

# Mathematics, Virtual Reality, and Programming

*Andy Yeh*

a.yeh@qut.edu.au

Faculty of Education

Queensland University of Technology

Australia

## Abstract

*In this paper, an online virtual reality learning environment named VRMath2 is introduced. The epistemological underpinnings for the VRMath2 learning environment are presented to support its design of linking mathematics with programming in virtual reality microworlds. The fractal geometry, specifically the Koch snowflake and Sierpiński triangle are demonstrated as applications in mathematics in this paper. By using the Logo programming language in VRMath2, many 2D or 3D fractals can be described, created and experimented in virtual reality space. And in doing so, mathematical understandings are developed and/or enhanced. There is great potential in using programming to create mathematical entities, particularly in 3D virtual reality space. VRMath2 is freely available online, and could be further explored as a teaching and learning resource for mathematics. However, more empirical studies about the use of VRMath2 learning environment should be evaluated to validate its theoretical underpinnings and to improve its practical applications in mathematics education.*

## 1 Introduction

New technological innovations often catch educators' and education researchers' attention for their potential to facilitate learning and knowledge construction. When just introduced, new technologies can often better engage and motivate students in learning, mostly due to the novelty of new technologies. However, the key challenges of learning with new technologies could ultimately depend on the epistemological underpinnings, from which the pedagogical design involving technologies could better ensure deep learning to occur, sustain and be transferable. The recent development in computer 3D graphics, mobile computing and affordable immersive virtual reality (VR) wearable devices such as the low-cost Google Cardboard<sup>1</sup>, mid-range Samsung Gear VR<sup>2</sup>, to high-end Oculus Rift<sup>3</sup> and HTC VIVE<sup>4</sup> etc. is looking to generate a wave of educational applications. When users experience for the first time with these wearable VR goggles for example, they are usually very impressed because of the new stereoscopic effect and perhaps some haptic feedback generated. However, when it comes to the learning of mathematics (and education in general), there is a need to further examine the issues and challenges pertaining the use of VR for mathematics education. Among these issues and challenges, the creation or consumption (who to create the VR mathematical microworlds) and the design and generation of VR experiences (how to create the VR mathematical microworlds) will be focused for discussion in this paper.

Since 2009, this author has begun the development of an online virtual reality learning environment (VRLE) named VRMath2, aiming to capitalize the potential of VR for the learning of mathematics, science, technology and engineering [12]. After a few prototyping and evaluations in

---

<sup>1</sup> For Google CardBoard VR, see <https://vr.google.com/cardboard/>

<sup>2</sup> For Samsung Gear VR, see <http://www.samsung.com/global/galaxy/gear-vr/>

<sup>3</sup> For Oculus Rift VR, see <https://www.oculus.com/rift/>

<sup>4</sup> For HTC VIVE VR, see <https://www.vive.com/>

the past few years, VRMath2 has evolved to be an online learning community for learners to design, create and share their 3D virtual worlds. In this paper, the epistemological underpinnings for the design of VRMath2 and its applications for the learning of mathematics will be presented.

## 2 The VRMath2 Learning Environment

### 2.1 Epistemological underpinnings

The epistemological underpinnings of VRMath2 are rooted in constructionism [3, 4], which asserts that we learn the best when we learn by making or building personally meaningful artefacts. Under this tenet, empowering learners to create their own VR space rather than consume VR space built by professionals or teachers is the foremost core design principle of the VRMath2 learning environment. This, however, raised the concern about how learners/users are able to create their own VR space with mathematical entities embedded virtual artefacts?

The roles of computer in education can be classified into three modes: tutor, tool, and tutee [5]. To explain these modes using VR (3D) applications for mathematics education, the computer as tutor can be envisaged as that learners explore in a professionally-designed VR space with pre-selected mathematical contents such as a geometry museum with already created 3D objects, or a specific parametric function visualized in 3D, where learners follow some pre-defined instructions to interact with the virtual objects. Examples of computer as tutor could be apps or games that teach/practice certain mathematical topics, or some augmented reality (AR) applications, in which learners explore/follow what are designed by the teachers or VR professionals. Computer as tool refers to that the software environment has some built-in facilities that enable learners to create and put together mathematical objects in order to investigate some mathematical properties, patterns and relationships. The computer assisted design (CAD), dynamic geometry systems (DGS), and computer algebra system (CAS) are main examples of computer as tool. Computer as tutee is that learners need to teach (program) the computer to create the mathematical virtual worlds by using a language it understands. The Logo programming language with its turtle graphics is an example of computer as tutee. The benefits of all three modes are recognized, but computer as tutee tends to confer special educational benefits because it shifts the focus of education in the classroom from end product to process, and from acquiring facts to manipulating and understanding them. With this in mind, the VRMath2 learning environment thus employed a Logo programming language for the learners to create (teach the computer) VR microworlds<sup>5</sup>. In addition to the three modes, researchers later added in a fourth mode as computer supported collaborative learning (CSCL), which utilizes the Internet for learners to share and learn from more knowledgeable members [1]. The idea of an online learning community based on social constructivism is therefore adopted for VRMath2.

### 2.2 Implementation

In order to meet the epistemological underpinnings described above, the VRMath2 learning environment has a community website (see Figure 2.1) for the social construction of knowledge through blogging, forum discussion and writing wiki pages. It also has a VRMath2 Editor website (see Figure 2.2) for the construction of personally meaningful artefacts of 3D virtual worlds by means

---

<sup>5</sup> There are a few definitions about the term “microworld”. In this paper, a (mathematical) microworld is a VR space, which has a symbolic core consisting of multiple representational modes, and programming language serves as one of them to link and interact with other modes [2].

of a Logo programming language. All these implementations are utilizing current W3C [6] standards such as HTML5 [7] and Web3D [8] for easy access and their long-term sustainability. A modern web browser such as Chrome, Firefox, or Safari with Internet connection will enable the participation and knowledge construction in the VRMath2 learning environment.

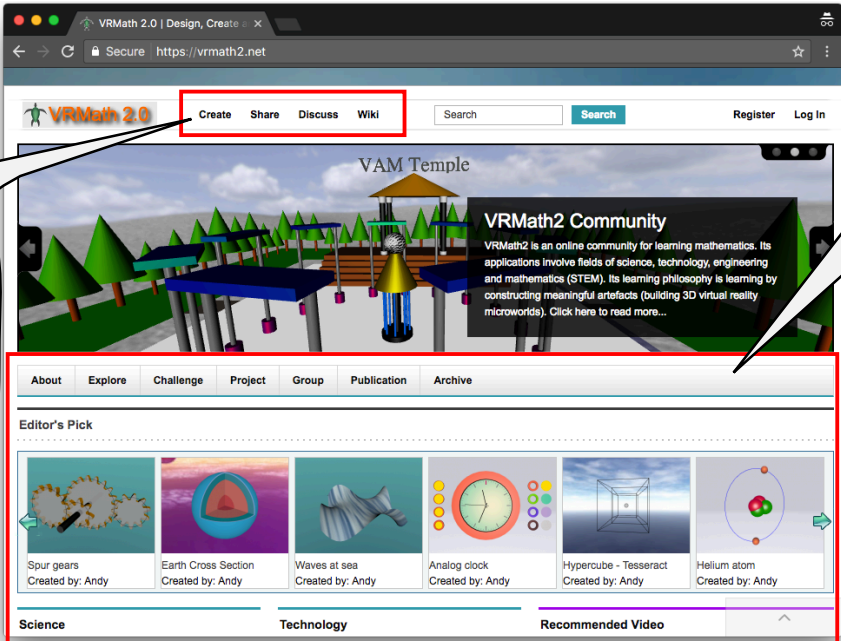
**Main menu:**

**Create:** Link to VRMath2 Editor site.

**Share:** Write a blog to share 3D virtual worlds and programming codes.

**Discuss:** Post or reply questions in the discussion forums.

**Wiki:** Write docs or tutorials about VRMath2 collaboratively.



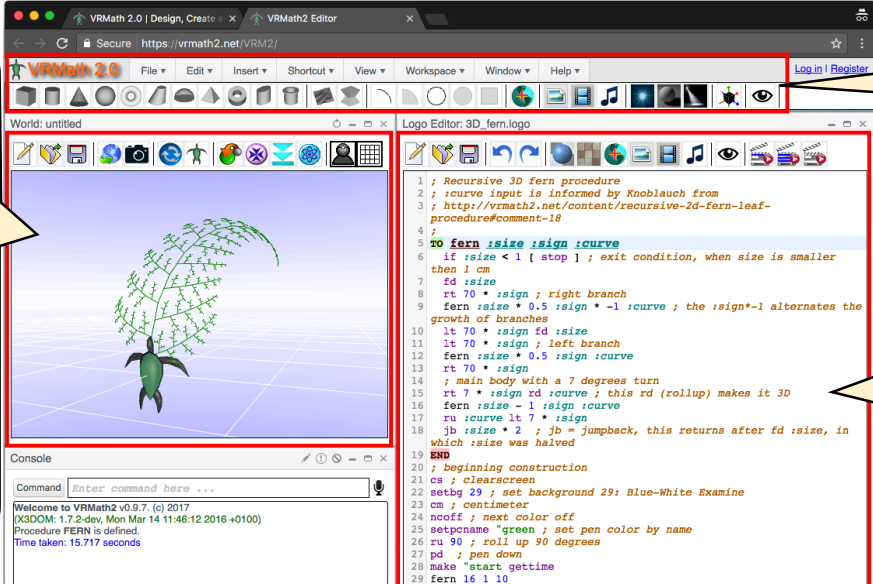
**Secondary menu and User blogs:**

This VRMath2 community portal site displays users' blogs in categories in the front page. Readers can read and comment on blogs as part of the process of social construction of knowledge. This site is a Web 2.0 style website, in which users can create rather than just consume.

**Figure 2.1** The VRMath2 Community Website (<https://vrmath2.net>)

**Virtual Reality interface:**

This is real-time and interactive 3D graphics. A 3D turtle serves as a reference point for location and direction. Users can freely navigate within this virtual space and interact with objects in it.



**Graphical user interface:**

Some widgets enable easy creation of 3D contents.

**Programming interface:**

A text-based and fully functional Logo programming language is implemented in VRMath2.

**Figure 2.2** The VRMath2 Editor Website (<https://vrmath2.net/VRM2>)

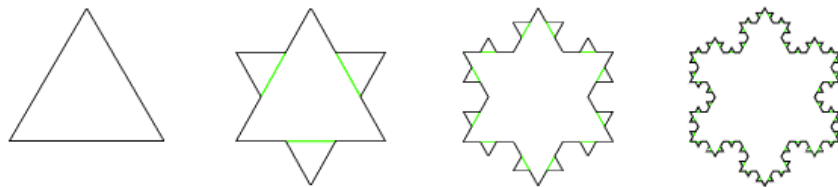
It should be noted that the design and implementation of this VRMath2 learning environment are constantly changing and evolving to improve user experience and learning.

### 3. Applications in Mathematics

The VRMath2 learning environment is now designed for science, technology, engineering and mathematics (STEM) education. Its applications in STEM education have been briefly reported elsewhere [11]. Because of its fully functional programming language and the turtle geometry, VRMath2 could have virtually unlimited applications for many mathematical topics. Using programming to create mathematical microworlds, the learners are offered the opportunity to describe, measure, and experiment the mathematical concepts/entities with codes to generate dynamic 3D worlds. In this paper, the Koch snowflake and Sierpiński triangle are selected as examples of applications to demonstrate the use of Logo programming for fractal geometry. These two fractals are perhaps well-known but will serve as appropriate examples for school or even undergraduate mathematics lessons.

#### 3.1 Koch snowflake

Fractal geometry is largely known by its self-similarity characteristic. They can be used to describe and simulate many naturally occurring objects such as fern leaves and trees. Artificially, defining patterns and recursive rules can also generate fractal arts that could stimulate further mathematical studies. Therefore, there could have two ways to investigate fractals using programming. The first way is to start from the fractal object, to describe its pattern and rules, and then write a program to create or simulate it. The second way is to start with defining/modifying patterns and rules (could be from some existing programs of fractals) then to experiment and generate variations or new fractals. For this Koch snowflake example, we can start from its fractal image (see Figure 3.1) to describe its patterns and rules then write a Logo program to create it.

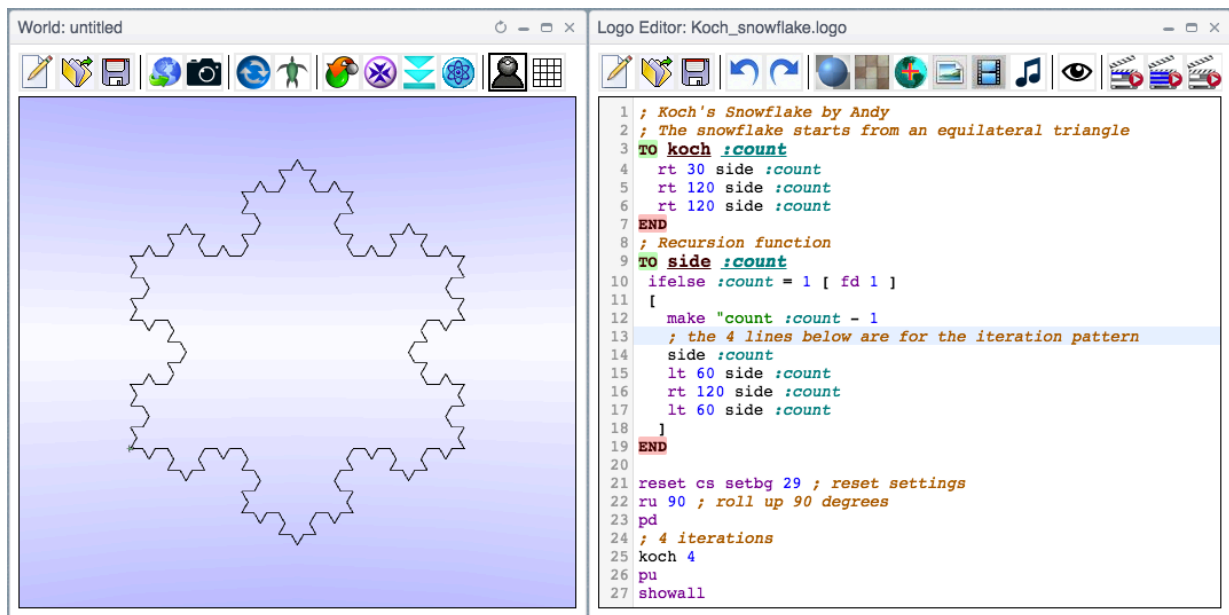


**Figure 3.1** The First Four Iterations of Koch Snowflake

(Image adapted from <https://upload.wikimedia.org/wikipedia/commons/d/d9/KochFlake.svg>)

As can be seen in Figure 3.1, the Koch snowflake starts from an equilateral triangle. It then follows a simple rule as: for each iteration, each side will be divided into three equal segments, then the middle segment is used as a side to form another equilateral triangle, this middle segment is then removed. This description about the iteration rule is key to the understanding of fractal. Other discussions about the changes in perimeter and area in the Koch snowflake would be good mathematical learning but they are not the focus in this paper.

Next, with the description of the iteration rule, there are still different ways to write a Logo program to generate this snowflake. In general, the programming will need to define the iteration pattern, and then put it in a recursive function to call itself with an exit condition, which is usually related to the level of iteration desired. The following program produces a 4-iteration snowflake in VRMath2 (see Figure 3.2).



**Figure 3.2** Koch Snowflake with 4 Iterations in VRMath2

As can be seen in Figure 3.2, the **koch** procedure is the base shape of an equilateral triangle. The **side** procedure is the iteration pattern for each side of the triangle. It calls itself (i.e., recursion) until the level (**:count**) of iteration reduces to 1. Once the program is completed, there are a few experiments learners may try to produce variations of fractals. The base shape (**koch** procedure) and the iteration pattern (in **side** procedure) can be replaced with different shapes and patterns (include 3D shapes and patterns). Learners of course can change the number of iteration and examine (mouse dragging) the 2D snowflake from any 3D perspective and distance in the interactive 3D virtual space.

### 3.2 Sierpiński triangle

The Sierpiński triangle (see Figure 3.3) can be constructed with a variety of methods [10]. Because it also has a base shape of an equilateral triangle, we can firstly try to create it from the Koch snowflake program presented in Figure 3.2.

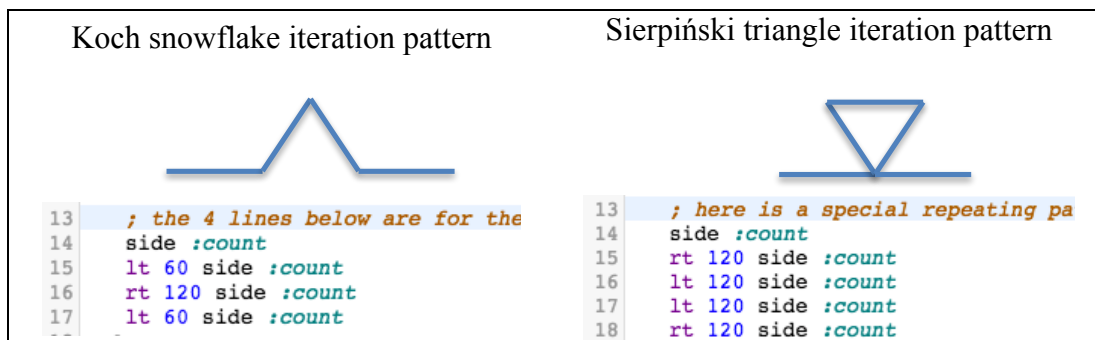


**Figure 3.3** The First Five Iterations of Sierpiński Triangle  
(Image adapted from

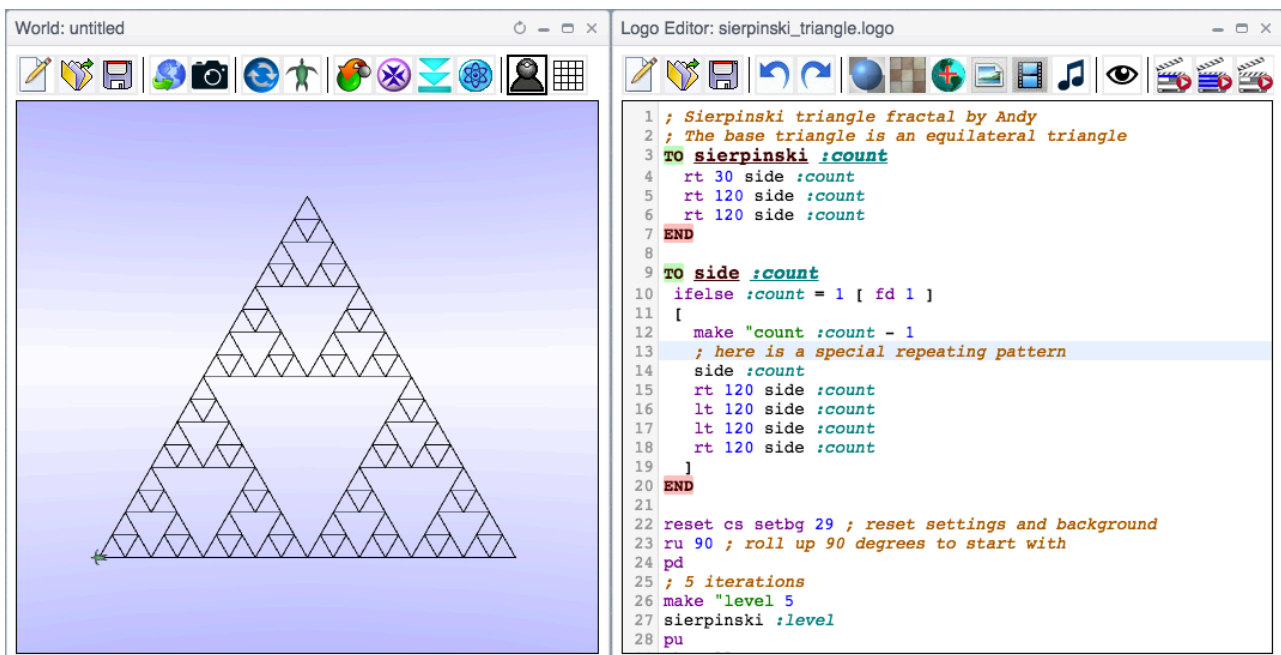
[https://upload.wikimedia.org/wikipedia/commons/0/05/Sierpinski\\_triangle\\_evolution.svg](https://upload.wikimedia.org/wikipedia/commons/0/05/Sierpinski_triangle_evolution.svg))

We take the same program but modify the iteration pattern as shown in Figure 3.4. After running the codes, the result is the Sierpiński triangle as shown in Figure 3.5. Using this approach, learners can design and try out different iteration patterns to create variations of fractals and arts (see [9]). This could be considered as an advantage of tutee mode of doing mathematics with programming in

VR space because learners can describe and manipulate more complex algorithms to experiment and understand about the mathematics.



**Figure 3.4** Iteration Patterns for Koch Snowflake and Sierpiński Triangle



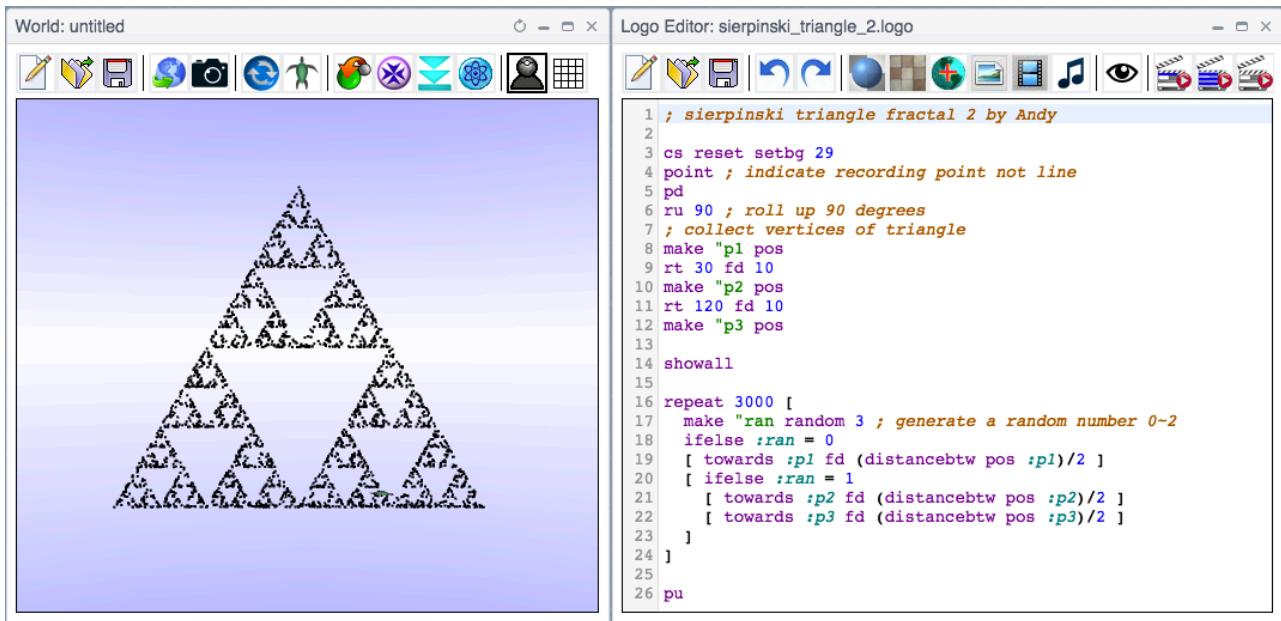
**Figure 3.5** Sierpiński Triangle with 5 Iterations in VRMath2

Other benefits of programming for mathematics include its flexibility of varying codes to achieve the same purpose and its ability to carry out steps of computation with flow controls such as *if*, *ifelse* and *repeat* (loop). Therefore, using a computer programming language can implement simple to complex mathematical algorithms, and in the case here, to generate fractals with ease. For example, the Sierpiński triangle can be constructed using the chaos game approach with the following steps [10]:

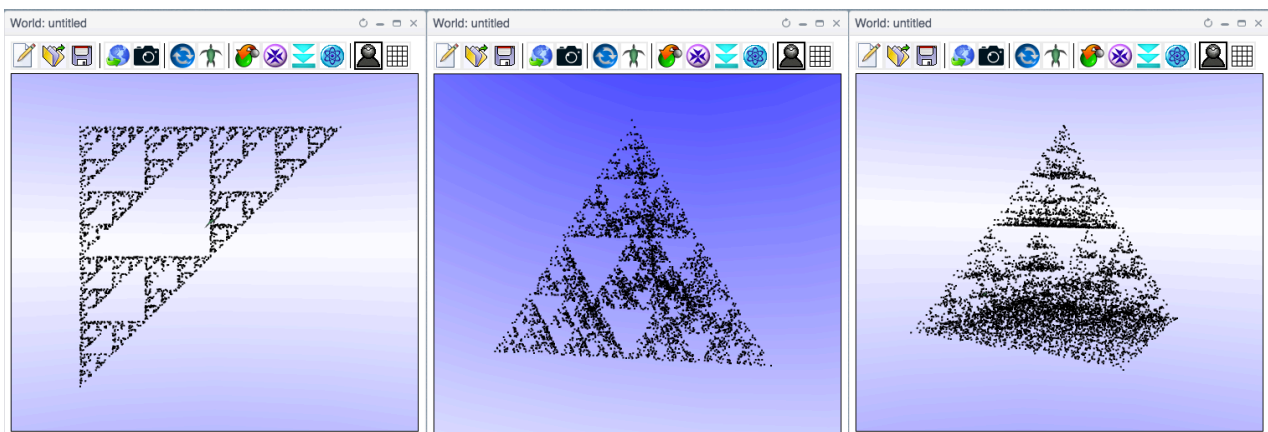
1. Take 3 points in a plane to form a triangle, you need not draw it.
2. Randomly select any point inside the triangle and consider that your current position.
3. Randomly select any one of the 3 vertex points.
4. Move half the distance from your current position to the selected vertex.

5. Plot the current position.
6. Repeat from step 3.

These steps can be coded easily in VRMath2's Logo programming language. With 3000 points plotted, the Sierpiński triangle is taking shape as shown in Figure 3.6. Using this random method, variations such as any triangles or even 3D tetrahedron or any pyramid can be experimented by slightly modifying the codes. Figure 3.7 shows the variations of Sierpiński triangle with right angle triangle (2D), tetrahedron and a rectangular pyramid (3D).



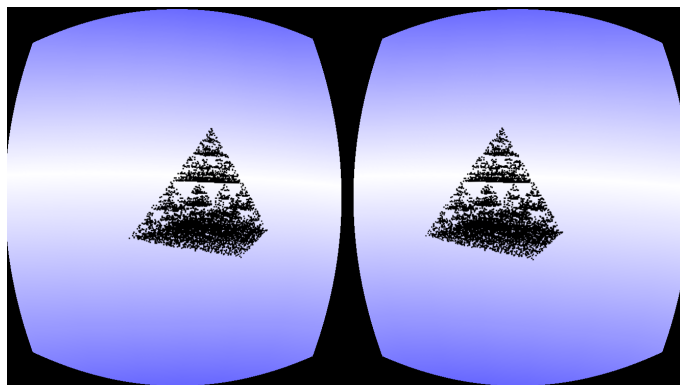
**Figure 3.6** Sierpiński Triangle Generated Using Random Method in VRMath2



**Figure 3.7** Sierpiński Triangle in Right Angle Triangle, Tetrahedron and Rectangular Pyramid

In the VRMath2's virtual reality space, users can freely navigate to examine the virtual objects and virtual worlds created. In fact, any virtual worlds created on VRMath2 website, can be viewed using stereoscopic VR such as Google Cardboard with a mobile device such as an iPhone. Figure 3.8 below shows the stereoscopic view of the Sierpiński pyramid. In this example, the pyramid fractal is

only examined in stereoscopic VR without further interactions. There are some animation and interaction (e.g., mouse over or click) functionalities offered in VRMath2 for creating more interactive and dynamic 3D virtual worlds. However, that is beyond the scope of this paper.



**Figure 3.8** Stereoscopic View of 3D Virtual World in VRMath2

#### **4. Conclusions**

Virtual Reality technology is now quite affordable and accessible. In time, it will inevitably become a commodity in our daily life. Like many new technologies, VR has the potential to facilitate learning in mathematics. However, if without careful consideration of the epistemological ground for the design of VR learning environment, the potential of VR could diminish after its novelty worn out. In this paper, an online virtual reality learning environment named VRMath2 is introduced. The design of the VRMath2 learning environment is informed by constructionism, and therefore, it has become a learning environment that empowers learners to create via programming rather than just consume information created by others.

Technology can play a role as tutor, tool or tutee. Each of the three roles has its strengths and weaknesses, and quite often a technology (e.g., a software or a learning environment) could have multiple roles. The VRMath2 learning environment could also be recognized as a tool but it has strongly adopted the tutee mode because of its inclusion of a Logo programming language as the main means of construction of 3D virtual worlds. Programming can be powerful and useful, as is demonstrated in this paper about creating fractals. The skill of programming is also inherently related to mathematics, and can be transferable to the learning in other disciplines such as science, technology and engineering. On the other hand, programming (even in Logo turtle geometry) can be difficult for learners. Learners would require time and opportunities in the classroom to problem solve with programming. The VRMath2 learning environment is online and freely accessible. Its potential could be further explored and integrated into the teaching and learning of mathematics. This presented VRMath2 learning environment and its theoretical underpinnings, however, should also be further empirically evaluated and validated, so that further insights on how to effectively utilize programming and virtual reality technology for mathematic can emerge.

**Acknowledgements** The author would like to thank Dr. Paul Grimm from University of Fulda, Germany, for his input on programming fractals in 3D computer graphics.



## References

- [1] Bruckman, A., and Bandlow, A. (2002). *HCI for kids*. In J. Jacko & A. Sears (Eds.), *Handbook of Human-Computer Interaction*. NJ: Lawrence Erlbaum Associates.
- [2] Hoyles, C., Noss, R., and Adamson, R. (2002). *Rethinking the Microworld idea*. *Journal of Educational Computing Research*, 27(1), 29-53.
- [3] Kafai, Y. B., and Resnick, M. (Eds.). (1996). *Constructionism in practice: Designing, thinking, and learning in a digital world*. Mahwah, NJ: Lawrence Erlbaum Associates.
- [4] Papert, S. (1991). *Situating constructionism*. In I. Harel, S. Papert & Massachusetts Institute of Technology. Epistemology & Learning Research Group. (Eds.), *Constructionism: research reports and essays, 1985-1990* (pp. 1-12). Norwood, N.J.: Ablex Pub. Corp.
- [5] Taylor, R. P. (Ed.). (1980). *The computer in the school: Tutor, tool, tutee*. New York: Teachers College Press.
- [6] The World Wide Web Consortium (W3C). (2017). See <https://www.w3.org/>
- [7] The World Wide Web Consortium (W3C). (2017). HTML5. See <https://www.w3.org/TR/html5/>
- [8] Web3D Consortium. (2017). See <http://www.web3d.org/>
- [9] Wikipedia. (2017). *Koch snowflake*. [https://en.wikipedia.org/wiki/Koch\\_snowflake](https://en.wikipedia.org/wiki/Koch_snowflake)
- [10] Wikipedia. (2017). *Sierpinski triangle*. [https://en.wikipedia.org/wiki/Sierpinski\\_triangle](https://en.wikipedia.org/wiki/Sierpinski_triangle)
- [11] Yeh, A. (2017). *Programming Driven 3D Modeling on the Web*. In *Proceedings of Web3D '17*, Brisbane, QLD, Australia, June 2017. DOI: <http://dx.doi.org/10.1145/3055624.3075953>
- [12] Yeh, A., and Chandra, V. (2015). *Mathematics, Programming and STEM*. 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.