming. This means that young children will probably not be able to utilise these Logos for learning about 3D geometry. It seems that only higher levels of students (e.g., students with prior
knowledge about vector) can possibly do some 3D geometry in these Logo environments. Nevertheless, some Logo programs such as Elica enables one to create 3D primitives such as \textit{uppitch}, \textit{downpitch}, \textit{leftroll}, and \textit{rightroll}. This is a very powerful feature in turtle geometry, and will be further reviewed in Section 2.4.4.

\textit{Dynamic geometry}

Dynamic geometry (DG) has a special usage in computers and education. Goldenberg and Cuoco (1998) elaborated that:

The term “dynamic geometry”, originally coined by Nick Jackiw and Steve Rasmussen, has quickly entered the literature as a generic term because of its aptness at characterizing the feature that distinguishes DG from other geometry software: the continuous real-time transformation often called “dragging.” This feature allows users, after a construction is made, to move certain elements of a drawing freely and to observe other elements respond dynamically to the altered conditions. As these elements are moved smoothly over the continuous domain in which they exist, the software maintains all relationships that were specified as essential constraints of the original construction, and all relationships that are mathematical consequences of these.

(Goldenberg & Cuoco, 1998, p. 351)

There are many DG programs available such as Cabri-Géomètre (Baulac, Bellemain, & Laborde, 1992, 1994), Cinderella (Richter-Gebert & Kortenkamp, 2002), Geometer's SketchPad (Jackiw, 1991, 1995), Geometry Inventor (Brock, Cappo, Dromi, Rosin, & Shenkerman, 1994), Geometric Supposer (Geometry SuperSupposer) (Schwartz & Yerushalmy, 1992), Geometry Assistant, Euklid, CaR, GeoNET, GeoLOG, Geometry, GRACE, DrGEO, Wingeom etc. (see Stothers, 2003). They are empowered by the advancement of the computer graphical interface (GUI) in which the mouse serves as the primary input and manipulation device. Therefore, the “drag mode” or “dragging” of mouse is essential in DG, and raises some issues in the teaching and learning geometry.

In general, DG software provides basic geometrical objects such as point, line, and circle and so forth for users to draw by mouse’s operations (e.g., click and move, drag). Users can also set constraints or special relationships in between those geometrical objects such as two lines are perpendicular or parallel, a point locates on the circumference of a circle, and the length of a line segment is the scale ratio of more line segments. Once the constraints and relationships among geometrical
objects are set, users can then drag those geometrical objects and observe the continuous change of the geometrical figure under the constraints. Figure 2.10 shows the conic sections as the locus of perpendicular bisectors by dragging the focus point in Geometer’s SketchPad.

![Figure 2.10 Conic section in Geometer’s SketchPad](image)

Some educational issues that have been raised by the introduction of DGs are:

1. How do students perceive the moving display?
2. How do students interpret what they see?
3. What is that picture on paper or screen? (Goldenberg & Cuoco, 1998)

Because of the moving display, students develop the notion of “figure” – attending to underlying relationships rather than to the particulars of a specific drawing. Hence, the gap between drawing and figure is bridged in DG. And because the relationships and constraints in between geometrical objects are set, the moving actions become the vehicle for proofs, reasoning, and justifications of many geometry theorems (Bennett, 1995, 1999; De Villiers, 1998; Hanna, 1998; C. Laborde, 2000; Little & Sutherland, 1995; Mariotti, 2000; Marrades & Gutiérrez, 2000; Rahim, 2002; Sanders, 1994; Scher, 1995; Shaffer, 1995; Thibault & Barre, 1996).

With regards to 3D geometry, many applications can be found in DG. Figure 2.11 below shows a 3D cube and a projected rectangular box in Cabri-Géomètre.
All geometrical objects in Figure 2.11 are 1D and 2D objects. However, after some constraints and relationships were applied, 3D objects were created. These 3D objects can then be resized, rotated and topologically morphed by the mouse dragging on some specific points.

To sum up, DG is successful in teaching and learning geometry because of its interactive style of direct manipulation of geometrical objects, which distinguishes it from Logo (J.-M. Laborde, 1996). However, it is still based on the nature of a 2D drawing canvas on screen and Euclidean geometry. The construction of 3D geometry from 2D geometry is neither easy nor a natural way to achieve 3D spatial sense. Most research studies about DG have been conducted with secondary and tertiary students with deductive reasoning and proofs of Euclidean geometry in mind. Therefore, the applicability of DG within primary school contexts is still open to question. However, despite this and its lack of 3D functionality, the DG paradigm of thinking and doing geometry provides much promise for the teaching and learning of geometry.

**Virtual Reality (VR)**

With the recent progress in the power of desktop computer graphics, much computer software now has the ability to create 3D Virtual Reality space behind the 2D computer screen (see Web3D Consortium, 2003b). Also, much VR hardware has been developed for the applications such as training, education, scientific simulation, medical program, and entertainment etc. Considering the affordability in educational
application, this section mainly focuses on the review of desktop VR (see Section 1.2).

VR is three-dimensional in nature. Therefore, VR environments have the advantage that any 3D geometrical objects within a VR environment can be viewed naturally with 3D perspectives. This is totally different from the precedent set by the turtle geometry and dynamic geometry environments. A review of the research literature found many studies that were relevant to teaching and learning 3D geometry in VR environments (Ainge, 1996; Allport et al., 1999; Barab et al., 2000; Chan et al., 1999; Kwon et al., 2002; Pasqualotti & Freitas, 2002; Song et al., 2000; Teixeira et al., 1999).

Ainge (1996) studied upper primary students’ construction and exploration of 3D shapes in VR environment and card nets. He found that VR had little impact on shape visualisation and name writing, but strongly enhanced recognition. He also noted that students had no major difficulty in using the VREAM (a VR software) program, and enthusiasm for VR was unanimous and sustained. Barab et al. (2000) conducted a Virtual Solar System project (VSS), in which students used Virtual Reality Modelling Language (VRML) to construct a cylinder with the codes like geometry Cylinder {}. The codes could also be edited and visualised in CosmoWorlds -- a VR modelling tool with further manipulations such as colouring, rotating, resizing, and applying textural images and so forth by mouse click and drag (see Figure 2.12). Findings in the VSS project regarding the use of 3D modelling software and VRML language were similar to Ainge, namely that the integration of 3D technologies into learning environments deepened students’ understanding of the content, but with much time devoted to the learning of the software.

Figure 2.12 A cylinder in CosmoWorlds
The direct manipulation of 3D geometrical objects in 3D modelling tools seems to be an advantage over dynamic geometry software. Indeed, VR provides another dynamic environment in which students can construct and manipulate three dimensionally. However, there are still critics. Leach et al. (1997) commented that most interactive 3D graphics applications employ a 2D cursor and 2D pointing device for interaction. Given that the user is manipulating 3D objects or worlds in these applications, this gives rise to a fundamental mismatch in dimensionality.

Kwon et al. (2002), and Pasqualotti and Freitas (2002) also utilised VRML but without 3D modelling tools. Instead, the VR environments were integrated into web environments. This reduced the ability of direct manipulation of 3D objects but increased the accessibility from Internet. Figure 2.13 and 2.14 present a screenshot of each application.

![Figure 2.13 Spatial visualisation software using virtual reality (Figure adopted from Kwon et al., 2002)](image)

![Figure 2.14 Examples of objects constructed using the MAT\textsuperscript{3D} (Figure adopted from Pasqualotti and Freitas, 2002)](image)
Similarly, the VR in web environments was found to be effective for learning and teaching 3D geometry, and improving students’ spatial visualisation ability (Kwon et al., 2002; Pasqualotti & Freitas, 2002). Pasqualotti and Freitas (2002) argued that VR is the best and probably the only strategy that permits students to learn by first-person experience. In DG environments, geometrical objects are manipulated; therefore the users perceive only the spatial visualisation in terms of McGee’s taxonomy of spatial ability (McGee, 1979) (see Section 2.3.1.2). In VR environments, both the geometrical objects and the user can be manipulated. When users walk or navigate in VR environments, their spatial orientation ability is called into action.

Elliott and Bruckman (2002), Edwards, Elliott, and Bruckman (2001), and Elliott, Adams, and Bruckman (2002) also utilised interactive computer 3D graphics for mathematics learning. They designed the software named AquaMOOSE 3D (Figure 2.15), which was a multi-user desktop 3D environment designed to help students learn about the behaviour of parametric equations (analytic geometry). In AquaMOOSE 3D, students were able to specify parametric equations and visualise in the 3D environment, and then use the mouse to navigate in the 3D underwater environment to examine the visualisation of the parametric equation. AquaMOOSE also employed the metaphor of avatar (see Section 2.4.5.3), in which students could become the avatar (e.g., a fish) to swim and view from the viewpoint of the avatar in a sine wave in x and a cosine in y to create a spiral (see Figure 2.15). They termed this as the experience of “mathematical movement”, which they hoped to facilitate the transfer of mathematical knowledge. They suggested that this 3D multi-user environment holds promise for helping high-school students learn mathematics. However, they also suggested that issues such as integrating local and remote interaction, helping students stay oriented in a 3D world, scaffold programming and mathematics through a programming environment, and more time and trials were needed for future iteration and research.
VR, therefore, offers many advantages that naturally facilitate the teaching and learning of 3D geometry. However, using a 2D device to manipulate 3D objects (Leach et al., 1997) and mere direct manipulation of 3D objects (Pasqualotti & Freitas, 2002) do not guarantee success. Furthermore, concerning the processes of constructing knowledge of 3D geometry, whether the VR environments should be built and explored by students (e.g., Ainge, 1996) or be built by experts then explored by students (e.g., Kwon et al., 2000), needs to be justified.

**Multimedia, Hypermedia and Internet**

Blakesley (2001) investigated the definition of multimedia from 15 sources. In short, multimedia means more than one medium; multimedia involves the media in forms of visual and audio; multimedia productions are typically developed and controlled by computer; multimedia is used to represent, communicate and receive information; multimedia is a human-computer interaction; multimedia is a form of semiotic communication model; multimedia is fundamental to teaching and learning; and the definition of multimedia is continuously growing.

In fact, according to these definitions, the previous discussion about turtle geometry, dynamic geometry, and even virtual reality should be also in the domain of multimedia. Multimedia has become a more general term to describe a mode of computer application.

There are numerous multimedia applications for teaching and learning 3D geometry. For example, Hidaka (1994) developed a 3D-LAB microworld for
learning and teaching some basic three-dimensional solid objects such as cone, pyramid, cylinder, or polyhedron. With 3D-LAB, students could rotate and modify 3D solids interactively, find segments that intersect with a cutting plane, open or close truncated solids, and measure lengths, areas, volumes and angles. Figure 2.16 below shows two similar multimedia for teaching and learning 3D geometry.

![3D Images](http://www.wkbradford.com/3dim1.htm)  ![Small Stella](http://home.connexus.net.au/~robandfi/Stella.html)

Figure 2.16 Examples of multimedia for teaching and learning 3D geometry

Multimedia provides a rich environment containing multiple means of communication such as text, image, diagram, animation, video, sound, and speech and so forth to aid teaching and learning. It usually incorporates a fancy and user-friendly computer interface, in which it draws the students’ attentions and interests. Its interactive and student self-paced style of learning can also accommodate different learning/cognitive styles and promote effective learning (Crosby & Stelovsky, 1995).

Hypermedia is seen as an extension of multimedia that combines multimedia with the concept of hypertext. Dede and Palumbo (1991) argued that hypermedia is a fundamentally innovative means of thinking and communicating. They gave four arguments for why it is a major advance over other media:

1. The associative, non-linear nature of hypermedia mirrors the structure of human long-term memory, empowering both intelligence and coordination through intercommunication.

2. The capability of hypermedia to reveal and conceal the complexity of its content lessens the cognitive load on users of this medium, thereby enhancing their ability to assimilate and manipulate ideas.
3. The structure of hypermedia facilitates capturing and communicating knowledge, as opposed to mere data.

4. Hypermedia’s architecture enables distributed, coordinated interaction, a vital component of teamwork, organizational memory, and other “group mind” phenomena.

Dede and Palumbo also noted that there are four potential limits in hypermedia:

1. People become disoriented when navigating through large hypermedia structures.

2. Traversing a hypermedia network imposes considerable cognitive overhead on the user.

3. Creating hypermedia structures involves a very large front-end investment of time and expertise.

4. “Tower of Babel” situations are likely in shared hypermedia systems.

Despite those arguments, many empirical researches have found hypermedia to be effective as an educational tool (Crosby & Stelovsky, 1995; Frey & Simonson, 1993; Liu & Reed, 1994, 1995). Teixeira et al. (1999) developed a hypermedia learning environment encapsulated in the form of HTML help file (Figure 2.17) for the teaching of descriptive geometry. They argued that the united use of texts, hypertext, images, animations and virtual reality in the hypermedia environment could create an atmosphere highly interactive and extremely rich in information, with great potential for use in any knowledge area.

![Figure 2.17 Hypermedia learning environment for teaching descriptive geometry](image)
The Internet is the pertinent platform for disseminating hypermedia. The World-Wide-Web (WWW) on the Internet is a hypermedia application that further extends the accessibility, the depth and the width of information of hypermedia. New networking technologies also enable web environments to generate dynamic hypermedia, integrate multiple perspectives, and build up virtual communities. Math Forum presents a collaborative geometry that links resources for knowledge sharing and building (Figure 2.18).

Figure 2.18 Math Forum web discussion in geometry
(http://www.mathforum.com/epigone/geometry-software-dynamic)

Multimedia, hypermedia and the Internet have many implications for the teaching and learning of 3D geometry. They are rich semiotic resources of physical and digital materials, personal thoughts, and social-cultural collaborations for communication, knowledge presentation and representation, and knowledge construction. They have been utilised broadly in many disciplines in education. Thus, they should be regarded as imperative computational paradigms in the teaching and learning of 3D geometry.

2.3.4 Synthesis

Section 2.3 has focused on a review of 3D geometry. It began with a discussion of the components of 3D geometry to target the knowledge domain for this research study. After this knowledge domain had been identified, the developmental models of the concepts and processes of the knowledge domain were
investigated. These developmental models of 3D geometry have implications for the teaching and learning of 3D geometry. Traditional approaches for teaching and learning 3D geometry have their strength and weakness. The strength is that they are spiral and diagnostic approaches. However, there are still some fundamental limitations in traditional pedagogy, which are addressed in the literature review of computers and 3D geometry in Section 2.3.3.2.

In the review of the components of 3D geometry, this researcher diverged from the traditional mindset that sees geometry as chapters of some content knowledge such as three-dimensional shapes, two-dimensional shapes, symmetry and tessellations, position and orientation, geometry on a sphere, deductive geometry, and analytical geometry and so forth. Instead, the 3D geometry was conceptualised within a semiotic viewpoint, in which the 3D geometry consists of three components, namely external material world, internal spatial ability, and communication. These three components form another semiotic triad, which is shown in Figure 2.19 below.

![Semiotic triad of 3D geometry](image)

**Figure 2.19** Semiotic triad of 3D geometry

This triad presents an inclusive model of 3D geometry. The external material world indicates that geometry is the study of the space in which we are living. It includes all physical objects with their properties and behaviours to be the content knowledge that interests this research study. As constructivists have argued, knowledge must be constructed by active learners and be integrated into their internal mental structures (see Duffy & Jonassen, 1992a). This semiotic model of 3D geometry therefore includes two other indispensable components: the internal spatial ability of learners signifying the capacities for building internal structures of knowledge, and the various signs communicating between the other two components.

The development and understanding of 3D geometry concepts and processes relate closely with the components of 3D geometry. The Piagetian model identifies
the use of a different frame of reference and the progress of geometrical concept from topological to projective to Euclidean based on physiological maturation. The van Hiele model suggests that the geometrical understanding moves from visualisation, analysis, abstraction, and deduction, to rigor according to experience. These two models of development and understanding of geometry share some important themes:

1. A generic sequence for developing geometrical concepts from concrete (visualisation) to abstract (abstraction), inductive (analysis) to deductive, informal to formal, and subjective (egocentric) to objective (rigor, global generalisation of coordinate system).

2. Real world experience, visualisation of objects, and use of language are important throughout the development of geometrical concepts. This has implications for the communication component of 3D geometry, which needs multiple forms of sign systems.

The teaching and learning of 3D geometry is much influenced by the developmental models of 3D geometry. There are many approaches for teaching and learning 3D geometry that support different aspects of the developmental models. The important idea behind these approaches is to use them spirally and diagnostically according to the teaching and learning contexts. This idea seems to create much flexibility and efficacy on to the teaching and learning of 3D geometry. However, traditional limitations (such as drawing using pencil and paper doesn’t get the comprehension of figures, and the manipulation in real world environments has difficulties with time, space and physical laws etc.) constrain the visualisation and advancement of teaching and learning in geometry.

Multiple paradigms of using computers to facilitate the teaching and learning of 3D geometry were reviewed. Included in this review were turtle geometry, dynamic geometry, virtual reality, and multimedia, and hypermedia and Internet. Some indications of how we can efficaciously unleash the power of computers in the teaching and learning 3D geometry have emerged from the literature review. These suggest that we can employ the computers’ abilities of:

1. Programming to link between language and visualisation, and promote logical and procedural thinking; (Turtle Geometry)
2. Direct manipulation of geometrical objects or figures to bridge the gap between drawing and figure; (Dynamic Geometry)

3. Real-time 3D graphics to facilitate both spatial visualisation and orientation abilities; (Virtual Reality) and

4. Organisation of thoughts of 3D geometry with hypermedia and networked collaboration through Internet web environments. (Multimedia, Hypermedia and Internet)

This section has focused on 3D geometry involving the understanding of knowledge domain, the understanding of learners, and a narrow understanding of technology focusing on current uses of computers specifically on 3D geometry. The next section will focus on technology including its role in education and some sustained metaphors, thus, to form a better understanding of technology.

2.4 TECHNOLOGY

The Oxford dictionary of current English (1993) defines technology as “knowledge or use of the mechanical arts and applied sciences.” According to Encyclopaedia Britannica (2000), technology is defined as “the application of scientific knowledge to the practical aims of human life or, as it is sometimes phrased, to the change and manipulation of the human environment.” The meaning of technology is broad and changes over time; it does not necessarily mean the electronic or digital computers. However, such meaning is often the case in mathematics education.

Technology innovations grow exponentially, especially in computer technology. Because of the rapid emergence of new computer technologies, it is not possible to always pursue the latest computer technology for mathematics education. Moreover, latest computer technologies are not always the best for education. Technologies must be examined with their natures, features and potentials, developed in teamwork, integrated with other technologies and educational theories, practiced in communities, and evaluated and revised to form new innovations.

The following sections explore sustainable and inspirational metaphors of computer technology that serve as principles and guides for the design and use of computer technology in mathematics education. Also explored is the new computer
technology “Virtual Reality” with its underlying philosophy, its distinguishable features, and its potential to create a learning environment for mathematics education.

2.4.1 The nature of technology use in mathematics education

The technology principle described in NCTM principles and standards for school mathematics states that:

Technology is essential in teaching and learning mathematics; it influences the mathematics that is taught and enhances students' learning. Electronic technologies -- calculators and computers -- are essential tools for teaching, learning, and doing mathematics. They furnish visual images of mathematical ideas, they facilitate organizing and analyzing data, and they compute efficiently and accurately. They can support investigation by students in every area of mathematics, including geometry, statistics, algebra, measurement, and number. When technological tools are available, students can focus on decision-making, reflection, reasoning, and problem solving.

(National Council of Teachers of Mathematics, 2000, p. 24)

This statement addresses the importance and potential of technology. The NCTM further elaborated on how we should use technology.

The effective use of technology in the mathematics classroom depends on the teacher. Technology is not a panacea. As with any teaching tool, it can be used well or poorly. Teachers should use technology to enhance their students' learning opportunities by selecting or creating mathematical tasks that take advantage of what technology can do efficiently and well — graphing, visualizing, and computing.

(p. 25)

Heid (1997) expressed some concerns about the students’ uses of technology. She noted that there are fears that when students use computers as a computational tool, they will misuse or overuse technology or use it for inappropriate topics; when students use programmed instruction, they may contribute too much knowledge to the computer that is functioning as their teacher; technology issues may overshadow the mathematical issues for students; and students shift attention from the mathematical activity to the tool.
Roschelle et al. (2001) argued that the mere presence of computers in the classroom does not ensure their effective use. They analysed 21 major studies on the effectiveness of computers as learning tools ranging from 1989 to 1999, and found that some computer applications were more successful than others and many factors influenced how well the applications were implemented. Among the analysis of several computer-based applications, four fundamental characteristics of learning were identified to improve how and what children learn if they are supported in the computer-based applications. They are (a) active engagement, (b) participation in groups, (c) frequent interaction and feedback, and (d) connections to real-world contexts.

Pea (1985) argued that computers are classically used as amplifiers of cognition. But alternatively, computers can be used as reorganizers of mental functioning. As an amplifier, the computer increases the number and range of examples for students to manipulate with ease. Thus, the computer is just amplifying what we currently do with mathematics. Contrarily, as a reorganizer, the computer provides new ways of thinking and doing mathematics. Thus, the computer alters the mathematics content and reshapes what we are. Kilpatrick and Davis (1993) argued that the computer is not merely an amplifier of general curriculum issues specific to the mathematics curriculum. It changes certain fundamental questions we need to consider such as “what is mathematics?”, “what knowledge of mathematics does tomorrow’s society demand?”, and “what mathematics should this pupil learn so as to be a wise and human citizen of that society?”

In a similar vein with the viewpoint of knowledge, Scardamalia and Bereiter (1993; 1996a; 1996b) argued that the strategies of using computers should be focused on creating knowledge building environments rather than knowledge reproduction environments. Knowledge reproduction such as “copy and delete” text is easy to complete with the aid of computer technology. However, it impedes students from judging and identifying the central point of the text. This information reproducing activity has its root in the belief in which knowledge is transmitted and learning consists of reproduction of knowledge in someone’s mind. In contrast to knowledge reproduction, knowledge building emerges from the constructivist viewpoint believing that conceiving a new idea and learning are fundamentally the same process. In this view, learning is not students displaying or reproducing what
they have learned, but is like scientific discovery and theorizing – a process of working toward more complete and coherent understanding.

2.4.2 The role of computer

Taylor (1980) was an early pioneer in the field of computers in education. He identified three modes of using computing in education namely tutor, tool and tutee. These three modes are summarised in Table 2.4.

Table 2.4
Tutor, Tool and Tutee – the Three Modes of Using Computing in Education

<table>
<thead>
<tr>
<th>Tutor</th>
<th>Tool</th>
<th>Tutee</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Rationale and process</strong></td>
<td>The computer is programmed by “expert,” the student is then tutored by the computer executing the program(s). The computer presents some subject material, the student responds, the computer evaluates the response, and, from the results of the evaluation, determines what to present next.</td>
<td>The computer needs only have some useful capability programmed into it such as statistical analysis, super calculation, or word processing. Students can then use it to help them in a variety of subjects.</td>
</tr>
<tr>
<td><strong>Features</strong></td>
<td>▪ If appropriately designed the computer can accommodate a wide range of student difference. ▪ Development of a good tutor program is time consuming and expert work.</td>
<td>▪ Their use can pay off handsomely in saving time and preserving intellectual energy by transferring necessary but routine clerical tasks of a tedious, mechanical kind to the computer.</td>
</tr>
<tr>
<td><strong>Examples</strong></td>
<td>CAI</td>
<td>Spreadsheet, word processing</td>
</tr>
</tbody>
</table>

72
Taylor indicated that the use of computer as tutor and tool can both improve and enrich classroom learning, and neither requires student or teacher to learn much about computers. However, he pointed out that neither tutor nor tool mode confers upon the user much of the general educational benefit associated with using the computer as tutee.

Bruckman and Bandlow (2002) also categorised genres of educational software in relation to Taylor’s classifications, and linked them with different learning theories (Table 2.5).

Table 2.5  
*Genres of Children’s Software*

<table>
<thead>
<tr>
<th>Genres</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entertainment</td>
<td>Games created solely for fun and pleasure.</td>
</tr>
<tr>
<td>Educational</td>
<td>Software created to help children learn about a topic using some type of pedagogy – an approach to teaching and learning.</td>
</tr>
<tr>
<td>- Computer as Tutor</td>
<td>Often referred to as “drill and practice” or “computer-aided instruction” (CAI), this approach is grounded in behaviourism. Children are presented with information and then quizzed on their knowledge.</td>
</tr>
<tr>
<td>- Computer as Tool</td>
<td>The learner directs the learning process, rather than being directed by the computer. This approach is grounded in constructivism, which sees learning as an active process of constructing knowledge through experience.</td>
</tr>
<tr>
<td>- Computer as Tutee</td>
<td>Typically, the learner uses construction kits to help reflect upon what he or she learned through the process of creation. This approach is grounded in constructivism and constructionism.</td>
</tr>
<tr>
<td>- Computer-supported Collaborative learning (CSCL)</td>
<td>Children use the Internet to learn from and communicate with knowledgeable members of the adult community. Children can also become involved in educational online communities with children from different geographical regions. This approach is grounded in social constructivism.</td>
</tr>
<tr>
<td>Edutainment</td>
<td>A mix of the entertainment and educational genres.</td>
</tr>
</tbody>
</table>

In their categorisation, Bruckman and Bandlow introduced a fourth mode of using computer as CSCL. They noted that most tools for learning have traditionally been designed for one child working at the computer alone. However, learning is generally recognised to be a social process. The new emerging computer technologies such as Internet and database systems have made it possible for children to learn from one another and from knowledgeable members in online communities.
2.4.3 Microworld

There has been a strong link between Logo and Microworld that has led people to think they are the same thing. In fact, the notion of microworld was first used by artificial intelligence (AI) specialists to describe a small, coherent domain of objects and activities implemented in the form of a computer program (Weir, 1987). It was then used by Papert (1980) with Logo the turtle graphics in which he described the idea of a microworld as:

… the Turtle defines a self-contained world in which certain questions are relevant and others are not… this idea can be developed by constructing many such “microworlds,” each with its own set of assumptions and constrains. Children get to know what it is like to explore the properties of a chosen microworld undisturbed by extraneous questions. In doing so they learn to transfer habits of exploration from their personal lives to the formal domain of scientific theory construction.

(Papert, 1980, p.117)

Although the term microworld arises in the context of introducing the Logo programming language, however, the idea of a microworld does not necessarily involve Logo programming, modification of Logo code or interaction with Logo.

Edwards (1995) examined literature and found that microworlds were analogously used in a variety of names such as simulations, intrinsic models, interactive illustrations, and discovery-based learning environments. He identified that those various uses of the idea of microworld could be categorised into two views: the structural view of designers and the functional view of learners. From the structural view of designers, a microworld might include the following elements:

1. A set of computational objects (defined formally through procedures or programs), which have been created to reflect the structure of mathematical or scientific entities within some sub-domain of mathematics or science.

2. Links between representations of the underlying mathematical or scientific entities or objects. Typically, these representations include a symbolic and a visual or graphical component, although it is an area for further exploration and investigation as to what other modalities might be usable in the design of microworld like environments (e.g., sound and motion).
3. Objects and operations combined to form more complex objects or operation; this is particularly so when one of the representations consists of a “language” for the entities and operations.

4. A set of activities, which may be pre-programmed into the environment or which may be instantiated in worksheets or verbal instructions, in which the user is challenged to use the entities and operations to reach a goal, solve a problem, duplicate a situation or pattern, and so forth.

And in the use of a microworld from the functional view of learners, the learners are expected to:

1. manipulate the objects and execute the operations instantiated in the microworld, with the purpose of inducing or discovering their properties and the functioning of the system as a whole. Experimentation, hypothesis generation and testing, and open-ended exploration are encouraged;

2. interpret feedback from these manipulations (feedback which may be provided through multiple, linked representations) in order to self-correct or “debug” his or her understanding of the domain; and

3. use the objects and operations in the microworld either to create new entities or to solve specific problems or challenges (or both).

Therefore, the power and value of microworlds are apparent. Edwards (1995) then concluded his review of microworlds as embodiments of mathematics. He argued that the value of microworlds went beyond their reifying link between the representation and the mathematical entity, to providing the opportunity for learners to kinaesthetically and intellectually interact with the designers’ construction of a system of mathematical or scientific entities, as mediated through the symbol system of a computer program.

Hoyles, Noss and Adamson (2002) noted that the microworlds environments led directly to the idea of constructionism, which argued that effective learning will not come from finding better ways for the teacher to instruct but from giving the learner better opportunities to construct. They viewed constructionism as giving the learner opportunities to build his/her own physical, virtual and mental knowledge structures, and programming as a means to do that. However, the programmability
of microworlds has been a controversial topic, which will be further discussed in next section.

2.4.4 Computer programming language

A programming language is defined as an artificial language used to write instructions that can be translated into machine language and then executed by a computer (The American Heritage dictionary of the English language, 2000). According to Google’s (2003) directory of computer programming language, there are at least 160 current languages in computing (see http://directory.google.com/Top/Computers/Programming/Languages/). They are used for a wide range of purposes such as mathematics computing, scientific analysing, accounting, computer system, database, AI, networking and so forth. The effects on human cognition and learning of computer programming language, in particular, have interested educators. It is the focus of the following section to discuss the educational values of computer programming languages. However, to view the computer programming languages from their own domain is also important.

2.4.4.1 The domain of computer science

According to the free on-line dictionary of computing, programming language is described as:

… a formal language in which computer programs are written. The definition of a particular language consists of both syntax (how the various symbols of the language may be combined) and semantics (the meaning of the language constructs). Languages are classified as low level if they are close to machine code and high level if each language statement corresponds to many machine code instructions (though this could also apply to a low level language with extensive use of macros, in which case it would be debatable whether it still counted as low level). A roughly parallel classification is the description as first generation language through to fifth generation language. Another major distinction is between imperative languages and declarative languages.

(Howe, 2002)

Two ways of classifying programming language are found in the above description; the first is the evolution of computer programming language from first to fifth generation, and the second is imperative language versus declarative language.
The five generations of computer programming language are briefly described in TechTarget (2003) as below:

1GL: the first-generation language is machine language or the level of instructions and data that the processor is actually given to work on (which in conventional computers is a string of 0s and 1s).

2GL: the second-generation language is assembler or assembly language (e.g., ADD 12,8). An assembler converts the assembler language statements into machine language.

3GL: the third-generation language is a “high-level” programming language such as Pascal, C, or Java. A compiler converts the statements of a specific high-level programming language into machine language. A 3GL language requires a considerable amount of programming knowledge. An example of Java language statements adding from 1 to 10 is:

```java
public class BasicsDemo {
    public static void main(String[] args) {
        int sum = 0;
        for (int current = 1; current <= 10; current++) {
            sum += current;
        }
        System.out.println("Sum = " + sum);
    }
}
```

4GL: the forth-generation language is designed to be closer to natural language than a 3GL language. Languages for accessing databases are often described as 4GLs such as SQL (Structured Query Language). For example:

```sql
SELECT * FROM customer WHERE purchases > 100
```

5GL: the fifth-generation language is programming that uses a visual or graphical development interface to create source language that is usually compiled with a 3GL or 4GL language compiler. Visual programming allows you to easily envision object-oriented programming class hierarchies and drag icons to assemble program components.

With respect to the second classification of programming language, the imperative language (also known as procedural language) is the language where the programmer specifies explicit sequences of steps to follow to produce a result, while declarative language (also known as relational language or functional language) describes relationships between variables in terms of functions or inference rules.
The language executor (interpreter or compiler) then applies some fixed algorithm to these relations to produce a result (Howe, 2002). For example, an imperative programming in Fortran 77 looks like:

```fortran
Integer function power(m,n)
    Integer m,n,p
    p=1
    if (n .gt. 0) then
        do i=1,n
            p=p*m
        end do
    end if
    power=p
end
```

And a general declarative programming looks like:

```fortran
power(m,n)=
    if n <= 0 then 1
    else m*power(m,n-1)
```

(Examples adopted from http://www.abo.fi/~atorn/FuncProg/Page11.html)

It may be said that imperative programming languages are designed for programmers, whereas functional programming is designed for problem solvers (Törn, 1999).

### 2.4.4.2 The educational context

Research with respect to teaching and learning computer programming is abundant, but with mixed and conflicting results. For example, when Oakley and McDougall (1997) and Clements (1999) reviewed the extensive research literature in this area, they found that computer programming has positive effects on:

1. mathematics achievement including geometry and spatial sense, number, arithmetic and algebra, ratio and proportion;
2. problem solving and higher-order thinking;
3. language arts and creativity;
4. social-emotional development;
5. transfer of learning from programming to other areas of curriculum;
6. learners’ self-confidence and feelings of empowerment in learning; and
7. the encouragement of learners’ thinking about thinking.
However, they also found negative or insignificant results for learning in computer programming environments. These included:

1. students did not tend to generalize;
2. most students did not show evidence of transfer of learning;
3. students failed to generalize learning; and
4. no significant differences in achievement between programming group and control group (in Logo).

These negative findings can be possibly attributed to the use of different methodologies (e.g., empirical versus ethnographic approaches, for different research projects), poor conceptualisation of the problem to be studied, heavy emphasis on anecdotal evidence, wide range of instruments used, variations between studies in the amounts and types of programming undertaken, variations in the classroom environment, small numbers of subjects used in some empirical research, and differing ages of subjects ranging from preschool to adult learners (Oakley & McDougall, 1997), and mere “exposure” without teacher planning and mediation of carefully planned sequences of computer programming activities (Clements, 1999).

Oakley and McDougall (1997) synthesised literature and identified nine key factors that facilitate the successful integration of programming (Logo) into education:

1. Time: Research suggests that achievement of expert programmer status takes perhaps 1000 hours. The best experimental treatments involve no more than about 100 hours. Twenty to forty hours is more typical and once an exploratory study involved only three hours of instruction.

2. The teacher: Teacher is expected to have knowledge of programming concepts such as recursion, and intensive training in the programming language.

3. Teaching method: Explicit teaching of the language generally facilitates skill acquisition. While indirect instruction is more likely to result in transfer.

4. Learning as apprenticeship and collaborative activity: Research found that programming particularly in elementary school classroom is often a
collaborative process. And the use of borrowed code is crucial and an integral part of the collaborative process.

5. Culture within the school: The school needs to foster a culture in which computers play significant roles in helping students and teachers to engage in higher-order thinking.

6. Relevance of the programming activities: Some functional significance needs to be attached to the programming activities to result in successful, transferable learning.

7. Pupil interest in programming: Programming is an important and deeply personal intellectual activity for those students who are interested in it.

8. Pupil age: Children from the age of pre-school and above have been reported in research that programming activities were a significant experience to them.

9. Administrative factors: This includes the supports from the principal, community, school policy statement, teaching staff and computing expert, and computers which pupils have a high level of time to access and interact with integration to curriculum.

Hoyles, Noss and Adamson (2002) also point out three practical objections to the idea of children programming computers: (a) programming takes too much time in what is already a crowded mathematics curriculum; (b) programming is too hard for teachers as well as students; and (c) programming diverts attention from the underlying knowledge goals. These three objections were partly addressed in Oakley and McDougall (1997). Hoyles et al. are, however, posing a new question, namely: Does it still makes sense to insist on programmability as a vehicle for creativity and constructionist learning, along the evolution of programming and the meaning of microworlds?

Oakley and McDougall (1997) noted that from different periods in the development of educational computing, certain features such as graphical user interface (GUI) widgets (e.g., slider, button) and graphic tools have broadened the possibilities for cognitive development, which might change the nature of programming language. Indeed, early programming activities involved mainly pure text-based coding, which has been the main source for the claim of time-consuming,
difficult and attention shifting in programming. Hoyles et al. (2002), however, do not dispute the value of programming in learning mathematics. They still remain convinced that the textual side of programming environments play an imperative role of linguistic formalisation in aiding learners’ mathematical expression. Nevertheless, in the evolution of computer programming, they give a slightly altered role to programming, namely to serve as the symbolic representation of a mathematical function as well as the glue that bound all the representational modes together. Hoyles et al. argued that a mathematical microworld has a symbolic core consisting of multiple representational modes, and programming language serves as one of them to link and interact with other modes such as GUI entities and visualisation. Moreover, they propose a bi-directionality between the different representational modes. For example, one can use programming language to manipulate graphical objects, or use a mouse or other means to directly manipulate graphical objects whilst the programming codes are generated at the same time.

2.4.4.3 The case of Logo

The first Logo language was invented (as a distant relative to Lisp language) by Feurzeig and colleagues in late 1960s as the first programming language for children. Papert (1980) then extended Logo to include “turtle graphics”, in which children learn geometrical concepts by moving a “turtle” around the screen (Bruckman & Bandlow, 2002).

There are differing versions or dialects of Logo developed in the last 2 decades such as Comenius Logo, Imagine Logo, Logo Gráfico, LogoWriter Win, LCSI Microworlds, Terrapin Logo, VLogo, WIN-Logo, LEGO-Logo, Elica, Geomland, A Logo Interpreter in Scheme, Logo in Scheme, LogoFE, MSWLogo, NetLogo, P-Logo, rLogo, StarLogo, TinyLogo, and UCBLogo (see (Logo Foundation, 2000)). There were also GeoLogo (Clements & Meredith, 1993), MultiLogo (Resnick, 1991), OpenLogo (Blaho, Kalas, & Tomcsanyi, 1999), Turtle Tracks (Azuma, 1999), and YoYo (Media Laboratory at MIT, 2001). In general, the Logo learning environment consists of a 2D drawing window, an immediate command centre, and a procedure editor window. The various versions of Logo also contributed to the improvement of Logo environment along the evolution of computer technology, such as:
1. Graphical User Interface (GUI) with visual widgets
2. Integration with multimedia and hypermedia
3. Network and Internet accessibility
4. Cross platform implementation
5. Language extension
6. Object Oriented Programming (OOP)
7. Parallel computing
8. Semi three dimensional functionality

The integration of these technologies into the Logo environment has broadened and deepened Logo’s mathematical applications. Some of these technologies seemed to push the Logo from 3GL to 5GL.

Fay and Mayer (1988) identified two major components of programming language namely syntactic knowledge and semantic knowledge in Logo. The syntactic structure of programming language consists of three components: the elements, the rules for combining elements into commands, and the rules for combining commands into programs or program segments. The elements include keywords, punctuation marks, and arguments. In Logo for example, the keywords are FORWARD, BACK, LEFT, and RIGHT; the punctuation marks are spaces, parentheses, and double quote; and the arguments are numbers or variables that store numbers. The rules define the sequence or order for elements. In Logo, the rules for combining elements into commands are:

```
KEYWORD space ARGUMENT
```

And the rules for combining commands into programs are:

```
TO procedure-name [variables list]
    KEYWORD space ARGUMENT
 .........
END
```

Fay and Mayer (1988) found that the acquisition of the Logo syntax can be difficult for some children and it may require extensive practice and experience before mastery will occur. A study (Heller, 1986) showed that after 3 weeks of Logo experience, the mean percentage correct on a Logo syntax test was about 48%, and after 12 weeks, performance rose to 65% correct.
The semantic knowledge of programming language has a logical structure that is dependent on the functional components of the programming domain. There are three functional components in programming language: (a) operations, which are the actions that can be performed; (b) objects, which are the entities that are operated upon; and (c) locations, which refer to where in the computer the operation occurs (Fay & Mayer, 1988). Fay and Mayer elaborated that an operation is comprised of two subcomponents: an action and an action modifier. In Logo, there are four elementary actions: clockwise rotation, counter clockwise rotation, forward movement, and backward movement, which correspond with the keywords RT (Right), LT (Left), FD (Forward), and BK (Back) respectively. The action modifier is the numerical argument that accompanies the action. The second functional component – objects - in Logo refers to the turtle. The turtle has spatial features (e.g., front, back, left side, and right side) and a current state (e.g., orientation and position). A Logo command “RT 90” can be then interpreted as “rotate the turtle 90 degrees to its right” or as “rotate the turtle 90 degrees clockwise from its current orientation.” The third functional component of semantic knowledge is location, which refers to where the operation on the object is to take place. In the Logo graphics environment, the most common location is the computer screen. Or it can be the floor if a robotic turtle is used. Nonetheless, both of them are dealing with a 2D plane.

Fay and Mayer (1988) indicated that acquiring semantic knowledge could be difficult for two reasons: The learner lacks the necessary conceptual skills (e.g., the egocentric bug in young children in which they cannot recognise the turtle’s right corresponds to their left when the turtle is in 180 degree orientation), or the learner inappropriately maps the meanings that the keywords have in everyday language onto the command in Logo environment. However, they did point out that repeated experiences in Logo could improve learners’ acquisition of the semantics in Logo.

Fay and Mayer (1988) argued that much of the impetus for teaching Logo to children is based on the claim that experience with Logo could enhance general and high-level thinking skills. And this phenomenon is often termed as transfer, which means the knowledge acquired in one domain is applied, without the benefit of learning or experience, to a novel domain. They proposed that there are two conditions that must be met in order for transfer to occur. The first is the relevant
syntactic and semantic knowledge components must be acquired to some degree in the original learning environment, and the second is the knowledge acquired in the original domain must be recognised as relevant for the new domain.

Fay and Mayer also indicated that during the teaching of programming languages such as Logo, syntactic knowledge should be taught prior to semantic knowledge. However, a review of the research literature has failed to locate unequivocal support for this assertion. Furthermore, the veracity of Fay and Mayer’s claim runs counter to the researcher and many other programmers’ experiences in learning programming languages. During conversations with other computer programmers, he has found that without exception, they all claim that they found that understanding the semantics of a programming language was a necessary condition for the learning of the programming languages syntax.

2.4.5 Virtual Reality

Virtual Reality (VR) constitutes a perfect oxymoron, and was first coined by Jaron Lanier in 1980s. In a recent interview by Sun Microsystems, Jaron Lanier expresses his own idea that “Virtual Reality is a somewhat broad idea, and the definition isn’t fixed. It means different things to different people” (Sun Microsystems Inc., 2003). Indeed, many scholars have given a definition for “Virtual Reality” to fit their particular research domains (M. Bricken, 1990; Bricken & Byrne, 1992; Byrne, 1996; Davis et al., 1996; Hedberg & Alexander, 1994; Heudin, 1999; Macpherson & Keppell, 1998; Moore, 1995; Osberg, 1993, 1997; Winn & Bricken, 1992). In the field of educational research, VR will probably be defined as “a computer generated, multi-dimensional space, which allows users to move viewpoints freely in real time, interact and communicate with entities in the space, gain the real-world-like experience, and thus change cognition.”

VR technology has also been classified into various categories such as cab simulators (e.g., flight simulators), projected reality, artificial reality, augmented reality, tele-presence, desktop VR, and visually coupled displays. (Pimentel & Teixeira, 1995) or simply non-immersive, semi-immersive, and fully immersive VR (Kalawsky, 1997). No matter what kind of VR is used, the very core idea of using VR in education is encapsulated by the following statement from Lanier:
What I envision is not so much a pre-programmed virtual world that you might play as a game, but rather a virtual world that you can change from the inside that people use as a form of expression in which they're creating things together.

Virtual Reality Pioneer – Jaron Lanier
(Sun Microsystems Inc., 2003)

VR technology has been so profound that it is beyond the scope of this research to do an extensive review of it. Some most important aspects of the value of VR in education have been addressed in Section 1.2, and some of VR’s applications in mathematics education with their advantages and critiques in Section 2.3.3.2. This section aims at targeting the specific VR technology (which is desktop VR) that was utilised in this research study and analysing the special features relevant to this research knowledge domain (which is the knowledge construction of 3D geometry concepts and processes).

2.4.5.1 Scope

Considering the affordability in current educational and technological contexts, desktop VR was the ideal VR technology for this research study. There are also a few technologies available for desktop VR such as Virtual Reality Modelling Language (VRML) and Java3D. VRML⁴ (pronounced as vermal) (Web3D Consortium, 2003a), was used to implement the Virtual Reality Environment (VRE) in this research study.

The current version of VRML contains two parts: VRML97 (ISO/IEC 14772-1:1997) and VRML97 External Authoring Interface (EAI, which is a set of Java API) (ISO/IEC 14772-2:2002). According to its specification,

VRML is a file format for describing interactive 3D objects and worlds. It is designed to be used on the Internet, intranets, and local client systems. VRML is also intended to be a universal interchange format for integrated 3D graphics and multimedia. VRML may be used in a variety of application areas such as engineering and scientific visualisation, multimedia presentations, entertainment and educational titles, web pages, and shared virtual worlds.

(Web3D Consortium, 2003a)

⁴ VRML is an international standard (ISO/IEC 14772) file format for describing interactive 3D multimedia on the Internet.
VRML has great potential in education. It has some special features that are distinct from traditional 2D graphics, and has fundamental influences upon the teaching and learning of 3D geometry that need to be addressed. In addition, the employment of VRML also sets the limitation to the VR power within its current development.

2.4.5.2 General environment

A VRML file defines a 3D space through a 2D rectangle on the computer screen. Although the specification of VRML specifies the conformance and minimum support requirements of the 3D space, current implementations of software (VRML browsers) pushes the limits of the hardware. Therefore, the 3D space defined in VRML has almost infinite space and can contain as many objects (e.g., geometrical objects, lights and dynamic information) as the computer hardware resources (e.g. memory) can support.

Since the 3D space of VRML is to reflect the reality, the pixel on traditional 2D screen is no longer valid in 3D space. The unit of distance described in VRML is meter. This is advantageous in terms of authentic environment in which students experience the virtual space just like the real world setting. However, it is also possible that students will be conflicting with their mental model if they have previously built upon the use of pixel. And if they want to specify some thing smaller than one meter, the decimal expression has to be used.

In VRML’s virtual space, a Cartesian coordinate system (see Figure 2.20) is used to maintain the location of geometrical objects. The x-axis can be seen as the width, the y-axis the height, and the z-axis the depth.

![Figure 2.20 Coordinate system in VRML space](image-url)
The default viewpoint in VRML’s virtual space is at the position (0,0,10) looking towards negative z direction. The perspective of looking towards some direction is called orientation of the viewpoint. Any other geometrical objects in this space have a location called translation, as it is related to some hierarchical levels of objects. All geometrical objects have also an orientation called rotation, which is to distinguish object from subject (viewpoint of the user). This semantics of orientation in VRML space informs the important idea of being subjective or objective, a notion which will be discussed in Section 2.4.5.3.

VRML allows users to have multiple preset viewpoints in a virtual space. Users can then travel among those viewpoints and get a better pre-defined perspective of the virtual space. One important attribute of viewpoint is the field of view. The field of view specifies a preferred minimum viewing angle from the viewpoint in radians. A small field of view roughly corresponds to a telephoto lens; a large field of view roughly corresponds to a wide-angle lens. The field of view shall be greater than zero and smaller than π. The default value for the field of view in the virtual space is 0.785398, which is close to normal human vision. The changing of the field of view may be valuable as it creates the effect of viewing from different optical conditions. For example, a cube will be distorted or stretched to a rectangular box and two parallel lines may cross over if under a large field of view.

2.4.5.3 Avatar

Avatar is used as a metaphor to represent oneself in a multi-user virtual world. An avatar is a virtual person who can be any kind of creature or non-creature such as a robot, animal or person (see Figure 2.21). The free use of avatars can bring great enjoyment, motivation, and a sense of security for users to participate in virtual worlds.

![Avatars in a multi-user virtual world](image)

*Figure 2.21 Avatars in a multi-user virtual world*
In the virtual space, users normally only see other users’ avatars because they are in the avatars’ viewpoint. However, it is also possible that users can jump out of the avatar’s viewpoint and see themselves. The ability to imagine seeing the virtual world from an avatar’s viewpoint (eyes) has been termed as “first-person” imagery, while seeing oneself from the outside as an external observer was termed “third-person” imagery. It was suggested that giving opportunity to switch between first- and third-person imagery might be of great benefit for the virtual traveller to anticipate new vantage points and appropriate action (Amorim, Trumbore, & Chogyen, 2000). When users are seeing themselves in virtual worlds, they are monitoring themselves from an objective viewpoint. This may improve users’ metacognition and reduce the workload of first-person imagery.

2.4.5.4 Objects

VRML can create any possible geometrical objects that one can imagine. Basic geometrical primitives supported include the Box, Cylinder, Cone, Sphere, and Text. For arbitrary models, VRML provides PointSet, IndexLineSet, and IndexFaceSet. It also provides ElevationGrid and Extrusion methods for easy constructing terrains and extrudable objects. Geometrical objects in VRML are very rich in the expression of colours and textures. The colouring model in VRML was described in Section 2.3.1.1. Any surface of geometrical objects can be wrapped with images, animated images, and even movies. A sphere wrapped with an image of earth will closely resemble the planet earth.

In addition to geometrical objects, natural objects without any physical occupation of space such as light, sound, background, fog, and time can also be described in VRML. Hence, the virtuality is very much enhanced with the complex interactions among the lighting and colouring models, 3D sound, and geometrical objects.

Most objects in VRML can be grouped and constrained in transformation hierarchies. Thus, geometrical objects can be scaled, translated, rotated, and orientation-scaled to form topological transformations and more complex geometrical relations.
2.4.5.5 Behaviours

The current standard VRML has a former name as “Moving World” as it describes some dynamic aspects of the real world. Some properties of geometrical objects such as translation, rotation, and scale can be interpolated to perform movements. Objects in VRML are able to send and receive events. Thus, objects can inform the other objects that they are performing actions while reporting values (e.g., position, orientation), and be informed by others with others’ current status.

One mechanism to aid performance behaviours is a sensor. Sensors act as triggers to report information, invoke behaviours, and interact with a pointing device (e.g., a mouse or wand). There are nine sensors: Anchor, which adds a hyperlink to an object; Collision, which prevent the user from colliding into objects when navigating; CylinderSensor, which maps pointer motion into a rotation on an invisible cylinder; PlaneSensor, which maps pointing device motion into two-dimensional translation in a plane; ProximitySensor, which generates events when the viewer enters, exits, and moves within a region in space (defined by a box); SphereSensor, which maps pointing device motion into spherical rotation about the origin of the local coordinate system; TimeSensor, which generates events as time passes for driving continuous simulations and animations, controlling periodic activities (e.g., one per minute), and initiating single occurrence events such as an alarm clock; TouchSensor, which tracks the location and state of the pointing device; and VisibilitySensor, which detects visibility changes of a rectangular box as the user navigates the world.

These sensors offer the possibility for users to directly manipulate and interact with objects, and perceive continuous spatial visualisation in virtual space. The virtual space hence becomes a dynamic instead of a static world. The full utilisation of these sensors and behaviours can create realistic simulations. However, it also creates complexity which may lead to increased cognitive load. The concept of objects sending and receiving events to perform behaviours is a programmer’s mental model, which must be introduced and mastered by learners or designers before they can take advantages of these computational functionalities. Moreover, the predominant use of a 2D point device – mouse – in this 3D desktop VR environment, will possibly create the fundamental mismatch in dimensionality (Leach et al., 1997).
2.4.5.6 Navigation modes

Navigation in 3D VR space is another kind of direct manipulation. However, unlike other forms of direct manipulation (e.g., dragging the objects), this form of manipulation changes not the object itself, but the views of the user; the navigation in 3D space allows users to move around in the virtual world and change their position and orientation. Thus, users can create any possible perspectives from any viewing angle and perceived visual orientation. Because it is three-dimensional space, the common 2D pointing device (e.g., mouse) in desktop VR is not sufficient to express the dimensionality. For example, most desktop VR software will match the mouse’s xy dimension to xz dimension in 3D space. Therefore, novice users find it difficult to operate on the y dimension. To pertinently navigate in 3D space requires some special devices such as SpaceMouse or SpaceBall (Figure 2.22). However, the current cost of such a device can be as much as one personal computer, which is not affordable at this moment but may be feasible in the future.

![SpaceMouse and SpaceBall](http://www.3dconnexion.com/)

*Figure 2.22 SpaceMouse and SpaceBall*
(Source: http://www.3dconnexion.com/)

To solve the problem of the mismatch in dimensionality, the 2D mouse is used in combination with some function keys on keyboard. Therefore, the mastery of using the 2D mouse and the keyboard to navigate in 3D space is considered a necessity in this research study.

There are five navigation modes described in VRML specification:

1. Walk: This mode simulates the walking experience in real world. Navigation in this mode is good for exploring a virtual world as if walking on foot or driving in a vehicle that rests on or hovers above the ground.
The gravity effect will apply in Walk mode and thus create the terrain following and gravity experience.

2. Fly: This mode simulates the flying experience in real world. Navigation in this mode is similar to Walk mode except that terrain following and gravity are disabled or ignored.

3. Examine: Navigation in this mode is for viewing individual objects and often includes (but does not require) the ability to spin around the object and move the viewer closer or further away.

4. Any: User can freely change among the three navigation modes above.

5. None: Navigation is disabled. All browser-specific navigation user interfaces are removed, forcing the user to navigate using only mechanisms provided in the scene.

In normal VRML browser’s user interface, users can use some keys on the keyboard to perform a third dimension operation. For example, in Cortona\(^5\) VRML browser, dragging the mouse straight forward to walk forward in Walk mode will become “pitch up” while space key is pressed.

Navigation in 3D space may be valuable for gaining spatial ability. However, it is also a worry that users may become disorientated in the space (Salzman, Dede, Loftin et al., 1999). Hence, VRML browsers often provide special orientation functions such as “align to horizontal”, “travel among pre-set viewpoints”, “restore previous viewpoint”, and “fit all objects in view.” For a better VR learning environment, it may also be a good strategy to inform the users about their position and orientation by adding a grid to tell the distance, a compass to tell the direction, some landmarks to relate position, and a 3D arrow to tell the coordinate.

### 2.4.6 Design of technology-rich learning environments

There has been a strong appeal for educational reform and integration of technology into mathematics curriculum because of the important role that technology can play on teaching and learning (Cuoco & Goldenberg, 1996; Heid, 1997; Roschelle et al., 2001). However, the mere presence of computers in classrooms does not guarantee efficacious teaching and learning. Furthermore,

\(^5\) Cortona is a product of ParallelGraphics. See http://www.parallelgraphics.com/products/cortona/
although there is much software for learning developed in the form of advanced technologies such as multimedia and hypermedia, many of them are criticised as only fancy textbooks. Papert (1996) commented that 99% of the software for “learning math” in the market was teaching “junk math” because its idea about what math is, and why the kids should learn it is flimsy. Therefore, a technology-rich learning environment needs to be well designed, and good design is informed by theory (Wilson, 1999).

2.4.6.1 Assumptions and principles of design

Battista (2001) stated that there are three basic assumptions in designing technological tools for teaching mathematics:

1. Technological tools for teaching mathematics must be reasonably easy for students and teachers to use. The technological interface must quickly recede into the background so that the focus of attention is on doing mathematics, not operating the technology.

2. Because use of technological tools is an important component of modern applications of mathematics, there are clear advantages when students’ use of technological tools for teaching mathematics helps them learn to use technological tools for doing mathematics.

3. Most importantly, technological tools for teaching mathematics must support instruction that is consistent with current professional standards for teaching mathematics and with modern research on mathematics learning.

Clements and Battista (1994) took a functional viewpoint and drew some implications from research studies for the design of geometrical computer environments for geometry education, in which the environment should: (a) allow students to build on their visual strengths to more powerful geometrical thinking; (b) connect representations between visual and symbolic modes bi-directionally; (c) expand primitives and tools to facilitate students’ construction of geometrical notions and increases analytical, rather than visual approaches; (d) facilitate the processes of change and exploration by providing functions such as undo and redo; (e) enable the creation of procedures for students to reflect upon; (f) balance freedom within
constraints; (g) plumb the depths of simple tasks; and (h) explicitly underlie some pertinent theoretical models.

In a similar vein, Hoyles (2001) noted that if we want to design investigative environments with computers that will challenge and motivate children mathematically, we need software in which children have some freedom to express their own ideas, but are constrained in ways so as to focus their attention on the mathematics.

Oliveria (2000) proposed a set of semiotic principles for underlying the concept of interface design: (a) the interface should be understood as a group of entities that communicate, one or more of which are human beings; (b) every entity in the interface possesses a semiosis capacity perceived by the human; (c) every entity in the interface has a possibility of sign emission perceived by the human; and (d) the language used by the entities of the interface emerges from the conviviality among them. The last principle in Oliveria comes from semiotic viewpoint of interface design, and resonates Hoyle’s (2001) critical idea of motivating and focusing students’ attention.

Roblyer, Vye and Wilson (2001) set the following criteria for the future learning software, in which the software: (a) is Internet deliverable or Internet enhanced such as web-based learning environments; (b) emphasises visual and three-dimensional problem solving; (c) has the availability of visualisation or model building; (d) provides “rich” learning resources (e.g., hyperlinks to locate resources); and (e) has more apparent relative advantages that can be provided by an alternative, or take the place of previous methods of teaching and learning. These criteria make the World-Wide-Web an ideal platform for developing learning software.

Campbell (1999) proposed six conceptual frameworks for the design of web learning environments. They are: (a) multiple representations of reality, (b) authentic tasks, (c) real-world, case-based contexts, (d) fostering reflective practice, (e) knowledge construction, and (f) collaborative learning. These frameworks are actually in line with constructivist principles of learning, which informs that the design of technology-rich learning environments and learning theories are two sides of a coin, corresponding closely.
2.4.6.2 Process of design

A traditional research approach where all functions and conditions are controlled and the treatment varied is inappropriate for most research investigating information technology’s role in the reform of schooling (Hawkins & Collins, 1992). As Bruckman (2002) wrote:

The design of any piece of technology intended for human use – whether for entertainment, work, or education – is ideally iterative and user-centered. Designers cannot anticipate all the needs of users, but most begin with a prototype and revise it based on user feedback. This is even more true of online learning communities, where designers must understand the needs not just of individual users, but of groups of users and their complex inter-relationships, as facilitated by the technology. Designers begin with theory, prototype, test, and then revise. However, it is not just the technology that can be revised, but also the underlying theory. Technological design and pedagogy have the potential to co-evolve in this new medium6.

(Bruckman, 2002)

Bruckman’s notion of “iterative design” and “co-evolve” of technology design and pedagogy (or theory) are analogous to the notion “design experiments” or “design-based research method”7 generated from a group of researchers and educators (Bannan-Ritland, 2003; Bell, 1998; Bereiter, 2002; Brown, 1992; Cobb, Confrey, diSessa, Lehrer, & Schauble, 2003; Collins, 1993; Hawkins & Collins, 1992; Hsi, 1998; Kelly, 2003; Lobato, 2003; McCandliss, Kalchman, & Bryant, 2003; The Design-Based Research Collective, 2003) that has been reviewed in Section 1.4.

There is a consensus within the research literature that the process of designing technology-rich learning environments is an iterative one which begins with a prototype set up by the original designers with some informing theories. Once the prototype is taken into practice in community, the iterative process begins. During the iterative process, the whole environment (prototype) is changing through the enactment of multiple sources of input such as students, teachers, parents, classroom ethos, theories and new technologies. These multiple inputs can simultaneously serve as designers, researchers, observers, actors, and/or practitioners

---

6 The new medium refers to the MOOSE Crossing project, which began to develop in 1992 in Georgia Institute of Technology, as an exploration of the application of constructivist learning theory to the design of an online learning community.

7 According to The Design-Based Research Collective (2003), these two terms are different names for the same research methodology.
as their roles are also changing at times. The change of the environment can generate implications for revising and building theories. New goals for design and research can also emerge because of new potentials being discovered during the course of design research, and so are new practical applications and dissemination.

2.4.6.3 Evaluation of design

According to the design-based research methodology, the evaluation of design is a retrospective analysis. Cobb, Confrey, diSessa, Lehrer, and Schauble (2003) pointed out that a primary aim when conducting a retrospective analysis is to place the design experiments in a broader theoretical context, thereby framing it as a paradigmatic case of the more encompassing phenomena specified at the outset. It involves systematically working through the extensive, longitudinal data sets such as discourses, designed artifacts, activity structures, institutions, scaffolds, and curricula generated in the course of the design experiments. This claim is promising, however, it is also challenging.

Usability for industrial engineering in producing learning software is a domain of knowledge concerning human factors with computers. Conyer (1995) defined that usability evaluation is the analysis of the design of a product or system in order to evaluate the match between users and a product or system within a particular context. Usability is a dynamic process throughout the life cycle of a product or system. This domain of knowledge could provide explicit guidelines for the evaluation of design in design experiments.

Nielsen (1993) listed five “usability attributes”, which formed the foundation for usability inspection methods:

1. Learnability: Ability for users to learn the system easily.

2. Efficiency of use once the system has been learned: Ability for users to save time in their work once they’ve learned the system.

3. Memorability: Ability for users to come back to the system and remember how to use it once they’ve been away from it for some time.

4. Error recovery & prevention: When the system presents an error message to users, it gives enough information for them to be able to continue with their work. Better yet, the system helps to prevent errors.
5. Subjective user satisfaction: Users’ overall feelings about the system. Is it pleasant to use?

There is a number of usability inspection methods and data collection methods well documented in the literature. Conyer (1995) summarised six usability inspection methods:

1. Heuristic evaluation: This method uses a predefined list of usability principles to identify usability problems so that they can be attended to in an iterative design process.

2. Pluralistic walkthroughs: The goal of this method is to systematically review the usability of an interface and its flow from a task-based user-centred perspective whilst at the same time considering the design constraints.

