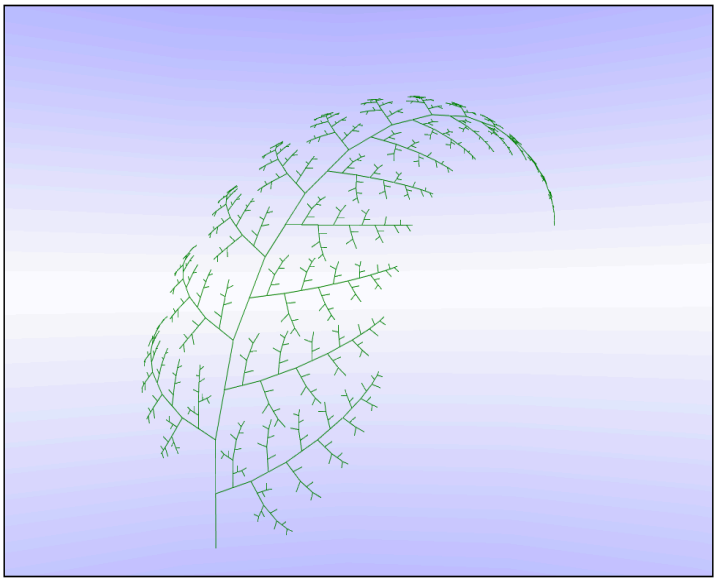
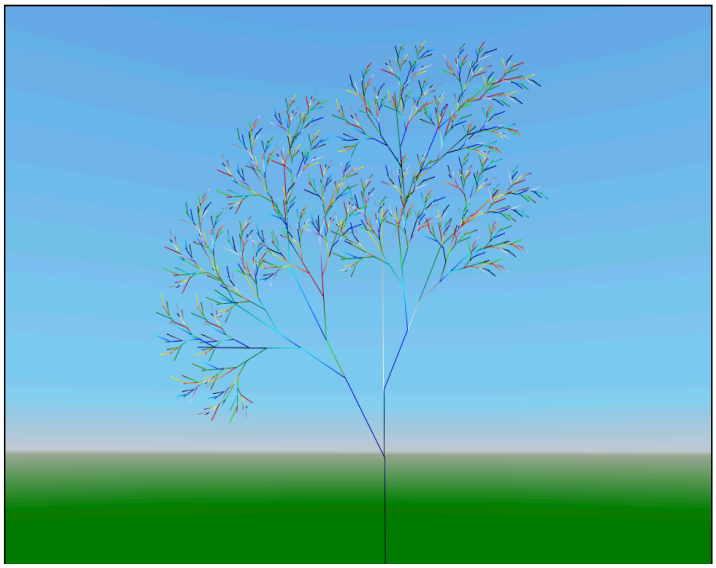
Programming Fractals in 3D Workshop 1:

Fractals in nature





```
; recursive tree
;
TO tree :size
 if :size < 5 [forward :size jumpback :size stop]
 forward :size/3
 left 30 tree :size*2/3 right 30
 forward :size/6
 right 25 tree :size/2 left 25
 forward :size/3
 right 25 tree :size/2 left 25
 forward :size/6
 jumpback :size
END
clearscreen
cm ; change to cemtimeter
line ; change to line mode
nextcoloron ; rotate pen color
ru 90
pd tree 100 pu
```

Programming fractals in 3D VR

Workshop for Asian Technology Conference in Mathematics (ATCM)

Date/Time: 16th December 2017, 4 pm ~ 5 pm

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

Handbook for Programming Fractals in 3D Virtual Reality: Fractals in Nature

VRMath2 Workshop 1

VRMath2 Community

https://vrmath2.net

is a free online learning environment for STEM

Dr Andy Yeh

a.yeh@qut.edu.au

Queensland University of Technology

Contents

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 brief introduction to Logo programing and turtle geometry, then we will focus on creating fractals in the nature, specifically on creating:

1. Fern leaves and variants in 2D and 3D.
2. Trees variants in 2D and 3D.

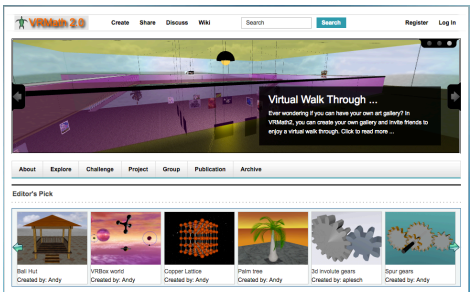
Requirements

- Computer lab (Windows or Mac) with Internet access.
- Web browser (Chrome, Firefox or Safari).
- No Logo programming experience required.
- 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

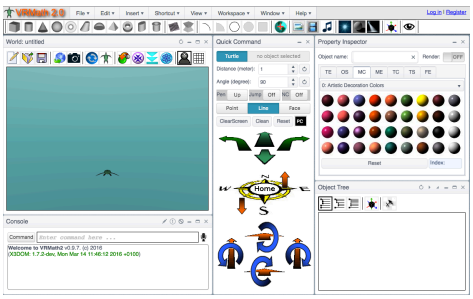
VRMath2 stands for **V**irtual **R**eality **Math**ematics **2**.0. It started in Year 2001 as an online learning environment for constructing knowledge of 3D geometry. Over the year, it has evolved to become a STEM (Science, Technology, Engineering and Mathematics) integrated learning environment. The 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)

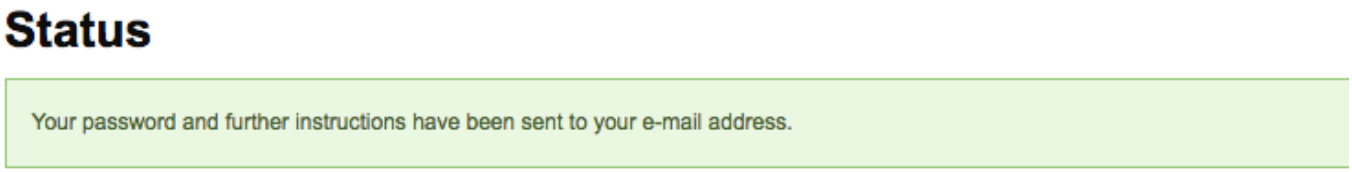
This is a recent application and is still developing. Using a CardBoard VR Box with a mobile phone (e.g., iPhone 6 and above, Android 6 and above), you can view all your 3D contents created and all 3D contents shared in VRMath2 websites. In this workshop, you will be able to see your 3D models or any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

2.1 Create an account

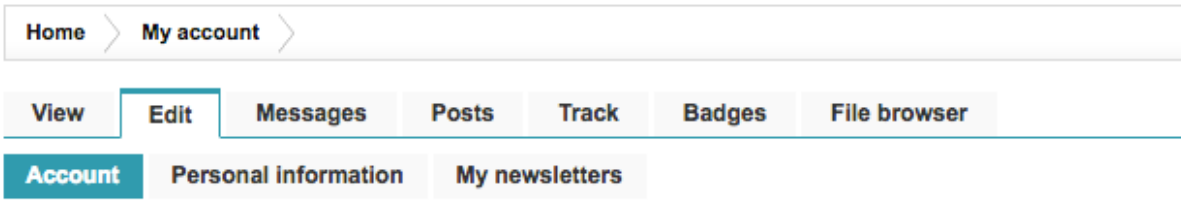
VRMath2 is a free online learning environment. Without needing an account to login, visitors can browse most of the website contents and create 3D contents with VRMath2 Editor. However, with an account, users can further share, discuss, comment and save files to personal folders.

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).



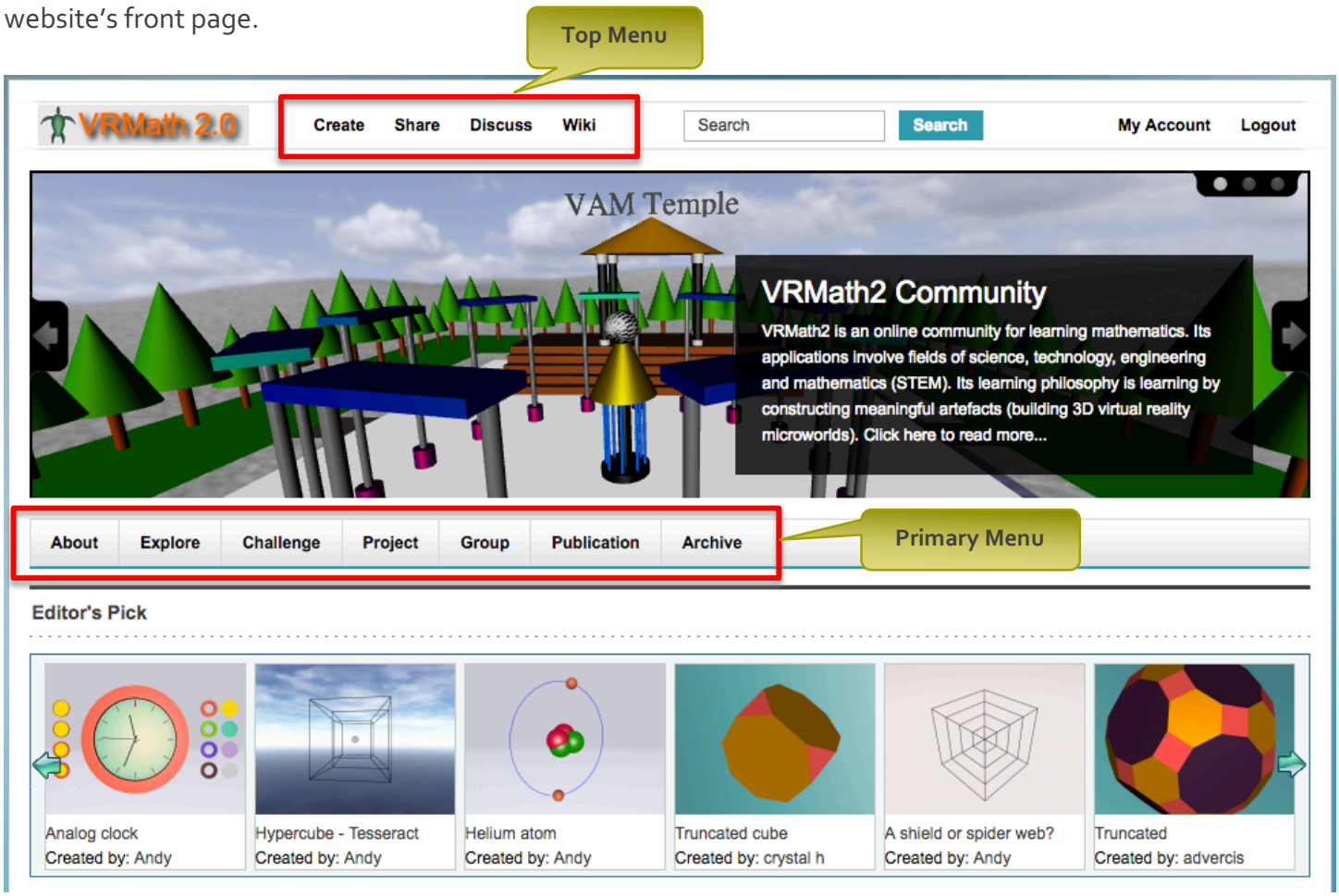
It is highly recommended that you click on the one-time login link, which will bring you to your account. Then you can click on the **Edit** tab to change password, select/upload own account picture, add your real name and other information.



