Mathematics, Programming, and STEM

Andy Yeh
Queensland University of Technology
<a.yeh@qut.edu.au>

Vinesh Chandra
Queensland University of Technology
<v.chandra@qut.edu.au>

Learning mathematics is a complex and dynamic process. In this paper, the authors adopt a semiotic framework (Yeh & Nason, 2004) and highlight programming as one of the main aspects of the semiosis or meaning-making for the learning of mathematics. During a 10-week teaching experiment, mathematical meaning-making was enriched when primary students wrote Logo programs to create 3D virtual worlds. The analysis of results found deep learning in mathematics, as well as in technology and engineering areas. This prompted a rethinking about the nature of learning mathematics and a need to employ and examine a more holistic learning approach for the learning in science, technology, engineering, and mathematics (STEM) areas.

In the early 2000s when the personal computers were powerful enough to handle complex three-dimensional (3D) real time rendering, the author started developing an online 3D virtual reality-learning environment (VRLE) for learning mathematics. The focus of this VRLE (Figure 1) is on mathematical meaning-making of 3D geometry. This 3D VRLE was seen as an information and communication technology (ICT) tool, tutor, and/or tutee (Taylor, 1980) to facilitate the learning or knowledge construction of mathematics.

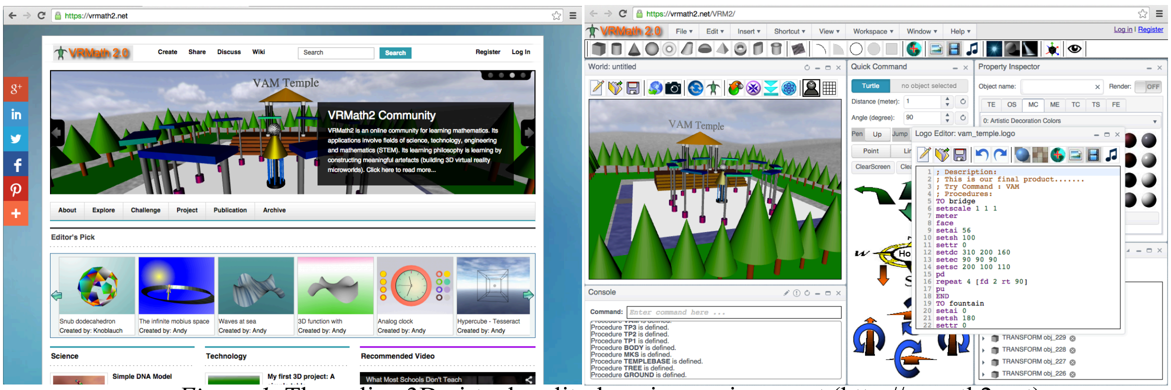


Figure 1: The online 3D virtual reality learning environment (http://vrmath2.net)

The design of the VRLE was informed by a semiotic framework (Yeh & Nason, 2004), which asserted the need of multiple semiotic resources for mathematical meaning-making. The implementation of the VRLE was elaborated elsewhere in Yeh (2007). The main components of the VRLE included an interactive 3D virtual space for visualising 3D objects, a customised Logo programming language for creating 3D virtual worlds, and an online forum for social discussion and presentation of the created 3D virtual worlds.

Among the semiotic resources, the programming language plays an imperative role of linguistic formalisation in aiding learners' mathematical expression. It serves as the symbolic representation of a mathematical function as well as the glue that binds all the representational modes together (Hoyles, Noss, & Adamson, 2002). Feurzeig, Papert and Lawler (2010) further confirmed the values of programming languages to the teaching and learning of mathematics. They argued that programming (in the example of Logo) not only

2015. In M. Marshman, V. Geiger, & A. Bennison (Eds.). *Mathematics education in the margins* (Proceedings of the 38th annual conference of the Mathematics Education Research Group of Australasia), pp. 659–666. Sunshine Coast: MERGA.

enables pupils to access an accurate understanding of some key mathematical concepts, but also develops problem-solving skills and facilitates the expansion of mathematical culture to topics in biological and physical sciences, linguistics, etc.

In this paper, the authors report a review of a teaching experiment, in which the focus was originally on the learning of mathematics by primary students in the VRLE. In this review, we found that through programming, the primary students not only developed deeper understanding of mathematics, but also applied and practiced their problem solving skills in designing, creating, and engineering their 3D virtual worlds.

Theoretical and Conceptual Frameworks

The theoretical framework for this research is rooted in semiotics, from which all human cognition is viewed as meaning-making endeavour within systems of signs. Lemke (2001) proposed a mathematical account of semiotics and classified the mathematical signs or representations into three categories of semiotic resources, namely typological, topological, and social-actional resources. Typological resources are those signs that signify meaning by discrete means. Languages (including programming languages) and symbols are typical typological resources. Conversely, topological resources signify meaning by continuous means. They could be animations, change of colour spectrum or sound pitch, and continuous variation of viewpoints such as changing from the top view of a square to side view of a square to recognise the different shapes a square can transform into due to different 3D perspectives (Yeh & Hallam, 2011). The social-actional resources are non-exclusive to the above two. They are the means of meaning-making by cultural activities or gestures, or from discussion to negotiation of doing things together such as when building a house or designing a garden. Informed by this mathematical account of semiotics, the VRLE was designed and implemented. Initial research and evaluation (see Yeh, 2007, 2013; Yeh & Hallam, 2011) have reported deep learning of mathematics within rich typological (e.g., Logo programming), topological (e.g., interactive 3D virtual space), and social-actional (e.g., discussion forum and group project) resources.

Upon the review of the teaching experiment in Yeh (2007), the authors further confirmed that the Logo programming language (a typological resource) of the VRLE played a central role to connect other semiotic resources for mathematical meaning-making. Moreover, it was found that the social-actional resources (e.g., a building project in the teaching experiment) also contributed to the learning of, not only mathematics, but also other science, technology, engineering, and mathematics (STEM) areas including technology and engineering. A new conceptual framework is thus formed as shown in Figure 2 below.

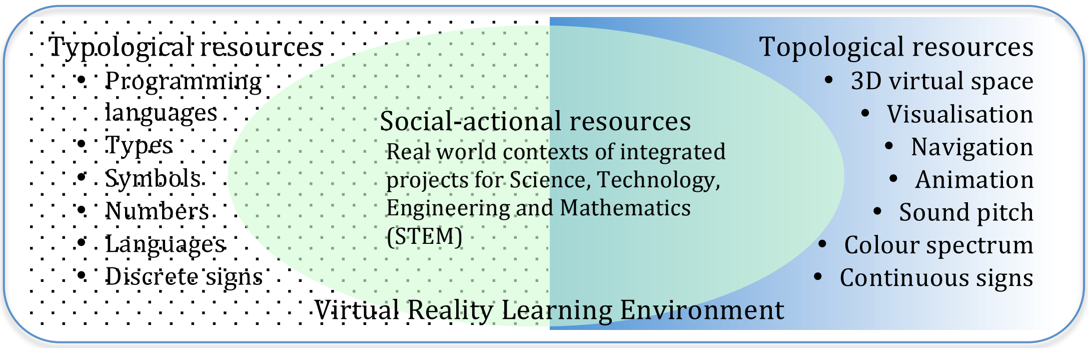


Figure 2: Semiotic framework for learning in STEM in VRLE

This new conceptual framework (Figure 2) is an initial attempt to explain what we found about the learning occurred in the VRLE. The framework postulates that the social-actional resources in the VRLE can provide the contexts of integrated projects in STEM areas. The rich semiotic resources in the VRLE thus enable learning beyond the field of mathematics, which in turn informs that the nature of learning (or meaning-making) about mathematics is not confined within mathematics itself, but in an interdisciplinary manner. This reasoning can also be applied to the learning in science, technology, and engineering, and even beyond STEM. In the next section, we will report on the reviewed teaching experiment to elaborate on this new conceptual framework. Due to the scope of this conference paper, the focus of this report is on how the programming connects all meaning-making resources to expand the learning from mathematics to STEM.

The Teaching Experiment

Three Year 5 students (Pseudo names: R2D2, Victor, Alekat20) participated in a 10-week (2 sessions per week, 1 hour per session) teaching experiment involving learning 3D geometry in the VRLE. In the first 8 weeks, the three participants were made familiar with the VRLE and were able to write Logo programs including 3D movement commands and procedures in the carefully designed learning activities by this teacher-researcher. In the last 2 weeks, the three participants had to work together as a team, choose a design project, and create the 3D virtual world for the project. This paper reports only on the participants' project in the last 2 weeks of the teaching experiment.

Data collected included video and audio recordings, participants' programming codes, 2D drawings, 3D virtual artefacts, and the teacher-researcher's field notes. For the purpose of this paper, we focus on the analysis of the programming process and report on the participants' learning in the STEM areas.