3. Formal usability inspections: Usability issues are reviewed within the context of specific user profiles and defined goal-oriented scenarios by applying a task performance model and heuristics.

4. Empirical methods: Data are collected in an experimental test to prove or disprove a hypothesis. For example, the number of correct responses and errors made by a user under controlled conditions.

5. Cognitive walkthroughs: These are used to evaluate the ease of learning to use a product, particularly by exploration. The method is a formalised way of imaging people’s thoughts and actions when they use a product interface for the first time.

6. Formal design analysis: This is based on the premise that understanding of the requirements of the task to be performed is the key to understanding behaviour.

Conyer (1995) also outlined six data collection techniques for those methods:

1. Verbal report: Users provide a verbal report soon after completing their evaluation. This information can then be informally reviewed or formally classified into categories for evaluation.

2. Concurrent think-aloud method: Evaluators verbalise their thoughts while interacting with a product. The purpose of this method is “to show what
and why the users are doing it while they are doing it, in order to avoid later rationalisations."

3. Questionnaire: Questionnaires can be composed of items that address information and attitudes. It is important to keep questions specific rather than general and to ask questions about actual product experience rather than hypothetical questions about possible product changes.

4. Video analysis: One or more videos can be used to capture data about user interactions. For example, in a software usability evaluation, three different video cameras could be used to capture the user's keyboard actions, the screen activity and the user's verbal and non-verbal responses. Video Analysis is then used as a tool in the process of interpreting what usability problems occur and why. Even more powerful is to use the video to create a multimedia document, which includes annotations of the usability problems.

5. Auto-logging programs and audit trails: Auto-logging programs can be used to track user actions with respect to duration and frequency of use, like number of keystrokes, button clicks, and requests for help, duration and path of errors.

6. Software support: Software can be designed to support the evaluator during the evaluation process and to provide an assessment summary. For example, during the evaluation of a system user interface, the test items are presented on the screen with accompanying usability criteria and a rating scale. The evaluator selects the usability criteria, giving each a rating and writes an explanation of each rating. The software calculates an average mark for each criterion and sorts the results by usability components.

Karat, Campbell, and Fiegel (1992) provided guidelines for considering the usability of educational software:

1. Use a simple and natural dialog.
2. Provide an intuitive visual layout.
3. Speak the user’s language.
4. Minimize the user’s memory load.
5. Be consistent.
6. Provide feedback.
7. Provide clearly marked exits.
8. Provide shortcuts.
10. Allow user customisation.
11. Minimize the use and effects of modes.
12. Support input device continuity.

These guidelines can serve as the starting points for the principles to be used in the design and implementation, and evaluation phases of the research study.

2.4.7 Synthesis

Section 2.4 has explored issues of computational ideas and paradigms including the nature of the use of technology, the educational roles for computers, the microworlds metaphor, the programming paradigm, the desktop VR technology with particular reference to VRML, and the design of a technology-rich learning environment.

The use of technology in mathematics education is not a panacea but without doubt valuable and potential. Technology can visualise abstract mathematical ideas, enhance students' learning opportunities, and focus students on decision-making, reflection, reasoning, and problem solving. However, it can also divert students’ attention from mathematics to technology itself. Key factors identified regarding the effective use of technology involve the following seminal thoughts:

1. Technology is misused if it is seen as only a computational tool or a replacement of teachers (Heid, 1997).

2. Successful use of computers in learning involves 1) active engagement, 2) participation in groups, 3) frequent interaction and feedback, and 4) connections to real-world contexts (Roschelle et al., 2001)

3. Going beyond the amplification of cognition to the reorganisation of mental functioning (Pea, 1985).

In addition, Taylor’s (1980) three modes of using computers suggested that the “computers as tutee” mode of using computer could bring much more benefit to learning than as tutor or tool. Bruckman (2002) also highlighted that the CSCL mode of using computers to address the importance of collaborative learning. All thoughts for effective use of technology are evident to be consistent with constructivism, constructionism, and social constructivism. Therefore, a consensus has emerged that technology should be treated more like a thinking, building, and communicating tool that promotes and engages students in new ways of thinking, doing, creating, and sharing new ideas within communities.

So far the perspective of using technology has been set. The next step we will need to know is what technologies are available and what kind of technologies can achieve those thoughts above. Technology is able to create or simulate a coherent domain of objects and activities in which mathematical ideas are permeated through. The coherent domain has been termed as “Microworld.” Microworld is the ideal vehicle for developing mathematical knowledge because it is the embodiment of mathematics (L. D. Edwards, 1995). Microworlds are dynamic in nature, therefore should allow interventions to be made to change themselves. The programming language is a powerful means to manipulate and change microworlds.

Programming language serves as the role to link multiple representations such as formal language, mathematical symbols, visual graphics, and kinaesthetic actions in microworlds. However, traditional programming language, based on a text only interface, has been found to be time consuming and difficult for young children to operate. The review of research literature indicated that along the evolution of programming language, a 5GL with visual aided interface and a functional style of programming language might ease the difficulties in programming. Logo is a programming language, which has had a long history of educational research. Its syntax is easy to understand for children and adults, and its semantics is full of geometrical and mathematical sense. Many computational paradigms such as a visual interface, multimedia and hypermedia, network, and parallel computing have been integrated into Logo. This made Logo the ideal language for this research study.
In addition to programming language, another major component of microworlds is the visual or graphical component. VR technology with VRML can fulfill the requirements of this component, particularly for 3D geometry. VR of VRML reflects the dynamic nature of the real world. However, this computational reality raises some issues different from traditional 2D visualisation, which are considered as important pre-requisites for learning. These issues include the understanding of:

1. the general environment of VR such as the distance is expressed as meters or centimetres (not pixels), the coordinate system or direction;
2. avatar in which the users can be subjectively moving or objectively observing in VR worlds;
3. describing geometrical objects and behaviours of virtual worlds; and
4. using keyboard and mouse to navigate in 3D virtual worlds.

The Logo language and VRML have formed a powerful combination for a mathematical microworld. These technologies, however, need to be implemented as informed by relevant theories. Research literature identified the iterative nature of designing a technology-rich learning environment, which is termed as “design experiments” or “design-based research” (The Design-Based Research Collective, 2003). Therefore, this research study began with a “prototype” of a microworld and a “prototheory”, and then both of the “prototype” and “prototheory” evolved through the iterative processes of practice in community. Moreover, the usability evaluation was embedded as a dynamic process throughout the iterative cycle of the development of the VRLE.

2.5 CONCEPTUAL FRAMEWORK FOR THE RESEARCH STUDY

The literature review has explored three threads of thought including an understanding of the domain knowledge (e.g., mathematics, 3D geometry and mathematics education), an understanding of the learner (e.g., cognitive development and spatial abilities), and an understanding of computational ideas and paradigms (e.g., microworlds, programming, and VR technology) (Resnick, 1996). These three threads have revealed many interrelated issues involving the fields of philosophy,
From this literature review, a two-facet conceptual framework to inform the research being conducted in this study was generated. The two facets contained within this conceptual framework are:

1. Specifications to inform the design of a VRLE for 3D geometry; and
2. Perspectives to inform the evaluation of student learning of 3D geometry within the VRLE.

2.5.1 Specifications to inform the design of the VRLE

From the literature review, it has been possible to distil a set of guiding assumptions to inform the design of the VRLE. These assumptions can be classified into four categories: (a) epistemological assumptions, (b) learning/instructional assumptions, (c) assumptions about 3D geometry, and (d) assumptions about the use and the role of technology.

Epistemological assumptions

The following set of four epistemological assumptions was derived from the analysis and synthesis of the research literature from the fields of mathematics, mathematics education, and semiotics:

1. Mathematics is fallibilist in nature.
2. Mathematics is a meaning-making activity of human kind.
3. There are multiple ways of knowing mathematics.
4. There are multiple perspectives and representations of mathematics.

Learning/Instructional assumptions

Consistent with the four epistemological assumptions listed above, the following learning/instructional assumptions were generated from the analysis and synthesis of research literature from the fields of learning and development, and mathematics education.

1. Learning and instruction in mathematics should focus on meaning-making.
2. Learning and instruction based on constructivist/constructionist viewpoints are the most efficacious means of ensuring meaning-making occurs during mathematical learning activities.

**Assumptions about 3D geometry**

The following assumptions were derived from the fields of semiotics, psychology, mathematics education, and technology.

1. From a semiotic viewpoint, 3D geometry involves three indispensable components: external material world (objects), internal spatial ability (interpretants), and communication (signs).

2. The development of the knowledge of 3D geometry requires familiarity with and the coordination of these three frames of reference: egocentric, fixed, and coordinate system.

3. Multiple paradigms of using computers should address some limitations of traditional teaching and learning of 3D geometry; these paradigms include the focus on language (programming), the manipulation of dynamic geometrical objects, real-time 3D graphics, and web environments.

**Assumptions about the use and the role of technology**

The following assumptions are consistent with learning/instructional and 3D geometry assumptions above.

1. The most effective use of technology is when technology is used as a thinking tool rather than computational tool, for the reorganisation of mental functioning rather than the amplification of cognition, and for knowledge building rather than knowledge reproduction.

2. Computer microworlds combining real time 3D visual graphic component (virtual reality) and programming language component (Logo) could form powerful embodiments of mathematics and 3D geometry.

3. Computer microworlds in combination with CSCL environments could act in synergy to enhance the social construction of knowledge about 3D geometry.
4. Sustained educational innovation using technology requires the framework of design experiments or design-based research methodology that continuously advances both theories and instructional design for bringing about new forms of learning.

2.5.2 Perspectives to inform the evaluation of student learning of 3D geometry within the VRLE

Most traditional ways of thinking about, and teaching and learning geometry tend to focus particularly on the content of geometry (e.g., notions of shape and structure, transformation and symmetry, and location and arrangement). However, most of these traditional ways of thinking about, and teaching and learning geometry have been generated within contexts quite different to that produced within the context of a VRLE.

Unlike most other previous contexts that have been generated to facilitate the teaching and learning of geometry, the VRLE is based on a semiotic scheme in which the domain of geometry is categorised into these three components: external material world (objects), internal spatial abilities (interpretants), and the communications (signs) that mediate between them (see Section 2.3.1). Also, the VRLE has been designed to provide students with the possibility of exploration within a 3D virtual environment quite different to that provided by traditional classroom geometry learning activities of computer-based activities such as Logo.

Therefore, it was envisaged that the VRLE would provide students with new ways of thinking and doing 3D geometry. Because of this, it was hypothesised that traditional perspectives for evaluating learning of 3D geometrical concepts and processes would need to be augmented by modified and/or additional perspectives in order to accurately evaluate how and what students learn about 3D geometry as they interact with the VRLE. Perspectives for evaluating learning that were investigated within this study were:

1. How do students’ spatial abilities as defined by McGee (1979), Hoffer (1977), and Bishop (1983) (see Section 2.3.1.2) change and develop within the 3D VRLE? (PE1: Perspective for Evaluation 1)

2. How does the ability to interact and link within real time 3D visualisation (VR interface, topological semiotics), procedural language communication
(Programming interface, typological semiotics), and hypermedia forum (Social semiotics) during the construction of 3D geometry artifacts influence how and what students learn about these artifacts? (PE2)

3. How do students use and understand the use of the different frames of reference (e.g., egocentric, fixed, and coordinate systems) to effectively and/or intentionally move the turtle to construct geometrical objects? (PE3)

More particularly:

4. How does the ability to engage in 3D transformations of turns, rolls and tilts impact on students’ knowledge about the geometrical concepts such as similarity and congruence? (PE4)

5. How does the ability to coordinate and integrate the turtle movements and navigation in 3D space influence students’ construction of geometrical objects? (PE5)

6. How does the ability to explore 3D objects from different viewpoints or perspectives within the VRLE influence students’ conceptions about 3D geometrical objects? For example, how does the fact that the square faces of a cube may not look like squares in many viewpoints in VRLE’s 3D space influence students’ conceptualisation about the nature of a cube? (PE6)

7. How does the ability to distinguish the integrated movement (e.g., moving while turning in real situation) and component movement (e.g., moving and turning are separate in turtle geometry) in students influence their construction of 3D geometrical objects? This is particularly significant as in the VRLE the component movements of the turtle are not commutative. For example, movements of a roll followed by a tilt are different to a tilt followed by a roll. (PE7)

8. How will the children’s lack of kinaesthetic experience in tilting and rolling within a 3D environment affect their ability to operate with these movements within the VRLE? (PE8)
CHAPTER 3
DESIGN OF PROTOTYPE VRLE

3.1 OVERVIEW

Chapter 3 reports on the Stage 2 of the study: Design of prototype VRLE. The design of the prototype VRLE was informed by the conceptual framework constructed in Stage 1 (and presented in Chapter 2). The researcher designed a prototype VRLE named VRMath to be utilised in Stage 3 (the enactment and evaluation of the VRLE).

This chapter first discusses the technical considerations that impinged on the development of the prototype VRLE. During this discussion, issues such as the limitations and the availability of current technologies for creating VRLEs that placed constraints on the design of the VRMath will be reviewed. Next, the chapter presents the five phases that underlay the evolution and the development of VRMath:

- Phase 1. Identification of components.
- Phase 2. Design of the topological component (VR Interface).
- Phase 3. Design of the typological component (Programming Interface).
- Phase 4. Design of the social-actional component (Hypermedia and Discussion Forum).
- Phase 5. Fine-tuning the coordination between the components.

3.2 TECHNICAL CONSIDERATIONS

VRMath (VRLE) is a new paradigm of utilising ICT that employs VR technology. There were many possible ways in which a variety of available 3D technologies could have been used to construct VRMath. However, the technical considerations underlying the design and construction of VRMath were informed by Roblyer, Vye and Wilson’s (2001) criteria for the future of learning software. According to Roblyer et al., learning software should:

1. be Internet deliverable or Internet enhanced such as web-based learning environment;
2. emphasise visual and three-dimensional problem solving;

3. have the availability of visualisation or model building;

4. provide “rich” learning resources (e.g., hyperlinks to locate resources); and

5. have more apparent relative advantages that can be provided by an alternative or take the place of previous methods of teaching and learning.

In accordance with Roblyer et al.’s Criteria 1 and 4, the first decision made was that VRMath would be implemented on a World-Wide-Web platform. The implementation of VRMath on WWW platform has many advantages such as making VRMath an Internet deliverable, hypermedia rich resource that has the potential to mediate collaborative learning over the Internet (e.g., CSCL and discussion forums). To this, the Apache\(^8\) HTTP server was adopted to setup the web environment.

In accordance with Roblyer et al.’s Criteria 2 and 3, a second decision was made to utilise the ISO standard Virtual Reality Modelling Language (VRML, see Section 2.4.5.1) and Java\(^9\) during the construction of VRMath. VRML was specifically developed for computer 3D graphics on Web environments. There were two major reasons for the adoption of Java. First, the Java computer language can communicate with VRML and thus enables more sophisticated manipulation and construction of 3D models within VRML 3D virtual spaces. Second, Java has the power to implement a Logo-like language that can employ rich geometrical, logical, and procedural semantics for 3D model building and problem solving.

One of the important components of mathematical model-building within mathematical communities of practice is the knowledge-building discourse that occurs between members of the community (Scardamalia & Bereiter, 1993, 1996a, 1996b). Therefore, consistent with Roblyer et al.’s Criteria 3 and the notion of knowledge-building communities, it was decided that VRMath should have some mechanisms that enable users of VRMath to engage in online discussion and collaborative learning. To this end, an online discussion forum and data tracking

---

\(^8\) The Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server for modern operating systems.

\(^9\) Java is a cross platform computer language developed by Sun Microsystems, Inc.
system were built into the design of VRMath. The MySQL\textsuperscript{10} database system and PHP\textsuperscript{11} scripting language were utilised to implement these mechanisms.

The technologies (VRML, Java, Web environment (HTML, PHP), and database system (MSQL)) chosen for this project were capable of constructing a VRLE. However, they also set constraints in some aspects of the VRLE. For example, the use of VRML and Java on a web environment raised some issues such as the Java virtual machine (VM) must be the internal VM of the web browser. In this case, the web browser Internet Explorer (IE) seemed to be the only choice because most other web browsers do not have an internal VM. Moreover, the VRML browser was constrained to Cortona, which is the only VRML browser compliant to the current VRML standard\textsuperscript{12}. Therefore, the design of the VRLE must consider the user interfaces and functionalities of IE and Cortona.

The importance of these technical considerations is that technical issues have pre-shaped and guided the future design of the VRLE. It is important to point out that these technical considerations can influence the design of the VRLE and are out of control of the designer. The designer should always pay attention to the change of these technical situations, and reflect on them to improve the future design of the VRLE.

3.3 PHASE 1: IDENTIFICATION OF COMPONENTS

The VRLE is an information rich environment that facilitates meaning-making in 3D geometry. Informed by Lemke’s (1998; 1999; 2001) notion of multiple semiotic resources for meaning making, three main components of the VRLE were identified:

1. Topological component: This component refers to any meaning by degree or continuous representations of geometry such as visual graphics, images and 3D shapes and space. The main part of this component is the virtual reality interface that provides rich representation in colours, textures,
geometrical objects and behaviours, and allows for real-time navigation within the 3D virtual space.

2. Typological component: This component refers to any meaning by kind or discrete representations of geometry such as language, texts, numbers, icons and buttons. The main part of this component is the Logo-like programming language interface that engages students in logical procedural thinking and the tutee mode of computer usage (Taylor, 1980), and links to topological component while the students are programming to manipulate and build within the 3D virtual reality microworld.

3. Social-actional component: This component refers to many facilities such as online discourse that stimulate thinking, and the provision and sharing of information. The main part of this component is the web-based online discussion forum that aggregates information and scaffolds discourse. Students thus can contribute ideas, search for information and ask for help from more knowledgeable peers.

To incorporate these three components into the VRLE, a layout of the web interface was formed as shown in Figure 3.1.

![Figure 3.1 Prototype of VRMath](image)

**Figure 3.1** Prototype of VRMath

### 3.4 PHASE 2: DESIGN OF TOPOLOGICAL COMPONENT (VR INTERFACE)

The topological component mainly refers to the VR interface that provides meaning by degree and/or continuous visualisation of 3D geometry. The VR
interface is a 3D virtual space embedded in the 2D screen (see Figure 3.1). Through the rectangular 2D screen, users can use the 2D input devices such as mouse and keyboard to freely navigate within the almost infinite computational virtual space. Informed by the review of the research literature (see Section 2.3.3.2 of virtual reality and Section 2.4.5), issues such as mismatch of dimensionality (Leach et al., 1997) and disorientation (Elliott & Bruckman, 2002; Salzman, Dede, Loftin et al., 1999) in navigation were critical to the VR interface. Therefore, a tool bar containing multiple icons with associated functions to aid the navigation and discrimination of 3D VR space was developed. This section elaborates on the 3D navigation and the design of the VR tool bar within the VR space.

### 3.4.1 3D navigation

The navigation in 3D space is very important for getting proper viewpoints to observe and examine the 3D virtual worlds. It is necessary that users are familiar with them before they go into the construction activities. As was stated in Section 3.2 technical considerations, the VR space of VRMath utilises Cortona VRML browser as the VR space renderer. Therefore, the way to utilise the 2D input devices (e.g., mouse and keyboard) for 3D navigation is pre-defined in the design of Cortona system. To overcome the mismatch of dimensionality, Cortona system utilises the combination of mouse and keyboard to perform a full range of 3D navigation.

Users can navigate by using the mouse, keyboard, or both mouse and keyboard. To navigate with mouse, the pointer is positioned anywhere in the 3D window and the left mouse button is pressed. The mouse is then moved while its left button is held down. The direction in which users drag the mouse determines the navigation direction. The distance that user drags the mouse determines the speed of navigation. If users stop moving the mouse, the navigation will continue until the mouse button is released. To move around a 3D world using the keyboard, the mouse first is clicked into the 3D space to get focus for key events. The Up, Down, Left, and Right arrow keys are used to move around in 3D world. Users can combine Up and Down keys with Left and Right keys to perform two actions at the same time. For example, holding the Up and Left keys at the same time in Walk mode results in moving forward while turning left. To accelerate the navigation speed, the Shift, Ctrl or Shift+Ctrl keys are held down while navigation occurs.
Also, there are three main navigation modes in 3D space: Walk, Fly, and Examine as defined in VRML specification (see Section 2.4.5.6). These three navigation modes are usually performed by mouse dragging in conjunction with holding Alt and Space keys. For example:

1. Walk mode: In this mode, users can drag mouse or use arrow keys to move forward or backward, left turn or right turn. As the name walk implies, users cannot leave the plane (ground) because of the gravity effect on walking entities. However, users can follow the terrain (e.g., stairs) to walk up or down as a result of collision detect. Collision detect with objects is usually on in this mode. When navigating in Walk mode, holding the Alt key down while dragging left/right or pressing left/right keys results in sliding left or right, and holding Space key while dragging up/down or pressing up/down keys will mean to look up/down. It is important to notice that users always walk forward or backward on the plane they are standing on, not forward or backward in the direction they are looking.

2. Fly mode: In this mode, users are free to move and turn in any direction just like a flying airplane. There is no collision detect in this navigation mode. When navigating in Fly mode, dragging the mouse or pressing down on direction keys results in flying forward or backward, or turning left or eight according to the dragging direction or key(s) pressed. Holding Alt key while navigating with mouse or keys means to slide up, down, left, or right. Users cannot navigate forward or backward when Alt key is held. Holding Space key while dragging up/down or pressing up/down keys means to pitch (or roll) up/down the airplane's head. Holding Alt and Space keys together while dragging left/right or pressing left/right keys causes yawing (or tilting) left/right of the airplane's body and view. It is important to notice that users always fly forward or backward according to the direction they (the airplane) are facing.

3. Examine mode: This mode is also called Study mode, in which users can rotate the whole 3D world to examine or study it. There is no collision detect with objects in this navigation mode. When navigating in Examine mode, dragging the mouse or pressing down direction keys rotates the 3D
world according to your dragging direction or key(s) pressed. Holding Alt key while rotating with mouse or keys means to slide up, down, left, or right. No rotation will be made when Alt key is held. Holding Space key while dragging up/down or pressing up/down keys leads to zoom in/out the 3D world. Holding Alt and Space keys together while dragging left/right or pressing left/right keys means to yaw (or tilt) left/right of the 3D world. It is important to notice that it feels like the 3D world is rotating, but in fact it is the user him/herself changing his/her position and orientation facing to a rotation centre point.

The design of 3D navigation in Cortona system is sound. However, it is envisaged that the complexity of 3D navigation in VR interface will be difficult for young students to master. Therefore, some navigation aids (see Section 3.4.2) have been designed, and time and activities for users to get familiar with 3D navigation within VRMath also have been prepared (see Section 3.8).

3.4.2 VR tool bar

This is a GUI (Graphical User Interface) tool bar placed above the 3D VR space containing many icons with associated functions to aid the 3D navigation and the construction of 3D geometrical microworlds. The current design of this tool bar includes the following icons and functions:

1. Walk, Fly, and Examine are the three modes of navigating in VR space. These three modes of navigation were defined in VRML specification (Web3D Consortium, 2003a). Walk and fly modes reflect the navigation in real world. The examine mode, which is not possible to perform in real world when the target scene is large (e.g., a building), is advantageous in virtual reality environments because it enables users to examine 3D objects from many different viewpoints. These icons enable users to switch the navigation modes to whichever is the best one to get to the viewpoint they wish.

2. Avatar View switches between avatar’s view (turtle’s view) and free view. The avatar’s view is the viewpoint of the turtle. When in avatar’s view, free navigation in 3D space using mouse and keyboard is
temporarily disabled. The navigation is instead done by commanding the turtle to move the user’s viewpoints. Thus, users will be able to experience a mathematical movement from the viewpoint of the turtle in a manner similar to that employed by Edwards, Elliott, and Bruckman (2001) and Elliott and Bruckman (2002).

3. Rotation Centre sets the turtle’s current position as the rotation centre for Examine mode. This function is adopted with the special feature provided by Cortona VRML browser. Examine mode of navigation has a centre point to rotate the whole scene. It is envisaged that users can examine the virtual world easier with this function.

4. Horizontally Straighten sets the view to be parallel to the ground. In fly or examine mode of navigation, users can easily lose the horizon. This function can help restore the horizon.

5. Restore Viewpoint returns to the initial view. Because disorientation can easily occur in VR space, this function enables the users to restore the initial viewpoint. Thus, users can always restart the navigation whenever they feel lost in VR space.

6. Fit Screen fits all objects into view. This function provides another convenience for navigation. It calculates and brings the user to a viewpoint where all objects in the 3D space can be viewed. This can be utilised to get a quick holistic view of the scene and also helps to overcome disorientation.

7. Full Screen enlarges the VR space to full screen. The 3D VR space occupies only about a third of the 2D screen. This function can enlarge the 3D space to full screen. Thus it enables users to have a better examination and exploration of the virtual space.

8. Head light switches the light on and off. The light in VR space is an environmental cue that influences the colouring model and rendering of the geometrical objects within the virtual space. This function enables
users to experience the 3D stereo visualisation when the headlight is on and 2D picture effect when the headlight is off.

9. Collider switches the collision with objects facility on and off. This function is used in conjunction with walk mode of navigation. In walk mode, there is collision detection to geometrical objects. For example, user will be stopped when navigating to a wall (face). This function also makes it possible to walk through the wall.

10. Compass switches the compass on and off to shows the absolute direction (e.g., east, west, north, and south). The compass can provide a constant message to remind users about direction. This also helps users from getting disoriented.

11. Wired Frame and solid turns 3D objects into wire frames or solid shapes. The wire frame mode makes solid shapes transparent. Thus users can have both inside and outside perspectives of solid geometrical object.

12. Grid, Chessboard, and Sand Ground toggle the grid line, chessboard, and ground on and off. The grid line is 40 by 40 square meters with 1 meter intervals, which provides a sense of distance in the 3D space. Users can also get a reference point from grid, chessboard, and sand ground to improve their 3D sense in VR space.

13. Show/Hide Turtle toggles the presence of the turtle in 3D space. This function can temporarily turn the turtle invisible for better observation of the construction of geometrical objects. This function can also be done by giving command st (showturtle) and ht (hideturtle).

14. Show or hide the 3D coordinate axis. The 3D axis shows the 3D coordinate (red: x, green: y, and blue: z) used in VR space. This will help when the 3D coordinate system in geometry is introduced and when coordinate commands (e.g., setxyz) are used.

During the design of these icons, careful consideration was made about the meaning emission capacity of the icon images in terms of Peirce’s classification of signs: icon, index and symbol (Cunningham, 1992b). That is, the icon images can
quickly and efficiently communicate the actions (functions) embedded in the icons. Also, a help text describing the function was designed by the researcher to display when users placed the mouse pointer over the icons. Therefore, it was envisaged that users would be able to easily remember the functions associated with these icons. Moreover, the icons’ display was also considered with respect to icons’ actions. For example, the grid icon is in bright colours when grid function is activated, or it is in dimmed colours when grid is not activated. This is consistent in all implementation of icons (Karat et al., 1992).

3.5 PHASE 3: DESIGN OF TYPOLOGICAL COMPONENT (PROGRAMMING INTERFACE)

The typology component mainly refers to the programming interface that provides meaning by type and/or discrete and symbolic representation (e.g., language) of 3D geometry. The programming interface includes an extended Logo language and many GUI widgets (e.g., buttons, menus, and dialogs) to help easily create 3D objects and write procedures.

3.5.1 Extension of Logo language in VRMath (commands)

Logo language has had a long history, an extensive research literature and resources (see review in Section 2.3.3.2 turtle geometry and Section 2.4.4.3). VRMath is an integrated VRLE (Virtual Reality Learning Environment) or microworld, in which the Logo language is an important component. However, the Logo language in VRMath is not just another dialect or extension of Logo. It is a very distinctive Logo -- a 3D Logo. Because of the 3D real time graphics (VR) in VRMath, the Logo language here has significant conceptual differences. For example:

1. Traditional 2D Logo has a 2D graphical window, and the turtle is bound in the very limited window. When turtle moves out of the boundary, the WRAP command or scroll bars must be used to follow the turtle. The 2D graphical window is limitedly small. In contrast, in VRMath the VR graphical interface is an almost infinite 3D space that allows users to walk, fly, or examine in the 3D virtual space.
2. Traditional 2D Logo uses pixels as the unit of distance. While in VRMath, the unit of distance is metre or centimetre as we use in the real world.

3. Traditional 2D Logo has two turns only (left and right). In VRMath, the 3D space requires six turns (left and right, rollup and rolldown, tilt left and tilt right). The 2D Logo has limitations for the research on human's 3D spatial ability, while VRMath has the potential to investigate on human's full range of 3D spatial ability.

4. Traditional 2D Logo draws pixels on screen, while VRMath creates dynamic objects in 3D space, which can perform behaviours (animation) to reflect the real world.

As a result, the Logo language in VRMath has a set of 3D related commands, and has eliminated some unnecessary 2D commands. Generally speaking, to use VRMath, one has to go beyond one’s 2D-based preconceptions and imagine oneself is in a 3D world just like in the world we are living in. This set of 3D related commands has been designed and classified into five categories: Move commands, Turn commands, Create commands, Material commands, and Setting commands.

### 3.5.1.1 Move commands

The move commands can change the turtle's position in 3D space. Move commands do not change the turtle's orientation. Informed by the notion of multiple frames of reference (Darken, 1996; McCormick et al., 1998; Olson & Bialystok, 1983; Piaget & Inhelder, 1956; Sachter, 1991; Salzman, Dede, & Loftin, 1999; Yakimanskaya et al., 1991), VRMath employs three sets of move commands based on three frames of reference:

1. Egocentric: This set of commands is based on turtle's viewpoint. They are FD or FORWARD, BK or BACK or BACKWARD.

2. Fixed: This set of commands is based on something or directions that are fixed and not to be changed. They are UP: the turtle goes up towards sky, DOWN or DN: the turtle goes down towards ground, EAST: the turtle goes towards east, WEST: the turtle goes towards west, NORTH: the turtle goes towards north, and SOUTH: the turtle goes towards south.
3. Coordinate: This set of commands is based on the Cartesian coordinate system adopted in VRMath (see Figure 2.20). These commands are: SETPOS, SETXYZ, SETX, SETY, and SETZ etc.

There is also a set of jump commands associated with every move command. The jump series commands add the prefix "JUMP" before every move command (e.g., JUMPFORWARD or JF, JUMPBACK or JB, JUMPEAST or JE, JUMPUP or JU, JUMPXYZ or JXYZ, JUMPX or JX etc.). This jump series commands only have effect when you are creating turtle tracks objects (PENDOWN or PD). They mean temporarily no tracks, or end of a face.

3.5.1.2 Turn commands

The turn commands will not change the turtle's position, but change its orientation (or rotate the turtle). To fully operationalise the 3D rotation, another four turn commands were designed in addition to the two traditional 2D Logo turn commands: LEFT and RIGHT. As was reviewed in Section 2.3.3.2, both Elica Logo and MSWLogo have adopted UPPITCH, DOWNPITCH, RIGHTROLL, and LEFTROLL to perform 3D rotation. These were considered by the researcher when choosing the appropriate commands for VRMath. However, considering the short command for UPPITCH will be conflicting with UP command, the researcher therefore adopted ROLLUP or RU, ROLLDOWN or RD, TILTLEFT or TL, and TILTRIGHT or TR. Figure 3.2 illustrates the turn commands used in VRMath’s 3D space.

Figure 3.2 3D rotation in VRMath
3.5.1.3 Create commands

This set of commands creates geometrical objects in 3D space according to the turtle's position and orientation. These commands include:

1. PENDOWN or PD: Start recording turtle's track for point set, line set, or face set.
2. PENUP or PU: End of recording turtle's track.

One PD and PU creates one object. Using FACE, LINE, or POINT command before PD determines the object type. There is a PENERASE or PE command for undoing one track when creating a PD object. Figure 3.3 shows a frame of cube and a face created by using PD commands in LINE and FACE mode.

![Figure 3.3 A frame of cube and a face created in VRMath](image)

The other Create commands in VRMath are:

1. BOX or CUBE: Create a box object.
2. CAN or CYLINDER: Create a cylinder object.
3. CONE: Create a cone object.
4. BALL or SPHERE: Create a sphere object.
5. LABEL: Create a label object. Only this command in this set needs a parameter (e.g., LABEL “VRMath”).

Point set and line set objects created by PD will be affected by pen color. Users can use Pen Color Editor (see Section 3.3.3.2) in Tool menu or pen color related commands to change pen color. Face set object created by PD may be affected by pen color when PENCOLORON or PCON command is specified or by material settings when PENCOLOROFF or PCOFF command is specified. Users can change material settings by using Material Editor (see Section 3.3.3.2) in Tool
menu or material related commands. All other objects will be affected by material settings. All objects will be affected by SCALE series commands, which can be found in Setting commands. Figure 3.4 shows these objects created in VRMath.

![VRMath](image)

*Figure 3.4 Primitive objects of VRMath*

### 3.5.1.4 Material commands

There are six elements in 3D objects material as described in Material Editor (see Section 3.3.3.2). The material setting of objects will be rendered with the lighting model and thus create a realistic scene. Each of the six elements has one associated command:

1. **SETDIFFUSECOLOR** or **SETDC**: set diffuse colour of object.
2. **SETEMISSIVECOLOR** or **SETEC**: set emissive colour of object.
3. **SETSPECULARCOLOR** or **SETSC**: set specular colour of object.

Each of the above three commands has three parameters: red, green, and blue ranging from 0 to 1000 in integer. For example, **SETDC 0 0 1000** means set diffuse colour to blue. Another 3 commands are:

1. **SETAMBIENTINTENSITY** or **SETAI**: set ambient intensity for object.
2. **SETSHININESS** or **SETSH**: set shininess for object.
3. **SETTRANSPARENCY** or **SETTR**: set transparency for object.

Each of the above three commands has one integer parameter ranging from 0 to 1000. For example, **SETTR 500** makes the object created semi-transparent.
3.5.1.5 Setting commands

This set of commands influences the numbers used by move commands or the object's material or type.

1. CENTIMETER or CM: set the distance unit to centimetre long.
2. METER: set the distance unit to meter, which is the default unit in VRMath.
3. FACE: tells the turtle to create face when PD.
4. LINE: tells the turtle to create line, which is the default turtle track.
5. POINT: tells the turtle to create point when PD.
6. PENCOLORON or PCON: use pen colour for FACE object.
7. PENCOLOROFF or PCOFF: use material settings for FACE object.

The PCON and PCOFF commands are only useful when FACE and PD are used together. These commands (except METER and CENTIMETER) should be specified before PD command. Once the pen is down (PD), the pen types (i.e., POINT, LINE, and FACE) and pen colour (i.e., PCON and PCOFF) for turtle track objects cannot be changed. This is due to the convenience of implementation. But it is considered that users may wish to change these settings during PD. The users’ demands with respect to this will be investigated in the enactment stage of this project.

Another important series of setting commands is the scaling command, which will affect all objects when creating in 3D space. The scale of geometrical objects is of great importance with respect to the geometrical concepts about topology and congruence.

1. SCALE x y z: sets the scale for the geometrical objects being created within VRMath by defining the relationship between the three axes. The x, y, and z are floating point numbers and they ignore the number sign (positive or negative). The default ratio of 1 1 1 is most commonly used. Give either of x, y, or z the value of 0 means objects will lose one dimension. For example, SCALE 1 0 1 BALL will create a face of a circle (a disc) (Figure 3.5).
The following commands can be used to change one dimension only. The geometrical vocabularies: width, height, and depth were utilised in addition to x, y, and z for the familiarisation of young students.

2. SCALEX or SCALEWIDTH or SCALEW: change the scale of the x dimension.

3. SCALEY or SCALEHEIGHT or SCALEH: change the scale of the y dimension.

4. SCALEZ or SCALEDEPTH or SCALLED: change the scale of the z dimension.

Apart from this set of 3D related commands developed by this researcher, most other traditional Logo commands and syntax were kept including the structure of procedure, the naming of variables, and variable types. The implementation of VRMath Logo language was derived and modified from an open source project named TurtleTracks developed by Azuma (1999) in Java language. Further extensions such as animation of geometrical objects’ properties of the Logo language in VRMath are under development. This research study focuses on the use of current implementation of the Logo extension commands by young students to conceptualise the next stage of VRMath’s research and development.

3.5.2 Graphic User Interface (GUI) of programming interface

As was reviewed in Section 2.4.4.3, Logo has evolved to include many interface features such as GUIs, multimedia, and Internet accessibility etc. to facilitate the interaction between human and computer. VRMath employed these features to improve the interaction between user and programming interface, facilitate the learning of programming commands and procedures, and serve as the symbolic representation of a mathematical function as well as the glue that bound all
the representational modes together (Hoyles et al., 2002). Figure 3.6 below shows the layout of the programming interface GUI.

![Figure 3.6 The layout of programming interface GUI](image)

This GUI utilised computer graphical widgets such as menu, button, and text display and input that encompassed links to all functions of programming within this small rectangular area on screen. More graphical widgets (e.g., icons and scrollbars) were utilised in other windows or dialogs (e.g., Material Editor and Quick Command from tool menu). The utilisation of these graphical widgets were carefully considered according to the semiotic principles for underlying the concept of interface design (Oliveira & Baranauskas, 2000, see Section 2.4.6.1). There were four menus: Project, Edit, Tool, and Environment to integrate and classify functions. The text display area in the centre of the interface (see Figure 3.6) is the Interactive Message Centre that provides information and responds to users’ command, which also can be used for debugging purposes. The Command Input Box at the bottom of the interface can interpret users’ commands and reflect on the VR interface straight away. The Help button, when clicked by mouse, links to an information web page about VRMath commands displayed in the hypermedia interface (see Section 3.6.1).

### 3.5.2.1 Project menu

Project menu (Figure 3.7) enables easy project management. Work in progress projects can be saved into a network database. Later, users can easily retrieve (open) their projects to continue. Projects can be made public by their authors in order to share projects with other users. In this prototype VRMath, projects can only save the information including project title, date, description, public status, and procedures just like other traditional Logo. The 3D microworlds
created in projects cannot be saved. However, they can be exported to VRML files or recreated by executing the saved procedures.

![Project menu](image)

**Figure 3.7 Project menu**

### 3.5.2.2 Edit menu

Edit menu (Figure 3.8) consists of three groups of functions. The first group is related to procedure management. New procedure function invokes a procedure editor (Figure 3.9) that enables procedures to be created, edited, and executed. Opening the Editor function (Figure 3.10) presents a dialog listing the current defined procedures, and allows those procedures to be modified (edited) or deleted. The second group is linked to the maintenance of the geometrical objects created in VR virtual space. Undo function can remove the last geometrical object added into VR space. In contrast, Redo function puts back the last geometrical object removed by Undo action. The design of Undo and Redo functions was new and unprecedented in traditional pixel-drawing Logo environments. They were probably only applicable in VRMath because the geometrical entities were treated as objects not pixels.

![Edit menu](image)

**Figure 3.8 Edit menu**
3.5.2.3 Tool menu

The Tool menu (Figure 3.11) provides six important tools to aid the construction and construction process in 3D microworlds. They appear as separate dialog windows when selected by mouse. Each of them is described in turn in the following paragraphs.

**Background Chooser**

The Background Chooser (Figure 3.12) allows users to change the background of the 3D virtual space. As the name Chooser implied, it does not enable
users to customise the background. The background in the 3D VR space is defined as an infinite far away phenomenon which can never be reached. There are two sets of background provided in Background Chooser: Color and UniversalMedia\textsuperscript{13}. The Color set of background is rendered in colours. Multiple colours can be used for sky and ground colours. Thus, it creates the gradient effect of colouring to simulate day time, night time, or under water background. In contrast, the UniversalMedia background consists of panoramic pictures to mimic the real world scenes (see Figure 3.12). Background is an important environmental cue which gives an authentic sense to users. It is envisaged that the panoramic background can provide a reference point to keep users’ attention on directionality. Different backgrounds may also help the construction processes depending on the context of the microworld being constructed.

\textbf{Figure 3.12 Background Chooser}

\textbf{Pen Color Editor}

Pen Color Editor (Figure 3.13) was designed to visually utilise the full colouring potential of computer graphics for point, line, and face objects. Therefore, users can visually select a colour from the provided palette, the colour canvas, or the scrollbars of red, green, and blue. Traditionally, there were colouring commands such as SETPC, COLOR, SETRGB, and PALETTE etc. in Logo. VRMath has a set

\textsuperscript{13} UniversalMedia is a Web3D Consortium Solution, which increases the realism of online Web3D worlds (VRML, Java 3D, and other online 3D technologies) and decreases network downloads by defining a small, cross-platform library of locally resident media elements (textures, sounds and 3D objects) and a uniform resource name (URN) mechanism by which Web3D content creators can incorporate these media elements into their worlds. See http://web3dmedia.com/UniversalMedia/
of colouring commands inherited from traditional Logo language. However, the values of red, green, and blue are ranged from 0 to 1000 in VRMath instead of from 0 to 255 in versions of Logo. This may overcome young users’ curiosity about the number 255. The Copy button will generate a SETRGB command for the current selected colour, and store the command in clipboard. Users can then paste into the editing procedures.

![Pen Color Editor](image1.jpg) ![Font Chooser](image2.jpg)

**Figure 3.13 Pen Color Editor**

**Figure 3.14 Font Chooser**

**Font Chooser**

Font Chooser (Figure 3.14) was designed to aid the construction of LABEL object. By default, the label will be placed on the right hand side of the turtle, facing opposite to the turtle’s direction, and above the turtle’s plane. These settings can be easily changed in this Font Chooser dialog. There are also six commands to associate with these six font settings. They are SETFF (set font family), SETFJ (set font justification), SETFS (set font style), SETFH (set font horizontal/vertical), SETFL (set font left-to-right/right-to-left), and SETFT (set font top-to-bottom/bottom-to-top). This Font Chooser may help users to easily use and associate with commands. Similar to Pen Color Editor, the Copy button generates six font setting commands for users to easily use in procedures. That is, when the Copy button is clicked, the font commands will be generated in the system’s clipboard. The users can then paste the font commands into Procedure Editor when writing procedures.

**Material Editor**

Material is a new concept that applies to VR technology. Material Editor (Figure 3.15) provides immediate visualisation of colouring and lighting effects of
objects. Thus, objects are perceived in stereo like real world objects. There are 16 categories of material predefined in Material Editor allowing users to make a quick selection. Users can also customise their own material by manipulating the scrollbars and buttons. Associated commands (see Section 3.5.1.4) can also be manipulated in procedures to create dynamic materials. Material Editor is designed to help users to understand and use the material commands. The material also provides more authentic meaning to motivate construction within the virtual world. The Copy button generates six material commands for easy use in procedures. That is, when the Copy button is clicked, the material commands will be generated in the system’s clipboard. The users can then paste the material commands into Procedure Editor when writing procedures.

Figure 3.15 Material Editor

Quick Command

Quick Command (Figure 3.16) was designed to help the learning of the moving and turning commands. There are 33 commands that can be generated by this Quick Command dialog. When users point the mouse cursor to the moving and turning icons, the corresponding commands are shown in the command text field at the bottom of this dialog. If users click the icon or button, the corresponding command will be sent and execute, and immediately reflected in the 3D VR space. This is to help the semiosis of the geometrical meaning of moving (change location) and turning (change direction) by linking to multiple semiotic resources such as icon,
button, symbol, language, and action etc. Quick Command can be a good start for novice users for learning the moving and turning commands.

In this prototype, the Options window (Figure 3.17) only consists of the manipulation of turtle waits, which is the delay between every moving and turning command expressed in milli-seconds. When the turtle is moving or turning as it executes procedures, it is very fast as the default for the turtle waits is 20 ms. Therefore, it is necessary that users are able to increase the delay between movements in order to better observe and understand the commands and the movements. The dragging of the scrollbar in Options dialog will have effect even when the turtle is in action. The SETWAIT command will only take effect after the turtle stops movement.

Environment menu

Environment menu (Figure 3.18) provides two important functions. It is possible that users accidentally or intentionally create an endless loop in a procedure or without giving a proper exiting condition to a recursive procedure resulting in the execution of procedure never stopping. In this circumstance, users can use the Interrupt function to arbitrary stop the execution of procedure. Reset Environment
will also interrupt the execution of the executing procedure, and also execute the
RESET command, which restores VRMath to its initial state.

<table>
<thead>
<tr>
<th>Project</th>
<th>Edit</th>
<th>Tool</th>
<th>Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Reset Environment</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Interrupt</td>
</tr>
</tbody>
</table>

*Figure 3.18 Environment menu*

3.6 PHASE 4: DESIGN OF SOCIAL-ACTIONAL COMPONENT
(HYPERMEDIA AND FORUM INTERFACE)

This component refers to facilities such as online discourse that stimulate
thinking, and the provision and sharing of information. It exists not only for human-
computer interaction, but also for distant and asynchronised human to human
interaction, and to enable the collaborative learning. This is a web-based online
discussion forum that aggregates information and scaffolds discourse. Students can
contribute ideas, search for information and ask for help from their peers. Also, there
is much documentation that provides necessary information to aid the use of the
VRMath system. All resources in this component are organised in the form of
hypermedia, which reflects the non-linear nature of human thinking and hyperlinked
to world wide Internet resources.

3.6.1 Hypermedia documentation

Considering the concerns about the potential limits of hypermedia (Dede &
Palumbo, 1991), this component implemented some basic browsing functions and
shortcuts linking important documents to prevent disoriented browsing and cognitive
overload. There are icons on the top of this component to associate with these
functions and shortcuts.

1. **Enlarge/reduce component frame:** Because of the limited space of the
   2D screen, these icons were designed to enable changing the size of
   component frames at the users’ convenience. Users could also adjust the
   size of frames by dragging the division line between components.

2. **Previous page:** The original browsing interface of the web browser
   was disabled in VRMath to help reduce the interference of non-relevant
   web functions. However, basic browsing functions are still important.
Therefore, this icon was designed to help users traversing in the browsing history.

3.  Next page: This icon was also designed to help users move to the next browsed page.

4.  Home page: This is a shortcut to go back to the initial page of this component. Thus, users could always have a chance to restart again.

5.  Forum page: This is an important shortcut to the forum page.

6.  Help Index page: This icon displays an index page containing links to important document pages such as introduction page, interface tour page, command library page, 3D navigation page, and tutorial page etc. This index page could help easily locate information.

7.  Exit VRMath: This icon provides a clearly marked exit (Karat et al., 1992) to improve the usability of VRMath.

In addition to the functional interface discussed above, there are also many important documentation pages for learning and instructional purposes. These pages are written in the form of hypermedia. Therefore, there are hyperlinks over these pages to organise the thoughts and reference to other pages whenever it is appropriate. These pages include:

1. Introduction: This page provides an introduction about the VRMath project including interface and general guides to the use of VRMath.

2. Interface tour: This page provides an interface tour for the three components and the functions associated with buttons, menus etc.

3. Command library: This page is a functional oriented guide of VRMath commands. Because commands are classified, it will do much to assist beginners learn the commands.

4. Command index: This page contains all commands in alphabetical order. It will assist advanced users to quickly locate a particular command.

5. 3D Navigation: This page contains information about 3D navigation using mouse and keyboard.
6. Tutorial page: This page contains information for using commands, writing procedures, and saving projects in VRMath.

7. Tips: This page provides some helpful tips to effectively use VRMath.

These hypermedia pages can be found in Appendix 2.

3.6.2 Online discussion forum

The online discussion forum plays an important role in the social construction of knowledge. It forms an online community where members can express ideas, seek help, and work together. The forum is named VRMath Forum. It contains many sub-forums to scaffold discussions (Figure 3.19). Current sub-forums include:

1. FAQ: This was designed for frequently asked/answered questions in VRMath.
2. 3D Virtual Space: This was designed for discussions about the 3D space including navigation and control of the 3D interface.
3. Geometry: This was designed for discussions about geometry in virtual and real worlds.
4. Programming: This was designed for discussions about the programming language in VRMath.
5. Show Room: This is a place where members can share their creations of the 3D virtual worlds.
6. Learning Tasks: This sub-forum is for discussions about developing learning tasks in VRMath system.

The design of the forum was adopted from an open source project named phpBB\(^\text{14}\). phpBB has many features that are suitable for building an educational community:

\[^{14}\text{phpBB is a high powered, fully scalable, and highly customisable open-source bulletin board package. phpBB has a user-friendly interface, simple and straightforward administration panel, and helpful FAQ. It is the ideal free community solution for web sites. More information about phpBB can be found at http://www.phpbb.com/}\]
1. User management: Members of the forum are the core of forum community. phpBB has sound user management including online registration, user privilege setup, user group control, and user profile etc. Users can also select an avatar, which is a picture of a character (e.g., human, cartoon, or animal etc.) to represent themselves in addition to their username. This motivates the users’ participation in the forum community.

2. Forum management: Multiple forums are allowed and easy to setup in phpBB. Forums can be set as public or private, hide or disabled, open to every member or moderated by some users, and a variety of privilege settings according to different privileged users.

3. Post management: It is easy to post messages in the forum. The design of the posting interface is very intuitive. The posting messages can be rich text format, in which different font, font size, colour, and style, images, and hyperlinks are allowed. Many built in smiley icons termed “Emoticon” are available for using in messages. This can help the affective expression within forum discussions.

4. Multiple language interfaces: As an international project, phpBB has included multiple languages contributed from world wide users. This may
be helpful for VRMath to attract a wide range of users from different language backgrounds.

5. Powerful search utility: phpBB provides a sophisticated searching facility for locating certain posts (information). Searching can accurately find relevant posts by specifying searching criteria such as keyword, author, forum, and category combined with boolean expressions (e.g., AND, OR, and NOT). This will be useful for retrieving information in forums when the amount of posts is big.

6. Private messaging system: For every registered user, a private messaging system for personal private communication is available (as opposed to public communication in forums). The composing of private messages is the same process as composing posts in forums.

Despite these features, however, phpBB is designed for general online community on web environment. In the future, cognitive scaffolds (e.g., theory building, opinion, model building, model critiquing, plan, and plan critiquing etc.) such as those currently used in Knowledge Forum ® (Scardamalia & Bereiter, 2002) may be incorporated into VRMath Forum. This would be expected to enhance deep knowledge construction and knowledge-building activity within VRMath Forum.

3.7 PHASE 5: FINE-TUNING THE COORDINATION BETWEEN COMPONENTS

The fine-tuning of the coordination between components mainly concerned the unification and consistency of VRMath system, and the interaction among the three components for the overall usability of VRMath. VRMath utilises a variety of open resources, which were developed under different purposes. Therefore, it was necessary and important to coordinate and integrate all resources toward achieving the aims of VRMath (i.e., to be a microworld for knowledge construction of 3D geometry).

User database consolidation

VRMath Forum has a built-in user database for the operation of online community. Similarly, the project management in programming interface also needs a user database for creating and saving projects. These two user databases have been
consolidated into one database. Therefore, users will only need to login once to participate in the forum community and manage their projects.

*Interface consistency*

This is one of the usability considerations for educational software made by Karat, Campbell, and Fiegel (1992). Other considerations include the use of simple and natural dialogs, provision of intuitive visual layout, speaking of the user’s language, provision of good help, and the support of input device continuity. For example, in the design of icons, the following points are consistent:

1. The size and colouring scheme (e.g., bright colour means in action, dimmed colour means not in action) of the icons.
2. The display of user-friendly help-texts and roll over effect for mouse over icon event.
3. The action taking of mouse click event on icons.

*Interaction among components*

The three semiotic components: topological, typological, and social-actional have been identified to be important for mathematical meaning making. And technically, the technologies (i.e., VR, Java, and HTML) used in the three components are able to communicate each other bi-directionally (i.e., send and receive messages). Hence, there is no doubt and no problem on why and how to interact among components. However, when and what to communicate are the two key questions for the design and evaluation (i.e., PE2) of this VRLE. In this prototype VRMath, the following interactions were designed:

1. Immediate visualisation of geometrical language: This is the interaction between typological and topological components. When geometrical commands are sent into programming interface, the VR interface provides the visualisation of the geometrical objects. For example, repeat 4 [fd 1 rt 90] from programming interface results in the turtle walking and creating a square in VR interface. This interaction strengthens the users’ formalisation of geometrical language to its associated geometrical objects through the movement and direction of the turtle to the understanding of geometrical shape, patterns and properties. In addition, A SETWAIT
command was designed in this interaction to better facilitate the users’ understanding of geometrical processes.

2. Command facilitator: This refers to the design features that facilitate the use of commands in the programming interface. For example, the hypermedia documents and the users’ comments and discussions in the forum provide information on commands. The Material Editor and Pen Color Editor provide topological variance in material and colour to facilitate users’ understanding of material and colour commands, and use in procedures (e.g., Copy button).

3. Multiple shortcuts: There are many ways to perform the same thing in VRMath. For example, to hide the turtle in 3D VR space, users can either click on the turtle icon in VR interface or use HIDETURTLE command in programming interface. To display the help page in the hypermedia component, users can either follow the hyperlink provided in forum or use the HELP command or click on Help button in programming interface.

4. Data tracking: This is used for evaluating the users’ learning within VRMath, users’ work history including time and commands sent can be recorded. This relies on the interaction between forum interface and programming interface.

3.8 SUMMARY

This chapter presented the design of the prototype VRLE named VRMath. The design of the prototype VRLE was informed by the review of research literature and the conceptual framework developed in Chapter 2. As a result, VRMath consisted of three essential components: topological (VR interface), typological (programming interface), and social-actional (hypermedia and forum interface) semiotic resources for mathematical meaning making. The design of the three components (interfaces) was elaborated in details, and the physical implementation of VRMath was completed. With the completion of the prototype VRMath, this research study was now ready to proceed to the next stage of the design experiments (i.e., enactment and evaluation).
CHAPTER 4

ENACTMENT AND EVALUATION: ITERATION 1

4.1 OVERVIEW

Chapter 4 reports on the first iteration of Stage 3 of the design experiments: the enactment and evaluation of the VRLE. The focus of Iteration 1 was on the design of learning activities. Because of the novelty of the conceptual framework and the prototype VRLE, it was necessary to design new types of 3D geometrical learning activities that would utilise the unique learning environment provided by the VRLE.

Therefore, the primary aim of this iteration was to design, evaluate and refine a set of five learning activities for the VRLE. Concurrently with this, both the design of VRLE and students’ learning within the VRLE were also evaluated. The evaluation of the design of VRMath involved a usability inspection, while the evaluation of the users’ learning focused on knowledge construction processes and conceptual changes with respect to 3D geometry as a result of the experiences with VRMath.

The following sections present the design and method, results and discussion, implications, and refined learning activities for Iteration 2.

4.2 DESIGN AND METHOD

4.2.1 Participants

The two participants in this iteration were Emilie (a Year 6 student) and Anya (a Year 7 student) who both attended an inner city school in eastern Australia. Neither participant had prior experiences in programming and Logo. Nor had either participant had any experience with 3D computer graphics.

4.2.2 Data collection and analysis

As the students participated in each of the five activities, data were collected from three sources:
1. Observations and video and audio recordings of the students’ interaction with VRMath interface and intellectual engagement with the five activities.

2. Focus interviews with the two students after each session on the usability issues of VRMath.

3. Computer records of the participants’ inputs in VRMath including forum messages, procedures in their projects, and the geometrical objects created.

The analysis of data utilised the grounded theory approach (Strauss & Corbin, 1990, 1998). Data analysis in this iteration thus followed the steps below.

1. Transcription: The data derived from tape-recorded interviews, and video recordings were transcribed in full. Because non-verbal cues such as communication of embarrassment or emotional distress, or simply a pause for thought are important elements (Hancock, 1998; Lacey & Luff, 2001), in addition to words spoken, non-verbal cues also were recorded at appropriate places in the transcriptions.

2. Organisation of data: This step reduced the large amount of data from the transcription. Data were organised into easily retrievable sections with pseudonyms and/or code numbers.

3. Familiarisation: Before the formal analysis began, the researcher reviewed the data including audio/video tapes and written notes, and made memos and summaries from the original data.

4. Coding: During this step, the data were categorised and coded.

5. Identifying Themes or Commonalities: In this step, the data coded in the previous step were further analysed to identify new themes and the major and minor categories under each themes. Analysing techniques such as charting, comparing and mapping concepts were utilised in this step. Themes with respect to differentiating between spatial orientation and spatial visualisation, frames of references (e.g., egocentric vs. fixed), and 3D navigation were identified.
6. Conclusion drawing/verification: This step interpreted the phenomena and relationships in/between themes and their sub-categories.

4.2.3 Procedure

Both participants interacted with VRMath together for six sessions (one hour per session, one session per week). In the initial session, the participants were given a brief introduction to VRMath. During the next five sessions, they engaged in the following five pre-designed learning activities:

1. Become a member: In this activity, participants registered themselves into VRMath Forum. This enabled them to participate in online discussions and to create projects within the learning environment.

2. Turtle’s eyes: In this activity, the participants were presented with a game in which they had to discover the secret about the turtle’s eyes. That is, the turtle’s eyes only change colour when they navigate close to the turtle using the mouse. The purpose of this activity was to develop the participants’ skills in 3D navigation (walk, fly, examine) within VRMath using the mouse and keyboard.

3. Turtle dance: This activity focused on developing an understanding of basic commands for changing the turtle’s position and orientation within VRMath. These commands included six turning commands (left, right, rollup, rolldown, tiltleft, and tiltright) and position changing commands (Egocentric: forward, back Fixed: east, west, north, south, up, down). Repeat command was introduced in this activity for animating the turtle.

4. Formula of polygon: The purpose of this activity was to have the participants develop a pattern or command for constructing a polygon by using the repeat command. The desired formula or command for a polygon is repeat side [forward 1 right 360/side].

5. Creating a cube: In this activity, participants were required to write a procedure for constructing a frame of cube, and then save it as a public project where it could be examined by other members in the learning community.
At the beginning of each session, the researcher explained the purpose of the activity (e.g., “We are going to find a secret about the turtle’s eyes,” or “We are going to ask the turtle to dance”) to both students. Then the researcher introduced knowledge about the VRLE necessary for the successful completion of tasks to the students (e.g., the three modes of navigation, the semantics and syntax of commands). This was done by reading through with the students the information pages for the particular activity. These information pages were located in VRMath Forum. Each student then went to her own computer and worked on the assigned activity. When any difficulty was encountered, the students were able to seek help from the researcher or from the other student. Rather than merely showing the student how the difficulty could be overcome, the researcher actively tried to scaffold student resolution of the difficulty by replying to their questions with questions of his own in a manner suggested by Papert (1993). At the end of each session, the researcher and the two students met in a focus group to discuss what they had learnt. Each focus group session concluded with the administration of a set of questions (see Appendix 3) that focused on the usability of VRMath and other computer-human interface issues with respect to VRMath.

4.3 RESULTS AND DISCUSSION

4.3.1 Learning Activity 1: Become a member

To become a member of the VRMath Forum community, participants have to register themselves by filling in an online form. The two participants found the process of filling in the online form was easy. They were excited when choosing a nickname and in particular were very intrigued by being able to select a personal Avatar (i.e., a picture image).

After becoming a member, the participants were asked to post messages in the forum. They learnt how to post and send messages very quickly using the information given on the webpage with very few verbal instructions from the researcher. When writing a message, they also utilised the small ‘smilies’, which are termed “emoticons” to express emotions. They did not post public messages; instead they preferred to communicate by sending private messages to one another and to the researcher. The messages sent to the researcher did not focus on the geometry but instead were mainly social in nature. Because VRMath was just in its rudimentary
stage, there were only few casual users in the forum community. This resulted in no
effective social construction of knowledge in the forum. This may explain the
participants’ preference in communicating through private messages.

4.3.2 Learning Activity 2: Turtle’s eyes

During this activity, the two participants seemed to find it difficult to navigate
in 3D space using a mouse. Two main difficulties were apparent:

1. Controlling the speed of navigation: In VRMath, the distance the mouse
   is dragged determines the speed with which the user navigates around the
   3D environment; the further the mouse is dragged, the faster the user
   moves within the environment. This was explained to participants at the
   beginning of this activity. However, during this session both students had
   great difficulty in controlling speed as they attempted to navigate within
   the 3D virtual environment.

2. Controlling the direction of navigation: To navigate in different directions
   within the 3D virtual environment requires more than the four directions
   provided by arrow keys or by the mouse being dragged over the two-
   dimensional mouse pad. Therefore, to navigate within the 3D virtual
   world required other keys (e.g., Alt, Space) to be used in conjunction with
   the arrow keys or the mouse dragging over the mouse pad. This was
   further complicated by the three modes of navigation within the 3D
   virtual world: walk, fly, examine. The students found the coordination of
   the Alt and Space keys with the dragging of the mouse rather difficult.
   However, by the end of this session after many trial-and-error
   explorations, they had mastered the control of navigation.

