CONSTRUCTING A TETRAHEDRON IN MICROWORLD: POTENTIAL FUTURE RESEARCH AND PRACTICE

ANDY YEH
Queensland University of Technology
a.yeh@qut.edu.au

MATTHEW PEILE
Queensland University of Technology
matthew.peile@student.qut.edu.au

Traditionally, the teaching and learning about 3D shapes has utilised concrete materials such as solid wood, plastic connectors and paper nets for modelling. While they are good materials, limitations such as time, accuracy, manipulability and creativity etc. could apply. This paper presents how a 3D microworld named VRMath 2.0 (VRMath2) can be utilised to construct a tetrahedron from within a cube, which would be difficult to do with real materials. The use of virtual materials in addition to real materials would have implications on how the teaching and learning about 3D geometry could be enhanced, and how the human spatial cognition and abilities can be further researched.

Introduction

Human spatial ability is a factor of intelligence, and is an important predictor of future career paths, especially in scientific research, engineering and the arts (Ivie & Embretson, 2010). Pittalis and Christou (2010) also pointed out that spatial abilities are closely related to academic achievement particularly in mathematics and geometry. In other words, learning about geometry would contribute to the development of spatial abilities.

The teaching and learning about 3D shapes (topic in geometry) in primary school classrooms usually start with real objects in the natural environment, then formalise into more regular shapes such as sphere (ball), cube (box), cylinder (can), cone, prism and pyramid. Typical activities include students constructing drawings, interpreting diagrams, and constructing physical models using nets or concrete materials. Concrete materials play an important role for school children to recognise, identify, visualise, and examine the properties of shapes. However, the use of concrete materials could be time consuming for preparation and construction; the use of flexible materials (e.g., sticks and foam balls, rubber bands, straws and pipecleaners etc.) has accuracy issues when measuring and they are not easy to put together; the use of solid materials (e.g., timber, plasticine, paper and cardboard etc.) may have visibility issues that prevent students from visualising invisible edges; the use of commercial shape connectors with some standard units could also be inflexible for construction (e.g., designed only for certain 3D shapes, could only connect in certain angles, no diagonal length). These conditions of concrete materials could limit the creativity and further development of geometrical understanding of 3D shapes and spatial abilities such as spatial visualisation, spatial orientation, and spatial relations (for details about the three spatial abilities, see Lohman, 1988).

The use of information and communication technology (ICT) tools can address most of the issues above. ICT tools such as LOGO microworld and Dynamic Geometry Software (DGS) are intuitive to use, accurate in constructing geometric objects, easy to manipulate geometric objects once created, and expressive[1] to create geometric objects. The power of LOGO is its ability to express natural geometric movement and the programming language, which links multiple representations of mathematical entities (Hoyles, Noss, & Adamson, 2002). However, most of the LOGO environments are lacking of interactive graphics, particularly on 3D capabilities. DGS, on the contrary, has great interactive 3D graphics that allow direct manipulation of virtual geometric objects, but often has no natural expressions of movement or does not focus on programmability. It is often perceived that DGS operates on more formal geometric thinking and reasoning as it is mostly based on Euclidean geometry. Constructing 3D shapes using ICT tools usually force and require learners to use and develop their understanding of geometrical relationships to produce dynamic “figures”, rather than static “drawings” (Jones, 2000). This is one key aspect for developing deep geometric understanding.