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.



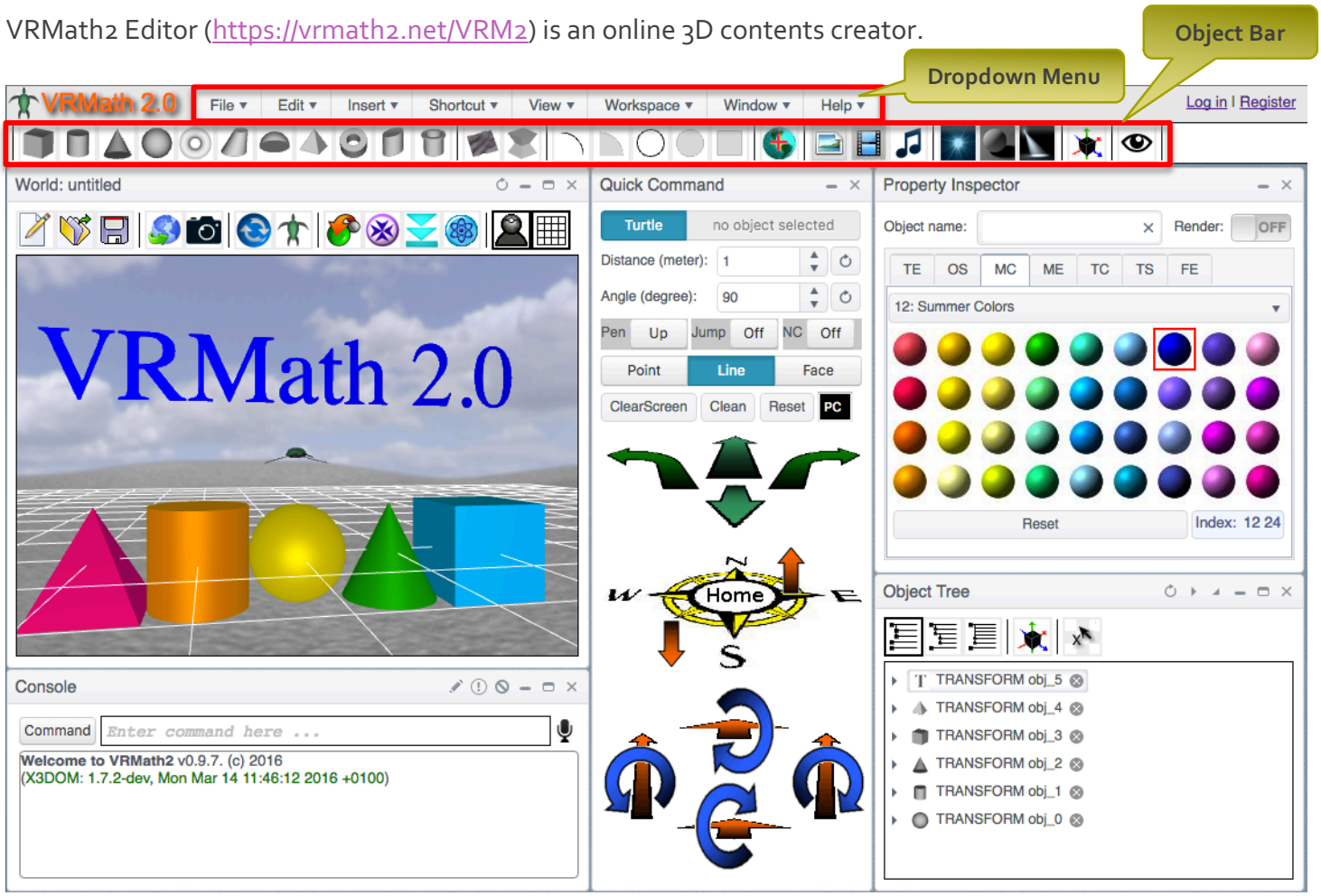
The **Top Menu** has four main functions:

- **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.



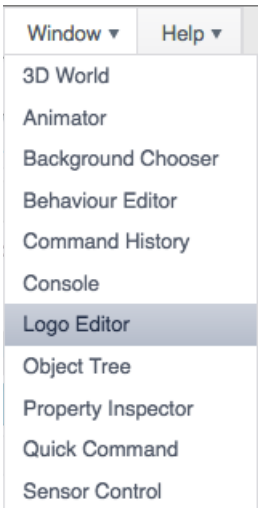
The **Dropdown Menu** has access to most functions in VRMath2 Editor. For example, all **Object Bar** functions (insert primitive objects) can be accessed as well from the **Dropdown Menu**. It is suggested that you click on these menus and try out, particularly the **Workspace** and **Window** menus.

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

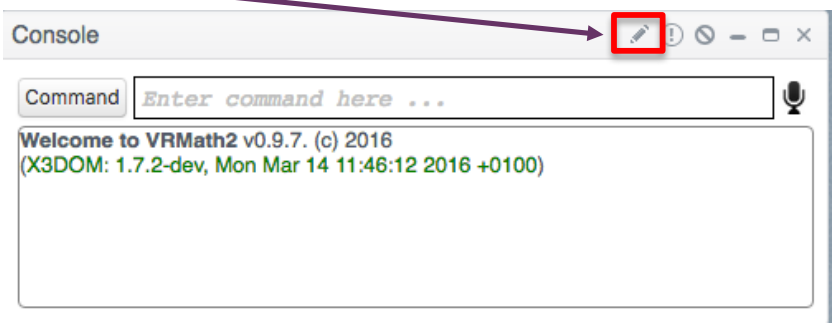
3D Navigation

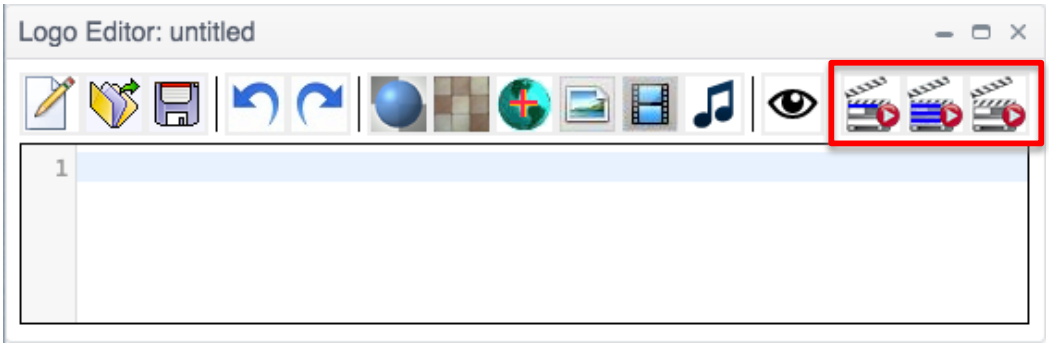
It is essential to navigate in the 3D window so to inspect where the turtle is and what you have created. You can try out the icons in the Toolbar, particularly the Reset Viewpoint  , Fit All  , and Set Rotation Centre  are very useful when navigating in 3D virtual space. When using a touch screen, you 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.

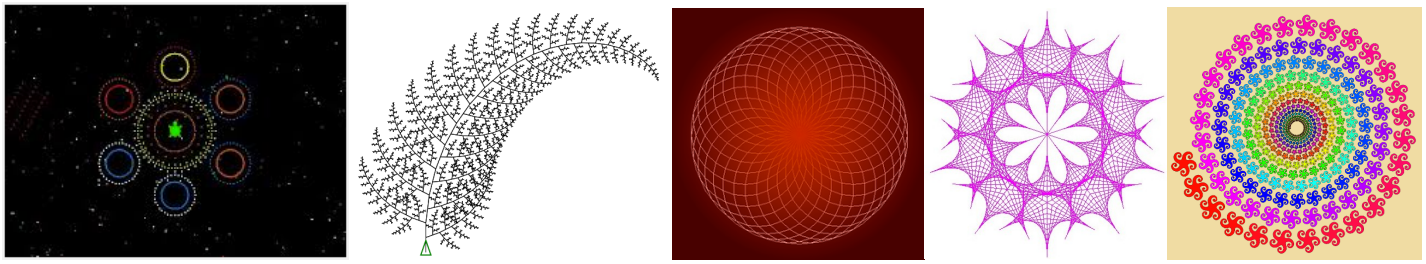




The three icons at the left will run your codes in **per line**, **per block** (selection), or **all** program.