The completion of the activity (to navigate within 0.5 metre of the turtle’s
eyes and see the eyes turn red) was not achieved simply by navigating using the three
modes of navigation. The two participants found that they had to experiment with
other navigation aids before they were able to satisfactorily complete the activity.
These other navigation aids such as Change Navigation Mode, Avatar View, Set
Rotation Centre, Align to Ground, Restore Viewpoint, and Fit Screen etc. were
displayed as icons above the 3D window. Because it was impossible for these icons
to fully deliver their meanings of functionality, a mechanism of showing a Help Text
when the mouse was pointing on the icon was built into the design of the system. Both participants found the Help Text very useful, especially when they began the activity. During their navigation in 3D space, the “Restore Viewpoint” icon was found to be used the most often, as the students got lost very easily in 3D virtual space. After investigating all of the other navigation aids, the participants found the best way to complete this task was to use Fit Screen function, which brought all objects into the 3D screen. Emilie contributed her experience and wrote in the forum:

You make the turtles eyes go red by hitting the fit screen button and then rotating the turtle by using the examine mode.

In VRMath’s 3D space, the users navigate themselves about the 3D virtual environment with the mouse and the navigation keys. Using the mouse and the navigation keys has no effect on the turtle. The turtle can only be moved by specific written commands (e.g., FD 1, BK 10 etc.). As the participants were navigating towards the turtle using the mouse and navigation keys, they were asked, “Is the turtle moving?” Both participants replied very confidently that the turtle was moving. This indicated that as the participants were navigating within VRMath, their differentiation between spatial visualisation and orientation (McGee, 1979) was lost. Normally in the real world, one can differentiate between spatial visualisation and orientation by referring to points of reference located in space or to one’s kinaesthetic sense. Thus, if the person perceives that her location with respect to these points of reference are staying constant whilst her perspective of an object is changing and her kinaesthetic sense indicates that she is not moving, then the person knows that the object is moving and her spatial visualisation of the object is changing. By contrast, if the person perceives that both her location to the points of reference and her perspective of an object is changing, then the person knows that she is moving and thus her spatial orientation with respect to the object is changing. Also, in the real world this is further confirmed by kinaesthetic feedback if they are moving. It was conjectured that because neither kinaesthetic feedback nor specific points of reference were provided in this desktop VR, the participants were unable to

---

15 The forum contributions were quoted exactly the same looks as they were posted. This is to preserve any clues of social expressions such as the use of font size, colour, slang, typo and emoticon for grounded analysis.
differentiate between spatial visualisation and orientation and thus intuitively came to the incorrect conclusion that the turtle was moving.

To overcome their incorrect intuition that the turtle rather than they themselves were moving, the researcher asked the participants to pay attention to the compass at the top of the window and the background stimuli in VRMath. When they paid attention to the compass and/or the background stimuli, they soon discovered that in fact they were moving through the virtual 3D world.

4.3.3 Learning Activity 3: Turtle dance

In this activity, the participants were introduced to VRMath’s Logo-like programming language that enabled changes to be made to the turtle’s position and orientation. To facilitate learning of these commands, the researcher designed a tool named “Quick Command,” which included these commands within a GUI (Graphic User Interface) dialogue (see Figure 3.16).

In Quick Command, any icons or buttons clicked will produce the corresponding command sent to the Command Input Box (see Figure 3.6). After a few tries on Quick Command and watching the effect on the turtle, the participants built up some basic knowledge about these commands. Then the participants tried these commands again by typing commands in the Command Input Box. Some general mistakes occurred when the participants were typing these commands. These mistakes included misspelling of commands, no space between command and argument, and lost cursor focus of Command Input Box.

During their practice in using VRMath’s commands, the researcher found that Emilie preferred using GUI Quick Command, while Anya preferred typing directly into Command Input Box. They both, however, preferred typing when they knew that they could recall the command history by using Up and Down arrow keys. The function of recall command was particularly useful when more complex commands such as the REPEAT command were introduced.

The researcher was very cautious when introducing the semantics and syntax of VRMath’s commands to the participants. The semantics of command were introduced and discussed before the syntax of command. For example, when introducing the REPEAT command for the purpose of animation (turtle dance), the researcher explained, “If we want to repeat some actions, we need to tell the turtle
how many times to repeat with some commands” and then brought in the syntax “REPEAT repeat_counts [commands list].” Because of careful scaffolding provided by the researcher, both participants were able to make sense of the syntax and build VRMath commands such as repeat 12 [rt 30] (which made the turtle repeat twelve times of turning right 30 degrees).

The participants were able to substitute the rt command in repeat 12 [rt 30] with other turning commands such as LT (left turn), RU (roll up), RD (roll down), TR (tilt right), and TL (tilt left) to animate the turtle. They also tried to change the counts and degrees in that command to play with the turtle. During their play with the turtle dance, they were able to predict the turtle movements before they inputted the commands. This showed that the participants could use the turtle as the reference point for performing rotation in 3D virtual space. However, it was found that the participants could not mentally think of the all rotations in 3D. When asked about the directions of 3D rotation without using computer, the participants talked about moving left and right and rollup and rolldown but both ignored the rotations of tiltleft and tiltright.

4.3.4 Learning Activity 4: Formula of polygon

In previous activity, the participants learnt the repeat command. In this activity, the researcher began with a discussion about how a square could be constructed by repeat 4 [fd 1 rt 90] commands. The two participants physically paced out a square to generalise the commands. They were then asked to use the similar syntax to construct polygons.

Anya first tried to create a pentagon. She drew a pentagon on a piece of paper and tried to measure the degrees by estimation. Because the pentagon she drew was not equilateral, she found that there were two right angles, two obtuse angles and one acute angle in the pentagon. The researcher then advised her that we could make all angles the same in the pentagon. She then started to guess the degree of the angle and replace the degree in the following formula:

\[ \text{repeat 5 [fd 1 rt degree]} \]

Anya tried degrees such as 110, 120, 180, 190, and 280 etc. After a shape was drawn by the turtle, she navigated to view the shape. She found that 290 degrees was almost right for a pentagon.
Emilie also tried to generate a formula for drawing a pentagon on VRMath. She was unsuccessful in this endeavour. However, she found that if 120 degrees was the input to the procedure `repeat 5 [fd 1 rt degree]`, she ended up with an equilateral triangle. The researcher then generated a table (Table 4.1) to organise their exploration and stimulate their thinking.

Table 4.1  
*Generalisation of the Polygon*

<table>
<thead>
<tr>
<th>Name</th>
<th>Shape</th>
<th>Sides (repeat)</th>
<th>Degree</th>
<th>Relationship</th>
</tr>
</thead>
<tbody>
<tr>
<td>Triangle</td>
<td>△</td>
<td>3</td>
<td>120</td>
<td>3 x 120 = 360</td>
</tr>
<tr>
<td>Square</td>
<td>□</td>
<td>4</td>
<td>90</td>
<td>4 x 90 = 360</td>
</tr>
<tr>
<td>Pentagon</td>
<td>⬤</td>
<td>5</td>
<td>290 (?)</td>
<td>5 x (?) = 360</td>
</tr>
<tr>
<td>Hexagon</td>
<td>⬤</td>
<td>6</td>
<td>?</td>
<td>6 x ? = 360</td>
</tr>
</tbody>
</table>

Emilie used a calculator to get 72 degrees for a pentagon. Anya also calculated 45 degrees for an octagon. The researcher then informed the participants that the mathematical operation could be written in the formula. Thus with the help of the researcher, the final formula for polygon was formed by the participants as: `repeat sides [fd 1 rt 360/sides]`. The researcher then posed a question “What will it look like if there are 360 sides?” The participants answered “a circle”, which they verified with the final formula within VRMath.

### 4.3.5 Learning Activity 5: Creating a cube

In this activity, the two participants were asked to create a frame of cube, write commands in a procedure, and save a public project. Emilie first tried on Quick Command tool to construct a cube. She was very confident in using the tool. During the process of using Quick Command, although the construction didn’t look like a cube, she didn’t navigate to change the viewpoint. Thus, two traditional errors for egocentric reference that have been termed the egocentric bug (Fay & Mayer, 1988) emerged. That is, when the turtle is facing Emilie, she often intuitively clicked on forward for back and back for forward, and left for right and right for left. This didn’t happen when the participants were typing commands. It was conjectured that because of the GUI, the arrow icons misdirected her to this egocentric bug.
The GUI interface of Procedure Editor was found to be easy to use. The participants also easily accepted the format of a procedure (i.e., to procedure_name commands end) and started to write down commands while mentally thinking the position and rotation of the turtle. It was found that the participants had great difficulty remembering the orientation of the turtle especially when more than one dimension of rotation was used (e.g., left and right is one dimension, rollup and rolldown is another dimension). The researcher then advised them to use other frame of reference commands such as up and down to avoid using rollup or rolldown. It was found as stated in the literature (Yakimanskaya et al., 1991) that the use of fixed frame of reference in addition to egocentric frame of reference aided the construction of a cube significantly. The participants then successfully completed their procedures (see Table 4.2).

Table 4.2  
*Procedures of a Cube*

<table>
<thead>
<tr>
<th>Anya’s procedure</th>
<th>Emilie’s procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TO acube</strong></td>
<td><strong>TO ecube</strong></td>
</tr>
<tr>
<td>cs</td>
<td>cs</td>
</tr>
<tr>
<td>pd</td>
<td>pd</td>
</tr>
<tr>
<td>repeat 4 [fd 1 rt 90]</td>
<td>repeat 4 [fd 1 rt 90]</td>
</tr>
<tr>
<td>up 1</td>
<td>up 1</td>
</tr>
<tr>
<td>repeat 4 [fd 1 rt 90]</td>
<td>repeat 4 [fd 1 rt 90]</td>
</tr>
<tr>
<td>fd 1 down 1</td>
<td>east 1</td>
</tr>
<tr>
<td>rt 90</td>
<td>dn 1</td>
</tr>
<tr>
<td>fd 1 up 1</td>
<td>north 1</td>
</tr>
<tr>
<td>rt 90</td>
<td>up 1</td>
</tr>
<tr>
<td>fd 1 down 1</td>
<td>west 1</td>
</tr>
<tr>
<td>pu</td>
<td>dn 1</td>
</tr>
<tr>
<td><strong>END</strong></td>
<td><strong>bk 1</strong></td>
</tr>
</tbody>
</table>

These two procedures from Anya and Emilie were very similar. Both procedures correctly produced a frame of a cube. However, the researcher found that the use of fixed frame of reference commands instead of egocentric commands actually produced different results. For example, when the turtle is in tiltleft 45 orientation, the two procedures above will not create a cube (see Figure 4.1). This could be further developed in the activity and discovered by students.
4.3.6 Usability inspection

The usability inspection was undertaken through the observation of the participants’ use of VRMath system and the focus interview at the end of each session. The observation and interview focused on the five attributes of usability namely learnability, efficiency of use, memorability, error recovery and prevention, and subjective user satisfaction (Nielsen, 1993).

The researcher observed that the participants had some difficulties in using VRMath system. However, when interviewed the participants claimed that it was easy to learn and remember. For example, Emilie spent about 12 minutes in finding the Fit Screen icon. Despite this, she still thought that it was easy to use and remember. The researcher found that the participants could fluently switch between and utilise the VR interface, programming interface, and hypermedia forum interface. During the five sessions, no systemic errors occurred. Both the participants indicated that they had enjoyed VRMath very much.

However, three main suggestions regarding the interface and activity design of VRMath were made by the two participants.

1. The dialogues such as Quick Command and Material Editor should stay in foreground instead of being sent to background when users temporarily switched to navigate in 3D virtual space. This issue had been noted by the researcher. However, due to the limitation of the programming language, it was not possible to change in the short duration of this iteration.

2. The current prototype of VRMath could only save/store the users’ constructions of procedures in projects. The participants wished that VRMath could also save/store their constructions of 3D microworlds.
created in the VR interface, so they could share with each other easier. This was an excellent suggestion as the 3D visualisations are rich and intuitive representations of 3D geometry. The researcher will consider implementing this in the future iterations of the design of the VRLE.

3. The requirements for more games such as the “turtle’s eyes” to be designed in activities. Their reason for making this suggestion was that they found that “doing” the game was much fun. This suggestion has been adopted by the researcher. However, since the purpose of games such as “turtle’s eyes” is to facilitate the students’ skill in 3D navigation, the design of game-like activities needs to avoid shifting the students’ attention away from learning.

4.4 IMPlications for iteration 2

The findings from Iteration 1 that reflect on issues of the design of VRMath and learning of 3D geometry within VRMath were identified. These findings, such as the emergence of the traditional egocentric bug and the use of multiple frames of reference to aid the construction of 3D geometrical objects, have confirmed results from past research literature (e.g., Fay & Mayer, 1988; Yakimanskaya et al., 1991). But more importantly for the future phases or iterations of the design experiments research study, some of the findings from this study have raised issues that hitherto have not emerged in the research literature. Some of these findings and their related issues have great import not only for the later phases or iterations of this study but also for other research and development focusing on the learning of 3D geometry in 3D VR environments. The following implications have emerged from the findings for Iteration 2 of the study.

The first implication to emerge was that the 3D VR environment in VRMath should provide users with the opportunity to operationalise both their spatial visualisation and orientation abilities. This is in contrast with most other ICT tools that enable users only to operationalise their spatial visualisation abilities. However, because kinaesthetic feedback from VRMath only comes from the mouse and keyboard, users often think that they are manipulating the geometrical objects as they do with other ICT tools. Therefore, users seem to perceive that the turtle rather than they are moving within the 3D virtual environment. To overcome this problem and to
enable users to successfully utilise their spatial orientation abilities within the VRMath environment, environmental cues such as background and compass need to be emphasised to allow users to distinguish between movement by objects and movement by themselves within the 3D virtual environment. In addition, the scaffoldings used by the researcher can help. For example, the researcher can use “if you can navigate or walk to see the turtle’s eyes…” in discourse with users.

During this iteration, the participants found the process of navigating in the 3D virtual space of VRMath rather challenging. VRMath enables users to navigate themselves to get multiple viewpoints in the 3D virtual space. This is invaluable in the process of constructing knowledge about 3D geometry. However, the participants often found that it was difficult to control their navigation speed and direction. To overcome the difficulties in 3D navigation, the three modes of navigation Walk, Fly, and Examine need to be explicitly discussed with the users before giving information about the use of mouse and keyboard. The three modes of navigation are actually informative metaphors for building conceptions about 3D navigation. For example, the walk mode has a gravity effect. These discussions may help alleviate the memory load of 3D navigation. Also, more time should be devoted for users’ practice of 3D navigation prior to their engagement with complex design of 3D objects tasks.

The third implication derived from Iteration 1 was that the use of multiple frames of reference commands could aid the construction of 3D geometrical objects significantly. But more importantly, the different frames of reference commands have different effects on the construction of 3D geometrical objects. The implications of this finding for later phases or iterations of the study are that the different effects of constructing 3D geometrical objects emanating from the use of different frames of reference should be developed and integrated into the learning activities. Also, the users’ understanding about the use of different frames of reference are worthy of further investigation.

4.5 Refined Learning Activities

Based on the analysis of data from Iteration 1, the five learning activities were refined and four more learning activities were designed for next iteration of enactment and evaluation. Each activity was presented with a title, purpose, description, time and applicable PEs (Perspectives for Evaluation, see Section 2.5.2).
In addition, usability inspection was included in the evaluation of all learning activities.

Learning Activity 1: Become a member

Purpose: To register as a member of VRMath community and familiarise with forum interface.

Description: User is required to register online into VRMath Forum. This involves setting the user’s profile including login name, password, email, signature, and avatar. After registration, the user is also required to post in the Test Forum, and send and reply private messages to other users. This enables the user to participate in online discussions and to create projects within the learning environment.

Time: 1 hour.

Evaluation: Usability, PE2.

Learning Activity 2: The secret and the legend of the turtle

Purpose: To learn 3D navigation and the use of navigation aids.

Description: Two games were designed to help the learning of 3D navigation and the use of navigation aids. The turtle’s eyes change colour when the user navigates very close to the turtle’s eyes (approximately 17cm in virtual space). The legend of turtle is that when user navigates to the position down below the turtle about 7 metres, the turtle will turn into a jet plane. The user is required to see these two changes of the turtle through 3D navigation in VR space, and post findings in forum.

Time: 1 hour.

Evaluation: Usability, PE1, PE2.

Learning Activity 3: Turtle dance

Purpose: To learn basic commands for moving and turning the turtle.

Description: The moving commands (egocentric: FD, BK fixed: EAST, WEST, SOUTH, NORTH, UP, and DOWN) and turning commands (RT, LT, RU, RD, TL, TR) are introduced in this activity. Also, REPEAT command is introduced here to animate the turtle. For example, repeat 12 [rt 30] will result in the turtle right turn a round.
Learning Activity 4: Shapes and scale

Purpose: To learn scale in 3D

Description: In this activity, five primitive 3D objects (CUBE, SPHERE, CYLINDER, CONE, and LABEL) and scaling commands (SCALE, SCALEWIDTH, SCALEHEIGHT, and SCALEDEPTH) are introduced. User is required to predict and experiment with different scales on different objects. For example, what will a cone look like when the depth scale is set to zero? The answer is a triangle.

Time: 1 hour.
Evaluation: Usability, PE1, PE2, PE5.

Learning Activity 5: Climb up stairs

Purpose: To build a first “microworld” using previously learnt commands.

Description: In this activity, the user utilises previously learnt commands including moving, turning, shape primitives, and scale etc. to build staircases, and experience walking on the staircases. User is required to post findings in the Forum.

Time: 1 hour.
Evaluation: Usability, PE1, PE2, PE7.

Learning Activity 6: 3D rotation

Purpose: To find out the fact of non-commutativity in 3D rotation.

Description: In this activity, user is required to: (1) solve a question: Will you face the same direction when you turn right 45 degrees first then roll up 45 degrees, or you roll up 45 degrees first then turn right 45 degrees? and (2) post findings in the Forum.

Time: 1 hour.
Evaluation: Usability, PE6, PE7, PE8.

Learning Activity 7: Formula of a polygon
Purpose: To generalise a formula for a polygon.

Description: In this activity, user is asked to generate a formula using the REPEAT command. The desired formula is \texttt{repeat :side [fd 1 rt 360/:side]}. User will try to generalise the formula through the pattern and relationship found in the experiments of triangle, square, pentagon, hexagon, heptagon, and octagon etc. Table will be utilised to scaffold the pattern and relationship of polygons.

Time: 1 hour.

Evaluation: Usability, PE1, PE2. PE4.

Learning Activity 8: A frame of a cube

Purpose: To create a frame of cube, write a procedure and save a project.

Description: User will be using Quick Command first to create a frame of cube. Then user mentally imagines the process of walking a cube and write commands into a procedure using Procedure Editor. The procedure will then be saved into a project.

Time: 1 hour.

Evaluation: Usability, PE1, PE2, PE3, PE5, PE6, PE8.

Learning Activity 9: Final project

Purpose: To collaboratively work in a team and integrate knowledge about VRMath.

Description: User will be working in a group of three students for a geometry project. The topic of the project will focus on landscape of the real world. The group has to discuss the topic and plan the tasks of the project. Progressive reports about the project will be reported in forum.

Time: 6 hours.

Evaluation: Usability, PE2, PE6

These Learning Activities (LAs) and their relevant Perspectives for Evaluation (PEs) are summarised in Table 4.3. As can be seen, all the eight PEs are covered at least twice through the nine LAs. This is, however, a tentative establishment of links between LAs and PEs for Iteration 2 at this stage. More
connections between LAs and PEs have emerged during Iteration 2, and were reported in Chapter 10: Reflection and Redesign.

### Table 4.3

*Learning Activities and Perspectives for Evaluation*

<table>
<thead>
<tr>
<th>LA1</th>
<th>LA2</th>
<th>LA3</th>
<th>LA4</th>
<th>LA5</th>
<th>LA6</th>
<th>LA7</th>
<th>LA8</th>
<th>LA9</th>
</tr>
</thead>
<tbody>
<tr>
<td>PE1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>PE2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>PE3</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>PE4</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>PE5</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>PE6</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>PE7</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>PE8</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

### 4.6 SUMMARY

This chapter reported on the first iteration of the enactment and evaluation stage of the design experiments. This first iteration (Iteration 1) was short yet important to guide the next iteration (Iteration 2). In summary, Iteration 1 has set out the following research protocols for Iteration 2:

1. The design and method of the enactment and evaluation: This included the selection of the participants (two primary school students in Year 6 and 7), data collection and analysis (using grounded qualitative methodology), and procedures to carry out the field testing of the VRLE.

2. The design and enactment of the five preliminary learning activities: The five learning activities utilised many special features (e.g., 3D navigation, construction of virtual objects in 3D virtual space using programmable language) to enable new ways of thinking and doing 3D geometry for learners, and new ways of investigating and researching the learning for teachers and researchers.

3. The findings from Iteration 1 confirmed some findings from the research literature such as the egocentric bug and the use of multiple frames of reference to aid the construction of 3D virtual artefacts. The findings from Iteration 1 also identified new research interests such as the confusion of spatial orientation and visualisation in the 3D virtual space, the difficulty
in 3D navigation using 2D devices (usability issue), and the effect of using
different frames of reference (i.e., egocentric and fixed) commands to
construct 3D objects. These had implications and became major research
foci for Iteration 2.

4. The five learning activities were refined and four new learning activities
were designed for Iteration 2. These nine learning activities were linked to
the perspectives for evaluation (see Section 2.5.2) for more structured and
in-depth investigation.
CHAPTER 5
ITERATION 2: DESIGN AND METHOD, AND PRE-INTERVIEW

5.1 OVERVIEW

Iteration 2 was a more extensive enactment and evaluation of this design-based research. Because of this, the report on Iteration 2 has been spread over Chapters 5-9.

Chapter 5 reports on the design and method, and the pre-interview for Iteration 2. The design and method of Iteration 2 built on from and extended from Iteration 1 to include the selection of the participants, data collection, data analysis, and experimental procedures. The number of participants was increased from 2 to 6 younger aged students (Year 4 and 5). Data collection methods were enhanced by pre- and post-interviews, concurrent think-aloud protocols in interviews and teacher-student interactions, and improved computer auto-logging programs. Data analysis was further strengthened to focus on and develop themes as informed by the perspectives for evaluation (see Section 2.5.2). The procedures clearly set out the steps and timeline for the interviews and enactment of the nine learning activities.

The pre-interview aimed to familiarise the researcher with the participants. It served as a preliminary analysis of the participants’ prior experiences with computers and 3D graphics, mathematics beliefs, preferences and achievements. The understanding of the participants’ mathematical and technological background was important as this information was used to inform the choice of instructional strategies utilised with each group. The results of the pre-interview were recorded as the initial state of the participants for later comparison during the enactment of the nine learning activities and in post-interview.

5.2 DESIGN AND METHOD

5.2.1 Participants

According to research literature on programming, learners from the age of preschooler to adult can all enjoy programming in Logo (see Oakley & McDougall,
1997). However, most of the geometrical concepts to be explored within the context of this study are scheduled in the Queensland Mathematics Syllabus (Queensland Studies Authority, 2004) at Year Levels 4-5. Accordingly, the participants selected for Iteration 2 were two groups of three Year 4-5 students (Aged 9-10 years-old) from two urban schools in south-eastern Queensland (School C and School K). The students were selected only on one criterion, which was that the students should have basic skills of operating a mouse and keyboard to avoid the frustration and waste of time familiarising themselves with the computer hardware.

Each of the participants was allocated a pseudonym. The pseudonyms for the participants from School C were: Bonbon, Rosco, and Grae. The pseudonyms for the participants from School K were: Alekat20, R2D2, and Victor. These pseudonyms were actually the nicknames the participants chose to use in VRMath Forum.

The small size of the sample could be justified by the large quantity of data that was generated by each participant and also by the fact that a major aim of the study was theory building. According to Cohen, Manion, and Morrison (2000), theory-building qualitative research tends to operate with such small samples. To enable the investigation of collaborative learning, the two groups interacted through VRMath Forum, the social-actional component of VRMath.

5.2.2 Data collection

Data were derived from the following five sources:

1. Observations: During the orientation and learning activities, the students were observed. In order to facilitate the process of data analysis, videotaped recordings of children’s interaction with their peers and the VRLE were made and transcribed in full.

2. Concurrent think-aloud interviews: During the orientation and learning activities, the students were asked why and how they were thinking and doing when conceptual changes or new ideas emerged. Students were required to answer straight away to reflect their immediate thoughts. These data were noted and video-audio recorded as much as possible.

3. Focus group and individual interviews: There were interviews for students after each session on the VRLE. Focus group interviews were conducted
prior to individual interviews. The interviews focused on: (a) system design issues (usability questions), and (b) issues pertaining to the construction of 3D geometry concepts and processes. The interviews were videotaped for later transcription and analysis.

4. Auto-logging program: A specially designed mechanism in VRLE recorded each student’s interaction within VRLE including time and command history for further analysis.

5. Artifacts: These data included students’ construction of 3D microworlds, projects (e.g., Logo procedures), and the discussion in hypermedia forum.

5.2.3 Data analysis

The data collected were qualitatively analysed. Data analysis in this iteration of the study followed the steps below.

1. Transcription: The data derived from tape-recorded interviews, and video recordings were transcribed in full. Because non-verbal cues such as communication of embarrassment or emotional distress, or simply a pause for thought are important elements (Hancock, 1998; Lacey & Luff, 2001), in addition to words spoken, non-verbal cues such as facial expression and body movement were also recorded at appropriate places in the transcriptions.

2. Organisation of data: This reduced the large amount of data from the transcriptions. Data were organised into easily retrievable sections with pseudonyms and/or code numbers.

3. Familiarisation: Prior to formal analysis, the audio/video tapes and written notes were reviewed, and memos and summaries from the original data were made.

4. Coding: The data were categorised and coded into five usability attributes (see Section 2.3.6.3) and into categories based on the eight perspectives for evaluation (see Section 2.4.2).

5. Identifying Themes or Commonalities: The coded data were further analysed to identify new themes and the major and minor categories.
under each themes. Analysing techniques such as charting, comparing and concept mapping were utilised.

6. Conclusion drawing/verification: This involved interpretation of the phenomena and relationships in/between themes and their sub-categories. There were confirmative and generative findings drawn in verification, comparison and justification with informing theories and design principles of the VRLE.

5.2.4 Procedure

There were three steps in this phase of the study.

Step 1: Pre-interview

The purpose of the interview was to gain general background knowledge about each of the participants. In this interview, the researcher questioned the participants about their:

1. Mathematics preference and achievement;
2. Beliefs and conceptions about mathematics;
3. Geometrical understanding about angle, shapes and their properties;
4. Computer experience about programming, Internet, 3D games, and online community; and
5. History of involvement in research.

Based on the review of the literature, it was felt that these factors could influence how and what each participant experienced and learnt during the following learning activities. For example, it was anticipated that participants with prior 3D graphics experiences with computer games had the potential to master the intricacies of VRMath more easily than other participants and thus learn more from VRMath. Also, it was anticipated that participants who considered mathematics being a creative endeavour where one is engaged in the construction of new knowledge would interact quite differently from participants who considered mathematics as a set of static information to be memorised. Questions about history of involvement in research were asked because it was necessary to find out whether they had had prior experiences in collaborative learning communities. Prior experiences in such
learning communities would possibly facilitate how the participants utilised VRMath Forum. The pre-interview questions are listed in Appendix 4.

**Step 2: Enactment of the nine learning activities**

The enactment of the learning activities generally followed their design, but varied in time or sequence depending on real situation and contingency factors. For example, more programming commands were introduced whenever it was appropriate during activities. During learning activities, the participants engaged in within-group discourse. They were also encouraged to communicate with the other group of participants through the public forum and private messages. Each activity began with the sharing of other participants’ works from the previous activity, and ended with the posting of the outcomes from the current activity on to VRMath Forum.

**Step 3: Post-interviews with each group and participant**

Individual interviews were conducted after the participants had completed the final learning activity. These interviews focused on the participants’ understanding and learning about geometry and programming. Focus group interviews were conducted after the individual interviews focusing on usability issues. The questions administered in these interviews are listed in Appendix 5.

**Timeline for iteration**

Iteration 2 was conducted over eight consecutive weeks, two sessions per week, and one hour per session. Table 5.1 presents the timeline for the iteration.

<table>
<thead>
<tr>
<th>Table 5.1</th>
<th>Timeline for the Enactment of VRMath</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week</td>
<td>1</td>
</tr>
<tr>
<td>Session</td>
<td>X</td>
</tr>
<tr>
<td>Interview 1</td>
<td>X</td>
</tr>
<tr>
<td>Activity 1</td>
<td>X</td>
</tr>
<tr>
<td>Activity 2</td>
<td>X</td>
</tr>
<tr>
<td>Activity 3</td>
<td>X</td>
</tr>
<tr>
<td>Activity 4</td>
<td>X</td>
</tr>
<tr>
<td>Activity 5</td>
<td>X</td>
</tr>
<tr>
<td>Activity 6</td>
<td>X</td>
</tr>
<tr>
<td>Activity 7</td>
<td>X</td>
</tr>
<tr>
<td>Activity 8</td>
<td>X</td>
</tr>
<tr>
<td>Activity 9</td>
<td>X</td>
</tr>
<tr>
<td>Interview 2</td>
<td>X</td>
</tr>
</tbody>
</table>
During Iteration 2, data were collected through the five methods described in Section 5.2.2.

5.3 PRE-INTERVIEW

The pre-interview was a preliminary analysis of the participants’ prior experiences with computers and 3D graphics, mathematics beliefs, preferences and achievements. The understanding of the participants’ mathematical and technological background was important as this information was used to inform the choice of instructional strategies utilised with each group.

The pre-interview was also utilised to familiarise the researcher with the six participants, and to familiarise the participants with the research project. The researcher first gave a brief introduction about the VRMath project (e.g., try out the VRMath software), the research timeline (e.g., over a period of eight weeks and session time), and the environments (e.g., meeting room and computer settings) etc. The semi-structured interview followed this introduction and included a list of questions related to the participants’ mathematics and computer background. The process of the interview was flexible: when appropriate, follow up questions were administered.

5.3.1 Results

Mathematics preference and achievement

When asked “Why were you chosen for this project?” R2D2 replied “Because we are good at maths and computers”. This was not expected by the researcher as the selection criteria for choosing the participants was that the participants should possess basic skills of operating computers with keyboard and mouse, regardless their school performance\(^{16}\). Later on, the researcher found that School K participants Victor, Alekat20, and R2D2 achieved high distinction, distinction, and distinction respectively in the Australia mathematics competition of Year 5 students. School C participants Rosco, Bonbon, and Grae were more average mathematics students according to the school Principal.

\(^{16}\) The researcher communicated with school principals about the selection criteria in the meetings prior to this iteration of the study.
School C participants expressed that they didn’t have particular interest in mathematics but were “OK” with mathematics. They said that they were interested in English and reading. Similarly, School K participants expressed that they didn’t particularly like mathematics, but were “OK” with mathematics. They also expressed that they didn’t like to write down the sums or do calculations.

**Beliefs and conceptions about mathematics**

With respect to mathematical beliefs and conceptions, the researcher asked three main questions: “What is mathematics?”, “Is mathematics discovered or invented?”, and “Where do you use mathematics?”

When asked what mathematics is, the participants all replied with strong connection to numbers and operations. For example,

*Alekat20: Mathematics is a way of presenting numbers and showing space (She gave an example of $1 + 1 = 2$).*

*Victor: Maths is like calculating things; like to find out where the sun is gonna go, and the Earth is (spinning with his hand gesture). For example, “95 squares”.*

*R2D2: Algebra, adding things together and multiply… to get the sum.*

Bonbon and Rosco also replied that mathematics is about numbers and calculations such as addition, substraction, multiplication and division. They gave an example of using mathematics to buy things in the tuckshop. School K participants gave more examples of where they used mathematics, which also contained a strong sense of “school maths” and mathematics as being numbers and calculations. For example,

*Alekat20: At school for examination for learning.*

*R2D2: Money, using money in tuckshop.*

*Victor: At home, when you are counting the pokers that you get 1, 2 3...*

From the reactions on the participants’ faces, it was quite apparent that the question "Is mathematics discovered or invented?" was not a common question for these young participants. However, when asked, the participants from School K generated very insightful replies. For example,

*R2D2: Invented. You can’t really discover mathematics. You have to make a few things up.*

*Victor: Invented. Like the base 10 thing, is like made up by Indians.*

*Alekat20: Invented by some mathematicians with curly hairs. I think mathematics is discovered, and you can invent it as well. Because you can discover it … you can’t exactly discover it by … oh, there is a map, there is...*
a map and turn it around. You can only discover it by … I guess … your mind.

Their conceptions of mathematics seemed to be quite fallibilist (Ernest, 1999) in nature. In contrast, the participants from School C had a more absolutist view about mathematics. Rosco and Bonbon thought that we can both discover and invent mathematics, while Grae thought we can only discover mathematics. When challenged to invent some mathematics, they seemed to be puzzled by this challenge and were unable to give an example.

**Geometrical understanding**

With respect to geometrical understandings, the researcher asked the participants questions about 3D shapes and their properties such as vertex, edge, face, and angle etc.

The participants from School K appeared quite knowledgeable about 3D shapes considering their ages. For example, when asked if they could name some 3D shapes, participants from School K replied cube, sphere (Aleket20 said circle), cylinder, triangular prism, rectangular prism, triangular based pyramid. And when asked: Is a pyramid different from a prism? the following conversation ensued:

**Alekat20 and R2D2:** Yes.
**Alekat20:** A pyramid only has three sides... no, that's four sides. It has a vertices (according to her pronunciation) at the top, and then all the sides come down. A pyramid has one vertices, and multiple vertices down the bottom.
**Researcher:** You talked about vertex?
**Alekat20:** Vertices. It is a point where multiple sides meet.
**Researcher:** sides?
**Alekat20:** multiple edges.
**Researcher:** About angles, how many degrees is a right angle?
**All:** 90
**Researcher:** How about the triangle? There are three angles...
**R2D2:** Acute angle.
**Researcher:** What is an acute angle?
**Alekat20:** An angle that is less then 90 degrees.
**Researcher:** How about greater then 90 degrees?
**Alekat20:** obtuse angle, that is greater then 90 degrees.
**Victor:** And then there is reflex angle.
**Alekat20:** Reflex angle that goes like ... (hand gesture for a circle)
**Researcher:** If you turn around, that is how many degrees?
**All:** 360 degrees.
**Researcher:** What is the sum of the three angles in a triangle?
**Victor:** 180.
**Researcher:** Are you sure? any type of triangles?
**Victor:** Yah.
Researcher: How about rectangle?
Victor: 360.
Researcher: How about pentagon?
Alekat20: Pentagon has five sides, which will be... 450 degrees (She doesn’t have a pattern to figure out, just guess).

All participants from School C were able to name 3D shapes such as sphere, cone, cylinder, cube, prism, and pyramid. They could also use correct terminologies such as vertex, edge, and face on 3D shapes. However, Bonbon and Rosco didn’t know what a right angle was. Grae was a little bit unsure what cylinders and pyramids were.

All six participants were asked to draw perspective 3D shapes with dotted lines to represent hidden edges of the following shapes: cube, cone, cylinder, triangular based prism, and rectangular based pyramid. Their drawings are displayed in Table 5.2.

Table 5.2
Participants’ Drawing of 3D Shapes in Pre Interview

<table>
<thead>
<tr>
<th>School K</th>
<th>School C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Alekat20</td>
<td>R2D2</td>
</tr>
<tr>
<td>Rosco</td>
<td>Bonbon</td>
</tr>
<tr>
<td>Cube</td>
<td>[Image]</td>
</tr>
<tr>
<td>[Image]</td>
<td>[Image]</td>
</tr>
<tr>
<td>Cone</td>
<td>[Image]</td>
</tr>
<tr>
<td>[Image]</td>
<td>[Image]</td>
</tr>
<tr>
<td>Cylinder</td>
<td>[Image]</td>
</tr>
<tr>
<td>[Image]</td>
<td>[Image]</td>
</tr>
<tr>
<td>Triangular based prism</td>
<td>[Image]</td>
</tr>
<tr>
<td>[Image]</td>
<td>[Image]</td>
</tr>
<tr>
<td>Rectangular based pyramid</td>
<td>[Image]</td>
</tr>
<tr>
<td>[Image]</td>
<td>[Image]</td>
</tr>
</tbody>
</table>

These perspective drawings revealed some difficulties in their 3D visualisations. For example, R2D2, Rosco, Bonbon and Grae couldn't visualise the perspective drawing of a cube. Alekat20 and Victor drew a cube with perspective lines correctly, but they couldn't differentiate between a prism and a pyramid. Rosco drew a triangular based pyramid instead of a rectangular based pyramid. Alekat20,
Victor and Grae couldn't correctly visualise the perspective of the rectangular based pyramid. In these five 3D shapes, only cone and cylinder were successfully drawn by all participants.

**Computer use background**

With respect to computer use background, all participants appeared to be very adroit in using mouse and keyboard. When asked about what they did with computers, School K participants enumerated: type stories, make slide shows, go to Internet, emails, live chat rooms, download CDROMs, and download games etc. These seemed to be the common activities that inner city students do with computers. Boys seemed to play more computer games; for example, both R2D2 and Victor mentioned that they played some 3D video games. Alekat20 didn't play any 3D games. However, she was the only one with some computer programming experience. She claimed that she had done a small amount of HTML for web pages (she showed some html code on screen). In particular, she mentioned that she did Logo programming in Grade 2.

School C participants seemed to have high Internet usage. They had used Internet quite frequently but mainly for web browsing in some popular children sites such as Neopets and Battleon. They had also used search engines such as Google and Yahoo for school study. Grae had some experience on using MSN messenger for online chatting. They all had played three dimensional games either on PC or video (e.g., Play Station or Nintendo)

**History of involvement in research**

All participants except R2D2 and Victor had never been involved in any research project. R2D2 and Victor (from School K) reported that they had previously been involved in a research project Hypatia conducted by QUT, in which they had experienced online discussion and collaboration in the Knowledge Forum® environment.

---

17 Neopets® is a virtual pet site on the Internet. It is also an online community where users exchange pets’ information and battle their virtual pets. See http://www.neopets.com/

18 Battleon (AdventureQuest) is a free online RPG played using a web browser. See http://www.battleon.com/

19 Knowledge Forum is an electronic group workspace designed to support the process of knowledge building. It is developed by Learning in Motion, Inc and the Ontario Institute for Studies in Education. For more information about Knowledge Forum, please see http://www.knowledgeforum.com/
Summary

The outcomes from the pre-interview are summarised in Table 5.3.

Table 5.3
Summary of Pre-Interview about Participants

<table>
<thead>
<tr>
<th></th>
<th>Alekat20</th>
<th>R2D2</th>
<th>Victor</th>
<th>Rosco</th>
<th>Bonbon</th>
<th>Grae</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>School</td>
<td>K</td>
<td>K</td>
<td>K</td>
<td>C</td>
<td>C</td>
<td>C</td>
</tr>
<tr>
<td>Age</td>
<td>10</td>
<td>10</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>Gender</td>
<td>F</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>F</td>
<td>M</td>
</tr>
<tr>
<td>Year</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Mathematics background</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Preference</td>
<td>Yes but not calculation</td>
<td>Yes</td>
<td>Yes</td>
<td>Ok and prefer English</td>
<td>Ok and prefer English</td>
<td>Yes, OK, Reading</td>
</tr>
<tr>
<td>Beliefs</td>
<td>Both discovered &amp; invented</td>
<td>Invented</td>
<td>Invented</td>
<td>Both</td>
<td>Both</td>
<td>Discovered</td>
</tr>
<tr>
<td>Geometry</td>
<td>All participants are able to name and recognise 3D shapes, but show some error in perspective drawings.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computer background</td>
<td>HTML, Logo (little)</td>
<td>No</td>
<td>No</td>
<td>Yes, Robot Lego Mindstorms</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Programming</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Internet</td>
<td>No</td>
<td>Video games</td>
<td>Video games</td>
<td>PC &amp; video games</td>
<td>PC &amp; video games</td>
<td>Video game</td>
</tr>
<tr>
<td>3D graphics</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Neopets frequently</td>
<td>Neopets frequently</td>
<td>MSN</td>
</tr>
<tr>
<td>Online community</td>
<td>Yes frequently</td>
<td>No</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.3.2 Discussion

With respect to mathematics achievement and preferences, the participants from School K seemed to have better mathematics performance and more positive feeling about mathematics than those from School C. Having this understanding in mind, School K participants were extended further than School C participants whenever it was appropriate in the learning activities.
Another finding that emerged was that the participants’ conceptions about mathematics were quite limited to "school maths", in which mathematics is about numbers and calculations. This may indicate that the current implementation of mathematics curriculum and teaching practices in both schools still focuses very much on the instrumental aspects of mathematics. It was very interesting to find that School K participants tended to think that mathematics was invented while School C participants expressed a much stronger belief about mathematics being discovered. This may indicate a relationship between beliefs about mathematics and achievement in mathematics. If mathematics is perceived as invented, then doing mathematics can be more interesting. Therefore, students who like mathematics more may achieve better in learning mathematics. This research study adopted a fallibilist view (Ernest, 1999) about mathematics as a design principle of VRMath to encourage learners to invent and create mathematics.

From the discourse in the interviews and the participants’ perspective drawings of 3D shapes, it was found that the participants’ geometrical understanding was in Level 2 in terms of van Hiele’s (1986) level of geometrical understanding (see Section 2.3.2.2). The participants could recognise and name properties of geometrical figures, but they did not see relationships between these properties. For example, the participants could recognise and name prism and pyramid, but they still showed some confusion in their drawings. Each of the participants showed some errors in perspective drawings, which indicated the limitation in their mental visualisations. It was conjectured that VRMath could provide an environment for learners to experience and improve understandings about perspective within 3D shapes.

The six participants not only had excellent keyboard and mouse operating skills but also had experience on using computers particularly on the Internet. All participants had had Internet experience including interaction within online communities such as chat rooms and instant messengers. Alekat20 was particularly good at posting online messages. R2D2 and Victor reported that they didn’t have similar online experience as to Alekat20. However, they had actually interacted with online users when they participated in Hypatia Project. Rosco and Bonbon had broadband Internet access at home. They reported that they had adopted many electronic or digital pets at Neopets, where they often met cyber friends to discuss
about pets and have pet competitions. Grae also reported to have some MSN messenger experience before but was not using it at the moment. In light of their experiences in using computers and Internet, it was decided that less time was needed on basic computer instructions and more time could be spent to focus on the use of VRMath system.

Half of the six participants reported that they had had programming experience in Logo or Lego, which were highly related to the Logo programming in VRMath. However, when questioned about their programming experience, they didn't seem to remember any commands or how the turtle graphics worked. This prior experience (i.e., programming logics and controls) was taken into account but it was still necessary to rebuild an understanding about commands and the turtle graphics.

Their 3D graphics experience in games was also valuable. With prior experience of 3D navigation in games, it was decided to focus on the participants' utilisation of 3D navigation in learning. However, in most 3D games, only first-person and/or third-person navigation are provided. The users are either walking or driving on terrains (as to the Walk mode in VRMath), or flying in the sky (as to the Fly mode in VRMath). VRMath provided a third navigation mode "Examine", which is commonly used in CAD (Computer Aided Design) or 3D models building software. The Examine mode provides the advantage of quickly changing viewpoints of the 3D scene. Therefore, the Examine mode is what the researcher expected to be used frequently when users are building 3D models. And it is perhaps the navigation mode that needed more instruction and practice during the activities.

5.4 SUMMARY

This chapter reported on the design and method, and the pre-interview for Iteration 2. Six students (Year 4 and 5) from two primary schools (three students from each school forming two groups) were chosen to interact with the VRMath system for a period of eight weeks (two sessions per week, one hour per session). Data collection methods included pre- and post-interviews, observations and video and audio recordings, concurrent think-aloud protocol in interviews and teacher-student interactions, computer auto-logging programs, and the participants’ creations or artefacts (e.g., forum discussions, 3D virtual objects and programming codes).
Data analysis was linked to the perspectives for evaluation (see Section 2.5.2) to generate themes for discussion.

The major findings from the pre-interview included:

1. School K participants had better achievement in mathematics and more positive attitude towards mathematics than School C participants.

2. The participants’ levels of geometrical understanding were identified as Level 2 in terms of van Hiele’s (1986) classification, in which the participants could recognise and name properties of geometrical objects, but could not discern which properties are necessary and sufficient to describe and draw the objects.

3. The participants had excellent keyboard and mouse operating skills and Internet experience, which allowed the researcher to focus on using VRMath system rather than on using computers.

4. Some participants also had programming experience in Logo or Lego, and 3D graphics, which could advance the progress of using the VRMath system, and could be used to justify the later enactment and evaluation.

The results of the pre-interview were recorded as the initial state of the participants for later comparison during the enactment of the nine learning activities and in post-interview.
CHAPTER 6

ITERATION 2: INTRODUCTORY LEARNING ACTIVITIES

6.1 OVERVIEW

Iteration 2 was a more extensive enactment and evaluation of this design-based research. There were nine learning activities in Iteration 2, which were classified into three categories: introductory, specific concepts, and application types of learning activity. Chapter 6 reports on the enactment and evaluation of the introductory type of learning activities. This type of learning activity consisted of the first three learning activities aiming to familiarise the participants with the VRMath system.

Learning Activity 1: Become a member focused on the hypermedia and forum interface (social-actional resources) of the VRMath system. The participants were required to register in VRMath Forum and communicate with each participant through public forums or private messages.

Learning Activity 2: The secret and legend of the turtle focused on the VR interface (topological resources) of the VRMath system. The participants had to master the 3D navigation skills in the 3D virtual space to solve the two game-like problems.

Learning Activity 3: Turtle dance focused on the programming interface (typological resources) of the VRMath system. The Logo-like programming language of VRMath was introduced to the participants for commanding the 3D turtle to move (change position or location) and turn (change orientation or direction).

6.2 LEARNING ACTIVITY 1: BECOME A MEMBER

This learning activity required the participants to register themselves into VRMath Forum by creating an account name and a password. After their accounts were created, the participants logged into the forum, posted a message and wrote private messages to other participants. Various functionalities such as user profile
management, read and post messages and private message of VRMath Forum were introduced. This was crucial because once the participants had their own account and could login to the forum, the VRMath system could then trace their usages such as time, command specified and communications within forum for future analysis.

6.2.1 Results

Knowing that there were participants from another school, all participants were excited and keen to communicate with each other through the forum. The activity generally went very well as all participants seemed to have some Internet experience and good operating skills on computers.

To start the registration, the participants were instructed to go to the forum page by clicking on the forum icon (see Figure 6.1), then clicking on the register link in the main function list of the forum (see Figure 6.2).

![Figure 6.1 Forum icon](image1)

Figure 6.1 Forum icon

![Figure 6.2 Forum register link](image2)

Figure 6.2 Forum register link

To register, the user then created their username and password, and provided a valid email address (see Figure 6.3).

![Figure 6.3 Information required for Forum registration](image3)

Figure 6.3 Information required for Forum registration
The username can be any name that the participants wish to be called. The researcher suggested them to use a nick name other than their real names. The participants each selected a name as Alekat20, R2D2 and Victor from School K, and Rosco, Bonbon and Grae from School C. These names became pseudonyms and were used throughout this research study.

The email address is the unique identification for a member in the forum community. Alekat20 seemed to be an experienced Internet user. She immediately supplied her email address from which it became apparent that the name “alekat20” was already in use by her. As a result, Alekat20 quickly registered in the forum without any difficulty. R2D2 and Victor halted in this registration process as they didn’t have an email address. The researcher suggested them to give a false one as the use of email was not planned in this research. R2D2 then chose r2d2@somewhere.com.au and Victor typed in victorchen@yuhgiohdungeon.com. Both participants then went through the registration process. This, however, caused the forum system to generate an error message to the forum administrator (i.e., the researcher) as:

The original message was received at Thu, 16 Oct 2003 09:50:53 +1000 (EST) from cobia.ed.qut.edu.au [131.181.58.18]
----- The following addresses had permanent delivery errors -----
   r2d2@somewhere.com.au

Successful registrants got an auto-generated welcome email from the forum system. School C participants all had their own email address. Rosco and Bonbon both use free email from Yahoo. Grae uses the email provided by the school. Their registration process went through very well. However, they didn’t get the auto-generated welcome email as the VRMath system was setup to run in School C’s local Intranet.

After registration, the participants could login to the Forum system. The researcher expected to have some login problems such as the forgotten password. Surprisingly, no login problems occurred throughout Iteration 2. After the participants could login freely, they were introduced them to some of the community

---

20 School C used the Internet infrastructure of Education Queensland (EQ). For security reasons, EQ’s firewall disabled most direct Internet connections such as database connection. Therefore, the researcher had to install a copy of VRMath system in School C’s local network as VRMath system required the clients to establish a direct connection to its database. And as a result, the auto-generation of emails was disabled.
members by using the Forum’s “Memberlist” function. The first member introduced was the researcher. By clicking on a user’s name in the member list, the user’s profile can be seen as in Figure 6.4.

![Figure 6.4 Viewing user profile in forum](image)

When viewing the users’ profile, participants could obtain information such as number of posts, location and website etc. But they were most intrigued by the visual avatar image. Being motivated by this, participants requested and were then shown how to edit their personal profile, where they could change their personal information and select their own avatar from over 600 images in the forum system. To edit profiles was easy, but to choose an avatar was time consuming as the participants tried to browse through hundreds of avatar image.

It was interesting to find that girls only selected an avatar once but boys changed their avatars over time. Boys seemed to be more interested in changing their avatar. The avatars chosen seemed to reflect well on the participants' personalities and characteristics (see Figure 6.5). For example, female participants (i.e., Alekat20 and Bonbon) both chose a girl’s image or a neutral cartoon’s image, while male participants chose their favoured male cartoon characters.

![Figure 6.5 Initial Avatars of participants](image)
Other personal information the participants changed in their profile included mainly the location and signature. For the location, the participants seemed very keen to show where they were from. Therefore, they typed in locations such as “aussie”, “Qld Brisbane”, and even “Sunnybank Hills”. The signature is used when posting messages in the forum. Five out of the six participants utilised the signature. Male participants wrote general greeting such as “Hi peoples im rosco!!! 😊😊 hello” from Rosco and “uerdo 😊😊” from R2D2 in their signatures. Female participants created different signatures from males. Bonbon put a question in her signature “How do you get the turtles eyes to turn Red?” and Alekat20 had a two lined signature:

Can You Talk To Me????????? Please!!!!!
Can You Make the Turtles Body A Different Colour????????

The participants changed their profiles during the course of Iteration 2. The maintenance of the profiles behaviours are further analysed in Section 9.4.3.

The participants showed good computer operating skills and the VRMath system worked well. Instruction then began on the two main tasks of this learning activity: posting discussions in the Forum and sending private messages to other participants. These two functions use very similar web posting forms (Figure 6.6) that supports rich text, hyperlink and images.

![Figure 6.6 Web posting form for sending private message](image-url)
The visual layout of this web posting form appeared clear and intuitive to the participants. They managed to utilise most functions including rich text format, attach signature, preview and emoticons in this form. The participants then practiced the posting of messages in Test Forum, and the writing of private messages to other participants in COLAB user group\(^{21}\). The results showed that the participants were able to communicate through forum discussions and private messages. Without much instruction, the participants worked out how to post messages quickly by the clear visual layout of the web posting form (see Figure 6.6). During the practice time, it was found that the participants spent more time on choosing emoticons and asked for more emoticons to help express their emotions. However, sometimes they just inserted many emoticons in their messages for fun, which made their messages somewhat distractive. The researcher introduced them to online etiquette and encouraged them to discuss learning through public forums and send private messages for personal matters. For the remainder of sessions, the participants were allowed about 15 minutes before and after the learning activity to communicate using the Forum and private message. They were also free to post messages anytime during sessions if they wished. Throughout this iteration, a total of 159 posts were generated by the six participants and the researcher in the Forum, and over 150\(^{22}\) private messages were sent among COLAB group members. These posts and private messages were presented in different sections of learning activities where appropriate, and were further analysed in Section 9.4.1 and Section 9.4.2.

### 6.2.2 Discussion

By the end of this activity, together with the pre-interview, all participants have acquired a basic understanding about VRMath system particularly with the forum interface. The forum interface of VRMath provides mechanisms to enable communication and collaborative learning among users. The participants were now able to communicate through posting public and private messages. However, as a social-actional component, the forum has other social-actional resources than just posting messages. For example, users can setup a poll to collect other users’

---

\(^{21}\) COLAB was established to include all participants as a small community. User group is a function provided in this phpBB forum system to further complex the management of forum privilege. However, in this study, COLAB is only used to reveal participants’ information such as number of posts.

\(^{22}\) 150 private messages were calculated at the end of Iteration 2. During the iteration, some private messages were deleted by the participants.
opinions; they can search user information and discussions in the forums; and there is a built-in rating system for users based on their contributions to the forum. The researcher didn’t plan instructions for all these resources. Instead, the participants were expected to discover these resources and use them. It was found that most of these resources or functions such as setting a poll were utilised by the participants in later activities.

While the forum interface provides a channel for communication and enabling collaboration, the researcher was interested in both their affective and functional aspects of using forum interface (these are usability issues). The researcher investigated: (a) Did the participants get a sense of being a member, belonging to a community and not alone? (b) Did the participants enjoy using the system? (c) Could the participants easily operate the forum interface?

It was evident from the observations that all participants showed great “excitement” and were keen to get private messages or replies from other participants especially from the other school. Alekat20, for example, posted her first message in Test Forum:

😃Hi Peoples!! Can any one talk!! Hello!!! 😃I want some one to talk to!!

Can You Talk To Me?????????? Please!!!!!
Can You Make the Turtles Body A Different Colour??????????

In her post above, the last two lines signature further intensified her wish to interact with some else. This, however, was brought to the researcher's attention as the signature was used to ask questions rather than "formal signature" that displayed name and some personal information. It was thought that the signature in online communities has become a handy text ready to be attached to posts. Futher, for some online communities where users were anonymous, there was no need to display personal information in the signature. Therefore, the signature was utilised as a handy text to include question, wish, or motto etc. It could be argued that some participants might have incorrectly used signature. However, the researcher did not intend to interfere with the use of signature. Instead, the participants' spontaneous use of signature was observed and recorded for further analysis.

The participants also asked the researcher for other school participants’ personal information such as age, gender and school etc. The researcher then encouraged them to communicate through private messages for personal matters.
Danevans was a user of VRMath Forum as well as the grandfather of Rosco and Bonbon. He was concerned about his grand children’s participation in this VRMath Project and therefore, he kept in contact with the researcher by using private messages in the forum. After this “Becoming a member” activity, he sent this researcher messages stating that:

I'm sure that we will all get something out of this experience. I know that Rosco and Bonbon will as they are quite excited about their next session with you.

and

They enjoy experimentation and the joy of discovery. If they find a subject too easy they get bored - just the same as other kids. If it gets too hard then they tend to give up as well - but no so easy. If you give me instructions on what they should do I'm sure that I can cajole them into leaving Neopets and Operation Flashpoint.

It was found that the participants liked to use emoticons (e.g., 😊, 😎) when composing messages. However, sometimes it was felt that the use of emoticons was distractive. Most of their messages contained some proper emoticons that aided their expression. Because it was only one click to insert an emoticon, two participants in particular, had too many inappropriate emoticons inserted into their messages. This didn’t help the communication. Instead, it became an emoticon battle within sending messages.

It was impressive that all the participants were able to use the forum component within two sessions (the actual time for practicing was about one hour). They could quickly figure out the use of rich text and emoticons in posting message and the maintenance of their profiles (e.g., edit signature and avatar). This may be partly due to their previous experience on computers and Internet, or it may indicate that the forum interface was easy to use.

During this activity, there were no critical system errors except for one uncontrollable technical issue, which was the forum system notification email. When a new user registered or a private message was sent to another user, the forum system generated an automatic email to welcome or notify the user. Due to security reasons, the VRMath system installed in School C could not send email (no access to email service). Therefore, some error messages were displayed on screen to School C participants and sent to the admin account in the forum (i.e., the researcher). Fortunately, this didn’t interrupt the activity.
The use of an email address as a unique identification in an online community was adequate and popular. However, it may present some difficulty to young children because they are not yet using one or are controlled by their parents for security and child safety reasons as was the case for Rosco, Bonbon, Victor, and R2D2. The forum system provides each user a personal message box, and integrates it with a formal email system to aid online communication. However, this could not be fully exploited in this study.

6.3 LEARNING ACTIVITY 2: THE SECRET AND THE LEGEND OF THE TURTLE

This activity aimed to facilitate the participants’ 3D navigation skills in the virtual reality interface of VRMath system. The VR interface is virtually an endless 3D space embedded on a 2D screen, on which users can navigate in 3D space by mouse and keyboard. Because of the inherent dimensional mismatch, 3D navigation is usually done by the combined use of mouse and keyboard. And it is necessary to practice 3D navigation skill in order to gain vantage viewpoints in 3D space for learning. Therefore, two game-like tasks were designed to engage and scaffold participants in learning 3D navigation skills.

The first task (the secret of turtle’s eyes) was the same game designed in Iteration 1, in which a ProximitySensor sized 1 cm³ was placed between the turtle’s two eyes. If the user navigated within the range of this sensor, the turtle’s eyes changed to a random colour (in Iteration 1, they only changed to red colour). If the user navigated away from the range of the sensor, the eyes returned to their original colour of blue (see Figure 6.7).

![At a distance away the turtle shows two blue eyes](image1)

![Navigate closer to the turtle, its eyes will change colour](image2)

*Figure 6.7 The secret of the turtle’s eyes*
The second task (the legend of turtle) was that when user navigates to the position down below the turtle about 7 metres into the range of another ProximitySensor, the turtle will turn into a jet plane (see Figure 6.8).

![Navigate down below the turtle and look up](image1). ![Move closer into the range of the sensor, the turtle turns into a jet plane](image2).

*Figure 6.8 The legend of the turtle*

Before the sessions, the researcher posted four messages in the Forum introducing the documentations about 3D navigation, the secret of the turtle’s eyes, and the legend of the turtle in the Hypermedia and Forum interface. The participants were asked to read the information in the Forum, solve these two tasks and post their results in the Forum. This learning activity was conducted in Iteration 1 (see Section 4.3.2). The findings from Iteration 1 suggested that careful language and instructions should be given to avoid confusion in the users' operation of their spatial orientation and visualisation abilities. This was one of the main focuses in this learning activity.

### 6.3.1 Results

School C participants started this activity first. Before the activity, the participants were guided to read through the Forum posts about this activity and the 3D navigation document in the forums. Each of the icons (i.e., navigation mode, avatar view and restore viewpoint etc.) and its functionality in the navigation toolbar was introduced. After the introduction of the navigation toolbar, the researcher conducted a test (see Appendix 3) to identify the appropriateness of the pictorial icons and their associated functions for usability inspection. The result of this test is reported in Section 9.3: Usability inspection. After the test, there was a brief discussion followed about the three navigation modes and the use of mouse together with meta-keys (Shift, Alt, Ctrl, and Space) was briefly mentioned. Then the participants went on to solve the secret and legend problems.
The secret problem

Before the researcher introduced the turtle's eyes problem, Rosco had already navigated to see the turtle's eyes. Bonbon said that she had found the answer in “chat room” (i.e., VRMath Forum) but she didn’t really try it. She knew that the turtle's eyes will change their colour to red as that was posted by the two participants Anya and Emilie during Iteration 1.

Examine and Fly modes can basically accomplish the 3D navigation to any viewpoints. However, the navigation in the two modes only by using the mouse is not used. The use of Examine and Fly modes can basically accomplish the 3D navigation to any viewpoints. However, the navigation in the two modes only by using the mouse is not used. The use of Examine and Fly modes can basically accomplish the 3D navigation to any viewpoints.

To address this, the researcher reprogrammed the secret of the turtle's eyes. Rosco thus was surprised to find that the eyes' colour changed every time when navigating away and close to the eyes. Bonbon was excited and she joined in to make the eyes change colours in her computer. The researcher noticed that both of them used Examine mode to rotate the scene (i.e., rotate viewpoint around the turtle) and Fly mode to go near the eyes. The meta-keys were not used. The use of Examine and Fly modes can basically accomplish the 3D navigation to any viewpoints. However, the navigation in the two modes only by using the mouse is the integrated movement similar to real life movement in which we orientate (changing direction) and translate (changing location) seamlessly together. The researcher would like to involve them with the component movement by introducing meta-keys with the mouse. The question "what is virtual reality?" was first posed to Bonbon and Rosco to try to compare the difference and similarity of 3D navigation in VR and real world.

Researcher: this is virtual reality (pointing to 3D interface), what is virtual reality?
Both: ....
Researcher: is that real?
Rosco: no, but it looks real.
Rosco: it looks exactly like how we are looking in real life if its from digital..
Researcher: so is that 3D space real because you can walk and you can fly?
Bonbon: yeah... how you are supposed to fly in the water?

23 Grae was late for about 38 minutes, and he had missed most of this activity.
Researcher: yes, sometime the background doesn’t make sense, see the background is underwater; it is just a simulation, not actually underwater. You can not feel the water, and you can still breathe.
Rosco: yes, it’s kind of weird.
Researcher: yes, we just use Virtual Reality to simulate something.
Rosco: simulations that what we call it.....
Researcher: now we have three modes, that’s the important thing, and we have to know how to walk, fly, and how to examine. Do you know how to walk?
Bonbon and Rosco: yes.
Researcher: if you walk, can you jump very high?
Both: no
Researcher: because there is gravity you cannot leave the ground. But when you are walking, you can still turn your head, so you can look up and look down.