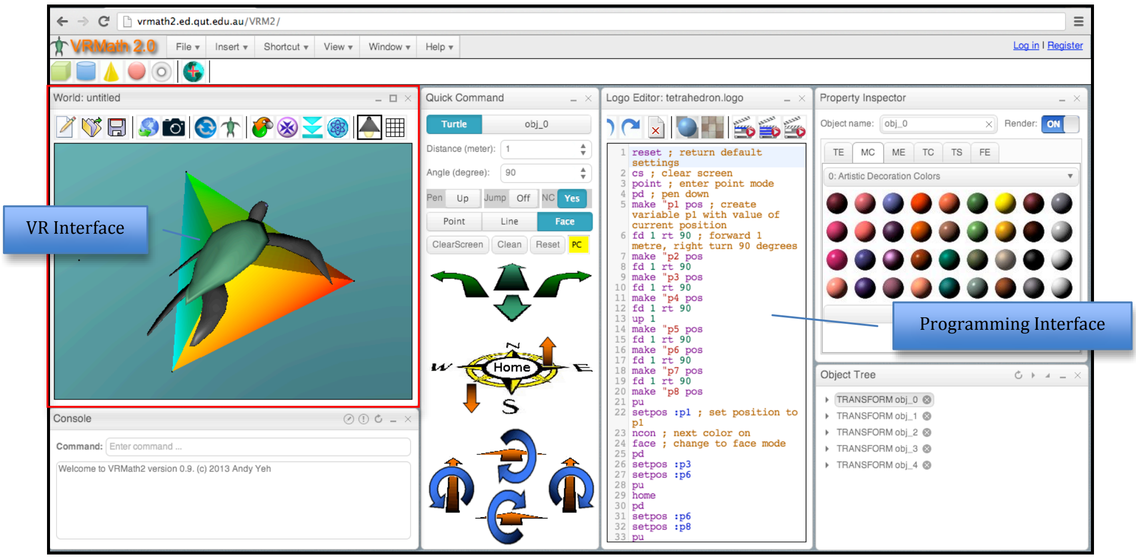
Based on the information presented above, it was thought that an ICT tool empowered by natural expressions of geometric movement, programming ability, and a 3D interactive graphic, could not only enable children to develop deeper understanding about 3D geometry, but also develop a wide range of spatial abilities through new ways of thinking and doing in the new computational environment (Resnick, 1996). Therefore, the purpose of this paper is to demonstrate how to create a tetrahedron in such an ICT tool environment named VRMath 2.0 (VRMath2) as an example, and discuss its implications for potential future research and practice.

The learning environment: VRMath2

VRMath2 is an open online learning environment. It has three main components: (1) an interactive virtual reality (VR) 3D interface, (2) a 3D programming interface, and (3) a Web 2.0 style sharing and collaborating interface. Due to the scope set, this paper focuses on the first two interfaces (see Figure 1).

The VR interface is virtually an unlimited 3D space, in which learners can navigate to see the virtual world from any perspective, and interact with the objects created in the virtual space. As a VR space, the objects in it include visible geometric objects with rich colours and textures, and invisible environmental objects such as light, sound and camera viewpoints. This interface helps develop spatial visualisation when manipulating geometric objects (particularly in small scale), and spatial orientation and relations when navigating (particularly in large scale) in the virtual world. One thing to note about this 3D VR graphic is that its unit of measurement is not based on pixels on screen as of other 2D or 3D computer graphics. As its name “virtual reality” implies, this VR 3D space utilises units that can be interpreted as metres to reflect on the real world settings and to work with the Cartesian coordinate system.

[1] The ICT tools provide facilities such as movement commands, mouse dragging actions, and colours and textures for users to express their ideas.



Welcome to VRMath2 version 0.9. (c) 2013 Andy Yeh

```
reset ; return default settings
cs ; clear screen
point ; enter point mode
pd ; pen down
make "p1 pos ; create variable p1 with value of current position
fd 1 rt 90 ; forward 1 metre, right turn 90 degrees
make "p2 pos
fd 1 rt 90
make "p3 pos
fd 1 rt 90
make "p4 pos
fd 1 rt 90
up 1
make "p5 pos
fd 1 rt 90
make "p6 pos
fd 1 rt 90
make "p7 pos
fd 1 rt 90
make "p8 pos
pu
setpos :p1 ; set position to p1
ncon ; next color on
face ; change to face mode
pd
setpos :p3
setpos :p6
pu
home
pd
setpos :p6
setpos :p8
pu
```

Figure 1. VRMath2.

The programming interface includes fully functional LOGO programming language, and a set of graphic user interface (GUI) to assist with the creation of virtual worlds. Because of 3D, this LOGO has an extended set of 3D movement commands (Figure 2) and 3D geometric primitives commands.