Results

The three participants decided to create a Temple, and started a generic problem solving cycle of think, plan, do, and check. They thought about what to have in the virtual Temple space, and then drew some initial plans (see Figure 3) in a collaborative and cooperative manner. The Temple was then divided into three parts and each participant designed and programed a part of the Temple.

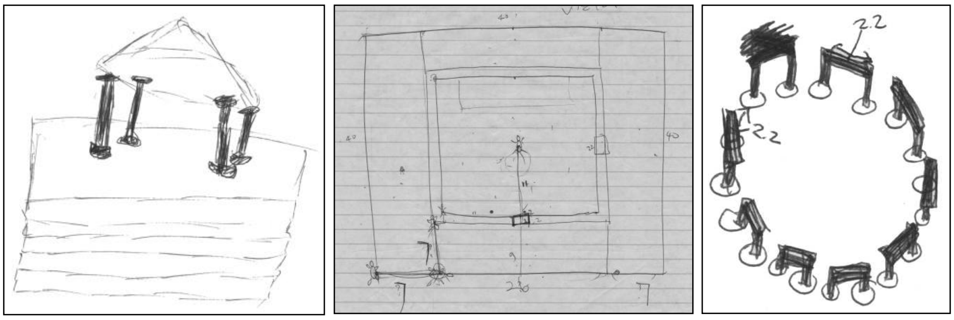


Figure 3: Three parts of the Temple plan

R2D2 was responsible for the stage with stairs and a kiosk. He observed his drawing design and thought that it could be achieved by scaling shapes of a cone, cylinders, and boxes. In doing the stairs, he experimented (i.e., trial and improvement) with different scales and started building the stair cases one by one:

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 …

Challenged by the teacher-researcher, he noticed a pattern firstly in the drawing of stairs then in the above commands as the scale of depth keeps decreasing by 0.5. After a few trials he came up with using a repeat to complete this stairs stage:

repeat 4 [up 0.4 fd 0.25 scaled 6-repcount*0.5 box]

The kiosk consisted a label, a cone roof, and four cylindrical posts, which were created in many cycles of think-plan-do-check, with calculations and trials of different sizes, locations, directions, and movements. R2D2 put every command in a procedure named *body* so the whole Temple stage could be created by just a simple command as *body*. In his spare time, he also wrote a *fountain* procedure with carefully chosen materials settings for running water (blue cylinders) and marble top (two overlapping spheres with different colours). He was very proud of his invention of the marble top because he found that although the two spheres were created at the same location, if they have different orientation then it will have the colour alternating effect. His creation of Temple part is shown in Figure 4 below.

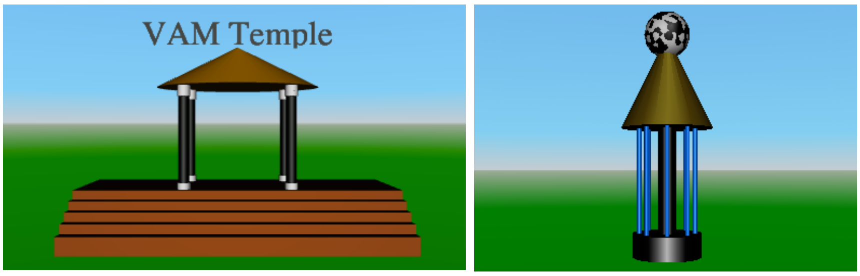
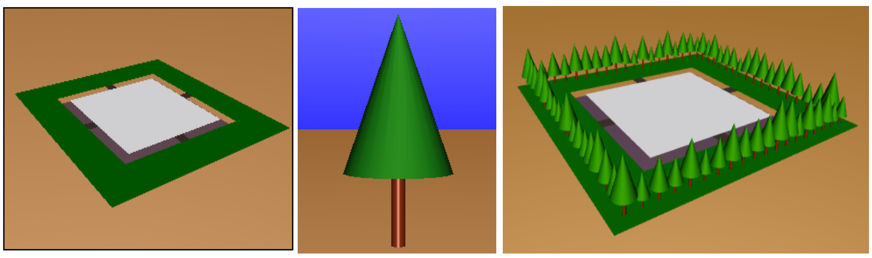


Figure 4: R2D2's Temple stage and fountain

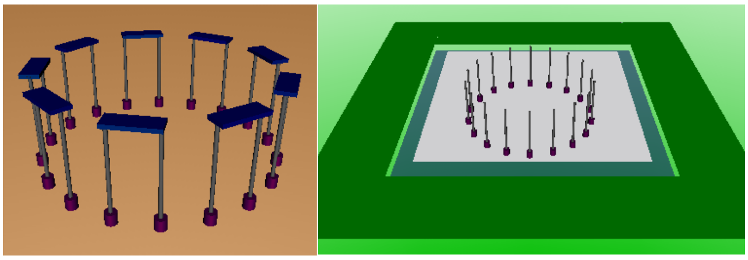
Victor was responsible for the Temple ground. He had more detailed plan with pre-calculated dimensions of the ground, centre court, and four bridges. Similar to R2D2, Victor decided to write a *ground* and a *bridge* procedure to create simple faces (planes) to achieve his design. He moved the turtle (i.e., the reference point in Turtle Geometry or Logo) in the 3D virtual space and recorded the track to create the 2D faces for the ground and bridges. After the construction of 2D faces, he wrote a *tree* procedure to create a simple 3D tree consisting of a cone and a cylinder. In order to generate different sizes of tree (a challenge by the teacher-researcher), the *tree* procedure was modified to take in an input. To create trees with random sizes (ranging from 1.1 to 2 times), Victor tried with brackets and eventually wrote: *tree (((random 10)+1)/10)+1*. The tree procedure was then repeated to create a tree fence surrounding the Temple ground (Figure 5).



```
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
```

Figure 5: Victor's Temple ground and trees

Alekat20 took the design of part 3 of the Temple. She drew a circular structure to be placed at the centre of the Temple. Because in previous activities she had learnt about a polygon formula: *repeat :side [fd 1 rt 360/:side]*, she was able to apply the formula (challenged by the teacher-researcher) to create 18 cylinder posts but only 9 top slates (Figure 6) with many trials. To place this structure at the right place is another challenge. In Logo, a 2D circle or polygons can be easily created using the above polygon formula, which forwards same distance and turns same angle for many times (i.e., repeat). However, to find out the centre of a polygon or a circle is difficult for primary students. Eventually, Alekat20 solved this by a few trials of different starting locations and relying on the feedback she got between her Logo program (typological) and the navigation in 3D virtual space (topological) to place her *templebase* structure at a centre-south location.



```
TO templebase
home meter setscale 0.2 5 0.2
up 2.5 south 5 west 6.5
repeat 18 [cylinder fd 2.2 rt 360/18]
setscale 0.6 0.6 0.6 dn 2.2
repeat 18 [cylinder fd 2.2 rt 360/18]
setscale 1 0.2 2.5 up 4.7
repeat 9 [fd 1.1 box fd 1.1 rt 360/18 fd
2.2 rt 360/18]
END
```

Figure 6: Alekat20's Temple base on Temple ground

After all participants had completed their design of their Temple part, the final effort was to put everything together. Because there were many procedures created, the teacher-researcher suggested them to each create another procedure such as *tp1* (Temple Part 1) and include all their procedures in it. The team then quickly sorted out the sequence and merged all procedures and thus a final virtual Temple was created with a main procedure named *VAM* (an acronym related to their first names) (Figure 7).

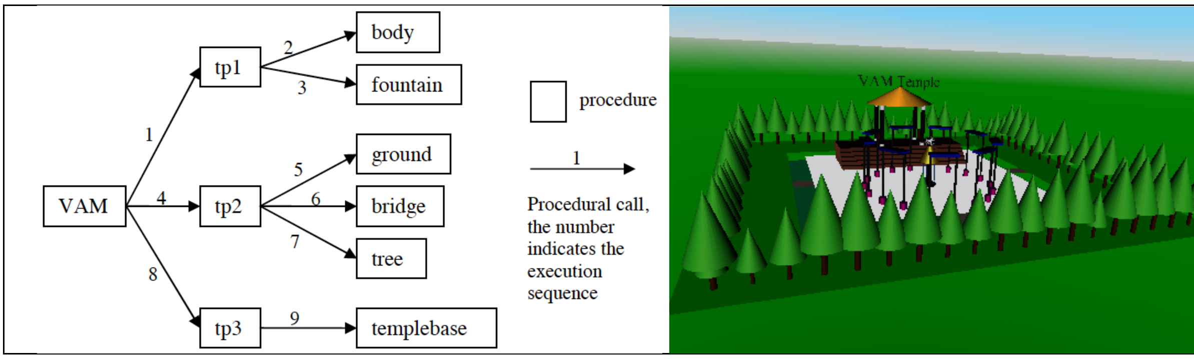


Figure 7: VAM Temple's procedural structure and virtual world

Discussion

From the results, we hope that we have shown deep mathematical learning among the three participants. Initially, the teaching experiment focused on the learning of geometry with 2D and 3D shapes, maps/plans, and location, direction, and movement. However, further analysis identified that learning in the areas of technology and engineering has also occurred. It can be sometimes difficult to separate the learning into individual disciplines. But here we will first try to discuss the learning we found according to disciplines, and then discuss the inter-relationships or integration of disciplines.