The researcher had tried to link the 3D navigation (Walk) to participants' real life experience. The use of Space key (when walking, press down the Space key) to look up and look down was introduced here. And whenever appropriate, the use of navigation toolbar such as restore viewpoint was utilised and reinforced. To further assimilate and differentiate 3D navigation in virtual space and real world, the researcher compared the two modes Walk and Fly in VRMath.

Researcher: ok, when you walk, you can look up and you can look down. If you look up and you still walk, what will happen?
Rosco: you will crash into something
Researcher: you are still on the ground, right?
Both: yes
Researcher: ok, the different thing is when you are flying, the airplane doesn’t have a head.
Rosco: yes
Researcher: So the airplane cannot look up or look down, but the airplane can do what (with hand gestures)?
Rosco: Pitch up and pitch down.
Researcher: Yes, but in VRMath we call it roll up and roll down. So when you pitch up or when you roll up, you look at that place (with hand gesture), at this time, if you go forward, where will you go to?
Both: You will go up (pointing to the heading direction).
Researcher: Yes, you go the direction you are looking at.

By now, the difference between Walk and Fly modes was discussed. In Walk mode, we are actually moving on a 2D plane. The direction that we are moving forward isn't necessarily the direction we are looking at. However, in Fly mode, we move forward to the direction we are heading (looking) to. Bonbon went on to use Examine mode. The researcher than discussed the Examine mode with participants.

Bonbon: What about the other one, the arrows... (the Examine icon has two arrows).
Researcher: Examine? Yes, it is kind of weird thing. If you examine, you are rotating the world, but who is moving? (Rosco was using Examine mode)
Both: us.
Researcher: Yes, not the turtle, not the other object, it's you! But how can you move that fast in the real world?
Both: no.
Researcher: just in the virtual reality.

By now the three navigation modes had been discussed and linked to participants' real life experience. The researcher then allowed some time for the participants to play with the 3D navigations. While they were playing, the researcher introduced the combination use of mouse and keyboard in the three modes and linked them to the participants' body movements. For example, when the Alt key was pressed down while dragging, it would mean to slide. The researcher then asked Bonbon to stand up and slide. Key questions such as "when you slide, did you change your direction?" and "when you walk, can you slide up and down?" were posed to further differentiate the three modes. By using the analogies to their body movements, it became very easy for the participants to make sense of the component movement and understand the 3D navigation in VRMath. As can be seen in the discourse and questioning, the researcher also consciously used sentences such as "you walk" or "you go" to remind participants that they were moving or navigating (acting on spatial orientation ability) rather than the turtle was moving (acting on spatial orientation ability).

The two participants were now able to navigate to see the turtle's eyes changing colours. The researcher observed that they could effectively use mouse and keyboard together to navigate. The Restore Viewpoint and Fit Screen in the navigation toolbar were utilised frequently. When using the Examine mode, Rosco found a problem in which he couldn't rotate (examine) around the turtle. This problem was due to the rotation centre was not set to the turtle's current position. Sometime during the activity, Rosco used Quick Command to move the turtle away from home position (coordinate 0,0,0), and clicked on Set Rotation Centre to a point other than coordinate 0,0,0. To solve this problem, the researcher instructed him to type in command HOME to move the turtle back to coordinate 0,0,0, and then clicked on Set Rotation Centre icon.

The participants could now navigate quite easily to see the turtle's eyes especially by using Examine mode. The researcher then challenged them to use one
navigation mode only to see the turtle's eyes, and asked "which is the best mode?" to see the secret of the turtle's eyes. Rosco was the first to give his opinion. He said that Examine mode (with the use of Space key) is the best way to see the turtle's eyes. He then posted in the forum about his finding that

```
when you get close to da turtle eyse, they turn different coulors!!! 😄😄😄
that the best way to get close is to do what hieoples said
```

and Bonbon posted her opinion about the best mode to see the turtle's eyes.

```
You use examine mode to see the turtles eyes turn a different colour. They turn purple, blue, yellow, green or white 😄😄😄😄😄
```

**The legend problem**

The legend problem was introduced by reading the message posted by the researcher in the Forum:

```
Hi all,

The turtle is really smart, it has evolved again...
When you go beneath the turtle and look up....
When the turtle fits into the sun........
Let's find out!
😊
do not change background
```

Rosco immediately replied that it was "simple". He then tried a couple of times but found it not very easy. It was observed that when he was kind of lost in 3D space; he always used Restore Viewpoint to start again. He could use Examine mode to quickly navigate beneath the turtle. However, when the turtle fit into the sun (in background), both the turtle and the sun were not in the centre of the screen (see Figure 6.9). If they were not in the centre of the screen, then the user was not right beneath the turtle. Therefore, the user wouldn't be in the range of the ProximitySensor to turn the turtle into a jet plane.

*Figure 6.9. Beneath the turtle and heading up*
Bonbon thought that she did it and called the researcher excitedly. However, she was doing exactly the same as Rosco. The researcher gave her a hint that both the turtle and the background sun must be in the centre of the screen; and that could be done by using Examine mode to centre the sun and slide to centre the turtle. Just then Rosco called out. He had navigated to the right location and found the legend of the turtle. Both Rosco and Bonbon were very happy to discover the legend of the turtle and went on to post their findings in the forum. Rosco wrote that

The turtle turns into a jet fighter.

and Bonbon built on Rosco’s post and wrote that

must keep the sun in the middle of the screen
Get right under the turtle in examine mode then press space and zoom out with the turtle in the sun it will turn into a jet.

Hey Peoplz!

School K participants started this activity on the next day. Before they explored the secret and legend of the turtle, they had read School C participants' posts in the forum. Therefore, they had known the answers from the forum. However, they were still excited to try out for themselves. The researcher discussed about the three navigation modes with School K participants in a way similar to that with School C participants, and tried to link to their real world experience by asking them to act the three navigation modes. In a manner similar to that used by School C participants, they utilised the meta-keys with mouse, and navigation aids in the navigation toolbar. When the researcher prompted them to solve the secret and legend problems, Alekat20 thought that it was difficult. However, all three participants soon navigated correctly to experience the secret and legend of the turtle.

Their processes of navigating to solve the secret and legend problems were similar to School C. They frequently used Fit Screen, Restore Viewpoint, changing navigation modes, and meta-keys. Alekat20 encountered the same problem as Rosco. She unintentionally commanded the turtle to move and clicked on Set Rotation Centre icon to set a new rotation (examine) point.

When challenged to use one mode only to see the turtle's eyes, the participants had different opinions. All participants thought that Examine mode was the easiest mode to see the turtle's eyes except R2D2, who thought Fly mode to be the best. He wrote in the forum:
man was that hard i just made the turtles eyes change colour without using fit
screen in walk, fly and examine mode. the hardest is walk mode and the easiest
is fly, if someone else tries remember that you won't get in your first try and
use slide!
remember to do them at different times(not on same turn)

In the next session, the researcher videotaped each participant's navigation for
the secret and legend of the turtle. All participants were able to use just one
navigation mode only and without using Fix Screen function to solve the two
problems. The researcher observed that School C participants all navigated by
dragging the mouse on any arbitrary angles rather than in a cross direction. The
effect of dragging on an angle is that the user will navigate while changing both
location and orientation together (see for example Figure 6.10). This is the
integrated movement similar to how we function in real world. But it seemed that
with the integrated movement, it was more difficult to solve the two navigation
problems.

![Diagram](image)

**Figure 6.10** Dragging to navigate in Walk mode

Navigation speed was too fast at times as the participants dragged a long line.
Arrow keys on the keyboard could be of better control for navigation speed. Shift
and Ctrl keys were rarely utilised to control navigation speed. Alt key for sliding and
Space key for zooming were utilised often. Bonbon seemed to prefer changing
navigation mode than using meta-keys.

When the participants were navigating, the researcher asked them in an
ordinary voice: is the turtle moving? Surprisingly, all participants answered in a firm
voice and said "No". They consciously knew that they were navigating and the turtle
was not moving.
At the end of this activity, School C participants came up with an idea. Instead of navigating to discover the secret and legend of the turtle, they wanted to stay unmoved and command the turtle to move by using Quick Command. This is a type of problem solving strategy termed "reverse thinking" (Baturu & Cooper, 1993). This was unexpected and was developed as an additional task. The researcher then posted a mesage in the forum titled “re: other ways to make the turtle's eyes change colour”:

Hi all,

What a good finding by grae, bonbon, and rosco...

We have been trying different ways (e.g., walk, fly, and examine) to approach the turtle's eyes so they can change colour.

A total different way of doing that is to move the turtle instead of moving yourself in the 3D virtual space....

Remember that we have a Quick Command tool?

Can you restore viewpoint, and do not move yourself, just try to move the turtle by giving commands from the Quick Command tool. If you can move the turtle to just in front of you (within 0.5 meter) then its eyes should also change colour....

Because we are using commands, this post is also relevant to the Programming forum, you can also see this post in the Programming forum.

Have fun! 😄

_________________

Andy *^___^*

and another message titled “How about move the turtle for jet.....”:

Hi all,

Same as move the turtle to see its eyes change colour..

I have some info about you in the 3D space...

When you restore viewpoint, you are 1.6 meter above the ground, six meters away from the turtle.

The trick is if you are 10 meters away from the turtle and face to the turtle's bottom, then the turtle should turn into a jet too.

PS: this post is relevant to Programming forum, you can also see this post in Programming forum..

Cheers, 😊

_________________

Andy *^___^*

Victor posted in programming forum:

type
up 1.6
rd 90
south 3
Another contingency that related to social interaction was found in this activity. Alekat20 found how to setup a poll in forum without the researcher’s help. She setup the first poll (see Figure 6.11) on the 3D navigation modes:

![Figure 6.11 A poll in VRMath Forum about 3D navigation](image)

After this first poll, many other polls were setup by all participants except Grae during other learning activities.

### 6.3.2 Discussion

This activity focused on developing the users’ 3D navigation skills in the 3D VR space and evaluating the design of the interface of 3D VR space. The 3D navigation in VRMath using mouse and keyboard was mastered by these Year 4-5 participants after two hours of engagement (the actual time practiced in VR space was about one hour). Some characteristics about 3D navigation similar to Iteration 1 were found. These included the extensive use of navigation tool bar (i.e., Restore Viewpoint, and Fix Screen etc.), and some difficulties in controlling navigation speed and direction (i.e., mouse dragging). New characteristics were also identified.

The use of navigation tool bar to help 3D navigation was again confirmed in this Iteration. However, some functions were actually causing confusion in this activity. For example, the Set Rotation Center resulted in the participants being unable to properly navigate in Examine mode. In this activity, it seemed that the Navigation Mode, Avatar View, Restore Viewpoint, Fit Screen, and Compass would be required. The Set Rotation Center and other icons (functions) were not needed. However, it was natural that the participants would try every one of them especially when they were free to explore in VRMath. Each of the icons (functions) was only briefly explained. It was evident that the researcher's brief introduction about icons was not retained in participants' memory except those that were used in activities. These icons (functions) were designed for certain purposes. It was thought that
specific activities for utilising specific icons should be designed and made clear for future iterations.

With respect to the difficulties in navigation speed and direction, a critical navigational behaviour was observed. That was, when navigating using the mouse, the participants dragged between forward and right, causing the users to change location and orientation simultaneously. This type of navigation is similar to the navigation in real world and is termed as integrated movement. In virtual space, however, the integrated movement appeared to be a source of difficulty. It was conjectured that due to the lack of kinaesthetic feedback and limited field of view, the users couldn't get a good sense of their location in 3D virtual space.

To solve the integrated movement problem in navigation, it was suggested the participants to use meta-keys (e.g., ALT for slide) and drag the mouse on vertical and horizontal lines. Thus the navigation became component movement, in which the users only changed orientation (direction) or position (location). This made the navigation easier when users tried to approach the turtle's eyes or navigate beneath the turtle. The integrated movement and component movement issue identified here is different but relevant to PE7. The PE7 stated in Section 2.5.2 is about user's differentiation of integrated movement and component movement upon the manipulation of the turtle when constructing 3D geometrical world, while the issue identified here is about the effect of integrated movement and component movement upon 3D navigation. The former focuses on the turtle's movement for constructing 3D objects and the later focuses on user's movement for getting vantage viewpoints. This new issue about integrated and component movement is included in the reflection and redesign chapter of the study (see Section 10.2: Reflection).

Overall, it was observed that the participants in Iteration 2 had developed much better 3D navigation skills than the participants in Iteration 1. The researcher conjectured this being due to:

1. The participants in Iteration 2 having some prior experiences in 3D graphics such as 3D PC and video games, and a little longer exposure to 3D navigation in VRMath’s 3D virtual space.

2. The researcher providing better instructions and scaffolds in Iteration 2 such as discussion about the differences between each navigation mode,
linking navigation modes to participants’ real life experience, and practice in one navigation mode only to solve the secret and legend problems.

3. There were more social interactions among six participants within and between groups in Iteration 2 than that of the two participants in Iteration 1.

With respect to the second cause above, the researcher intentionally used instructions and languages such as "if you move to" and "you navigate to" to inform users that they were changing their orientations instead of manipulating the object (i.e., the turtle) when navigating. The effect of this was that the participants were conscious about they were moving themselves rather than moving the turtle. This was evident when the researcher questioned "is the turtle moving?" during their navigation, and all participants replied "no" immediately and confidently. In terms of PE1, the participants successfully distinguished and developed both their spatial visualisation (i.e., moving the object) and orientation (i.e., moving themselves) abilities (McGee, 1979).

A subsequent effect of this consciousness about spatial visualisation and orientation was conjectured to emerge. That is, the idea of commanding the turtle to move rather than navigate to solve the secret and legend problems, as proposed by School C participants. Since we could navigate close to see the turtle changing its eyes' colour, naturally we should be able to command the turtle to move close to us and get the same result. The experiments by the participants proved that it worked. This was a valuable reverse thinking or working backward problem solving.

It was obvious that there were more social interactions between and within groups in this Iteration, especially through the mediation of the Forum interface. An opinion poll could be an important social interaction in VRMath Forum. The researcher didn’t plan to utilise it because he tried to minimise the complexity of the design experiments. However, it occurred naturally. The participants were given time (for about 15 minutes) in every session to freely interact with VRMath Forum. It was found that they fluently interacted with the three interfaces (VR, Programming, and Forum). 3D VR space and the programming interface (e.g., direct command or Quick Command etc.) was the fun place to play. However, it was found that the participants also spent a great deal of their time in the forum, telling their findings, socialising, and showing their abilities. It was noted that a total of 14 polls
were created during this iteration. These polls played a crucial role in facilitating interaction and discussion among participants. They were also a means for participants to show off themselves. The fact that the participants learnt how to setup polls without any instruction also indicated that the forum interface or environment was easy to use. However, in the future, the use of polls need more scaffolds in order to maximise their benefits. For example, in the poll generated by the participants shown in Figure 6.11, the two options are not polar opposites. Therefore, users of this poll were not provided with a legitimate choice when asked for their preference.

6.4 LEARNING ACTIVITY 3: TURTLE DANCE

Following Learning Activity 1 (Become a member) that focused on Forum interface and Learning Activity 2 (The secret and legend of the turtle) that focused on VR interface, Learning Activity 3 focused on the programming interface and formally introduced VRMath's Logo-like language about moving and turning commands to the students. There are three sets of moving commands in VRMath: egocentric, fixed and coordinate system (see Section 3.5.1.1). The coordinate moving commands involve the 3D coordinate system and variables X, Y, and Z, which were deemed to be too complicated for the participants. Therefore, only egocentric (i.e., FORWARD and BACK) and fixed (i.e., EAST, WEST, NORTH, SOUTH, UP, and DOWN) moving commands were introduced in this activity. Although the coordinate system is not introduced in this activity, the turning commands are actually based on the X, Y, and Z axes.

In VRMath's 3D space, there are six turning commands LEFT (LT), RIGHT (RT), ROLLUP (RU), ROLLDOWN (RD), TILTLEFT (TL), and TILTRIGHT (TR) (please refer to Section 3.5.1.2 for details). In addition to these moving and turning commands, the commands REPEAT and HOME are essential and were introduced in this activity.

6.4.1 Results

The researcher began with a discussion about the six turning commands with School C participants. He first asked the participants to use their hands to simulate
the turtle (one hand on top of another), and then spoke the commands to link with hand gestures:

*Researcher:* See this is the turtle. The turtle can turn right (hands turn to right), left (hands turn to left), roll up, how to roll up? (hands' finger top rolls up), and roll down (with hands' finger top rolls down). What others?

*Rosco:* (thought for few seconds) tilt (he tilted his hands).

*Researcher:* Good. Show me tilt left (all performed tilt left with hands), good, and tilt right (all performed tilt right with hands).

The introduction of the six turns went smoothly with the help of hand gestures. The researcher then challenged the participants with their 3D thinking. He asked, "There are only six turns. Do you think the six turns can turn to any directions?" and pointed to some directions in space. The participants thought and moved their hands (using the turtle gesture). Rosco turned his hands in a manner reflecting how the VRMath turtle turns within the VRMath virtual environment. For example, he seemed to make only two turns to get to any directions the researcher pointed to. After that the participants replied "yes", the researcher reinforced this by saying "yes, the turtle probably can go anywhere".

To formally introduce the turning commands, the researcher then asked the participants to use the Quick Command tool. Bonbon and Grae were watching Rosco operating the computer. Rosco quickly brought out the Quick Command tool from the Tool menu (see Figure 3.11). When the Quick Command tool appeared in a separate window, he immediately dragged the Quick Command window away from the 3D space and clicked in the 3D space. The researcher observed that it seemed to be Rosco's computer using habit, in which he liked to move or resize windows to get a better working environment on screen. However, when he clicked into the 3D space (VRMath window), the Quick Command window was sent to the back and disappeared. The participants were a little bit puzzled but Rosco quickly found the Quick Command window again by clicking from the Tool menu.

The researcher then introduced the commands to the participants by moving the mouse pointer over the pictorial icons (see Figure 6.12). When the mouse is over the icons, the associated commands will appear in the text field down the bottom. The researcher encouraged the participants to click and try each icon, read the command, and see the action of the turtle in the 3D virtual space. It was hoped that this would facilitate the participants gaining knowledge about the commands.
The participants quickly used the mouse clicking-on icons and verified the movements of the turtle in 3D space. They tried all fourteen icons including the six turning and eight moving commands. The HOME command was introduced here to bring back the turtle to its starting position when the participants had commanded the turtle to move far away. To help the participants better understand about turning and moving in VRMath, the researcher asked two key questions as illustrated in the following transcription.

Researcher: If you turn the turtle, does the turtle change its position?
Rosco: urh... (He tried in VRMath) no, it doesn't move away from its position.
Researcher: but if you command the turtle to go forward, does the turtle change its direction?
Rosco: Yes, oh no no, ... (He tried in VRMath) no, if you do this (click on turning icons)... it changes its direction.

The researcher observed that the participants focused their attention more between the icons and the actions, and the commands seemed to be ignored. Therefore, the researcher asked the participants to type commands in the command text field instead of using Quick Command. The use of Up arrow key to recall previous commands was introduced here and was often used by the participants. For example, the researcher asked Rosco (Three participants were together, Rosco was operating the computer) to type in a command _tl 15_ and press Enter key to make the turtle tilt left 15 degrees. The participants then used Up key to recall _tl 15_ command and Enter key to execute the command. By quickly pressing the Up key and then Enter key, the turtle was repeating the turning quickly as was the mouse clicking on the icon in Quick Command.

After some practice on the six turns, the researcher introduced the REPEAT command. The REPEAT command is actually the command to make the turtle dance without having to click the icons or press the keys many times. When the researcher
mentioned that there was a command called REPEAT, the three participants immediately turned to their computers and couldn't wait to try the command. However, before they tried, the researcher asked the following questions.

**Researcher:** Tell me what is REPEAT?
**Bonbon:** Repeat means …
**Rosco:** Repeat every command …
**Researcher:** Tell me if you want to repeat, what do you need to tell the turtle?
**Rosco:** Repeat and then the commands…
**Researcher:** Yes, sure, but how many information I have to give the turtle? If you just say repeat and enter, what will the turtle say?
**Rosco and Bonbon:** I don't know how to repeat. (This is the standard feedback in Logo language when an unknown command was entered. However, since repeat is a known command in Logo, the feedback should be "Not enough inputs". Their answer indicated that they had tried some of their own commands before.)
**Researcher:** That's right, or probably some other answer…
**Rosco:** wait wait wait… (He turned back to his computer to try). Not enough inputs.
**Researcher:** Ok, the turtle needs more inputs after REPEAT.
(Rosco typed in DANCE)
**Rosco:** I don't know how to dance.
**Researcher:** Yes, you didn't teach the turtle how to dance. Now, REPEAT, first thing you have to tell the turtle is how many times you want to repeat, and then you have to tell the turtle what to repeat, a command list in square brackets.

After the discussion about the semantics of REPEAT command, the syntax of REPEAT command was formed as `REPEAT repeat_count [command list]`. The participants then tried typing in commands such as `repeat 12 [ru 30]` to see the turtle "roll up" 30 degrees for 12 times. They were also encouraged to substitute the RU with other turning and/or moving commands and change the repeat count and degrees. But before they sent in their commands, they were asked to predict how the turtle would act. The participants were then having fun making the turtle dance. The command SETWAIT was introduced here to control the turtle's moving and turning speed. For better observing the turtle's movement, the participants were asked to set 300 to 500 mini-second delay (e.g., `setwait 350`) as the default setting was 20 mini seconds. However, it was noted that Rosco changed the delay to zero (no delay) and said to Bonbon excitedly when he turned the turtle "see how fast it goes".

It was noted that commands such as `repeat 12 [tl 30]`, `repeat 360 [tl 30]`, `repeat 10 [cone]`, `repeat 12 [tl 30 cone]`, `repeat 360 [tl 100]`, and `repeat 360 [tl 309]` etc. were created by School C participants. In particular, when the command `repeat`
12 \[\text{tl 30 cone}\] was entered, the participants "Wowed" and said "I created a monster". And when the cone was replaced with box, Rosco said that he created a "gear" (see Figure 6.13).

![repeat 12 [tl 30 cone]  repeat 12 [tl 30 box]](image)

**Figure 6.13** Objects created by repeat command

School K participants went through similar processes about turtle dance activity as School C participants. Alekat20 was the first to know that this session was about turtle dance because she had sent the researcher a private message asking about next session and the researcher had replied. When she checked her private message, she spoke out in a surprising voice "What? We are going to teach the turtle dance? It's kind of funny". It was observed that at the beginning of sessions, participants were busy communicating using private messages and forums. VRMath's social-actional interface had become a facility for the participants to easily communicate with each other.

The two key questions about turning and moving were posed to School K participants but they were not bothered at all. They were very conscious about that turn does not change position and move does not change direction. The researcher tried to further clarify this component movement characteristic in VRMath by asking Victor to act out walking to a computer nearby, in which he naturally turned and moved seamlessly. The researcher then reminded them that in fact, the movement in 3D space consisted of two separate components turn: changing direction, and move: changing position.

When using the turning commands, the researcher didn't suggest specific degrees to turn. R2D2 tried \textit{rt 360} but didn’t see any movement of the turtle. He then found out that the turtle had actually turned around a full circle but because it turned so quickly therefore it was difficult to notice. Victor always had different ideas. He played and tried turning the turtle one degree each time. It seemed that he wanted to
test the turtle if it could do such a small turn. However, when he tried the moving command such as FORWARD or FD, he typed `forward 1000000000000` to test the turtle. As a result, the turtle disappeared because it had moved forward a long way.

The researcher noted that School K participants were confused with the command TL (Tilt Left) and TR (Tilt Right). They thought that TL stood for turn left and TR stood for turn right. The researcher again asked them to use Quick Command to review the commands and actions. When using Quick Command, R2D2 quickly turned the turtle into a Jet plane (For details on how to turn the turtle into a jet plane please refer to Section 6.3: Learning Activity 2), which they called "Jet dance". They tried all commands from Quick Command tool. They seemed to see the difference between egocentric and fixed frame of reference commands. An episode about introducing METER and CM commands occurred here. The default unit of distance is meter in VRMath. CM command will change the unit from metre to centimetre. Before the researcher introduced the project management in VRMath, Victor had found how to open existing projects. He opened and executed a symmetrical tree project, in which the unit of distance was changed from metre to centimetre. Therefore, when Victor tried to make the turtle go forward (`FD 1`), the turtle only went a centimetre forward; and it was difficult to notice its movement. By this chance, the researcher introduced the METER and CM commands. Later on, the researcher found that Alekat20 was using a command METRE, which couldn't be recognised by the turtle because of the Australian spelling. Hand gestures were also utilised to scaffold the link between commands and actions. The researcher tested their understanding by asking them to perform a series of commands (e.g., `rt 30 ru 45 east 1 up 1 tl 30` etc.) using their hand gestures. It was observed that the three participants were following the instruction correctly.

The REPEAT command was introduced in the way similar to School C. Victor was particularly curious about repeat. He tried `repeat 720` (incomplete command), `repeat 1000 [cone up 0.5]` (create 1000 cones), and `repeat 1000000 [cone up 0.5]` etc. Because the repeat count was too large, the interrupt function was introduced here to stop the repeat.

The researcher then challenged the three participants with the commands `repeat 10 [tl 30 repeat 5 [ru 30]]`. When seeing this, Victor was excited and couldn't
wait to try but was stopped by the researcher. The researcher asked for their interpretation about that repeat within repeat commands.

*Alekat20: It will turn left 30 degrees, 10 times…*
*Researcher: 10 times of ..?*
*Alekat20: And then it will go right up … what is ru mean again?*
*Researcher: Roll up.*
*Alekat20: yes, and then it will roll up 30 times.*

Alekat20's interpretation was obviously incorrect. She showed a misunderstanding about the abbreviation of commands, and the confusion about degrees and repeat count. The researcher then explained and showed them step by step (*setwait 1000*) in the 3D space. They realised the commands after the researcher’s explanation and seeing the actual movement of the turtle, which they exclaimed "cool". However, they didn’t try more of this repeat within repeat commands due to the session ending.

6.4.2 Discussion

This learning activity "turtle dance" was designed to formally introduce moving and turning commands. As a result, many relevant commands such as HOME, SETWAIT, METER and CM etc., and GUIs such as Quick Command, Navigation Toolbar were learnt and utilised. From the participants' complex and dynamic interactions within VRMath environment, the following implications emerged.

Firstly, with respect to perspectives for evaluation, many operations of the participants' spatial abilities (PE1) were observed. For example, when the participants were commanding the turtle to move and turn, they were able to predict (create mental images) the turtle's movements. When manipulating the turtle (object) it was the spatial visualisation ability (McGee, 1979) in action. It was found that the participants also navigated to view the turtle from different viewpoints (spatial orientation) while using Quick Command or typing commands to manipulate the turtle. However, when doing this the Quick Command tool was sent to back and disappeared. This usability issue was identified in Iteration 1 but not rectified due to limitation in Java programming. It was noted that this usability issue still bothered the participants in Iteration 2.
In terms of spatial perception abilities (Hoffer, 1977), the VRMath environment and turtle dance activity also provided opportunities to operationalise these abilities. For example, when navigating to see the turtle from different viewpoints and/or command the turtle to turn a different orientation, the participants would have perceived the perceptual constancy, in which they recognised the size and shape of the turtle being unchanged in spite of the variation of its impression as seen from different viewpoints. When the participants had commanded the turtle to move outside of view, it was observed that they could easily follow and navigate to see where the turtle was. This is related to the position in space perception, perception of spatial relationships, and visual memory of Hoffer's (1977) spatial perception abilities. It was evident that the participants tried to make sense of the turtle's location and direction when they had commanded the turtle to move and turn. For example, Victor tried to catch the turtle when he sent command forward 1000. He navigated forward, which was the correct direction to look for the turtle. But eventually he used command HOME to bring back the turtle because it was too far to navigate. In addition, the Restore Viewpoint was utilised to bring back the initial viewpoint. It was thought that activities such as "search for the turtle" (e.g., after commands right 60 forward 100 was sent) could be designed to facilitate the development of position in space perception, perception of spatial relationships, and visual memory of Hoffer's (1977) spatial perception abilities.

With respect to PE2, it was found that the participants could easily interact with the three interfaces: VR (topological semiotics), programming interface (typological semiotics), and hypermedia forum interface (social-actional semiotics). The turtle dance activity was the first activity to observe the participants' holistic use of the three interfaces as all the three interfaces were formally introduced through the first three learning activities. The VR interface provided continuous visualisations (topological transformation) when users navigated to see the turtle or geometrical objects. This is an example of what Hoyles (1996) refers to as the gap bridging power of computers that facilitates the participants' ability to differentiate between a static drawing and a dynamic figure. However, the manipulation of turtle using programming commands is not quite continuous. For example, the command \texttt{tl 30} resulting the turtle to tilt left 30 degrees seemed observable. But the command given by R2D2 \texttt{tl 360} didn't make sense to him because the turtle turned around too quickly.
and didn't show the direction it turned. It was thought that maybe a slower animation for turning some degrees or moving a distance should be designed on the turtle in 3D virtual space.

It was evident that the participants' engagement in body movement could facilitate their understanding of 3D movement in VRMath. For example, in using hand gestures to simulate the turtle, the participants could then act and think of the 3D movements and rotations naturally and thus identified the six turns on three axes. This confirmed what had been previously noted in the research literature about the promotion of the use of kinaesthetic feedback or body movement to aid the development of mathematical thinking (see for example, L. D. Edwards, 1995; Lemke, 2001). It was felt that the participants tended to use right and left more or prior to the rolling and tilting commands. For example, from their hand gestures, they took turns to get to any direction; and the first turn was always left and right. Alekat20 also interpreted that TL means "turn left", and RU means "right up". It was speculated that because inherently we live on a 2D plane, most of time we only turn left and right; and we do less tilt and roll in real life.

Certainly, it can be argued that the participants only learnt these commands in a short period of time, and more time should be devoted to master these commands. But clearly, there were confusions about the abbreviation of commands. This reflects the semiotic principle that a sign signifies something but it also signifies something else (Cunningham, 1992b). The design of these commands was discussed in Section 3.5.1, in which the naming of commands was carefully considered. It was thought that since the abbreviation could cause confusion, then maybe the use of full commands such as RollUp and TiltLeft should be practiced together. In addition, the spelling difference between Australia and American English such as METRE and METER was found to be a usability issue. The social-actional resources such as the forum and private message were found to be highly utilised by the participants. Alekat20 again setup a poll for this turtle dance activity (see Figure 6.14). However, the usages of these resources so far tended to operate on the socialising and "showing off" levels. It was expected that collaboration and negotiation will occur in later activities when they engage in the construction of 3D geometrical objects.
Figure 6.14 A poll for turtle dance activity

With respect to PE3, the egocentric and fixed frames of reference were explored by the participants. However, due to the time constraint, the researcher only focused on the turning commands. Although the participants seemed to see the actions of egocentric (i.e., FORWARD, BACK) and fixed (i.e., EAST, WEST, NORTH, SOUTH, UP, and DOWN) movement of the turtle, key ideas such as egocentric movements depend on turtle's orientation while fixed movements don't, was not investigated at this time by the researcher.

The PE7 about component movement was manifested in this activity when the participants were using commands. In real life, often we turn and move together and thus ignore the component movement. However, the thinking and awareness of component movement are important factors for the study of rigorous and analytic geometry. It is the nature of the "Turtle Geometry" that places the users into the thinking of component movements. For example, Rosco had to take two turns using his hand gestures to get to any arbitrary directions. In VRMath, we can only command the turtle to either change direction or location each time. This was made clear to the participants when the researcher questioned "if you turn the turtle, does the turtle change its position? And if you command the turtle to go forward, does the turtle change its direction?"

One important command introduced in the turtle dance learning activity was the REPEAT command. It is a more complex command because it has two inputs or parameters. Some syntactic guesses such as repeat, repeat 12, 10 repeat [box fd 1 up 0.5] were observed from the participants. These examples of incorrect syntax showed a lack of understanding and thinking about the meaning of repeat. After the discussion about the meaning of repeat, the participants could make sense about repeat and thus form the correct syntax for the REPEAT command. The processes of learning programming commands here were semantical first then syntactical, which differed from the view that syntactic knowledge should be taught prior to semantic
knowledge (Fay & Mayer, 1988). It could be argued that the participants could also master the programming commands if syntactic knowledge was taught first. However, the syntax is just about the rule and structure, and doesn't carry any mathematical concepts or meanings. It was felt that the understanding about the meaning of commands had facilitated the participants' use of commands. Furthermore, in VRMath's Logo-like language, users can create new commands (via procedure), in which they must have semantic understanding to design their own syntax. Therefore, this researcher would prefer to have the semantical understanding of the language (programming commands) ready within the participants then naturally emerge into the syntactical understanding.

It was observed that using REPEAT could easily create a pattern of 3D objects or movements. This was also the "WOW" factor that "sparkled" the participants. After this activity, the researcher reflected and realised that the use of programming commands (non-figural data) to create 3D objects (visual images) was actually the visualization ability VP (visual processing of information) identified by Bishop (1983). For example, Rosco used commands repeat 12 [tl 30 box] to create a gear (see Figure 6.13). Although he created the gear by accident, this kind of activity seemed to help the development of his VP ability. Moreover, if the participants could interpret from the gear (visual image) to form the commands, their interpretation of figural information (IFI, see Bishop, 1983) ability would have been developed. For example, the researcher can ask the participants to interpret the composition of the gear in Figure 6.13. The participants may identify that in fact three boxes are enough to create the gear. And thus commands such as repeat 3 [tl 30 box] may be formed to represent the 3D gear. For experienced users, real world objects such as a fern leaf, a flower, and stair cases etc. could be given for interpreting into more complex commands and vice versa.

Lastly, the repeat within repeat commands (e.g., repeat 10 [tl 30 repeat 5 [ru 30]]) seemed to surprise the participants. For an experienced programmer, it would be common to use a loop within a loop. But for a novice programmer, this repeat within repeat usage may be an inspiring point to encourage creativity and flexibility in programming. It was thought that a "Cool Dance" activity that uses the repeat within repeat syntax could be designed as an extention of the turtle dance activity for the future iterations.
6.5 SUMMARY

This chapter reported on the enactment and evaluation of the introductory type of learning activities, which included the first three of the nine learning activities.

Learning Activity 1: Become a member focused on the hypermedia and forum interface (social-actional resources) of the VRMath system. The participants showed excitement to communicate with other participants via the forum discussion and private messages. They also showed their good ability to use the forum from registering and login, to posting messages and using emoticons and avatar images. However, these young participants didn’t have a proper email address to fully utilise the forum facilities due to security concerns from parents.

Learning Activity 2: The secret and legend of the turtle focused on the VR interface (topological resources) of the VRMath system. The participants first had some difficulties in controlling 3D navigation speed and direction, but they gradually mastered this in about one hour of practice. The 3D navigation to solve the two problems was aided by the use of Navigation Toolbar. Due to the lack of kinaesthetic feedback of the integrated 3D navigation, the participants couldn’t get a good sense of their location in 3D virtual space. However, with the aid of meta-keys, the participants were in the end able to navigate to solve the problems more easily (e.g., ALT key for sliding to change location but keep direction). It was found that confusion between spatial visualisation (moving objects) and spatial orientation (moving self) could be avoided with instructional scaffolds.

Learning Activity 3: Turtle dance focused on the programming interface (typological resources) of the VRMath system. It was found that the use of Quick Command GUI could aid the learning of turning and moving commands. However, there was some confusion about the naming of commands such as METRE for METER (Australian spelling vs. American spelling), and RU (Roll Up) for Right Up. The turtle dance activity enabled the investigation and development of both spatial visualisation (move or turn the turtle) and orientation (navigate to see from different viewpoints) abilities (McGee, 1979), as well as some of Hoffer’s (1977) spatial perception abilities.
7.1 OVERVIEW

Iteration 2 was an extensive enactment and evaluation of this design-based research. There were nine learning activities in Iteration 2, which were classified into three categories: introductory, specific concepts, and application types of learning activity. Chapter 7 reports on the enactment and evaluation of the specific concepts type of learning activities. This type of learning activity focused on developing some specific mathematical and geometrical concepts. Learning Activities 4-8 were of this type.

Learning Activity 4: Shape and scale introduced the five 3D primitive objects commands (BOX, CAN, BALL, CONE, and LABEL) and a set of scaling commands to investigate the concepts of projective shapes, similarities, and transformations.

Learning Activity 5: Climb up stairs utilised previously learnt commands to build a staircase. This developed the idea of moving and turning the 3D turtle as a reference point in 3D virtual space for placing 3D objects. The participants were also able to experience the virtual walk through of climbing up stairs.

Learning Activity 6: 3D rotation developed the concept of non-commutativity of 3D rotations. The participants needed to utilise the VRMath environment to investigate whether the two sets of 3D rotation (RIGHT 45 ROLLUP 45 and ROLLUP 45 RIGHT 45) would end up with the same direction or not.

Learning Activity 7: Formula of polygon investigated a mathematical formula for 2D regular polygons. The target formula to develop was “REPEAT :side [FORWARD 1 RIGHT 360/:side]”.

Learning Activity 8: A frame of a cube investigated the properties of a 3D cube. The participants first used Quick Command to construct a cube with immediate visual feedback. Later they wrote necessary commands into a Logo procedure to create a cube.
7.2 LEARNING ACTIVITY 4: SHAPES AND SCALE

This activity introduced SCALE command and five 3D primitive objects BOX (CUBE), CAN (CYLINDER), BALL (SPHERE), CONE, and LABEL. The five 3D primitive objects along with 2D line and face are the fundamental building blocks for constructing 3D virtual worlds. The SCALE command involves the scale in three dimensions X, Y, and Z, and thus it has three inputs or parameters for the three dimensions. Simple scaling commands SCALEX, SCALEY, and SCALEZ were designed to take one input only for each dimension. Alternatively, SCALEW (SCALEWIDTH), SCALEH (SCALEHEIGHT), and SCALLED (SCALEDEPTH) were also designed for young children who haven’t learnt about 3D coordinates (for more details please refer to Section 3.5.1.5). These scaling commands enabled the investigation about similarities and transformations, and the construction of rich 3D virtual worlds using scaled 3D primitive objects.

7.2.1 Results

Before this activity, all five 3D primitive objects except LABEL had been learnt and used by all participants. The participants from both schools told the researcher that they found the four primitive objects BOX, CAN, CONE, and BALL from the image in VRMath homepage. From the commands history log, it was found that the participants had also guessed some other primitives such as pyramid, triangle, prism, and triangular prism etc. Because those were not built-in commands, they got replies such as "I don't know how to pyramid" from the programming console.

The researcher then formally introduced the five primitives to the participants. This included the synonyms of BOX for CUBE, BALL for SPHERE, and CAN for CYLINDER; Font Chooser (see Figure 3.14) for LABEL object; and Material Editor (see Figure 3.15) for changing colour and material of objects.

R2D2 first tried the commands CYLINDER, CONE, and SPHERE together and created overlapping objects. He didn't want to clear the screen so he asked the researcher how to undo it. The commands UNDO and REDO, and the undo and redo functions in the Edit menu (see Figure 3.8) were then introduced. The researcher reminded them that they could first move or turn the turtle to the place and direction they wanted the objects to be, then use the primitive commands. The use of Material
Editor (see Figure 3.15) to change the 3D colouring before creating objects was also recommended. Then the researcher asked the participants to create a virtual world similar to the one they saw in the VRMath homepage. After some practices, the participants built their first 3D virtual world similar to as seen in Figure 7.1.

![Five primitive objects in VRMath](image)

*Figure 7.1 Five primitive objects in VRMath*

It was observed that all participants spent much time on choosing materials for objects. As this was the first time they utilised the Material Editor, the researcher didn't interrupt their exploration on the tool. After their extensive use of the Material Editor, School K participants found a material mismatch (a bug) in the Sheen Colors set of materials. Bonbon set up a poll in the Forum for Material Editor and found that Winter Colors to be their favourite colour (see Figure 7.2).

![Poll for Material Editor](image)

*Figure 7.2 A poll for Material Editor*

The construction of the virtual world in Figure 7.1 was quite awkward for the participants as they had to switch among the command input, material editor, and the 3D virtual space. Some frustration was observed when the participants clicked into the 3D VR space and lost the Material Editor (receded behind the main window), or when they typed the commands into the 3D VR space (the command input text field was not focused). However, the participants soon adjusted and successfully built the first virtual world. When Alekat20 created labels using `label [i love VRMath]` and `label [hi I made this thing me alekat20]`, she excitedly called out, "Cool, look what I
have got on the screen". After completion, the participants then enjoyably navigated to view their virtual worlds from different angles.

Scaling commands were then introduced. When the command SCALE was mentioned, Alekat20 tried scale and scale 2, and got "not enough inputs" feedback. The researcher then explained that the SCALE command needs three inputs as width, height, and depth. And alternatively, the researcher encouraged the participants to use SCALEWIDTH (or SCALEW), SCALEHEIGHT (or SCALEH), and SCALEDEPTH (or SCALED) for individual dimensions. For School K participants, the researcher also mentioned about SCALEX, SCALEY, and SCALEZ as equivalent to SCALEW, SCALEH, and SCALED. However, it was found that they preferred using the W, H, and D commands. Perhaps it was because they were more familiar with the terms width, height, and depth.

The researcher instructed School K participants to (a) create a ball, (b) change the scale to scale 2 0 2, then (c) create another ball. Alekat20 replied in a surprising voice after seeing what had been created. She said "It's a planet" (see Figure 7.3).

![Figure 7.3 Planet Saturn](image)

This utilisation of scale seemed to inspire School K participants. They navigated to examine the planet. R2D2 wanted to do this again, but he kept creating a disc with the BALL command. He asked "How can I make the scale back to normal?" The researcher replied and instructed them to use scale 1 1 1 to get back to the "normal" scale. Alekat20 asked whether she could save the 3D scene if she created some cool stuff. The researcher answered "Yes" but he knew that to "save" the 3D scene in VRMath involved the project management (save the programming codes) or export to a VRML file. Those were rather complicated and therefore the researcher didn't explain it to her. The three participants were then excitedly busy making their 3D virtual worlds.
For School C participants, the researcher challenged them with the following questions.

Researcher: *Now I have a ball* (two hands make a ball shape), *but if the height of the ball is 0* (hands clap together from top and bottom), *the ball will become…*
Grae: It will be flat.
Rosco: *The ball will turn into a circle.*
Researcher: *Good. How about a cone, and the height is 0* (hand gestures too)?
Bonbon: Circle.
Researcher: *How about a cylinder?*
All: Circle.
Researcher: *How about a cube?*
All: That will be a square.
Researcher: *Excellent! Now the height is 1, stay the same* (hand gesture), *width the same* (hand gesture), *depth, shu…* (hands clap together for depth), 0, *tell me what is a ball with depth 0?*
(All thought for few seconds)
Rosco: Circle.
Researcher: *How about a cone with depth 0?*
Rosco: Triangle.
Researcher: *How about a cylinder?*
Grae and Rosco: Circle.
Bonbon: Rectangle.
Researcher: *Ah.. rectangle. (Bonbon smiles), how about a cube?*
All: Square.

The three participants from School C then tried *depth 0* with the four primitives in VRMath (see Figure 7.4), and were satisfied with the results.

| scaled 0 box | scaled 0 cone | scaled 0 can | scaled 0 ball |

*Figure 7.4 Scaled primitive objects in VRMath*

All participants now seemed to have some ideas to try. The researcher then allowed some time for them to try and asked them to contribute to the Forum with their findings. Grae first posted in the Programming Forum titled "How to change the scale of objects...". However, his post didn't yield any response.

Hi all,

I learnt three commands:
scalew -- means change the scale of width
scaleh -- means the height changes
scales -- means the step will change
for example,
Victor posted his virtual world in Programming Forum titled "tower". The result of this tower program is presented in Figure 7.5 below.

```plaintext
scaleh 0 cone
you will get a circle
how about scaled 0 scaleh 1 cone ???

Figure 7.5 A tower made by scaled primitive objects
```

The researcher post replied to Victor and questioned "Hi victor, I just wonder why you want to scale width 3 times and depth 2 times... if I switch the scale of width and depth, will your tower still look the same...". This inquiry was intended to develop his ideas about similarity, congruency, and transformation. Unfortunately, Victor didn't reply to the question.

Bonbon created a candle using the commands learnt in this activity. She posted, “I can make a candle” in Programming Forum. The artefact of the 3D candle is presented in Figure 7.6.

```plaintext
~Hi all
To make a candle you choose the colour from material editor and then put in this command: scaleh 7 scalew 0.9 scaled 0.5 cylinder
Then choose a yellow, red or orange colour then use this command: up 4
after that use this command: scalew 0.5 scaleh 5 ball
You should have a perfect candle!~

Figure 7.6 A candle made by scaled primitive objects
```
Alekat20 tried Bonbon's commands by copying her codes from the Forum and pasting into the programming console. She replied to Bonbon in the Forum:

You are very smart bonbon!!!!
can you make any other COOL things!!!!

_________________
Can You Talk To Me?????????? Please!!!!!
Can You Make the Turtles Body A Different Colour??????????

The researcher also responded to Bonbon and tried to encourage a discussion about scale and distance by posting:

Hi,
I am thinking that ....
Why do you up 4?
Does that have anything to do with the scale?

_________________
Andy *^___^*

Again, Bonbon did not respond to the researcher's inquiry in the forum.

7.2.2 Discussion

This "shapes and scale" learning activity was the first activity in which the participants started to construct 3D objects. The five 3D primitive objects were simple and easy to use for these Year 4 and 5 participants. However, the young participants found the scaling of these primitives quite complicated. 3D scaling involves a factor in each of the three dimensions with 1 being the original or normal scale, and 0 being the exclusion of the corresponding dimension. Scaling also involves an operation of multiplication of both integers and decimal numbers. Although the researcher was concerned about the participants' understanding of the concepts of scale, ratio, and decimal numbers, he still encouraged the participants to try the commands and see what was happening in the 3D virtual space. The results of the participants' use of the commands turned out to be very creative.

So far, the participants had used the VRMath system for about five hours. It was observed that the six participants could all fluently interact within the three interfaces particularly within this activity. They knew when to resize the Forum and VR frames for browsing in the Forum and navigating in the VR space. They didn't miss typing of commands into the command text field after they had navigated in 3D space. This indicated an easy-to-use aspect of usability, and that approximately five
hours were required for these young children to master the operation of VRMath system.

The use of the three interfaces also indicated a progressive utilisation of the three semiotic resources (i.e., VR: topological, programming language: typological, and forum: social-actional) for constructing knowledge about 3D geometry. For example, to build a virtual world, the participants generally had to: (a) position and orientate the turtle by using commands (programming language), (b) select a material for 3D shapes from Material Editor, (c) select a 3D primitive command to create a 3D object, and (d) repeat the previous steps. During these processes, the participants tried to create any object at first. When seeing the result in 3D virtual space, they decided whether it was what they wanted or not. They used UNDO command or CS (CLEARSCREEN) to discard objects if they didn't want them. Gradually, they became more confident in making predictions or mental images of the 3D world before they sent in the commands to the console. The construction of 3D virtual worlds was further complicated by the use of REPEAT and SCALE commands. This was evident in Bonbon's creation of a candle, in which she tried many times manipulating the language and the 3D object to get the right scale. For simple objects such as a candle (see Figure 5.20), it was easy for the participants to visualise from the commands. However, complex 3D structures such as Victor's tower (see Figure 5.19) could be difficult to visualise. It was thought that the construction of simple objects could better facilitate these young participants' transfer between the visual (topological resource) and language (typological resource), or their spatial abilities in terms of Bishop's (1983) VP and IFI. Nevertheless, the use of complex commands to build the hard-to-imagine 3D structures was always fun and provided a potential context for advanced learners to advance their knowledge about 3D geometry.

During this learning activity, it was noted that the participants started to interact beyond the socialising level within the Forum. They utilised the Forum to post their programming codes and seemed very keen to show what they had achieved in constructing 3D objects to other participants. It was observed that the Forum had become a place not just for exchanging information but also handy for copying the programming codes. For example, Alekat20 read Bonbon's post about making a candle. She recreated the 3D candle quickly by copying the commands from the forum and pasting into the programming interface. The types of social interaction
also increased when Alekat20 praised, "You are very smart bonbon!!!!" and challenged her, "Can you make any other COOL things!!!!". In addition, Bonbon's poll on "Favourite colour on material editor" was setup timely in this activity, which led to the participants to extensively examine the Material Editor.

Although the participants' use of the Forum had shifted from general conversations to more programming-focused conversations, it was felt that more scaffolds were needed in order to elicit knowledge building discourse. To address this issue, the researcher posted two follow up questions in the Forum trying to engage the participants into deeper knowledge construction of 3D geometry. The first question was in response to Victor's tower. The researcher questioned that if the scaling commands `scalew 3 scaleh 1 scaled 2` were swapped to `scalew 2 scaleh 1 scaled 3`, will the tower still be the same? This question involved the concepts of similarity and congruency. For a box (cube), exchanging the scaling factors of width and depth will not change the shape of the box. In other words, the two boxes will be congruent. However, because Victor's commands `repeat 30 [box fd 0.3 up 0.3 rt 30]` created 30 boxes, together with the UP and RT, the tower would be different if the scale of single box was changed in width and depth. The second question related to Bonbon's creation of a candle. Bonbon used commands `scaleh 7 scalew 0.9 scaled 0.5 cylinder` to create the body of the candle. She then commanded the turtle `up 4` to be the centre position for placing the flame. In VRMath, the turtle's position is always at the centre of the primitive objects (except LABEL) no matter what the scale is. It was obvious that because the scale of height was set to 7 and the turtle represented the centre position of the candle, therefore the top of the candle should have 3.5 above the turtle. Command `up 4` would then bring the turtle 0.5 above the candle, the right position to place the flame. The researcher assumed that Bonbon must have figured out the calculation but the researcher still questioned about why `up 4`? Unfortunately, none of the participants responded to those two questions in the Forum.

Time was a possible reason for not replying in the Forum. The follow up questions were posted after the session of this activity. The participants' reading of the posts was a few days later. By then the details from the activity may have faded in their memory about their posts. During the course of this iteration, the researcher had been asking the participants to read every new post in the Forum. According to
the auto-logging mechanism of the Forum, the participants did read every single post. However, whether or not to reply was still their free choice. This might indicate that replying to these focused questions was not a priority of these young participants. Perhaps a face-to-face direct discussion about those questions and more time should be given to foster a knowledge building culture in the Forum.

The exploration of the virtual 3D primitive objects together with the scaling commands in VRMath led to the emergence of some new ways of thinking about and doing 3D geometry. For example, the exclusion of a dimension using a zero factor had created the 2D projective transformation of the 3D shape. Asking questions such as squashing a cone's height to zero etc., using the programming commands and visualising within 3D virtual space had provided an arena to operationalise the participants' 3D thinking. After this activity, it was found that additional activities could be designed to develop the concepts of similarity, congruency, and transformation using the 3D primitive objects and scaling commands. For example, using the same scale factor for each of the three dimensions can produce reduced or enlarged similar shapes. If these shapes (e.g., box) are overlapped, they can then show the relationship between the scale and volume of the 3D shapes. Many real world objects such as plants, animals, and buildings can also be simplified and constructed by these scaled primitive objects. Activities like that can further operationalise and challenge users' abilities to scale, position, and orientate in 3D space.

7.3 LEARNING ACTIVITY 5: CLIMB UP STAIRS

The purpose of this learning activity was to have the participants construct a real world artefact (i.e., a staircase) using previous learnt commands. The staircase was chosen because it has simple structure and pattern. If properly built, users could actually climb up stairs in WALK mode, and experience the virtual walk through and viewpoints from different height on the stairs.

7.3.1 Results

School C started this activity first. The researcher told School C participants that they were going to build a staircase using REPEAT command. The three participants seemed to be puzzled. (The researcher later realised that School C
participants might have tried but were not familiar with a moving command being used within a repeat command.) The researcher then explained that BOX could be used to be the stairs, and we needed to create a box then forward and up a little bit, and repeat. The three participants thought for a while and then seemed to understand. They began to explore on their computers.

Grae first tried commands `repeat 12 [box fd 1]`. He was a bit surprised to see a long box that he created (Figure 7.7). The researcher explained that an UP command was needed to make a staircase.

![Figure 7.7 A long box created by 12 cubes](image)

At about the same time, Bonbon and Rosco called out excitedly, "I made a staircase". They both came over to see Grae's long box. Rosco wondered and said, "How did he do that?" Soon he realized that was 12 boxes together. Then, the three participants created a staircase by using commands `repeat 12 [box fd 1 up 1]` (see Figure 7.8).

![Figure 7.8 A staircase created by 12 cubes](image)

Rosco tried to walk up the stairs. He navigated using Fly mode and as a result he flew through (penetrated) the stairs and couldn't climb up. He then asked the researcher for help. This is of the nature of virtual reality. In VRMath, the collision detection can be turned on manually but the default setting for Fly mode is off. The researcher then suggested him to use Walk mode. Rosco tried Walk mode, but soon
he encountered another problem. He now collided with the objects. The staircase was too high (1 meter each step). Therefore, he still couldn't walk up. The researcher then suggested to them to use \textit{up 0.5} instead of \textit{up 1} in their repeat commands. A walkable staircase was eventually built by Rosco using commands \textit{repeat 12 [box fd 1 up 0.5]} (see Figure 7.9).

\textit{Figure 7.9} A walkable staircase created by 12 cubes

Now the height of each stair was half a meter, which might be still higher than in reality but was acceptable in both real and virtual space. The researcher advised the participants to turn on the ground (i.e., sand or chessboard ground) to make it more realistic. And with the sand ground turned on (from Navigation Toolbar), the gravity would apply in Walk mode. The participants then enjoyed climbing up and down the stairs. When they reached to the top of the stairs, they could see the turtle. They viewed the virtual space from different height on the stairs, and fell sometimes to the ground. They also tried cone and ball in the repeat command to create a cone staircase or a ball staircase. Interestingly, they could still climb up the cone or ball staircase, and got the up and down feedback along the surface of the cones or balls. Bonbon tried to create a winding staircase by using commands \textit{repeat 12 [box lt 0.5 up 0.5]} and \textit{repeat 12 [cone lt 0.5 up 0.5]}. However, she forgot to go forward and the effect of right turn 0.5 degree was too small to see the winding effect. Before the session ended, Rosco posted a message in the Forum titled "How to build a staircase":

To build a staircase you copy this command:
\textit{repeat 12 [box fd 1 up 0.5 ]}

And you can make other things using the other shapes 2 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈😈😈 😈😈.getBounds()

Hi peoples im rosco!! 😈😈hello
School K participants built a staircase quickly with the help of Rosco's post. They recreated staircases by copying from Rosco's commands and played with walking on them. Alekat20 tried a sphere staircase. Occasionally it could be heard from R2D2 as he said, "Oh.. I fell down" when he fell from the staircase. The staircase was created easily. However, the commands were copied from the Forum. Therefore, the researcher tried to explore their idea about "repeating a box, forward a bit and up a bit" to build a staircase.

Researcher: Do you know how to make the turtle go forward?
Victor: Yes, forward 1.
Researcher: Good, I create a box here (with body gestures) and I forward 1, and then... put another box here?
Victor: no.
Researcher: no...for a staircase you have to go up a little bit, so after forward 1...
Victor: up 1
Researcher: Yes, up 1, and then?
Victor: up 1 forward 1 up 1 forward 1 up 1 forward 1 .... Reseracher: put another box here. Then you can put the box, forward and up commands into the repeat command.

The researcher then asked the participants to improve the staircase by using scaling commands and selecting different materials. It was noted that R2D2 was constructing commands `repeat 12 [cone lt 0.5 up 0.5]`, which was exactly as Bonbon did before. The researcher realised that R2D2 was trying to build a winding construction. Therefore, the researcher also challenged the participants to add in a turning command in the repeat command to build a "spiral staircase". It was felt that the participants didn't seem to understand about a turning command in the repeat command. But after a few tries, they soon got the idea.

It was observed that Alekat20 tried commands `scaleh 0.3 scaled 0.5 scalew 2 repeat 12 [box up 0.5 fd 1 rt 30]`, and `repeat 12 [box up 0.5 fd 0.5 rt 10]`. She also tried `scale 0.5` (not enough inputs), `scaleh 0.5 scaled 0.5 scalew 2`, and `sphere up 0.5 fd 0.5 rt 10`. She utilised the Up key to repeat the `sphere up 0.5 fd 0.5 rt 10` commands for 134 times just to observe slowly how the spiral staircase was built. R2D2 tried `repeat 24 [box up 0.5 fd 1 rt 30]`, `repeat 24 [cone up 0.5 fd 1 rt 30]`, `repeat 24 [sphere up 0.5 fd 1 rt 30]`. He also tried the build several staircases by moving the turtle away to start a new staircase. An interesting trial by R2D2 was that he commanded `cs rd 180 repeat 10 [cone up 0.5 fd 1 rt 30] rd 180 repeat 10 [cone
up 0.5 fd 1 rt 30], which created up-side-down cones in the spiral structure (Figure 7.10).

![Figure 7.10 A spiral structure of up-side-down cones](image)

Victor was a playful boy. He tried scaled 0 scaleh 0 scalew 2 box, which created nothing in the 3D space. He commanded repeat 30 [box fd 0.3 up 0.3 rt 30], and used Up key to repeat the command for another 34 times, which created a total of 1050 stairs. He was then happily climbing up the stairs to try to catch the turtle. By the end of session, Alekat20 posted a message in the forum titled “How to Make a better stair case” in reply to Rosco’s “How to build a staircase.”

Today with Andy we learnt that you can make a better stair case.
The command to Type Is ....
scaleh 0.5
scaled 0.5
scalew 2
repeat 12 [box up 0.5 fd 0.5 rt 10]
Good luck....
TIP: you must type in the scale commands first.... (before typing in repeat 12.........
Alekat20

Can You Talk To Me??????????? Please!!!!!
Can You Make the Turtles Body A Different Colour??????????

The staircase in the post above is presented in Figure 7.11 below.

![Figure 7.11 A spiral staircase from knowledge building discourse](image)
R2D2 seemed to be inspired by the spiral staircase structure. He experimented with making the DNA structure in VRMath, and posted:

```
hi can you you make a DNA structure i can. all you have to do is make a spiral staircase (see my other post) but write in sphere at start. then say scalew
```

R2D2 didn't provide the programming codes for his DNA model. However, according to his command history in the auto-logging file, his DNA model could be reconstructed as shown in Figure 7.12 below.

```
Cs repeat 24 [sphere up 0.5 fd 1 rt 30]
repeat 24 [sphere up 0.5 fd 1 rt 30]
HOME
east 5
south 5
repeat 24 [sphere up 0.5 fd 1 rt 30]
repeat 24 [sphere up 0.5 fd 1 rt 30]
repeat 24 [sphere up 0.5 fd 1 rt 30]
repeat 24 [sphere up 0.5 fd 1 rt 30]

Figure 7.12 A spiral DNA structure
```

Alekat20 found it "funny" that every participant was using number 12 in the repeat command. She wrote in the Forum titled "how come???????":

```
When rosco first posted a message telling every one how to make a stair case he had .... repeat 12...... and now every time some one types in a post they write repeat 12....

Why Is This....
```

or Is Rosco a bit of a trend setter???????

```
Can You Talk To Me???????????? Please!!!!!
Can You Make the Turtles Body A Different Colour???????????
```

She also setup a poll regarding this finding (see Figure 7.13). The poll was quite humorous and provoking. The researcher tried to scaffold this discussion by pointing out `repeat 12 [rt 30]` in the turtle dance activity, and there was a relationship for 12 and 30 (i.e., to make a 360 degree full turn). However, this didn't generate much attention among the participants. Only four including the researcher voted.
7.3.2 Discussion

This activity was a culminating activity because it involved the participants using all previously learnt 3D navigation skills and programming commands. In this staircase activity, it was observed that all participants had experienced and utilised the following operations and facilities in VRMath:

1. Use scaling commands for a better size of stairs.
2. Select material from Material Editor.
3. Use navigation toolbar including Sand Ground and others.
4. Build a straight up staircase by forward a little bit and up a little bit.
5. Build a spiral staircase by turning a little bit each stair.
6. Judge the distance and degree to move and turn.
7. Climb up stairs and race to the top to see the turtle.
8. Experience different views from different heights on the stairs.
9. Use different primitive objects such as box, ball, cone and can.

In doing these operations, the participants actually improved their spatial abilities. For example, the 3D structures (e.g., a long staircase) they built were quite large. When the large 3D structures were created, only a small part of the 3D structures could be seen through the limited 2D square viewport on the screen. At first construction, these participants were somewhat surprised by the 3D structures created by the simple repeat commands. But gradually, it was noted that the participants were utilising visual imagery in their mind when they used commands to create those large 3D structures (i.e., IFI, see Bishop, 1983). This was evident from the construction process, in which the participants could use more commands before they navigated to check. Therefore, even though the whole 3D structure was not in

---

Figure 7.13 A poll about why number 12 is so popular

<table>
<thead>
<tr>
<th>Why Is The Number 12 so popular??</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>because 1 and 2 are side by side on the keyboard!!!</td>
<td>25% [ 1 ]</td>
</tr>
<tr>
<td>I have No IDEA What So Ever!!!</td>
<td>0% [ 0 ]</td>
</tr>
<tr>
<td>Rosco is A TREND SETTER!!! (No offence Rosco)</td>
<td>0% [ 0 ]</td>
</tr>
<tr>
<td>I Have a Different answer!!! (If You Do Please Post To the forum)</td>
<td>75% [ 3 ]</td>
</tr>
</tbody>
</table>

Total Votes : 4
view, the participants were able to visualise and navigate to view the hidden parts (i.e., visual memory, see Hoffer, 1977).