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*

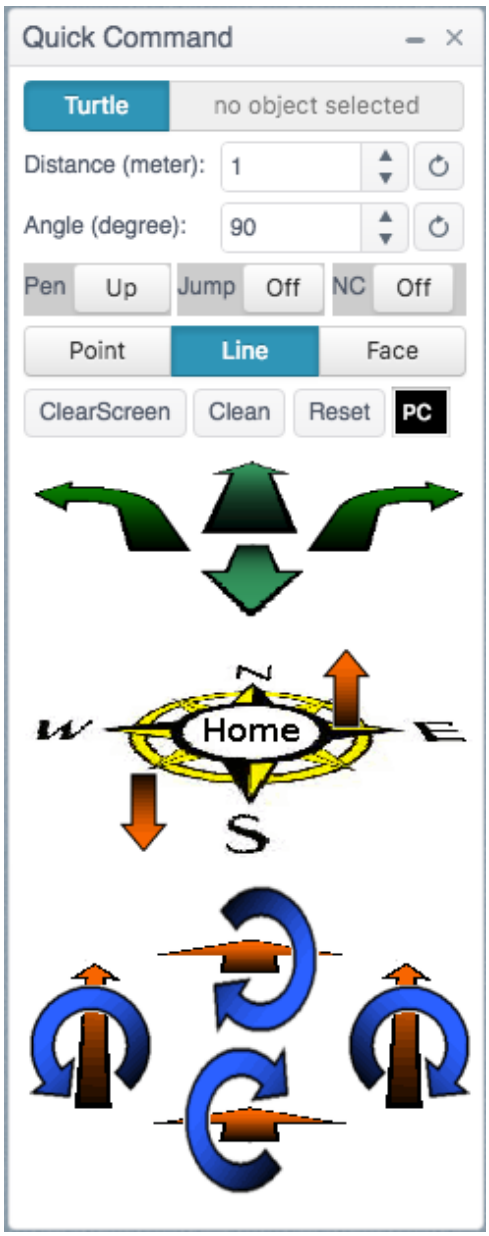
The distance is specified as METER or CENTIMETER. The above commands can be trialled in the **Quick Command window**.

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:

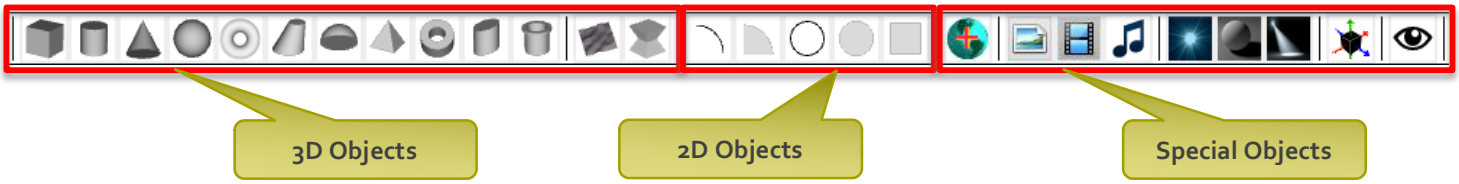
https://vrmath2.net/content/how-move-turtle-3d-virtual-space



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.

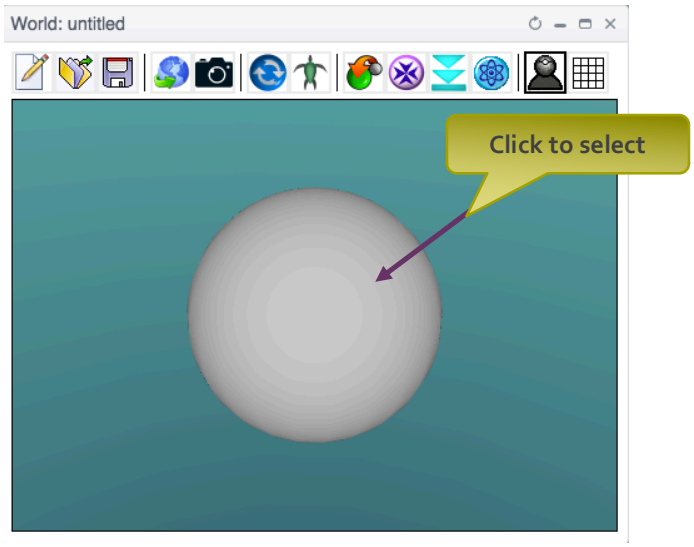
```
HOME  ; this brings the turtle to home location (coordinate 0 0 0), facing north (-Z).
TILTLEFT 180  ; tilt left 180 degrees is the same as flipping the turtle
CONE  ; this cone will be up-side-down as the turtle
```

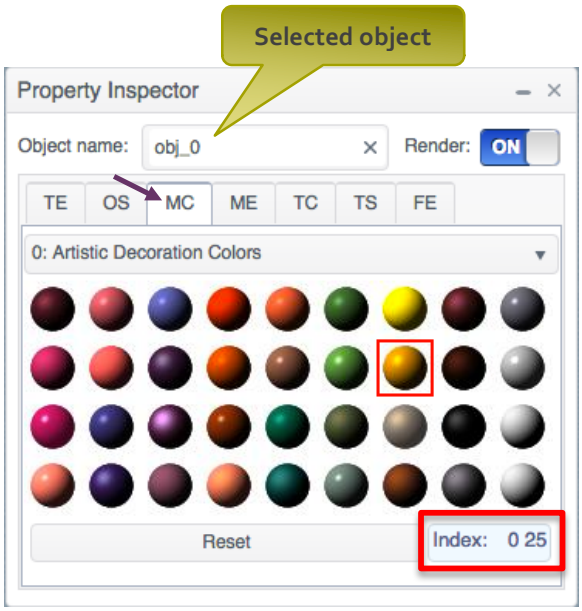
3.3 Material settings

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.

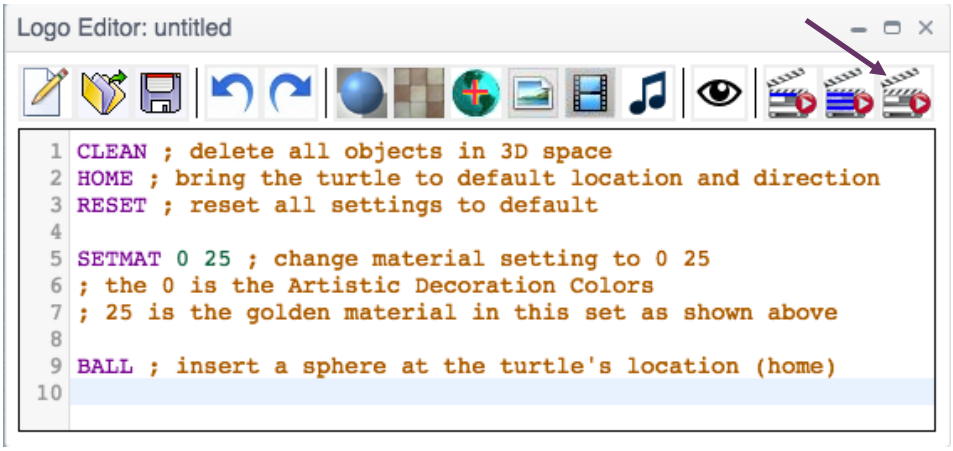
Via Material Chooser (MC) tab in Property Inspector window.

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



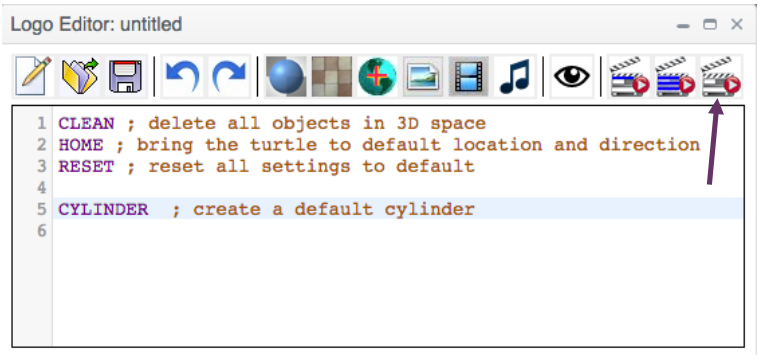
The **SETMAT** or **SETMATERIAL** command will make all subsequent objects the same material setting.

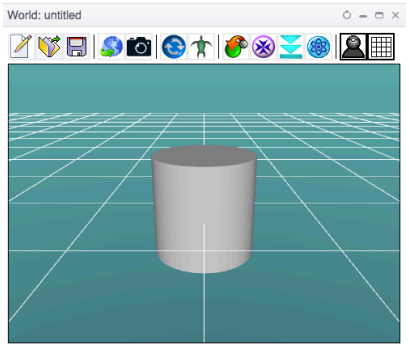
For advanced material settings, you can try the **ME** (Material Editor) tab next to **MC** tab, to further customised materials.

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

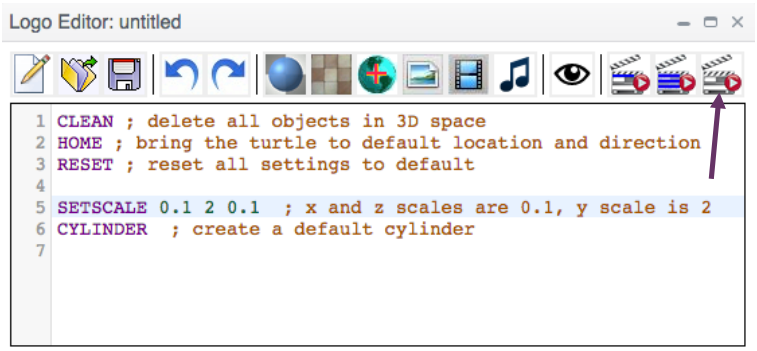
3.4 Scaling objects

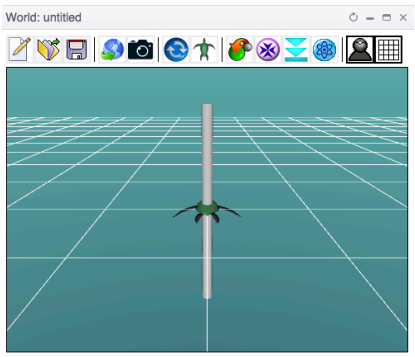
Before inserting an object, you can specify a scale so the object can be scaled to the size you need. In VRMath2's LOGO, we use **SETSCALE** *x y z* to change the scale before creating a 3D or 2D object. For example, a default **CYLINDER** has height and diameter as 1, see below:





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**.

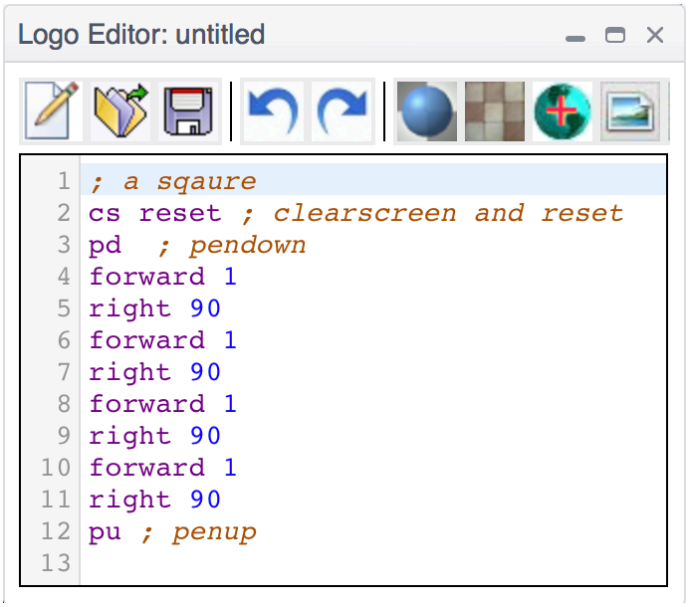
THINK: What if you want a cylinder that is tilted 45 degrees? What if you want the cylinder to have different material (colour)?

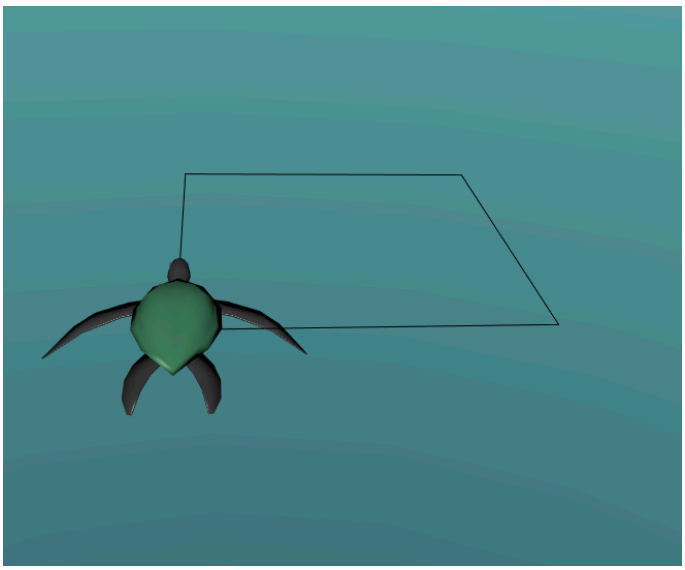
3.5 Loop

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:





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:

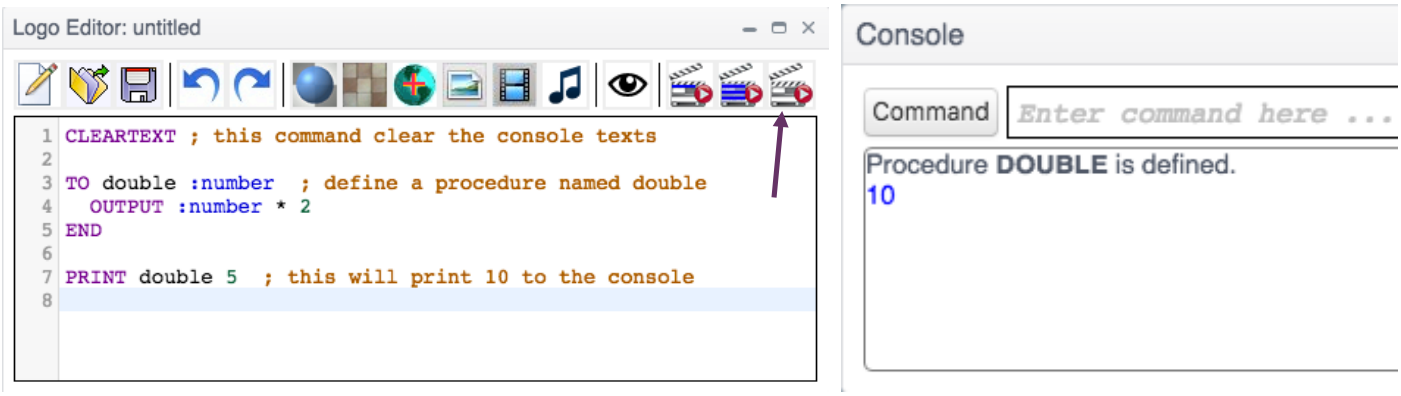
REPEAT 4 [forward 1 right 90] ; note that there is a space between command and input

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,

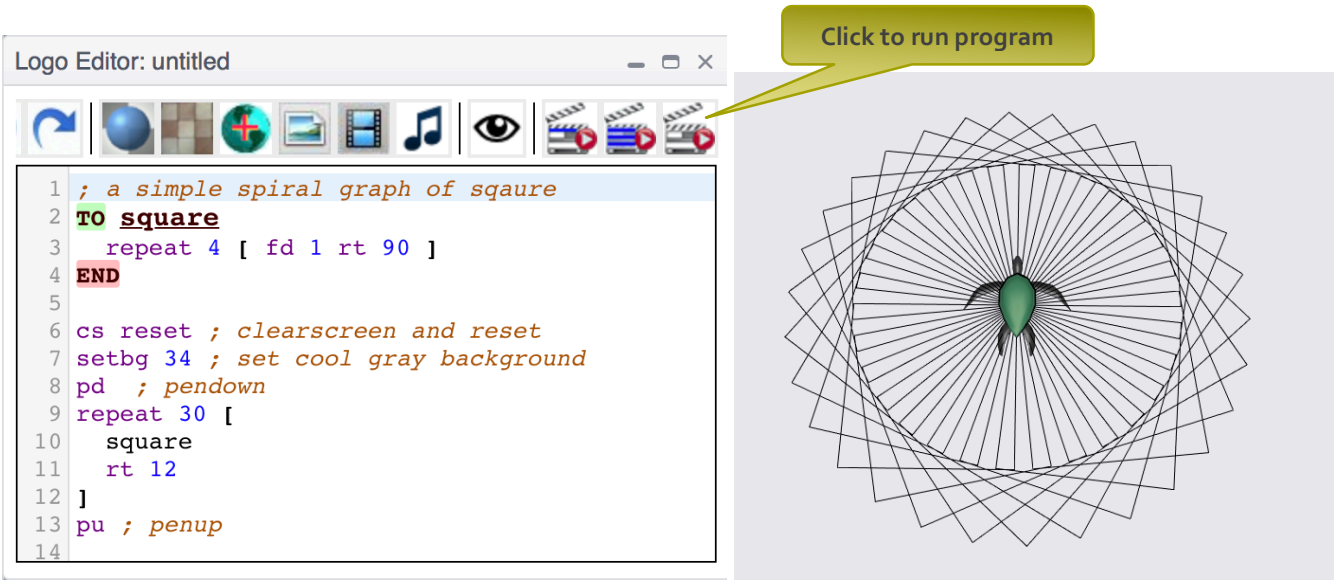
REPEAT *100* [print #] ; This will print number 1 to 100 in the **Console window**.

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?

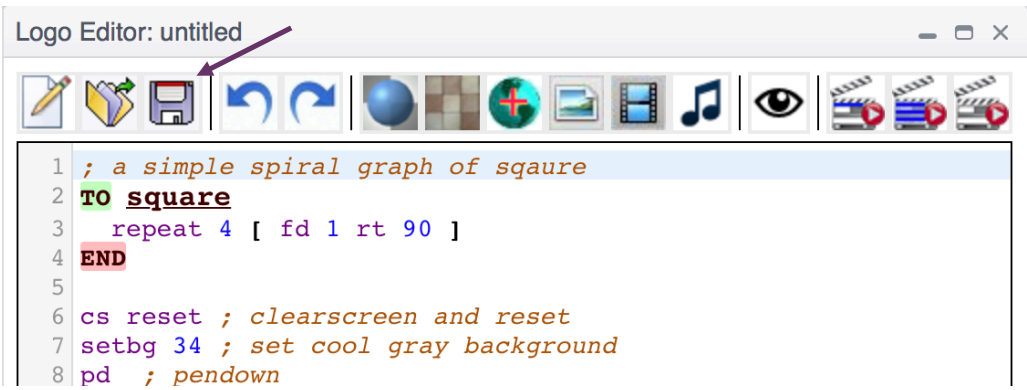
3.7 Save your works

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

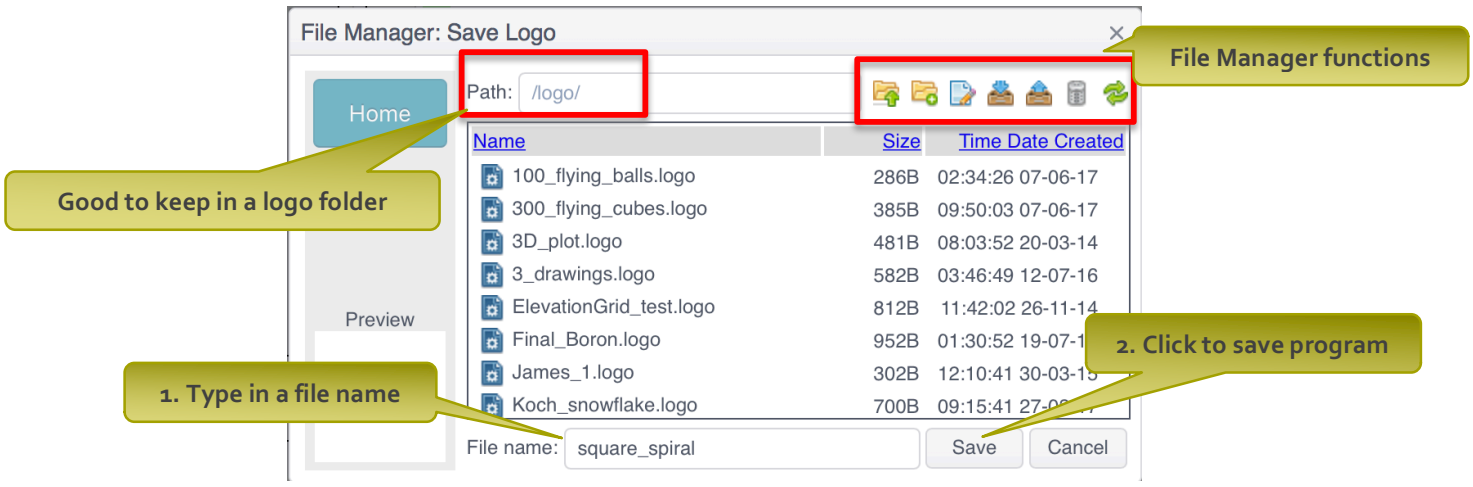
In VRMath2 Editor, you can save 4 types of files:

- .vrm file: the vrm 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 the LOGO program.



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



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). In this workshop series, we will focus on two methods using Logo programming language:

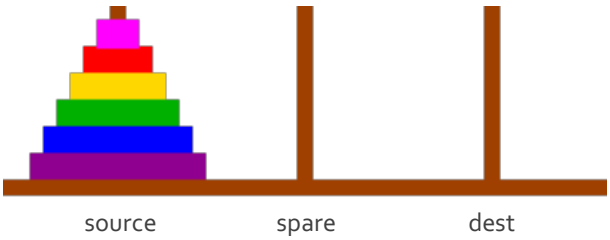
1. Iterated or recursive functions
2. Random or chaotic games

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 expresses 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 Fern leaves

There are different types of ferns, and below are some pictures of typical fern leaves.



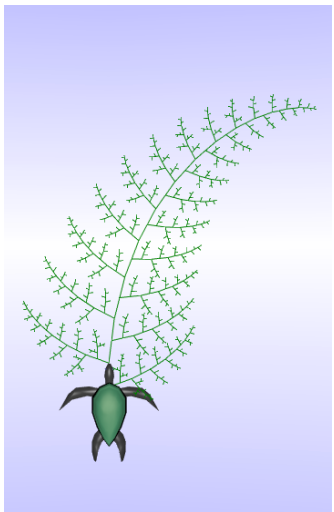
In Logo Editor, open the Logo file from:

Example → fractal → 1_fern_tree → fern.logo

Logo Editor: fern.logo

Click to run program

```
1  ; Recursive fern procedure
2  ;
3  TO fern :size :sign
4    if :size < 1 [ stop ]
5    fd :size
6    rt 70 * :sign fern :size * 0.5 :sign * -1 lt 70 * :sign
7    fd :size
8    lt 70 * :sign fern :size * 0.5 :sign rt 70 * :sign
9    rt 7 * :sign fern :size - 1 :sign lt 7 * :sign
10   jb :size * 2   ; jb = jumpback
11 END
12 cs ; clearscreen
13 cm ; centimeter
14 ncoff ; next color off
15 setpcname "green ; set pen color by name
16 ru 90 ; roll up 90 degrees
17 pd  ; pen down
18 fern 16 1
19 pu  ; pen up
```



Questions: (predict)

1. What is the exit condition of the **fern** procedure?
2. What is the purpose of the variables ***:size*** and ***:sign***?
3. Why rt 7 in line 9?
4. Why jump back ***:size*** * 2 in line 10?

Experiments:

1. Try giving different ***:size*** in line 18, then run the program to see the result. For example, **fern** 10 1
2. Try giving different angles in line 6, 8 and 9, then run the program to see the results.
3. Try changing the decreasing factor 0.5 and 1 in line 6, 8 and 9, then run the program to see the results.

Questions:

1. Answer the previous questions again to see if any different ideas you have after experiments.
2. What is the purpose of line 6 and 8?
3. What is the purpose of line 9?

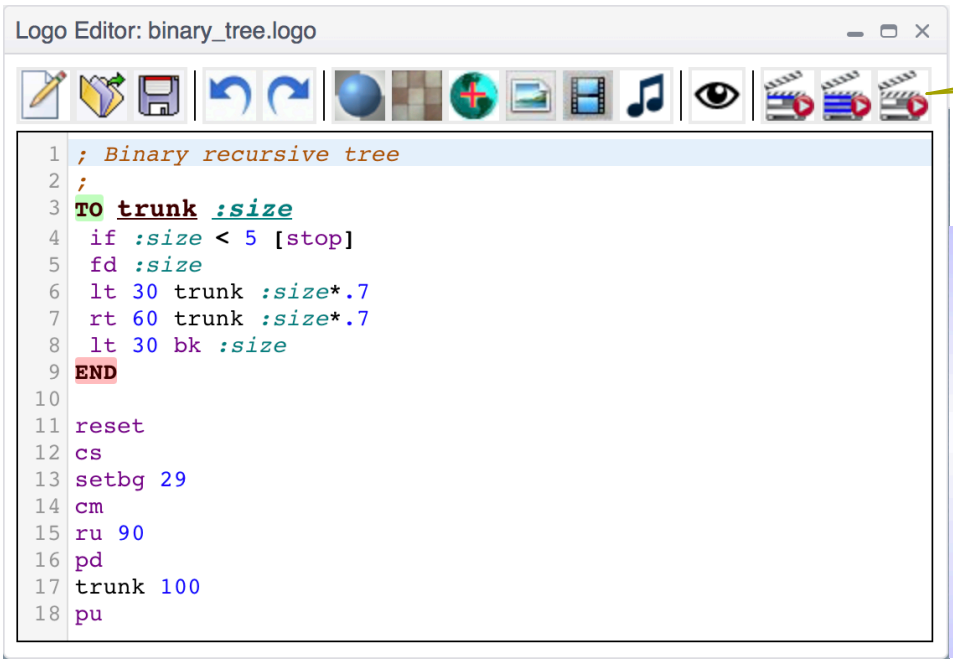
Challenge:

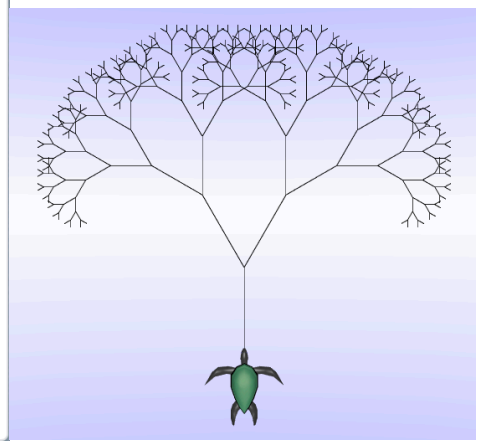
Using the other rotation commands (see page 8), add in some commands (programming codes) to make the fern leave curl in 3D.

(You may open the file from Example → fractal → 1_fern_tree → 3D_fern.logo for an example of 3D fern leave)

4.3 Trees

In Logo Editor, open the Logo file from: Example → fractal → 1_fern_tree → binary_tree.logo





Questions: (predict)

1. What is the exit condition of the **trunk** procedure?
2. What is the purpose of the variable ***:size***?
3. Why the angles turned in line 6, 7 and 8 are 30, 60, and 30?

Experiments:

1. Try giving different ***:size*** in line 17, then run the program to see the result. For example, **trunk** 50.
2. Try giving different angles in line 6, 8 and 9, then run the program to see the results.
3. Try changing the decreasing factor 0.7 in line 6 and 7, then run the program to see the results.

Questions:

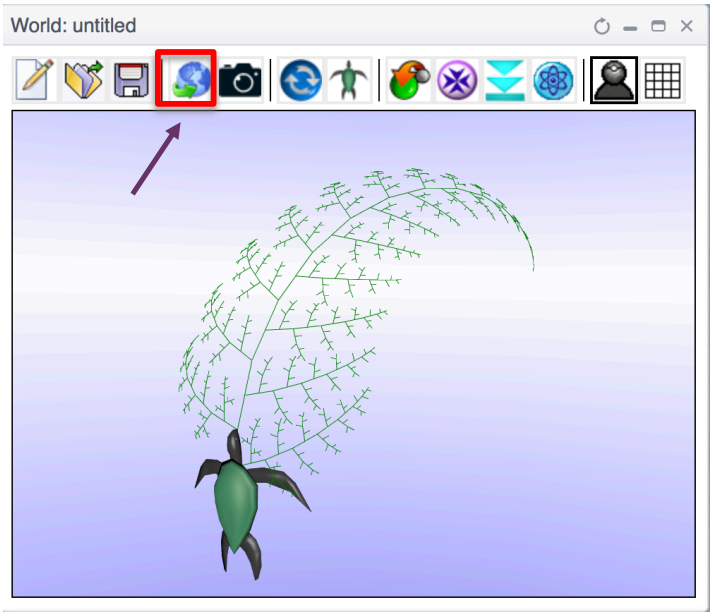
1. Answer the previous questions again to see if any different ideas you have after experiments.
2. What is the purpose of line 6 and 7?
3. What is the purpose of line 8?

Challenge:

Using the other rotation commands (see page 8), add in some commands (programming codes) to make the tree has more branches in 3D.

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.

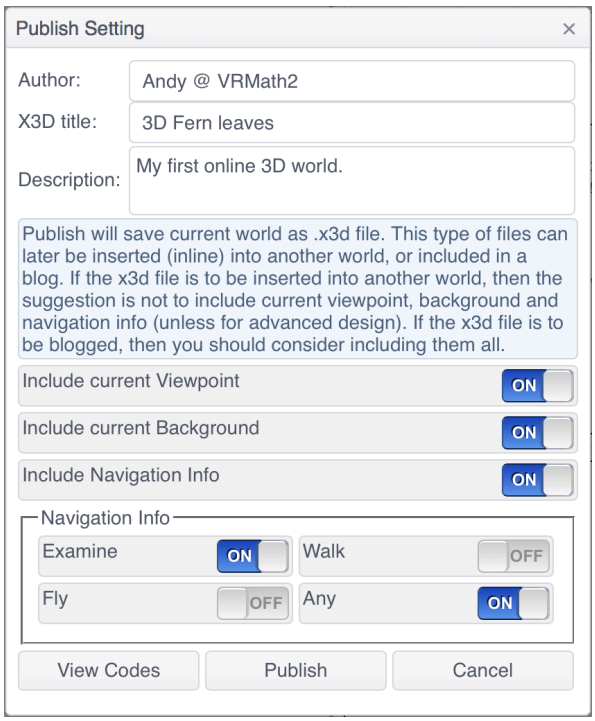
Blogging in VRMath2 is like writing a personal diary about your learning journey. Through blogging, you can practice your writing and communication skills. And when you write about your programming and virtual worlds, you will be learning and applying science, technology, engineering and mathematics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

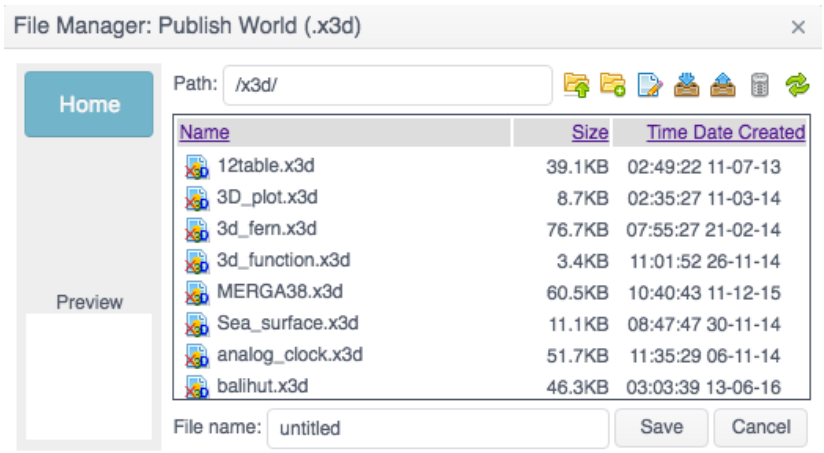
5.1 Publishing 3D model

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.

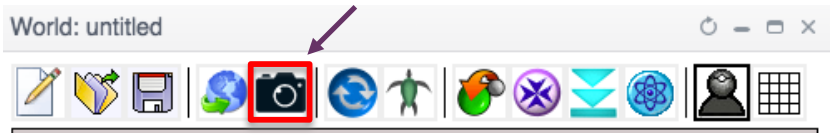


By default, you will save in the **x3d folder**. Provide a sensible file name then click on **Save** button to finish.

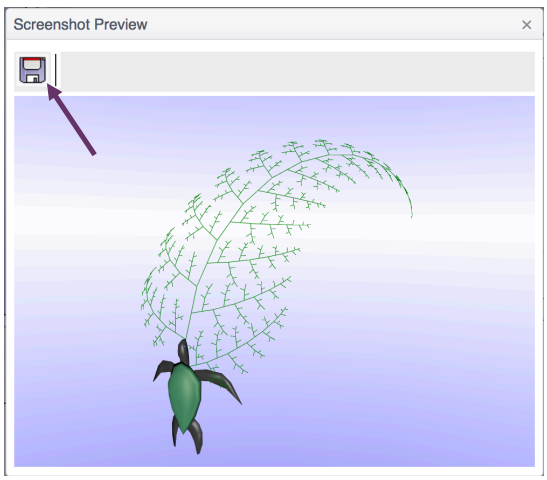
5.2 Save an image for blog

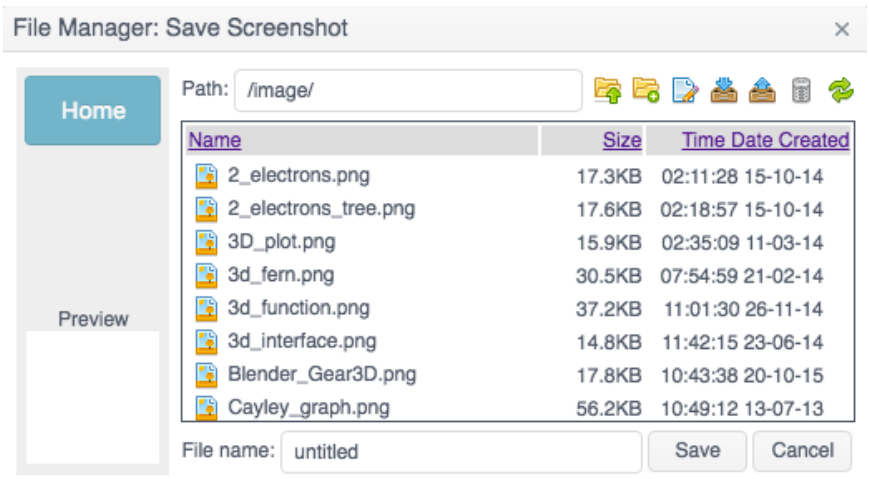
An image of your 3D model can be uploaded when blogging, as a thumb nail for your blog. Readers can then see a picture of your 3D model before they click to read your full blog. This is a key aspect of blog in VRMath2 Community.

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,

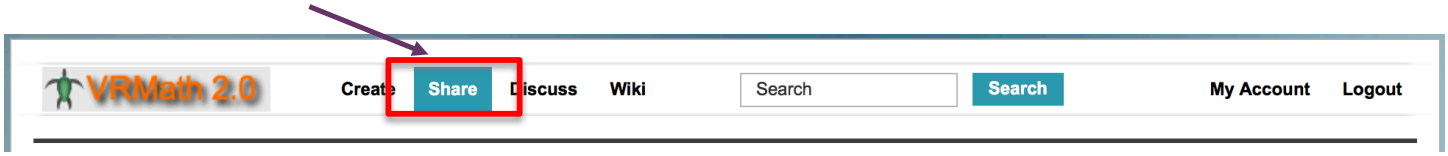




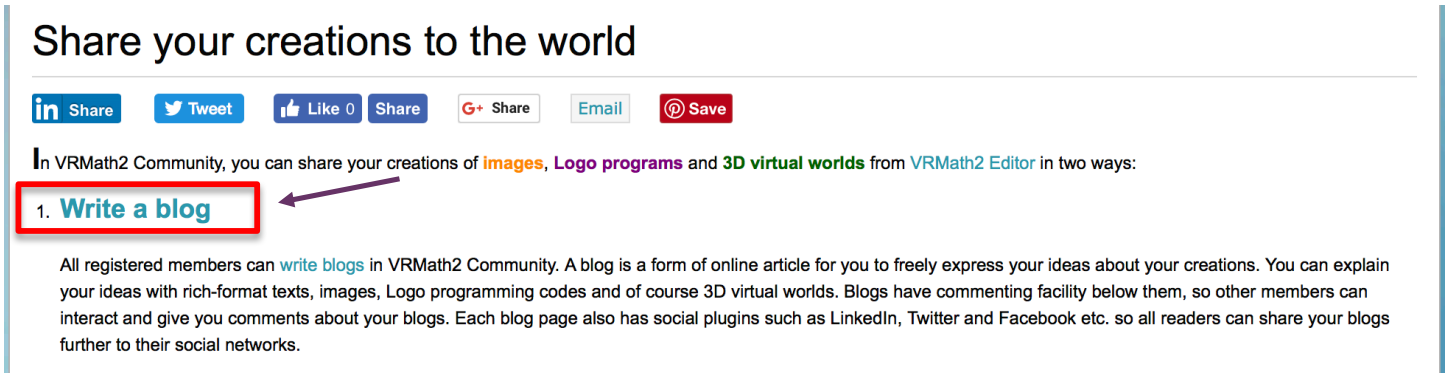
The default folder for saving screenshots is the **image folder**. Provide a sensible file name then click on **Save** button to complete.

5.3 Write a blog

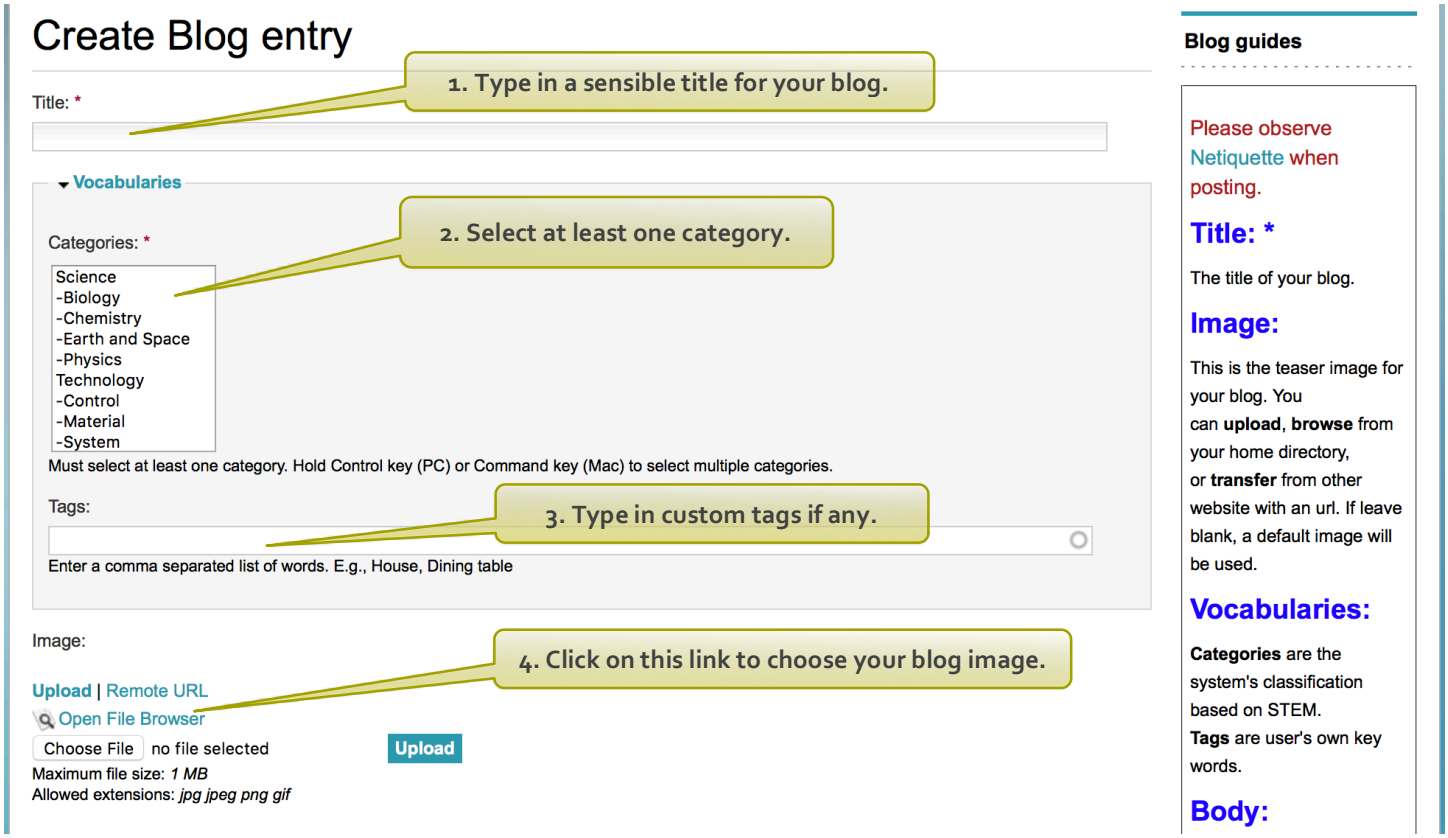
To write your blog about your 3D virtual world, click on the **Share** in the main menu of home page.



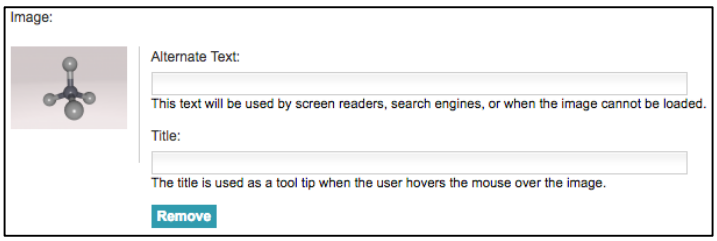
Then, in the share page, click on **Write a blog**.



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.

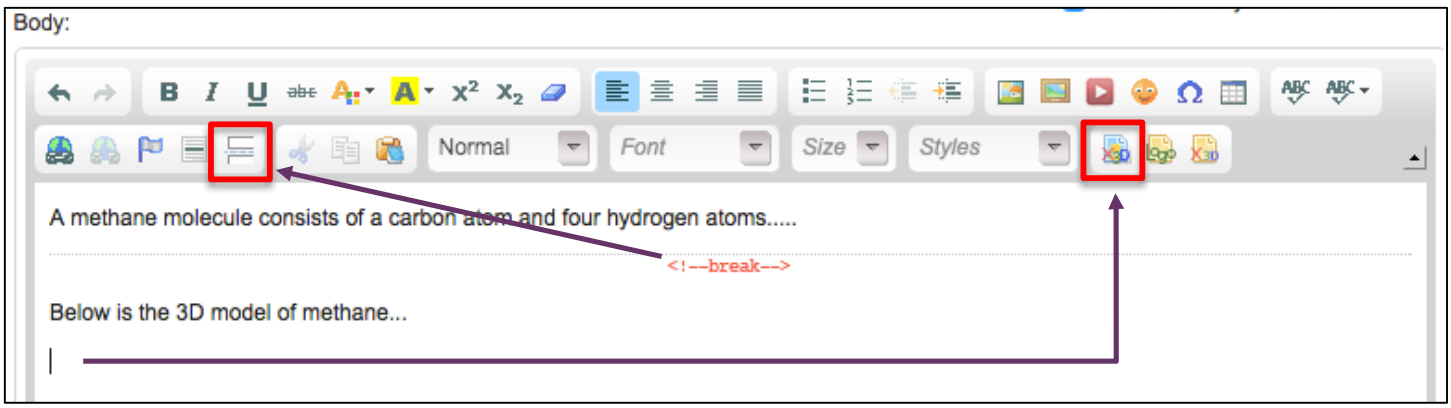


The **File Browser** will open in a new tab. You need to click into your **image folder** to find the image you saved in previous section (6.2). Select your image file then click on **Insert file** to close the dialog.

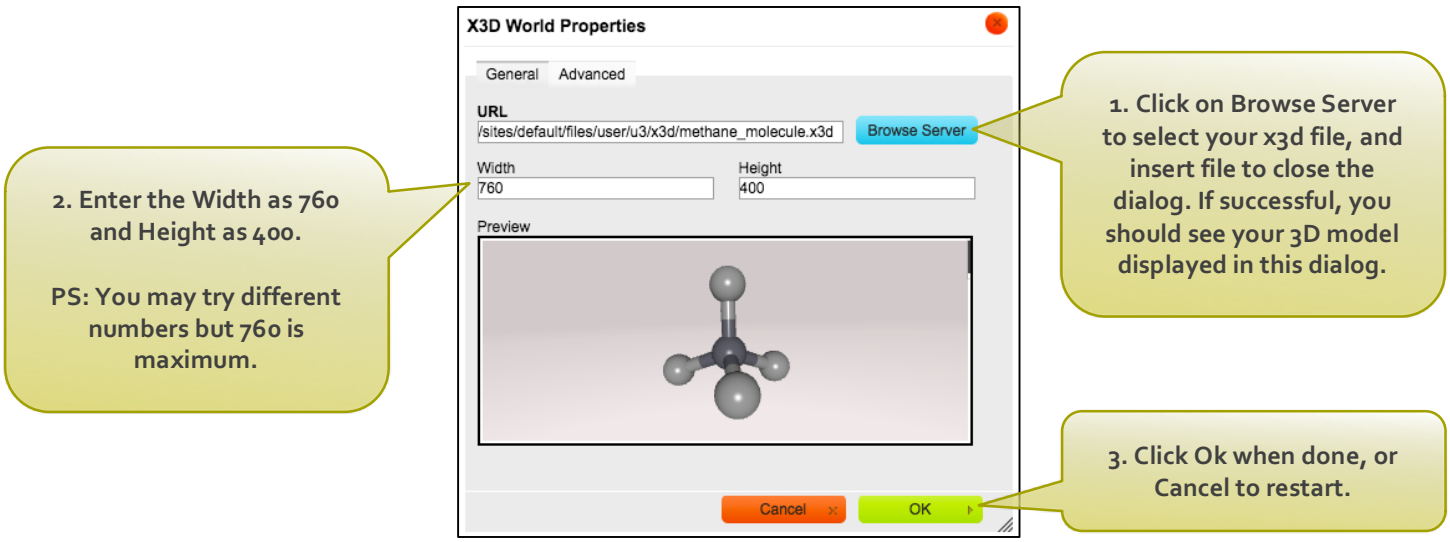


If successful, you should see your image displayed and you are asked to enter **Alternate Text** and **Title** for the image. Please do provide a text and title for the image.

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.



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



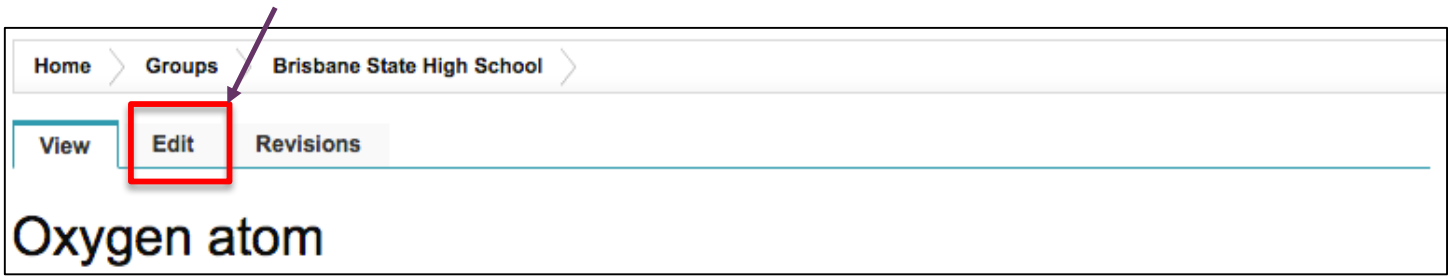
After the 3D model, you could write some paragraphs to explain about your 3D virtual world and programming. At the end of the blog, insert links to your 3D model and Logo program files.

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

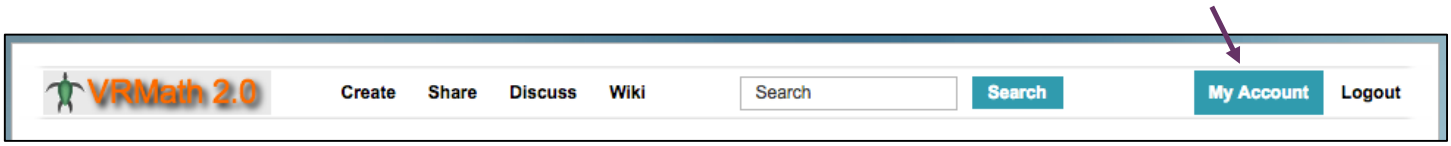
You can edit later your blog as many times as you wish. See next section for how to **Edit**.

5.4 Edit own blog

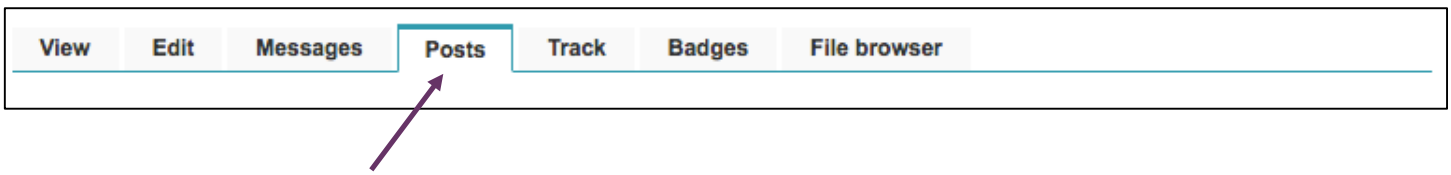
You can edit all your posts when you are viewing them. Simply click on the Edit tab to edit. For example,



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.

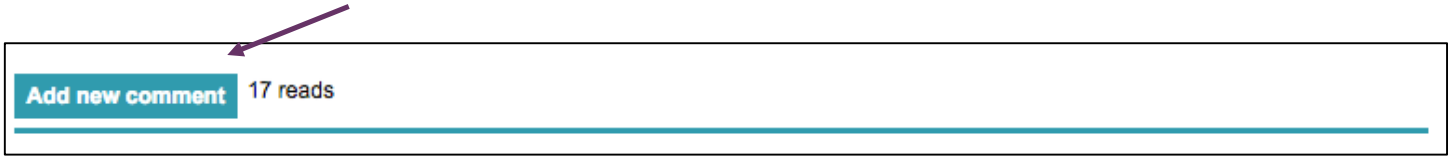


5.5 Commenting blogs

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.

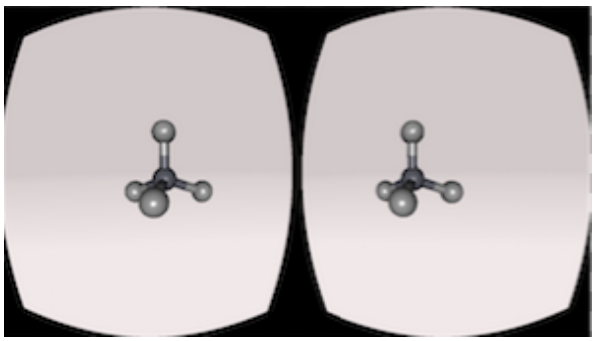


You can also reply to comments by clicking on the **Reply** button in the comments.

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 assessable 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.

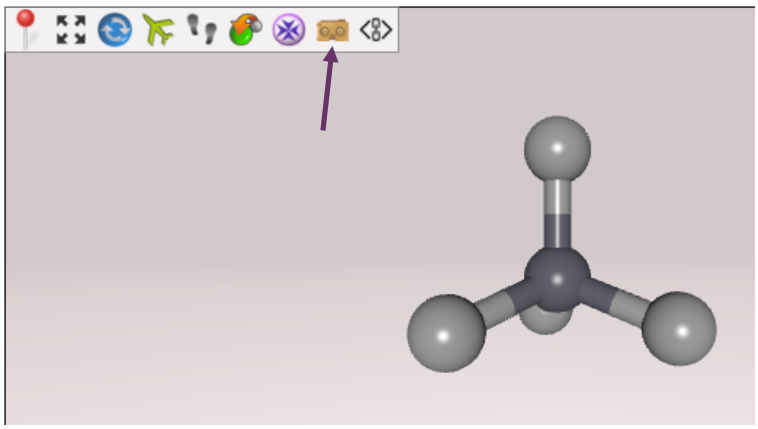
VRMath2 VRBox is an application to view 3D models in VR CardBoard. It has a web address for mobile device only at https://vrmath2.net/vrbox

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

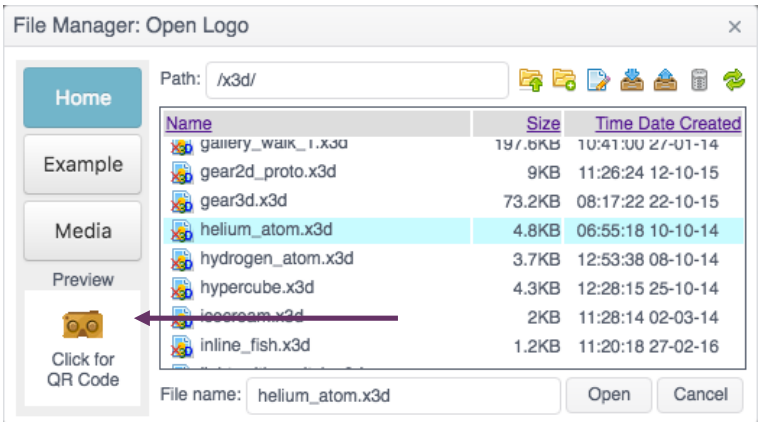
When viewing a blog with a 3D model in it, click on the square to get a menu as below:



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.

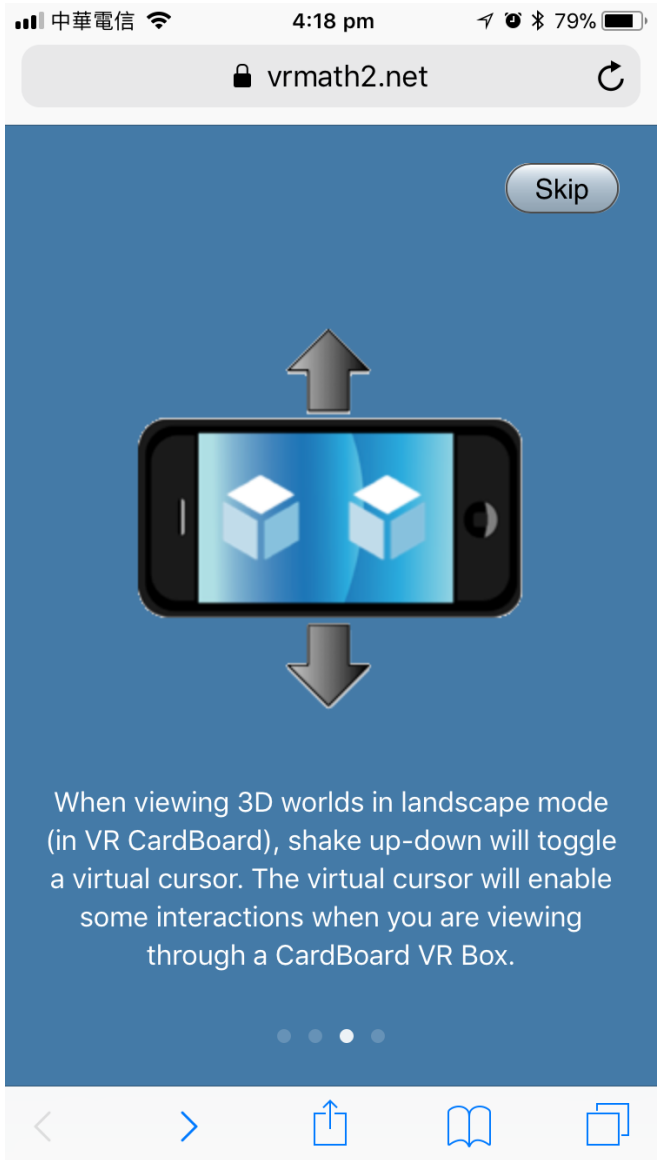


Once a x3d file is selected, click on the CardBoard icon in the Preview area. A QR code will be displayed. Scan the QR code with a mobile phone to open the VRMath2 VRBox application in a mobile browser.

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

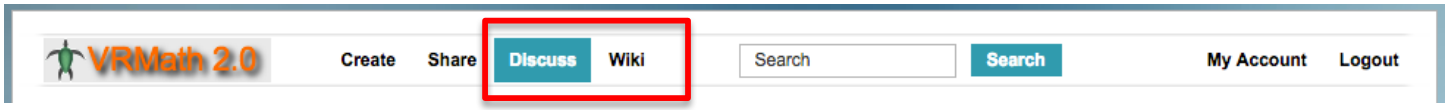
6.2 VRMath2 VRBox

The VRMath2 VRBox App in 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.

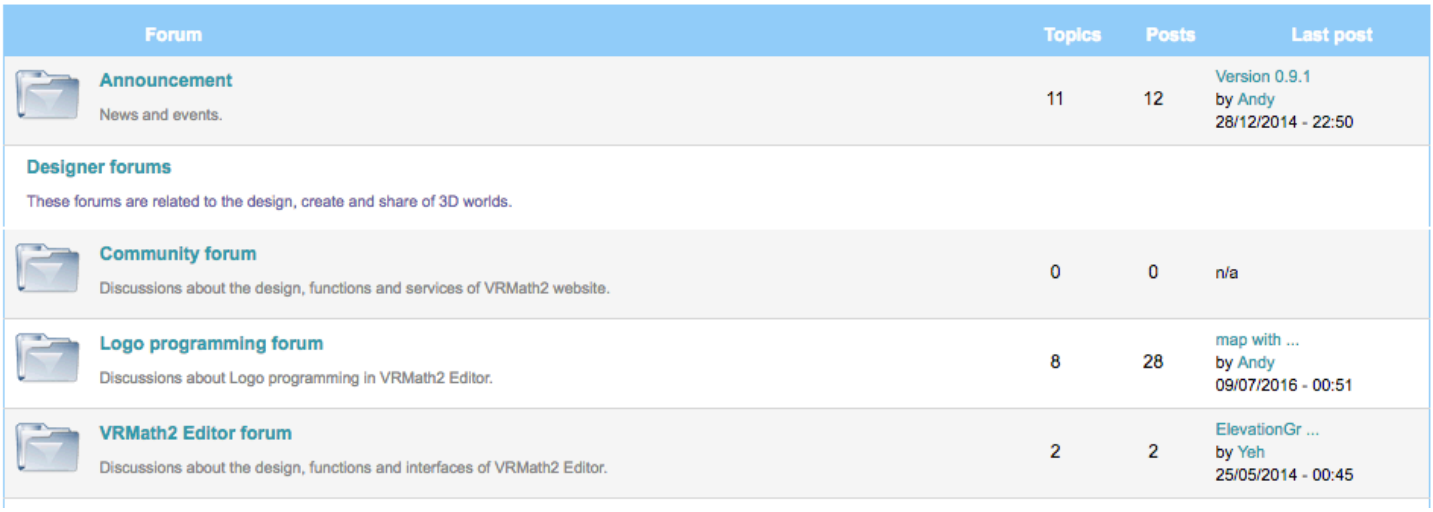


7. Forums and Wiki for help

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.



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.