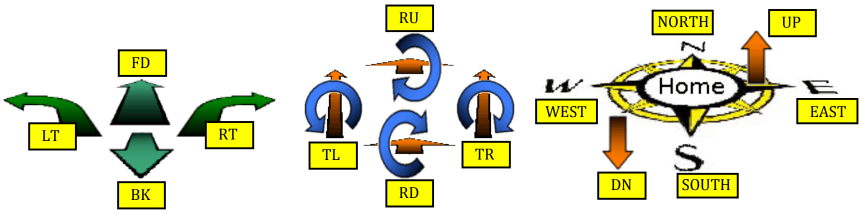


Figure 2. GUI of 3D movement commands.

Using the concept of frame of reference (FOR), these 3D movements can be classified into two categories: egocentric and fixed movements. The egocentric movements include FORWARD[2] (FD), BACK (BK), and the six turns of RIGHT (RT), LEFT (LT), ROLLUP (RU), ROLLDOWN (RD), TILTRIGHT (TR), and TILTLEFT (TL). These egocentric movements are based on the orientation of the turtle in the 3D space. The fixed movements are the four compass points EAST, WEST, NORTH, and SOUTH, plus the UP and DOWN. These movements express a relationship to a fixed direction and are not related to the turtle's orientation. HOME is a special fixed movement as it brings the turtle back to home position (0, 0, 0), face to north and back up. There is also a set of coordinate commands (universal FOR) that changes the turtle's position (or location) by specifying the x, y, and z coordinate. For example, SETPOS [x, y, z] will move the turtle to the specified coordinate [x, y, z]. With these natural expressions of 3D movement, the turtle is able to turn to any direction and move to any location to create objects in the 3D virtual space.

[2] The VRMath2 Logo commands are case insensitive. For ease of reading, we use capital letters for Logo commands in the paragraphs. If programmed in Lego Editor, we simply use small letters for Logo commands.

As a programming language, LOGO can create variables to store information and customise commands. For example, MAKE “p1 POS will store the turtle’s current position to the variable named p1. The command POS is a built-in command, which returns the turtle’s coordinate. The MAKE command takes two inputs, and stores the value of second input into the first input. Once this is done, SETPOS :p1 will then take the turtle to the coordinate (location or position) stored in p1.

To customise a command in LOGO is also very easy. For example, if SETPOS is not easy for young children to remember, we may create an easier command called GOTO instead. To do so, we can type in the Logo Editor with the following program and execute:

```
TO GOTO :place
SETPOS :place
END
```

The key words TO and END specify the beginning and end of a procedure or command. The GOTO is the procedure’s name. The :place specifies that the procedure GOTO will take in an input. This new GOTO procedure will then pass on the input :place to SETPOS command. Therefore, after the procedure GOTO is defined (executed), GOTO :p1 will be doing the same as SETPOS :p1.

Construction of tetrahedron in VRMath2

Tetrahedron is one of the five Platonic solids, which are among the regular 3D shapes constructed in the primary school classrooms. The construction of a tetrahedron can be a simple process when using paper nets or commercial shape connectors (see Figure 3).

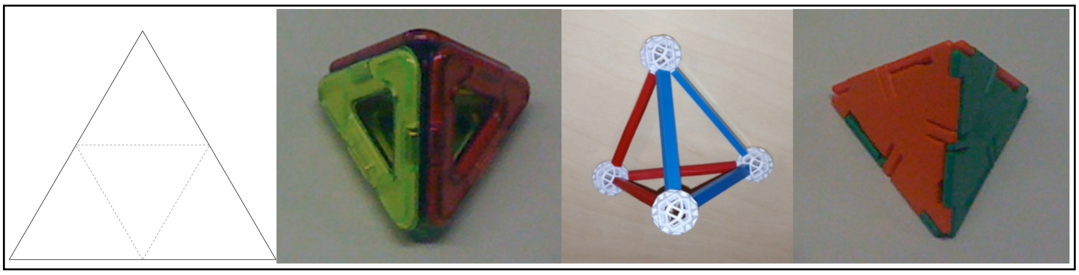


Figure 3. Tetrahedron net and models.

The use of concrete materials, such as paper nets and commercial shape connectors, allow children to easily identify the properties of tetrahedron as four faces of equilateral triangles, four vertices and six edges. And as of most 3D shapes, the angles (between two faces and between an edge and a face) within the 3D shapes are not easy to find or measure. Therefore, it is in fact more complex and difficult to construct in ICT tools.

The construction of a tetrahedron from within a cube is an extension of learning about the relationships between a cube and a tetrahedron. It is a good activity to develop children’s geometric understanding, and spatial visualisation and orientation abilities. The use of concrete materials can be beneficial; however, they start to reveal some limitations. As can be seen in Figure 4, the shape connectors do not have diagonal connectors. Therefore, blu-tack was used to secure the diagonal lines (edges of tetrahedron) on some vertices. Although it is possible to use concrete materials to achieve this task, the cube frame could distract the visualisation of the tetrahedron in the cube. The tetrahedron inside the cube is also only a frame. It could also be difficult for children to visualise the four triangle faces. Using concrete materials to create the four faces could be very difficult. ICT tools such as VRMath2 can address the limitations that arise from using concrete materials.

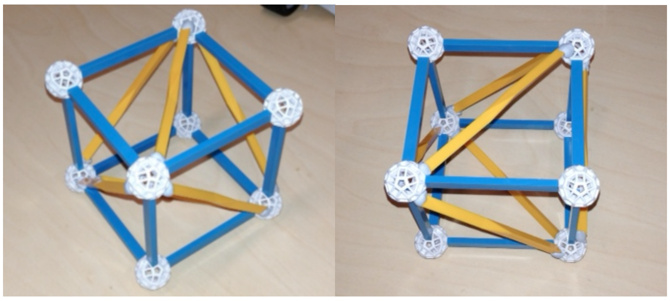


Figure 4. Tetrahedron in cube.

To construct a tetrahedron in VRMath2, one can use trigonometry to find out the angles to turn, then forward or move the turtle to create edges or faces. For example, the angle between an edge and a face is $\arccos\left(\frac{1}{\sqrt{3}}\right) = \arctan(\sqrt{2}) \cong 54.7356°$, and the angle between two faces (dihedral) is $\arccos\left(\frac{1}{3}\right) = \arctan(2\sqrt{2}) \cong 70.5288°$. Then in Logo, one can simply type in RU acrtan(sqrt(2)) to pitch up the angle between an edge and a face.

However, for young children who do not know about trigonometry, it is suggested to start with discussions about a cube on a drawing or using concrete materials (Figure 5).

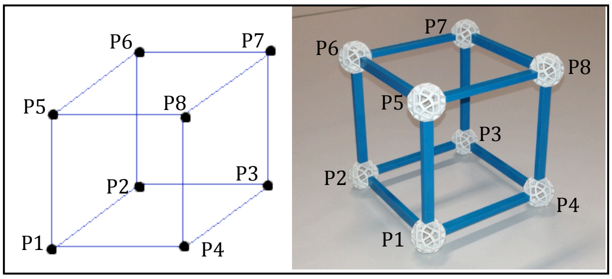


Figure 5. Cube with labelled vertices.

In these discussions, the eight vertices of the cube become the eight positions (or locations), which can be labelled as p1 to p8. The 3D Cartesian coordinate system is not necessary here, although they may be used in upper primary or secondary students who have prior knowledge about 3D Cartesian coordinates.