The idea of using the turtle as a reference point for positioning and orientating 3D objects was further developed in this activity (i.e., spatial visualisation, see McGee, 1979). This was evident when the participants (a) changed the turning degree for the spiral staircases; (b) moved the turtle to a new position to start another staircase; and (c) used command \textit{rd 180} (roll down) to create up-side-down cones. Climbing up stairs also provided opportunities to operationalise the participants' spatial orientation ability (McGee, 1979) especially when climbing up hundreds of spiral stairs to catch up the turtle, and then view from top of the staircase.

The utilisation of \textit{REPEAT} command was found not easy in this activity. Although the participants had used \textit{REPEAT} command to create many 3D structures, it was felt that the participants didn't develop the idea that the things to repeat were actually the patterns of the 3D structure. The first possible reason for that could be due to the nature of programming, in which the syntax of \textit{REPEAT} command was not clear to the participants. This could be addressed in the future by advising students that more than one or two commands can be put in between the square bracket (i.e., \texttt{[ ]}) of the \textit{REPEAT} command. The second and possibly more plausible reason could have been that the researcher may have inadvertently focused the participants' on one way thinking only where the participants only practiced from commands to visualisations (i.e., only utilised IFI, see Bishop, 1983). Therefore, it was felt that in the future, participants should also be encouraged to interpret from visualisations to form the commands (i.e., also utilise VP, see Bishop, 1983). For example, the participants could observe a real staircase; investigate the pattern between each stair (e.g., distance and height). This may help the formation of the \textit{REPEAT} command and the establishment of the relationship between the real world patterns (i.e., objects) and the abstract commands (i.e., signs).

When a staircase was created by School C participants using commands such as \textit{repeat 12 [box fd 1 up 1]}, the researcher was concerned about the transfer of knowledge from the virtual staircase to a real world staircase. The virtual staircase did look like a real staircase (see Figure 7.8). However, the proportions of the staircase were problematic. A stair with one metre in height and depth doesn't make...
sense. Fortunately, the VR environment in VRMath provided mechanisms to raise the participants' awareness regarding this. It was noted that the VR environment, as was entitled "virtual reality", had more to offer than general 3D graphics. The VR environment of VRMath could simulate gravity and collision detection in Walk mode. When R2D2 tried to walk on the staircase with one metre high each stair, the VR environment gave feedback and disallowed the action. This gave an opportunity for the researcher to raise the issue of the scale and size, and the distance to move the turtle for constructing a proper staircase.

The scaling commands SCALEW, SCALEH, and SCALED were favoured by the participants above utilising SCALE with three parameters (i.e., \texttt{scale x y z}). The SCALE command, though easier to type, caused more problems for the participants because of the requirement of three parameters. For example, Alekat20 again in this activity used command \texttt{scale 0.5}, which generated a "not enough inputs" feedback. She didn't bother and went on to use SCALEW, SCALEH, and SCALED as the other participants. This however, gave an idea to the researcher for the future implementation of VRMath, in which command such as \texttt{scale 0.5} can be accepted to mean all three dimensions are using 0.5 scale.

When the scaling commands were specified, the subsequent 3D shapes would be scaled. This raised the issue to move the turtle accordingly to place the next stair. For example, Alekat20 tried commands \texttt{scaleh 0.3 scaled 0.5 scalew 2 repeat 12 [box up 0.5 fd 1 rt 30]}. She found that there was a small gap in height and a bigger gap between each stair. She was then able to adjust the parameters for UP, FD, and RT. Among her adjustments that were recorded were \texttt{repeat 12 [box up 0.1 fd 1 rt 30]}, \texttt{repeat 12 [box up 0.5 fd 0.1 rt 30]}, \texttt{repeat 12 [box up 0.5 fd 1 rt 10]}, \texttt{repeat 12 [box up 0.5 fd 0.5 rt 10]}, \texttt{repeat 12 [box up 0.5 fd 0.5 rt 30]}, and \texttt{scaleh 0.5}. And finally she was satisfied with \texttt{repeat 12 [box up 0.5 fd 0.5 rt 10]}, and posted it to the Forum. This indicated that an understanding about the relationship between scaling factor and size was developing while constructing a proper staircase. Nevertheless, a lack of thinking about real world staircases was also evident because stairs with half a metre high are not common in real life. It was thought that the measurement of a real staircase may be included in this activity to better connect "virtual" and "reality".

It was observed that confusion about the distance to move and the degree to turn might have occurred. This was evident when both Bonbon and R2D2 used
commands *repeat 12 [cone lt 0.5 up 0.5]*. To turn left a degree as small as 0.5 is rare and not easy to observe in the 3D virtual environment. It didn't seem that the participants were using the small turning degree purposefully. Therefore, the use of 0.5 degree might indicate a lack of thinking about the commands and the understanding about angle. According to the log files, Rosco had also specified turning angles such as *lt 1000* and *lt 1260*. Grae had commands such as *tl 303*, *tl 305*, *tl 2000*, *tl 2003*, and even *tl 2000355524452155544121455525555555555*. These degrees were not relevant to 360 degrees and its derivatives such as multiples and factors. However, it was observed that the participants were very engaged in experimenting with the commands. After a few tries of commands and observations of the turtle, the participants had picked up some observable degrees such as 10, 20, and 30 etc. in their commands. This indicated that the VRMath environment could guide learners to the formal thinking about angles and provide a sense of the approximation of degrees.

Excellent coordination in the use of the three interfaces (semiotic resources) was observed among the participants. In particular, new types of social interactions were found in this activity. For example, Alekat20 built on Rosco's idea about making a staircase. She posted it to add in scales and a turning command to make the staircase spiral. It was noted that after reading her posts in the Forum, School C participants also followed to create many spiral staircases. As the time by which these participants could spend in the Forum was limited, discussions between the two schools' participants were valuable. These discussions can be referred to as the knowledge building discourse (Scardamalia & Bereiter, 2002), which is an important practice of the Forum community. However, because of the small number of community members (participants) and the constraint of time, it was not easy for the knowledge building discourse to occur. Another example is the "why repeat 12?" question asked by Alekat20. She also set up a poll to induce discussion. Unfortunately only R2D2 and the researcher replied. R2D2 thought that 12 was a popular number. In fact, "repeat 12" was first used in the turtle dance activity to turn the turtle a full round (using commands *repeat 12 [rt 30]*). It was thought that "why repeat 12?" could be a good question to involve the participants' thinking about angles. However, there was reluctance to generate discussion despite the researcher's encouragement in the Forum.
Creative uses of command were noted. These included Victor's exploration of two zero dimensions, and R2D2's spiral DNA structure. The researcher will consider integrating these activities into future iterations.

7.4 LEARNING ACTIVITY 6: 3D ROTATION

This learning activity was designed to investigate the non-commutative nature of 3D rotation using VRMath environment. In this activity, a question “Will you face the same direction when you turn right 45 degrees first then roll up 45 degrees, or you roll up 45 degrees first then turn right 45 degrees?” was posed to the participants. They were asked to predict the result and then experiment with VRMath.

7.4.1 Results

The initial thinking of all participants was that the two 3D rotations (RU 45 RT 45 and RT 45 RU 45) would result in same direction regardless of the performing sequence. This thinking was challenged when the participants interacted with VRMath. The processes by which each of the participants changed their conceptual understanding of 3D rotation will be described.

Rosco’s experiment: Avatar View

When Rosco was asked to justify his thoughts about the 3D rotation problem, he immediately came up with the idea of using the “Avatar View” in VRMath. Avatar View is a function by which the user temporarily becomes the turtle and views actions within the 3D virtual space from the turtle’s perspective. In this mode, 3D navigation by mouse and keyboard in VR space are disabled to prevent changing the viewpoint. The use of commands is the only way to manipulate the turtle’s position and orientation and to change the viewpoint. Bonbon suggested that Rosco switched on the Compass in VR space in order to see the degrees. Rosco thus began his experiment as illustrated in Figure 7.14.
To his surprise, Rosco found that the views of Picture 4 \((RU \ 45 \ RT \ 45)\) and Picture 6 \((RT \ 45 \ RU \ 45)\) in Figure 5.28, which he originally thought would be the same, looked different. Because of the different part of the sky he (or the turtle) saw, he then started to think that a different order of two 3D rotations may end up with different directions. He also contributed his idea of using Avatar View in the Forum in the following posting titled “How to determine if .......”:

```
How to determine if ru 45 lt 45
and lt 45 ru 45
have da same viewpoint
1. go to avatar view
2. copy this text ru 45 lt 45
3. copy this text lt 45 ru 45
4. SEE FOR YOURSELF
```

Bonbon’s exploration: Look at the turtle

Bonbon used her hands to simulate the two 3D rotations, and was sure that the two 3D rotations were the same. She conducted an experiment by watching the turtle turns, but she decided to try on RU and LT (left) instead of RU and RT (right). The processes of her experiment are illustrated in Figure 7.15.
Bonbon carefully compared the two views of picture 3 (RU 45 LT 45) and 6 (LT 45 RU 45) and noticed that they were different. However, before she made a conclusion, she also tried tilting rotations (TL and TR) with RU and smaller degrees, and together with Rosco’s Avatar View experiment, she convinced herself that the two 3D rotations resulted in different results.

**Grae’s experiment: Create 3D objects**

After seeing Rosco’s and Bonbon’s experiment, Grae could not think of any idea to show the difference between the two 3D rotations. The researcher encouraged him to try to create a 3D object after each 3D rotation. Grae then decided to create a sphere after each 3D rotation. He used commands “RU 45 RT 45 BALL” to create the first sphere, and then “HOME RT 45 RU 45 BALL” for the second sphere. The processes are illustrated in Figure 7.16.

Grae originally thought that the two spheres should be somewhat overlapped but located in a different place. However, he was confused when he navigated to see the two balls from different viewpoints; they seemed to be one ball. The researcher
then suggested him to try on CUBE instead of BALL and with different colours. Figure 7.17 shows the processes of creating cubes after each rotation.

![Diagram of creating cubes](image)

**Figure 7.17 Create 3D objects experiment about 3D rotation (2)**

Grae was then satisfied with this result, and with the help from the researcher, Grae posted a message titled “two turns must take turns” in the Forum:

Hi,
if you lt 45 ru 45, or if you ru 45 lt 45 Will these be the same?
you can check the answer by doing:
1. home lt 45 ru 45 cube so you have a cube...
2. you pick another color from the material editor.
3. home lt 45 ru 45 cube
so you have another cube but this time the turtle go lt 45 first then ru 45
do you think that the two cubes are in the same place???
--
graef

After Rosco, Bonbon, and Grae had completed their solutions to the 3D rotation problem, the researcher challenged their spatial thinking again with another question “if you do **RU 45, RT 45, RD 45**, and then **LT 45**, will you go back to the original direction?” Interestingly, even though they had found that the two 3D rotations produced different result; they still thought that the turtle could go back to its original direction. They, however, found their prediction to be incorrect.

**Alekat20 and R2D2’s reasoning**

Alekat20 and R2D2 tried to solve this 3D rotation problem on the next day. Therefore, they had the chance to see the other three students’ posts in the Forum and followed their ideas. However, before they used the ideas in the Forum and interacted with VRMath, the researcher challenged their thinking about 3D rotation. Interestingly, they came up with some ideas quite different from the other group.

---

24 Victor from School K was absent in this 3D rotation activity.
In addition to the use of hands, or other objects, to simulate the two 3D rotations, Alekat20 and R2D2 used their head movement and eyes to determine whether or not the two 3D rotations would generate the same outcome directionally. They practiced turning their heads \( RU \ 45 \) then \( RT \ 45 \), and tried to remember the spot they saw or faced to. Then they did the same thing with \( RT \ 45 \) first then \( RU \ 45 \). However, both of them claimed that the two 3D rotations would produce the same outcome because they were seeing the same spot.

Alekat20 started to draw grids on paper (see Figure 7.18). She reasoned that:

\[ \text{Alekat20: If you go FD 1 first and then EAST 1, you will end up with the same position as you go EAST 1 first and then FD 1.} \]
\[ \text{Researcher: But we are talking about turning...} \]
\[ \text{Alekat20: Oh, ya., but I think they are the same.} \]

![Figure 7.18 Reasoning from moving for turning](image)

Alekat20 and R2D2 then followed the ideas posted in the Forum. After they tried the Avatar View and created objects, they concluded that the sequence of performing 3D rotations does matter. However, they still had some doubts in mind, as they replied to Grae in the forum:

\[ \text{R2D2: no because me and alekat20 just tried and guess what thier (sic.) not the same. (strange 😕)} \]
\[ \text{Alekat20: R2D2 And I Found That this is not true!!!!!!!! How Weird... 😞保修} \]

7.4.2 Discussion

From “the two 3D rotations are the same” at the beginning to “the sequence of performing 3D rotations does matter” at the end, the five young participants experienced a conceptual shift after their interactions with VRMath.

The non-commutative nature of 3D rotation may be easily understood by one who can perform trigonometry in 3D coordinate system, but it would be very difficult for most people if we can only use our body movements, senses or feelings, mental reasoning, and other concrete objects. It is evident that although students live
in a 3D space, they have limitations in imaging or thinking three dimensionally. In this activity, Alekat20 and R2D2 utilised a traditional way to solve the 3D rotation problem by moving their heads to see and feel the direction. Unfortunately, there was a lack of accuracy in this, an insufficiency that became apparent to these students once the rotations were investigated within the VRMath 3D virtual environment.

The VR interface of VRMath which enabled the students to switch between first- and third-person experiences facilitated dynamic visualisations of the 3D rotations. Rosco, for example, utilised the Avatar View to simulate the body movement, which was a typical example of using a computer to address a limitation with real world experiences within 3D space. In the Avatar View, Rosco temporarily became the turtle and viewed the rotations from the turtle’s perspective. At the same time, he also manipulated the turtle’s orientation by using 3D rotation commands. This operation of switching from third-person experience (watching the turtle turning) to first-person experience (turning himself) allowed Rosco to see different portion of the sky, and as a result, to realise the non-commutative nature of 3D rotations and thus correctly solve the 3D rotation problem posed by the researcher. Rosco’s experiences confirmed the benefit of switching between first- and third-person imagery (Amorim et al., 2000), and also addressed the lack of accuracy of using body (e.g., head) movement as performed by Alekat20 and R2D2.

Bonbon and Grae used the Logo-like programming language to manipulate the turtle and build 3D objects in VR space to solve this 3D rotation problem. Bonbon’s experiment demonstrated again that the computational environment VRMath easily and accurately showed the two 3D rotations were different, which was in contrast to the use of her hands to simulate the 3D rotation. Grae’s experiment of creating objects was another approach to successfully solve this 3D rotation problem. Nevertheless, he also found that creating a sphere after each set of 3D rotation would not show any difference of the two 3D rotations because as long as the turtle doesn’t move, the centre for spheres remains the same. His accidental use of a sphere actually revealed its unique radius property against other 3D objects such as a cube. This provided a valuable opportunity for learning about 3D shapes.

One important misconception about 3D rotation found in this activity was Alekat20’s reasoning from moving to turning. She argued that changing direction should be the same as changing location. In knowledge construction, it is common
that one tries to assimilate new problems from his/her similar past experience. Unfortunately, moving on a 2D plane can be a false analogy for 3D rotation. This type of reasoning has been identified in the research literature as the "intuitive rules: same A same B", in which the students react to the new mathematical or scientific problems with intuitions from their similar experiences (Stavy, Tsamir, Tirosh, Lin, & McRobbie, 2002). 3D rotations, however, counter with many of our intuitions. This also indicates that though we live in a 3D world, 3D thinking is difficult, and a VRLE such as VRMath can provide opportunities for 3D thinking and thus advance and facilitate the knowledge construction of 3D geometry.

Another common misconception about 3D rotation found in participants in this activity was thinking that a turning could be eliminated by its opposite turning performed later in a series of 3D rotations. For example, in the four rotations RU 45 RT 45 RD 45 LT 45, students with this misconception believe that RU can be eliminated by RD and RT by LT. However, as VRMath showed, a rotation of another dimension in between the two opposite turns means that the two rotations of the same dimension still cannot eliminate each other. This is an intriguing question and can be further examined in later studies. For example, as showed by VRMath, the four rotations RU 45 RT 45 RD 45 LT 45 cannot bring the turtle back to its starting orientation. The question "what direction will the turtle face?" can then be posed. A trial in VRMath shows that a cone shape can be formed with commands `pd repeat 180 [fd 1 bk 1 rt 45 ru 45 lt 45 rd 45] pu` (Figure 7.19).

![Figure 7.19 A cone shape formed by 3D rotations](image)

More questions can then be derived from the four 3D rotations. For example, what if the degree turned is 30 or 60, what shape will it be? The answer is actually different size of cones will be formed. In this example, the complexity of 3D rotations is still difficult to comprehend. However, it presents a new way of thinking
about and exploring a cone that may lead to some new and interesting discoveries by students about mathematics.

7.5 LEARNING ACTIVITY 7: FORMULA OF POLYGON

This learning activity was designed to investigate the pattern of regular polygons. In this activity, the participants were asked to generate a formula using REPEAT command. The desired formula is \texttt{repeat :side [fd 1 rt 360/:side]}. As this activity had been conducted in Iteration 1, many useful scaffolds were carried over into this activity. For example, free exploration of triangle, square, pentagon, hexagon, heptagon, and octagon etc. with body movement, and a table were utilised to scaffold the discovery of the patterns and relationships between those regular polygons. After many explorations in 3D shapes from previous learning activities, this learning activity introduced PD (PenDown) and other relevant commands to draw 2D lines in 3D space.

7.5.1 Results

As usual, the participants started with reading and replying new posts and private messages. Before this activity, the researcher had sent a private message summarising previous learning activities to all participants. Rosco read it out aloud while he was looking at the private message sent by the researcher.

\texttt{Hi COLABers,}

\texttt{It is so quick.....}
\texttt{We have had seven hours with VRMath, half way to the end of this project.}
\texttt{I hope that you have enjoyed very much within VRMath playing with the turtle and knowing each other, haven't you?}
\texttt{So far, we have learnt}
\texttt{ chats how to communicate each other in forum,}
\texttt{ chats how to navigate in 3D virtual space,}
\texttt{ chats and how to command the turtle to move and turn.}
\texttt{ chats We can also build some basic 3d shapes with different scales and materials..}
\texttt{ chats We can repeat commands to build stair cases or DNA models.}
\texttt{ chats We also found that two turning commands must take turns, otherwise you won't get the same direction (weird 😐)}
\texttt{What else??? you can think of more...}
\texttt{ chats By knowing all these things in VRMath, you are probably able to build a virtual world....}
You can start to think of a virtual world that you would like to build at the end of this project… such as a city, a math pattern, an animal, or a car...

I can't wait to seeing your creations..... 😊

Let's move on! 🎉

PS: Special thanks all of you for posting in the right forum and giving many positive feedback and comments.. 😊

_________________

Andy *^___^*

After reading the message, Rosco deleted the message straight away. The researcher asked him, "Why not keep the message?" Rosco said that he replied to most of the private messages, but he didn't think that he needed to reply to this one. Therefore, he deleted the message sent by the researcher. It seemed that he didn't like to keep messages in his Inbox. However, the researcher found that he didn't delete messages in the Sentbox, and he didn't know that he could organise messages by moving important messages into Savebox. At the same time, Bonbon was using the search function in the forum. She searched Alekat20's posts to see if there was any follow-up question. She also voted in the Forum and edited personal information in her profile as other participants did.

Bonbon was asked to join the discussion with Rosco being the operator of the computer. The researcher first asked them to bring out the Quick Command, from which the PenDown (PD) and PenUp (PU) commands were introduced. The researcher explained that the Pen Color Editor (see Figure 3.13) could be used to select colours for the lines created by PD command. Bonbon had been changing the pen colours in previous learning activity. Until now she understood why the Pen Color Editor hadn't worked on the 3D shapes. Rosco quickly tried pd and fd 1 commands. However, he was puzzled to see that the turtle forwarded 1 metre but the line created was longer than 1 metre. This was because Rosco had changed the scale some minutes ago. The researcher then introduced RESET command, which resets the scale, material, and colours to the default settings (i.e., scale 1 1 1, gray material and black pen colour).

The researcher asked Rosco to draw a shape by repeatedly clicking on FD and RT in the Quick Command. As the default degree in Quick Command was 30, Rosco ended up with a 12-side dodecagon. However, because of the viewpoint is not
the top view of the polygon (it was the original viewpoint), both of the participants saw the shape and exclaimed "It's an oval". Rosco then navigated to examine the polygon from different viewpoints. Both Bonbon and Rosco didn't know the name of dodecagon but they described it as "many sides". They also counted and found that there were 12 sides in the polygon. The researcher tried to link the degrees turned and the number of sides by saying, "The turtle is turning 30 degrees and that makes 12 sides". The polygon was created by quick mouse clicking in a short time. It seemed that the participants didn't quite follow the process of the moving and turning. Therefore, the researcher asked Rosco to stand up and pace out a square.

Grae was about 20 minutes late. He arrived at this time and joined in the activity.

Rosco followed the researcher's commands FORWARD 1 LEFT 90 FORWARD 1 LEFT 90 FORWARD 1 LEFT 90 FORWARD 1 LEFT 90. From his body movement, the participants noticed that he had walked a square. The researcher again tried to link the degrees and the sides by saying "You turned 90 degrees so you get a square". The researcher then asked them to try in Quick Command and told them to change the degrees for a square. It was observed that the participants restored viewpoint, cleared the screen, brought the turtle home, and then started the construction of a square using Quick Command.

Bonbon first typed and tried a sequence of clicks in Quick Command. She called the researcher and said that she had created a triangle (see Figure 7.20). The researcher asked her to hide the turtle and navigate to examine. She found that it was not quite a triangle.
It was noted that Bonbon didn't change the degree to 90. She tried another
sequence of commands. However, with the original viewpoint, it was very difficult to
draw a square (see Figure 7.21).
PD FD 1 FD 1 FD 1 FD 1 FD 1 FD 1 FD 1 FD 1 FD 1 FD 1 FD 1 FD 1 FD 1 FD 1 FD
1 FD 1 RT 30 RT 30 RT 30 FD 1 FD 1 FD 1 FD 1 FD 1 FD 1 FD 1 FD 1 FD 1 FD 1
FD 1 FD 1 FD 1 FD 1 RT 30 RT 30 RT 30 BK 1 BK 1 FD 1 FD 1 FD 1 FD 1 FD 1
FD 1 FD 1 FD 1 RT 30 RT 30 RT 30 FD 1 FD 1 FD 1 FD 1 FD 1 FD 1 FD 1 FD 1 FD
1 FD 1 FD 1 FD 1 FD 1 RT 30 RT 30 RT 30 FD 1 FD 1 FD 1 BK 1 BK 1 BK 1 BK 1
BK 1 BK 1 BK 1 BK 1 BK 1 BK 1 BK 1 BK 1 BK 1 BK 1 BK 1 LT 30 FD 1 FD 1.

Original viewpoint

Top view

Figure 7.21 Bonbon's attempt for a square
Grae changed the degree to 90 in Quick Command. Similarly, he tried to
draw a big square as Bonbon did. After many clicks in Quick Command, he didn't
get a square but created some lines on different planes (see Figure 7.22).

Figure 7.22 Grae's attempt for a square
The researcher gathered the three participants to discuss regular polygons.
The researcher first explained that FD 1 in VRMath means forward 1 metre.
Therefore, there was no need to forward one metre a lot of times. And because we
are making 2D shapes, we only need RT or LT, FD or BK commands to draw on one
plane. The researcher drew a triangle and a square on a paper (see Figure 7.23) and
questioned:

228


Figure 7.23 Polygon scaffold

Researcher: If the turtle starts here (pointing to the turtle at the triangle), tell me what should the turtle do first?

Participants: Forward...

Researcher: Forward 1, good, then what degrees to turn?

Bonbon: 45...

Rosco: 50?

Researcher: This is already 90 degrees (pointing to the dash line), it must be...

Rosco: 45?

Researcher: must be over 90. I will give you a hint. For a triangle, you have to turn three times (tracing the three sides of the triangle) and that makes 360 degrees for a round.

(The participants thought for a while)

Researcher: So we must fd 1 and then rt some degrees...

Bonbon: 120 (exclaimed)

Researcher: OK, so we forward 1 and turn, for how many times?

Rosco: Forward 1 rt ...

Researcher: Because it is a triangle, so how many times?

Rosco: You can repeat it.

Researcher: OK! Good.

The researcher wrote down the commands as repeat 3 [fd 1 rt ?] with the help of the participants. Bonbon said, "I know I know" and couldn't wait to try in VRMath. The researcher suggested the participants to start with a square because they knew the degrees to turn were 90. The participants began to use commands instead of Quick Command. Bonbon quickly created a triangle using cs pd repeat 3 [fd 1 right 120] pu. She called out excitedly, "I did a triangle". Grae typed commands cs pd repeat 4 [fp 1 rt 90] but didn't get a square due to the typo fp (should be fd).

After their initial exploration with REPEAT command, the researcher gathered the three participants again to create a table for generalising polygons (Table 7.1).
Table 7.1

*Generalisation of the Polygon (1)*

<table>
<thead>
<tr>
<th>Sides</th>
<th>Degrees to turn</th>
<th>Shape</th>
<th>Relationship</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>120</td>
<td>Triangle</td>
<td>3 x 120 = 360</td>
</tr>
<tr>
<td>4</td>
<td>90</td>
<td>Square</td>
<td>4 x 90 = 360</td>
</tr>
<tr>
<td>5</td>
<td>?</td>
<td>Pentagon</td>
<td>5 x ? = 360</td>
</tr>
<tr>
<td>6</td>
<td>?</td>
<td>Hexagon</td>
<td>6 x ? = 360</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The researcher continued to ask:

*Researcher: Can anyone guess what will be the degrees to turn for a pentagon?*
*Bonbon: Pentagon is five, right?*
*Researcher: Five sides, yes.
Rosco: This is hard.*

*Researcher: OK, did you see anything about that? (pointing to the relationship: 3 x 120 = 360, 4 x 90 = 360)*
*Rosco: Why is that 360?*
*Researcher: Because you just turn around (tracing to triangle) and turn around (tracing the square), and that is 360, so the pentagon is….*
*Bonbon: 5 times ….*
*Researcher: something is 360, or you can….*
*Rosco: Do a divide.
Bonbon: 5 times 12 is 60…*
*Researcher: May be you can try this hexagon first.
Bonbon: Hexagon is six, right?*
*Researcher: Yes.*

It seemed that the participants had difficulties in doing the calculations of degrees. The researcher then introduced the PRINT command, which could calculate the given number and maths operations in the parameter. Bonbon quickly typed command *print 360/5* and said, "72". Rosco also got 72 and said, "Hey, this is good. I am going to try something". He tried *print 1+1* and got the result 2. After he tested the command, he said, "I can use this to do my maths homework".

The researcher reminded the participants to use the 72 degrees in the REPEAT command. At the same time, Bonbon had already typed in *pd repeat 5 [fd 1 right 72] pu* and created a pentagon in the 3D virtual space. She also tried *pd repeat 5 [fd 0.8 right 72] pu* and *pd repeat 5 [fd 4 right 72] pu* to get two different
size pentagons. Rosco got 60 from \textit{print 360/6} but he couldn't formulate the command. He then asked for help from Bonbon and copied her pentagon commands. Both of them hid the turtle and examined the pentagon from different continuous viewpoints. After that, Rosco made a hexagon easily and was very happy about that. Bonbon went on to find 36 from commands \textit{print 360/10}. However, she didn't create a decagon because she didn't change the repeat count: \textit{repeat 5 [fd 4 right 36]}. It didn't take long for her to debug the repeat count problem.

Grae had been very quiet in this activity. Because his late coming in some sessions, he seemed not to understand the commands well. It was observed that he got into trouble with the following commands:

\textit{print 360/72} (got 5 from the programming console)  
\textit{repeat 5} (not enough inputs)  
\textit{repeat 10} (not enough inputs)  
\textit{repeat 102} (not enough inputs)  
\textit{repeat 102232} (not enough inputs)  
\textit{repeat 5/ print} (not enough inputs)  
\textit{repeat 5/print} (not enough inputs)  
\textit{360/7} (got "You don't say what to do with 51.42857142857143")

The researcher then assisted him with some direct instructions, and asked the participants to try a 7-sided septagon. Rosco was surprised to find the 51.42857142857143 from \textit{print 360/7}. He tried \textit{repeat 7 [fd 1 rt 52]} and \textit{repeat 7 [fd 1 rt 51]}, but found something wrong about the septagon when he hid the turtle and examined (see Figure 7.24).

\textit{Figure 7.24 Overlap and gap in septagons (1)}

Bonbon said, "I know I know I know", and tried 51.5 degrees in the commands. She hoorayed, "I did it". But when she navigated closer to see, she found
that the septagon did not connect. She went on to try 51.3, 51.4, and 51.45 degrees, and was puzzled by the results (see Figure 7.25).

![Figure 7.25 Overlap and gap in septagons (2)]

Rosco kept saying that the number 51.42857142857143 was weird. The researcher then introduced that mathematical operations could be put into the commands such as `repeat 7 [fd 1 rt 360/7]`. The participants replied "cool" and tried it in VRMath. Together, the researcher and the participants filled in and completed the polygon table (Table 4.7). The formula for regular polygon was also formed as `repeat number [fd 1 rt 360/number]`, from which the participants correctly explained that the number was the number of sides of the polygon. The researcher further questioned, "If we create a 360-sides polygon, what will it look like?" The participants were able to predict it as a circle and verified this in VRMath.

School K participants went through similar processes as the School C participants in this activity. At the beginning of this session, Alekat20 was counting how many messages she had posted in the forum. In comparison with other participants, she was the most active user in the Forum. R2D2 and Victor were also busy checking new posts and new private messages. It was observed that Victor was deleting his private messages and editing his posts. This was brought to the researcher's attention because the Forum system did not reserve the original content of those edited posts. It was noticed that Alekat20 and R2D2 also edited some of their posts.

Before using Quick Command to draw lines, the PRINT command was introduced to the participants. The researcher posed a question `print 1+2*3` to the participants. They calculated to get 9 but were surprised to see 7 calculated by the turtle. Victor immediately realised that the turtle was correct because multiplication should be performed before addition. They were much motivated and went on to test
the turtle with some calculations such as \( \text{print } 1 + 2 \times 5 + 9 \times 3 / 7 - 1 + 34 / 1 \times 2345 - 1234 \times 222 - 555 \).

Quick Command was utilised to introduce PD, PU and other related commands. When PenDown and PenUp were introduced, R2D2 asked why not just use PEN as the command to draw lines. The researcher explained that because the turtle needs to know when to draw and when to stop drawing, therefore it needs two commands PD and PU. Alekat20 also asked: does PenUp mean to pick up the pen to draw, and PenDown means to put the pen down to stop drawing. This was just opposite to what PU and PD meant. The researcher explained that it was just like holding a pen above a paper. Therefore, we could PenDown to draw on the paper and PenUp to lift up the pen away from the paper. After the researcher's explanation, they accepted that PD was to start drawing and PU was to stop drawing.

The researcher then allowed some time for the participants to freely draw 2D lines in the 3D virtual space using Quick Command. It was noted that both Alekat20 and R2D2 utilised most functions in Quick Command. They both drew some colourful lines (changed pen colour) in some planes (used roll and tilt commands). Victor didn't change pen colour at all; he only drew lines on one plane (used only FD, BK, RT and LT).

After the free drawing of 2D lines, the researcher gathered the participants and asked Alekat20 to walk a square. The researcher analysed by pointing out the processes as FD 1 RT 90 FD 1 RT 90 FD 1 RT 90 FD 1 RT 90. Before the researcher asked them to find the pattern in the processes, Alekat20 immediately replied, "I know a quick way to do that". She wrote down "repeat 4 [fd 1 rt 90]". R2D2 and Victor agreed and seemed to understand the commands.

The researcher then drew a Table (Table 7.2), and began to challenge them about other polygons. The researcher questioned, "We know to use 90 degrees for a square, how about a triangle, pentagon…" The participants seemed to know the names of polygon well. They named all the polygons up to 10 sides and put those names into the Table. R2D2 was very interested in this and he asked, "what about 360 sides?" The researcher then asked to find out by using the REPEAT command, and assigned Alekat20, Victor, and R2D2 to try for a triangle, pentagon, and hexagon respectively.
Table 7.2
Generalisation of the Polygon (2)

<table>
<thead>
<tr>
<th>Sides</th>
<th>Shape</th>
<th>Degrees to turn</th>
<th>Relationship</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Triangle</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Square</td>
<td>90</td>
<td>Repeat 4 [fd 1 rt 90]</td>
</tr>
<tr>
<td>5</td>
<td>Pentagon</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Hexagon</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Septagon</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Octagon</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Nonagon</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Decagon</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Alekat20 tried \texttt{repeat 3 [fd 1 rt 70]}, \texttt{repeat 3 [fd 1 rt 30]}, \texttt{repeat 3 [fd 1 rt 2]}, and then \texttt{repeat 3 [fd 1 rt 120]} to get a triangle. She recorded 120 degrees into the Table for a triangle. Victor predicted 70 degrees for a pentagon. He typed in \(360/5\) and got a feedback "You don't say what to do with 72". Instead of using \texttt{REPEAT} command, he used \texttt{Quick Command} and tried \texttt{PD FD 1 RT 72 FD 1 RT 72 FD 1 RT 72 FD 1 RT 72 FD 1}. He then recorded 72 degrees into the Table for a pentagon. R2D2 typed \texttt{repeat 3}, the researcher remind him that he was to create a hexagon. Hearing that, R2D2 changed the 3 to 6 right away. He then formulated commands \texttt{cs pd repeat 6 [fd 1 rt 70] pu} and tried 100, 10, 20, 30, 50, and finally 60 degrees. He was very happy and recorded 60 into the Table.

The researcher assigned new polygons for the participants: septagon for Alekat20, octagon for R2D2, and nonagon for Victor. Alekat20 tried \texttt{repeat 7 [fd 1 rt 200]} and \texttt{repeat 7 [fd 1 rt 50]}. She claimed that she had done a septagon. However, when she navigated closer to see, she changed the degrees to 52, 51 and was puzzle about the overlapping sides and the gap. R2D2 tried 50, 40 degrees for an octagon. After seeing the result in 3D virtual space, he thought and said, "Must be 40 something". And soon he got 45. He went to help Alekat20 and they got 51.5 for a septagon. Victor easily got 40 degrees for a nonagon. He had actually found the relationship between the sides and degrees. He quickly did the 11-sided and 13-sided polygons to further confirm his finding.
Alekat20 kept trying 53, 50, 50.5, 51.3, 51.4, and claimed 51.4 to be the degrees for a septagon despite a small gap could be seen between the starting and ending sides. Victor calculated and suggested Alekat20 to use 51.42857142857143, which she tried to successfully create a septagon.

The research then called for a group discussion to find out the relationship between the angle to turn and the number of side of polygons. Victor was very excited because he was the only one so far to find the relationship. He told everyone to use 360 divided by the number of sides to get the degree. Alekat20 seemed to figure out and said, "Oh, 'cause it's going right around, and the 360 degrees is like the whole shape". They wanted to post their findings in the Forum. But before that, the researcher questioned how about 360 sides? R2D2 answered, "It is 1 degree". The researcher told them, it would certainly be a polygon, but what will it look like? At the same time, Alekat20 had done the 360-sides polygon and replied "a circle".

Victor posted his finding in the Programming Forum titled “Shape 2D”:

1. you divide 360 by the number by the sides on the 2-D shape you are doing
2. you go onto the quick command
3. you type in the anwser into the degree box
4. then you just start to fd your anwser then you do it or rt your anwser
5. keep ondoing it until it has no gaps
p.s I came up with the idea first

Victor posted his finding in the Programming Forum titled “Shape 2D”:

1. you divide 360 by the number by the sides on the 2-D shape you are doing
2. you go onto the quick command
3. you type in the anwser into the degree box
4. then you just start to fd your anwser then you do it or rt your anwser
5. keep ondoing it until it has no gaps
p.s I came up with the idea first

Alekat20 also posted her analysis titled "COOL SHAPE FACTS!!!!" in the 3D Virtual Space Forum to mark the end of this session.

Alekat20 also posted her analysis titled "COOL SHAPE FACTS!!!!" in the 3D Virtual Space Forum to mark the end of this session.

Today we Learnt the formula for making shapes...
It Is...... [number of sides] divided by 360= the number of deegrees
Here are some facts
s d
3 120
4 90
5 72
6 60
7 51.42857142857143
8 45
9 40
10 36
We also found that the number of sides on a shape and the number of deegrees on
In this activity, the six participants successfully investigated the formula of polygons in VRMath. This confirmed that the instructional strategies and scaffolds from Iteration 1 were useful for learning. However, unlike in Iteration 1 with Year 6-7 students, this formula of polygon problem was successfully solved by Year 4-5 students. It would be interesting to know if younger children could do this activity within the VRMath environment, and if so, how young it could be?

It was noted that the concepts about angles and decimal numbers were difficult for the participants especially the Year 4 students Bonbon and Rosco. In the pre-interview (see Section 5.3), it was noted that Rosco didn't know what a right angle was. This was confirmed here in this activity when the researcher questioned about the angles to turn for a triangle (see the transcription for Figure 7.23), in which he seemed to think the 90 degrees right angle as 45 degrees. Furthermore, the many digits after the decimal point such as 51.42857142857143 (i.e., 360/7) were found to puzzle these young students. From the results, it was observed that these Year 4-5 students, when encountered such a decimal number, would first try to use the whole number (i.e., 51). When conflict was generated from the VR space (navigated closer to see the septagon), they would then try half more which is 51.5. This was a good strategy to guess and narrow down the possibility. However, it also showed a lack of understanding about rounding and estimating the decimal number. Fortunately, the VRMath's 3D space was able to precisely visualise the decimal numbers and the programming language allowed the use of mathematical operations in the commands.

One thing special about this activity was the idea of investigating 2D lines and shapes in a 3D virtual space. Traditionally, the investigation of 2D lines and shapes was done on a 2D plane such as a paper and 2D Logo environment. It may be sufficient and easier to do 2D lines and shapes in 2D environment. However, it was found that VRMath's 3D environment could offer and stimulate new thinking about
2D lines and shapes. For example, as discussed above, the 3D virtual space could provide high precision visualisation to stimulate the investigation about angles. If this was on a 2D paper or pixel-based 2D microworld, it would not be possible to visualise the decimal numbers. Moreover, when taking a different perspective (viewpoint) in 3D rather than the normal top view in 2D environments, learners would have the opportunity and be able to recognise that a 2D square was a square even though it didn't look like a square from different 3D perspectives. This was observed when the participants constructed the polygons from the original viewpoint. They were a little bit confused about the shapes at the beginning. After multiple examinations in 3D space, they built up confidence and could recognise polygons from different viewpoints.

The ability to construct and recognise 2D shapes in 3D space was related to the PE6, in which different viewpoints or perspectives within VRMath influence students' conceptions about geometrical objects. When constructing a square from original viewpoint (not top view) using Quick Command, the participants tended to think that they were at the top view and think of the 3D space as was a 2D plane on the screen. This was evident from Grae and Bonbon's attempts to draw a square using Quick Command. They both forwarded some metres at the beginning and tended to draw a big square because the FORWARD 1 looked just like forwarding a little bit from the original viewpoint. When turned 90 degrees, they forwarded less because it looked long enough. This is illustrated in Figure 7.26 below.

![Figure 7.26 Drawing of a 2D square in 3D virtual space](image)

It seemed that the 3D perspective had generated a wrong perception for the participants. Using Quick Command to draw a square became difficult if the perspective was not top view. And from a perspective other than top view, the
participants seemed to lose their awareness about the distance (length) walked. This mis-perception however, disappeared when they navigated to examine the 2D shapes.

The egocentric bug (Fay & Mayer, 1988) was also found when the participants were using Quick Command to draw shapes. This was evident in Bonbon's triangle and Grae's square drawings. When the turtle had been turned 180 degrees to face them, they used BK command for forward instead of FD command. It was observed that because of the Quick Command interface had fixed icons for FD and BK, and RT and LT, therefore, they only worked when user's viewpoint followed the turtle's viewpoint. It was thought that if user's viewpoint could automatically follow the turtle's viewpoint, the egocentric bug could have been avoided. Or as had to be pointed out to the participants that when using Quick Command, we should always take the turtle's perspective to move or turn (when using egocentric frame of reference).

School K participants seemed to have better understanding about polygon, angle and REPEAT command. Little instruction was given, but they could easily determine the pattern to repeat and the relationship between degrees to turn and the number of sides. This reflected their high achievement in the mathematics competition as identified in pre-interview. Victor was the first one to find the relationship between degrees and number of sides. Normally he was not interested in posting in the forum. This time, however, he was motivated by his own finding and posted a good message with, "I came up with this idea first".

The utilisation of social-actional resources was observed. In addition to posting or replying to messages, it was noted that the participants were managing their private messages and editing their posts in the Forum. This was brought to the researcher's attention because the Forum system wouldn't keep a history record of those deleted or modified messages. It was thought that it may be valuable to keep all messages in order to better analyse the learning and social interactions.

The researcher was concerned about deep knowledge construction. All participants were able to use the formula in REPEAT command. They were able to see the relationship between the number of sides and degrees to turn. However, it was thought that the properties of polygons could have been further investigated. For example, with further explorations, the participants might have associated 72 degrees
with pentagon without noticing that the 72 degrees was the exterior angle not the interior angle of the pentagon. Discussion and exploration about exterior and interior angles, and decimal numbers should be included in future iterations.

7.6 LEARNING ACTIVITY 8: A FRAME OF A CUBE

Following the construction of 2D shapes in 3D space in Learning Activity 7, this learning activity challenged the participants to create a frame of a cube, a 3D structure in 3D space. To create a frame of a cube, the participants were free to choose which turning and moving commands to use. The six turning commands (i.e., RT, LT, RU, RD, TL, and TR) and FD and BK commands have an egocentric frame of reference, which means the turning and moving are referenced from the turtle's perspective. The other six moving commands EAST, WEST, NORTH, SOUTH, UP, and DOWN have a fixed frame of reference in six fixed directions. In addition, the writing of procedures and project management were introduced in this activity.

7.6.1 Results

The researcher first asked School C participants to use the Quick Command to walk a frame of a cube. When started, an efficient way of using VRMath was observed from the three participants. It was noted that the participants adjusted the frame sizes of the Forum and VR space. They also changed PenDown colour, background colour or panorama from the Background Chooser, cleared the screen, restored viewpoint, and moved the Quick Command dialogue's position on the screen and so forth. Very quickly, they had set up the environment ready to use Quick Command to “walk” a cube.

Rosco started with turning 30 degrees. After seeing the turtle turning in the 3D virtual space, he immediately changed to 90 in the Quick Command. He was the first one to create a frame of a cube. A mixture of egocentric and fixed frames of reference commands was utilised (see Figure 7.27).

```
PD FD 1 RU 30 RU 30 UP 1 reset setrgb 0 718 1000
FD 1 BK 1 PD FD 1 LT 30 LT 30 FD 1 LT 30 FD 1 PE
PE RT 30 RT 30 RT 30 LT 90 FD 1 LT 90 BK 1 FD 1
PE PE FD 1 RT 90 LT 90 LT 90 FD 1 UP 1 FD 1 PE
RT 0 FD 1 PE LT 90 LT 90 FD 1 LT 90 FD 1 LT 90
FD 1 LT 90 FD 1 FD 1 PE LT 90 FD 1 DN 1 LT 90 FD
1 UP 1 LT 90 FD 1 DN 1
```

Figure 7.27 Rosco's cube by Quick Command
Rosco didn't attempt to navigate around the shape he had created to inspect it; he knew that it was a cube. Bonbon had a look at Rosco's cube and said, "No". But when Rosco navigated around the shape that he had created, Bonbon realised that it was a cube. At the same time, Grae said, "I did it". He had created a colourful frame of a cube. Similar to Rosco, he had utilised both egocentric and fixed frames of reference commands as shown in Figure 7.28.

![Grae's cube by Quick Command](image)

**Figure 7.28 Grae's cube by Quick Command**

Bonbon also drew a 3D cubic shape. She said to the researcher, "I think I draw a cube". It was noted that Bonbon didn't navigate in 3D space while she was drawing. The 3D structure she had created did look like a cube from the original viewpoint. However, when she navigated around the VR space to examine the object she had created, she found out that it was a rectangular box (see Figure 7.29).

![Bonbon's cube by Quick Command](image)

**Figure 7.29 Bonbon's cube by Quick Command**

Next, the researcher called the three participants together and introduced them to writing a procedure in VRMath's Logo-like programming language. The researcher demonstrated an example of a pentagon procedure, which created the new command PENTAGON once the procedure was executed and defined.
TO *pentagon*

`pd repeat 5 [fd 1 rt 72] pu`  
END

The researcher then tried the new command *PENTAGON* to show the participants. Unexpectedly, the command *PENTAGON* didn't create a pentagon as was predicted. The researcher carefully examined and found that there were actually five points drawn in the 3D virtual space. The researcher realised that it was because Rosco had clicked the POINT command in Quick Command earlier. Thus, by chance, the researcher introduced the POINT, LINE, and FACE commands to the students. The effect of the three commands on the pentagon procedure is illustrated in Figure 7.30.

![Figure 7.30 Drawing of a 2D square in 3D virtual space](image)

As the participants appeared to understand the procedure well, the researcher decided to introduce the syntax of parameter into a procedure. The pentagon procedure was then modified as below:

```
TO *pentagon* :side
  pd repeat :side [fd 1 rt 360/:side] pu
END
```

The participants tried typing in *pentagon* but got a "Not enough inputs" feedback. The researcher explained that the PENTAGON command now needs an input as does the FD command. Then the researcher typed in *pentagon 6*, which resulted in creating a hexagon in the 3D virtual space. The researcher continued to try *pentagon 7* and *pentagon 12* to create a heptagon and a dodecagon. Bonbon asked, "Why don't you change the name for the procedure?" Rosco and Grae wanted to try 360 sides, which they did and wowed when they saw a circle was created by command *pentagon 360*.

Next, the researcher asked the participants to create their own procedure for a cube. It was observed that all participants could correctly operate the Procedure
Editor. However, it seemed not easy to start writing. Grae had syntactic problems with the procedure, in which he wrote “TO cube 10”. The researcher then suggested him to use "TO cube10" because space was not allowed in the procedure names. It was noted that Rosco used "RT 1" in his procedure. Although he had changed the degree to 90 in Quick Command and seen the "RT 90" command frequently, he still seemed confused about the parameter for moving and turning commands. Bonbon used her hands to simulate the turtle. She moved and turned her hands while composing her procedure. The participants were now using their mental imagery or visualisation ability to compose commands in their procedure. However, despite having a few tries, the participants still seemed to have difficulties in “walking” a cube mentally in their mind.

The researcher gave them a hint to use REPEAT command, and that a square might be a good start. The participants then started to include the REPEAT command into their procedure. Bonbon was the first to complete the procedure. Grae and Rosco still had difficulties at completing the cube. The researcher then helped Grae to think and try it out step by step. Finally he completed the cube with actually four squares. Rosco was unable to think three dimensionally. Bonbon then helped him to create a procedure similar to hers. The three procedures are presented in Table 7.3.

Table 7.3
Procedures of a Cube by School C Participants

<table>
<thead>
<tr>
<th></th>
<th>Bonbon</th>
<th>Grae</th>
<th>Rosco</th>
</tr>
</thead>
<tbody>
<tr>
<td>TO Cubonbon</td>
<td>pd</td>
<td>repeat 4 [fd 1 rt 90]</td>
<td>TO cube10</td>
</tr>
<tr>
<td></td>
<td>repeat 4 [fd 1 rt 90]</td>
<td>ru 90</td>
<td>pd</td>
</tr>
<tr>
<td>up 1</td>
<td>repeat 4 [fd 1 rt 90]</td>
<td>repeat 4 [fd 1 rt 90]</td>
<td>repeat 4 [fd 1 rt 90]</td>
</tr>
<tr>
<td>fd 1</td>
<td></td>
<td>fd 1 rd 90</td>
<td>fd 1</td>
</tr>
<tr>
<td>DN 1</td>
<td></td>
<td></td>
<td>repeat 4 [fd 1 rt 90]</td>
</tr>
<tr>
<td>east 1</td>
<td></td>
<td>fd 1 rd 90</td>
<td>fd 1</td>
</tr>
<tr>
<td>up 1</td>
<td></td>
<td></td>
<td>repeat 4 [fd 1 rt 90]</td>
</tr>
<tr>
<td>rt 360/2</td>
<td></td>
<td>pu</td>
<td>up 1</td>
</tr>
<tr>
<td>fd 1</td>
<td></td>
<td>END</td>
<td>fd 1</td>
</tr>
<tr>
<td>dn 1</td>
<td></td>
<td></td>
<td>dn 1</td>
</tr>
<tr>
<td>nc</td>
<td></td>
<td></td>
<td>rt 90</td>
</tr>
<tr>
<td>pu</td>
<td></td>
<td></td>
<td>fd 1</td>
</tr>
<tr>
<td>END</td>
<td></td>
<td></td>
<td>up 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>nc</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>pu</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>END</td>
</tr>
</tbody>
</table>
School K participants went through similar processes as the School C participants. They began with using Quick Command and the fluent operation of VRMath was also observed. Victor utilised the Show Grid and took a side view, which formed an isometric kind of drawing. He also turned the turtle into a jet plane (see Turtle Dance Activity in Section 6.4) before drawing the cube. It was noted that the three participants didn't change the degree to 90. However, they could just click three times on a turning icon to get 90 degrees as the default degree is 30. R2D2 was the first one to complete a cube using Quick Command. It was observed that he didn't change viewpoint at all when constructing the cube; he used both egocentric and fixed frames of reference commands. He called the researcher and navigated to show the cube he created. He then found that he had one less click on RD for the last line of the cube (see Figure 7.31).

![Figure 7.31 R2D2's cube by Quick Command](image)

R2D2 restarted the construction by clearing the screen. He drew a cube very quickly by using only the fixed frames of reference commands (i.e., EAST, WEST, NORTH, SOUTH, UP, and DOWN). The turtle didn't make any turn at all. Alekat20 liked to change viewpoint when constructing the cube. She also utilised both egocentric and fixed frames of reference commands. When she had completed a square, she navigated to the top view but the turtle appeared flipped. Because of the turtle was now 180 degrees rotated, she then did some wrong clicks on UP as DN and DN as UP (i.e., manifested the egocentric bug). Victor also liked to navigate when constructing. However, instead of making a cube, he tried to draw a pyramid. It was noted that he mostly only used egocentric frames of reference commands. The angles within the pyramid were very difficult for Victor to figure out. What he could do was just to get a visual feeling from the 3D virtual space. In the end, he did make a pyramid but lines were not perfectly connected at the vertices. During his
construction, he also took BK as FD because of the icon BK in Quick Command (see Figure 7.32).

![Quick Command](image)

**Figure 7.32 Victor's pyramid by Quick Command**

After the researcher introduced the Procedure Editor and the syntax (format only) for procedure, the participants were asked to write a procedure for a cube. Different to School C, the researcher showed a real frame of a cube and a small turtle toy for School K participants to practice before writing the procedures. They practiced on the wireframed cube with commanding the turtle. This seemed to aid the mental thinking relating to a cube. After a few trials and errors, the participants had each completed a procedure for a cube. However, due to the network problem, the database server was disconnected from the programming interface. Therefore, their procedures needed to be posted into the Forum instead of saved as projects. Their procedures are presented in Table 7.4 below.

<table>
<thead>
<tr>
<th>Table 7.4</th>
<th>Procedures of a Cube by School K Participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>R2D2</td>
<td>Alekat20</td>
</tr>
<tr>
<td>TO mycube2</td>
<td>TO mycube</td>
</tr>
<tr>
<td>pd</td>
<td>pd</td>
</tr>
<tr>
<td>repeat 4 [fd 1 rt 90]</td>
<td>square ru 90 fd 1 rd 90</td>
</tr>
<tr>
<td>ru 90 fd 1 rd 90</td>
<td>square</td>
</tr>
<tr>
<td>repeat 4 [fd 1 rt 90]</td>
<td>rt 90</td>
</tr>
<tr>
<td>repeat 2 [fd 1 rd 90 fd 1 ru 180 fd 1 rd 90 rt 90]</td>
<td>fd 1</td>
</tr>
<tr>
<td>pu</td>
<td>lt 90</td>
</tr>
<tr>
<td>END</td>
<td>square</td>
</tr>
<tr>
<td></td>
<td>fd 1</td>
</tr>
<tr>
<td></td>
<td>rd 90</td>
</tr>
<tr>
<td></td>
<td>square</td>
</tr>
<tr>
<td></td>
<td>pu</td>
</tr>
<tr>
<td></td>
<td>END</td>
</tr>
</tbody>
</table>

TO square
repeat 4 [fd 1 rt 90]
END
In the beginning of next session, School C participants reviewed their's and School K participants' procedures. The researcher showed them the real cube wireframe and the small turtle toy as in School K. Rosco experienced trouble with the cube procedure last session. When the researcher asked him to command the toy turtle to walk the wireframed cube, he seemed to understand better about the moving and turning in 3D space. Both Rosco and Bonbon used the same strategy to construct a cube, which was first to walk the top and the bottom squares, and then use UP and DOWN to connect the two squares. Grae did well in last session with VRMath. But when using the real cube, he seemed to show a different kind of 3D difficulty. He gave the commands and manipulated the toy turtle. At the same time, he also rotated the wireframed cube. This confused him because he forgot where he had walked.

The researcher then tried to show the effects of different frames of reference commands. The researcher questioned, "What if the turtle begins with a \textit{tilt 45} orientation? Will you still get a cube from your procedure?" Bonbon thought and replied, "It will be a tilt cube". The researcher then opened each participant's procedure and tested the procedures with TL 45 orientation. These effects are illustrated in Figure 7.33.

![Figure 7.33 The effects of different frames of reference on cubes](image.png)

Seeing this, Rosco thought that it was cool but weird. Bonbon thought and said, "I think it's the UP and DOWN thing..." Rosco and Bonbon then modified their procedures to get rid of the EAST, UP, and DOWN commands, and saved to new projects.

School K participants also tried their procedures with TL 45 orientation. However, because they didn't use any fixed frames of reference, the effect of using two different frames of reference was not seen. When the participants tried their procedures of cube from TL 45 orientation, the cubes were all the same to Grae's tilted cube as shown in Figure 5.47 above. From their procedures, the researcher
identified two ideas from the participants and asked them to incorporate these in their future projects. The first idea was about the use of procedures in procedures as Alekat20 did in her “mycube” procedure (see Table 7.4). The second idea is to think of the turtle’s position and orientation at the end the cube. This could be utilised to create certain patterns of 3D structures. For example, this occurred when Alekat20 was asked by the researcher to repeat her “mycube” procedure six times. This created the pattern of six cubes illustrated in Figure 7.34

![Image of six cubes](image1)

*Figure 7.34 Alekat20's six cubes*

Alekat20 opened Victor's cube procedure. Victor wasn't very happy about that. He kept asking how to open others' projects and how to prevent others from opening his projects. The researcher had talked about this before but explained again to him that it was because of the "public" option when saving projects. Meanwhile, Alekat20 had done some modifications on Victor's procedure. She said, "I have made your procedure better". Figure 7.35 shows a pattern of some cubes created by the modified procedure.

![Image of repeated cubes](image2)

*Figure 7.35 Repeated cubes of Victor's procedure*

It was noted that Victor soon changed his projects to private because he didn't want to share his programming codes unless he felt the codes were perfect. Later on, the researcher advised them to look at School C participants' projects, and showed them the effect of different frames of reference commands. The participants said,
"Cool". They also inquired about other VRMath users such as anna1 and Hipeoples and tried projects created from those users.

After they reviewed each other’s procedures, Alekat20 posted in the Forum titled “Why ???”:

why is every ones cubes the same ?

_________________

Can You Talk To Me???????????? Please!!!!!
Can You Make the Turtles Body A Different Colour??????????

Victor replied:

because it just is

_________________

victor

R2D2 also replied:

mine is different (only problem is that it is so long)

_________________

MKS

R2D2 later found that there was an error in his procedure “mycube2” (see Table 7.4). It should be repeat 3 instead of repeat 2 in his procedure. He then corrected the procedure and saved to a project named “cubef”. It was noted that R2D2 also created a project named “dud”, a procedure that utilised only fixed frames of reference commands to construct a cube.

TO dud
pd
north 1 west 1 south 1 east 1 up 1
north 1 west 1 south 1 east 1 north 1 dn 1
west 1 up 1 south 1 dn 1
pu
END

He wrote in the project description explaining that:

try this out it only uses n s e w and up and down
ps:dud stands for (direction & up & down)

He also posted “dud ‘n’ cubef” in the Forum, showing and inviting people to try his procedures.

please go into public project and try either dud or cubef (i recommend dud)

_________________

MKS
Near the end of this session, a new user named "James" was registered and started to join the Forum discussion and sent private messages to all participants except Rosco. Soon Alekat20 and R2D2 found that James was created by Victor. The researcher didn't query Victor about why he created a new user. But something could be certain was that he felt better to participate if he could remain anonymous.

7.6.2 Discussion

This activity enabled many PEs (Perspectives for Evaluation) to be addressed. With respect to PE1, the participants’ operationalisations of spatial abilities were observed when they navigated to examine the cubes (spatial orientation) and command the turtle to move and turn (spatial visualisation). It was found that generally there were two types of behaviours utilised when constructing a frame of a cube using Quick Command. The first was constructing with 3D navigation and the second was constructing without 3D navigation. In this activity, it was found that only Bonbon from School C and R2D2 from School K didn’t navigate around the object when constructing using Quick Command. The other participants mostly navigated in 3D space during their construction of 3D cubes. One problem about navigating that emerged when constructing was that the Quick Command window would disappear (sent to back) when clicking into 3D virtual space to navigate. However, it was observed that these participants were not bothered by this. They just quickly recalled the Quick Command window from the Tools menu.

Bonbon and R2D2 didn’t navigate (i.e., utilise spatial orientation) while using Quick Command to construct a cube. They only manipulated the turtle (utilising spatial visualisation) and viewed from the original viewpoint. This resulted in the creation of incorrect cubes as shown in Figure 7.29 and Figure 7.31. However, Bonbon’s problem was that she drew the cube based on the visualisation from the original viewpoint. She drew a 2 by 1 by 1 box, which looked like a cube from the original viewpoint, disregarding what constituted a cube or the properties of a cube. This indicated that she didn’t think three dimensionally and she didn’t understand or did not think about the properties of a cube (e.g., all edges are the same length). R2D2, in contrast, just missed out the final line because of the viewpoint did not show a proper rolldown of 90 degrees. However, his understanding about a cube was

---

25 PE1 is about the evaluation of how students’ spatial abilities change and develop within the 3D VRLE?
generally accurate. Bonbon and R2D2's examples illustrated that different viewpoints or perspectives within the VRLE did influence their conceptions about 3D shapes (i.e., Perspective for Evaluation 6 or PE6\(^{26}\)). VRMath revealed Bonbon’s misunderstanding about the congruency of all edges of a cube. The clear implication that emerges from this is that the construction of 3D shapes from only one viewpoint can lead to incorrect perspective drawings. Therefore, the operationalisation of both spatial visualisation (turning and moving the turtle) and spatial orientation (changing viewpoints) are recommended when constructing 3D shapes within the VRMath environment.

Interestingly, Rosco and Grae utilised Quick Command to successfully draw a cube but they both were struggling in transferring the commands into a procedure. Bonbon used the Quick Command and drew an incorrect cube. However, she did the best in writing the procedure. When using Quick Command, the participants could get immediate feedback of 3D visualisation in VR interface. By contrast, when writing the cube procedure, the participants had to mentally visualise or trace the turtle's position and orientation. Obviously, writing the cube procedure required a higher level of operation of spatial abilities. Bonbon did well in writing the cube procedure showed her good spatial abilities and 3D visual memory. This might be attributed to her strategy of using hand gestures in space. Rosco and Grae didn't have any strategy when doing the 3D cube. And without Quick Command and 3D virtual space, there was no immediate visual feedback. Therefore, when encountered with writing procedure, it was difficult for Rosco and Grae to think and visualise the 3D cube and the turtle in their mind. Their difficulty in writing the cube procedure indicated a lack of visual memory to maintain the turtle’s position and orientation in their mind. The implications from this vignette are that (a) the real time 3D visualisation (with navigation) could aid the construction of 3D artefact (However, the 2D perspective of 3D space (without navigation) could impede the development of 3D thinking); (b) the writing of procedures (link to abstract language) could serve as a diagnostic tool to identify student's capacity of visual memory; and (c) it might pose a gender issue on spatial ability such as boys are better in utilising visual feedback while girls are better in using language description for future research.