Learning in Mathematics

The mathematical concepts developed and applied in this students' final project not only included the intended 3D geometry, but also involved:

- Number and operations: This was evident where the participants designed and calculated the dimensions of their Temple parts. Whole numbers and decimal numbers were used throughout their programming. Operations were applied brilliantly to create intended results. For example, in R2D2's stairs stage, the use of *6-repcount*0.5* showed a decreasing mechanism by starting with a larger number 6.
- Measurement: This was evident from the plans they drew and the scales they wrote in the Logo program to change the sizes of 3D objects. The decimal scales such as 0.2 or 1.5 were not specifically discussed by the teacher-researcher with the participants. However, with the feedback from 3D virtual space and programming, the participants demonstrated good understanding and uses of decimal scales. It was usually a trial and improve process, in which they guessed a decimal scale in Logo, saw the 3D objects created, then made a sensible change of scales.
- Patterns: The teacher-researcher had this planned and thus always challenged the participants to use the *repeat* command to simplify the programming codes. By observing the geometrical patterns and the number patterns, participants developed ideas from describing (e.g., getting smaller), generalising (e.g., decreasing by 0.5), and then formalising in Logo programs.
- Algebra: The Logo programs created by the participants naturally contained many algebraic expressions. For example, the random sized tree *(tree (((random 10)+1)/10)+1)* involved variable (as an input of a procedure), order of operations, and functional thinking. In fact, a procedure in Logo is a sub-routine or a function. All participants were able to create and name their sub-routines and execute function calls.
- Chance: This was a contingency when the teacher-researcher challenged Victor to create different sizes of tree. The idea of random could be difficult to Victor. However, as demonstrated by his codes, he was able to utilise *random* command/function to generate a number range from 1.1 to 2.

The processes of learning mathematics in the VRLE are subtle and dynamic. We can confirm that the mathematical concepts and skills developed and applied in the VRLE are the results of the meaning-making (semiosis) within the multiple semiotic resources afforded in the VRLE. Further, what is common in the above discussion about the mathematical learning, is that it started and involved the Logo programming language. It is of course the nature of this VRLE, for its inclusion and design of Logo programming language as a core component. However, this reminds and informs that language, particularly a programming language can serve as a formalising agent for mathematical abstraction and logical thinking and reasoning.

Learning in Technology

Technology as framed in the Australian Curriculum, includes Design and Technologies, and Digital Technologies (ACARA, 2015). In this Temple project, the learning in technology was evident in:

- Design thinking and solutions (product): The whole process in this Temple project demonstrated a technological process as similar to the think-plan-do-check problem-solving cycle. The final VAM Temple was created through many refinements (prototyping) in designs of shapes, colours, and materials. There was a final product (solution) as a virtual Temple, albeit *virtual* but a different kind of *reality*. The virtual Temple is very tangible and real in a sense that the participants can see it, navigate, and walk in it.

- Computational and systematic thinking: As an ICT tool, the VRLE is a technology that engages the participants in designing and implementing digital solutions. The Logo programming language is a natural match for computational thinking, in which the participants practiced the problem solving, generated procedural and systematic codes to provide a solution (the virtual Temple), and evaluated the solution. The participants together managed the project and were able to create a systematic structure of the VAM Temple procedures (see Figure 7).

Learning in Engineering

The engineering design process is in a way similar to the generic problem solving cycle but can be broken down into more detailed cyclic steps: (1) identify the need or problem; (2) research the need or problem; (3) develop possible solution(s); (4) select the best possible solution(s); (5) construct a prototype; (6) test and evaluate the solution(s); (7) communicate the solution(s); and (8) redesign (Massachusetts Department of Education, 2006). In this Temple project, there was not enough time to go through the full cycle a few times. However, it was clear that the participants had gone through selecting materials and tools (programming commands and/or graphic user interface) to prototyping, testing, communicating, and redesigning their artefacts (e.g., the 3D stage, trees, centre court structures, and the programming codes etc.). The aspect of procedural thinking in the programming is also a key component of engineering design process. It involves flow-charting, data/variables, mathematical computations, and comparisons (e.g., greater and/or less than), logical operations (e.g., and, or) and controls (e.g., if, else). Some of them were not evident in this Temple project but they are certainly provided in the Logo programming language in this VRLE.