Then in VRMath2, these eight positions can be stored in eight variables exactly named as p1 to p8. To do so, we need to command the turtle to move into those eight positions. This can be achieved by using the movement icons (Figure 2) in Quick Command and Console (Figure 1). Or if students are familiar with movement commands, they can use their mental imagery (with or without the help of Figure 5) and program in the Logo Editor as below:

```
cs ; clear screen
point ; enter point mode
```

```
pd ; pen down
make "p1 pos ; create variable p1 with value of current position
fd 1 rt 90 ; forward 1 metre, right turn 90 degrees
make "p2 pos
fd 1 rt 90
make "p3 pos
fd 1 rt 90
make "p4 pos
fd 1 rt 90
up 1 ; fixed FOR command to move turtle up 1 metre without changing its direction.
make "p5 pos
fd 1 rt 90
make "p6 pos
fd 1 rt 90
make "p7 pos
fd 1 rt 90
make "p8 pos
pu ; pen up
```

In the above program, the semicolon sign (;) denotes what follows are comments only. The POINT command in second line specifies the pen mode (turtle track) to be point only. Other pen mode commands include LINE and FACE. The pen mode commands need to be given before PENDOWN (PD), and during PD, the pen mode cannot be changed. As can be seen in the program, the turtle will be moving through the eight positions, and will leave eight points track in the 3D space when executed (see Figure 6).

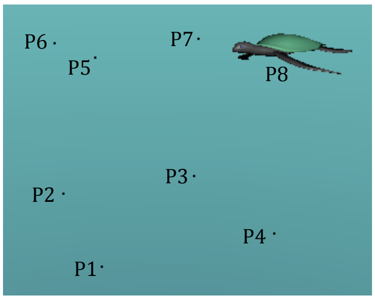


Figure 6. Cube vertices in VRMath2.

Next, the discussion will focus on the properties of tetrahedron. Since the faces of tetrahedron are equilateral triangles, we need to find four sets of three positions out of the eight and connect them to form four triangles. Another idea could be to find the four vertices in the eight positions, then connect the four vertices with lines, or utilise the four vertices to construct the four triangle faces. There could be a couple of possibilities and it is a good idea to try from the first position p1. With the help of concrete materials (e.g., Figure 5) or the navigation in the VR space, students should be able to find that it cannot be adjacent positions and the four edges would be the diagonal of the square faces on the cube. The following program demonstrates how to start from p1 to create a tetrahedron in the cube.

```
setpos :p1 ; set position to p1
ncon ; next colour on, this will help create colour rich triangles
face ; change to face mode
pd ; pen down for first triangle
setpos :p3
setpos :p6 ; as the turtle has moved through p1, p3 and p6, a triangle face will be created
```

```
pu ; pen up for the end of first triangle
setpos :p1
pd ; this pd creates the second triangle with vertices in position p1, p6 and p8
setpos :p6
setpos :p8
pu ; end of the second triangle
pd ; this pd creates the third triangle with vertices in position p8, p6 and p3
setpos :p6
setpos :p3
pu ; end of the third triangle
pd ; this pd creates the fourth triangle with vertices in position p3, p1 and p8
setpos :p1
setpos :p8
pu ; end of the fourth triangle and the tetrahedron
```

When the above program is executed, a colourful tetrahedron will be created in the 3D virtual space. We can hide the turtle by clicking an icon or giving a command HIDETURTLE (HT) for better views of the tetrahedron. The cube vertices and each of the four triangle faces can also be hidden to examine the construction process. Figure 7 below shows the process of construction.

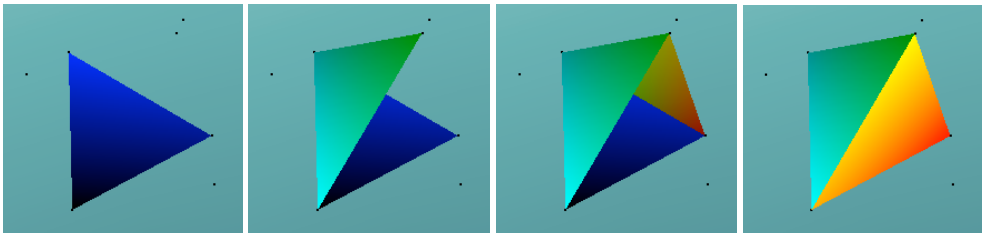


Figure 7. Construction process of tetrahedron in VRMath2.

Implications for future research and practice

In this paper, we have demonstrated the construction of a tetrahedron from within a cube in a powerful ICT tool or microworld named VRMath2. VRMath2 has a VR 3D interface, a programming interface, and a social (web 2.0 style) interface. The VR 3D interface allows users to navigate in virtual worlds and examine and interact with virtual objects. The programming interface allows users to use natural expressions of 3D movement and programming language to create virtual objects in virtual worlds. The social interface enables users to share their creations of virtual worlds and collaborate with other users. We think that VRMath2 represents a new paradigm of computing, and its new paradigm of thinking and doing mathematics would have some profound implications for future research and practice.

For research, VRMath2 is a pertinent vehicle for investigating and developing human spatial cognition and abilities. Because of its VR 3D interface, users will naturally tend to think 3D and work 3D. Unlike the 3D interfaces in other ICT tools that focus on one or few classical geometric shapes, this VR 3D space enable users to think and reason in larger scale of 3D worlds as well as on smaller scale of geometric objects. The larger scale of virtual worlds provides opportunity particularly for the operation and development of spatial orientation ability. The smaller scale of geometric objects will particularly facilitate the operation and development of spatial visualisation ability. The construction of virtual objects and worlds in VRMath2 using LOGO programming language will also assist the operation and development of spatial relations ability.

VRMath2 is rich in facilities that encourage users to express their thinking and ideas. When users express their thinking and ideas via programming and/or blogging their creations, their artefacts will help research into how we think, reason and develop our spatial abilities.

For practice, VRMath2 is also a vehicle to foster creativity. The unlimited virtual space and programming means endless possibilities of design and creation. When creating a virtual world, there are always many ways of doing and solving problems. For example, to connect or move from p1 to p3 in the tetrahedron task, instead of SETPOS :p3, a more natural way of TOWARDS :p3 FORWORD DISTANCEBETWEEN :p1 :p3 can also achieve the same result. Furthermore, creating a virtual world in VRMath2 is not just for making classroom geometric objects. With rich textures and environmental effects, the creation and construction of virtual worlds in VRMath2 can have virtually any objects such as a building, a table, a chair or a tree etc., as in the real world. Learning in VRMath2 is truly by designing and doing. Teachers can design learning and assessment activities using VRMath2, with possibilities to integrate disciplines such as mathematics, arts, science, engineering and technology.

To conclude, this paper has presented a new microworld named VRMath2 and demonstrated the construction of a tetrahedron in this microworld. We hope that the construction of tetrahedron in VRMath2 serves as a good example to encourage more future research and practice. And from more future research and practice, VRMath2 can evolve towards a better learning environment for all.

References

Hoyles, C., Noss, R., & Adamson, R. (2002). Rethinking the microworld idea. *Journal of Educational Computing Research, 27*(1&2), 29-53.

Ivie, J. L., & Embretson, S. E. (2010). Cognitive process modeling of spatial ability: The assembling objects task. *Intelligence, 38*(3), 324-335.

Jones, K. (2000). Providing a Foundation for Deductive Reasoning: Students' Interpretations when Using Dynamic Geometry Software and Their Evolving Mathematical Explanations. *Educational Studies in Mathematics, 44*(1), 55-85. doi: 10.1023/A:1012789201736

Lohman, D. (1988). Spatial abilities as traits, processes and knowledge. In R. J. Sternberg (Ed.), *Advances in the psychology of human intelligence* (Vol. 40, pp. 181-248). Hillsdale: LEA.

Pittalis, M., & Christou, C. (2010). Types of reasoning in 3D geometry thinking and their relation with spatial ability. *Educational Studies in Mathematics, 75*(2), 191-212. doi: 10.1007/s10649-010-9251-8

Resnick, M. (1996). New paradigms for computing, new paradigms for thinking. In Y. B. Kafai & M. Resnick (Eds.), *Constructionism in practice: Designing, thinking, and learning in a digital world* (pp. 255-267). Mahwah, NJ: Lawrence Erlbaum Associates.