---

\(^{26}\) PE6 evaluates how the students' ability to explore 3D objects from different viewpoints or perspectives within the VRLE influence their conceptions about 3D geometrical objects?
Bonbon and Rosco's procedures were different from the other participants' procedures in terms of 3D rotation. The two procedures created by Bonbon and Rosco only involved turning in one dimension, while the other participants' procedures involved turning at least in two dimensions. In fact, Bonbon and Rosco only used one turning command (RT) in their procedures. They first created the bottom square using the commands `repeat 4 [fd 1 rt 90]` learnt from the polygon activity. Then they used `up 1` and then created the top square. The rest of three edges were then drawn by using UP and DN. This strategy was simple and efficient for creating a cube but might not be enough for other 3D shapes such as a tilted cube or a pyramid. Moreover, this strategy actually reflected human nature, in which we predominantly turn on one plane (dimension) only. It was thought that the use of only RT or LT command might indicate the participants' lack of experience in tilting (TL and TR) and rolling (RU and RD). Therefore, for some participants, there was a tendency to use the turning in one dimension only. This partly addressed Perspective for Evaluation 8 (PE8\(^{27}\)), in which the lack of kinaesthetic experience in tilting and rolling affected the use of commands in VRMath environment.

Victor, R2D2, and Alekat20 generally had no problem with creating a frame of a cube. They successfully created a frame of a cube by using Quick Command and the writing of a procedure. It was thought that they went smoothly in this activity partly because a real frame of a cube and a toy turtle were available. The use of a real wireframed cube and a toy turtle provided a visual image and helped clarify what to do in this activity. However, when Grae rotated the cube and moved and turned the cube and toy turtle together, he was unable to remember where the turtle had walked on the wireframed cube. This had some implications. Firstly, the real wireframed cube had limitation as it couldn't show the processes of turtle track as did in the 3D virtual space. Secondly, Grae rotated the wireframed cube because he wanted to maintain a viewpoint following the turtle so he could probably avoid the egocentric bug. Thirdly, this indicated that it might be a good teaching strategy to fix the wireframed cube and just change the viewpoint by moving one’s own body around if necessary. Lastly, the manipulation of both the wireframed cube and the turtle at the same time was putting both spatial orientation and visualisation abilities in action. This was a higher level of spatial ability and thinking, which researchers should be

\(^{27}\) PE8 is about how children’s lack of kinaesthetic experience in tilting and rolling within a 3D environment affects their ability to operate with these movements within the VRLE?
careful about when the novice learners are trying to operate with both spatial abilities.

The egocentric bug was manifested by all the participants but only when they were using Quick Command. When writing procedures, there was no egocentric bug as the participants were naturally thinking the 3D movements from the turtle’s perspective. The egocentric bug is a common misconception that can happen even with adults with well developed 3D abilities. The researcher considered that the egocentric bug as a unique usability issue that only applied on this Logo-Turtle graphics environment. It may be utilised to contrast users’ thinking and raise the users’ awareness about thinking from the turtle’s perspective. However, if the VRMath environment is to be user friendly, this usability issue with respect to the egocentric bug needs to be addressed. As was mentioned in the previous activity (formula of polygon), a camera (animated viewpoint) can be designed to follow the turtle’s viewpoint. This is similar to the Avartar view but the turtle should still be visible. This feature will be implemented in next iteration to try to eliminate the egocentric bug.

It was observed that the fixed frames of reference commands aided the construction of a cube. This was evident in participants’ use of Quick Command, where they all used some fixed frames of reference commands (e.g., UP, DN, EAST, WEST, NORTH, and SOUTH). Rosco even utilised only the six fixed frames of reference commands to construct a cube in the shortest time. The use of egocentric frames of reference commands was more difficult and was the source that led to the emergence of the egocentric bug. This finding again confirmed what Yakimanskaya et al. (1991) found that the predominant use of some one particular frame of reference (most often the human body) often impedes successful problem solving particularly in descriptive geometry, and the suggestion that it is necessary to use several frames of reference simultaneously.

It was interesting to find that the School K participants had no fixed frames of reference commands in their procedures (see Table 7.4) despite using some fixed frames of reference commands when using Quick Command. It was conjectured that the use of only egocentric frames of reference commands was because of their practice on the wireframed cube with the toy turtle. Furthermore, they had only used LT, RT, RU, RD, and FD commands. No TL, TR, and BK were used in their
procedures. This might indicate the participants’ preferences and reflect that tilting and going backward are less performed in real life. The researcher considered that all egocentric frames of reference commands could contribute to the development of 3D spatial abilities. However, users would naturally use only certain commands. It was thought that in the future, some activities should be specifically designed to engage the use of certain commands. For example, the construction of different 3D shapes such as pyramid may engage the use of tilt commands. Or perhaps a direct instruction to limit the users to use certain commands can be integrated into activities.

The last point about the use of different frames of reference commands was that this activity was set to investigate how the participants use different frames of reference intentionally. However, within the limited time (about two hours) of this activity, it was difficult to achieve this goal. The students had no chance to try and explore further the use of different frames of reference commands. The researcher has taken a note that more time should be allowed and a specific context (e.g., build some 3D artefact) should be given in future iterations to investigate users' intentional uses of different frame of reference commands.

With respect to the syntax of VRMath’s Logo procedure, all participants except Grae were able to follow the syntax (e.g., TO procedure_name command_list END) easily. It seemed that the syntax of a procedure didn’t make sense to him. When introducing procedure, the researcher didn’t focus on the syntactic knowledge such as keywords (e.g., TO and END) and format (e.g., space between words, and space is not allowed in a procedure’s name). Instead, the researcher just showed an example of a procedure, and focused on the meaning of the procedure (e.g., the procedure contains many commands in sequence and the procedure will become a new command). The semantics of TO and END marked the start and end of a procedure. The TO keyword also implied that there was something after it (e.g., procedure name and parameter). The syntax of Logo procedure was succinct. However, it might not be explicit enough for some users. This led the researcher to a thought about the adoption of keywords (or reserved words) in different programming languages. For example, Pascal is one of the first procedural programming languages. It has an easy to understand structure for its procedures as:
procedure procedure_name (arguments);
var var_name : VAR_TYPE;
begin
commands
end;

In Grae’s case, it was not until more syntactic and semantic information had been given to him that he could then write procedures correctly. This brought to the researcher's attention how a novice user’s perceptions of the language (a type of typological semiotics) in programming can influence the user’s ability to construct a geometrical artefact (PE2).

In School C only, the researcher introduced the use of parameter in procedures. This was an incidental instruction, which happened because of a timely appropriateness of the group dynamics in School C. The researcher just carried on this "parameter in procedure" instruction after the introduction of the PENTAGON procedure. The syntax of using parameter (TO procedure_name :para_name) was new to the participants, which included a colon before the parameter as a mandatory syntactic rule. The participants saw the following pentagon procedure created by the researcher. They could see how the parameter ":side" was used in the procedure as the researcher asked for their attention on the parameter. However, they couldn't think of how to use the new command until the researcher demonstrated the command "pentagon 6".

TO pentagon :side
pd repeat :side [fd 1 rt 360/:side] pu
END

The parameter in programming is similar to a variable, which is also related to the important concept of "unknown" in algebra. Considering the young age (9 and 10 years old) of these participants, this might have been their very first experience with variables -- an abstract and typological thinking notion. The researcher was confident that these young children were able to learn about procedures and parameters when they could see the visualisation of procedures in the 3D virtual space. This was evident from their use of the procedure with parameter after the researcher's instructions. In using the PENTAGON procedure with a parameter, Bonbon also found that the name "pentagon" was not adequate for the procedure. Later on, the pentagon procedure was renamed to "polygon" by the participants. The
naming of procedures was considered as a valuable abductive thinking (see Section 2.2.1.2), in which learners invent signs to make sense of their new experience. Although "polygon" was not something invented by these participants, it was truly a new sign to them at that time. Similarly, R2D2 also invented a "dud" procedure to represent his new experience in using only fixed frame of reference commands to create a cube. This indicated that VRMath environment had the potential to encourage abductive thinking.

The last discussion about procedure was the use of a procedure in another procedure. For School K participants, the researcher gave an example of SQUARE procedure (i.e., TO square repeat 4 [fd 1 rt 90] END). When constructing a cube procedure, only Alekat20 utilised the square procedure in her cube procedure. In computer programming, it is common to write a procedure or sub routine for reuse in other part of the programming codes. However, this idea wasn't easy for novice users to think of. The researcher didn't plan to use this idea in this activity either. Alekat20's use of the square procedure in the cube procedure indicated her different ways of thinking about geometry. She has noticed that a cube consisted of some squares rather than just some lines to walk through. And by this chance, the researcher pointed out her idea to other participants as preparation for the next project activity.

With respect to the Forum activity, it was noted that School C participants didn't contribute any posts during this activity. As usual, the researcher still allowed some time and reminded the participants to interact in the Forum. However, School C participants seemed to turn their interest to the programming interface after the introduction of procedures and projects. They spent much time on creating procedures and saving their projects. They also found how to open others' projects and try to recreate some 3D virtual worlds in the 3D virtual space.

In contrast, School K participants always enjoyed using the Forum. In this activity, they were still keen to post the results of their learning in the Forum. For example, Alekat20 posted a question asking why everyone's procedure was similar. This questioning had the potential to stimulate knowledge-building discourse. Unfortunately, it only generated responses from School K participants. R2D2 used the Forum to advertise his procedures. Victor also used the Forum to show off his programming codes. He, however, took a different approach to publicise the
ingenuity of his procedure - he posted reply to his own post! When the researcher questioned him on-line within the Forum about why he had done this, he created a new user to anonymously post reply to himself and other participants. The researcher reflected on Victor's behaviour and realised that there was a social need in the Forum that should be catered. That is, users liked to receive feedback, comments or praise on his/her own posts. And sometimes users would like to be anonymous to feel secure and comfortable in the online community.

Lastly with respect to usability issues, the project management was found easy to use by the participants. The interface was fairly intuitive; the participants were able to use it without needing much instruction. When saving projects, the project description was utilised by the participants. This helped the researcher toanalyse the participants' projects. For example, the "dud" procedure created by R2D2 would be meaningless if R2D2 didn't put comments "dud stands for (direction & up & down)" in the project description. While the design of project description worked well, a drawback in the data collection was found by the researcher. When analysing participants' procedures, the researcher would like to see the participants' debugging of their procedures. However, the current VRMath system could only show the final procedures saved in projects. It would better facilitate the analysis of users' learning if the processes of debugging of procedures could be recorded.

7.7 SUMMARY

This chapter reported on the enactment and evaluation of the specific concepts type of learning activities, which included learning Activities 4-8.

Learning Activity 4: Shape and scale introduced the five 3D primitive shapes, and a set of scaling commands to investigate the concepts of projective shapes, similarities, and transformations. The major findings were:

1. About five hours of interaction with VRMath was needed for these Year 4 and 5 participants to effectively operate within the VRMath environment.

2. The 3D shapes and scaling commands enabled creative construction of 3D objects. The participants started to view and construct real world objects from scaled 3D primitive objects. These activities facilitated the transfer between visual and language, and between VP and IFI spatial abilities.
3. The VRMath environment enabled the investigation of 2D projective shapes from 3D objects by scaling one dimension to zero, similarity from scaling all three dimensions to produce enlarged/reduced objects.

**Learning Activity 5: Climb up stairs** utilised previously learnt commands to build a staircase. The major findings were:

1. The staircase activity facilitated the development of visual imagery when specifying commands and of visual memory when the participants were able to visualise the hidden part of the 3D structure.

2. To scaffold the use of REPEAT command, the participants should be informed that more than one command can be repeated in one REPEAT command, and be encouraged to observe and find the patterns of the real world structure (e.g., a real staircase) before forming the command.

3. It is important to link the virtual with reality. A virtual staircase with half metre height between each of the stairs may look like a real staircase yet doesn't make sense to a real world staircase.

4. The VRMath environment encouraged new ways of thinking and doing 3D geometry such as scaling two dimensions to zero, and creative DNA structures by using REPEAT command.

**Learning Activity 6: 3D rotation** developed the concept of non-commutativity of 3D rotations. The major findings were:

1. The investigation of non-commutativity in 3D rotation was limited by using body movement to see and feel. The inaccuracy in body movements and intuition often led to misunderstanding about 3D rotation.

2. The VRMath environment enabled multiple ways for the participants to solve the 3D rotation problem. These included the turning of the turtle in 3D virtual space, viewing from the turtle’s perspective in Avatar mode, and creating 3D objects after 3D rotation.

3. Two misconceptions were identified. The first was the false analogy of using 2D movement for 3D rotation. The second was that a series of 3D rotation could be conteracted by the opposite rotation in same order.
Learning Activity 7: Formula of polygon investigated a mathematical formula for 2D regular polygons. The major findings were:

1. Scaffolds such as a table for recording results and finding patterns and relationships were found to be effective in this learning activity. With the scaffolds and the VRMath environment, this learning activity could be learnt by children as young as Year 4.

2. VRMath provided the opportunity to investigate decimal numbers, which the Year 4-5 participants experienced through 3D navigation to see the gap in the 2D polygons.

3. The VRMath’s 3D environment allowed multiple perspectives for viewing 2D shapes. However, some 3D viewpoints may present a wrong perception of the 2D shapes if the 3D viewpoints are static (see Figure 7.26).

Learning Activity 8: A frame of a cube investigated the properties of a 3D cube by using Quick Command and writing procedures to draw a frame of a cube. The major findings were:

1. The use of both egocentric and fixed frames of reference commands aided the construction of a 3D cube.

2. The use of Quick Command to construct 3D objects was likely to cause the egocentric bug. An animated camera that follows the turtle’s viewpoint may be able to prevent the egocentric bug.

3. Quick Command provided an immediate visual feedback when constructing 3D objects, while writing a procedure forced the participants to think three dimensionally and thus developed the participants’ mental spatial ability.

4. Static 3D viewpoints (i.e., using Quick Command without navigation) could cause a mis-perception about a 3D cube (see Figure 7.29), which also indicated that the participant was not aware of the properties of a cube (i.e., all edges are equally long).

5. The common strategy to create a cube was to draw the top and bottom squares, and then connect the two squares by using UP and DOWN
commands. This reflected our nature of moving on a 2D plane, and a lack of tilting and rolling in 3D space.

6. The effect of using different frames of reference commands (see Figure 7.34) could be understood by the young participants.

7. Writing a cube procedure could be difficult if the participants operated both spatial visualisation and orientation abilities together.

8. The naming of projects and procedures encouraged abductive thinking, in which the participants invented signs to make sense of their new ideas.

9. The participants’ debugging processes in procedures are important for justifying their learning. This will be implemented in the next prototype VRMath.
CHAPTER 8

ITERATION 2: APPLICATION LEARNING ACTIVITY

8.1 OVERVIEW

Chapter 8 reports on the enactment and evaluation of the application type of learning activities. This type of learning activity investigated how the participants utilised their geometrical understanding and the facilities provided by the VRMath system to collaborately construct a virtual reality microworld. Learning Activity 9 was the only learning activity in this category.

**Learning Activity 9: Final project** required each of the two groups of participants to collaboratively build a 3D virtual world. The main focus of the final project activity included a holistic evaluation of the learning of 3D geometry, the collaboration within and between groups, and the usability issue about the design of the VRMath environment to facilitate the collaborative construction of 3D virtual worlds.

Through group discussions, School C participants decided to construct a house with some indoors and outdoors items. School K participants decided to construct a temple with a stage, some cylindric structures, and trees surrounding the temple. After six hours of interactions with group members and VRMath, the two final projects were successfully completed and recorded into online database.

8.2 LEARNING ACTIVITY 9: FINAL PROJECT

The final project activity required each of the two groups of participants to build a virtual world and save it as a project in VRMath. It was expected that the participants would utilise the skills (e.g., 3D navigation, programming commands and procedures, and communication in forums) learnt from previous activities to collaboratively build a virtual world with group members. Therefore, this activity would probably enable the observation of all Perspectives for Evaluation (PEs). However, because the researcher didn't prescribe any particular virtual world for the participants to build, the actual applicable PEs would depend on how the group
progressed during this activity. In addition, the usability about how the participants collaborate within group in utilising VRMath environment was inspected.

8.2.1 Results

School C participants started this activity. In trying to engender ideas for the group project, the researcher asked the participants to open and try some public projects in VRMath system. All three participants had tried most of public projects in VRMath's 3D space during previous activities. However, in this activity, the researcher encouraged them to examine the procedures rather than just to recreate some 3D virtual worlds. Among those public projects examined by the participants were Polyprism, Street Lights, Symmetric tree, Recursive random tree, and Random 3D Tree (see Appendix 7).

The three participants together examined the procedures. This was done to reinforce their knowledge about the syntactic structures (e.g., TO, END, and parameters) and conventions (e.g., the name "init" as initial procedure) subsumed within the programming language. Bonbon asked about jump commands (e.g., JB and JF) in the tree procedures. She also asked about the difference between pen colour and material settings. These were explained by the researcher. After the explanation, the participants seemed to understand each individual command. They enjoyed using "init" command to recreate the 3D artefacts. They called out, "Cool" when they saw those 3D virtual worlds recreated. However, the logic flow of the whole programming codes seemed to be too complex for them to comprehend.

Next, the researcher focused on a post in the Programming Forum named "Random ball" to begin a discussion.

```
  TO randomball :number
  repeat :number [
    ; set random position
    setxyz (random 20)-10 (random 20)-10 (random 20)-10
    ; set random diffuse color
    setdc (random 1000) 0 0
    ; set random transparency
    settr (random 1000)
    ; create a ball
    ball
  ]
END
```
From the randomball procedure, some programming features and techniques were discussed. These included the use of RANDOM command, semicolon as comment mark, set commands for material (e.g., SETDC and SETTR) and location (e.g., SETXYZ), and the use of “Copy” button in Material Editor (see Figure 3.17). The Cartesian coordinate and colouring model (i.e., red, green, and blue) were briefly mentioned. The researcher then asked Bonbon to copy the randomball procedure from the Forum to a Procedure Editor. After the procedure was executed and defined, Bonbon typed in `randomball 50` and was amazed by the semi-transparent reddish random balls randomly created in the 3D virtual space (see Figure 8.1).

![Figure 8.1 50 random balls in space](image)

After seeing the random balls, Grae and Rosco went back to their computers to try out the procedure. Without the researcher’s help, both Grae and Rosco created the randomball procedure. Grae kept using Up and Enter key to quickly send many `randomball 50` commands into the programming interface. He enjoyed watching the random balls filling into the 20 by 20 by 20 metres space. In the meantime, Rosco excitedly called the researcher over to see his different random balls. He accidentally created 50 random balls with lines connecting them (see Figure 8.2) because he had inserted the PD command before the `randomball 50` command. The lines connecting each ball showed the random effect even more clearly. Rosco actually navigated to check the sequence of balls by the connecting lines.

![Figure 8.2 50 random balls connected with lines in space](image)
Meanwhile, Bonbon had worked on a new project named "Wierd World!!!(sic.)". She merged the "Cubonbon" procedure into the project, and created the "randomball" and a new procedure named "WierdWorld" (sic.). When writing the "WierdWorld" procedure, she typed in:

```
TO Wierd World
Command list …
END
```

With a space between “Weird” and “World” in the procedure, she got an error message "Error while running: Parameter name must be preceded by a colon" when she executed the procedure. This was corrected after the space between Wierd and World was removed. After completion, the researcher was very impressed because she had utilised most programming features and techniques discussed earlier in the WierdWorld procedure and the project (see Appendix 8). The weird world turned out to be the kind of scientific-fictional virtual world when she changed to dark background and switched on the grid (see Figure 8.3).

![Figure 8.3 A weird virtual world by Bonbon](image)

After seeing and examining those existing projects, the researcher began the discussion for their final project. The researcher first proposed "Our School" to be the project for both schools.

*Researcher: What do we have in our school? Pod (A big building comprises four classrooms)?*  
*Bonbon: Yeah.*  
*Researcher: So we have a procedure called "Pod". (The researcher wrote down TO Pod END). Then you have to use cube, penup, pendown, face, line, whatever to draw a pod. If you move the turtle to somewhere and say pod, you will get a pod here. But you have to measure how big your school is then you move the turtle…*  
*Rosco: Ok so what we have to do is…*  
*Researcher: move the turtle and get a pod here, and get another pod there. And then you can probably draw some roads…*  
*Bonbon: Do we have to do our school?
Researcher: No. And some random trees here and there, but how can you put them together? You can create another procedure, the one I used is init, then I put all the commands I have in it to create pods, roads, and trees. Can you understand this?

The researcher here tried to scaffold the use of procedures in procedure, and to separate procedures for their future cooperative work.

Researcher: So you may have some procedures, pod, road, and tree... (The researcher wrote on a paper).
Rosco: And then you put all of them together...
Researcher: In this init procedure or any procedure you name it. 
Bonbon: Ok, so you just press ah... we will call it ccc or something for our school, and then you put ccc here (pointing to the command centre).
Researcher: Good. So if you save the project, next time you use VRMath, you only need to type in ccc command and everything will be recreated.
Bonbon: Cool.
Researcher: Ok, so what do we do? If you think our school is not a good topic, do you want to do a scientific or fictional virtual world?
Rosco: Like this? (pointing to the randomball virtual world on his screen)
Bonbon: You should see mine, you should see mine, mine is gonna look really mad, ok cs wierdworld go... (she typed in her computer).
Researcher: Or maybe you can create a floor here, and another floor here, and another floor here. Then you use your spiral staircase to connect them, and you can actually walk on it...
Bonbon: How do you setwait? (She wanted to speed up the weirdworld)
Researcher: (Look at the Weird World) So what project is that?
Rosco: This is weird world (He had opened the Weird World project in his computer)

It seemed that the participants didn't like the topic of "our school". However, the discussion seems to have generated some ideas. The researcher kept asking what they would like to do in their project. Finally, Grae came up with an idea to do a house. Everybody agreed. Bonbon started to name things in the house such as a bed, sofa, and chair. Rosco also suggested a TV. Bonbon replied that, "TV is easy though, all it is a box with two cylinders poking on the board on the top". The researcher then gave the participants homework, which was to draw the plan of the house (2D top view) with some measurement in metres. Rosco was already busy building a house using some 3D shapes brought in by the researcher. The participants were quite motivated now as Bonbon said that the turtle could live in the house, and Rosco added that the turtle could sit on the sofa and watch TV. Grae had been very quiet listening. Bonbon thought an open door was needed to walk into the house. She and Rosco also thought about windows and asked how to do a hole on a wall. The researcher then drew and explained that more faces were needed for the wall (see Figure 8.4).
Figure 8.4 A window on a wall

The initial idea about School C's project had formed. The researcher appointed Grae as the team leader to report their project in the Forum. Grae then posted in the Show Room Forum titled "project of house":

Hi all,
We have chosen a topic for our final project.
Project name: A House.
Team member: grae, bonbon, rosco
Steps:
1. Each of us will draw picture of the house and we will look at the drawing and we will put the thing that we agree on the house.
2. We will generate a list of items that we want to have in or out the house.
Items such as windows, pool, spa, bedroom door.
to be continued.....

School K participants began by reading the new posts in the Forum. When Victor was checking the member list in the Forum, he suddenly asked, "Who is Ken?" The researcher replied, "Ken is a Uni lecturer in UK". Victor continued to ask about other users such as "saehn" and "test" in VRMath Forum. The researcher knew that it would be invaluable for the participants to communicate with knowledgeable members such as "ken" and "test" (a senior member in the Web3D and VRML community), but he didn’t encourage the participants to spend time on it.

When told that School C participants had decided what to do in their project, Alekat20 soon found Grae's post in the Show Room Forum and said out loud, "They are making a house!" She posted immediately to "CONGRATZ" School C's commencement of the house project. However, she posted into a different Forum: 3D Virtual Space. After reading in the Forum, the participants started to discuss their project.

*Researcher: You can have your own projects. But I require you to do a project together.*
Victor: I have an idea. Why don't we just do an airport? We have the airplanes, we can make a jet, and then we have trees in the airport... Or we can make a road, with streetlights on it...
Researcher: Streetlights? That's my project.
Victor: We can make cars then.
R2D2: I can try to make a car, I can do pretty good.
Researcher: Yes. But before that, how can we get started?
Victor: We have to use the resources...
R2D2: By looking at others' projects?
Victor: We have to use the resources through... The airport might be a good decision because we have a plane that would work... and then a road might be good since it has the streetlights.
Researcher: Ok, good, an airport, then you turn the turtle into a jet plane...
Victor: Then you are on the road.
Researcher: So you can command the plane to forward, forward, forward, forward, then up a little bit, up a little bit, and up a little bit. Then it looks like an airplane flying at the airport.
R2D2: I want to do my own car, and she wants build a balcony.

It seemed that the participants had different ideas about their project. The researcher then suggested them to check out some public projects in VRMath. The three participants gathered at one computer with R2D2 as the computer operator. The first project examined was the randomball in the Forum. In a manner similar to that employed with School C participants, the researcher explained that a parameter or argument (i.e., :number) was needed for the randomball procedure. The number was used as the number of repeat. Therefore, `randomball 100` would create 100 balls. When asked what is random, R2D2 replied, "difficult?" and Victor said, "you don't know how much it is". The researcher then asked R2D2 to try `commands print random 5`, and the first result returned was 3. The researcher questioned, "random 5 is 3, is that correct?" Alekat20 was puzzled and said, "What do you mean?" The researcher then asked R2D2 to use Up and Enter keys to get more random number printed. A sequence of random number, "1, 3, 2, 2, 1, 1, 4, 4, 4, 1, 0" was generated. The researcher continued to question, "Is it possible to get a 5?" R2D2 tried a few more `print random 5` and didn't get any 5 printed. The researcher then concluded that, "random 5 will get a number from 0 to 4 but you don't know which number you will get because it is random". Victor tried `random 1` (always returns 0), `random 1000` (returns from 0 to 999), and `random 0` (always returns 0). R2D2 said, "Cool, so we can play like Bingo on that, cool!" From the randomball procedure, SETXYZ and material commands were introduced.

Victor was asked to be the computer operator for examining the Recursive Tree project (see Appendix 7). Instead of opening the project, the researcher asked
Victor to locate the project's procedure in the Forum. Victor followed the instructions and showed good computer operating skills in copying the procedure from the Forum to a Procedure Editor. The researcher explained that the tree procedure has itself in its procedure, which makes it recursive. The idea of recursive procedure seemed to generate no interest in the participants. However, they enjoyed it when the turtle was constructing the symmetrical and random trees. From the tree procedure, jump commands, and CM and METER commands were reviewed.

Alekat20 was asked to be the computer operator for examining the polyprism procedure (see Appendix 7). Alekat20 quickly copied the procedure from the Forum into a Procedure Editor. The polyprism procedure required three parameters as side, distance, and height, which were expressed in metres. Previously for the tree procedure, the unit had been changed to centimetre by CM command. Alekat20 typed in "METRE" again (see Learning Activity 3: Turtle dance in Section 6.4) to change the system unit, but the command "METRE" was not recognised. She soon realised why and typed in "METER" instead. To use the polyprism procedure, Alekat20 formulated a command polyprism 11 .5 7. When she typed in the command, she was puzzled to get the "I don't know how to polyprism" feedback. This was because she didn't execute to define the polyprism procedure in the Procedure Editor. After she executed the polyprism procedure, she finally created an 11-sided polygon based prism. From the polyprism procedure, FACE, LINE, POINT, and NC (NextColour) were reviewed, and PCON (pen colour on) and PCOFF (means to use material for objects) were introduced.

After examining those projects and procedures, the researcher encouraged the participants to review them at anytime, and if applicable, copy and use them in their projects. Victor asked everyone about the group project. He still wanted to build an airport and some roads with street lights. R2D2 still thought about a car for the project. Victor argued that cars were too difficult to make. He proposed a new idea to make a turtle, but no one was interested. Alekat20 also expressed that she was not interested in cars; she proposed making a chess set with the built-in Chess Board. R2D2 came up with an idea to make a garden. Victor suddenly agreed but he wanted to change it slightly to a greenhouse. The researcher asked Alekat20 if she would like to join the greenhouse project. Unfortunatetly, Alekat20 objected. Alekat20 had been
thinking quietly. She proposed, "I want to do a Temple" because she saw the Stonehenge picture on her computer desktop (see Figure 8.5).

![Figure 8.5 Stonehenge for Temple project](image)

Victor showed great interest about the Temple idea. He talked about Egyptian Temples having some cylinders arranged as a rectangle on a square box. And suddenly he realised and corrected that he was talking about the Greek's Temple. Finally, the participants had agreed to do a Temple together for their group project. The researcher asked the participants to draw a Temple on a paper. R2D2 drew one quickly and showed everyone (see Figure 8.6).

![Figure 8.6 R2D2's Temple drawing](image)

From Figure 8.6, the researcher suggested that a scaled, squashed box with REPEAT command could be utilised to create such a Temple. Meanwhile, Alekat20 began to create the Stonehenge structure utilising the polyprism procedure, but it didn't work out. She explained that the turtle kept repeating forward, right turn, and back. Unfortunately, she didn't save the procedure into a project. Therefore, the researcher was unable to analyse her procedure. Victor asked about how big the Temple would be. The researcher then asked him to draw the ground plan with measurements on it. Victor didn't want to draw. He replied, "10 squares?" The researcher then asked the participants to estimate the size of the room and justify
how big the Temple should be. After estimation, R2D2 said that 20 by 20 metres should be enough.

It was observed that the participants started to use language and commands such as FORWARD, RT, and UP to describe their Temple. They also asked questions such as how to change background and use smilies, and answered each other’s questions. When asked to nominate a team leader, the three participants all wanted to be the leader. To avoid argument, the researcher appointed R2D2 to be the team leader and asked him to post about their project. R2D2 then posted "project temple" in the Show Room Forum,

hi we've also chosen our project it is of course a temple (like it said in title).
These are our jobs:
  victor: he's doing the roof and other items in and out of the temple and he's also
doing the floor!
  alekat20: she's doing the border around the temple with some decorations as well!
  r2d2: i'm doing the body of the temple maybe with some stairs!
you'll be hearing more about this later(next time) so see ya!

School C participants found School K's Temple project in the Forum. They started to make drawings and discuss a list of items for their house project. The researcher suggested the participants to think about both indoors and outdoors items, and find items that appeared many times so it would be worthwhile to write a procedure and reuse it. Rosco wanted to use the streetlights procedure for creating many streetlights. Bonbon named many items she wanted to do including two sofas, a cooktop, and a sink. Rosco said that the sink was difficult to make. He suggested making a TV set and a garage. The researcher gave Bonbon some hints for the cooktop, which were to use scale commands to resize the box and cylinder, and four small flattened cylinders could be placed on top of the box to be the hotplates. The researcher reminded the participants that the turtle has to be moved to the right location to place or create objects. Bonbon seemed to understand and said, "Ok!"
When the participants were talking about a door and windows on the walls, the researcher suggested to them to make a transparent face for the window. Rosco immediately said that the glass material could be used. Bonbon also mentioned the PCOFF command, which was correct if the glass material was to be applied on a 2D face object. Grae drew his house plan showing that he wanted a pool, four trees and
a fence for the house. The 40 by 40 ground was the size of the built-in Grid. The researcher suggested the participants build the house within and upon the 40 by 40 grid. With the built-in Grid, the participants felt more comfortable to start with. Bonbon and Rosco also drew their design plans. The three plans are shown in Figure 8.7.

![Figure 8.7 Grae, Bonbon, and Rosco's house plans](image)

The researcher asked if the participants could identify separate tasks for individuals to work on. Because of the discussion and the plan each participant drew, the individual tasks were soon identified. Grae, as the team leader, posted in the Show Room Forum titled "Project of house - cont 1".

```
today we have decided what we are doing
Grae is doing the out side of the house.
Rosco is doing the house body
Bonbon is doing the in door items
to be continued !!!!!
```

School C participants then began the construction of their house. During the process of construction, it was observed that the participants engaged in much geometrical thinking. They talked and acted using the turtle commands, and they were involved in thinking about shapes and line, and direction, location and movement. The construction of 3D objects was a long process involving the debugging in procedures and visualising and examining the objects they had created in 3D virtual space. The following paragraphs report on each participant's construction.

**Bonbon's indoor items**

Bonbon wanted to construct many indoor items. However, she only managed
to make a cooktop, a bed, and a table during this learning activity. The first item she constructed was the cooktop. Before she started, Bonbon talked about her idea for the cooktop. She said, "I was thinking that it would be easier. All you have to do is just do the cube, and then on top of the cube just a flats … flat cube, you know that's scale height 0, and with a different colour, and then cylinder zero, right, forward, cylinder zero…” It was observed that Bonbon started to experiment on each part (e.g., body and hot plates) of the cooktop in a cooktop procedure following what she had told the researcher previously. This process involved multiple times in copying material settings from Material Editor, changing scales, and moving and turning the turtle in the procedure. It was observed that Bonbon had a debugging cycle of: (a) writing initial codes, (b) executing and defining procedures, (c) creating in 3D virtual space, (d) navigating and examining 3D virtual objects, and (e) modifying codes. The code changing was not recorded. However, her codes consisting of cooktop, bed, and household items procedures were recorded in her project titled "For the house project" (see Appendix 8).

Bonbon was very happy about the cooktop and the bed she created. However, the size of the bed was too small when compared with the cooktop. The cooktop was about one metre cube, while the bed was only 1.2 meter wide, 0.5 metre deep, and 0.2 metre high (i.e., `scale 1.2 0.5 0.2 box`). The researcher then helped modify the bed procedure to include a scale parameter (see Appendix 9). To do so, the researcher showed Bonbon to add a parameter into the bed procedure as:

```
TO bed :s
  Command list ...
END
```

And in the command list, the scale command and all moving commands such as FD and UP were multiplied by `:s`. After modification, the bed procedure was tested by commands `bed 1`, `bed 2`, and `bed 3`, which created three proportional beds (see Figure 8.8).

![Figure 8.8 Three proportional beds](image)
Bonbon followed the same instruction and modified the cooktop procedure. She found that the :s parameter also worked for the cooktop procedure (see Appendix 9). After some trials, Bonbon decided to use cooktop 1 and bed 3 in the final house project, which created a cooktop about 1 cubic metre and a bed about 3.6 (width) by 1.5 (depth) by 0.6 (height) cubic metre in size. In the final house project, a table procedure created by Bonbon was included (see Appendix 9). The table was made of four scaled cylinders and one scaled box. Its dimension was 2.5 (width) by 1.5 (depth) by 1 (height). Figure 8.9 illustrates these three household items in their final size and colour.

![Bonbon's household items](image)

*Figure 8.9 Bonbon's household items*

**Rosco's house body**

Rosco started his house body in a procedure named house. He first tried to build walls of the house.

```plaintext
TO house
  cs pd face
  repeat 4 [fd 10 rt 90]
  up 12
  repeat 4 [fd 10 rt 90]
  fd 10 dn 12 rt 90 up 12 fd 10 dn 12 rt 90 fd 10 up 12
  pu
END
```

When executed and specified the command *house*, Rosco got the feedback "You can't change pen type when pen is down!" from the programming console. This was because the *face* command was given after the *pd* command. The researcher instructed Rosco to move the *face* command to before the *pd* command. Rosco followed the instruction but soon he called for help because the faces (walls of the house) were unpredictably created (Figure 8.10).
The researcher explained that for every face (wall) created, a jump command was needed to tell the turtle the end of the face and to begin a new face (wall). In the house procedure above, the researcher instructed Rosco to change the up 12 between the two repeat commands to ju 12 (JumpUp). After changing to ju 12, the house procedure could create the top and bottom faces, but the rest of commands still acted oddly. The researcher asked why up 12? Rosco replied that he wanted to build a house with two or three stories. Because of the limited time for this learning activity, the researcher suggested Rosco to just build a flat house, and discussed a detailed plan with him (see Figure 8.11).

From the detailed plan above, Rusco was then able to reason out the commands. The researcher suggested that three wall procedures could be enough to build the house body. The first wall procedure was the front one containing the door, which consisted of 3 faces. The second wall procedure was the side wall containing one window, which could be reused for both side walls. The third wall procedure
was for the back wall. It only needed to build half of the wall as the back wall has a pattern repeated twice.

Rosco gradually wrote wall_1 procedure for the side wall, wall_2 procedure for the front wall and door, and wall_3 procedure for the back wall. A procedure named hbody was created to call those wall procedures. It was a long process for Rosco to write those procedures. He carefully calculated the distance to move and direction to turn the turtle. The difficult part observed was when to use a jump command. At times, Rosco was frustrated by the wrong distance calculated and was puzzled by unexpected faces created due to the absence of a jump command. The researcher suggested to him to slow down the turtle (i.e., SETWAIT command) for easy debugging. Copying the material settings from Material Editor was found easy for Rosco. Lastly, a roof procedure was created by Rosco using a scaled cone. Again, Rosco tried a few scales and moved the turtle to the right position to place the cone as the roof of the house.

**Grae's outdoor items**

Grae was interested in making a fence utilising the polyprism procedure. He copied the polyprism procedure from the Forum into a procedure named fence,

```latex
to fence :side :length :height
  repeat :side [fd :length ru 90 fd :height ru 90
    fd :length ru 90 fd :height ru 90
    jf :length rt 360/:side]
end
```

and started to experiment different parameters. Grae first tried `fence 100 0.5 1.5`, and with the Grid turned on, he created a fence on the right side of the Grid (see Figure 8.12).

![Figure 8.12 Grae's fence experiment 1](image)

Grae wanted the fence to be bigger and centred in the Grid. Therefore, he moved the turtle to `bk 15 west 15` location to start the fence with `fence 100 1 1.5` command. The researcher added a NC command at the end of the repeat's square
bracket (i.e., repeat […… nc]) to make the fence colourful. The fence resulted as shown in Figure 8.13.

![Figure 8.13 Grae's fence experiment 2](image)

The researcher suggested Grae to start the fence from home west 20 location. To get the right size of the fence, Grae also manipulated the parameters such as fence 120 1 3 and fence 200 1 1.5. The researcher reminded him to consider the dimension of a real fence. Finally, Grae was happy with fence 40 3 1.6, and with starting location at home west 20. This fence, however, was still not centred within the Grid (see Figure 8.14).

![Figure 8.14 Grae's fence experiment 3](image)

Next, Grae wrote a ground procedure, which created a green coloured, 40 by 40 metre squared face to replace the Grid. When he followed the plan (see Figure 4.61) for the pool, Grae was puzzled by two problems. The first problem was that the pool was a rectangle. The command repeat 4 [fd .. rt ..] didn't work for a rectangle. This was solved when the researcher pointed out to him that the width and length together could be repeated twice. Therefore, the command repeat 2 [fd 12 rt 90 fd 5 rt 90] was formulated to build a face as the pool. The second problem was that when the pool face was created on the same plane (i.e., y=0) as the ground, the two faces would overlap and flash from different viewpoints. This was solved by lifting the pool face up 0.01 to avoid overlapping. Lastly, Grae utilised the 3D Random Tree procedure, and created four trees beside the pool (see Figure 8.15). The trees,
although created by other's procedure, took Grae about 20 minutes to arrange next to
the pool.

![Figure 8.15 Grae's pool and trees](image)

School K participants also drew their plan for their part of the Temple (see
Figure 8.16). The following paragraphs report on each participant's construction.

![Figure 8.16 R2D2, Victor and Alekat20's Temple plans](image)

**R2D2's Temple part 1**

R2D2 started with a procedure named body (see Appendix 8). He wanted a
five steps stage and a kiosk on top of the stage as shown in Figure 8.16. He knew
that the structure he planned could be done by scaled boxes, cylinders, and a cone.
He then started with the construction of the stage using scaled boxes. The researcher
reminded him that in order to be able to walk up the stage, the high of each step
should be a reasonable height as the real stairs. R2D2 then tried some scales and
decided to start with \textit{scale 15 0.8 6}. The 0.8 in the scale command was the height,
which was 80 centimetres. Because this was the first stair, half of the stair was under
the ground. Therefore it was actually 40 centimetres as the height of each stair.
Figure 8.17 illustrated this with the Grid and Sand Ground.
The turtle was at the centre of the box. R2D2 experimented and wrote `up 0.4 fd 0.25 scaled 5.5 box` for the second stair. Following this pattern, he wrote

`up 0.4 fd 0.25 scaled 5.5 box`
`up 0.4 fd 0.25 scaled 5 box`
`up 0.4 fd 0.25 scaled 4.5 box`

into the body procedure. As can be seen, each stair was decreased by 0.5 metre in depth. The researcher introduced him with the `REPCOUNT` command, which returns the repeat count in `REPEAT` command. Therefore, from the second to the fifth stair, the four stairs were created by command:

```
repeat 4[up 0.4 fd 0.25 scaled 6-reepcount*0.5 box]
```

The construction of the kiosk was experimented many times with scaled cylinders and a cone as the roof. It was observed that R2D2 changed the scales according to how it looked in the 3D space. At times, he wasn't consciously aware of the turtle's position and orientation. Therefore, the researcher had to remind him to keep in mind of the turtle's position and orientation, and to move the turtle to the centre of where the 3D object was to be placed. For example, if the height of a cylinder was 5 metres (`scaleh 5`), the turtle should then `up 2.5` in order to place the whole cylinder above on the ground or floor.

After the completion of the body procedure, R2D2 built a fountain (see Figure 8.18) without any help from the researcher. The fountain procedure is included in Appendix 8.

```
Figure 8.17 R2D2's Temple stage
```

```
Figure 8.18 R2D2's Temple fountain
```
R2D2 explained that the eight blue thin cylinders were the water flowing down from the ball at the top. He was very proud of the ball on the top, which were two spheres placed at same location but with different orientations and materials. He accidentally found this special effect from the 3D rotation learning activity, where Grae placed two spheres at same location but with different orientations (see Section 7.4: Learning Activity 6 and Figure 7.16).

**Victor's Temple part 2**

Victor was responsible for the Temple ground and some trees. He didn't want the ground to be too simple. Therefore, Victor thought of a moat with four bridges around the Temple (see Figure 8.16 above). The Temple ground was then similar to Rosco's window, which was consisted of many faces. However, in contrast to Rosco's use of jump commands, each face of the Temple ground was constructed within a pair of PD and PU commands. From his plan, Victor calculated the distances and commanded the turtle to walk out the ground in a procedure named ground (see Appendix 8). The ground comprised five rectangular faces and a lower face as the water in the moat. A bridge procedure, which creates a rectangular face as a bridge, was also created by Victor. To successfully place the bridge at right position, Victor had carefully calculated the distance to move and turn the turtle. The calculation didn't bother Victor much as he started from home position for each of the four bridges. The ground and the bridges are illustrated in Figure 8.19.

![Figure 8.19 Victor's Temple ground without/with bridges](image)

Next, Victor wanted to have some trees. However, instead of using the tree procedure in some public projects, he decided to take the researcher's suggestion using a scaled cylinder and a scaled cone for a tree. He first wrote a tree procedure with proper scales for the cylinder and the cone, which creates a simple tree (see Figure 8.20).
Similar to Bonbon's bed procedure, the researcher introduced a scale parameter for Victor's tree procedure. The researcher told Victor the principle of adding a scale parameter was that every scale and move commands in the procedure had to multiply by the scale parameter. The tree procedure thus was modified as:

```
TO tree :s
  scale 0.2*:s 2*:s 0.2*:s
  ; some material settings here
  up 1*:s cylinder
  scale 1.5*:s 2.2*:s 1.5*:s
  ; some material settings here
  up 1.1*:s cone
  dn 2.1*:s
END
```

The modified tree procedure required an input or a parameter. Victor experimented with some inputs such as `tree 1`, `tree 2`, and `tree 3`, and verified the results in 3D virtual space. He then came up with an idea to use the RANDOM command for the input. He tried command `tree random 3` a few times. Because `random 3` returned a number from 0 to 2, when it returned 0 there would be no tree at all. The researcher then introduced a common programming strategy to translate the returned number from 1 to 3 by using command `tree (random 3)+1`. Victor tried the command for a couple of times. He was happy with the random trees in different sizes. However, he found that `tree 3` was too big. He wanted trees that were scaled between 1 and 2. The researcher then helped Victor to form the command `tree (((random 10)+1)/10)+1`, in which the random number generated would range from 1.1 to 2.0. Victor copied the command and used it in his procedure. However, he expressed that he still didn't fully understand about the command even after the researcher's explanation.
Alekat20 proposed the Temple project from the picture of the Stonehenge. Therefore, she chose to do some 3D structure like the Stonehenge. The strategy she incorporated was to utilise the polygon formula `repeat :side [fd 1 rt 360/:side]`. However, instead of using PD to draw lines, cylinders were created when the turtle moved to each vertex of the polygon. Alekat20 experimented with some scales for the cylinders, different numbers of sides, and distance to forward. She eventually decided to create 18 cylinders in her temple base procedure (see Appendix 8). The distance between two adjacent cylinders was 2.2 (i.e., `repeat 18 [cylinder fd 2.2 rt 360/18]`). The reason for why 18 and 2.2 was because that the structure created by the two numbers could fit well onto the Temple ground (see Figure 8.21).

![Alekat20's temple base on Temple ground](image1)

*Figure 8.21* Alekat20's temple base on Temple ground

The next step Alekat20 did was to place a scaled box on top of every two cylinders. Because there were 18 cylinders, she knew that there would be 9 scaled boxed. She then wrote the commands `repeat 9 [fd 3.3 rt 360/9 box]` but found that 3.3 was not the right distance to place the scaled box. Alekat20 experimented with many different distances to try to move the turtle to the middle of two cylinders. However, none of the distances worked. The researcher then reminded her to think about the pattern to repeat. Finally, Alekat20 figured out the pattern and wrote `repeat 9 [fd 1.1 box fd 1.1 rt 360/18 fd 2.2 rt 360/18]` to successfully solve the problem (see Figure 8.22).

![Alekat20's final temple base](image2)

*Figure 8.22* Alekat20's final temple base
So far each participant had completed their individual tasks for the group project. The next task was to combine all the procedures of each participant into the final group project. To do this, the participants experienced similar problems. Rosco first encountered the confusion of his latest project. It was found that Rosco saved projects using the same name. When asked to open his latest project, Rosco had to open the projects with the same name and check the latest edited time. The integration of procedures from different projects into a new project was also problematic. The first problem was that some of the participant's projects were marked as private, which meant that only the creator had access to the project. To solve this problem, the researcher had to ask the participants to mark all their projects in this learning activity as private except the latest one as public. The second problem was that there was no easy way to merge procedures from different projects. To merge procedures, the participants had to: (a) open the first participant's project and save as a new project (group project), (b) open the second participant's project and copy the procedures, (c) open the group project, paste the procedures and save the group project, (d) open the third participant's project and copy the procedures, and (e) open the group project again, paste the procedures and save the group project.

After integration of procedures, some individual tasks could not create 3D objects correctly. This was because some procedures had changed the scale and the turtle's location and direction. Therefore, minor editing was needed for each procedure to define scale (e.g., \texttt{scale 1 1 1}) and starting location and direction (e.g., \texttt{home}) at the beginning of each procedure. The researcher requested the participants to create a procedure as the final command to create the 3D virtual world. School C participants then used a procedure named hbody (Rosco's procedure) while School K participants used a VAM (the acronym of the three participants' names) procedure to include all other procedures. The final group projects were saved as "Ninja's house" for School C, and "VAM Temple" for School K (see Figure 8.23).

After completion of the group projects, the participants enjoyed much navigating in both virtual worlds. Both groups presented their projects in the Show Room Forum. They gave compliments such as "Your house is really cool !!!!! I Like it" and "da vam temp was very cool" to each other. Rosco was interested in the Temple trees and questioned in the Forum, "How did u get da twees?" However, due
to the time constraint, the participants didn't examine other's procedures, nor did they reply to the questions about their projects in the Forum.

![Ninja's house](image1.png) ![VAM Temple](image2.png)

**Figure 8.23 Final group projects**

### 8.2.2 Discussion

In this learning activity, the process of doing a project by the participants was a dynamic one, in which the teaching and learning depended on the contextual and concurrent interactions, and contingencies among the participants and the researcher. As a preliminary study of student's project in VRMath, this learning activity had implications for the instructional strategies for future student's project, as well as for the learning that occurred and the system's usability.

With respect to instructional strategies, four stages of instructions were utilised. The four stages were examining existing projects, forming topic for the project, implementing the project, and publishing the project.

1. Examining existing projects: The purposes of this stage were to engender ideas for possible projects, and to reinforce the structure of project and procedure. It was found that by examining exemplary projects, the participants could be motivated to learn advanced programming skills to apply to their projects.

2. Forming topic for the project: A student-centred approach could be employed to discuss topics for their project. Students were responsible and in charge of their own deign of the project. Drawings of the project plan and discussion about the geometrical composition of 3D objects with relevant programming language could facilitate the learning and construction of 3D artefacts.
3. Implementing the project: Students collaboratively or cooperatively carried out the writing of procedures. In this learning activity, the project was divided into smaller tasks for individual group members. Individual students only needed to implement a part of the project. This form of cooperative work might have reduced individual work load and accelerated the completion of the project. It, however, minimised the interaction between students. The adoption of the form of collaboration or cooperation needed to be considered and justified depending on the context and complexity of the project.

4. Publishing the project: Students utilised the project management facilities provided in VRMath system to merge procedures from individual's projects. New final group project was marked as public to share with wider users of VRMath. Forums could also be utilised to publish the final group project.

In addition to the four stages above, it was thought that a fifth stage to comment on the other group's project could be included in future learning activities. That is, students could examine and comment on other's thinking in programming, and receive comments from the others to reflect on their own. This would be a different form of collaboration to improve the knowledge construction of 3D geometry and the construction of the 3D virtual world.

With respect to the learning that occurred, many conceptual changes were observed and identified. These conceptual changes could be classified into two categories: programming related and 3D geometry related.

**Programming related conceptual changes**

The syntactic structure of a procedure was reinforced and fully internalised in this learning activity. This was evident that all participants confidently wrote "TO procedure_name END" when they started a new procedure. The only syntactic error occurred was a space in the procedure's name in Bonbon's "Wierd World" (sic.) procedure. This gave an opportunity for the participants to see the feedback from programming console, and for the researcher to introduce the syntax for parameter. The syntax of a procedure was correctly used by the participants as evident in such as Alekat's templebase procedure and Rosco's wall_1, wall_2, and wall_3 procedures.
Bonbon and Victor also utilised the parameter of procedure, in which they created a $:s$ parameter (for scale) and knew that the $:s$ could be used in the procedure as something given after the command. The parameter is a form of variable or unknown, and can contribute to the development of algebraic thinking.

The linear progression of programming logic was further developed in this learning activity. The Logo-like programming language in VRMath, unlike modern programming languages such as Java that has multi-thread of execution line, has only single thread of execution line. A Logo procedure, as it literally means, contains commands that step by step accomplish the procedure's goal. In the construction of 3D virtual objects, the participants wrote down each step of the turtle's movements then followed by scale commands and create commands (e.g., box and cylinder). It was noticed that sometimes the participants didn't choose a material for the 3D shapes when first completing the procedure. However, they could edit the procedure later to insert material commands at the right point before the create commands. The participants' procedural thinking was also evident when they were debugging their procedures, in which they were aware of steps and could identify the error commands in the procedure. The procedural thinking was also displayed in the structure of the project's procedural calls. School K's VAM Temple project, for example, had very succinct procedural calls as shown in the diagram below (see Figure 8.24).

![Figure 8.24 VAM Temple's procedural structure](image)

The concept of reuse of procedures was developed in this learning activity. A procedure could be seen as a pattern to be repeated. This was evident in Rosco's wall_1 procedures, which was used twice, and Victor's tree procedure, which was
used 73 times to create 73 trees. Bonbon also expressed that if she had time to write a chair procedure, she could have 4 or more chairs easily created for the table. The concept of reuse can be emphasised to focus students' attention on identifying the patterns in real world environment.

**3D geometry related conceptual changes**

In this learning activity, the use of programming language was central to the construction of 3D virtual worlds. The successful completion of 3D virtual artefacts in this activity was the result of interaction to both the programming interface and the VR interface. During the activity, the participants were constantly engaged in geometrical thinking by using language (i.e., typological resources) and 3D visualisation (i.e., topological resources). The learning of 3D geometry and the construction of 3D virtual artefacts occurred through a series of semiosis among semiotic resources. This series of semiosis began with the drawings of 2D plans, to linguistic procedures, and then to 3D virtual space. The participants (interpretant) then constantly examined to make sense between signs and objects (e.g., from programming language to 3D virtual objects, and from 3D virtual objects to real world objects). This addressed the PE2 (perspective for evaluation 2), in which the learning about the 3D artefacts relied on the ability to interact between 3D visualisation and procedural language. To make sense, both signs and virtual objects were manipulated. The programming codes were modified and the 3D virtual objects were examined from different viewpoints, until the interpretant visually or mathematically satisfied to equate the 3D virtual objects to real world objects. From the observation and result, the participants had considered both visually and mathematically the 3D virtual objects. However, it seemed that visual sense (visualising from different viewpoints) had dominated and caused potential problems in learning about the 3D virtual objects and real world objects.

For example, when the topics for the group projects were formed, the participants started to think of real world objects from their experience and draw 2D plans. Before the transfer from 2D plans to language procedures, the participants had simplified the real world objects to primitive 3D shapes or 2D faces. The use of primitive 3D shapes to represent real objects required scale command while 2D faces did not. Bonbon's cooktop, for example, consisted of a cube, a scaled box, and four scaled cylinders. In constructing the cooktop, she mathematically manipulated the
scale, move, and turn commands. She then navigated to visually examine the 3D virtual objects created by those commands. And from the visual feedback, she modified the parameters of those commands. In her final codes, Bonbon created a cooktop about 1 cubic metre in size, which was mathematically appropriate when compared to a real cooktop. However, she also created a bed as long as 3.6 meters in length, which was uncommon for a real bed but still visually appropriate as shown in Figure 8.9. This indicated a potential risk in VR environment, where 3D visualisation (from some viewpoints) may mislead the judgement of numerical value. The fundamental problem in VR environment is that, after all, the VR cannot be equated to the real world. A possible solution to this risk may be to just measure the real objects if available, and to maintain the consciousness of being in a simulated VR environment.

It was found that the construction of 2D faces was difficult. This was evident in Rosco's walls of the house, Victor's ground and bridges, and Grae's ground and pool. To create a rectangular face, the turtle has to walk through four points or locations. The programmer has to know exactly how far the turtle will be moving. Rosco couldn't complete the walls' procedure until the researcher had discussed detailed house plan (including measurements) with him (see Figure 8.11). Another difficulty of creating faces was the use of jump commands. The jump commands indicated the end and start of a face. They were designed for the internal language of VR interface (i.e., Virtual Reality Modelling Language). Victor avoided using jump commands by just using PD (PenDown) and PU (PenUp) commands for each 2D face. This may be a good strategy for beginners.

The construction of 3D shapes was easier because the turtle was always at the centre point of the 3D shape. However, when the scale was changed, the distance to move the turtle to the centre point would also change. The way to calculate the centre position was to half the scales. For example, to place a scale 0.2 5 0.2 cylinder on the ground, the turtle needs to go up 2.5. It was found that the participants were able to adjust movement for scaled objects such as Alekat20's cylinders in templebase procedure (see Figure 8.21), and R2D2's cylinders in fountain procedure (see Figure 8.18). It was also found that some participants didn't calculate according to the scales. In VR space, objects could overlap. Therefore, it was possible that the participants ignored the mathematical meaning of scale and movement, and just
relied on visual meaning-making. This was illustrated in Figure 8.25 below with Bonbon’s semi-transparent bed (half of the pillow in bed) and Rosco’s semi-transparent Temple stage.

Figure 8.25 Objects overlapping in VR space

A special characteristic or potential limitation of turtle geometry was identified. When utilising the polyprism procedure to create the fence, Grae found it difficult to put the fence around the centre of the ground (see Figure 8.14). Alekat20 also had the same problem when she tried to place the templebase centred on the fountain. This issue was identified in Abelson and diSessa (1981), in which the turtle geometry was intrinsic, local, and procedural as opposed to extrinsic, global, and equational aspects of coordinate geometry (see Section 2.3.3.2: Computers and 3D geometry). The features of turtle geometry enable easy creation of circular structures such as a fence. However, they make it difficult to define a centre point for the circular structures. It was thought that how to bridge the turtle geometry and the coordinate geometry might be a valuable topic for future research.

One important conceptual change in the learning activity was that the participants began to see patterns of the objects and transfer them into programming language. This was evident in the construction of rectangular faces where the formula for a square repeat 4 [fd length rt 90] was transformed into repeat 2 [fd length rt 90 fd width rt 90]. It was also evident in Alekat20’s templebase where she formulated repeat 9 [fd 1.1 box fd 1.1 rt 360/18 fd 2.2 rt 360/18] for putting a scales box on top of two cylinders (see Figure 8.22). This ability to see and analyse patterns from real world environment is invaluable for constructing 3D artefacts and problem solving. Other learning occurred during the project included the strategy to start from home location in procedures. It was noted that the home location played a crucial role as a reference point throughout the construction of 3D virtual worlds. The use of both egocentric and fixed frames of reference commands was also
observed. The UP and DN commands were particularly useful and were used by all participants in their procedures.

The use of the Forum during this activity still served as a channel for communication between groups. There were interactions about expressing ideas (e.g., announcing projects) and praising other group's work. Unfortunately, each participant focused much on his/her own tasks. The participants didn't comment on others' work. A new behaviour of using the Forum was found during this learning activity. Victor showed great interest in knowing other members in the Forum community. According to the Forum database, Victor did send two private messages to two Forum members during this activity. However, no replies had been received. The interaction between the participants and other Forum members would be valuable. However, the benefit wouldn't be significant when the Forum community was still small.

Lastly with respect to usability issues, the participants' interaction with the programming interface was examined. In this learning activity, the main focus of analysis was on whether the project management and the operations of editing and merging procedures had facilitated or impeded the collaboration and the construction of 3D virtual worlds. Although the participants had successfully completed group projects, the following usability issues were identified.

1. The default project status was public. This resulted in too many personal and experimental projects listed in the public project area. When group members tried to merge procedures by opening other's project, they were confused with too many public projects. The suggested solution was to set the default project status as not public, and ask the users to set only completed projects as public.

2. The project list only displayed the project names. Because it was allowed to have same project names in VRMath system, the users might confuse with the latest one if there were two or more projects with the same names. The suggested solution was to display more information of projects such as last edited time in the project list.

3. The editing of procedures often involved the use of multiple editor windows such as Procedure Editors, Material Editor, and Pen Colour
Editor. These editor windows could only be opened through the Tool Menu in the main window. Therefore, users had to frequently switch among main and editor windows. It was thought that some kind of mechanism to integrate multiple editor windows into one would facilitate the editing of procedures.

4. The merging of procedures from different projects was a tedious process, in which the users had to switch among projects and copy and paste procedures. It was thought that a procedure library or some mechanism to allow users to integrate procedures should be designed in the programming interface.

5. For collaboration purpose, some kind of group project space should be designed to enable easy sharing of procedures among group members.

These usability issues will be addressed and reflected in the next version of VRMath system for next iteration cycle of the design experiments.

8.3 SUMMARY

This chapter reported on the enactment and evaluation of an application type of learning activity: Activity 9.

In Learning Activity 9: Final project, each of the two groups of participants collaboratively built a 3D virtual world. School C participants constructed a house with some indoors and outdoors items. School K participants created a temple with a stage, some cylindric structures, and trees surrounding the temple. The major findings from this part of the research study were:

1. With respect to pedagogy, four stages of instructions should be utilised in projects such as this activity: (a) examine existing projects, (b) form topic for the project, (c) implement the project, and (d) publish the project. A fifth stage of instruction, which is commenting on other’s projects would be valuable for future iteration of this learning activity.

2. With respect to programming related to conceptual change, this learning activity internalised and developed the participants’ understanding about the syntax of a procedure, algebraic thinking through the use of parameters
in procedures, the linear progression of programming logic, and the reuse of procedures.

3. With respect to 3D geometry-related conceptual change, the results showed sophisticated learning of 3D geometry through engaging in 3D visualisation and navigation (topological resources), and programming language (typological resources) could be achieved.

4. 3D visualisation could potentially mislead the judgement of numerical value of real world objects. It is important to not equate the VR environment to the real world.

5. The construction of 2D faces was difficult because the dimension of the 2D faces needed to be specified by moving the turtle. The drawing of a 2D plan could help the participants to plan and construction 2D faces. This was in contrast to the construction of 3D shapes using built-in primitive shapes, in which the turtle was always at the centre of the 3D shapes.

6. Turtle geometry could easily create a circle (forward a little bit and turn a little bit), but could not identify the centre point of the circle easily.

7. The participants started to recognise the patterns in real world objects. This was reflected in their use of REPEAT command. The participants also learnt to start from home location in their procedures, and used both egocentric and fixed frames of reference commands to construct 3D objects.

8. The participants focused much on their own tasks and didn't comment or discuss through the VRMath Forum in this learning activity. Some participants showed interests on contacting other members. It seems that the benefits that can accrue from the Forum probably will not be significant if the Forum community has few participants.

9. Some usability issues regarding the project management for collaboration were identified. These included: (a) the confusion of too many public projects (b) the confusion of projects with the same names, (c) too many Editors' windows, (d) tedious process of merging procedures, and (e) a group project space should be designed.
9.1 OVERVIEW

Chapter 9 reports on the post analysis including post-interview, usability inspection, and artefact analysis of Iteration 2.

