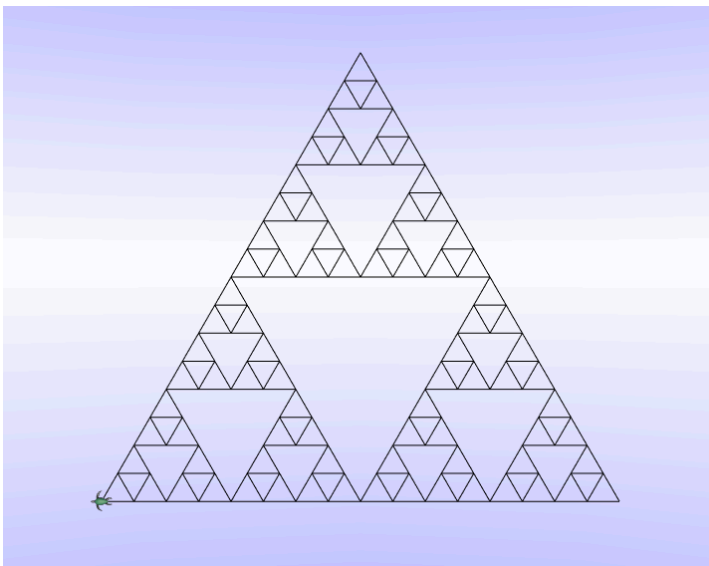
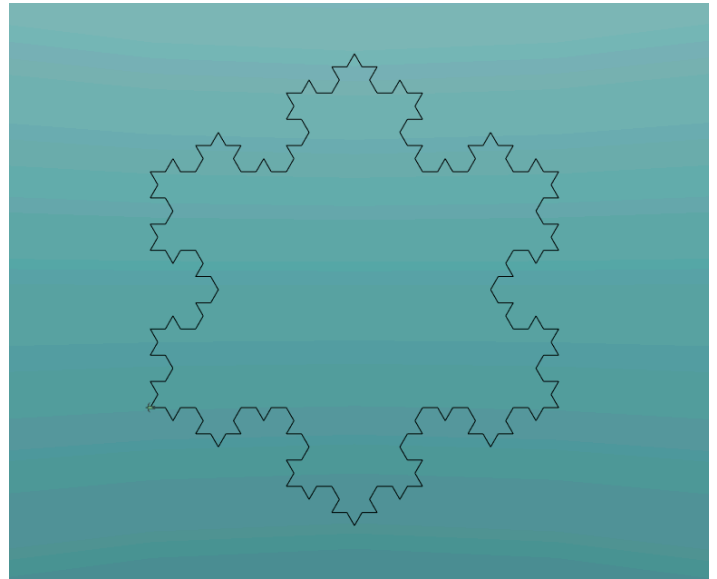


# Programming Fractals in 3D Workshop 2 Snowflake & Triangle



```
1 ; Koch's Snowflake
2 TO side :count :length
3   ifelse :count = 1 [fd :length]
4   [
5     make "count :count - 1
6     side :count :length
7     lt 60 side :count :length
8     rt 120 side :count :length
9     lt 60 side :count :length
10  ]
11 END
12
13 TO koch :count :length
14   rt 30 side :count :length
15   rt 120 side :count :length
16   rt 120 side :count :length
17 END
18
```

## Programming fractals in 3D VR

Workshop for Asian Technology Conference in Mathematics (ATCM)

Date/Time: 17<sup>th</sup> December 2017, 2:30 pm ~ 3:30 pm

Venue: Computer Lab 504, Science Building II  
Chung Yuan Christian University, Taiwan

Handbook for  
Programming Fractals in 3D  
Virtual Reality:  
Koch snowflake  
Sierpinski triangle

VRMath2 Workshop 2

VRMath2 Community

<https://vrmath2.net>

is a free online learning environment for STEM

Dr Andy Yeh

[a.yeh@qut.edu.au](mailto:a.yeh@qut.edu.au)

Queensland University of Technology

# Contents

1. Aims and requirements .....	1
2. VRMath2 introduction .....	2
2.1. Create an account .....	3
2.2. Community site overview .....	4
2.3. VRMath2 Editor overview .....	5
2.4. 3D Navigation and Logo Editor .....	6
3. 3D LOGO programming .....	7
3.1. 3D rotation and movement .....	8
3.2. Primitive objects .....	9
3.3. Material settings .....	10
3.4. Scaling objects .....	11
3.5. Loop .....	12
3.6. Procedures .....	13
3.7. Save your works (Logo files) .....	14
4. Programming fractals .....	15
4.1. Recursive procedure .....	16
4.2. Koch snowflake .....	17
4.3. Sierpinski triangle .....	19
5. Blogging and Publishing .....	21
5.1. Publishing 3D model .....	22
5.2. Save an image for blog .....	23
5.3. Write a blog .....	24
5.4. Edit own blog .....	27
5.5. Commenting blogs .....	28
6. Viewing 3D models in CardBoard .....	29
6.1. Scan QR Codes to view in CardBoard .....	30
6.2. VRMath2 VRBox .....	31
7. Forums and Wiki for help .....	32

# 1. Aims and requirements

## Aims

In this hands-on workshop, participants will use a Logo programming language to create fractal geometry in an online 3D virtual reality learning environment named VRMath2. VRMath2 is freely available at <https://vrmath2.net>.

Using the Logo programming language in VRMath2, fractals such as Fern leaves, Trees, Koch curve (snowflake), Sierpinski triangle and carpet, Dragon and Peano curves etc. can be described, experimented, and created using the Logo turtle geometry, and recursive and random capabilities of the programming language. Further, these fractals can be extended to 3D (not to confuse with fractal dimension) in VRMath2's virtual reality interactive space, and presented online in web browsers and/or viewed with Cardboard VR.

Participants will be introduced with a basic structure of recursive function and Logo programming, then they will be able to experiment and invent variations of fractals in 3D. All these will be done online in a web browser. Participants can also publish their fractals online in the VRMath2 website.

In this Workshop, there will be a very brief introduction to Logo programming and turtle geometry, then we will focus on creating recursive and random fractals of:

1. Koch snowflake, Sierpinski triangle, and variants in 2D and 3D.
2. Sierpinski carpet (2D) and Menger's sponge (3D) (Optional, depending on time).

## Requirements

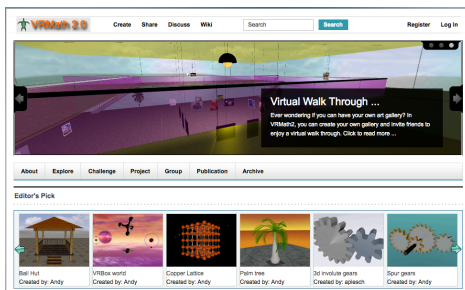
- Computer lab (Windows or Mac) with Internet access.
- Web browser (Chrome, Firefox or Safari).
- A mobile phone (e.g., iPhone 6 or newer, Android 6 or newer phone) is optional but encouraged for viewing the created 3D models. Mobile phone needs to have a QR Code App (e.g., i-nigma) installed.
- CardBoard or VR box is optional. There will be some provided in the workshop.
- Time: 1 hour



## 2. VRMath2 introduction

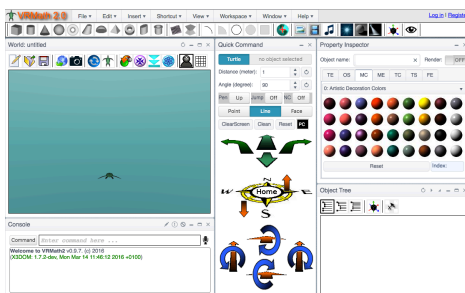
**VRMath2** stands for **Virtual Reality Mathematics 2.0**. It started in Year 2001 as an online learning environment for constructing knowledge of 3D geometry. Over the year, it has evolved to become a STEM (Science, Technology, Engineering and Mathematics) integrated learning environment. The 2.0 indicates that the website is a Web 2.0 style website, which allows users to design, create, share and discuss about the 3D virtual worlds they create. In this workshop, for example, you will be creating 3D atomic models in digital virtual space. While you are constructing a 3D atomic model for science, you will also need and learn about coding (technology, computational thinking), and use mathematics (e.g., geometry, trigonometry) to design and solve problems (engineering your atomic models).

### VRMath2 Community site (<https://vrmath2.net>)



The community website is the portal site for VRMath2. In this website, you can browse through community members' blogs, provide your comments, ask and answer questions in the discussion forums, create wiki pages, and most importantly, blog to share your creations of 3D virtual worlds.

### VRMath2 Editor (<https://vrmath2.net/VRM2>)



The Editor is a powerful online tool for creating 3D contents. You just run it as a software in a web browser such Google Chrome (PC) and mobile Safari (for iPad). It has friendly user interface and a LOGO programming language for creating 3D contents. All works in the Editor can be saved online in your account space and share (blog) later.

### VRMath2 VRBox (for mobile device only, <https://vrmath2.net/vrbox>)



This is a recent application and is still developing. Using a CardBoard VR Box with a mobile phone (e.g., iPhone 6 and above, Android 6 and above), you can view all your 3D contents created and all 3D contents shared in VRMath2 websites. In this workshop, you will be able to see your 3D models or any models published in VRMath2 in a VR CardBoard.

## 2.1 Create an account

VRMath2 is a free online learning environment. Without needing an account to login, visitors can browse most of the website contents and create 3D contents with VRMath2 Editor. However, with an account, users can further share, discuss, comment and save files to personal folders.

To create an account, simply follow the link below to register.

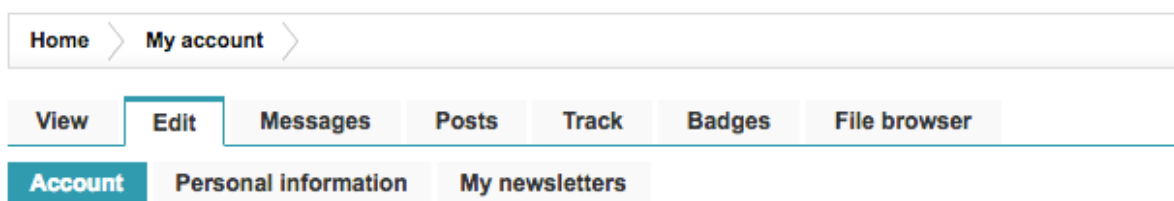
<https://vrmath2.net/user/register>

You should choose a sensible username, and enter a valid email address to register. After registering, an email will be sent to your registered email address. This email includes an auto-generated password and a one-time login link (can be used once only).

### Status

Your password and further instructions have been sent to your e-mail address.

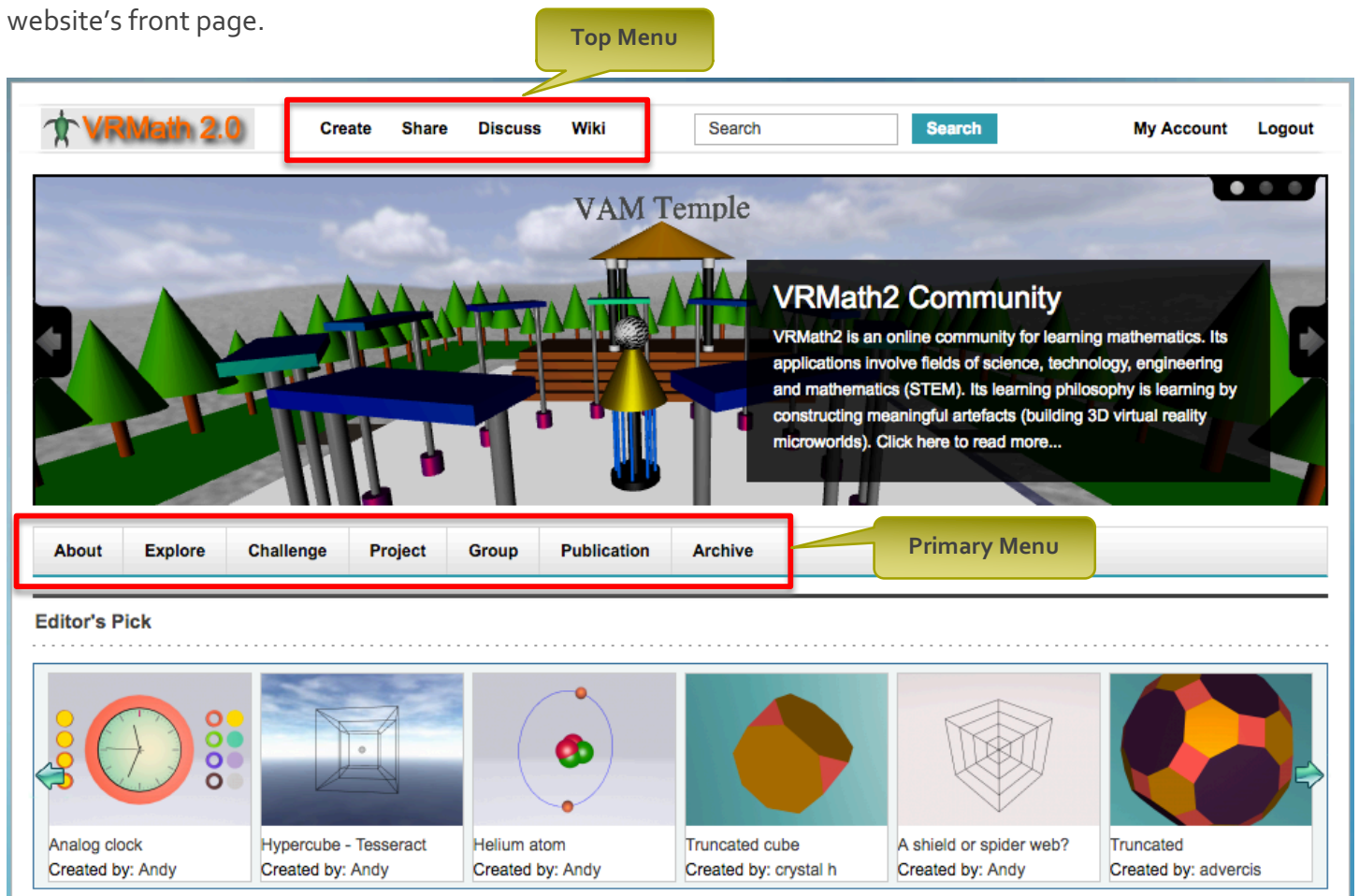
It is highly recommended that you click on the one-time login link, which will bring you to your account. Then you can click on the **Edit** tab to change password, select/upload own account picture, add your real name and other information.



You should be able to login and logout freely. In case that you forgot your password, you may request a new password from this link: <https://vrmath2.net/user/password>

## 2.2 Community site overview

The VRMath2 Community site (<https://vrmath2.net>) is easy to navigate. Below is a screenshot of the website's front page.



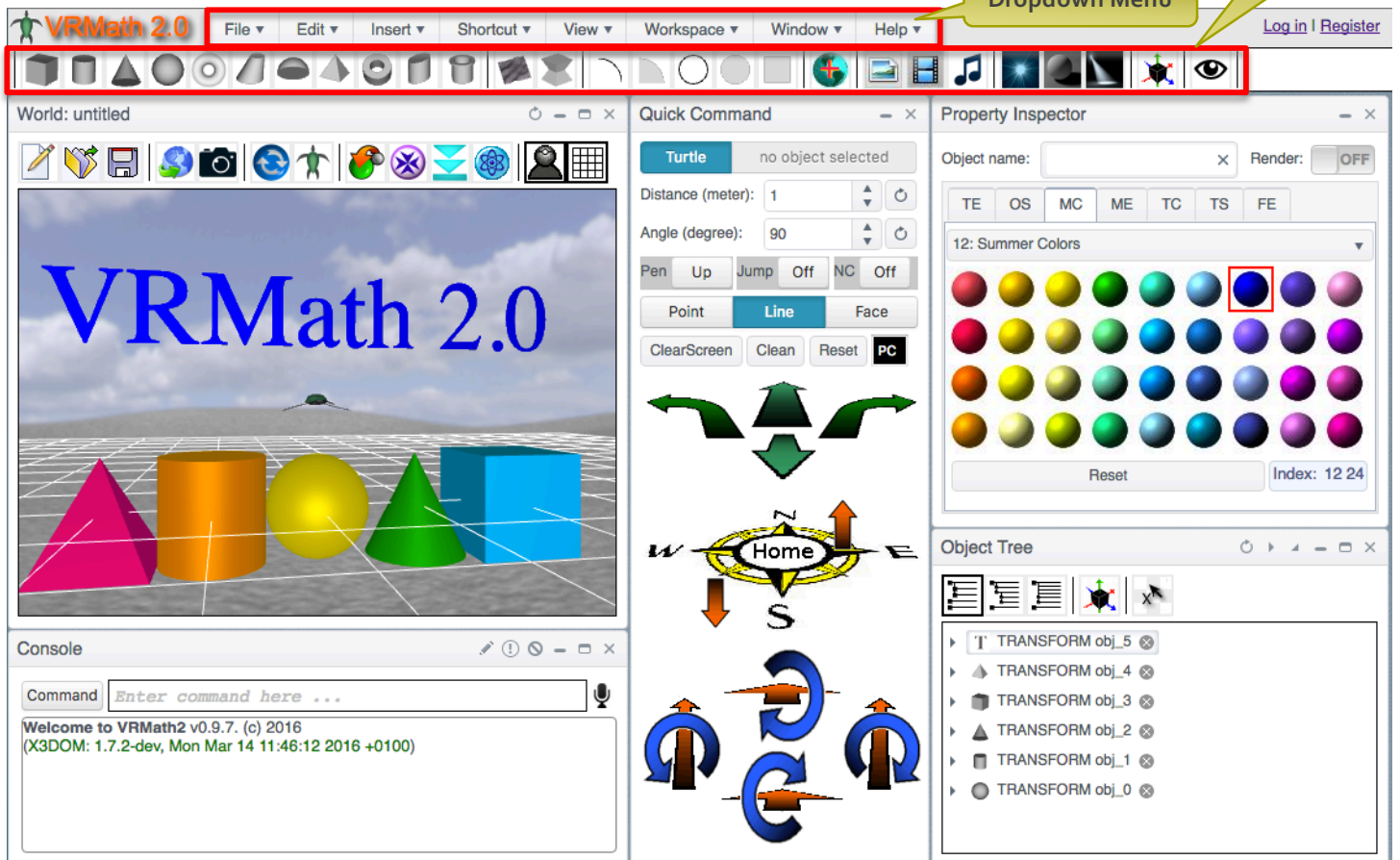
The **Top Menu** has four main functions:

- **Create:** click to open the VRMath2 Editor in a new window
- **Share:** click to write a community blog or deposit your creations
- **Discuss:** click to visit forums to ask and/or answer questions
- **Wiki:** click to visit wiki for documentations

The **Primary Menu** has categorized blogs and challenges, which you may be interested in taking. Please feel free to read a few blogs and see how community members are coding and creating 3D virtual worlds. You are also welcome to leave comments to blogs.

## 2.3 VRMath2 Editor overview

VRMath2 Editor (<https://vrmath2.net/VRM2>) is an online 3D contents creator.






The **Dropdown Menu** has access to most functions in VRMath2 Editor. For example, all **Object Bar** functions (insert primitive objects) can be accessed as well from the **Dropdown Menu**. It is suggested that you click on these menus and try out, particularly the **Workspace** and **Window** menus.

A simple click on the objects in the **Object Bar** will insert the clicked object at the turtle's location and direction. You can try to see if you can produce the same **3D World** as in the above image with **Quick Command** and **Property Inspector** windows.

## 2.4 3D Navigation and Logo Editor

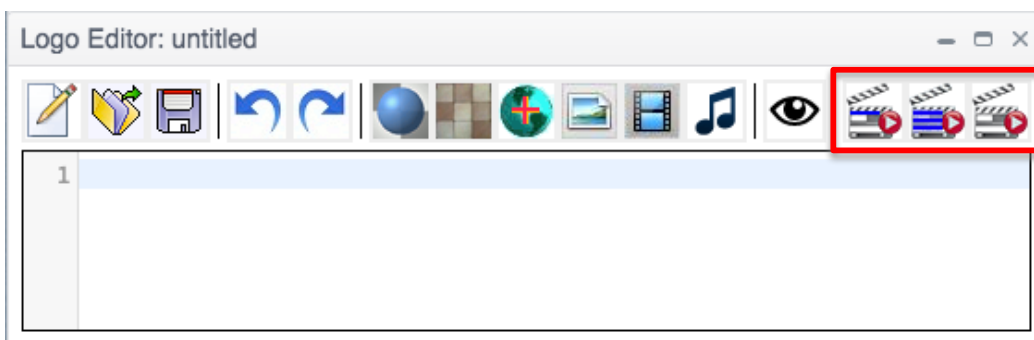
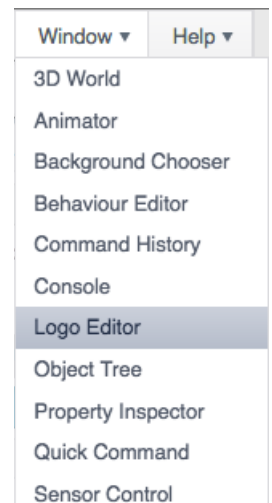
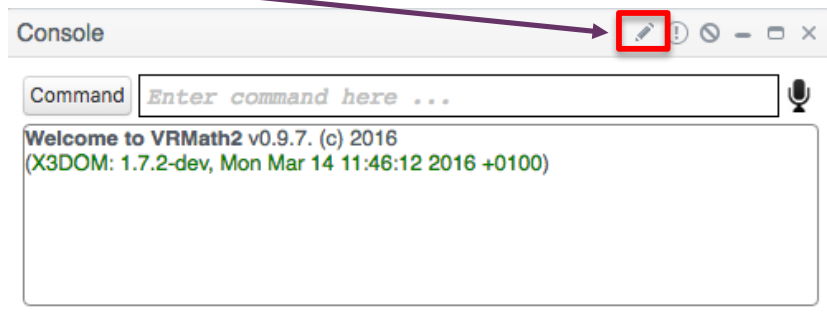
### 3D Navigation



It is essential to navigate in the 3D window so to inspect where the turtle is and what you have created. You can try out the icons in the Toolbar, particularly the Reset Viewpoint , Fit All , and Set Rotation Centre  are very useful when navigating in 3D virtual space. When using a touch screen, you can use two fingers to rotate, pan and zoom. For more details about 3D navigation, please visit wiki at <https://vrmath2.net/content/how-navigate-3d-virtual-space>

### Logo Editor

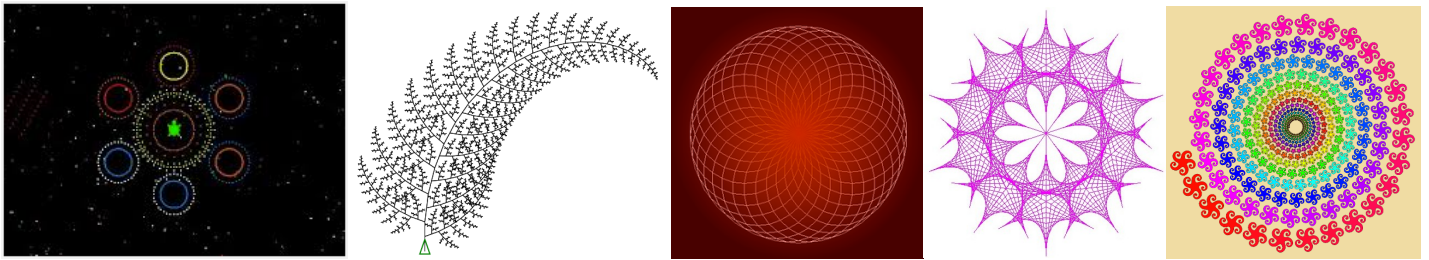
The Logo Editor window can be open from the **Window** menu, or by clicking on the pencil in the **Console** window.



The three icons at the left will run your codes in **per line**, **per block** (selection), or **all** program.

## 3. 3D LOGO programming

LOGO is a programming language developed in 1960s. It became popular mostly due to its Turtle Graphic, in which you program a turtle (a reference point in 2D space) to move and turn so to draw geometric pictures.



In VRMath2, the Turtle Graphic has become 3D. The 3D movements will be introduced in next section. LOGO commands are not case sensitive. As a programming language, it has features of what a programming language requires. Below are some key features for a programming language.

### Variables

To create a variable, we can use the MAKE command. The variable to be made needs a double quote (“) in front. To use the variable created, you need to have a colon (:) in front of the variable name.

```
MAKE "number 100 ; this line stores 1 into the variable named number  
PRINT :number ; then ask the computer to print the variable :number
```

### Flow control

Common control structures are **IF** or **IFELSE**. For example,

```
MAKE "number 100  
IF :number = 100 [ PRINT [ The number is 100 ] ] ; this will print "The number is 100"  
IFELSE :number > 50 [ PRINT "Big ] [ PRINT "Small ] ; this will print "Big"
```

### Arithmetic

Programming language usually has many build-in mathematical functions. For example,

```
PRINT 360 / 8 ; this will print 45  
PRINT SQRT 2 ; this will print square root of 2
```

In LOGO, texts after a semi-colon (;) mean comments, not codes. For more details see:

<https://vrmath2.net/content/logo-guidereference>



## 3.1 3D rotation and movement

Traditional 2D LOGO has only 2 rotations (**LEFT** angle and **RIGHT** angle), and 2 movements (**FORWARD** distance and **BACK** distance). The *distance* is specified as number of pixels in traditional 2D LOGO.

For 3D space, the VRMath2 LOGO has 6 rotations,

- **LEFT** angle or **LT** angle
- **RIGHT** angle or **RT** angle
- **ROLLUP** angle or **RU** angle
- **ROLLDOWN** angle or **RD** angle
- **TILTLEFT** angle or **TL** angle
- **TILTRIGHT** angle or **TR** angle

And 8 movements:

- **FORWARD** distance or **FD** distance
- **BACK** distance or **BK** distance
- **EAST** distance
- **WEST** distance
- **NORTH** distance
- **SOUTH** distance
- **UP** distance
- **DOWN** distance

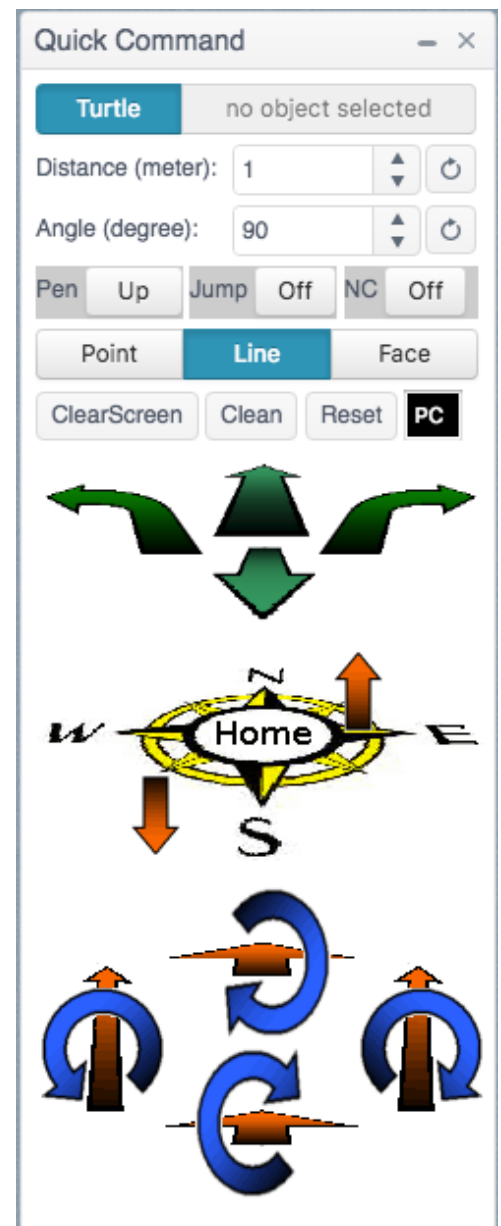
The distance is specified as METER or CENTIMETER. The above commands can be trialed in the **Quick Command window**.

There are also coordinate movement commands if you wish to move the Turtle to specific 3D coordinates.

- **SETX**  $x$
- **SETY**  $y$
- **SETZ**  $z$
- **SETXYZ**  $x y z$

For more details, please see wiki page at:

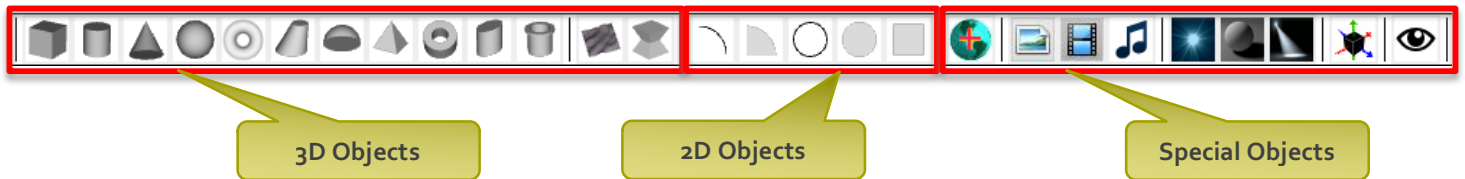
<https://vrmath2.net/content/how-move-turtle-3d-virtual-space>



## 3.2 Primitive objects

Like traditional LOGO, **PENDOWN** (or **PD**) will start recording the turtle's track, and **PENUP** (or **PU**) will stop recording the turtle's track.

In VRMath2's 3D LOGO, there are a few built-in objects as shown in the **Object Bar** below.



### 3D Objects

These include **CUBE (BOX)**, **CYLINDER (CAN)**, **CONE**, **SPHERE (BALL)**, **TORUS (DONUT)**, **SNOUT**, **DISH**, **PYRAMID**, **RECTANGULARTORUS (RECTORUS)**, **SLOPEDCYLINDER (SLOPEDCAN)**, and **NOZZLE**.

### 2D Objects

2D objects has zero Z dimension (no thickness or depth). These include **ARC**, **PIE**, **CIRCLE**, **DISK** and **RECTANGLE**.

### Special Objects

These special objects include **WORLD**, **PICTURE**, **VIDEO**, **SOUND**, **POINTLIGHT**, **DIRECTIONALLIGHT (DIRLIGHT)**, **SPOTLIGH**, **TRANSFORM**, and **VIEWPOINT**. Except for **WORLD**, **PICTURE** and **VIDEO**, other special objects have no geometry in the 3D space.

All objects, when inserted (either from programming or **Object Bar**) into the **3D World** window, will be inserted according to the turtle's location and direction. For example, if you want to insert an up-side-down cone, you will need to flip the turtle first. The codes below will create an up-side-down cone.

**HOME** ; this brings the turtle to home location (coordinate 0 0 0), facing north (-Z).

**TILTLEFT 180** ; tilt left 180 degrees is the same as flipping the turtle

**CONE** ; this cone will be up-side-down as the turtle

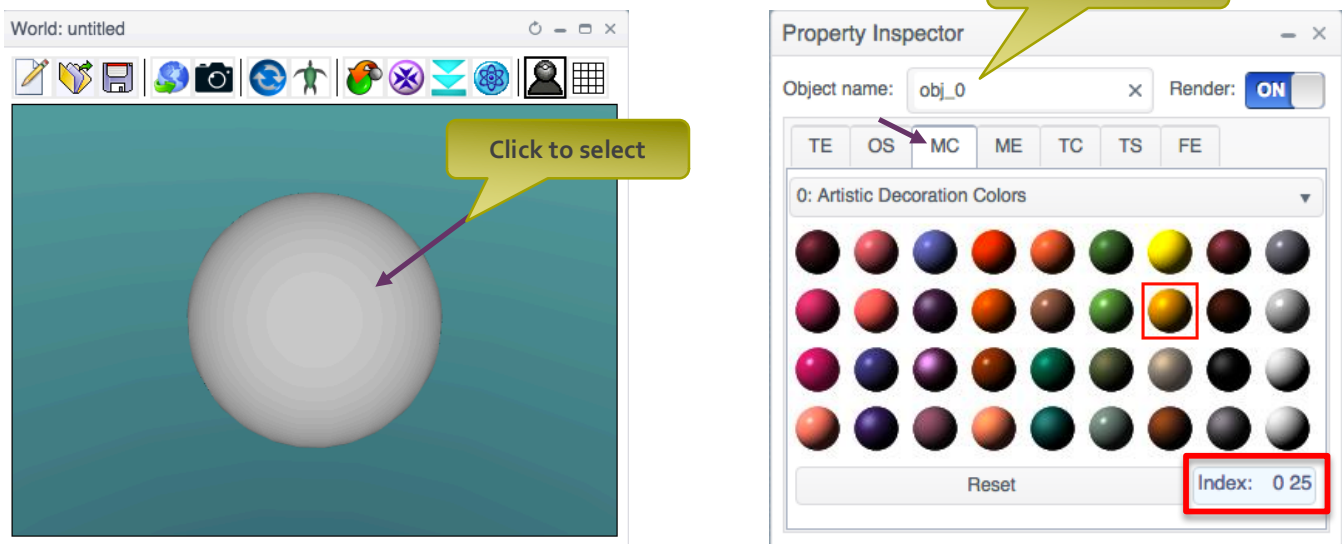


## 3.3 Material settings

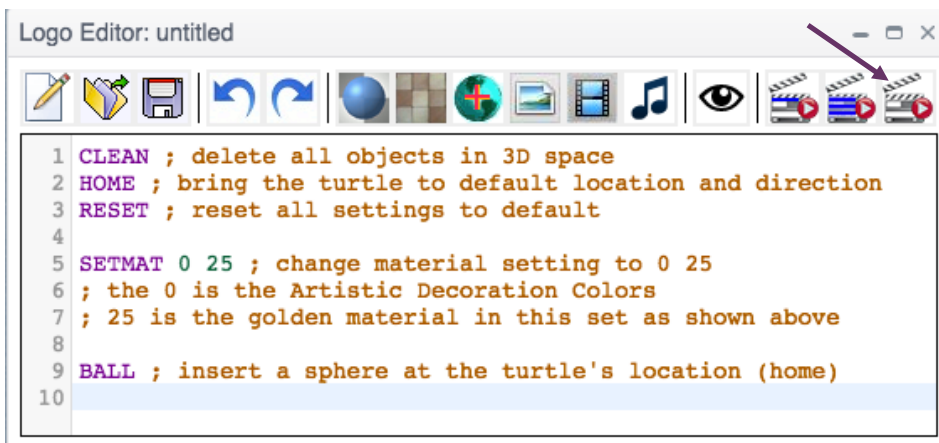
For 3D and 3D objects, you can apply material settings. Material settings define the object's colours with respect to environmental lightings. There are two ways to apply material settings onto geometric objects.

### Via Material Chooser (MC) tab in Property Inspector window.

Click on the geometric object in the 3D World window to select it, then click on a material in the Material Chooser to apply.



### Via LOGO programming



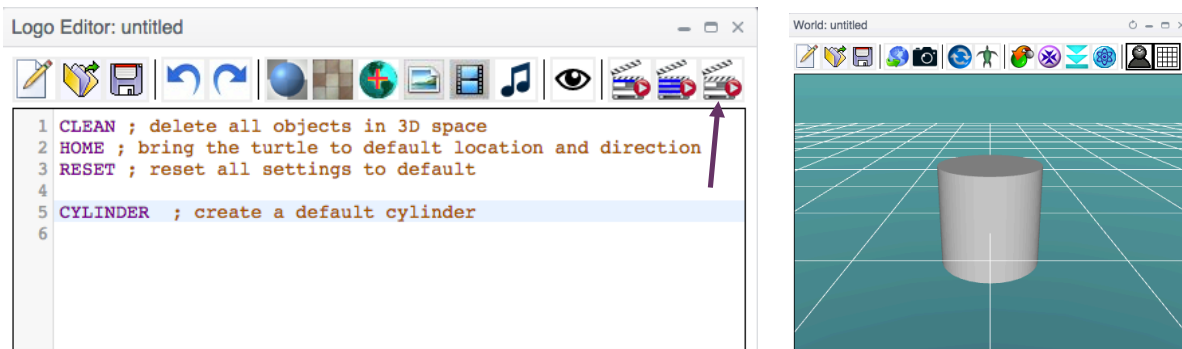
The **SETMAT** or **SETMATERIAL** command will make all subsequent objects the same material setting.

For advanced material settings, you can try the **ME** (Material Editor) tab next to **MC** tab, to further customised materials.

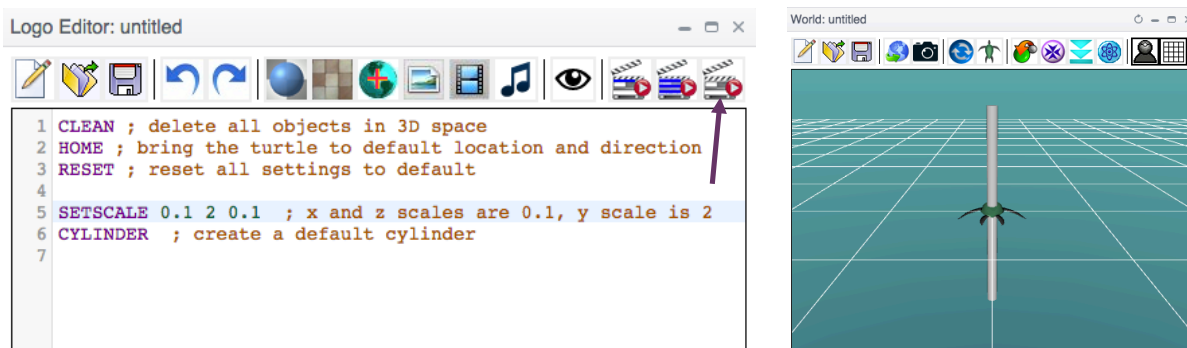
See <https://vrmath2.net/content/how-change-material-settings-virtual-object> for more details.

## 3.4 Scaling objects

Before inserting an object, you can specify a scale so the object can be scaled to the size you need. In VRMath2's LOGO, we use **SETSCALE**  $x$   $y$   $z$  to change the scale before creating a 3D or 2D object. For example, a default **CYLINDER** has height and diameter as 1, see below:



To make a thin and long cylinder (to represent a bonding between two atoms), we can use **SETSCALE** before the **CYLINDER** command. See below:



With the **SETSCALE 0.1 2 0.1** command, we created a cylinder that has height as 2, and diameter as 0.1. Note that the 2D and 3D objects will always refer to the turtle's location and direction at the time of their creation. And once the scale has changed, it will affect the rest of 2D and 3D objects being created, unless you specify a different scale such as **SETSCALE 1 2 0.5** or **RESET**.

**THINK:** What if you want a cylinder that is tilted 45 degrees? What if you want the cylinder to have different material (colour)?

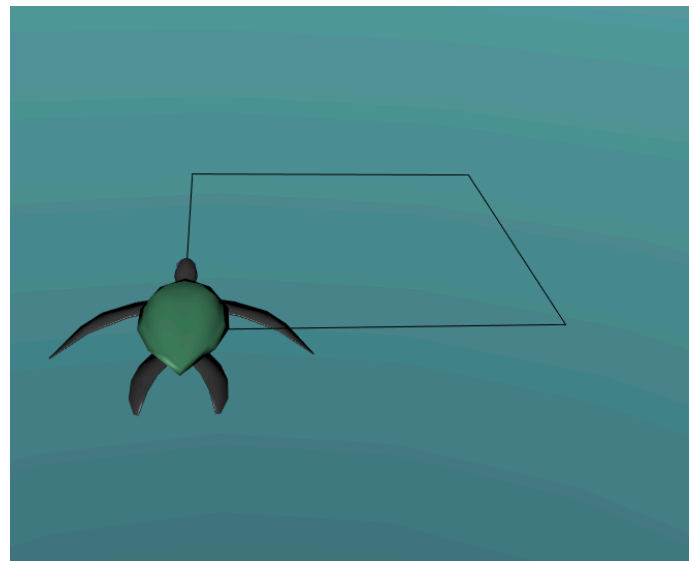
## 3.5 Loop

When you have identified a pattern or you just want repeat some codes, use a loop structure can make your codes more efficient. In LOGO, there are some loop structures and the most common one is the **REPEAT** command. The syntax of **REPEAT** is as below:

**REPEAT** *number\_of\_repeat* [ codes to repeat ] ; it must use square brackets

For example, we can program the turtle to walk a square using the following codes:

```
Logo Editor: untitled
1 ; a sqaure
2 cs reset ; clearscreen and reset
3 pd ; pendown
4 forward 1
5 right 90
6 forward 1
7 right 90
8 forward 1
9 right 90
10 forward 1
11 right 90
12 pu ; penup
13
```



In this program, we repeated `forward 1 right 90` for four times. Therefore, we can simply replace line 4 to line 11 with the following command:

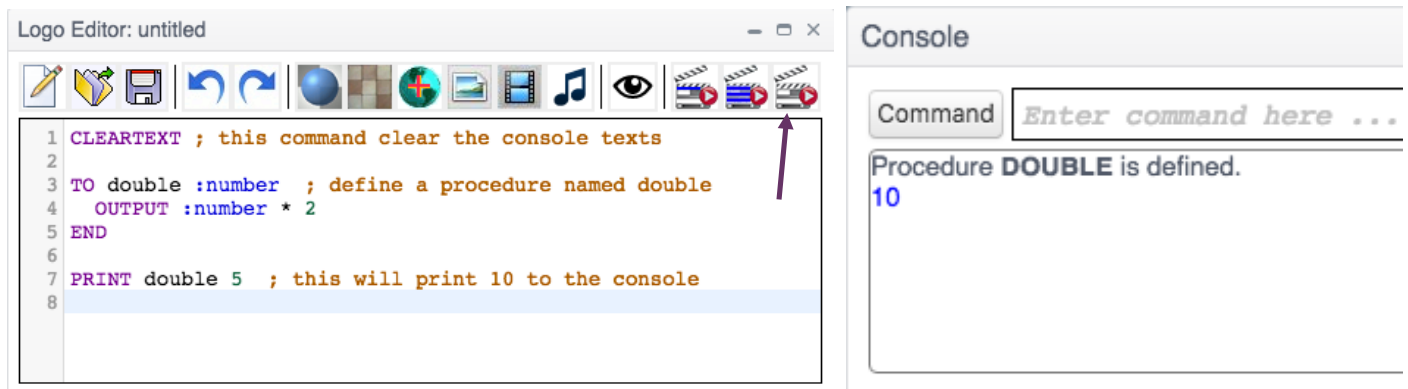
**REPEAT** 4 [ `forward 1 right 90` ] ; note that there is a space between command and input

Within the square brackets of a **REPEAT** command, if you want to know the current repeat number, you can use **RECCOUNT** or simply **#** to get the repeat number. For example,

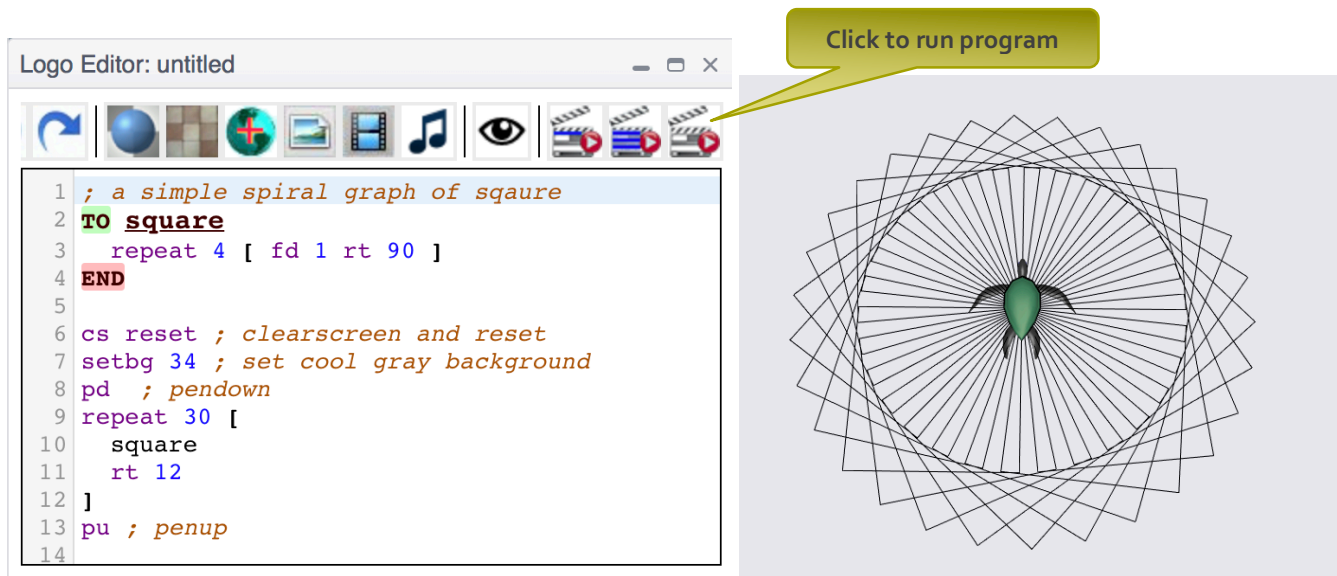
**REPEAT** 100 [ `print #` ] ; This will print number 1 to 100 in the **Console window**.

## 3.6 Procedures

In computer programming, a procedure is a block of codes that can be reused or used to handle some specific work such as to calculate sum of two numbers. Therefore, procedures are also called sub-routines and/or functions. In LOGO, to define a procedure, we use **TO** and **END** keywords. Below is an example of a function named **double**, which **OUTPUT** the double of the input number.



Note that a procedure can have zero or many inputs, and can have none or one output. In the previous square example, we can make a procedure named **square** to simplify the creation of square and reuse for a simple spiral graph. Below is an example of a simple spiral graph.



**THINK:** Can you use the 3D rotation commands in 3.1 to create some 3D spiral graphs?

## 3.7 Save your works

As a registered user and a community member of VRMath2, you can save your work in your personal folders.

In VRMath2 Editor, you can save 4 types of files:

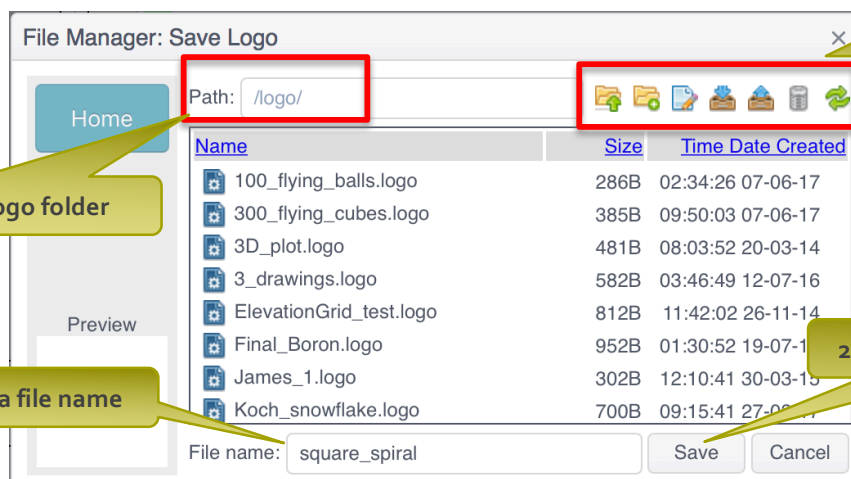
- .vrml file: the vrml file is the editing-in-progress 3D world. You can save the unfinished 3D worlds in the **3D World** window.
- .x3d file: the x3d file is an international web standard 3D files. X3d files can be published from the **3D World** window as well, and can be later used in blogs to show the whole world your creation.
- .png file: in the **3D World** window, you can take a screenshot of your 3D world and save as an image in .png format. You will need a .png image file when you blog.
- .logo file: the .logo file is the ultimate 3D generator in VRMath2. It is easier to modify the LOGO programs to change the 3D worlds than to directly modify the 3D worlds.

By now, we have created a LOGO program that can generate a simple spiral graph, so it is time that we save

```
1 ; a simple spiral graph of square
2 TO square
3 repeat 4 [ fd 1 rt 90 ]
4 END
5
6 cs reset ; clearscreen and reset
7 setbg 34 ; set cool gray background
8 pd ; pendown
```

the LOGO program.

In Logo Editor, click on the disk icon to bring up the file manager dialog.



File Manager functions

Good to keep in a logo folder

1. Type in a file name

2. Click to save program

The file name can only contain alphabets, numbers and underscore ( \_ ). Note that if the browser window is closed, any unsaved work will be lost.

## 4. Programming fractals

Fractals, when described mathematically, could be presented as some formulae, rules, and/or algorithms. They are abstract and their dimensions are not integers (and hence fractals). Computer programming and graphics are great tools to visualise fractal geometry in both 2D and 3D (

There are a few techniques for generating fractals. (see

[https://en.wikipedia.org/wiki/Fractal#Common\\_techniques\\_for\\_generating\\_fractals](https://en.wikipedia.org/wiki/Fractal#Common_techniques_for_generating_fractals)). In this workshop series, we will focus on two methods using Logo programming language:

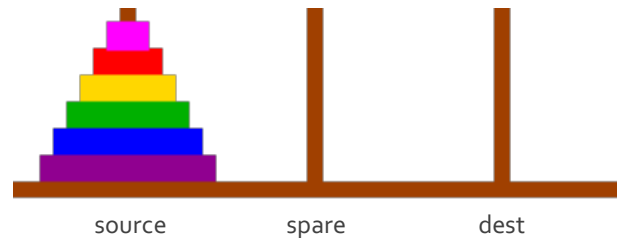
1. Iterated or recursive functions
2. Random or chaotic games

## 4.1 Recursive procedure

An active procedure that is invoked from within itself or from within another active procedure is a *recursive* procedure. Such an invocation is called recursion. In computer science, recursion also refers to the technique of having a function repeatedly call itself.

Recursive procedures can be used to solve and express many mathematical problems such as in combinatorics, sequence (e.g., Fibonacci sequence) and fractals etc. The **Towers of Hanoi** problem for example, can be expressed and solved using a recursive function (pseudocode) as below:

```
FUNCTION MoveTower(disk, source, dest, spare):  
IF disk == 0, THEN:  
    move disk from source to dest  
ELSE:  
    MoveTower(disk - 1, source, spare, dest)  
    move disk from source to dest  
    MoveTower(disk - 1, spare, dest, source)  
END IF
```



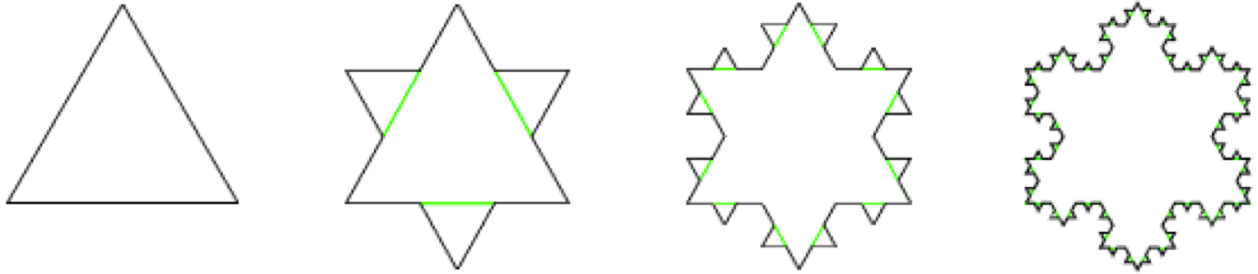
(source: <https://www.cs.cmu.edu/~cburch/survey/recurse/hanoiimpl.html>)

Because the recursive procedures or functions repeatedly call themselves, there are usually two key characteristics of recursive procedures.

1. There is an exit condition. In the above example, the  $disk == 0$  is the exit condition.
2. When the recursive procedure calls itself, it calls by reducing or increasing the condition variable, so the exit condition eventually can be met. In the above example, the  $disk - 1$  is the mechanism to meet the exit condition so the procedure will not fall into some infinite loop.

In this workshop series about programming fractals, the fractals and the recursive computer programs are both considered mathematical. The next sections will introduce how to write recursive procedures in Logo programming language to generate fractals.

## 4.2 Koch's snowflake



In Logo Editor, open the Logo file from:

Example → fractal → 2\_snowflake\_triangle → Koch\_snowflake.logo

```
1 ; Koch's Snowflake
2 ; The snowflake starts from an equilateral triangle
3 TO koch :count
4   rt 30 side :count
5   rt 120 side :count
6   rt 120 side :count
7 END
8 ; Recursion function
9 TO side :count
10  ifelse :count = 1 [ fd 1 ]
11  [
12    make "count :count - 1
13    ; the 4 lines below are for the iteration pattern
14    side :count
15    lt 60 side :count
16    rt 120 side :count
17    lt 60 side :count
18  ]
19 END
20
21 reset cs setbg 29 ; reset settings
```

Click to run program

### Questions: (predict)

1. What is the exit condition of the **side** procedure?
2. What is the purpose of the variable **:count**?
3. What is the purpose of the **koch** procedure?
4. From line 4 to line 6 in the **koch** procedure, what shape is implied?
5. What is the iteration pattern (or motif) as can be seen from line 14 to line 17?



## Experiments:

1. Try different numbers of iteration (**:count**) in line 25, then run the program to see the result. For example, **koch 1**, **koch 2** ... and so on.
2. What will happen if you comment out line 6 as below? (The symbol semi-colon ; denotes comment)

```
3 TO koch :count
4   rt 30 side :count
5   rt 120 side :count
6   ; rt 120 side :count
7 END
```

3. Line 4 to line 6 in the **koch** procedure is in fact the 3 sides of a equilateral triangle. What if we add one more line in this procedure, and change all the angles to 90 degrees (for a square shape)? What about other shapes or any arbitrary angles you like? For example:

```
3 TO koch :count
4   rt 45 side :count
5   rt 90 side :count
6   rt 90 side :count
7   rt 90 side :count
8 END
```

4. Line 14 to line 17 in the **side** procedure defines the recursion pattern (motif).

```
9 TO side :count
10  ifelse :count = 1 [ fd 1 ]
11  [
12    make "count :count - 1
13    ; the 4 lines below are for the iteration pattern
14    side :count
15    lt 60 side :count
16    rt 120 side :count
17    lt 60 side :count
18  ]
19 END
```

The pattern is looking like below:



What will happen if the pattern is changed to:



5. Carry on from the above, you may design different motif (pattern) to create different fractals.

## Questions:

Back to the original Logo program, for **koch 1**, we will get an equilateral triangle with perimeter as 3. What will the length of perimeter change to if the number of iteration increases?

## Challenge:

Using the other rotation commands (see page 8), add in some commands (programming codes) to make more snowflakes in 3D space (different planes).

## 4.3 Sierpiński's triangle

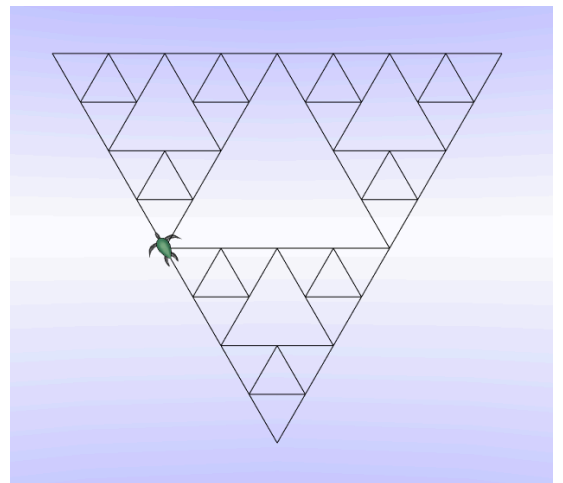
In previous snowflake activity, you should have noticed that the Sierpinski triangle can be created using the snowflake program with the motif (pattern as below):



In Logo Editor, you can open the Logo file from: Example → fractal → 2\_snowflake\_triangle → sierpinski\_triangle.logo to get the triangle as below:

```
Logo Editor: sierpinski_triangle.logo
1 ; Sierpinski triangle fractal by Andy
2 ; code modified from http://logo.twentygototen.org/_REo_2F2
3 TO side :count :length
4   ifelse :count = 1 [ fd :length ]
5   [
6     make "count :count - 1
7     side :count :length
8     lt 120 side :count :length
9     rt 120 side :count :length
10    rt 120 side :count :length
11    lt 120 side :count :length
12  ]
13 END
14
15 TO triangle :count :length
16   rt 30 side :count :length
17   rt 120 side :count :length
18   rt 120 side :count :length
19 END
20
21 reset
```

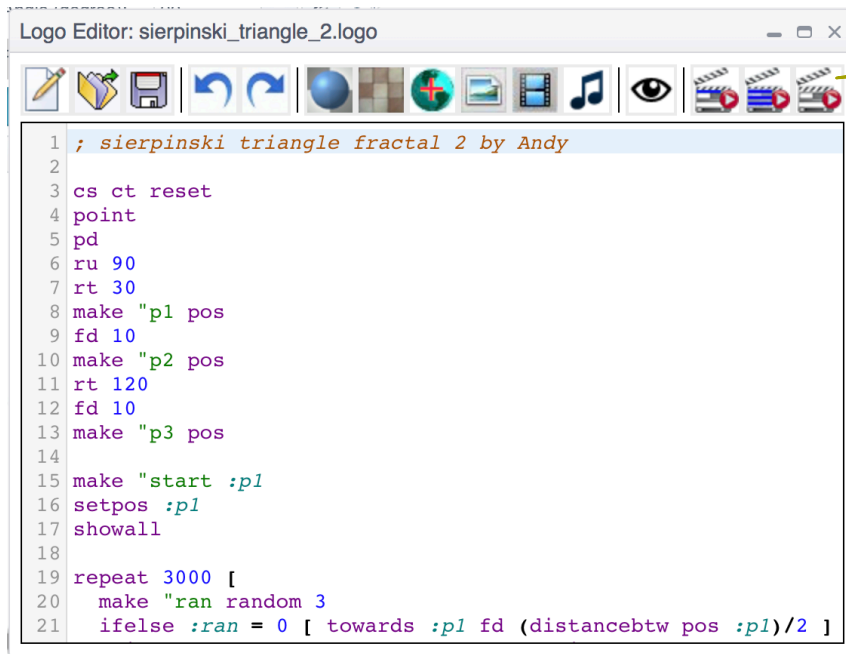
Click to run program



Here, we want to introduce a different method (not recursion) to generate Sierpinski triangle. This method is called the random or chaos games method. The Sierpinski triangle can be created if the following steps are followed:

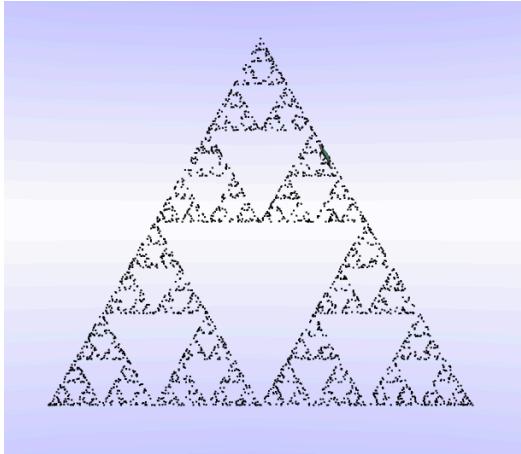
1. Take 3 points in a plane to form a triangle, you need not draw it.
2. Randomly select any point inside the triangle and consider that your current position.
3. Randomly select any one of the 3 vertex points.
4. Move half the distance from your current position to the selected vertex.
5. Plot the current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

The sierpinski\_triangle\_2.logo implements the about steps.



```
1 ; sierpinski triangle fractal 2 by Andy
2
3 cs ct reset
4 point
5 pd
6 ru 90
7 rt 30
8 make "p1 pos
9 fd 10
10 make "p2 pos
11 rt 120
12 fd 10
13 make "p3 pos
14
15 make "start :p1
16 setpos :p1
17 showall
18
19 repeat 3000 [
20   make "ran random 3
21   ifelse :ran = 0 [ towards :p1 fd (distancebtw pos :p1)/2 ]
```

Click to run program



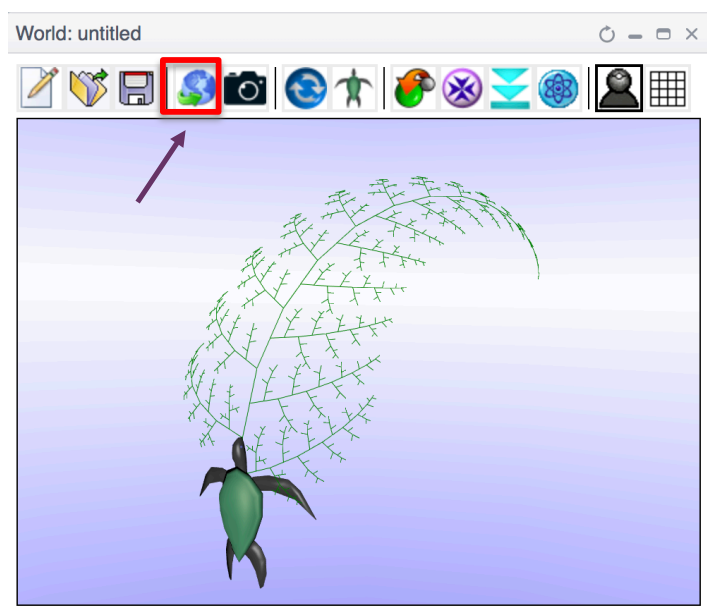
### Experiments:

1. Would this method work for any triangles such as right triangle and obtuse triangle?
2. What about four vertices of a square?
3. Would this work for 3D such as a tetrahedron or a square-based pyramid?

**Note:** you may open example 3D sierpinski logo files in the Example folder.

## 5. Blogging and publishing

Publishing 3D models (worlds) and blogging in VRMath2 website is the way we learn and share in VRMath2 online learning community.



The term “publish” in VRMath2, is specifically referred to publish 3D models and save as .x3d files.

However, until you blog your 3D models, they are just files in your personal folders and no one can see them.

Blogging in VRMath2 is like writing a personal diary about your learning journey. Through blogging, you can practice your writing and communication skills. And when you write about your programming and virtual worlds, you will be learning and applying

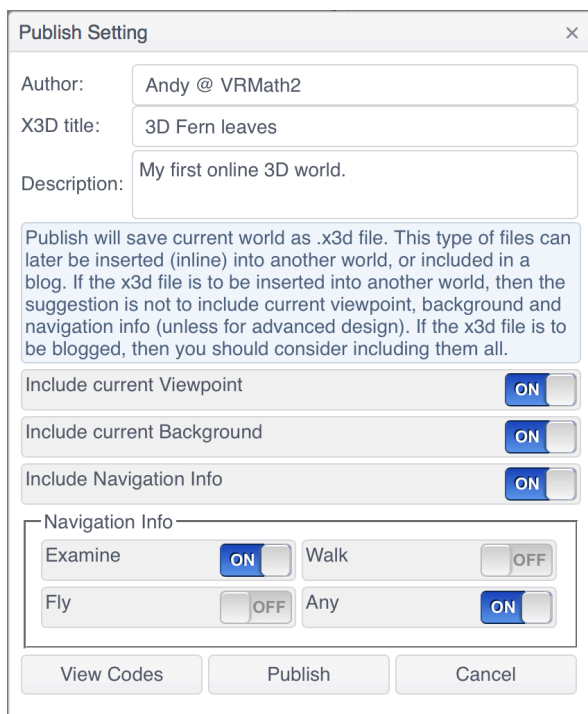
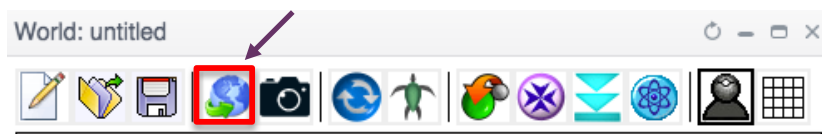
science, technology, engineering and mathematics (STEM).

To blog, you will need to prepare 3 essential files. They are the **.x3d** (3D model), **.png** (an image of 3D model) and **.logo** files. Please read on the following sections for more details.

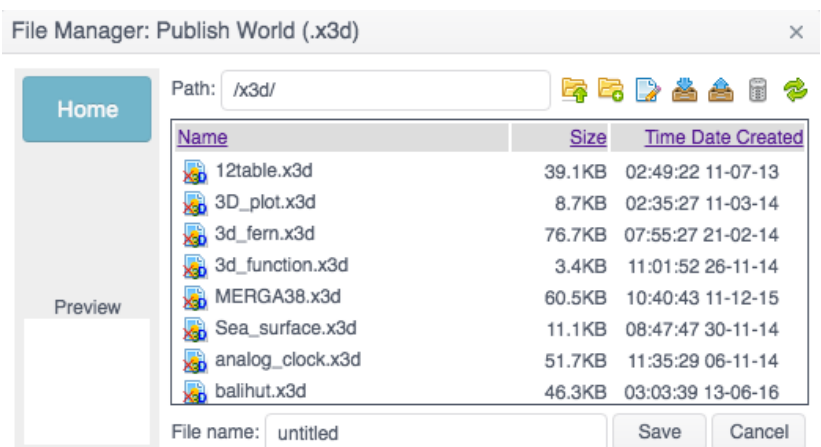
# 5.1 Publishing 3D model

The 3D models produced in VRMath2 Editor are in the international standard X3D format. This means that you can embed your 3D models online and share to the world in any modern web browsers. More details on <https://vrmath2.net/content/x3d-behind-scenes>

When you have created your 3D models (draft or complete) in the **3D World** window, you may publish and save your 3D models. To do so, simply click on the publish icon to bring up the **Publish Setting** dialog.



As indicated in the dialog, there are two types of x3d files to publish. If the 3D model is for blogging purpose, it is suggested that you include viewpoint, background and navigation info (as show in the image on the left). After filling in the Author, title and description info, click on the Publish button to bring up the **File Manager** dialog.

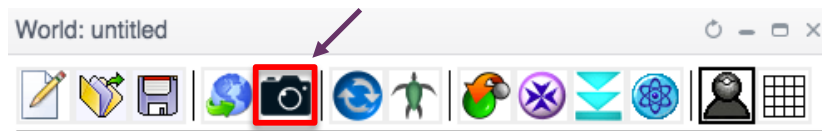


By default, you will save in the **x3d folder**. Provide a sensible file name then click on **Save** button to finish.

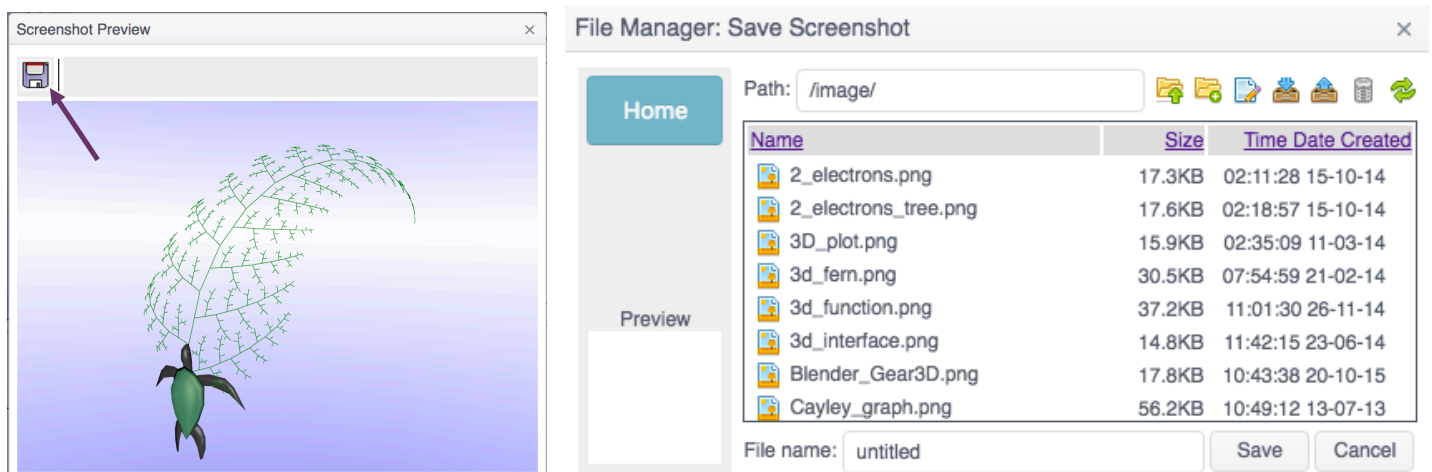
## 5.2 Save an image for blog

An image of your 3D model can be uploaded when blogging, as a thumb nail for your blog. Readers can then see a picture of your 3D model before they click to read your full blog. This is a key aspect of blog in VRMath2 Community.

To save an image of your 3D model, click on the camera icon in the toolbar as below:



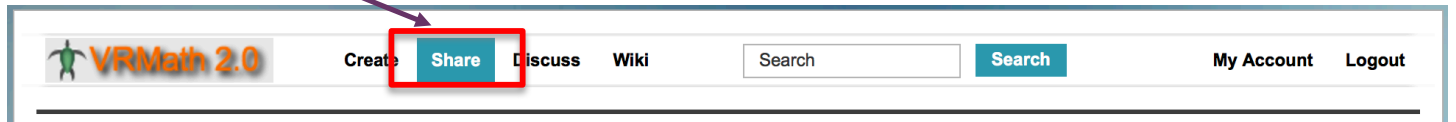
You will then see a **Screenshot Preview** window. If you are happy with the screenshot, click on the disk icon to bring up the **File Manager** dialog. For example,



The default folder for saving screenshots is the **image folder**. Provide a sensible file name then click on **Save** button to complete.

## 5.3 Write a blog

To write your blog about your 3D virtual world, click on the **Share** in the main menu of home page.



Then, in the share page, click on **Write a blog**.

### Share your creations to the world

[in Share](#) [Tweet](#) [Like 0 Share](#) [G+ Share](#) [Email](#) [Save](#)

In VRMath2 Community, you can share your creations of **Images**, **Logo programs** and **3D virtual worlds** from [VRMath2 Editor](#) in two ways:

- Write a blog**

All registered members can [write blogs](#) in VRMath2 Community. A blog is a form of online article for you to freely express your ideas about your creations. You can explain your ideas with rich-format texts, images, Logo programming codes and of course 3D virtual worlds. Blogs have commenting facility below them, so other members can interact and give you comments about your blogs. Each blog page also has social plugins such as LinkedIn, Twitter and Facebook etc. so all readers can share your blogs further to their social networks.

You will then have the blog form to fill in. It is recommended that you read the **Blog guides** on the right side first before writing. Then, the first part of blogging is title and blog image.

# Create Blog entry

Title: \*

1. Type in a sensible title for your blog.

Vocabularies

Categories: \*

- Science
- Biology
- Chemistry
- Earth and Space
- Physics
- Technology
- Control
- Material
- System

2. Select at least one category.

Must select at least one category. Hold Control key (PC) or Command key (Mac) to select multiple categories.

Tags:

3. Type in custom tags if any.

Enter a comma separated list of words. E.g., House, Dining table

Image:

Upload | Remote URL

Open File Browser

Choose File no file selected

Upload

Maximum file size: 1 MB

Allowed extensions: .jpg .jpeg .png .gif

4. Click on this link to choose your blog image.

## Blog guides

Please observe **Netiquette** when posting.

### Title: \*

The title of your blog.

### Image:

This is the teaser image for your blog. You can **upload**, **browse** from your home directory, or **transfer** from other website with an url. If leave blank, a default image will be used.

### Vocabularies:


**Categories** are the system's classification based on STEM.

**Tags** are user's own key words.

### Body:

The **File Browser** will open in a new tab. You need to click into your **image folder** to find the image you saved in previous section (6.2). Select your image file then click on **Insert file** to close the dialog.

Image:



Alternate Text:

This text will be used by screen readers, search engines, or when the image cannot be loaded.

Title:

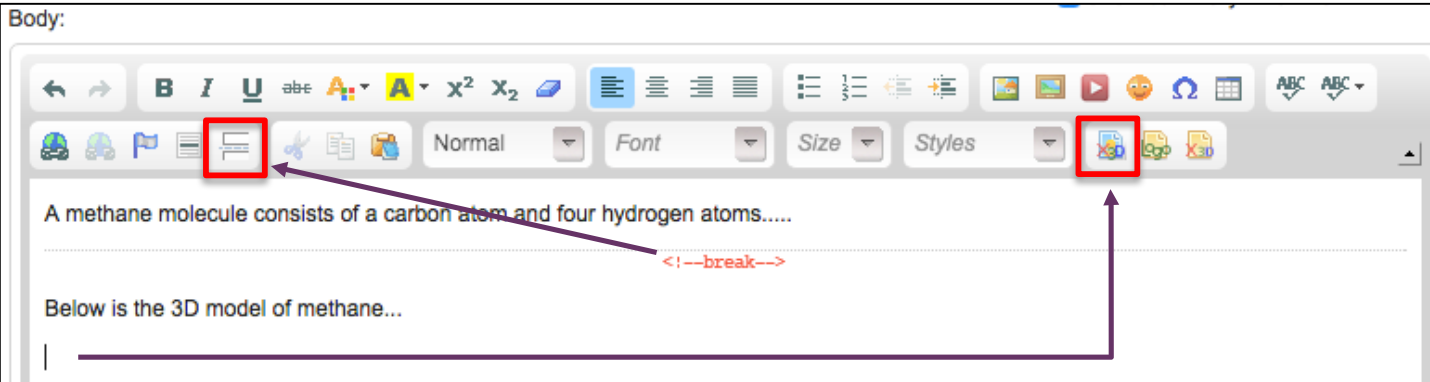
The title is used as a tool tip when the user hovers the mouse over the image.

Remove

If successful, you should see your image displayed and you are asked to enter **Alternate Text** and **Title** for the image. Please do provide a text and title for the image.

Then it will be the main blog body. You should write an introduction paragraph, then click on **Teaser break** icon. The content before the teaser break is treated as a summary to be displayed in the Group page.

Body:



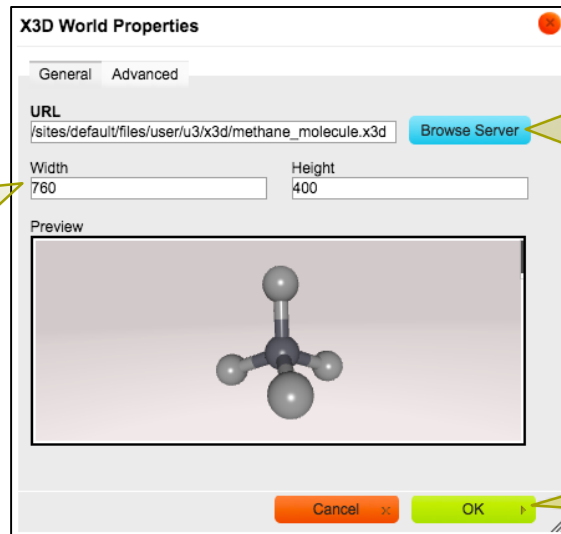
A methane molecule consists of a carbon atom and four hydrogen atoms....

<!--break-->

Below is the 3D model of methane...



Next, click on the X3D icon to insert your 3D model.



2. Enter the Width as 760 and Height as 400.

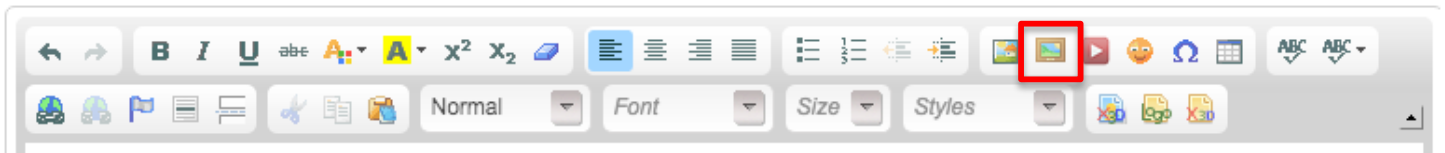
PS: You may try different numbers but 760 is maximum.

1. Click on Browse Server to select your x3d file, and insert file to close the dialog. If successful, you should see your 3D model displayed in this dialog.

3. Click Ok when done, or Cancel to restart.

After the 3D model, you could write some paragraphs to explain about your 3D virtual world and programming. At the end of the blog, insert links to your 3D model and Logo program files.

Body:

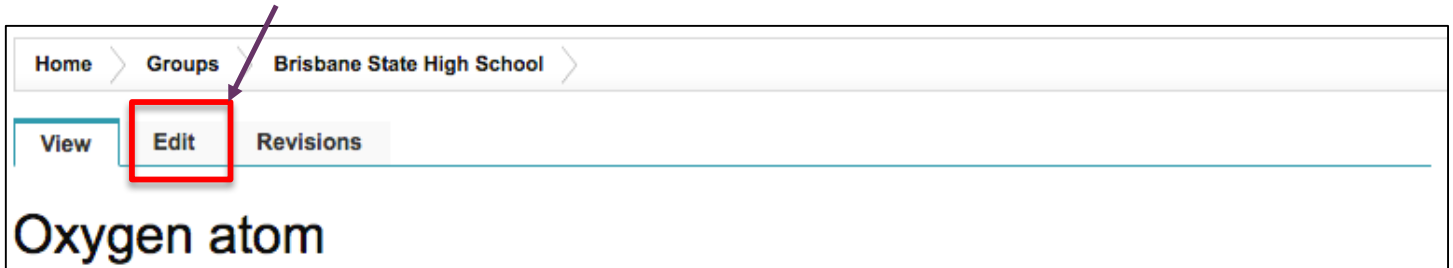


Finally, you can click on **Save** button to publish your blog.

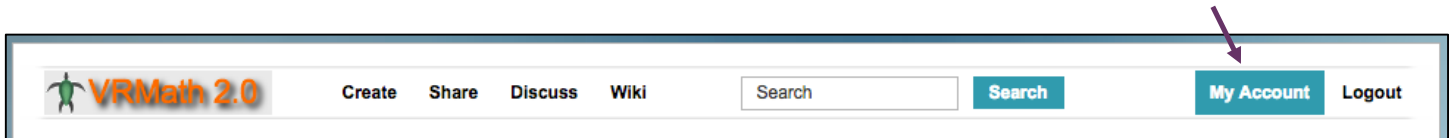
You can edit later your blog as many times as you wish. See next section for how to **Edit**.

## 5.4 Edit own blog

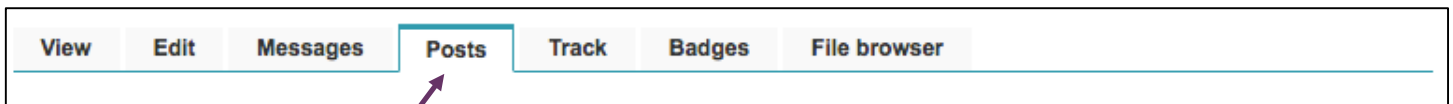
You can edit all your posts when you are viewing them. Simply click on the Edit tab to edit. For example,



In case that you cannot see your blog in the group, you can always find it from your account. Click on **My Account** at the top-right corner of website to view your account details.



Then click on Posts tab to see a list of your posts.



## 5.5 Commenting blogs

Social interaction is a key aspect of learning. When reading others' blogs, you can provide your comments and in the same manner that others may also provide their comments to your blog.

Commenting is not a part of assessment, but you are encouraged to provide your comments in the form of praise, critique, help and ask further questions.

To comment on blogs, simply click on **Add new comment** button at the end of blogs.

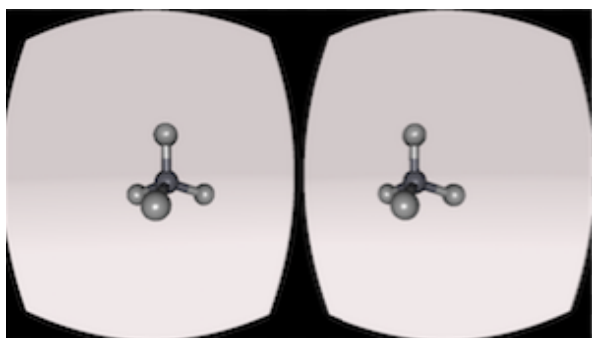


You can also reply to comments by clicking on the **Reply** button in the comments.

## 6. Viewing 3D models in CardBoard

There are different ways of sharing in VRMath2 Community. These can be blogging, commenting, discussing in forums and creating wikis etc.

In this section, we are introducing about sharing VR experiences on your 3D models using a mobile phone and a VR Cardboard.



In this workshop, you will be given a VR CardBoard to take away. The volunteer helpers will assist you to assemble the CardBoard VR box.

You will need to have a mobile phone such as iPhone 6 and android phone version 6 to better view the 3D models created and blogged in VRMath2 website.

Your mobile phone also needs to have a QR code scanner app such as i-nigma. Please download and install in your phone.

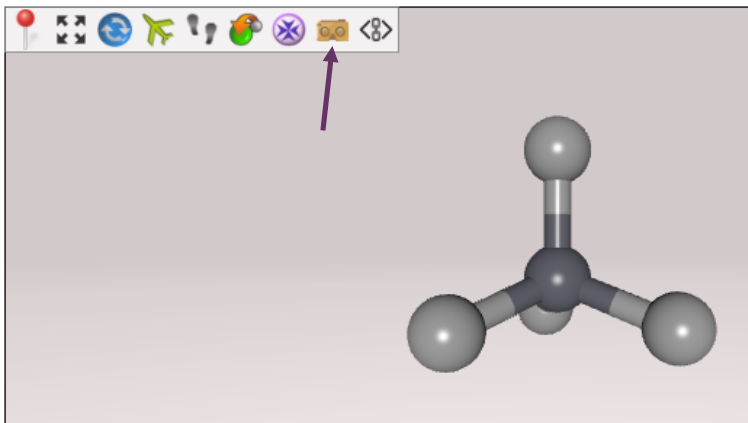
VRMath2 VRBox is an application to view 3D models in VR CardBoard. It has a web address for mobile device only at <https://vrmath2.net/vrbox>

# 6.1 Scan QR codes to view in CardBoard

There are two main ways to view your 3D models in CardBoard VR Box.

## From blogs

When viewing a blog with a 3D model in it, click on the square to get a menu as below:

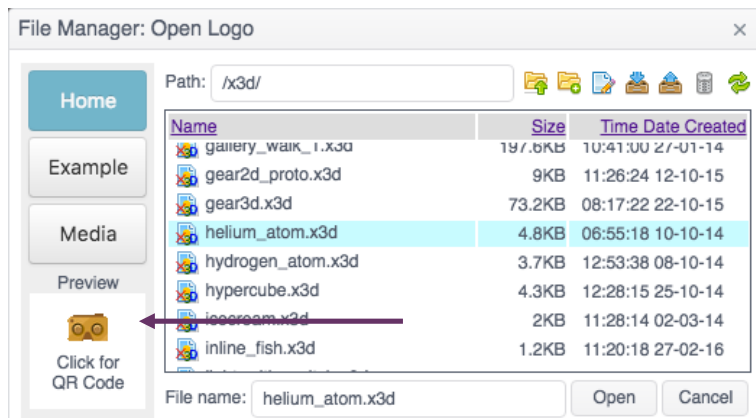


Then click on the **CardBoard** icon to get a QR code. Scan the QR code in your mobile phone to open the **VRMath2 VRBox** application in your mobile browser.



## From VRMath2 Editor

Use the **File Manager** to browser your x3d files.

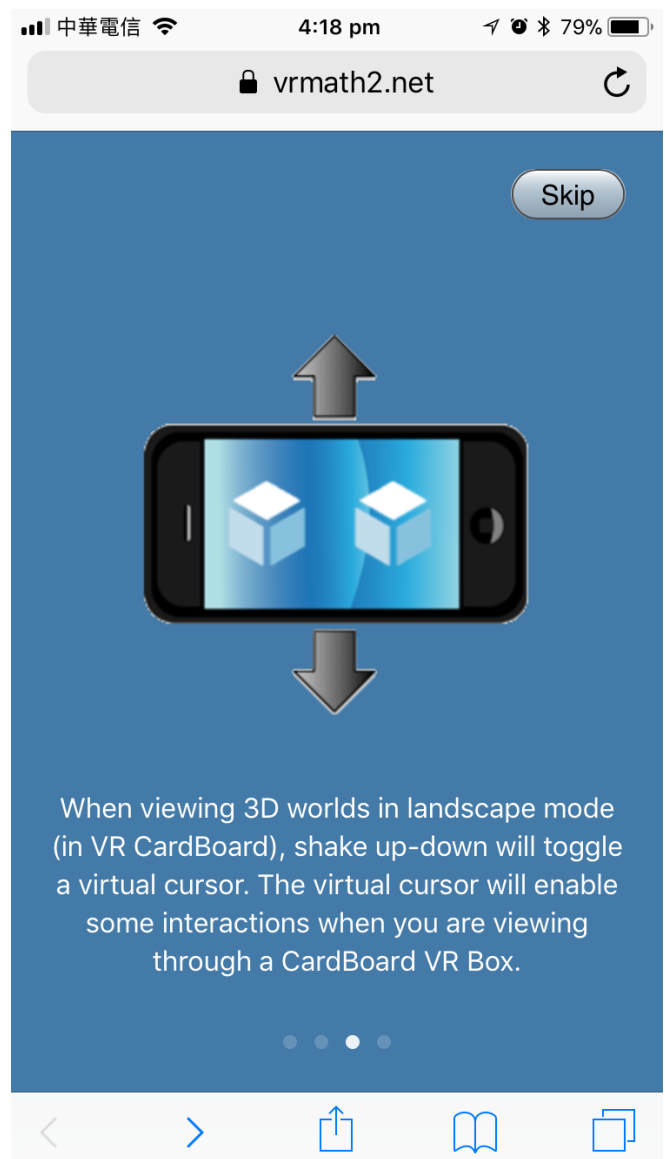


Once a x3d file is selected, click on the CardBoard icon in the Preview area. A QR code will be displayed. Scan the QR code with a mobile phone to open the VRMath2 VRBox application in a mobile browser.

Once opened, skip/end the tutorial, rotate the phone to landscape, insert your phone to your CardBoard box and enjoy.

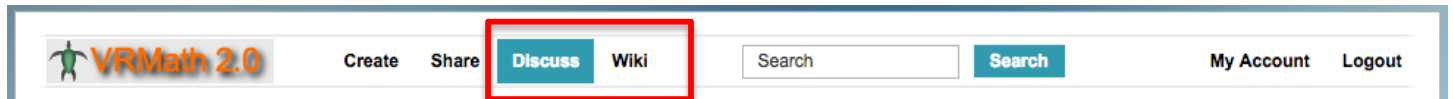
## 6.2 VRMath2 VRBox

The VRMath2 VRBox App is still in development. It can be accessed in mobile browsers directly from <https://vrmath2.net/vrbox>. In order to get the full screen in mobile browser such as Chrome and Firefox, please save the above URL on to the home screen of the mobile device. In the VRMath2 VRBox App, you can shake your mobile phone (device) to get function menu. This is introduced in a tutorial when first open the URL <https://vrmath2.net/vrbox>.







# 7. Forums and Wiki for help

After the workshop, you can still get support about VRMath2 through discussion forum and wiki. You can access forum and wiki from the top menu:



When in the **Discussion forums**, simply select an appropriate forum to ask questions and you may also answer other's questions.

Forum	Topics	Posts	Last post
 <b>Announcement</b> News and events.	11	12	Version 0.9.1 by Andy 28/12/2014 - 22:50
<b>Designer forums</b> These forums are related to the design, create and share of 3D worlds.			
 <b>Community forum</b> Discussions about the design, functions and services of VRMath2 website.	0	0	n/a
 <b>Logo programming forum</b> Discussions about Logo programming in VRMath2 Editor.	8	28	map with ... by Andy 09/07/2016 - 00:51
 <b>VRMath2 Editor forum</b> Discussions about the design, functions and interfaces of VRMath2 Editor.	2	2	ElevationGr ... by Yeh 25/05/2014 - 00:45

Wiki is a place of tutorials and documentations for VRMath2 Community and Editor. You may also help edit some of the wiki pages if you wish.

---

## The "What is" pages

Click on the above link to find some "what is" questions answered.

---

## The "How to" tutorial pages

Click on the above link to find some "How to" questions explained.

---

## The Logo Programming Guide/Reference

Click on the above link to go directly into the Logo programming guide.

---