Learning in Mathematics, Programming and STEM

In the discussion above about the learning in individual disciplines, there are clearly some overlapping developments among disciplines. The meaning-making of mathematics is a dynamic and complex process among systems of signs in the VRLE. In this Temple project example, we found that the geometrical Logo programming language was the pivotal point linking all types of mathematical representations. When programming in the VRLE's Logo and virtual space, learners will mathematically analyse the real world context; generalise according to patterns and relationships; logically sequence the steps and commands; semantically and syntactically write in the programming language; execute the codes to create the virtual world; navigate in the 3D virtual space to examine and see the continuous visual feedback; rethink, recalculate, and repeat the earlier steps; and sometimes restart, redefine, and redesign their solutions.

From mathematics to programming, we also found that what were learnt was beyond merely mathematics as we originally focused. In building a virtual Temple, many of the technology and engineering concepts were learned and practiced by the participants. They collaboratively and cooperatively solved a problem (i.e., build a virtual Temple) by applying the design process (i.e., technology or engineering processes); selecting materials (i.e., material settings on virtual objects), programming with variable and procedures (e.g., the tree procedure), operating and communicating with computers (e.g., use of ICT tools and forums), thinking procedurally (e.g., sequence procedural calls), and creating systems and controls (e.g., combine functional calls). We can say that these participants have engineered a virtual Temple with technologies and mathematics through programming in the VRLE.

Conclusion

In our new conceptual framework (Figure 1), the social-actional semiotic resources such as designing and building structures can and will most certainly involve projects from science, technology, engineering, and mathematics. This semiotic framework thus has implications for future teaching and learning, not just for mathematics, but STEM as an integrated whole for a more holistic meaning-making approach. We would like to conclude that learning mathematics now encompasses other disciplines, particularly with areas in STEM. The nature of learning mathematics may be still within mathematics itself, but in the current technological world, at least in this VRLE, knowledge and skills of mathematics, technology, and engineering developed simultaneously. We need to rethink and consider how mathematics can be taught and learnt in an integrated way and utilise what current technologies such as this VRLE can offer. We also need to examine closely how programming can and should be included in the curriculum.

References

Australian Curriculum, Assessment and Reporting Authority (ACARA). (2015). Australian Curriculum: Technologies. Retrieved 28 February 2015, from http://www.australiancurriculum.edu.au/technologies/rationale

Feurzeig, W., Papert, S., & Lawler, B. (2010). Programming-languages as a conceptual framework for teaching mathematics. *Interactive Learning Environments, 19*(5), 1-15. doi: 10.1080/10494820903520040

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

Lemke, J. L. (2001). Mathematics in the middle: Measure, picture, gesture, sign, and word. Retrieved 6 February 2011, from http://academic.brooklyn.cuny.edu/education/jlemke/papers/myrdene.htm

Massachusetts Department of Education. (2006). Massachusetts Science and Technology/Engineering Curriculum Framework. http://www.doe.mass.edu/frameworks/scitech/1006.pdf

Taylor, R. P. (Ed.). (1980). *The computer in the school: Tutor, tool, tutee*. New York: Teachers College Press.

Yeh, A. (2007). *Knowledge Construction of 3D Geometry in Virtual Reality Microworlds.* (Doctoral dissertation, Queensland University of Technology, Brisbane, Australia). Retrieved from http://eprints.qut.edu.au/16648/

Yeh, A. (2013). Constructing a frame of cube: Connecting 3D shapes with direction, location and movement. In V. Steinle, L. Ball, & C. Bardini (Eds.), *Mathematics educations: Testerday, today and tomorrow* (Proceedings of the 36th annual conference of the Mathematics Education Research Group of Australasia. Melbourne: MERGA.

Yeh, A., & Hallam, J. (2011). Young children's understandings about "square" in 3D virtual reality microworlds. In J. Clarrke, B. Kissane, J. Mousley, T. Spencer, & S. Thornton (Eds.), Mathematics : Traditions and [new] practices (Proceedings of of the 23rd biennial conference of the AAMT and the 34th Annual Conference of the Mathematics Education Research Group of Australasia) Alice Springs, NT: MERGA/AAMT.

Yeh, A., & Nason, R. (2004, December). *Toward a semiotic framework for using technology in mathematics education: The case of learning 3D geometry.* Paper presented at the International Conference on Computers in Education, Melbourne, Australia.