The post interview focused on the participants' elaboration about their procedures in their final projects, and opinions about the design of the VRMath system (i.e., usability issues). Data were qualitatively analysed and interpreted.

The usability inspection section reported a specific usability test on the design of the Navigation Toolbar icons. The Navigation Toolbar icon test was administered before Learning Activity 2, which was three days after the participants had been introduced about the icons in Learning Activity 1. The test result was quantitatively analysed and interpreted.

The artefact analysis focused on the participants’ utilisation of the forum interface (social-actional resources). Examined were the forum posts, private messages, and the maintenance of profiles. Data were quantitatively analysed and interpreted to draw implications about forum communication.

9.2 POST INTERVIEW

The post interview consisted of individual and focus group interviews. Individual interviews focused on the participants’ understanding and learning about geometry and programming. Focus group interviews were conducted after the individual interviews focusing on usability issues.

9.2.1 Results

Individual interviews

Before the individual interview, the researcher printed out each participant’s procedures from their group projects. The interviews were based on the participants' review and explanations of their procedures.
Rosco was asked to explain his wall_2 procedure. It had been about a week since he constructed the procedure. When asked, he seemed to forget what the wall_2 procedure was for. He then started to reason from the language codes and drew a diagram to explain to the researcher. He went through the commands and correctly explained about material settings, FACE, PCOFF, and METER commands. He followed the move and turn commands to draw the diagram. When he had completed the diagram, he knew that the wall_2 procedure was for the front wall (including the door way) of the house (see Figure 9.1).

![Figure 9.1 Rosco's explanation about wall_2 procedure](image)

When asked about his roof procedure (see Figure 9.2), Rosco explained that he first commanded the turtle to the middle of the house, and then commanded the turtle to go up for the cone roof. The researcher questioned why "up 6.4"? Rosco thought for a while then explained that that was the place for the cone roof. The researcher reminded him that the height of the wall was 4 metres, and the scale for the height of the cone was 5. Rosco then found out that the right location to place the cone should be "up 6.5". However, he also said that "6.4" was ok, because the cone roof would "overlap" with the walls.

![Figure 9.2 Rosco's explanation about roof procedure](image)
Bonbon was asked to explain her Cooktop procedure (see Appendix 8). She first looked at the material settings and said that she couldn't remember what those material commands meant, but she knew how to use them. Bonbon then explained very clearly about how to copy the material settings from Material Editor into Procedure Editor. When asked about the :s parameter, she explained that the :s meant "scale/size/space", and the "*" sign meant "times". However, when asked to explain further, she couldn't elaborate why some numbers had to multiply by the :s. Instead, she tried to calculate the multiplications and show those numbers to the researcher.

Grae had five procedures: pool, fence, tree, ground, and init in the house project (see Appendix 8). The fence and tree procedures were copied from other public projects. Grae was able to explain the pool and ground procedure. However, when asked about the tree and fence procedure, he expressed that he couldn't understand but he could use them. The researcher questioned about his command fence 40 3 1.6, in which the forty-three metres long fence wasn't common in the real world. Grae agreed but he expressed that that was for the right size of the fence.

Alekat20 had a templebase procedure (see Appendix 8 and Figure 8.22) utilising the polygon formula. She explained every single command in the procedure very clearly, and showed a full understanding of the commands as if she was seeing the real 3D scene instead of the programming language. For example, Alekat20 explained that the reason to command the turtle up and down was to move the turtle to the centre of the cylinder according to the scale of height. In her commands repeat 9 [fd 1.1 box fd 1.1 rt 360/18 fd 2.2 rt 360/18], Alekat20 not only explained why fd 1.1 but also if not fd 1.1 what would have happened.

R2D2 also explained his procedures very well. When asked about his fountain procedure (see Appendix 8), he could easily visualise the commands and told the researcher which command was for which part of the fountain. Through the explanation, R2D2 also found a redundant code scale 1.5 0.3 1.5, which should be removed because another scale command was right after it. In his body procedure (see Appendix 8), the researcher had helped R2D2 to form a complex command repeat 4 [up 0.4 fd 0.25 scaled 6-repcount*0.5 box]. When asked to explain, R2D2 couldn't remember what REPCOUNT command was. After the researcher explained to him about REPCOUNT, he immediately remembered and correctly explained that
in the four repeats the SCALED command would be \textit{scaled 5.5, scaled 5, scaled 4.5} and \textit{scaled 4} respectively.

Victor was first asked to explain his bridge procedure (see Appendix 8). He explained all the commands in the procedure correctly. These included the SCALE, METER, FACE, material commands, PD, REPEAT, and PU commands. Victor expressed that he liked to use SCALE command more than SCALEW, SCALEH, and SCALED commands because SCALE was "easier to use". The researcher suggested that a PCOFF command could be added into the bridge procedure to make sure the bridge was using the material rather than pen colour. Interestingly, Victor expressed that he didn't know about PCOFF command. When asked to explain the \texttt{:s} parameter in his tree procedure (see Appendix 8), Victor could use examples such as "\texttt{tree 1} you get size one tree", "\texttt{tree 10} you get ten times big tree". He also recalled that he used random number for the tree procedure. However, when asked to explain \texttt{tree (\texttt{((random 10)+1)/10)+1}}, he was unable to answer. After the individual interviews, the participants were asked to draw 3D shapes on papers as they did in pre-interview. The results of their perspective drawing of 3D shapes in this post-interview, together with the results from pre-interview, are presented in Table 9.1.

\textbf{Table 9.1}
\textit{Comparison of Participants' Drawing of 3D Shapes in Pre and Post Interview}

<table>
<thead>
<tr>
<th></th>
<th>Alekat20</th>
<th>R2D2</th>
<th>Victor</th>
<th>Rosco</th>
<th>Bonbon</th>
<th>Grae</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cube</td>
<td><img src="image1" alt="Cube PRE" /></td>
<td><img src="image2" alt="Cube POST" /></td>
<td><img src="image3" alt="Cube PRE" /></td>
<td><img src="image4" alt="Cube POST" /></td>
<td><img src="image5" alt="Cube PRE" /></td>
<td><img src="image6" alt="Cube POST" /></td>
</tr>
<tr>
<td>Cone</td>
<td><img src="image7" alt="Cone PRE" /></td>
<td><img src="image8" alt="Cone POST" /></td>
<td><img src="image9" alt="Cone PRE" /></td>
<td><img src="image10" alt="Cone POST" /></td>
<td><img src="image11" alt="Cone PRE" /></td>
<td><img src="image12" alt="Cone POST" /></td>
</tr>
<tr>
<td>Cylinder</td>
<td><img src="image13" alt="Cylinder PRE" /></td>
<td><img src="image14" alt="Cylinder POST" /></td>
<td><img src="image15" alt="Cylinder PRE" /></td>
<td><img src="image16" alt="Cylinder POST" /></td>
<td><img src="image17" alt="Cylinder PRE" /></td>
<td><img src="image18" alt="Cylinder POST" /></td>
</tr>
</tbody>
</table>
Focus group interviews

The focus group interviews were semi-structured interviews focusing on usability issues of VRMath system. The main interview questions are listed in Appendix 5. Based on those main questions and the responses of the participants, more questions were administered wherever appropriate.

When asked "Do you like using VRMath?", all participants immediately replied "Yes", and expressed that they enjoyed very much. The common reasons were "VRMath is fun", and "the 3D world is cool".

The question "Is VRMath easy to use?" also got almost immediate reply "Yes". However, when the researcher reminded the participants to consider 3D navigation, programming and commands, and Forums, School C participants revised their words to, "It is hard to do but easy to learn". School K participants also shared School C participants' idea. They expressed that VRMath was pretty easy to learn and use, but doing the project had given them a hard time. Alekat20 said that her frustration was when she typed in "hi" and got the "I don't know how to hi" feedback, she couldn't understand what that message meant.

For the question "Is 3D navigation in VRMath easy?", the short response was still "Yes". The researcher also questioned about the participants' preferences of navigation mode, when to use which mode, and the need of the three navigation modes. Rosco expressed that he rather like to fly than to walk. Bonbon said that, "It all depends on what you want to do something. If you want to walk through a door or up stairs, then you need walk mode." School K participants expressed that the Examine mode and Fit Screen were the best way to navigate. R2D2 and Victor mentioned that they could use meta-keys such as Shift, Ctrl, Alt, and Space in
conjunction with the three navigation modes. They both added that they use Shift and Ctrl keys a lot to increase the navigation speed.

When asked "Are VRMath commands easy to use and remember?", the participants had different responses. School C participants talked about the acronyms of commands such as PD for PenDown were easy to use and remember. However, School K participants expressed the acronyms were sometimes confusing and hard to guess or remember. For example, PenDown could mean to put away the pen so there should have no turtle tracks. TL (TiltLeft) and TR (TiltRight) could be thought of as TurnLeft and TurnRight. Victor and R2D2 expressed that they like to use Quick Command instead of typing the commands. Alekat20, in contrast, preferred to type in commands because "that can do more stuff". Victor expressed that it was not easy to create a face. He suggested that there should have commands like RECTANGLE, TRIANGLE, SQUARE, or CIRCLE. The researcher took this opportunity to bring in questions about procedures. The researcher explained that the current commands were probably enough as "building blocks", and those commands suggested by Victor could be created by defining procedures. When questioned why there was a bridge procedure separated from the ground procedure in Victor's Temple part (see Appendix 8), Victor replied that "because the ground procedure will be too long". Alekat20 answered that "because you can reuse it".

To the question "Is VRMath Forum easy to use?", all participants responded "Yes" and "kind of easy". They mentioned that they could use smilies (emoticons), change font colour and size, and create a poll easily. R2D2, however, suggested that the Forum should allow users to "put pictures in".

For the question "What other aspects are easy or difficult in VRMath?", School C participants replied that they didn't feel anything particularly difficult in VRMath. Once learnt, everything was kind of easy. School K participants, in contrast, mentioned some difficulties and frustrations. R2D2 said that there were quite a lot of commands to memorise. All the three participants talked about that the "save project" function in programming interface was very confusing. Alekat20 expressed that she was frustrated by the save project function, which couldn't save the 3D virtual worlds she created.

Up to this stage, it was pertinent to ask the last question "How can VRMath be improved?" To the researcher's surprise, the participants provided many
suggestions on the design of VRMath. The participants started to talk about the icons in the Navigation Toolbar. They examined each icon, talked about how and when they used those icons and functions in previous learning activities. Some arguments emerged during this discussion. For example, Victor and Alekat20 commented that the Collision Mode icon was useless. R2D2 reputed and gave example of using it for his Temple stage in Walk mode. At the end, the participants concluded that Headlight and Horizontally Straighten were useless, and Restore Viewpoint was the most important one. The researcher showed the participants their Navigation Toolbar test result (see Table 9.2 in Section 9.3), and asked if they could draw some icons to improve the use of icons. The participants' suggestions about icons are presented in Figure 9.3.

![Rosco's Full Screen icons](image1.png) ![Alekat20's Compass icon](image2.png) ![Bonbon's Headlight icon](image3.png)

*Figure 9.3 Suggestions for icons in Navigation Toolbar*

The participants also discussed about the tools such as Background Chooser, Material Editor, and Quick Command in Tool Menu (see Figure 3.11), and commented that those tools were all important for VRMath. Alekat20 suggested that the Quick Command (see Figure 3.16) should enable the input of command, not just displaying the command. This was an excellent suggestion, and was soon implemented after Iteration 2 (see Figure 9.4).

![Improved Quick Command interface](image4.png)

*Figure 9.4 Improved Quick Command interface*
9.2.2 Discussion

When the participants constructed their procedures in Learning Activity 9, they interpreted from the plans they drew (visual images) to form the commands. The participants were interpreting the figural information (IFI, see Bishop, 1983) into non-figural data (programming commands). In the individual interviews, the participants were asked to explain their programming codes. In doing so, the participants were actually translating from non-figural data (programming codes) to 3D visualisation (visual images). This type of activity is operationalising the visualization ability VP (visual processing of information) as identified by Bishop (1983). Therefore, the individual interviews provided an opportunity to develop the participants' VP ability, and enabled a deep probe into the participants' understanding of 3D geometry.

From the individual interviews, it was found that all participants had very good understanding about the programming commands they wrote. They could correctly interpret from the programming language into 3D visual or mental imagery. Furthermore, in the process of interpreting, the participants could also debug and improve their procedures. From the participants' bi-directional translating between 3D visual imagery and language, it could be confirmed that deep learning about 3D geometry had occurred; it occurred through the meaning-making activities within multiple semiotic resources.

From the individual interviews, however, two potential flaws were identified. The first potential flaw was the expression of "couldn't understand but could use" of some programming and procedures. These included the use of :s parameter, recursive tree procedure, polyprism procedure, and the random command for the Temple trees. From a programmer's perspective, it may be the nature of programming, in which the fundamental libraries of procedures or functions were already created; the programmer only needs to use them and doesn't need to understand the details in the procedures or functions. However, in terms of learning, it could be argued that the learners should comprehend the procedures as fully as possible. Although it was difficult for the participants to comprehend every detail of the procedures in such a short period of time, it should be set as a long term goal for future iterations if time allows. The second potential flaw was related to the virtuality of the VR environment as evident in Grae's unreal fence. This could be seen as the limitation of the desktop
VR environment. It also reflected the principle of semiotics, in which the VR environment was a sign; VR was not reality; and a sign only carried certain aspects of the object for different interpretants (Cunningham, 1992b). To avoid equating VR and real world by the participants, the consciousness of constructing in computational VR environment, and the reality in real world should be raised.

The drawing of 3D shapes was designed to identify the change of the participants' ability to visualise 3D shapes. The results in Table 9.1 showed a great improvement in drawing the cube. This was particularly significant in School C participants' drawings of a cube. It was conjectured that this was being the result of creating a frame of a cube in learning activity 8. The drawing of prism and pyramid was not the focus of this Iteration, and the results were a bit odd. School K participants' drawings of the triangular based prism were improved. However, for some unknown reason, School C participants' drawings of the prism and pyramid seemed to get worse.

The focus group interviews evaluated the usability of VRMath system. The questions asked were based around the ideas of Nielsen's (1993) five usability attributes. From the responses of focus group interviews, it was evident that VRMath was pleasant to use for all participants. This was perhaps the most important usability attribute because when the participants were having fun designing cool virtual worlds, they would be fostering a positive attitude towards learning with VRMath. Learning from doing and playing was also an underlying design principle of VRMath, which was confirmed to have been achieved from the interviews. The comment "hard to do but easy to learn" was somewhat perplexing. However, it became clear when questioned into each of the three interfaces (VR, programming, and Forum interface). The "easy to learn" was referring to the GUI interfaces and operations such as 3D navigation, using commands and procedures, and discussion in Forums, were easy to learn and use. The "hard to do" was referring to the content knowledge (e.g., geometry and programming skills) and the style of using turtle graphics to construct complex 3D structures. The researcher reflected that the usability regarding easy to learn and use should be improved to ease the "hard to do" part. The participants' suggestions such as the acronyms of commands and feedback in the programming console will be carefully considered in the future design of VRMath. However, the turtle graphics style is the key feature to engage learning.
How to change this style to compromise between "easy to do" and "learning" will be a complex topic for future design.

Lastly, the participants' enthusiastic suggestions on how to improve VRMath were very inspiring. They provided simple yet powerful designs of visual icons and Quick Command interface from users' point of view. It was thought that if this VRLE was to be used by students, then students should have a say on its design. Thus, the VRLE would be a more user centred design and better usability of the VRLE could be expected. In the research literature, the user's contribution to the system design has been termed as participatory design (PD). PD is a profound study relevant to the field of human-computer interaction (HCI) (see for example, Carroll, 1996; Kautz, 1996; Schuler & Namioka, 1993; Tec-Ed Inc., 2004). The researcher will integrate PD to inform the future design of VRMath.

9.3 USABILITY INSPECTION

The usability inspection was conducted throughout Iteration 2 while the participants interacted with VRMath system. This section specially reported a usability test about the Navigation Toolbar icons (see Appendix 6). The Navigation Toolbar icon test was administered before Learning Activity 2, which was three days after the participants had just been introduced briefly about the icons and their functions in Learning Activity 1. This was designed to identify the meaning emission capacity and the memorability of those icons. The test results are presented in Table 9.2.

Table 9.2

<table>
<thead>
<tr>
<th>Icon</th>
<th>Function</th>
<th>Correct identification (out of six)</th>
<th>Error links to</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="" /></td>
<td>Walk mode</td>
<td>6</td>
<td>Collision mode, Walk mode</td>
</tr>
<tr>
<td><img src="image" alt="" /></td>
<td>Fly mode</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td><img src="image" alt="" /></td>
<td>Examine mode</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td><img src="image" alt="" /></td>
<td>Avatar / Free view</td>
<td>4</td>
<td>Collision mode, Walk mode</td>
</tr>
<tr>
<td><img src="image" alt="" /></td>
<td>Set rotation centre</td>
<td>3</td>
<td>Horizontally straighten, Full screen x 2</td>
</tr>
<tr>
<td><img src="image" alt="" /></td>
<td>Horizontally straighten</td>
<td>2</td>
<td>Fit screen x 2, Set rotation centre, Collision mode</td>
</tr>
<tr>
<td><img src="image" alt="" /></td>
<td>Restore viewpoint</td>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>
From the test results, it could be seen that many icons possessed other meanings in addition to their designed meaning. For example, the Avatar View icon was designed to mean that a person was viewing in front of an object, but obviously it could be associated to collision mode and walk mode too. This confirmed with the literature review about semiotics (Cunningham, 1992b), in which a sign has some aspects that are irrelevant to the object. Those icons identified to possess strong irrelevant meanings will be carefully redesigned in the next iteration.

9.4 ARTEFACT ANALYSIS

As a learning environment that encourages learning from doing and playing, VRMath enabled its users to construct many meaning-making artefacts. These artefacts include the 3D virtual worlds, procedural language, and the
communications in the Forum. The 3D virtual worlds and procedural language constructed by the participants were presented and analysed in previous learning activity sections. This section focuses on the analysis of the Forum artefacts including the participants’ posts, private messages, and profiles.

9.4.1 Forum posts

During 32 sessions (16 sessions for each school, one hour per session) of interaction with VRMath system, there were a total of 159 posts generated in the Forum by the six participants and the researcher. Although the researcher had reminded the participants to post in every session, it was still voluntary for the participants to choose whether or not to post in the Forum. Table 9.3 presents a descriptive statistics of the Forum posts in Iteration 2.

Table 9.3
Statistics of Forum Posts in Iteration 2 (1)

<table>
<thead>
<tr>
<th></th>
<th>Alekat20</th>
<th>R2D2</th>
<th>Victor</th>
<th>Bonbon</th>
<th>Rosco</th>
<th>Grae</th>
<th>Andy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total posts</td>
<td>40</td>
<td>29</td>
<td>28</td>
<td>7</td>
<td>15</td>
<td>6</td>
<td>34</td>
</tr>
<tr>
<td>New posts (76, 48%)</td>
<td>23 (57.5%)</td>
<td>14 (48%)</td>
<td>12 (43%)</td>
<td>4 (57%)</td>
<td>5 (33%)</td>
<td>5 (83%)</td>
<td>13 (38%)</td>
</tr>
<tr>
<td>Reply posts (83, 52%)</td>
<td>17 (42.5%)</td>
<td>15 (52%)</td>
<td>16 (57%)</td>
<td>3 (43%)</td>
<td>10 (67%)</td>
<td>1 (17%)</td>
<td>21 (62%)</td>
</tr>
<tr>
<td>New posts replied (41, 54%)</td>
<td>11 (48%)</td>
<td>9 (64%)</td>
<td>6 (50%)</td>
<td>4 (100%)</td>
<td>4 (80%)</td>
<td>2 (40%)</td>
<td>5 (38%)</td>
</tr>
<tr>
<td>Post to show and tell (56, 53*)</td>
<td>14 (35%)</td>
<td>16 (55%)</td>
<td>5 (18%)</td>
<td>5 (71%)</td>
<td>8 (53%)</td>
<td>5 (83%)</td>
<td>3 (9%)</td>
</tr>
<tr>
<td>Post to ask questions (47, 19*)</td>
<td>15 (37.5%)</td>
<td>2 (7%)</td>
<td>1 (4%)</td>
<td>0</td>
<td>1 (7%)</td>
<td>0</td>
<td>28 (82%)</td>
</tr>
<tr>
<td>Post to comment (49, 35*)</td>
<td>9 (22.5%)</td>
<td>12 (41%)</td>
<td>5 (18%)</td>
<td>3 (43%)</td>
<td>5 (33%)</td>
<td>1 (17%)</td>
<td>14 (41%)</td>
</tr>
<tr>
<td>Post to praise (23, 11*)</td>
<td>4 (10%)</td>
<td>1 (3.4%)</td>
<td>4 (14%)</td>
<td>1 (14%)</td>
<td>1 (7%)</td>
<td>0</td>
<td>12 (35%)</td>
</tr>
<tr>
<td>Post for irrelevant topics (9)</td>
<td>3 (7.5%)</td>
<td>0</td>
<td>6 (21%)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total polls (15)</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Posts contain emoticon (109, 69%)</td>
<td>17 (43%)</td>
<td>29 (100%)</td>
<td>6 (21%)</td>
<td>6 (86%)</td>
<td>15 (100%)</td>
<td>2 (33%)</td>
<td>34 (100%)</td>
</tr>
<tr>
<td>Posts contain rich text (28, 18%)</td>
<td>5 (12.5%)</td>
<td>0</td>
<td>0</td>
<td>5 (71%)</td>
<td>1 (7%)</td>
<td>1 (17%)</td>
<td>16 (47%)</td>
</tr>
<tr>
<td>Posts contain signature (138,87%)</td>
<td>37 (93%)</td>
<td>29 (100%)</td>
<td>24 (86%)</td>
<td>2 (29%)</td>
<td>12 (80%)</td>
<td>0</td>
<td>34 (100%)</td>
</tr>
<tr>
<td>In 3D Virtual Space forum (101, 64%)</td>
<td>34 (85%)</td>
<td>21 (74%)</td>
<td>11 (39%)</td>
<td>3 (43%)</td>
<td>9 (60%)</td>
<td>1 (17%)</td>
<td>22 (65%)</td>
</tr>
<tr>
<td>In Geometry forum (2, 1%)</td>
<td>0</td>
<td>0</td>
<td>1 (4%)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1 (3%)</td>
</tr>
<tr>
<td>In Programming forum (30, 19%)</td>
<td>2 (5%)</td>
<td>4 (14%)</td>
<td>10 (36%)</td>
<td>2 (29%)</td>
<td>2 (13%)</td>
<td>1 (17%)</td>
<td>9 (26%)</td>
</tr>
<tr>
<td>In Show Room forum (17, 11%)</td>
<td>3 (7.5%)</td>
<td>2 (7%)</td>
<td>2 (7%)</td>
<td>1 (14%)</td>
<td>3 (20%)</td>
<td>3 (50%)</td>
<td>2 (6%)</td>
</tr>
<tr>
<td>In Test forum (10, 6%)</td>
<td>1 (2.5%)</td>
<td>2 (7%)</td>
<td>4 (14%)</td>
<td>1 (14%)</td>
<td>1 (7%)</td>
<td>1 (17%)</td>
<td>0</td>
</tr>
</tbody>
</table>

* disregard the researcher's (Andy) posts.

The number of total posts indicates the participants’ utilisation of this social-actional resource. For individual participant, Alekat20 had the most number of posts (=40) while Grae had the least number of posts (=6). Because of the short period of this study, a total of 40 posts over 16 sessions indicated good utilisation of the Forum.
resources (an average of 2.5 posts per session). In contrast, only 6 posts over 16 sessions indicated a lack of utilisation of the Forum resources. Judging by groups, School K participants had better utilised the Forum than School C participants. The number of total posts may also indicate the potential for learning to occur. However, the actual learning occurred and knowledge building activities also require qualitative analysis of the quality of posts.

The total posts could be divided into new posts and reply posts. New posts were the posts which started a new topic or new ideas. Reply posts were those responded to either new posts or reply posts. The number of new posts indicated the ability to initiate a new topic while the number of reply posts indicated the ability to interact with new topics. Both new and reply posts were considered essential for knowledge building discourse. Table 9.3 showed that all participants were able to initiate new topics and reply to others in the Forum. Ideally, the number of reply posts should be more than new posts, which could mean that the chance of getting replied for the new posts would be higher. From the results in Table 9.3, Alekat20 and Grae had more new posts than their reply posts. Grae, in particular, only posted replied once, which showed his minimal interaction with other participants through the public forums. The total reply posts (=83, 52% of total posts) were more than the total new posts (=76, 48% of total posts), which indicated a certain level of interaction within the Forum. There were 41 out of 76 new posts or 54% of the new posts were replied, which meant that almost half of the new posts didn’t generate discussion. The 41 replied new posts were further analysed in terms of within and between schools interaction and the depth of the discussion indicated by the number of replies (see Table 9.4).

<table>
<thead>
<tr>
<th>Replies</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>6</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of replied new posts</td>
<td>21</td>
<td>12</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>41</td>
</tr>
<tr>
<td>Between schools interaction</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>Within schools interaction</td>
<td>19</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>29</td>
</tr>
</tbody>
</table>

As can be seen in Table 9.4, there were more interactions occurred within schools (=29) than between schools (=12). The maximum number of replies for a new post was 6. Most interactions only consisted of 1 or 2 replies. It was conjectured...
that the asynchronous communication in the Forum was the main reason for the small number of replies and that it discourages the interaction between schools. Another possible reason would be the content of their posts, whether would the content engage discussion or not.

The analysis of the Forum posts identified some types of post in the Forum. These were presented in Table 9.3 above as post to show and tell, to ask questions, to comment, to praise, and for irrelevant topics. A single post may contain more than one type of post. As shown in Table 9.3, post to show and tell was the prevailing type of post (=53). All participants except Alekat20 posted to "show and tell" the most. The relationship between the type of post and the posts replied was intriguing but unclear. Each type of post could generate discussion even the irrelevant topics. However, it was more likely to have learning occurred when the participants posted to comment and ask questions. It was also noticed that post to praise was usually the end of discussion and interaction. To maximise the benefit of social interaction for learning, how to scaffold and engage the participants to question and comment should be a focus for future iterations.

Other factors to incur social interactions included the polls and the use of emoticon, rich text, and signature. It was found that most polls were voted by participants from both schools. The online poll in VRMath Forum was a new experience to the participants. All participants except Grae had created at least a poll. Of the total 15 polls, 4 were irrelevant to the learning activities. However, all polls had votes from 3 to 8\(^{28}\), which indicated a high interaction for a six participant community. It was found that the polls were easier to interact with. And because the polls were attached to posts, they could facilitate the expression of ideas and communication in the Forum. There was also a high usage of emoticons in the Forum posts. R2D2 and Rosco utilised emoticons in their every post. It was found that the participants could pertinently insert emoticons in the posts to express their emotions such as very happy, smile, sad, surprised, shocked, confused, cool, embarrassed, crying, and mad etc. The use of rich text (e.g., font size and colour) in posts was not as popular as the use of emoticons. R2D2 and Victor didn't even use any rich text. Bonbon and Alekat20 were found to use the rich text very well to emphasise their

---

\(^{28}\) The 8 votes are from the six participants and the researcher, and the account James created by Victor.
main ideas in their posts. The signature feature was also extensively used in posts except Bonbon and Grae.

VRMath Forum had many sub-forums setup to categorise discussions (see Section 3.6.2). These sub-forums were designed to classify and structure discussions. The statistics in Table 9.3 shows that 64% of the posts were posted in 3D Virtual Space forum. However, a review of those 64% posts indicated that only about a quarter of the 64% belong to the 3D Virtual Space forum. This indicated that the design was not functioning as expected; and the purpose of the sub-forums should be made clear to the participants in the future iterations.

9.4.2 Private messages

The private message is a means of communication between the forum users in the VRMath Forum system. It differs from the Forum posts, which are public for all users to read. Private message, on the contrary, can only be seen by the sender and receiver. Due to the ethical reason and to protect the participants’ privacy, the content of the participants’ private messages were not read and analysed. However, it was possible to report some quantitative figures. Table 9.5 presents the number of private messages sent and received by each participant.

Table 9.5
Statistics of Private Messages in Iteration 2

<table>
<thead>
<tr>
<th>To</th>
<th>From</th>
<th>Alekat20</th>
<th>R2D2</th>
<th>Victor (James)</th>
<th>Rosco</th>
<th>Bonbon</th>
<th>Grae</th>
<th>Andy</th>
<th>Danevans</th>
<th>Total received</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alekat20</td>
<td></td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>8</td>
<td>8</td>
<td>0</td>
<td>26</td>
</tr>
<tr>
<td>R2D2</td>
<td></td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>7</td>
<td>0</td>
<td>17</td>
</tr>
<tr>
<td>Victor (James)</td>
<td></td>
<td>5</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>9</td>
<td>7</td>
<td>0</td>
<td>34</td>
</tr>
<tr>
<td>Rosco</td>
<td></td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>Bonbon</td>
<td></td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>16</td>
</tr>
<tr>
<td>Grae</td>
<td></td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>Andy</td>
<td></td>
<td>7</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>21</td>
</tr>
<tr>
<td>Danevans(^{29})</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Ken(^{30})</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Birbilis(^{31})</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

\(^{29}\) Danevans was a user of VRMath Forum as well as the grandfather of Rosco and Bonbon. He participated in the learning activity 1.

\(^{30}\) Ken was a university lecturer in UK.

\(^{31}\) Birbilis was a senior specialist in VR and programming.
From Table 9.5, it was noted that R2D2 didn't send to Alekat20, Victor didn't send to Rosco, and Rosco didn't send to R2D2 and Bonbon any private message. This was the evidence for the deletion of private messages as all participants had been asked and sent at least one private message to each of the other participants during learning activity 1. Disregard the deleted private messages, 4 out of the 6 participants (i.e., Alekat20, Rosco, Bonbon, and Grae) had more between schools than within school communication using private messages. Among the six participants, Grae sent the most private messages (=31) to other participants. This was in contrast to his usage of the public forums where he had only 6 posts. This demonstrated that Grae was able to use the Forum resources. However, he had a preference to communicate privately rather than publicly.

Another important finding from Table 9.5 was the attempt to contact other users in the Forum community. Rosco sent a private message to Ken and Birbilis; and Victor sent a private message to yuchouchen. According to the database, both Rosco and Victor didn't receive any reply from those users. Ken and Birbilis are knowledgeable members of VRMath Forum. Learning from interacting with more knowledgeable members was an important aim of the Forum community. Rosco and Victor's attempt to contact other Forum members showed a potential for learning as well as the need for wider social interaction. This should be an imperative task for future iterations to encourage interactions with more Forum community members.

9.4.3 Maintenance of profiles

The user profile in the Forum contains user's personal information such as web address, location, interest, signature, and avatar etc. This personal information will be displayed in "view profile" (see Figure 6.4) and in the Forum posts. Because these informations are available to the public, it was often observed that the participants changed them during the course of this study. From the Forum database, the participants' behaviours of maintaining their profiles are presented in Table 9.6.

---

yuchouchen was an anonymous user in VRMath Forum.
Table 9.6

*Behaviours of Maintaining Profiles*

<table>
<thead>
<tr>
<th></th>
<th>Alekat20</th>
<th>R2D2</th>
<th>Victor</th>
<th>Bonbon</th>
<th>Rosco</th>
<th>Grae</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change avatar images</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>1</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Change location info</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Change interests info</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Change signatures</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

The numbers presented in Table 9.6 indicate the number of times each participant changed the particular information in their profiles. Data were collected at the end of each session. However, it was possible that the participants could have changed their profiles some times in one session. The change of avatar images was a time-consuming task as the participants usually had to browse through over 600 avatar images to select one. It was found that the participants were all interested in getting an avatar. However, it seemed that only the boys (except R2D2) enjoyed changing the avatar image. The avatar images selected were found to reflect their gender and favoured characters. For the location, all the participants except Rosco were keen to put in "Aussie" or "Brisbane" to show where they were from. Bonbon was the only one to show her interest (i.e., literature). The change of signature was among the most popular activities in the maintenance of profiles. However, Grae didn't utilise the signature at all.

The participants' maintenance of their profiles was totally autonomous. It was observed that they imitated each other when they saw other's profile. For example, some participants put "brisbane, QLD Australia" in the location when they saw the location "Patras, GREECE" of user Birbilis. Alekat20, R2D2, and Bonbon all have "Aussie" in the location of their profiles. An interesting signature "Don't talk to me" of James was created in response to Alekat20's "Can You Talk To Me???????????? Please!!!!!!". The effect of the user profile on learning was still not clear from the small amount of data. It was even considered a waste of time when the participants were selecting their avatars. However, the user profile does exhibit subtle social interactions, which have influence upon the interaction between users. For example, Grae had many private messages sent to Victor because both of them had chosen similar characters as their avatars.

The user profile had served as a venue for the participants to express themselves, and a place to show their identity. Although the effect of the user profile
is still unclear, it is certainly a part of the messy settings of the learning environment. In light of the design experiments methodology, the user profiles should be examined to provide possible explanations for the justification of the learning environment. In this Iteration, the user profiles were examined and the important components of the user profiles such as the avatar, location, and signature were identified. These components together with the identity issue and how to guide the use of the user profile to facilitate communication should be the focus for future iterations.

9.5 SUMMARY

This chapter reported on the post analysis including post-interview, usability inspection, and artefact analysis of Iteration 2.

The post interview focused on the participants’ elaboration about their procedures in their final projects, and opinions about the design of the VRMath system (i.e., usability issues). The major findings were:

1. The writing of procedures from 2D plans developed the interpretation of figural information ability (IFI, see Bishop, 1983), while the interpretation of programming codes into 3D visualisations or visual imageries developed the participants’ visual processing ability (VP, see Bishop, 1983). This bi-directional teaching and learning developed more holistic spatial abilities.

2. Two potential risks in programming were identified. The first risk was that the participants “couldn’t understand but could use” some programming procedures, which might discourage deep understanding of the programming and 3D geometrical knowledge. The second risk was that the virtuality of the VR environment might foster the participants to equate the VR with the real world, which is inconsistent with the semiotic principle.

3. The VRMath system was “easy to learn but hard to do” indicated a complex design issue to bridge the usability and the learning of 3D geometry.

4. The participants’ participatory design of the VRMath system could contribute to the usability of and the learning within VRMath.
The usability inspection section reported a specific usability test on the design of the Navigation Toolbar icons. The results confirmed the semiotic principle, in which a sign carried both relevant and irrelevant meaning of the object. The icons that possessed irrelevant meanings were identified and will be carefully redesigned in the next iteration.

The artefact analysis focused on the participants' utilisation of the forum interface (social-actional resources). The major findings were:

1. The number of total posts indicated a good utilisation of the forum communication. The forum facilities such as poll, emoticons, rich text, and signatures were highly utilised. However, posts also contained irrelevant topics and some were posted into inappropriate forums.

2. The analysis of private messages showed that some participants would prefer to communicate through private message rather than public forum. The private message also served as the first communication channel to contact other forum members.

3. The analysis of the maintenance of profiles revealed a sense of self-identity to show the avatar image, location and interests, and signatures. These helped the community members to express themselves and facilitate communications.
CHAPTER 10

REFLECTION AND REDESIGN

10.1 OVERVIEW

Chapter 10 reports on Stage 4 of the design experiments: Reflection and Redesign of the virtual reality learning environment (VRLE). This chapter has three sections: reflection, redesign, and conclusions of research.

The reflection section is a holistic review of this research study. It reflects on the aims and research design reported in Chapter 1, and the previous 3 stages of the design experiments reported in Chapters 2 to Chapter 9.

The redesign section highlights concrete recommendations specifically for future implementations and advancements that could be made to VRMath and the content and conduct of VRMath learning activities.

The conclusions of research section summarises this research study and consistent with the philosophy underlying design experiments methodology (Brown, 1992; The Design-Based Research Collective, 2003; Bereiter, 2002) draws practical and theoretical insights for both mathematics and technology education that are relevant for VRMath in particular and for VRLE contexts in general. Visions and directions for future research about VRLE are presented.

10.2 REFLECTION

The reflection is a "thinking about thinking" process (The Design-Based Research Collective, 2003). After a long journey in conceptualising, designing, implementing, enacting, and evaluating the VRLE, the researcher now steps back to review the whole scene and reflect on his thinking and doing processes throughout this design-based research. This section thus presents a holistic reflection covering the aims, research design, conceptual framework, design of prototype VRLE, enactment and evaluation of the VRLE, design of the learning activities, and limitations of the VRLE.
On aims

The two aims of this study were to: (a) design and evaluate a VRLE to facilitate the learning by upper primary school students of 3D geometry concepts and processes, and (b) generate theoretical and design principles that will have application both within and beyond the immediate research study. In reflection, this research study has followed the direction and achieved the aims as evident by:

1. The generation of a theoretical framework consisting of a set of specifications (see Section 2.5.1) to inform the design of the VRLEs.
2. The physical and sophisticated implementation of the VRMath system (see Chapter 3); and
3. The field experiments with upper primary school students, whose learning of 3D geometry was facilitated by the VRMath system and from which the VRMath system was evaluated (see Chapter 4 to Chapter 9).

The aims, however, can be further explored in future iterations of design experiments to generate new insights for designing and using VRLEs for mathematics education. In future iterations, this VRLE can be applied to different age group learners and across different disciplines.

On research design

The nature of design experiments was reflected in this research. This included the messy settings of the physical environment (Bereiter, 2002), dynamic interactions among the participants (Bereiter, 2002; Brown, 1992), contingencies, and quick prototyping of the VRLE (Hsi, 1998). The enactment of the learning activities was guided by some vision of as-yet-unrealised possibilities and characterized by emergent goals (Bereiter, 2002). Although the research procedures utilised in the study were informed by design-experiments methodology, the stages of research design (see Figure 1.4) did not necessarily reflect the actual conduct of the research.

On reflection, there is an iterative cycle of the design experiments progressing from (Stage 1) the development of conceptual framework, (Stage 2) the design of prototype VRLE, (Stage 3) the enactment and evaluation of the VRLE, and to (Stage 4) the reflection and redesign of the VRLE. The progression of the four stages presents a concise logic of the research. However, the progression was not
necessarily linear in nature. It is important to point out that although Figure 1.4 shows a linear progression of the four stages, the last stage (Stage 4) had inputs for each of the previous stages. This means that in addition to the iterative cycle from Stage 1 to Stage 4, there were also iterative cycles from Stage 2 to Stage 4, and from Stage 3 to Stage 4. For example, this research study reported two iterations in Stage 3, which were actually the small and quick iterative cycles from Stage 2 or Stage 3 to Stage 4. The small and quick iterative cycles reflect the notion of quick prototyping of design experiments to address the new emerging goals (Bereiter, 2002). For example, after Iteration 1 of the enactment stage, the new games (the secret and legend of the turtle, see Section 6.3) were quickly redesigned to enable further investigation of 3D navigation in Iteration 2. During Learning Activity 2, when the participants were commanding the turtle (operating spatial visualisation ability) instead of navigating (operating spatial orientation ability) to solve the secret and legend problems, a new goal for investigation emerged that led to the identification of hitherto new unrealised potential of the VRMath environment. Therefore, a revised diagram for the research design is presented in Figure 10.1.
Furthermore, the cycle of design experiments of this research study (particularly from Stage 2 to Stage 4) is essentially for "engineering a learning environment" as proposed in Brown's (1992) model of design experiments (see Figure 1.3). Stage 1 the development of conceptual framework and the reflection in Stage 4 produced outcomes as the contributions to learning theory bidirectional to the engineering of the learning environment. Stage 2 and Stage 3 of the research design signified the practical feasibility and dissemination of the learning environment. Throughout the engineering of the learning environment (i.e., VRMath), many messy and dynamic inputs such as the group ethos, student as designer, curriculum, and technology were assessed to output the key factors accountable for such a learning environment. On reflection, the research design of this research study was fully consistent with the essence of design experiments methodology.

On the conceptual framework

The development of the conceptual framework followed Resnick's (1996) three threads of thought model: (a) an understanding of the learner (e.g., cognitive development), (b) an understanding of domain knowledge (e.g., mathematical and geometrical knowledge), and (c) an understanding of computational ideas and paradigms (e.g., VR technology and microworlds). These three threads of thought enabled the development of a set of specifications and perspectives to inform the design and evaluation of the VRLE. On reflection, the three threads of thought were found to be sufficient for this research. However, the extension of the conceptual framework for future research should not be limited to these three threads of thought.

Of specifications

There are two questions concerning the specifications. The first is whether VRMath was implemented in accordance with the specifications or not? The second is whether the specifications have led to the development of an efficacious learning environment.

On reflection, the VRMath system was implemented to comply with the specifications. The VRMath environment implements the microworld (e.g., 3D visualisations and programming language) and CSCL (e.g., online Forum
community) paradigms to provide multiple rich 3D geometrical representations. The environment thus enabled and encouraged the perception of mathematics as being meaning-making activities by its users. However, the researcher wishes to point out that although VRMath is an open, generative, and constructionist learning environment, to maximise its potential for student learning, it requires the instructors to understand the epistemological and learning/instructional assumptions underlying the learning environment.

It was evident that the VRMath environment was effective in engaging the students in 3D thinking and doing. For example, in the 3D rotation activity (see Section 7.4), the students experienced using different ways of solving the 3D rotation problem. The four sets of assumptions (epistemological, learning/instructional, about 3D geometry, and about the use and the role of technology, see Section 2.5.1) remain feasible. More iterative cycles of the design experiments are needed for their validation and justification. Nevertheless, new specifications emerged as the curriculum evolved and the new technologies developed.

It was felt that these specifications are mainly addressed from an education perspective. As a technological innovation, the VRMath system should also comply with some kind of software engineering specifications such as usability validity and cycle of software engineering. This, however, requires literature review in the fields of human-computer interaction (HCI) and information technology (IT), and will be the subject in future iterations of the design experiments.

**Of perspectives for evaluation**

The eight perspectives for evaluation (PE) (see Section 2.5.2) were mostly addressed through the nine learning activities (see Table 4.3), but not fully answered. From the results of enactment and evaluation, the researcher reflected on each of the PEs to provide implications for future iteration.

1. Perspectives for Evaluation 1 (PE1): How do students’ spatial abilities as defined by McGee (1979), Hoffer (1977), and Bishop (1983) (see Section 2.3.1.2) change and develop within the 3D VRLE? The learners’ spatial abilities did change and develop within VRMath. In terms of McGee’s (1979) spatial abilities, they were developed through 3D navigation (spatial orientation) and the manipulation of the turtle’s orientation and
location (spatial visualisation). Bishop's (1983) VP and IFI abilities were
developed through the programming and debugging processes, in which
the learners translated between the 3D visualisations and language
(programming codes). Hoffer's (1977) spatial perception abilities were
mainly developed through the continuous visual feedback of 3D
navigation.

2. PE2: How does the ability to interact and link within real time 3D
visualisation (VR interface, topological semiotics), procedural language
communication (Programming interface, typological semiotics), and
hypermedia forum (Social semiotics) during the construction of 3D
gallery artifacts influence how and what students learn about these
artifacts? The analysis of the data indicates that the learning of 3D
gallery within the VRMath environment was the result of the interaction
with the VR, programming, and the forum interfaces. The VR virtual
space signifies the real-world objects and the programming language
signifies the abstraction of 3D geometry, were the two main
representations of 3D geometry for individual learning in this cycle of
design experiments. The forum interface also contributed to the learning of
3D gallery with social expressions and diverse ideas of constructing 3D
gallery artefacts. However, more participants are needed in the future to
enable an in-depth investigation of the social aspect of the learning
through the forum interface.

3. PE3: How do students use and understand the use of the different frames
of reference (e.g., egocentric, fixed, and coordinate systems) to effectively
and/or intentionally move the turtle to construct geometrical objects? The
participants' spontaneous uses of different frames of reference were
explored. However, their intentional uses of different frames of reference
commands to construct 3D artefacts were not identified as yet. The
intentional use of particular frames of reference requires firstly, an
understanding about the effect of different frames of reference, and
secondly the specific learning activities that could elicit the use of certain
frames of reference.
4. **PE4:** How does the ability to engage in 3D transformations of turns, rolls, and tilts impact on students’ knowledge about the geometrical concepts such as similarity and congruence? The concepts about similarity and congruence were explored through the use of scaling and turning commands in creating 3D shapes and in writing procedures with a scale parameter. However, these concepts need to be addressed by focus questions from the teacher during and after the learning activities for deeper knowledge construction to occur.

5. **PE5:** How does the ability to coordinate and integrate the turtle movements and navigation in 3D space influence students’ construction of geometrical objects? The ability to coordinate and integrate the turtle movement and 3D navigation influenced the way the participants constructed the 3D geometrical objects. It is natural that the users will navigate during the construction of 3D geometrical objects. Navigating to examine the 3D objects while moving the turtle to construct them is more likely to build the 3D objects accurately. However, when the 3D viewpoint is different to the turtle’s orientation, the egocentric bug is likely to occur. In future iterations, the egocentric bug may be addressed by designing an animated camera to provide a viewpoint that follows the turtle’s orientation. Some participants showed the behaviour, in which they manipulated the turtle to construct 3D objects without changing the viewpoint. This resulted in the wrong 3D objects being created, but it could be a good indicator of the development of 3D spatial ability if the participants could construct 3D objects from a static 3D viewpoint.

6. **PE6:** How does the ability to explore 3D objects from different viewpoints or perspectives within the VRLE influence students’ conceptions about 3D geometrical objects? For example, how does the fact that the square faces of a cube may not look like squares in many viewpoints in VRLE’s 3D space influence students’ conceptualisation about the nature of a cube? A 3D viewpoint has depth to convey a sense of perspective and projection. However, if the users do not navigate to change viewpoint, they are likely to be hindered by the static perspective or projection of the 3D objects. For example, a rectangular based prism may be recognised as a cube from a
front viewpoint. Observing the users constructing 3D objects from a certain viewpoint can be a way to diagnose the users’ understanding about the properties of 3D shapes (e.g., a cube has equal length, width and height). Examining the 3D objects from multiple viewpoints should be encouraged in order to facilitate the construction of knowledge about 3D objects.

7. **PE7**: How does the ability to distinguish the integrated movement (e.g., moving while turning in real situation) and component movement (e.g., moving and turning are separate in turtle geometry) in students influence their construction of 3D geometrical objects? This is particularly significant as in the VRLE the component movements of the turtle are not commutative. For example, movements of a roll followed by a tilt are different to a tilt followed by a roll. The component movement (i.e., separation of turning and moving) can facilitate the understanding of the geometrical processes and concepts about properties of 3D objects (e.g., angles within a cube). The component movement can be investigated when the users manipulate the turtle, and when the users navigate with the use of the meta-keys (e.g., Alt and Space keys) along with the mouse dragging. The former is for constructing objects, and the later is for getting a vantage viewpoint.

8. **PE8**: How will the children’s lack of kinaesthetic experience in tilting and rolling within a 3D environment affect their ability to operate with these movements within the VRLE? The lack of kinaesthetic experience in tilting and rolling in real world was reflected when the participants were solving the 3D rotation problem (see Learning Activity 6 in Section 5.4.6) and constructing a frame of a cube (see Learning Activity 8 in Section 5.4.8). The participants tended to use 2D analogy for reasoning in 3D, and to show the dominant use of 2D turning (i.e., left and right) in constructing a cube. This, however, requires a long term observation to identify the effect and relationship between real world experience and the virtual manipulation.

In addition to the eight PEs, the following new perspectives for evaluation have emerged during the enactment and evaluation stage of this study. These are
broad perspectives and should be considered in future cycles of the design experiments.

2. Types of social interactions and scaffolds in the forum interface.
3. Use of programming language and mathematical language development.
4. Relationship between the usability of and the learning within the VRMath environment.

On design of the VRLE

The design of the VRMath system was informed by a semiotic framework derived from Cunningham's (1992b) educational semiotics and Lemke's (1998) mathematical semiotics. It was this semiotic framework that made the VRMath system to comply with the specifications. The semiotic framework proposes meaning making of 3D geometry within topological, typological, and social-actional semiotic resources. Informed by this proposal, the design of the VRMath system thus consisted of three interfaces: VR interface (topological resource), programming interface (typological resource), and hypermedia and forum interface (social-actional resource).

On reflection, the researcher wishes to firstly point out that although the three interfaces represent the three types of semiotic resources, they don’t have a definite boundary of their types. This means that VR, as a venue for delivering the topological meaning of 3D geometry, can also convey typological meaning such as a particular view of a cube. Similarly, the programming interface that focuses on the symbolic language has the topological meaning of 3D geometry. However, the GUI components such as Background Chooser and Material Editor in the programming interface also transmit certain topological meanings of the 3D space. The hypermedia and forum interface that enables social interaction among users as oppose to just human-computer interactions in the VR and programming interfaces, can also contain both topological and typological resources. The three interfaces are for the designer to ensure that the VRLE has employed multiple semiotic resources for meaning making. However, it is also important to understand the nature and effect of different types of semiotic resources in order to justify and evaluate learning.
Secondly, the technologies selected haven’t been fully exploited. The desktop VR technology employed has the power to animate the 3D objects within the 3D virtual space. This, however, requires an extension of the Logo-like programming language to be designed and implemented. They have the potential to facilitate the communication and construction of microworlds – a coherent domain of objects and activities, and thus increase the learning possibilities. Moreover, there are alternative technologies such as wearable VR devices for immersive VR environment, and natural language input and processing for the programming interface. These new or alternative technologies may lay down new possibilities for learning. However, how to utilise and integrate them to form powerful VRLEs will require the designers’ understanding of those technologies.

Thirdly, the question “who designs the VRLE?” needs to be addressed. The VRMath system is apparently a collective of ideas from educators, mathematicians, semioticians, and computer programmers. However, the findings from the post-interviews (see Section 9.2) may suggest that the participatory design can facilitate and benefit the design of the VRLEs. Therefore, it is also important to seek collaborative design from “users”, which include the students and any other possible users.

Lastly, there are always usability issues accompanying with any computational learning environment. It is evident that usability of the computational learning environment can influence the learning within the learning environment. In this thesis, only a preliminary usability inspection based on the five usability attributes (Nielsen, 1993) has been conducted. More thorough and systematic usability inspection methods (see Section 2.4.6.3) should be conducted in future cycles of the design experiments.

On enactment and evaluation of the VRLE

The reflection on the enactment and evaluation of the VRLE concerns about the selection of the participants, the settings of the classroom environments, teaching interventions such as instructions and interview protocols, and data collection methods.

The selection of the participants in this cycle of the design experiments was set to the only criterion, which was the participants must possess basic computer
operating skills using the mouse and the keyboard. However, it was confirmed that the School K participants had high academic achievement in mathematics. This is critical for making any inferences about the learning that occurred within the VRMath environment. For future iterations, the selection of the participants may be targeted to more specific learner groups such as younger or older ages, genders, indigenous, socially disadvantaged, students at risk, and gifted students. The number of the participants could be increased to engage more social interaction both within and between groups.

The classroom or lab settings were made as normal a classroom environment as possible except for the small number of participants. During the enactment, there were sick and late coming participants; there were arguments between participants; and there were special ethos of groups. Each participant was assigned to a computer with Internet access to connect to the VRMath system. However, due to the network restriction, School C participants had to access a local VRMath system installed in School C’s intranet environment. This has caused an asynchronous database problem. While many of those messy settings could in hindsight have been prevented, they nonetheless reflected the dynamic nature of the classroom environment. It is important to take these into account for any justifications.

The teaching intervention followed the cycle of learning and teaching: awareness, exploration, inquiry and utilisation (Bredekamp & Rosegrant, 1992). Following Iteration 1, the nine learning activities were purposefully sequenced in Iteration 2, which laid a good foundation for future iterations in the enactment and evaluation stage. The timeline and frequency of engagement sessions in future iterations probably should be adjusted. In this enactment, the participants attended two sessions per week for a consecutive eight weeks. This should in the future probably be changed to two focused periods of enactment with more intense engagement sessions, but with a longer interval between the two periods. This will allow time for deeper reflection of teaching and help identify more learning and usability issues when the participants revisit the VRMath system. The interviews were found inconsistent in Iteration 2. The questions about the nature of mathematics (discovered or invented) in pre interview were not administered in the post interview. The participants were asked to talk and draw in responding interview questions. In
the future interviews, the participants should also be asked to respond in writing. This would provide solid evidence and facilitate the analysis and evaluation.

The data collection methods (see Section 5.2.2) were found to have the following limitations that need to be addressed in future cycles of design experiments:

1. The participants’ interaction with computers is crucial for analysis, but the video camera was unable to clearly capture the three computers. This can be solved in the future by using more video cameras or by special usability testing software that can record the full interaction on the computer screen.

2. The auto-logging program can record each participant’s commands. However, when the participants were gathered to use one computer, the auto-logging program was unable to distinguish who sent the commands.

3. The collection of artefacts such as the posts, private messages, and the Logo procedures may be incomplete. The researcher had backup after each session. However, this couldn’t keep track of the process of the artefacts due to the participants’ deletion and modifications of those artefacts within a session. In this enactment, the valuable debugging processes in writing procedures were mostly not recorded. Again, this may rely on more video cameras or special software to record the full screen interaction.

On the learning activities

The nine learning activities were well sequenced in Iteration 2. However, these learning activities can be further classified into types of introductory, specific concepts, and application activities.

The learning activities of introductory type are essential for new users and beginners to the VRMath system. The first three of the nine learning activities belong to this type. The first three activities (i.e., Become a member, The secret and legend of the turtle, and Turtle dance) focus on and familiarise the users with the hypermidea and forum interface, VR interface, and programming interface respectively. They should be administered at the earliest stage to build up basic skills in communication (through forum), 3D navigation, and turtle graphics commands for later learning activities.
The learning activities of specific concepts type are those focuses on developing one or more specific geometrical concepts such as similarity of scaled shapes, patterns and relationships of repeated 3D objects, non-commutativity of 3D rotation, patterns of 2D polygons, and properties and processes of a cube. The learning activities from Learning Activity 4 to Learning Activity 8 are of this type. There may not have a learning sequence for this type of learning activities.

The Learning Activity 9 in Iteration 2 is the only application type of learning activity. This type of learning activities is for more advanced users, who are required to utilise their knowledge about the VRMath system and 3D geometry to construct a virtual world. The application type of learning activity can be a group project of 3D modelling for problem solving activities, and an assessment of learning and a more holistic usability testing of the VRMath system.

On limitations of the VRLE

There are currently two major limitations of the VRLE. The first limitation is bounded by the nature of the technology. From a semiotic point of view, the VRLE is rich in signs (representations) to signify the objects (3D geometry). The interpretant (learner) then tries to make meaning of 3D geometry concepts and processes through constant challenges and validations of multiple representations. The fundamental question is that the virtual space and manipulatives do not equate to the real space and objects. The learning within the VRLE depends on how a learner can transfer the knowledge from virtual to reality. Nevertheless, the VRLE’s 3D virtual space does have much potential for investigating 3D geometry. The computational power allows one to perform the impossible or dangerous actions in virtual space. We can envisage that as the technologies such as immersive VR become a common utility, the gap between virtual and reality will be minimised.

The second limitation is the software limitation for dissemination. The VRMath environment is available online for public access. However, the software required to use the system has stopped developing\(^{33}\). This increased the difficulty to disseminate the VRMath system. A major redesign to adopt other technology (software) will be the first priority in the next cycle of the design experiments.

\(^{33}\) The software required refered to Microsoft® Java Virtual Machine (MSJVM). In a legal litigation with Sun Microsystems Inc., Microsoft has agreed to stop supporting MSJVM from 31, December, 2007. For more information, please see [http://www.microsoft.com/mscorp/java/](http://www.microsoft.com/mscorp/java/).
10.3 REDESIGN

This section highlights the accumulated findings from this cycle of the design experiments that are relevant to the improvement of the learning activities and the VRLE. Concrete recommendations are presented for the learning activities and the VRLE.

Of the learning activities

Based on the results of the enactment and evaluation, each of the nine learning activities in Iteration 2 is refined to provide information about the: (a) big ideas, (b) focus of observation, and (c) key focus questions for future iterations. The information about the purpose, description, time, and evaluation of the nine learning activities presented in Section 4.5 remains valid. The big ideas refer to the concepts and processes of 3D geometry to be developed. The focus of observation highlights possible students’ interactions with the VRMath system that can be the indicators for learning and usability issues. The key focus questions are inquiries to scaffold the students’ learning and development of spatial abilities.

Learning Activity 1: Become a member

Big ideas: Learning within an online community.

Focus of observation: (a) faults in registration process, (b) maintenance of profiles such as avatar and signature, (c) utilisation of emoticons and rich texts.

Key focus questions: (a) When and what do you write in public forums and in private messages? (b) How should we communicate through the forums and private messages (network etiquette)? (c) Do you like using emoticons?

Learning Activity 2: The secret and the legend of the turtle

Big ideas: (a) 3D navigation modes are different (i.e., Examine, Fly, and Walk), (b) 3D navigation is a spatial orientation ability.

Focus of observation: (a) Use of navigation modes, (b) Use of the mouse and the keyboard, (c) Use of Navigation Toolbar.

Key focus questions: (a) Is the turtle moving? (b) How is it different or same about 3D navigation in virtual space and real world? (c) If you slide, do you change your direction? (d) What is your favourite navigation mode and why?
Learning Activity 3: Turtle dance

Big ideas: (a) Component movement in computer environment, (b) There are six turning commands in 3D, (c) egocentric and fixed moving are different.

Focus of observation: (a) Use of Quick Command, (b) Ego-centric bug, (c) Use of angles and distance for turning and moving commands, (d) 3D Navigation while turning the turtle, (e) use of REPEAT command.

Key focus questions: (a) Can you use your hand to do the six turns? (b) Does EAST command change the turtle’s direction? (c) If you turn the turtle, does the turtle change its position? (d) If you move the turtle, does the turtle change its direction? (e) What information do we need to tell the turtle for doing a repeat? (f) How to do a repeat within a repeat for a cool dance? (g) How to command to turtle to solve the secret and legend problems in previous activity?

Learning Activity 4: Shapes and scale

Big ideas: (a) Name 5 primitive shapes in VRMath, (b) Scale in 3 dimensions, (c) 2D projection when one dimension is 0 in scale, (d) similarity of shapes in different scales, (e) Congrence of 3D shapes in different orientation.

Focus of observation: (a) Use of Material Editor and other GUI tools, (b) 3D navigation to see 3D shapes from different perspectives, (c) Coordination of using the three interfaces, (d) Preference of using scaling commands?

Key focus questions: (a) What are the others names for box, ball, and can? (b) Can you predict when x dimension is 0, what 2D shape will you get for x shape? (c) What if there are two 0 dimensions? (d) What shapes do you see for an icecream cone? (e) Can you create a shape tower from a cube, cylinder, cone, sphere, and a label? (f) Are there any similar shapes in different scale? (g) How are the shapes different and similar?

Learning Activity 5: Climb up stairs

Big ideas: (a) Pattern in a staircase, (b) Repeat for a pattern.

Focus of observation: (a) Use of REPEAT command, what are repeated, (b) If scaling commands are used, how the distance to move changed in the command? (c) 3D navigation to walk up stairs, (d) Use of built-in ground for gravity effect, (e)
Coordination of using the three interfaces, (f) The turning degrees specified for spiral structures, (g) Any special number of repeat and turning degrees specified.

Key focus questions: (a) How is the location of a upper stair different to a lower stair? (b) What shapes can be used for stairs? (c) What are the dimensions of a real stair in centimeters? (d) How can the staircase be improved? (e) How can we build a spiral staircase? (f) What is it like to look down from top of the staircase?

Learning Activity 6: 3D rotation

Big ideas: Non-commutativity in 3D rotation

Focus of observation: (a) Use of body movement to simulate 3D rotation, (b) Use of Avatar View to view 3D rotation, (c) Turning the turtle to see 3D rotation, (d) Creating a 3D shape after each set of 3D rotation, (e) Incorrect use of 2D analogy to explain 3D rotation, (f) Incorrect use of moving on 2D plane analogy for turning in 3D.

Key focus questions: (a) Are LT 45 RU 45 the same as RU 45 and LT 45? (b) How about LT 30 RU 30 and RU 30 LT 30? (c) Why do two spheres created after each set of 3D rotation exactly overlap? (d) If you LT 45 RU 45 RT 45 RD 45, will you end up the original direction?

Learning Activity 7: Formula of polygon

Big ideas: An n-sided regular polygon can be created using the formula REPEAT n [fd 1 rt 360/n].

Focus of observation: (a) Use of body movement to walk a square, (b) Use of Quick Command for creating a square and a triangle, (c) Observe the degrees used for turning for a square and a triangle, (d) Use of a table to record results, (e) 3D navigation to see the polygons created, (f) Use of decimal number and operation in commands.

Key focus questions: (a) What is the degree to turn for walking out a square? (b) What about a triangle? (c) How to use a table to record the degrees for polygons? (d) Can you predict the degrees to turn for a pentagon? (e) How can we predict the degree? (f) What pattern did you see from the table? (g) What is the relationship between the number of sides and the degrees to turn? (h) What is the degree for 7 sided polygon, (i) What will be the shape if you turn 1 degree for 360 times?
Learning Activity 8: A frame of a cube

Big ideas: (a) A cube may not look like a cube from different perspectives, (b) A cube has 12 equal edges, (c) The angles between any two adjacent edges is 90 degrees.

Focus of observation: (a) Use of Quick Command, (b) Use of different frames of reference commands, (c) Egocentric bug, (d) Is 3D navigation performed during the construction of the cube? (e) Are there rectangular prisms created because of a static 3D viewpoint? (f) Use of Procedure Editor, (g) Debugging of procedure, (h) Saving projects.

Key focus questions: (a) What is the degree to turn for constructing a cube? (b) Is using Quick Command easier to create a cube? (c) Can you write a list of command for the cube without using the Quick Command tool? (d) How is your cube procedure different/the same to others? (e) What is the best procedure for a cube? (f) What if you start the cube procedure at a TL 45 orientation, will you get a cube? (g) Does your turtle go back to its starting location and direction after the cube procedure? (h) What if you repeat your cube procedure, what will you get?

Learning Activity 9: Group project

Big ideas: Teamwork on problem solving activity.

Focus of observation: (a) Review existing projects, (b) Questioning of new commands, (c) Discuss and look for real world topics, (d) assignment of individual tasks, (e) Drawing of 2D plans, (f) utilisation of other’s procedure, (g) Translation between 2D plans to 3D objects, (h) Logic flows of procedures, (i) Integration of individual tasks into the final project, (j) utilisation of prior knowledge about VRMath.

Key focus questions: (a) What real world objects or structures would you like to build? (b) Are there any similar objects (patterns) in the real world structure? (c) What global parameters need to be set for each individual tasks (e.g., scale and unit in metre)? (d) What 3D and 2D shapes can be utilised for your 3D structure? (e) What scales are needed for the 3D shapes? (f) Can REPEAT command be utilised? (g) How many procedures are needed? (h) How do you put all procedures together?
Of the VRLE

The redesign of the VRLE (VRMath) is still informed by the semiotic framework to provide rich semiotic resources (multiple representations and actions) for mathematical meaning making. The three interfaces that represent the three types of semiotic resources remain but need to be refined in the light of the findings from the enactment and evaluation and the emergence of new technologies. The refinement of the three interfaces involves mainly the usability improvement to enhance interaction, communication and thus learning, and the breakthrough of the technological limitations for dissemination of the VRLE. The refinement of the three interfaces presented here is, however, at a proposal stage. The actual implementation will be carried out in the next or future cycles of the design experiments.

VR interface

1. An animated camera will be designed to following the turtle’s orientation. This will have the potential to eliminate the egocentric bug.

2. The icons in the Navigation Toolbar will be reconsidered by the meaning emission capacity and by the users’ recommendations (see Figure 9.3).

3. The texturing of objects in 3D virtual space will be added into the Material Editor. This will enable the construction of more realistic 3D virtual worlds.

4. The behaviours such as rotation and movement will be enabled on any objects in the 3D virtual space. Thus the 3D virtual worlds can be dynamic rather than static. This will enable more investigations on 3D spatial abilities, and allow learners to express and communicate their ideas.

5. Alternative technologies for VR such as X3D34 and immersive VR will be sought. The use of wearable 3D input device will be considered depending on their availability. This, however, will require the coordination of the programming interface.

---

34 X3D is the new standard of 3D contents on web. It is the next generation of the VRML that is currently utilised for the VR interface. For more information please see [http://www.web3d.org/](http://www.web3d.org/).
Programming interface

1. The confusion of the commands such as TL (TileLeft) for TurnLeft and TR (TiltRight) for TurnRight will be addressed.

2. The programming commands will be extended to work with all the functionalities in VR interface. For example, there was no associate command for the Background Chooser in this prototype VRLE. The goal is to enable the use of the programming language to control and express everything in this VRMath environment.

3. The Quick Command has been redesigned (see Figure 9.4). However, it can be further improved if new technologies such as voice recognition and natural language processing could be employed.

4. The Logo-like programming language will be re-examined. Other syntactic and semantic forms or generations of programming language (see Section 2.4.4) will be considered.

5. Facilities such as procedure library and teamwork space will be designed to facilitate the reuse of procedures and the collaboration of group projects.

6. A step by step execution interface will be designed to facilitate the writing and debugging processes of procedures.

Hypermedia and forum interface

1. The forum has been redesigned to enable attachment of pictures and 3D virtual worlds created in VRMath. This will facilitate the communication of 3D geometry through the forum discussions.

2. Scaffolding facilities such as sub-forums and descriptors (e.g., praise, question, and proposition) will be improved to facilitate the knowledge building discourse.

3. An online chat room facility for synchronous communication may be designed to supplement the asynchronous communication of the Forum system.
10.4 CONCLUSIONS OF RESEARCH

In conclusion, the researcher wishes to present again the two main practical outcomes that have implications for both mathematics education and technology education. These two outcomes are: (a) a semiotic framework to inform the design of VRLEs, and (b) the VRMath system to facilitate the knowledge construction of 3D geometry concepts and processes.

The two outcomes have implications for both mathematics education and technology education in three levels. The first level is the theoretical level. The semiotic framework argues that learning mathematics is a meaning making activity, in which mathematical meaning emerges from the semiosis within multiple semiotic resources (i.e., topological, typological, and social-actional resources), and from which the meaning-making is a continuous process of constant challenging, revising, and inventing of existing and new meaning. This semiotic framework is thus consistent and interlinks with the fallibilist philosophy about the nature of mathematics, and the constructivist and constructionist view about learning. The second level is the content level. From the semiotic framework, the mathematical content such as 3D geometry can be examined and interpreted to form the external material world (i.e., objects), internal spatial abilities (i.e., interpretants), and communications (i.e., signs), which leads to in-depth analysis of the “reality”, and thus clarifies the goals and the contents of this research. The third level is the application level. When the semiotic framework is applied to the production of a computational learning environment such as a VRLE, it facilitates the design of the computational learning environment in terms of what technologies to adopt, the representations of knowledge, the processes of learning, and the interactions between human and computers.

The research study demonstrated the power and potential of the VRLE. It also has the following three implications for the use of ICTs (of technology education) in mathematics education at large.

1. The first implication goes back to the very basic idea, which is the use of ICTs for communication and expression about mathematics (Hoyles, 2001). That is, to design of computational learning environments should allow children to have freedom to express their own ideas, but constrain in ways so as to focus their attention on the mathematics.
2. The ICT tools should encourage new ways of thinking and doing of mathematics. This goes back to Pea’s (1985) idea about reorganising rather than just amplifying our mental functioning.

3. The third implication is that technological innovation is not just invention. The technologies utilised in the VRMath system were not new technologies, but they were used in a way that inhibits the essence of learning theories and philosophies to become a new paradigm of using computers.

Lastly, the discussion ends with the visions and directions for future research about VRLE. To be an innovation with sustainability, the VRMath system will need to further evolve in future cycles of the design experiments. It can be envisaged that the next cycle of the design-experiments will take into account of the future social and technical changes; the next “prototype” of the VRMath system will include more new technological innovations; the enactment and evaluation will still encompass dynamic and messy settings; and the reflection and redesign will generate more theoretical and practical insights in the fields of mathematics and technology education.
REFERENCES


(Eds.), *Intelligent learning environments: the case of geometry* (pp. 113-132). Berlin New York: Springer.


Thibault, M.-F., & Barre, R. L. (1996). Some hyperbolic geometry with Cabri-
Affairs Division. (Eds.), Intelligent learning environments: the case of
geometry (pp. 218-230). Berlin New York: Springer.

In D. A. Grouws (Ed.), Handbook of Research on Mathematics Teaching and


improvements in simulation-based training. Educational Technology, 34(8),
56-64.

from http://www.abo.fi/~atorn/FuncProg/Page11.html

Tracey, D., Perry, B., & Howard, P. (1998). Teacher beliefs about the learning and
teaching of mathematics: some comparisons. In C. Kanes, M. Goos & E.
Warren (Eds.), Teaching mathematics in new times: proceedings of the 21st
Annual Conference of the Mathematics Education Research Group of
Australasia (Vol. 2, pp. 613-620). Gold Coast, QLD: MERGA.


van Hiele, P. M. (1999). Developing geometric thinking through activities that begin
with play. Teaching Children Mathematics, 5(6), 310.

(Eds.), Learning and teaching mathematics: an international perspective
(pp. 5-28). Hove, East Sussex: Psychology Press.

Vincent, J. (1998). Progressing through the van Hiele levels with Cabri-géomètre. In
C. Kanes, M. Goos & E. Warren (Eds.), Teaching mathematics in new times:
proceedings of the 21st Annual Conference of the Mathematics Education
Research Group of Australasia (Vol. 2, pp. 645-652). Gold Coast, QLD:
MERGA.


Academic Publishers Professional.

from http://ceo.cudenver.edu/~brent_wilson/dangers.html


Appendix 1: Languages Introduced in Space/Geometry

<table>
<thead>
<tr>
<th>Year level</th>
<th>Languages</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>• Position, plane shapes, 3D shapes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Relational</th>
<th>Operational</th>
<th>Geometric</th>
<th>Positional</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>same/not the same</em></td>
<td><em>the same</em></td>
<td><em>Shape</em></td>
<td><em>in/out</em></td>
</tr>
<tr>
<td><em>same/different</em></td>
<td><em>sort</em></td>
<td><em>flat</em></td>
<td><em>outside/inside</em></td>
</tr>
<tr>
<td><em>big/little</em></td>
<td><em>fit</em></td>
<td><em>round/curved</em></td>
<td><em>over/under</em></td>
</tr>
<tr>
<td><em>large/small</em></td>
<td><em>join together</em></td>
<td><em>surface</em></td>
<td><em>above/below</em></td>
</tr>
<tr>
<td><em>top/bottom</em></td>
<td><em>fit together</em></td>
<td><em>face</em></td>
<td><em>on/off</em></td>
</tr>
<tr>
<td><em>around</em></td>
<td><em>match</em></td>
<td><em>edge</em></td>
<td><em>top/bottom</em></td>
</tr>
<tr>
<td><em>next to</em></td>
<td><em>cover</em></td>
<td><em>straight</em></td>
<td><em>same/different</em></td>
</tr>
<tr>
<td><em>top/bottom</em></td>
<td></td>
<td><em>corner</em></td>
<td>up/down</td>
</tr>
<tr>
<td><em>around</em></td>
<td></td>
<td><em>side</em></td>
<td>around</td>
</tr>
<tr>
<td><em>same/different</em></td>
<td></td>
<td><em>circle</em></td>
<td>underneath/on top of</td>
</tr>
<tr>
<td><em>more/most</em></td>
<td></td>
<td><em>triangle</em></td>
<td>first/last</td>
</tr>
<tr>
<td><em>less/least</em></td>
<td></td>
<td><em>rectangle</em></td>
<td>in front/behind</td>
</tr>
<tr>
<td><em>large/larger/largest</em></td>
<td><em>spread</em></td>
<td><em>square</em></td>
<td>beside/beside by</td>
</tr>
<tr>
<td><em>great/greater/greatest</em></td>
<td><em>cover</em></td>
<td><em>box shapes</em></td>
<td>side/alongside/next to</td>
</tr>
<tr>
<td><em>small/smaller/smallest</em></td>
<td><em>trace</em></td>
<td><em>ball shapes</em></td>
<td>before/after</td>
</tr>
<tr>
<td><em>whole/part/half</em></td>
<td></td>
<td><em>can shapes</em></td>
<td></td>
</tr>
</tbody>
</table>

• Line
Informal concept development and use of the word *line*.

2 • Position, plane shapes, 3D shapes

<table>
<thead>
<tr>
<th>Relational</th>
<th>Operational</th>
<th>Geometric</th>
<th>Positional</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>same/equal/different</em></td>
<td><em>repeat</em></td>
<td><em>fence</em></td>
<td><em>between</em></td>
</tr>
<tr>
<td><em>more/most</em></td>
<td><em>fit</em></td>
<td><em>rectangular</em></td>
<td><em>in front of/behind</em></td>
</tr>
<tr>
<td><em>less/least</em></td>
<td><em>together/overlap</em></td>
<td><em>circular</em></td>
<td><em>beside/beside by</em></td>
</tr>
<tr>
<td><em>large/larger/largest</em></td>
<td><em>spread</em></td>
<td><em>triangular</em></td>
<td><em>alongside/next to</em></td>
</tr>
<tr>
<td><em>great/greater/greatest</em></td>
<td><em>cover</em></td>
<td><em>space</em></td>
<td><em>before/after</em></td>
</tr>
<tr>
<td><em>small/smaller/smallest</em></td>
<td><em>trace</em></td>
<td><em>cube</em></td>
<td><em>in/out</em></td>
</tr>
<tr>
<td><em>whole/part/half</em></td>
<td><em>fold/unfold/folded</em></td>
<td><em>sphere</em></td>
<td><em>inside/outside</em></td>
</tr>
<tr>
<td><em>open/closed</em></td>
<td></td>
<td><em>cone</em></td>
<td><em>over/under</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>pyramid</em></td>
<td><em>above/below</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>side</em></td>
<td><em>on/off</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>prism</em></td>
<td><em>top/bottom</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>cylinder</em></td>
<td><em>up/down</em></td>
</tr>
</tbody>
</table>

• Line
Informal concept development and use of the word *line*.

3 • Position, plane shapes, 3D shapes
### Consolidate and extend to include:

- **outside/inside**
- **all over**
- **underneath**

### Consolidate and extend to include:

- **level**
- **point**
- **boundary**
- **base**

### Consolidate, extend to include symmetry language

**Line**

- line, level, horizontal, vertical, straight, curved

**Symmetry**

- fold in half, cannot be folded into half, fold line

**Angles**

- turns, corners, square corners, half turn, full turn

4. **Position, plane shapes, 3D shapes**

- polygons, quadrilaterals, nets, vertex

**Symmetry**

- symmetry, symmetrical, not symmetrical, no symmetry, reflection line

**Angles**

- angle, quarter turns, less/greater than a square/half/full corner/quarter turn

5. **Position, plane shapes, 3D shapes**

- pentagons, hexagons, heptagons, octagons, nonagons, decagons, undecagons/hendecagons, dodecagon, trapeziums, parallelograms, centre, diameter, radius, circumference, scalene, isosceles, right angle, equilateral, apex

**Line**

- parallel, intersect, intersecting, perpendicular, line, segment, oblique

**Symmetry**

- lines of symmetry, turn symmetry, turn symmetry of order 2

**Angles**

- degree, right angle, acute, obtuse, straight angle, vertex, ray, protractor

6. **Position, plane shapes, 3D shapes**

- congruent, similar, parallelism, symmetry

**Coordinates**

- ordered pairs, coordinates, points, regions, grid, axis, axes, y-axis, x-axis, origin, starting point

**Symmetry**

- flip, slide, turn

**Angles**

- reflex, full rotation

7. **Position, plane shapes, 3D shapes**

- bisect

**Symmetry**

- reflection, rotation, translation, asymmetrical, axis of symmetry, rotational symmetry of order 2

**Angles**

- bisect, bisection, intersection
Appendix 2: VRMath documentation pages in hyperlink

5. VRMath Forum: http://vrmath.yeh.id.au/forum/
6. VRMath documentations and papers:
11. Programming interface page:
Appendix 3: Focus group interview questions in Iteration 1

1. Do you like using VRMath?
2. Is VRMath easy to use?
3. Is 3D navigation in VRMath easy?
4. Is VRMath Forum easy to use?
5. Are VRMath commands easy to use and remember?
Appendix 4: Pre interview questions for Iteration 2

Background:
1. Math preference:

2. Computer Technoracy:
   Programming:
   Internet:
   3D Gaming:
   Online Community:

3. Other research experience:

Math Beliefs:
1. What is maths (example)?

2. Discovered or Invented?

3. Where to use maths?

Geometry:
1. 3D Shapes:
   Sphere
   Cone
   Cylinder
   Box/Cube
   Prism
   Pyramid

2. Properties:
   Vertex
   Edge
   Face

3. Angle:
   Triangle 180
   Right angle 90
   Circle 360

4. Perspective drawing of 3D shapes
Appendix 5: Post interview questions in Interaction 2

Questions:
1. Do you like using VRMath?
2. Is VRMath easy to use?
3. Is 3D navigation in VRMath easy?
4. Is VRMath Forum easy to use?
5. Are VRMath commands easy to use and remember?
6. What other aspects are easy or difficult in VRMath?
7. How can VRMath be improved?

Observations:
1. Learnability: Ability for users to learn the system easily.
2. Efficiency of use once the system has been learned: Ability for users to save time in their work once they’ve learned the system.
3. Memorability: Ability for users to come back to the system and remember how to use it once they’ve been away from it for some time.
4. Error recovery & prevention: When the system presents an error message to users, it gives enough information for them to be able to continue with their work. Better yet, the system helps to prevent errors.
5. Subjective user satisfaction: Users’ overall feelings about the system. Is it pleasant to use?
# Appendix 6: Navigation toolbar usability test

<table>
<thead>
<tr>
<th>Link icons and their functions</th>
<th>Name:</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Walk mode](Walk mode)</td>
<td>Walk mode</td>
</tr>
<tr>
<td>![Fly mode](Fly mode)</td>
<td>Fly mode</td>
</tr>
<tr>
<td>![Examine mode](Examine mode)</td>
<td>Examine mode</td>
</tr>
<tr>
<td>![Avatar / Free view](Avatar / Free view)</td>
<td>Avatar / Free view</td>
</tr>
<tr>
<td>![Set rotation centre](Set rotation centre)</td>
<td>Set rotation centre</td>
</tr>
<tr>
<td>![Horizontally straighten](Horizontally straighten)</td>
<td>Horizontally straighten</td>
</tr>
<tr>
<td>![Restore viewpoint](Restore viewpoint)</td>
<td>Restore viewpoint</td>
</tr>
<tr>
<td>![Fit Screen](Fit Screen)</td>
<td>Fit Screen</td>
</tr>
<tr>
<td>![Full screen of 3D space](Full screen of 3D space)</td>
<td>Full screen of 3D space</td>
</tr>
<tr>
<td>![On / Off headlight](On / Off headlight)</td>
<td>On / Off headlight</td>
</tr>
<tr>
<td>![On / Off collision mode](On / Off collision mode)</td>
<td>On / Off collision mode</td>
</tr>
<tr>
<td>![Show / Hide compass](Show / Hide compass)</td>
<td>Show / Hide compass</td>
</tr>
<tr>
<td>![Show / Hide grid](Show / Hide grid)</td>
<td>Show / Hide grid</td>
</tr>
<tr>
<td>![Solid mode](Solid mode)</td>
<td>Solid mode</td>
</tr>
<tr>
<td>![Wire frame mode](Wire frame mode)</td>
<td>Wire frame mode</td>
</tr>
<tr>
<td>![Show / Hide Chess board ground](Show / Hide Chess board ground)</td>
<td>Show / Hide Chess board ground</td>
</tr>
<tr>
<td>![Show / Hide sand ground](Show / Hide sand ground)</td>
<td>Show / Hide sand ground</td>
</tr>
<tr>
<td>![Show / Hide turtle](Show / Hide turtle)</td>
<td>Show / Hide turtle</td>
</tr>
<tr>
<td>![Show / Hide coordinate axis](Show / Hide coordinate axis)</td>
<td>Show / Hide coordinate axis</td>
</tr>
<tr>
<td>![Reduce / Enlarge forum frame](Reduce / Enlarge forum frame)</td>
<td>Reduce / Enlarge forum frame</td>
</tr>
<tr>
<td>![VRMath home](VRMath home)</td>
<td>VRMath home</td>
</tr>
<tr>
<td>![Previous page](Previous page)</td>
<td>Previous page</td>
</tr>
<tr>
<td>![Next page](Next page)</td>
<td>Next page</td>
</tr>
<tr>
<td><img src="Forum" alt="Forum" /></td>
<td>Forum</td>
</tr>
<tr>
<td>![Help index](Help index)</td>
<td>Help index</td>
</tr>
<tr>
<td>![Exit VRMath](Exit VRMath)</td>
<td>Exit VRMath</td>
</tr>
</tbody>
</table>
### Appendix 7: Public projects in VRMath created by the researcher

<table>
<thead>
<tr>
<th><strong>Project:</strong> Polyprism</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description:</strong></td>
<td>try: init Usage: polyprism :side :length :height example: cs meter pd polyprism 8.8 3 pu</td>
</tr>
<tr>
<td><strong>Procedures:</strong></td>
<td>TO polyprism :side :length :height repeat :side [fd :length ru 90 fd :height ru 90 fd :length ru 90 fd :height ru 90 jf :length rt 360/:side nc] END TO init cs meter pd</td>
</tr>
<tr>
<td><strong>Screenshot:</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Project:</strong> Street Lights</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description:</strong></td>
<td>Create a street with many street lights. try : init for a street light usage: streetlight scale ex: streetlight 1</td>
</tr>
<tr>
<td><strong>Procedures:</strong></td>
<td>TO init reset rt 90 fd 4 lt 90 bk 10 repeat 10 [streetlight 1 fd 10] bk 10 lt 90 fd 8 lt 90 repeat 10 [streetlight 1 fd 10] bk 10 lt 90 scale 1 1 1 setai 43</td>
</tr>
<tr>
<td><strong>Screenshot:</strong></td>
<td>setsh 190 settr 0 setdc 760 660 290 setec 180 160 70 setsc 360 380 60 face pcoff pd repeat 2 [fd 8 lt 90 fd 100 lt 90] pu home ru 90 fd 2.5 rd 90 setai 93 setsh 410 settr 450 setdc 370 0 750 setec 0 0 0 setsc 910 910 910 setfj &quot;middle label [Street Lights] fd 90 label [The End]</td>
</tr>
</tbody>
</table>
TO streetlight :s
; initial setting
pu
meter
; create pole
scale 0.1*:s 3*:s 0.1*:s
setai 140
setsh 900
settr 0
setdc 610 610 610
setec 0 0 0
setsc 770 760 760
ru 90
fd 1.5*:s
rd 90
cylinder
; create short pole
ru 90
fd 1.3*:s
rd 90
lt 90
scale 0.05*:s 0.4*:s 0.05*:s
fd 0.2*:s
rd 90
cylinder
ru 90
fd 0.15*:s
; create light cover
scale 0.5*:s 0.4*:s 0.5*:s
setai 40
setsh 70
settr 0
setdc 250 170 70
setec 130 130 130
setsc 550 260 260
rd 90
fd 0.1*:s
ru 90
cone
rd 90
fd 0.2*:s
ru 90
; create light
scale 0.2*:s 0.25*:s 0.2*:s
setai 0
setsh 50
settr 0
setdc 290 290 0
setec 320 320 0
setsc 1000 1000 390
ball
; turtle go back
ru 90
fd 0.3*:s
rd 90
### Project: Symmetric Tree

**Description:**
try command:
init
tree, Cheers,

**Procedures:**
TO tree :size
IF :size < 5 [STOP]
FD :size
LT 45 tree :size*.7
RT 90 tree :size*.7
LT 45
JB :size
END

TO init
setpc 0
setwait 20
cs

### Project: Recursive random tree

**Description:**
try: init
A procedure named rtree is included in this project.
After opening this project, give the following command:
cmp
ru 90
pd
rtree 80
pu

**Procedures:**
TO rtree :size
; Recursive random tree
if :size < 5 [stop]
Project: Random 3D Tree

Description:
try command:
init
cheers,

Procedures:
TO tree :size
localmake "tt random 6
localmake "de random 45
IF :size < 5 [STOP]
FD :size
ifelse :tt=0 [lt :de] [
ifelse :tt=1 [rt :de] [
ifelse :tt=2 [ru :de] [
ifelse :tt=3 [rd :de] [
ifelse :tt=4 [tl :de] [
ifelse :tt=5 [tr :de] [
]]]]]
tree :size*.7
ifelse :tt=0 [rt :de*2] [
ifelse :tt=1 [lt :de*2] [
ifelse :tt=2 [ru :de*2] [
ifelse :tt=3 [rd :de*2] [
ifelse :tt=4 [tl :de*2] [
ifelse :tt=5 [tr :de*2] [
]]]]]
JB :size
END

TO init
setpc 0
setwait 20
cs
cm
ru 90
pd
tree 100
pu
END
Appendix 8: Users' projects in VRMath

<table>
<thead>
<tr>
<th>Project: Wierd World!!!</th>
<th>Author: Bonbon</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description:</strong></td>
<td>TO WierdWorld</td>
</tr>
<tr>
<td>I Hope you try My Wierd world That I made!!</td>
<td>repeat 24 [Cubonbon]</td>
</tr>
<tr>
<td>This is the command : WierdWorld</td>
<td>rt 90</td>
</tr>
<tr>
<td></td>
<td>repeat 24 [Cubonbon]</td>
</tr>
<tr>
<td></td>
<td>randomball 100</td>
</tr>
<tr>
<td></td>
<td>setai 0</td>
</tr>
<tr>
<td></td>
<td>setsh 30</td>
</tr>
<tr>
<td></td>
<td>sett 0</td>
</tr>
<tr>
<td></td>
<td>setdc 0 0 0</td>
</tr>
<tr>
<td></td>
<td>setec 0 1000 1000</td>
</tr>
<tr>
<td></td>
<td>setsc 900 900 900</td>
</tr>
<tr>
<td></td>
<td>randomball 100</td>
</tr>
<tr>
<td></td>
<td>setai 0</td>
</tr>
<tr>
<td></td>
<td>setsh 50</td>
</tr>
<tr>
<td></td>
<td>sett 0</td>
</tr>
<tr>
<td></td>
<td>setdc 0 0 0</td>
</tr>
<tr>
<td></td>
<td>setec 490 0 1000</td>
</tr>
<tr>
<td></td>
<td>setsc 620 620 620</td>
</tr>
<tr>
<td></td>
<td>randomball 100</td>
</tr>
<tr>
<td></td>
<td>setai 0</td>
</tr>
<tr>
<td></td>
<td>setsh 30</td>
</tr>
<tr>
<td></td>
<td>sett 0</td>
</tr>
<tr>
<td></td>
<td>setdc 0 0 0</td>
</tr>
<tr>
<td></td>
<td>setec 0 0 1000</td>
</tr>
<tr>
<td></td>
<td>setsc 0 920 1000</td>
</tr>
<tr>
<td></td>
<td>randomball 50</td>
</tr>
<tr>
<td></td>
<td>setai 0</td>
</tr>
<tr>
<td></td>
<td>setsh 50</td>
</tr>
<tr>
<td></td>
<td>sett 0</td>
</tr>
<tr>
<td></td>
<td>setdc 0 0 0</td>
</tr>
<tr>
<td></td>
<td>setec 740 1000 90</td>
</tr>
<tr>
<td></td>
<td>setsc 800 780 770</td>
</tr>
<tr>
<td></td>
<td>END</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Procedures:</th>
</tr>
</thead>
<tbody>
<tr>
<td>TO Cubonbon</td>
</tr>
<tr>
<td>pd</td>
</tr>
<tr>
<td>repeat 4 [fd 1 rt 90]</td>
</tr>
<tr>
<td>ru 90</td>
</tr>
<tr>
<td>fd 1</td>
</tr>
<tr>
<td>rd 90</td>
</tr>
<tr>
<td>repeat 4 [fd 1 rt 90]</td>
</tr>
<tr>
<td>rt 90</td>
</tr>
<tr>
<td>fd 1</td>
</tr>
<tr>
<td>rd 90</td>
</tr>
<tr>
<td>lt 90</td>
</tr>
<tr>
<td>fd 1</td>
</tr>
<tr>
<td>rd 90</td>
</tr>
<tr>
<td>fd 1</td>
</tr>
<tr>
<td>nc</td>
</tr>
<tr>
<td>pu</td>
</tr>
<tr>
<td>END</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TO randomball :number</th>
</tr>
</thead>
<tbody>
<tr>
<td>repeat :number [</td>
</tr>
<tr>
<td>; set random position</td>
</tr>
<tr>
<td>setxyz (random 20)-10 (random 20)-10</td>
</tr>
<tr>
<td>random 20)-10</td>
</tr>
<tr>
<td>; set random diffuse color</td>
</tr>
<tr>
<td>setdc (random 1000) 0 0</td>
</tr>
<tr>
<td>; set random transparency</td>
</tr>
<tr>
<td>settr (random 1000)</td>
</tr>
<tr>
<td>; create a ball</td>
</tr>
<tr>
<td>ball</td>
</tr>
<tr>
<td>]</td>
</tr>
<tr>
<td>END</td>
</tr>
</tbody>
</table>

**Screenshot:**

![Screenshot of Wierd World!! project](image-url)
**Project:** For the house project

**Description:**
I made a cooktop!
And A Bed!!!

**Procedures:**

| TO Cooktop | reset | setai 0 | setsh 30 | settr 0 | setdc 0 0 0 | setec 0 1000 1000 | setsc 900 900 900 | cube | up 0.5 | scale 0.9 0.09 0.9 | setai 0 | setsh 50 | settr 0 | setdc 0 0 0 | setec 490 0 1000 | setsc 620 620 620 | box | lt 90 | fd 0.5 | rt 90 | fd 0.5 | setai 137 | setsh 170 | settr 0 | setdc 570 580 580 | setec 0 0 0 | setsc 670 460 440 | east 0.3 | bk 0.3 | repeat 4 [scale 0.3 0.1 0.3 cylinder rt 90 fd 0.4] | END |
| TO bed | setai 0 | setsh 150 | settr 0 | setdc 1000 1000 1000 | setec 1000 1000 1000 | setsc 1000 1000 1000 | scalew 1.2 scaled 0.5 scaleh 0.2 | box | up 0.1 | scalew 0.5 scaled 0.01 scaleh 0.8 | setai 0 | setsh 30 | settr 0 | setdc 0 0 0 | setec 0 0 1000 | setsc 0 920 1000 | rt 90 | fd 0.2 | rd 90 | box | ru 90 | bk 0.6 | scalew 0.4 scaled 0.3 scaleh 0.2 | lt 180 | setai 0 | setsh 30 | settr 0 | setdc 0 0 0 | setec 0 1000 1000 | setsc 900 900 900 | box | END | TO householditems | cooktop | fd 2 | bed | END |

**Screenshot:**

![Screenshot of the project]
**Project:** Ninja's House 1  

**Description:**
The Big Project!!!!!!!!!!

**Procedures:**

```
TO wall_2
setai 0
setsh 150
settr 0
setdc 600 0 170
setec 290 0 80
setsc 500 0 140
home
fd 2 ru 90 face pcoff meter
pd repeat 2 [fd 4 rt 90 fd 5 rt 90]
jf 4 rt 90 jf 5
repeat 2 [fd 2 rt 90 fd 1.5 rt 90]
jf 2 repeat 2 [fd 5 rt 90 fd 4 rt 90]
pu
END

TO wall_1
face pcoff
setai 0
setsh 150
settr 0
setdc 6005 0 170
setec 290 0 80
setsc 500 0 140
pd
fd 4 rt 90 fd 2.5 rt 90 fd 4 rt 90 fd 2.5 rt 90
rt 90 jf 5.5 lt 90
fd 4 rt 90 fd 2.5 rt 90 fd 4 rt 90 fd 2.5 rt 90 jf
0 fd 1.5 lt 90 fd 3 lt 90 fd 1.5 lt 90 fd 3
lt 90 jf 3 fd 1 lt 90 fd 3 lt 90 fd 1 lt 90 fd 3 lt
90
pu
setai 63
setsh 60
settr 500
setdc 0 0 750
setec 0 0 0
setsc 0 840 880
pd
lt 90 fd 3 lt 90 fd 1.5 lt 90 fd 3 lt 90 fd 1.5 pu
END

TO roof
scale 0.4 2 2
cone
END
```
TO hbody
scale 1 1 1 pu
home meter
fd 10 rt 90 ru 90 face pcoff
wall_1
home
east 12 fd 10 rt 90 ru 90
wall_1
wall_2
wall_3
roof
END

TO pool
home scale 1 1 1 face
meter pcoff
setai 1000
setsh 1000
settr 154
setdc 0 0 0
setec 900 900 900
south 6 west 10 up 0.01
pd
repeat 2 [fd 12 rt 90 fd 5 rt 90]
pu
END

TO fence :side :length :height
repeat :side [fd :length ru 90 fd :height ru 90
fd :length ru 90 fd :height ru 90 jf :length rt 360/:side nc]
pd
END

TO ground
home scale 1 1 1 meter
bk 20 west 20 face pcoff
setai 0
setsh 0
settr 0
setdc 0 0 0
setec 148 787 296
setsc 0 0 0 pd
repeat 4 [fd 40 rt 90]
pu
END

TO tree :size
localmake "tt random 6
localmake "de random 60
IF :size < 5 [STOP]
FD :size nc
ifelse :tt=0 [lt :de] [ifelse :tt=1 [rt :de] [ifelse :tt=2 [ru :de] [ifelse :tt=3 [rd :de] [ifelse :tt=4 [tl :de] [ifelse :tt=5 [tr :de] []]]]]]
tree :size*(((random 5)+5)/10)
ifelse :tt=0 [rt :de*2] [ifelse :tt=1 [lt :de*2] [ifelse :tt=2 [rd :de*2] [ifelse :tt=3 [ru :de*2] [ifelse :tt=4 [tr :de*2] [ifelse :tt=5 [tl :de*2] []]]]]]
tree :size*(((random 5)+5)/10)
ifelse :tt=0 [lt :de] [ifelse :tt=1 [rt :de] [ifelse :tt=2 [ru :de] [ifelse :tt=3 [rd :de] [ifelse :tt=4 [tl :de] [ifelse :tt=5 [tr :de] []]]]]]
JB :size nc
END

Project: Ninja's House by grae

Description:
The Big Project!!!!!!!!!!!!

Procedures:

TO pool
home scale 1 1 1 face
meter pcoff
setai 1000
setsh 1000
settr 154
setdc 0 0 0
setec 900 900 900
south 6 west 10 up 0.01
pd
repeat 2 [fd 12 rt 90 fd 5 rt 90]
pu
END

TO fence :side :length :height
repeat :side [fd :length ru 90 fd :height ru 90
fd :length ru 90 fd :height ru 90 jf :length rt 360/:side nc]
pd
END

TO ground
home scale 1 1 1 meter
bk 20 west 20 face pcoff
setai 0
setsh 0
settr 0
setdc 0 0 0
setec 148 787 296
setsc 0 0 0 pd
repeat 4 [fd 40 rt 90]
pu
END

TO tree :size
localmake "tt random 6
localmake "de random 60
IF :size < 5 [STOP]
FD :size nc
ifelse :tt=0 [lt :de] [ifelse :tt=1 [rt :de] [ifelse :tt=2 [ru :de] [ifelse :tt=3 [rd :de] [ifelse :tt=4 [tl :de] [ifelse :tt=5 [tr :de] []]]]]]
tree :size*(((random 5)+5)/10)
ifelse :tt=0 [rt :de*2] [ifelse :tt=1 [lt :de*2] [ifelse :tt=2 [rd :de*2] [ifelse :tt=3 [ru :de*2] [ifelse :tt=4 [tr :de*2] [ifelse :tt=5 [tl :de*2] []]]]]]
tree :size*(((random 5)+5)/10)
ifelse :tt=0 [lt :de] [ifelse :tt=1 [rt :de] [ifelse :tt=2 [ru :de] [ifelse :tt=3 [rd :de] [ifelse :tt=4 [tl :de] [ifelse :tt=5 [tr :de] []]]]]]
JB :size nc
END
<table>
<thead>
<tr>
<th>Project: Temple Base (By Ali)</th>
<th>Author: Alekat20</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description:</strong></td>
<td>setr 0 setdc 0 10 500 setec 0 0 0 setsc 0 910 10 repeat 9 [fd 1.1 box fd 1.1 rt 360/18 fd 2.2 rt 360/18] END</td>
</tr>
<tr>
<td><strong>Procedures:</strong></td>
<td>TO tp3 templebase END</td>
</tr>
<tr>
<td>TO init</td>
<td><strong>Screenshot:</strong></td>
</tr>
<tr>
<td>pu ground</td>
<td></td>
</tr>
<tr>
<td>home west 20</td>
<td></td>
</tr>
<tr>
<td>face pcon pd fence 40 3 1.6 pu pool home cm line south 600 west 1100 ru 90 repeat 4 [pd tree 100 pu rd 90 fd 400 ru 90] END</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Project: VAM Temple 1</th>
<th>Author: R2D2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description:</strong></td>
<td>setr 0 setdc 0 10 500 setec 0 0 0 setsc 0 910 10 repeat 9 [fd 1.1 box fd 1.1 rt 360/18 fd 2.2 rt 360/18] END</td>
</tr>
<tr>
<td>This Is Some Of The Final Part</td>
<td><strong>Screenshot:</strong></td>
</tr>
</tbody>
</table>
### Procedures:

TO tp1
body
home south 3 fountain
END

TO body
setai 53
setsh 50
settr 0
setdc 350 160 120
setec 0 0 0
setsc 210 110 0
scale 15 0.8 6
meter
home
fd 7 box
repeat 4 [up 0.4 fd 0.25 scaled 6-
repcount*0.5 box]
up 6.2
scale 8 2.6
setai 117
setsh 400
settr 0
setdc 800 510 90
setsc 920 430 10
cone
scale 2 2 2 setfj "middle
setai 0
setsh 80
settr 0
setdc 0 0 0
setsc 0 0 0
setsc 290 300 290
up 1 label [VAM Temple] dn 1
scale 0.5 5 0.5
dn 6 east 2.5 south 1.5 up 2.5
repeat 2 [cylinder fd 3 lt 90 cylinder fd 5 lt 90]
scale 0.6 0.5 0.6
dn 2.25
setai 0
setsh 0
settr 0
setdc 800 800 800
setec 0 0 0
setsc 0 0 0
repeat 2 [cylinder fd 3 lt 90 cylinder fd 5 lt 90]
up 4.5
repeat 2 [cylinder fd 3 lt 90 cylinder fd 5 lt 90]
END

TO fountain
setai 0
setsh 180
settr 0
setdc 0 0 0
setec 0 0 0
setsc 720 720 720
scale 1.5 0.5 1.5 cylinder
up 1.5 scale 0.4 3 0.4 cylinder
up 1.5 scale 1.5 0.3 1.5
setai 36
setsh 110
settr 0
setdc 360 280 30
setec 0 0 0
setsc 740 800 330
scale 2 2 2 up 0.5 cone
setai 33
setsh 540
settr 0
setdc 20 240 530
setec 10 120 270
setsc 320 400 400
dn 0.5 north 0.6 scale 0.1 3.3 0.1 dn 1.5
cylinder south 1.2 cylinder
north 0.6 east 0.6 cylinder
west 1.2 cylinder
east 0.6 rt 45 fd 0.6 cylinder
rt 90 cylinder
rt 90 fd 0.6 lt 90 fd 0.6 cylinder
rt 180 fd 0.6 lt 90 fd 0.6 cylinder
rt 180 fd 0.6 lt 90 fd 0.6 cylinder
up 3 rt 180 fd 0.6 mks
END

TO mks
scale 1 1 1
rt 90 ru 60
setai 0
setsh 180
settr 0
setdc 0 0 0
setec 0 0 0
setsc 720 720 720
ball
setai 0
setsh 0
settr 0
setdc 800 800 800
setec 0 0 0
setsc 0 0 0
ru 30 rt 50
ball
END
**Project:** VAM Temple 2 real

**Author:** Victor

**Description:**
This Is Some Of The Final Part

**Procedures:**

```plaintext
TO bridge
scale 1 1 1
meter
face
setai 56
setsh 100
settr 0
setdc 310 200 160
setec 90 90 90
setsc 200 100 110
pd
repeat 4 [fd 2 rt 90]
pu
END

TO tree :s
meter scale 0.2*:s 2*:s 0.2*:s
setai 60
setsh 930
settr 0
setdc 510 170 60
setec 0 0 0
setsc 530 530 530
up 1*:s cylinder
scale 1.5*:s 2.2*:s 1.5*:s
setai 63
setsh 90
settr 0
setdc 110 550 90
setec 0 0 0
setsc 120 60 60
up 1.1*:s cone
dn 2.1*:s
END
```

```plaintext
TO ground
scale 1 1 1
meter
home
south 11 west 11
face pcoff
setai 0
setsh 0
settr 0
setdc 0 0 0
setec 812 812 821
setsc 0 0 0
pd repeat 4 [fd 22 rt 90] pu
south 2 west 2 down 1
setai 63
setsh 60
settr 500
setdc 0 0 750
setec 0 0 0
setsc 0 840 880
pd repeat 4 [fd 26 rt 90] pu
up 1 south 7 west 7
setai 0
setsh 200
settr 0
setdc 0 490 0
setec 0 150 0
setsc 500 500 500
pd repeat 2 [fd 40 rt 90 fd 7 rt 90] pu
rt 90
fd 7
pd repeat 2 [fd 26 lt 90 fd 7 lt 90] pu
fd 26
lt 90
pd repeat 2 [fd 40 rt 90 fd 7 rt 90] pu
fd 40 lt 90
pd repeat 2 [fd 26 lt 90 fd 7 lt 90] pu
END
```
TO tp2
home
ground
home south 13 west 1 bridge
home west 13 north 1 rt 90 bridge
home east 13 south 1 lt 90 bridge
home north 11 west 1 bridge
home south 18 west 18
repeat 19 [tree (((random 10)+1)/10)+1 fd 2]
  bk 2 rt 90
repeat 18 [fd 2 tree (((random 10)+1)/10)+1]
  rt 90
repeat 18 [fd 2 tree (((random 10)+1)/10)+1]
  rt 90
repeat 18 [fd 2 tree (((random 10)+1)/10)+1]
END
Appendix 9: Final group projects in VRMath

<table>
<thead>
<tr>
<th>Project: Ninja's House</th>
<th>By: School C</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description:</strong></td>
<td></td>
</tr>
<tr>
<td>The Big Project!!!!!!!</td>
<td></td>
</tr>
<tr>
<td>try: hbody</td>
<td></td>
</tr>
<tr>
<td><strong>Procedures:</strong></td>
<td></td>
</tr>
<tr>
<td>TO wall_2</td>
<td></td>
</tr>
<tr>
<td>setai 0</td>
<td></td>
</tr>
<tr>
<td>setsh 150</td>
<td></td>
</tr>
<tr>
<td>settr 0</td>
<td></td>
</tr>
<tr>
<td>setdc 600 0 170</td>
<td></td>
</tr>
<tr>
<td>setsc 500 0 140</td>
<td></td>
</tr>
<tr>
<td>home</td>
<td></td>
</tr>
<tr>
<td>fd 2 rt 90 face pcloff meter</td>
<td>setai 0</td>
</tr>
<tr>
<td>pd repeat 2 [fd 4 rt 90 fd 5 rt 90]</td>
<td>setsh 100</td>
</tr>
<tr>
<td>jf 4 rt 90 jf 5</td>
<td></td>
</tr>
<tr>
<td>repeat 2 [fd 2 rt 90 fd 1.5 rt 90]</td>
<td>settr 0</td>
</tr>
<tr>
<td>jf 2 repeat 2 [fd 5 rt 90 fd 4 rt 90]</td>
<td>setdc 0 0 0</td>
</tr>
<tr>
<td>pu</td>
<td></td>
</tr>
<tr>
<td>END</td>
<td></td>
</tr>
<tr>
<td>TO pool</td>
<td></td>
</tr>
<tr>
<td>home</td>
<td></td>
</tr>
<tr>
<td>scale 1 1 1 face</td>
<td></td>
</tr>
<tr>
<td>meter</td>
<td></td>
</tr>
<tr>
<td>setai 1000</td>
<td></td>
</tr>
<tr>
<td>setsh 1000</td>
<td></td>
</tr>
<tr>
<td>settr 154</td>
<td></td>
</tr>
<tr>
<td>setdc 0 0 0</td>
<td></td>
</tr>
<tr>
<td>setec 0 1000 1000</td>
<td></td>
</tr>
<tr>
<td>setsc 900 900 900</td>
<td></td>
</tr>
<tr>
<td>south 6 west 10 up 0.01</td>
<td></td>
</tr>
<tr>
<td>pd</td>
<td></td>
</tr>
<tr>
<td>repeat 2 [fd 12 rt 90 fd 5 rt 90]</td>
<td>setai 137</td>
</tr>
<tr>
<td>pu</td>
<td></td>
</tr>
<tr>
<td>END</td>
<td></td>
</tr>
<tr>
<td>TO wall_1</td>
<td></td>
</tr>
<tr>
<td>face pcloff</td>
<td></td>
</tr>
<tr>
<td>setai 0</td>
<td></td>
</tr>
<tr>
<td>setsh 150</td>
<td></td>
</tr>
<tr>
<td>settr 0</td>
<td></td>
</tr>
<tr>
<td>setdc 600 0 170</td>
<td></td>
</tr>
<tr>
<td>setec 290 0 80</td>
<td></td>
</tr>
<tr>
<td>setsc 500 0 140</td>
<td></td>
</tr>
<tr>
<td>pd</td>
<td></td>
</tr>
<tr>
<td>fd 4 rt 90 fd 2.5 rt 90 fd 4 rt 90 fd 2.5 rt 90</td>
<td>setai 170</td>
</tr>
<tr>
<td>rt 90 jf 5.5 lt 90</td>
<td></td>
</tr>
<tr>
<td>fd 4 rt 90 fd 2.5 rt 90 fd 4 rt 90 fd 2.5 rt 90</td>
<td>settr 0</td>
</tr>
<tr>
<td>END</td>
<td></td>
</tr>
<tr>
<td>TO Cooktop :s</td>
<td></td>
</tr>
<tr>
<td>scale 1*:s 1*:s 1*:s</td>
<td></td>
</tr>
<tr>
<td>meter</td>
<td></td>
</tr>
<tr>
<td>setai 0</td>
<td></td>
</tr>
<tr>
<td>setsh 30</td>
<td></td>
</tr>
<tr>
<td>settr 0</td>
<td></td>
</tr>
<tr>
<td>setec 0 0 0</td>
<td></td>
</tr>
<tr>
<td>setdc 900 900 900</td>
<td></td>
</tr>
<tr>
<td>setsc 620 620 620</td>
<td></td>
</tr>
<tr>
<td>box</td>
<td></td>
</tr>
<tr>
<td>lt 90</td>
<td></td>
</tr>
<tr>
<td>fd 0.5*:s</td>
<td></td>
</tr>
<tr>
<td>rt 90</td>
<td></td>
</tr>
<tr>
<td>fd 0.5*:s</td>
<td></td>
</tr>
<tr>
<td>setai 137</td>
<td></td>
</tr>
<tr>
<td>setsh 170</td>
<td></td>
</tr>
<tr>
<td>settr 0</td>
<td></td>
</tr>
<tr>
<td>setdc 570 580 580</td>
<td></td>
</tr>
<tr>
<td>setec 0 0 0</td>
<td></td>
</tr>
<tr>
<td>setsc 670 460 440</td>
<td></td>
</tr>
<tr>
<td>east 0.3*:s</td>
<td></td>
</tr>
<tr>
<td>bk 0.3*:s</td>
<td></td>
</tr>
<tr>
<td>scale 0.3*:s 0.1*:s 0.3*:s</td>
<td>repeat 4 [cylinder rt 90 fd 0.4*:s]</td>
</tr>
<tr>
<td>END</td>
<td></td>
</tr>
<tr>
<td>TO fence :side :length :height</td>
<td></td>
</tr>
<tr>
<td>-------------------------------</td>
<td></td>
</tr>
<tr>
<td>repeat :side [fd :length ru 90 fd :height ru 90 fd :length ru 90 fd :height ru 90 jf :length rt 360/:side nc]</td>
<td></td>
</tr>
<tr>
<td>END</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TO householditems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home</td>
</tr>
<tr>
<td>fd 9.5 east 0.5 up 0.5</td>
</tr>
<tr>
<td>cooktop 1</td>
</tr>
<tr>
<td>home</td>
</tr>
<tr>
<td>east 12-1.5/2 fd 10-1.8 rt 90</td>
</tr>
<tr>
<td>up 0.3</td>
</tr>
<tr>
<td>bed 3</td>
</tr>
<tr>
<td>home</td>
</tr>
<tr>
<td>fd 6</td>
</tr>
<tr>
<td>rt 90</td>
</tr>
<tr>
<td>fd 6</td>
</tr>
<tr>
<td>table</td>
</tr>
<tr>
<td>END</td>
</tr>
</tbody>
</table>

| TO bed :s                     |
| setai 0                       |
| setsh 150                     |
| settr 0                       |
| setdc 1000 1000 1000          |
| setec 1000 1000 1000          |
| setsc 1000 1000 1000          |
| scalew 1.2*:s scaled 0.5*:s scaleh 0.2*:s box up 0.1*:s |
| scalew 0.5*:s scaled 0.01*:s scaleh 0.8*:s setai 0 |
| setsh 30                      |
| settr 0                       |
| setdc 0 0 0                   |
| setec 0 0 1000                 |
| setsc 0 920 1000               |
| rt 90                         |
| fd 0.2*:s                     |
| rd 90                         |
| box                           |
| ru 90                         |
| bk 0.6*:s                     |
| scalew 0.4*:s scaled 0.3*:s scaleh 0.2*:s lt 180 |
| setai 0                       |
| setsh 30                      |
| settr 0                       |
| setdc 0 0 0                   |
| setec 0 1000 1000              |
| setsc 900 900 900              |
| box                           |
| END                           |

| TO table                      |
| scale 2.5 0.2 1.5             |
| up 1                          |
| setai 445                     |
| setsh 100                     |
| settr 0                       |
| setdc 350 160 120             |
| setec 0 0 0                   |
| setsc 210 110 0               |
| box                           |
| dn 0.5                        |
| bk 1.5/2                      |
| rt 90 fd 2.5/2 lt 90          |
| scaleh 1.2                    |
| scalew 0.2                    |
| scaled 0.2                    |
| rt 180                        |
| setai 66                      |
| setsh 20                      |
| settr 0                       |
| setdc 340 210 240             |
| setec 10 10 10                 |
| setsc 200 80 140              |
| repeat 2 [can rt 90 fd 2.3 can rt 90 fd 1.5] |
| END                           |

| TO hbody                      |
| Init                          |
| scale 1 1 1 pu                |
| home                          |
| meter                         |
| fd 10 rt 90 ru 90 face pcoff  |
| wall_1                        |
| home                          |
| east 12 fd 10 rt 90 ru 90     |
| wall_1                        |
| wall_2                        |
| wall_3                        |
| householditems                |
| setai 0                       |
| setsh 30                      |
| settr 0                       |
| setdc 0 0 0                   |
| setec 0 1000 1000             |
| setsc 0 920 1000               |
| roof                          |
| END                           |

| TO roof                       |
| Home                          |
| fd 6 rt 90 fd 6 lt 90 up 6.4  |
| scale 17 5 12                 |
| cone                          |
| END                           |
TO tree :size
localmake "tt random 6
localmake "de random 60
IF :size < 5 [STOP]
FD :size nc
ifelse :tt=0 [lt :de] [
ifelse :tt=1 [rt :de] [
ifelse :tt=2 [ru :de] [
ifelse :tt=3 [rd :de] [
ifelse :tt=4 [tl :de] [
ifelse :tt=5 [tr :de] [
]]]]
tree :size*(((random 5)+5)/10)
ifelse :tt=0 [rt :de*2] [
ifelse :tt=1 [lt :de*2] [
ifelse :tt=2 [rd :de*2] [
ifelse :tt=3 [ru :de*2] [
ifelse :tt=4 [tr :de*2] [
ifelse :tt=5 [tl :de*2] [
]]]]]
JB :size nc
END

TO wall_3
Home
fd 10
ru 90
setai 0
setsh 150
settr 0
setdc 600 0 170
setec 290 0 80
setsc 500 0 140
pd
repeat 2 [fd 4 rt 90 fd 2 rt 90] rt 90 jf 2
repeat 2 [fd 2 lt 90 fd 1.5 lt 90] jf 2
repeat 2 [fd 2 lt 90 fd 4 lt 90] lt 90 jf 3
repeat 2 [fd 1 lt 90 fd 2 lt 90] pu
setai 63
setsh 60
settr 500
setdc 0 0 1000
setec 148 787 296
setsc 0 840 880
pd lt 90
repeat 4 [fd 40 rt 90] pu
END

TO ground
home scale 1 1 1 meter
bk 20 west 20 face pcoff
setai 0
setsh 0
settr 0
setdc 0 0 0
setec 148 787 296
setsc 0 0 0
pd
repeat 4 [fd 40 rt 90] pu
END

TO init
pu ground
home west 20
face pcon pd fence 40 3 1.6 pu
pool home
cm line
south 600 west 1100 ru 90
repeat 4 [pd tree 100 pu rd 90 fd 400 ru 90] pu
END
**Project:** VAM Temple

**Description:**
This is our final product....... Try Command : VAM

**Procedures:**

TO bridge
scale 1 1 1
meter
face
setai 56
setsh 100
settr 0
setdc 310 200 160
setec 90 90 90
setsc 200 100 110
pd
repeat 4 [fd 2 rt 90]
pu
END

TO fountain
setai 0
setsh 180
settr 0
setdc 0 0 0
setec 0 0 0
setsc 720 720 720
scale 1.5 0.5 1.5 cylinder
up 1.5 scale 0.4 3 0.4 cylinder
up 1.5 scale 1.5 0.3 1.5
setai 36
setsh 110
settr 0
setdc 360 280 30
setec 0 0 0
setsc 740 800 330
scale 2 2 2 up 0.5 cone
setai 33
setsh 540
settr 0
setdc 20 240 530
setec 10 120 270
setsc 320 400 400
dn 0.5 north 0.6 scale 0.1 3.3 0.1 dn 1.5
cylinder south 1.2 cylinder
north 0.6 east 0.6 cylinder
west 1.2 cylinder
east 0.6 rt 45 fd 0.6 cylinder
rt 90 cylinder
rt 90 fd 0.6 lt 90 fd 0.6 cylinder
rt 180 fd 0.6 lt 90 fd 0.6 cylinder
rt 180 fd 0.6 lt 90 fd 0.6 cylinder
up 3 rt 180 fd 0.6 mks
END
TO vam
  cs
  tp1
  tp2
  tp3
END

TO tp3
  Templebase
END

TO tp2
  Home
  ground
  home south 13 west 1
  bridge
  home west 13 north 1 rt 90
  bridge
  home east 13 south 1 lt 90
  bridge
  home north 11 west 1
  bridge
  home south 18 west 18
  repeat 19 [tree (((random 10)+1)/10)+1 fd 2]
  bk 2 rt 90
  repeat 18 [fd 2 tree (((random 10)+1)/10)+1]
  rt 90
  repeat 18 [fd 2 tree (((random 10)+1)/10)+1]
  rt 90
  repeat 18 [fd 2 tree (((random 10)+1)/10)+1]
END

TO tp1
  body
  home south 3 fountain
END

TO body
  setai 53
  setsh 50
  settr 0
  setdc 350 160 120
  setec 0 0 0
  setsc 210 110 0
  scale 15 0.8 6
  meter
  home
  fd 7 box
  repeat 4 [up 0.4 fd 0.25 scaled 6-
repcount*0.5 box]
  up 6.2
  scale 8 2 6
  setai 117
  setsh 400
  settr 0

setdc 800 510 90
setec 0 0 0
setsc 920 430 10
cone
scale 2 2 2 setfj "middle
setai 0
setsh 80
settr 0
setdc 0 0 0
setec 0 0 0
setsc 290 300 290
up 1 label [VAM Temple] dn 1
scale 0.5 5 0.5
dn 6 east 2.5 south 1.5 up 2.5
repeat 2 [cylinder fd 3 lt 90 cylinder fd 5 lt 90]
scale 0.6 0.5 0.6
dn 2.25
setai 0
setsh 0
settr 0
setdc 800 800 800
setec 0 0 0
setsc 0 0 0
repeat 2 [cylinder fd 3 lt 90 cylinder fd 5 lt 90]
up 4.5
repeat 2 [cylinder fd 3 lt 90 cylinder fd 5 lt 90]
END

TO mks
scale 1 1 1
rt 90 ru 60
setai 0
setsh 180
settr 0
setdc 0 0 0
setec 0 0 0
setsc 720 720 720
ball
setai 0
setsh 0
settr 0
setdc 800 800 800
setec 0 0 0
setsc 0 0 0
ru 30 rt 50
ball
END
<table>
<thead>
<tr>
<th>TO templebase</th>
<th>TO tour</th>
</tr>
</thead>
<tbody>
<tr>
<td>home meter</td>
<td>meter</td>
</tr>
<tr>
<td>scale 0.2 5 0.2</td>
<td>home</td>
</tr>
<tr>
<td>up 2.5 south 5 west 6.5</td>
<td>up 1.6</td>
</tr>
<tr>
<td>setai 250</td>
<td>bk 15</td>
</tr>
<tr>
<td>setsh 600</td>
<td>setwait 500</td>
</tr>
<tr>
<td>settr 0</td>
<td>repeat 10 [fd 1]</td>
</tr>
<tr>
<td>setdc 400 400 400</td>
<td>END</td>
</tr>
<tr>
<td>setec 0 0 0</td>
<td>TO ground</td>
</tr>
<tr>
<td>setsc 770 770 770</td>
<td>scale 1 1 1</td>
</tr>
<tr>
<td>repeat 18 [cylinder fd 2.2 rt 360/18]</td>
<td>meter</td>
</tr>
<tr>
<td>scale 0.6 0.6 0.6 dn 2.2</td>
<td>home</td>
</tr>
<tr>
<td>setai 53</td>
<td>south 11 west 11</td>
</tr>
<tr>
<td>setsh 120</td>
<td>face pcoff</td>
</tr>
<tr>
<td>settr 0</td>
<td>setai 0</td>
</tr>
<tr>
<td>setdc 340 0 340</td>
<td>setsh 0</td>
</tr>
<tr>
<td>setec 0 0 0</td>
<td>settr 0</td>
</tr>
<tr>
<td>setdc 840 0 0</td>
<td>setdc 0 0 0</td>
</tr>
<tr>
<td>repeat 18 [cylinder fd 2.2 rt 360/18]</td>
<td>setsc 812 812 821</td>
</tr>
<tr>
<td>scale 1 0.2 2.5 up 4.7</td>
<td>setsc 0 0 0</td>
</tr>
<tr>
<td>setai 26</td>
<td>pd repeat 4 [fd 22 rt 90] pu</td>
</tr>
<tr>
<td>setsh 100</td>
<td>south 2 west 2 down 1</td>
</tr>
<tr>
<td>settr 0</td>
<td>setai 63</td>
</tr>
<tr>
<td>setdc 0 10 500</td>
<td>setsh 60</td>
</tr>
<tr>
<td>setec 0 0 0</td>
<td>settr 500</td>
</tr>
<tr>
<td>setsc 0 910 10</td>
<td>setdc 0 0 750</td>
</tr>
<tr>
<td>repeat 9 [fd 1.1 box fd 1.1 rt 360/18 fd 2.2 rt 360/18]</td>
<td>setec 0 0 0</td>
</tr>
<tr>
<td>END</td>
<td>setsc 0 840 880</td>
</tr>
<tr>
<td>TO tree :s</td>
<td>pd repeat 4 [fd 26 rt 90] pu</td>
</tr>
<tr>
<td>Meter</td>
<td>up 1</td>
</tr>
<tr>
<td>scale 0.2*:s 2*:s 0.2*:s</td>
<td>south 7</td>
</tr>
<tr>
<td>setai 60</td>
<td>west 7</td>
</tr>
<tr>
<td>setsh 930</td>
<td>setai 0</td>
</tr>
<tr>
<td>settr 0</td>
<td>setsh 200</td>
</tr>
<tr>
<td>setdc 510 170 60</td>
<td>settr 0</td>
</tr>
<tr>
<td>setec 0 0 0</td>
<td>setdc 0 490 0</td>
</tr>
<tr>
<td>setsc 530 530 530</td>
<td>setec 0 150 0</td>
</tr>
<tr>
<td>up 1*:s</td>
<td>setsc 500 500 500</td>
</tr>
<tr>
<td>cylinder</td>
<td>pd repeat 2 [fd 40 rt 90 fd 7 rt 90] pu</td>
</tr>
<tr>
<td>scale 1.5*:s 2.2*:s 1.5*:s</td>
<td>rt 90</td>
</tr>
<tr>
<td>setai 63</td>
<td>fd 7</td>
</tr>
<tr>
<td>setsh 90</td>
<td>pd repeat 2 [fd 26 lt 90 fd 7 lt 90] pu</td>
</tr>
<tr>
<td>settr 0</td>
<td>fd 26</td>
</tr>
<tr>
<td>setdc 110 550 90</td>
<td>lt 90</td>
</tr>
<tr>
<td>setec 0 0 0</td>
<td>pd repeat 2 [fd 40 rt 90 fd 7 rt 90] pu</td>
</tr>
<tr>
<td>setsc 120 60 60</td>
<td>fd 40 lt 90</td>
</tr>
<tr>
<td>up 1.1*:s</td>
<td>pd repeat 2 [fd 26 lt 90 fd 7 lt 90] pu</td>
</tr>
<tr>
<td>cone</td>
<td>END</td>
</tr>
<tr>
<td>dn 2.1*:s</td>
<td>END</td>
</tr>
</tbody>
</table>
Screenshots: