
MakersCAD Documentation

Documentation

Release 0.5.2 Alpha

Luis Arturo Pacheco

Feb 13, 2018

Contents

1	Welcome to MakersCAD's documentation!	1
1.1	Getting started	1
1.2	CAD	10
1.3	Logic	21
1.4	Loop	22
1.5	Math	27
1.6	Text	30
1.7	Lists	30
1.8	Map	30
1.9	Color	30
1.10	AdvancedJS	30
1.11	Variables	30
1.12	Function	30
1.13	Examples	30
1.14	Tutorials	30
2	Indices and tables	31

Welcome to MakersCAD's documentation!

Contents:

1.1 Getting started

1.1.1 Downloading and installing MakersCAD

MakerSCAD can be installed for Windows, Linux and MacOS, in order to download your copy go to [MakerSCAD Website](#)

Warning: Remember this is an early Alpha testing version expect the following things to happen:

- Things might break.
- Current version might lose compatibility with future versions.
- MakersCAD and its documentation are under heavy development things will change!

If you find any bugs and want to help us out send us an email at:

`hello@MakersCAD.com`

The site has the option to try MakerSCAD online, but we strongly recommend to install the software on your computer, to do that click on “I WANT THIS!”

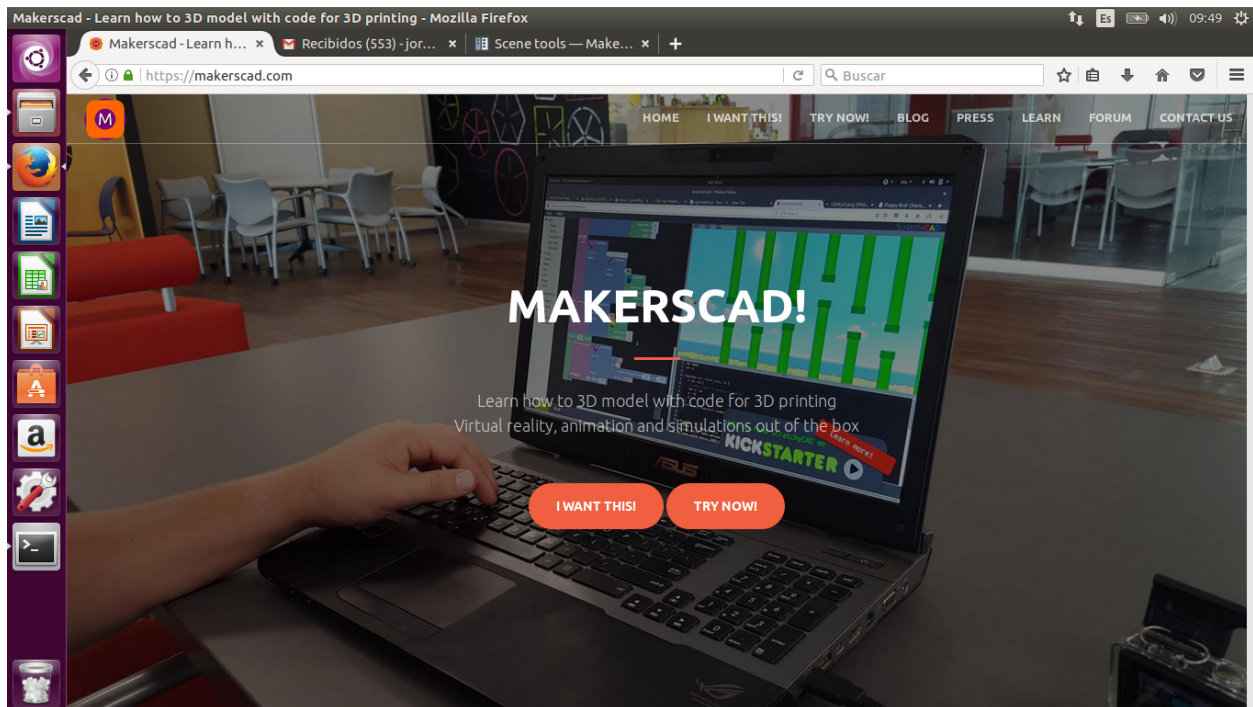
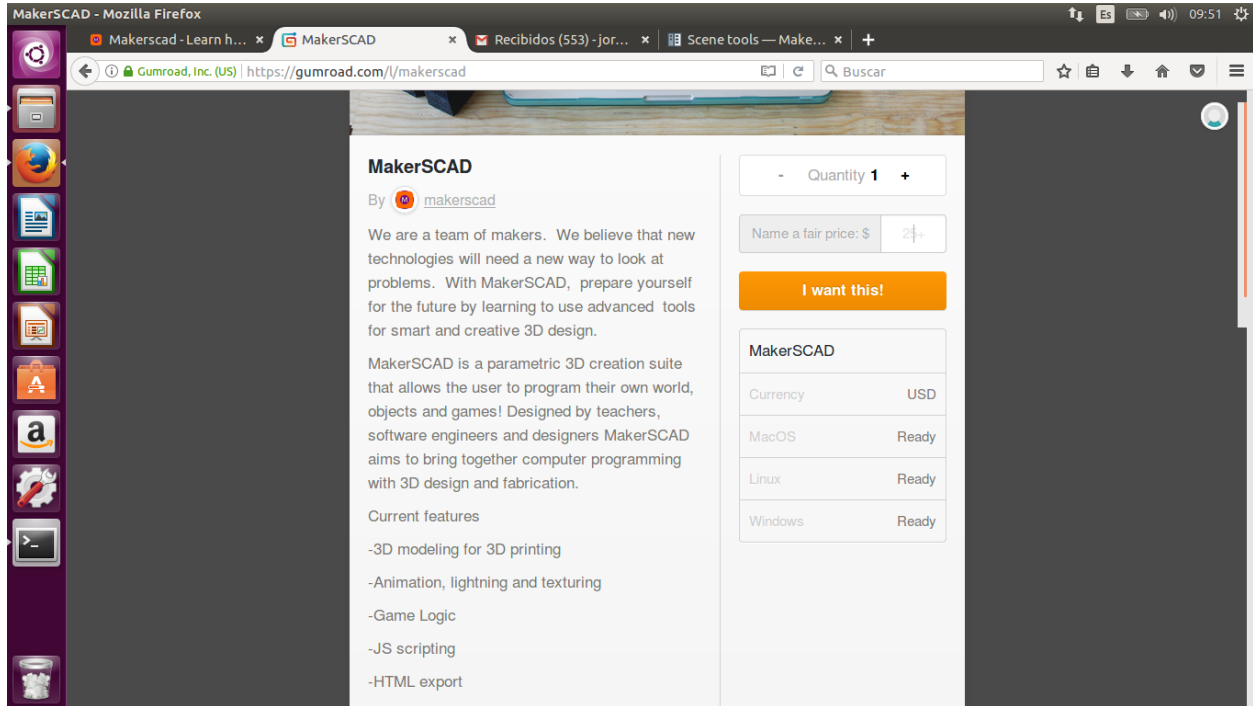
Enter your email address, name, and company in the form as shown in the picture.

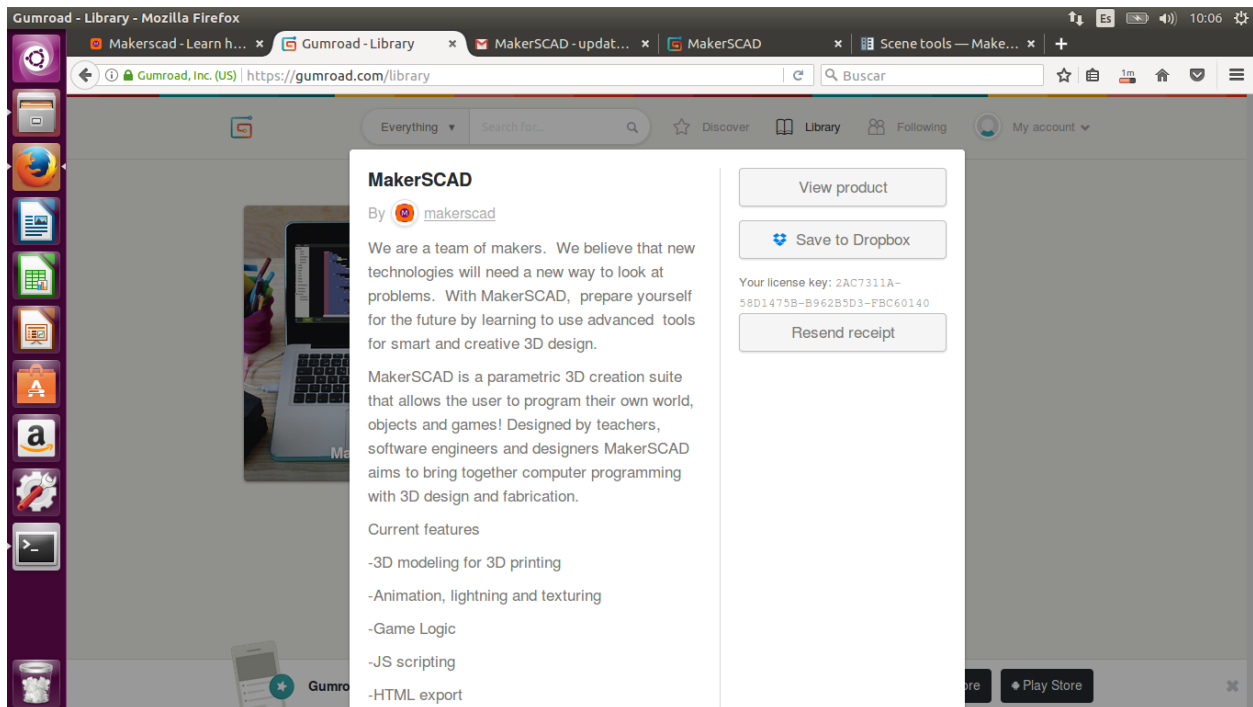
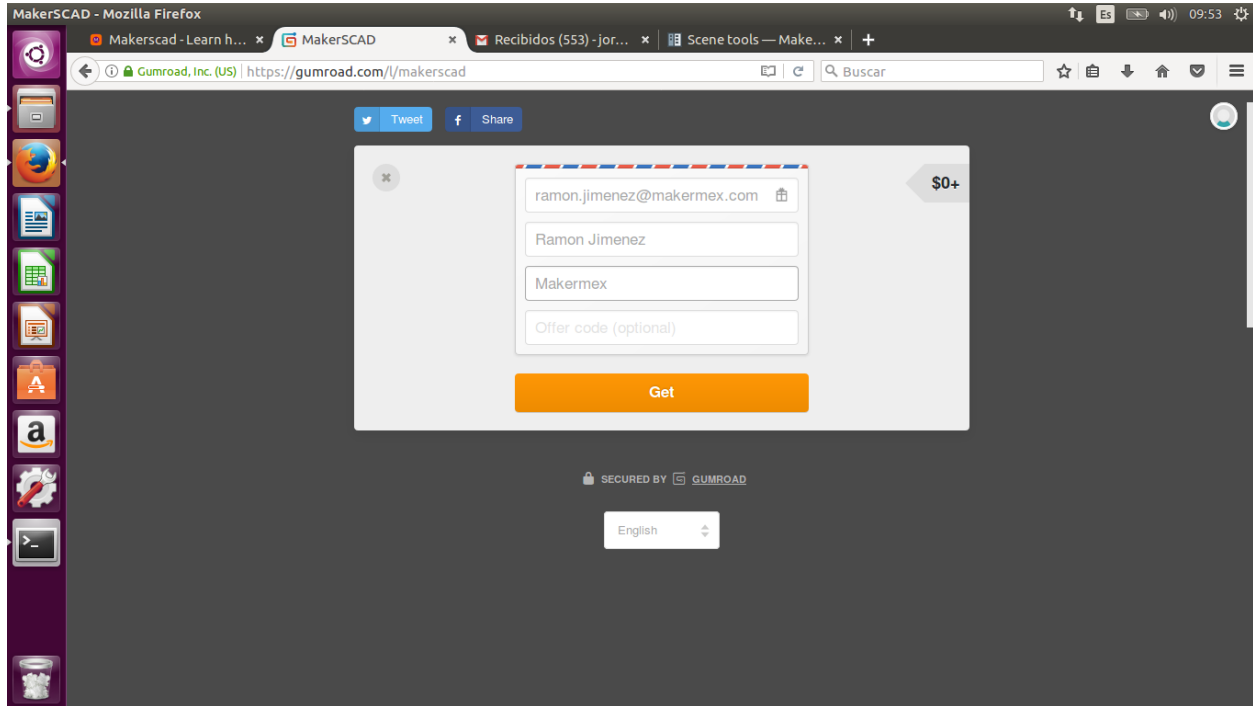
Pay what you want for MakerScad, that means that you decide how much the product is worth, so, enter the quantity you consider as a fair price and click on the button to proceed.

Go to view product and download the latest version for your operating system

in windows simply click on the installer and it will install the software

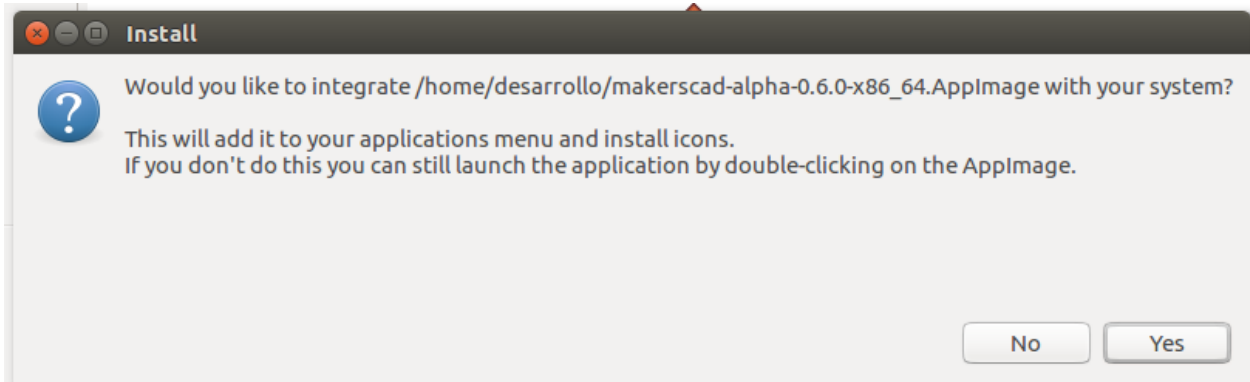
in linux open a terminal, go to the folder where the installer is located and type the following:





```
~$ chmod a+x makerscad-alpha-0.6.0-x86_64.AppImage
```

Then you will see this message, accept to make the file executable



1.1.2 Opening MakersCAD

First go ahead and visit [MakersCAD](#) and get your copy of MakerSCAD.

1.1.3 The MakersCAD interface

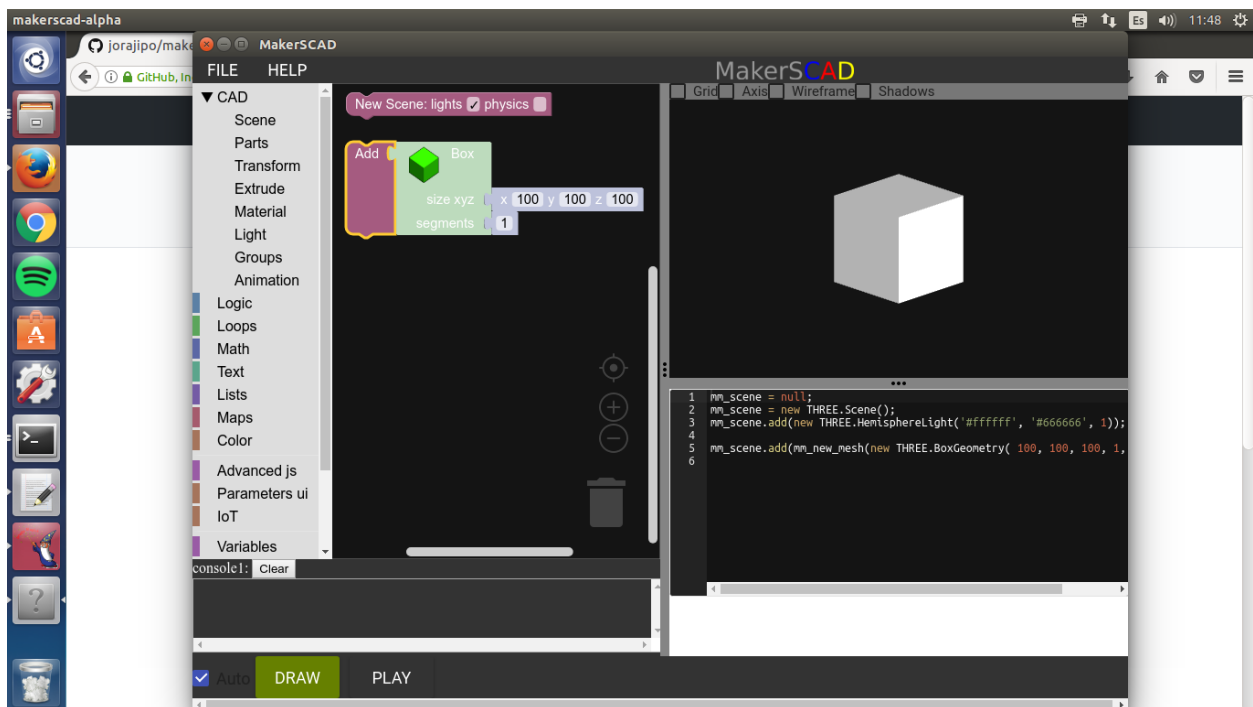


Fig. 1.1: The MakersCAD UI

First, we have the main toolbar

Here you can:

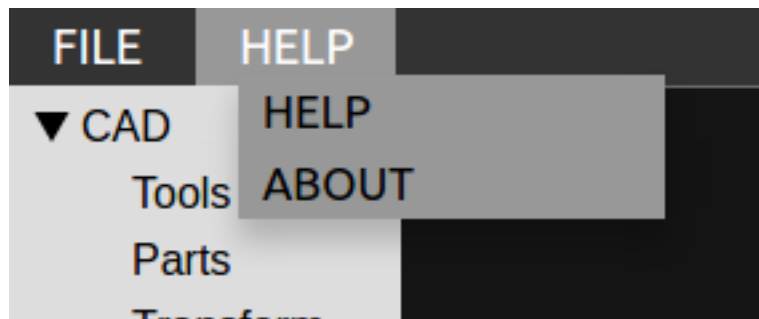
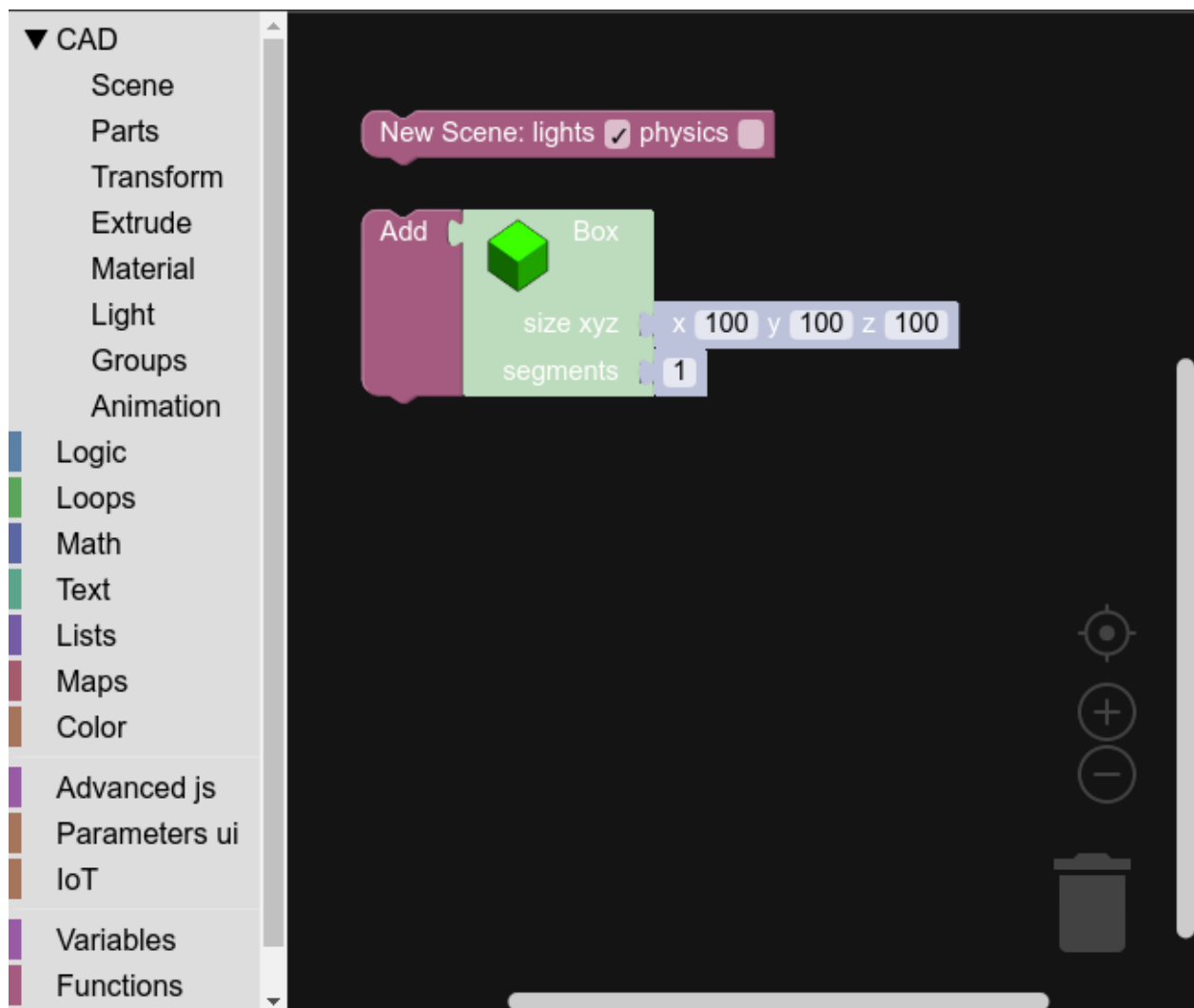


Fig. 1.2: the main toolbar

- SAVE your current work, it overwrites the latest save state.
- LOAD the latest saved version
- EXPORT your work into an xml file, for now the is test.xml increasing the file number, for example test1.xml,test2.xml... finding the next available number.
- EXPORT the current 3d objects as a STL file.
- EXPORT the current 3d objects as an OBJ file.
- LOAD an asset (experimental).
- Configure MakersCAD on the SETTINGS window.
- Go to the Documentation to get HELP learning MakersCAD
- Check the ABOUT window for authors, copyright licence and more.

Under the main toolbar at the left side is Work view



Here we have the block library.

- [CAD](#), Special blocks related to shapes, objects and parts.

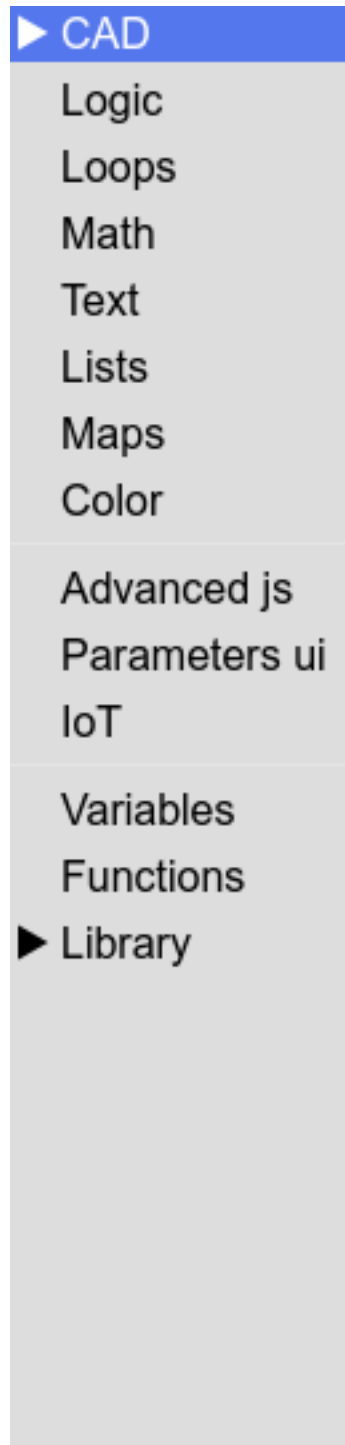


Fig. 1.3: The block library

- Logic: blocks related to logical operations, like comparison operators, conditional and boolean values (true/false).
- Loops: blocks for mathematical functions.
- Math: blocks for mathematical functions.
- Text: blocks for output, formatting and search operations with text strings.
- Lists: This section contain tools that allows the creation of object groups and search specific elements in them.
- Maps: Blocks to create data structures for keys and its values.
- Color: Blocks for coloring objects.
- Advanced js: These blocks can be used to write your own javascript code or import js files.
- Parameters ui: Blocks to enhance the GUI.
- IoT: Blocks to include connection to external webservices, invoke them, as well as send and receive parameters and messages with them in order to design IoT gadgets.
- Variables: Blocks to create programming variables and assign values to them.
- Functions: Blocks to declare and invoke functions.
- Library: Examples to learn from, and experiment with.

and the block workspace where we will be creating our 3D world.

Some useful controls for our block workspace:

- Pan the view up, down, right and left:

```
Right click + Drag
```

- Zoom in and Out:

```
Mouse wheel up or down
```

On the Left side at the the top theres our 3D view, where we can see a preview of our world.

Here you are able to:

- look around your current object in 3D:

```
Right click + Drag
```

- Pan the camera up, down, right and left:

```
Shift + Right click + Drag
```

or

```
MMB + Drag
```

- Zoom in and Out:

```
Mouse wheel up or down
```

or

```
Ctrl + Right click + Drag
```

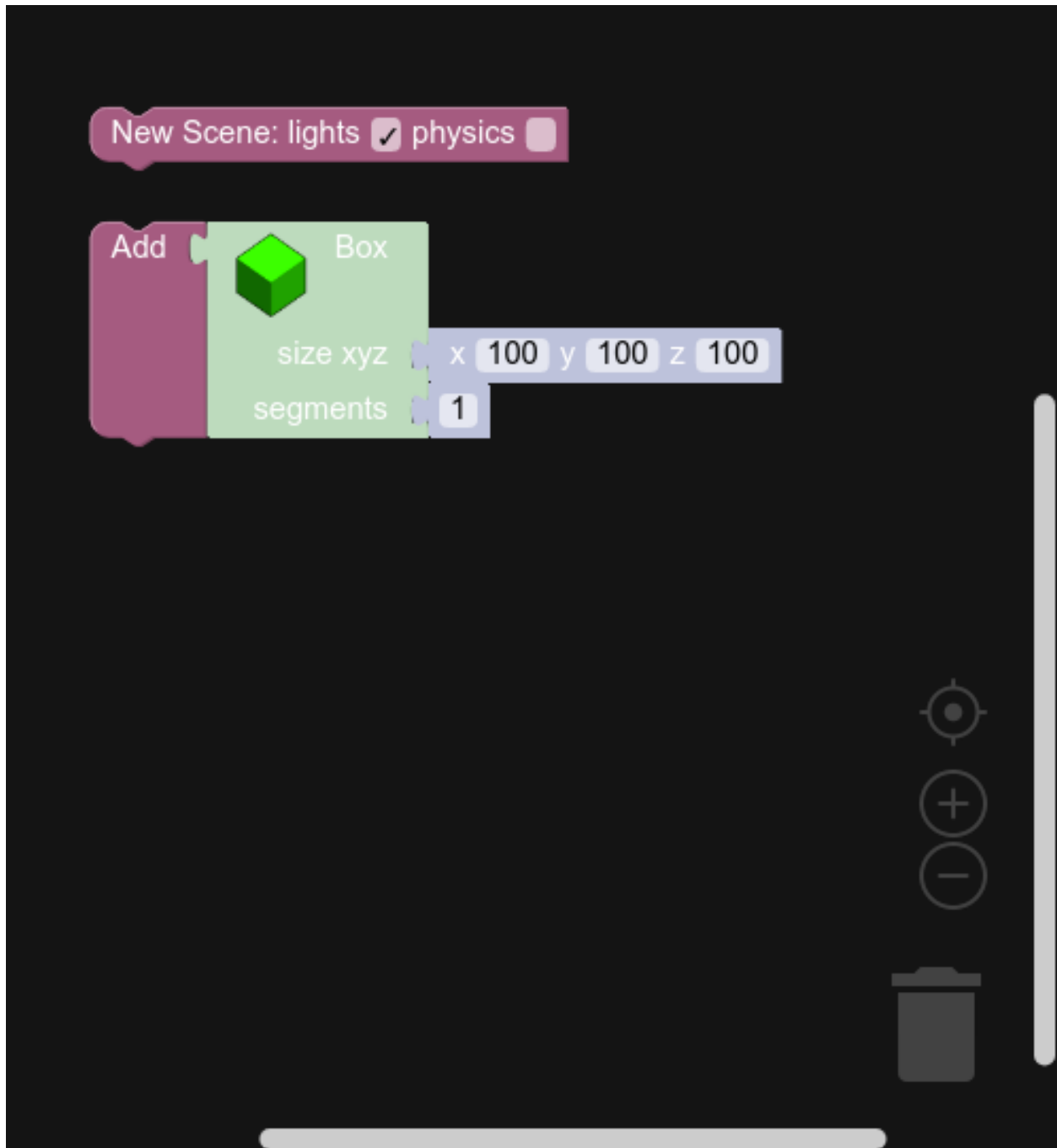


Fig. 1.4: The block workspace

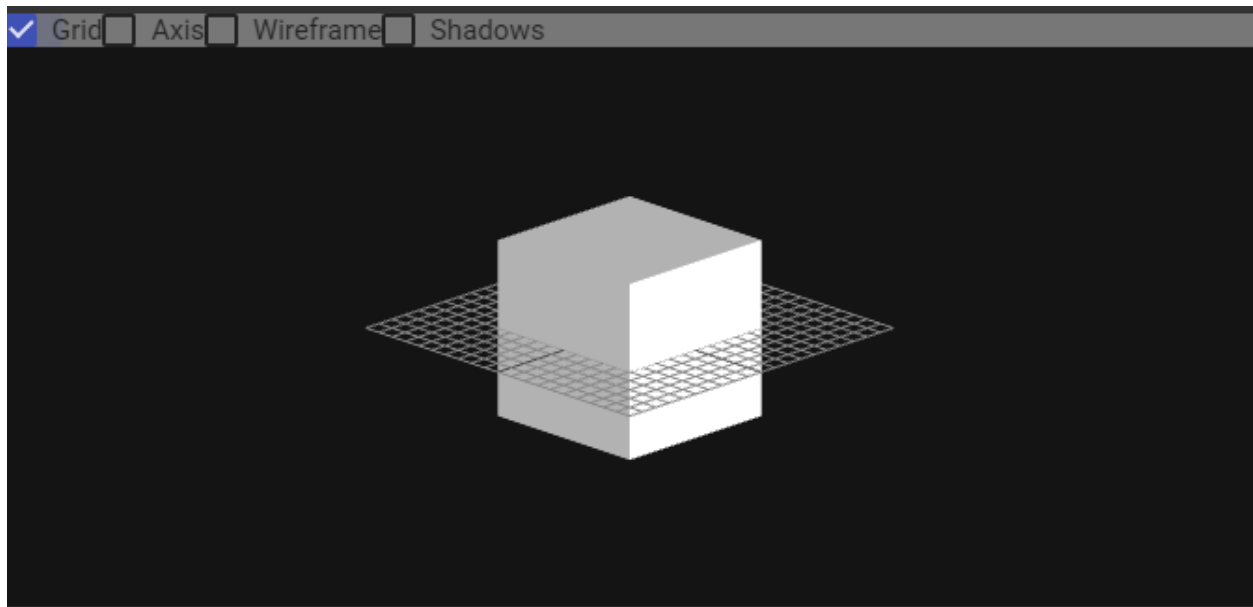


Fig. 1.5: The 3D view.

Under the 3D view there's a javascript viewer that allows us to view and debug the code created from our blocks to display our 3D world.

Finally at the bottom is our Draw Toolbar, here we can turn on and off the automatic update to see in realtime the changes we are making on our blocks, manually update and DRAW the 3D View or PLAY our world in a new window.

1.2 CAD

MakersCAD provides with CAD (Computer-Aided-Design) blocks.

They will help you create your 3d world and objects.

1.2.1 Scene tools

The basic scene

In order to start creating a 3D world we first need to set up a scene, in MakerSCAD we have preloaded a basic scene that comes with some basic lightning, material and shadow presets. The basic scene can be found in the block library under the CAD/Tools section.

It looks like this:

Go ahead and add a new basic scene to our workspace, right now it is an empty scene that we need to start adding objects to it in order to build our world.

Note: the new basic physic scene is similar to the basic scene but contains gravity, only use it when needed for game or animation logic.

Next we have the add block, this allows us to add an object to our scene objects include shapes, geometries, lights and more go ahead and take the scene.add block and plug it under the basic scene block.

```
1 mm_scene = null;  
2 mm_scene = new THREE.Scene();  
3 mm_scene.add(new THREE.HemisphereLight('ffffff', '#666666', 1));  
4 mm_scene.add(mm_new_mesh(new THREE.BoxGeometry( 100, 100, 100, 1, 1, 1 )));  
5
```

Fig. 1.6: The Javascript code viewer



Fig. 1.7: The draw toolbar

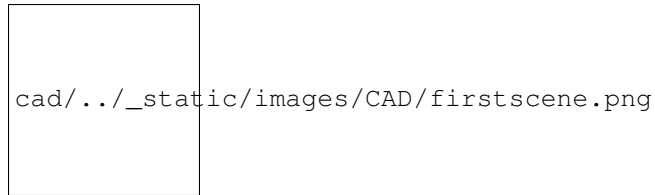


Fig. 1.8: the basic scene block.



Fig. 1.9: adding objects to our scene.

By default the add block comes preloaded with a cube, go ahead and press the DRAW button! here have succesfully added our first object to our scene.



Now that we know how to add objects to our world lets look at some of them:

Note: the add scene block in photorealistic mode, needs lights to work.

Object loader

Use an Object loader to load an obj or stl resource to the workspace, to add an object loader block go to CAD, then Parts, find the object loader block and click it.



Once you have an object loader in your workspace, click on the gear icon to open a file selection window, if you want to load a file from your PC click the “load” button” and browse your PC for an OBJ or STL file. Also you can enter or paste an URL in the text box and then click the “load url” button to load from a web location.



you will see a preview of the file, then you can enter or click the “close” button, to close the window and go back to your workspace.

Now you can add the loaded object to the scene as shown in the image

1.2.2 Shapes

The Box

The box geometry is found under the CAD/parts library Go on and grab the cube block and plug it to the scene.add block.

this is the MakerSCAD default box that measures 100x100x100 units.

The first parameter in the cube is asking for a size XYZ, for every parameter in ScratchyCAD that asks for XYZ we need to plug in a special block that contains a list of three numbers one for lenght (x), one for width (y) and one for

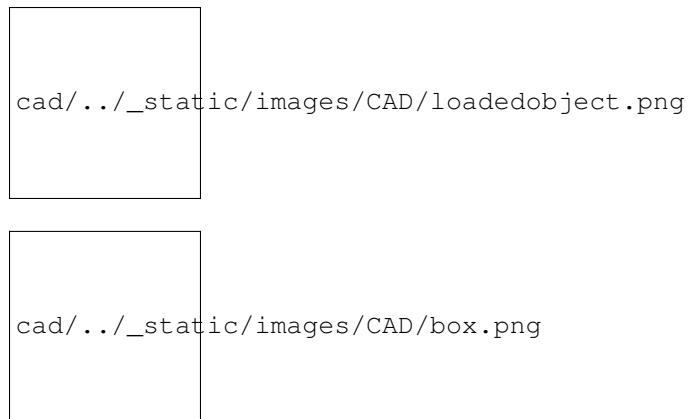


Fig. 1.10: ScratchyCAD's default box

height (z). For now grab lets use the preset and change the values on the x,y,z inputs. Activate the Auto checkbox on the left right corner and see how each value affects the box.

Under the Lists find the Vertical XYZ block this block will allow us to change the values to formulas or even variables.



Now we have a list we only need to get some values. Under the math library look for the first block that contains a value of 0, this is a number block, it returns the value that the user types on it, or the variable that's connected to it. Go ahead and grab three number blocks for each of the xyz inputs. Try different numbers, formulas or variables out, remember to turn on the AUTO in order to see the changes as you type.

The sphere

The sphere block can be found under the CAD/Parts library.

The sphere has two parameters:

- Diameter, a numeric value and defines the size of the sphere
- Resolution, a numeric value that defines the number of poligons that make the shape

The cylinder

The cylinder block can be found under the CAD/Parts library.

This block has four parameters:

- Diameter1, a numeric value that defines the size of the lower base.
- Diameter2, a numeric value that defines the size of the upper base.



Fig. 1.11: the default sphere block.



Fig. 1.12: the default cylinder block.

- Height, a numeric value that defines how tall is the cylinder.
- Sides, a numeric value that defines how many sides our prism will have.

The torus

A torus is a Donut shaped mesh, defined as a revolving circle in an axis coplanar with the circle.

The torus is found in the CAD/Parts library.



Fig. 1.13: the default torus block.

The torus has five parameters:

- diameter, a numeric value that defines the torus radius.
- tube diameter, a numeric value that defines how thick is the ring.
- radial segments, a numeric value that defines how many “sides” the section of the torus will have.
- tubular segments, a numeric value that defines the how many “sides” the torus itself will have.
- arc angle, a numeric value that defines if we only what a partial part of the torus.

The polygon

The polygon block defines a 2D shape, this shape is given by a list of at least three points or vertices. This given list of vertices creates a face in the x,y plane.

Go ahead and get the polygon block from the CAD/Parts library and plug it to your main function, by default it returns a triangle rectangle.



Fig. 1.14: the polygon block with default vertex list.

To make things easier we have included the OpenSCAD-polygon-editor that allows you to draw from scratch or trace a shape from an image, click on the gear in the top left corner of the polygon editor.



Fig. 1.15: the polygon editor interface.

From the selector menu (cursor icon) you can turn any point to a Bezier curve handle, or vertex.



Fig. 1.16: the polygon selector menu.

We can add new vertices by clicking on the + icon next to the selector menu.

We delete vertices by clicking on the - icon next to the + icon.

Finally we can add an image to trace, by setting the URL, uploading it and then scaling it..

Once we are done setting up the points for our shape click close and they will be automatically added to the polygon block.

Note: Polygons require to be extruded to be displayed on the 3D view

In order to see the polygon we just defined its important to add an extrusion block or else it wont be displayed.

The text3D

The 3Dtext block can be found under the CAD/Parts library.

It allows us to create a text based model, this block has the following main parameters:

- text : a string value with the characters we are going to draw.
- size: a numeric value that determines the thickness of the characters.

- height: a numeric value that determines the extrusion amount of the characters.



Fig. 1.17: the text3D block.

The part block

It's nice to be able to return different basic shapes, but most objects in real world are a mix of different shapes that create a more complex shape. That's what the Part block is for! This one is found under the CAD/tools library. It allows us to make some CSG (Constructive Solid Geometry) modeling operations, that work great for 3D printing.



Fig. 1.18: using the part block

The part block has three options:

- Union, adds two parts together.



Fig. 1.19: union of a cube with a sphere

- Difference, subtracts the second part from the first part, it's order sensitive.
- Intersection, returns only the overlap of both shapes.

1.2.3 Transforms

Translate

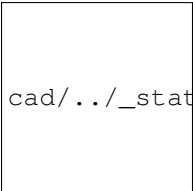
Translating an object is the same as moving it, plug in an object and set the amount of units you want to move that object to position it where you need it. The translation block can be found under the CAD/Transform menu.

Note: The translation is relative to the object's local coordinates.



cad/../../static/images/CAD/difference.png

Fig. 1.20: difference of a cube with a sphere



cad/../../static/images/CAD/intersection.png

Fig. 1.21: intersection of a cube with a sphere



cad/../../static/images/CAD/translate.png

Fig. 1.22: the translation block

Scale

Rotate

Extrude

Lathe

Path

Clone

1.2.4 Lights

Shadow map enable

Ambient Light

Directional light

Hemisphere light

Point Light

Spot light

Set shadow

Set light shadow

1.2.5 Materials

Set color

Set texture

Set material

Basic material

Lambert material

Phong material

Texture loader

Use a texture loader to load image resources as textures, to add a texture loader block go to CAD, then, Material, find the texture loader block and click it.

Once you have an object loader in your workspace, click on the gear icon to open a file selection window, if you want to load a file from your PC click the “load” button” and browse your PC for an OBJ or STL file. Also you can enter or paste an URL in the text box and then click the “load url” button to load from a web location.



```
cad/../../static/images/CAD/textureloader.png
```



```
cad/../../static/images/CAD/loadassetstexture.png
```

you will see a preview of the file, then you can enter or click the “close” button, to close the window and go back to your workspace

Then, to use the loaded texture you can go to CAD - Material, select a “setTexture” Block, and snap the texture loader instead of the default texture loader of the block, the result should look as follows:



```
cad/../../static/images/CAD/loadedtexture.png
```

1.2.6 Groups

New group

In MakerSCAD , you can group objects by declaring group variables, to do this, you can use the “Create Group” block

First go to “Variables” and select the “var” block, then go to the “Tools” section and select a “Create Group” block, snap the new group to the variable

Object add

When you declare a variable as a group, you can use a special block called “Create group”.

in the previous section, we created a variable as a group, now go to tools and select the block that says “group.add”

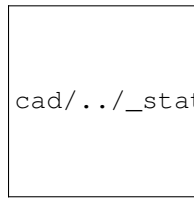
Change the word “group” for the name you used in your variable and snap a new part to it, for this example we will use a torus.

Now add another “create group” block, use the same variable name you declared before, snap a sphere to it.

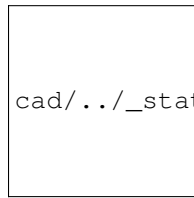
finally add the variable to the scene with a “scene.add” block, remember that you have to snap a “var” block to it

New Group

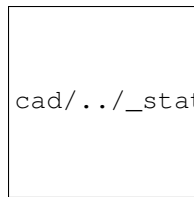
MakerSCAD also has a special block called “new group” that can be used to abbreviate the steps in the past sections, find this block in CAD - Groups. Note that it has a plus and a minus buttons, you can use them to add or remove objects, so, you can snap as many objects as you need.



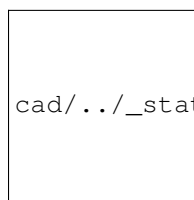
`cad/../../static/images/CAD/newgroup.png`



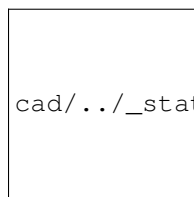
`cad/../../static/images/CAD/groupvariable.png`



`cad/../../static/images/CAD/creategroup.png`



`cad/../../static/images/CAD/groupadd.png`



`cad/../../static/images/CAD/othergroup.png`

1.2.7 Animation

Animation loop

1.2.8 Interaction

1.2.9 Physics

Basic physi scene

Physi material

1.2.10 Camera

1.2.11 Particles

1.3 Logic

In programming logic, is very useful to compare values of variables, MakerSCAD has Comparison operators to help with these tasks the first block is the “If” statement that allows us to compare between two variables, or compare if a variable has an specific value, or even if it is null or empty.



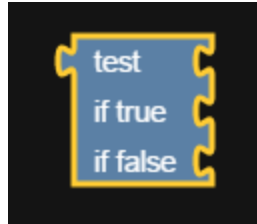
we can help us with another blocks, like the comparator block, that will be used to determine if a variable is equal, not equal, bigger than or smaller than another variable.



also there is a boolean logic operator block, it is used to use two conditions inside one single comparison “if” block, it has a dropdown list where you cn select “and” or “or” agregatio functions



There is also a “test” block, best known in programming as the “ternary operator”, it makes in a single line the comparison between variables using the if statement and an “else condition”, executed when the main condition is not true. in the generated code you can see that instead of “if” and “else” words it uses “?” and “:” symbols respectively

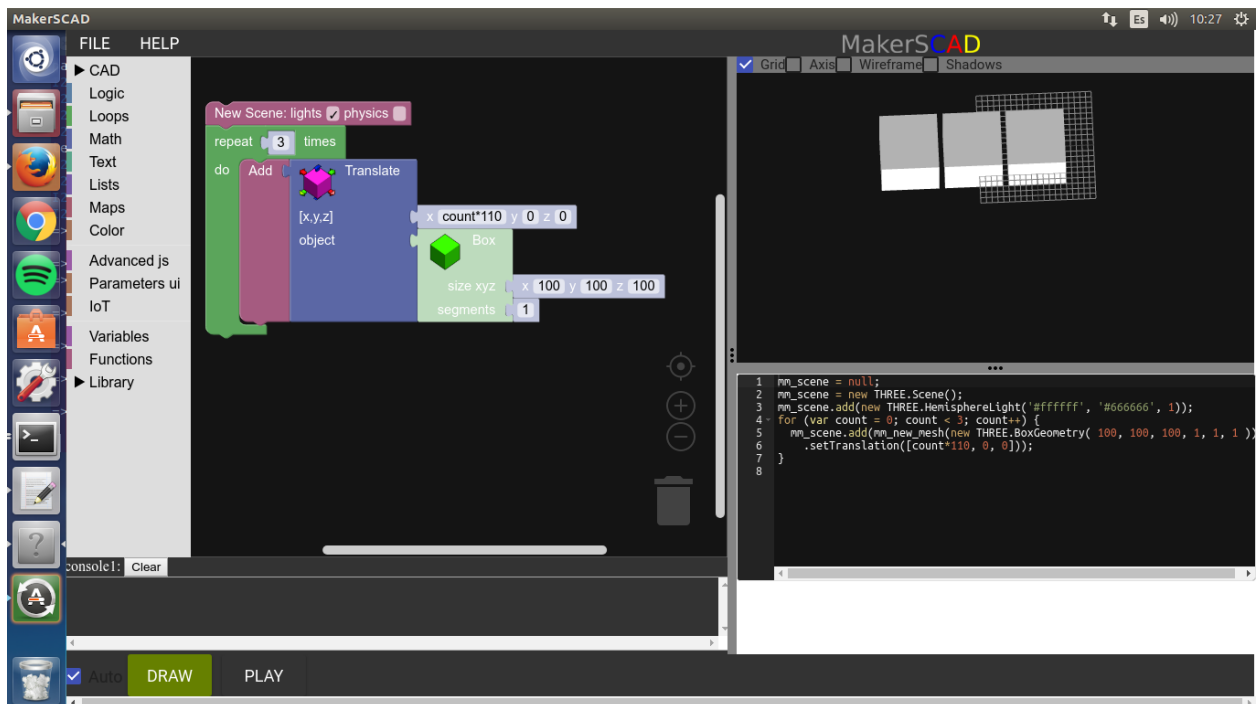


1.4 Loop

1.4.1 Repeat N times

You can use this block to make a loop in a known number of iterations, in the block you will see a text box that has a default value of ten, you can modify this value to the number of times you want the loop to be executed.

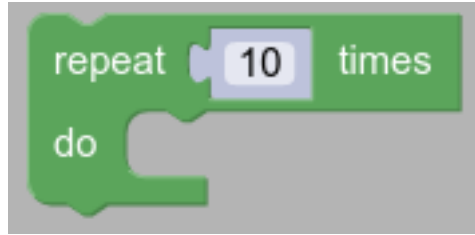
Lets make the example of the image above:



First, go to “Loops” and add a “Repeat n times” block, modify the value to 3.

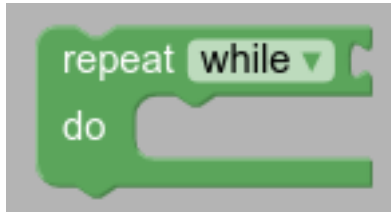
Now go to CAD - Scene, and bring an “Add” block, snap a Translate block to it (CAD - Transform -Translate), look at the X Y Z control, here we can put values, but also we can use programming expressions, we will put a formula in x to have each cube separated by 10 units from the others, (the size of each cube is 100), note in the generated code that there is a variable called “count” that begins with 0 and increments by 1 on each iteration, if we put “count*110” we are saying that the first cube, will have an x position of $0 * 110 = 0$, the second is $110 * 1 = 110$ and the third one is $110 * 2 = 220$

Now, you will have 3 cubes in the workspace separated to each other by 10.

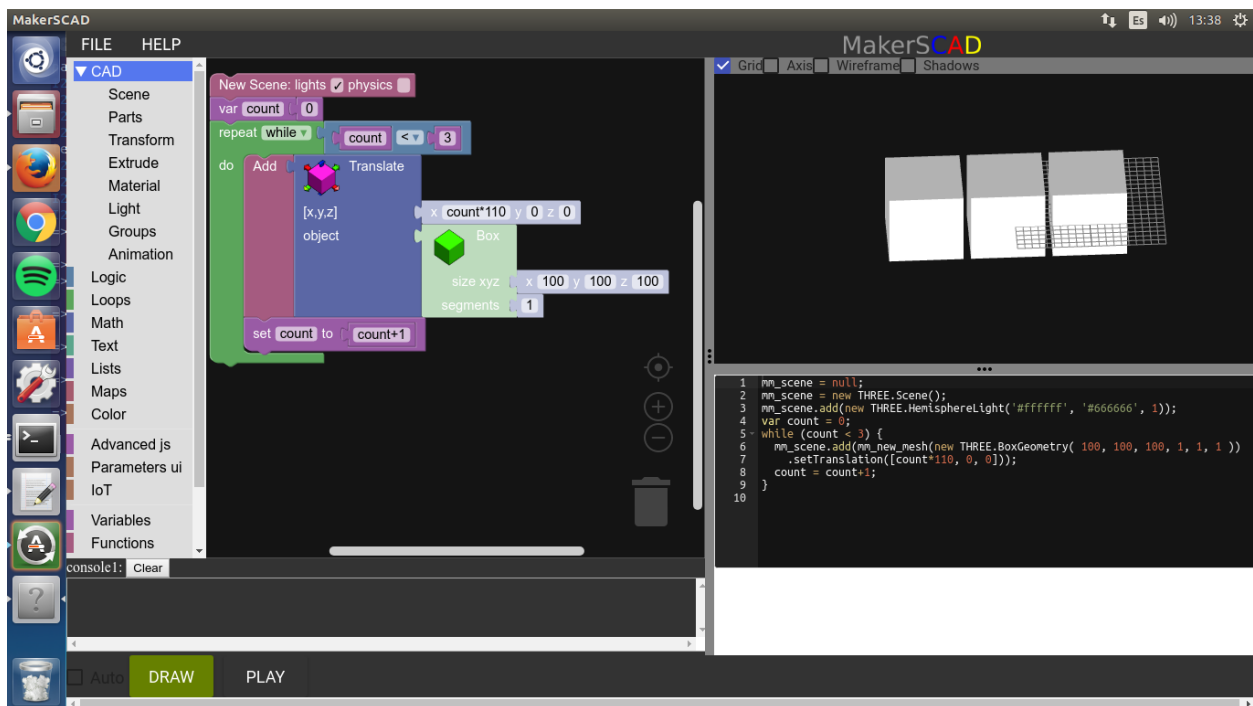


1.4.2 Repeat While

When you have a condition that must be true to continue the loop, but not necessarily know the exact number of iterations you can use the while loop.



To do the last exercise using a while loop go to “Loops” and add a repeat while” block to the workspace.



Warning: Be sure that the Auto render option is not checked in when you are beginning a while loop, and only click “Draw” once you have set the condition to exit the while”.

The first thing you will note is that there is a dropdown menu where you can select “while” or “until” the difference is that if you select “while” the loop will continue while the condition is true, but if you select “until” the loop will continue while the condition is false, for the exercise we will leave the “while option”

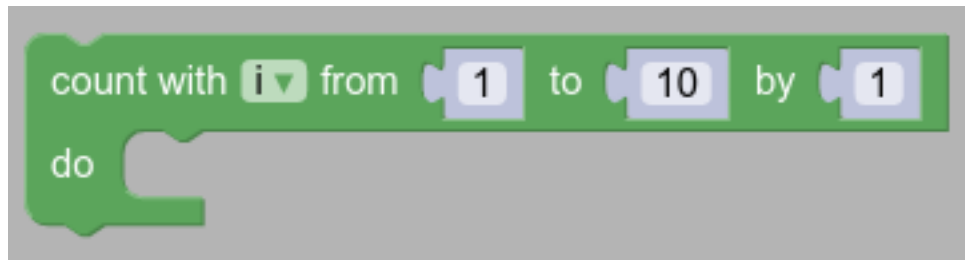
Then, go to Variables and select the block that says “Var1”, snap it to the while and write the following expression ::
count < 3

The expression says that when control variable “count” reaches the value of 3 the loop will break and will not do more iterations. If you had used “until” instead of while, the expression could had been :: count = 3

you can also use a comparator block and var blocks to avoid writing the whole expression

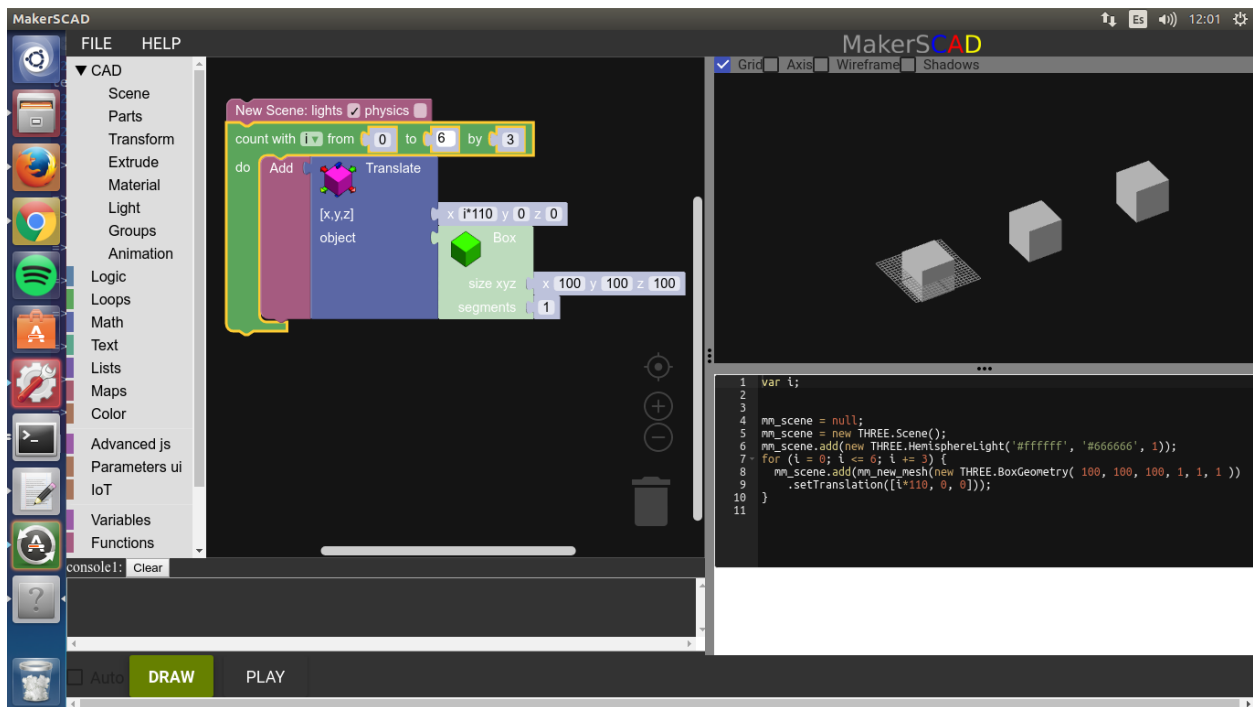
1.4.3 Count From/To

You can set the increment and the initial value of the control variable in a for loop, for that purpose MakerSCAD has the Count From/To block.



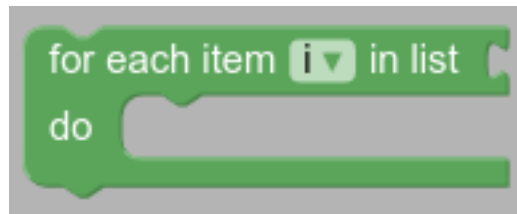
we will make an exercise very similar to the one on the Repeat n Times section, but the difference is that this time variable called “count” will be called “i” (in later versions you will change the name), begins with 0, ends in 6 and will have an increment of 3 on each iteration. That way we will have the same 3 cubes, the first cube will have an x position of $0 * 110 = 0$, the second is $110 * 3 = 330$ and the third one is $110 * 6 = 660$

Now, you will have 3 cubes in the workspace separated to each other by 230.



1.4.4 For each item in a list

In MakerSCAD there is a block to iterate between each element of a list.



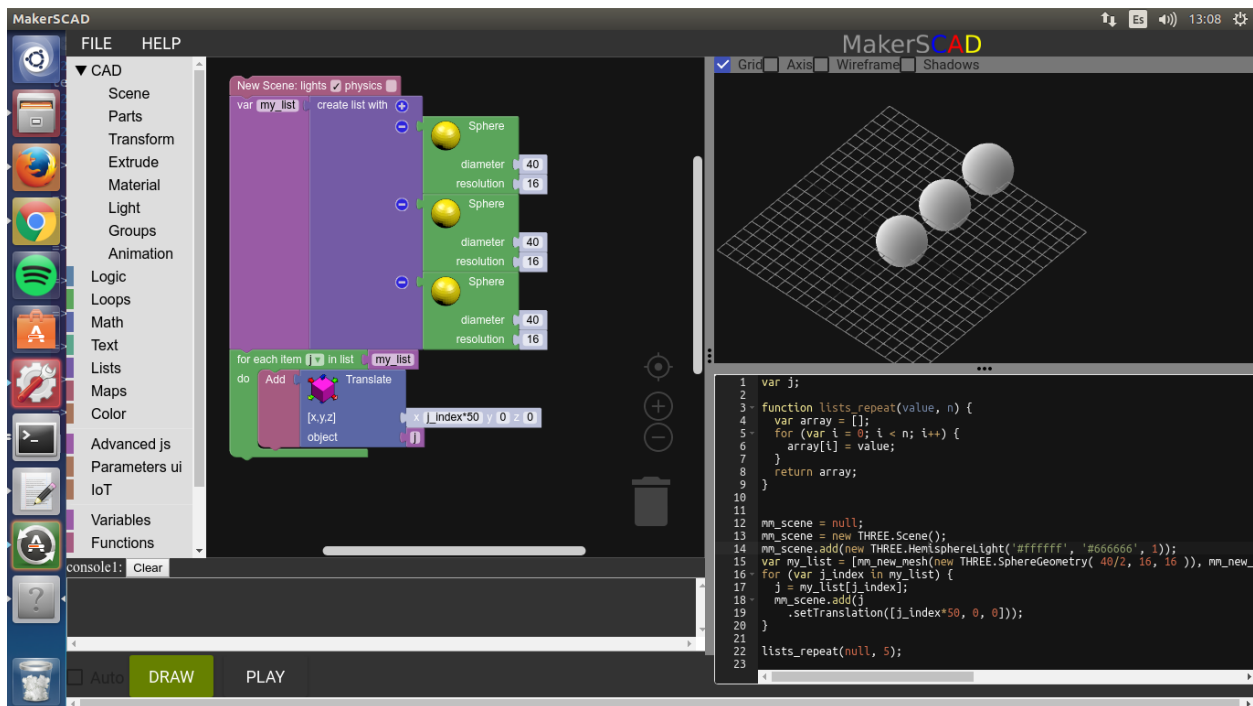
To do this example, let's create a variable for the list, go to Variables and pick a var block, name it "my_list" then go to "Lists", pick a "Create list with" block and snap it to the variable.

look at the add sign (+) in the block click it twice to create two slots more and snap a sphere on each one, you will have now a list with 3 spheres, set each one a diameter of 40.

Now go to Loops and pick a "For each item block", snap a var1 block and write "my_list" on it, that will make the loop to iterate over each item on "my list" and will call each element "j"

Go to CAD - Scene and pick an "Add" block, then CAD - Transform and pick a translate and use them into the loop, in the x control of the translate block write "j_index * 50" and finally add a var block to use the j variable.

your exercise should look like the picture, you will see three spheres

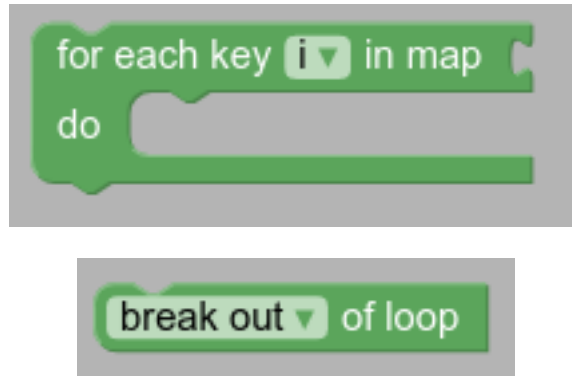


finally

1.4.5 For each key in a map

1.4.6 Break out of a loop

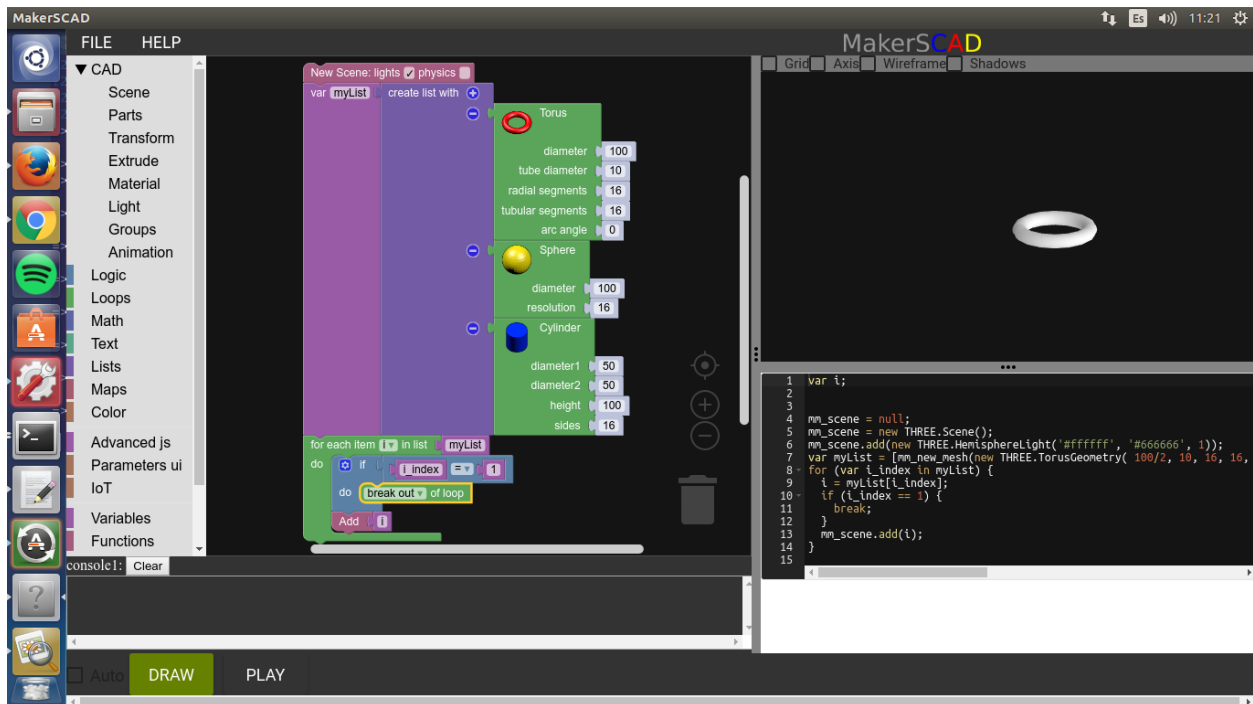
you can interrupt the execution of a loop with a special Block called break

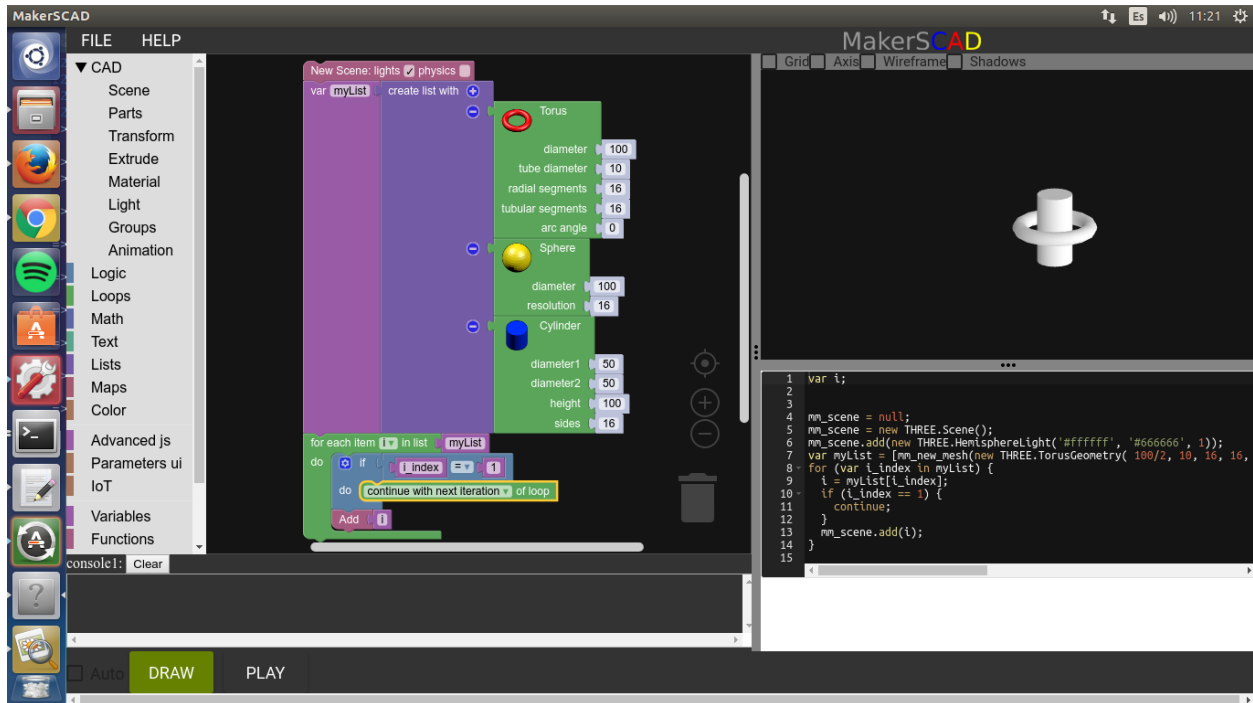


This is very useful for example when you are searching for an element in a list, you ask for the property you look for, and when you find it you can break out the loop to not continue.

the block also has an engine icon where you can change the expression from “break” to continue, the difference is that while the “break” expression interrupts the loop at all, the continue only pass by the iteration where the condition is accomplished but go immediately to the next iteration.

In the following image we can see a list with a torus, a sphere and a cylinder, in that order, then, a loop that iterates each item in the list, and a break expression that is executed if the index is 1, remember that in programming, a list or array begins with 0, so, what are we doing is to break out in the second iteration, and only the torus is added to the scene.





1.5 Math

Makerscad has useful mathematical functions, you can find them in the Math section of the menu

1.5.1 Number

In MakerSCAD there is a “Number” block, by default it has a zero but there you can enter the value that you need.

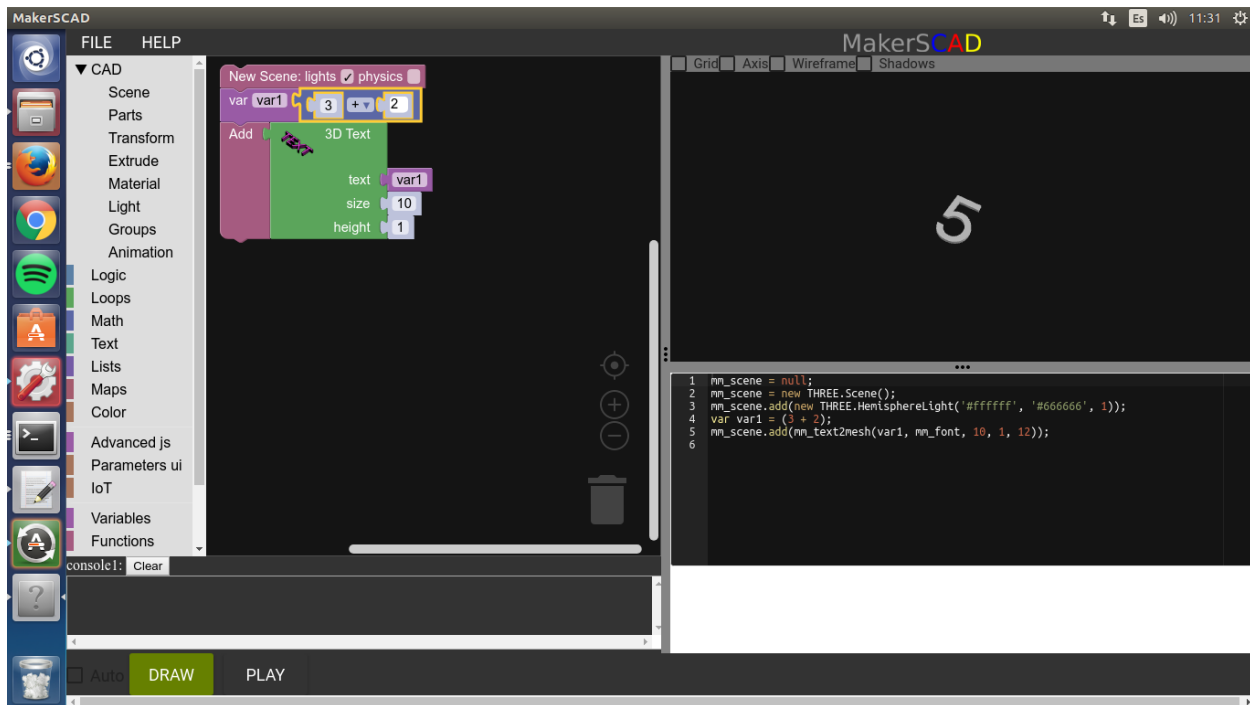


1.5.2 Basic operations block

This block allows you to perform basic arithmetic operations between two numbers like addition (+), subtraction (-), multiplication (*), division and power, to use it simply write a number on each side of the block and select the desired sign from the dropdown at the center with the little triangle.

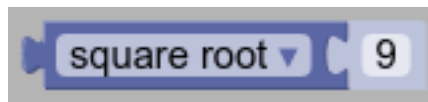


you can use a variable and a 3D text (CAD - Parts - 3Dtext) like in the picture to see the result



1.5.3 Advanced operations block

it includes operations like square root, absolute, negative, natural logarithm, log10, e power, 10 power, in this block you can snap another block with an expression, or simply write a number at the right of the block.



For this example lets start with the last example, add another variable block called “var2”, and then snap an “Advanced operations block” use the dropdown of the block to select “10” and snap a block of “var2”, look at the picture as a reference

1.5.4 Trigonometric functions

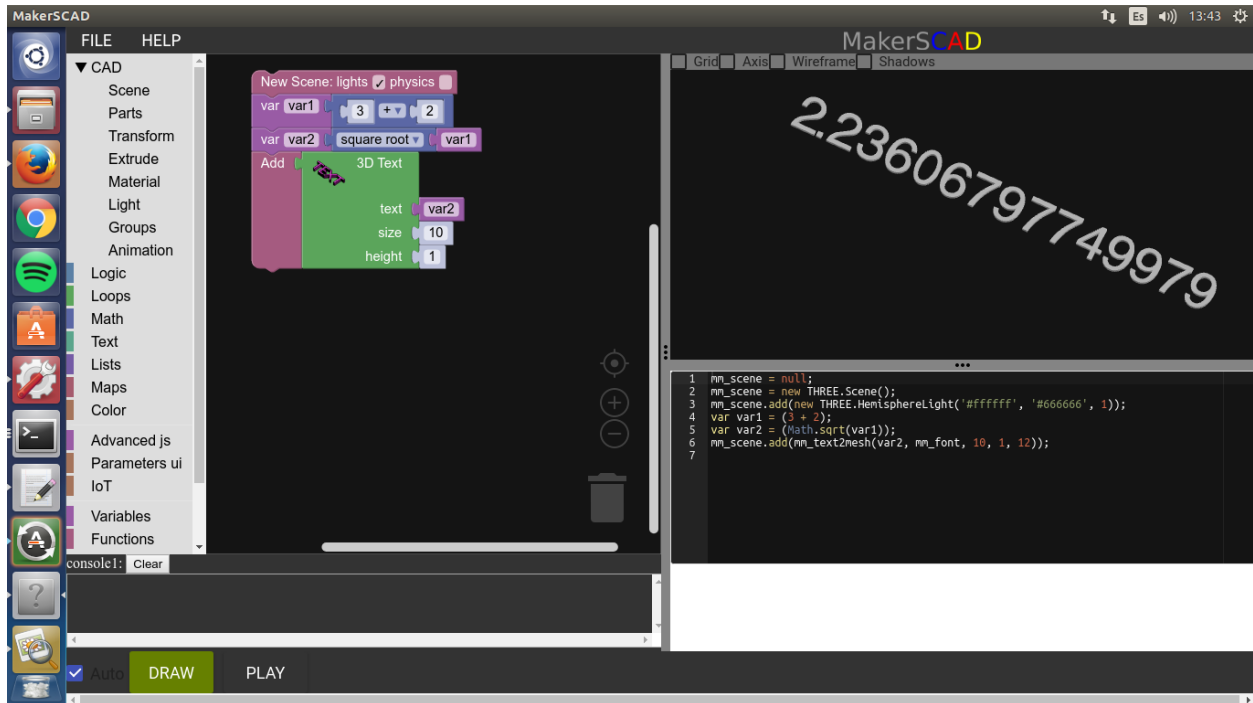
sin cos tan asin acos atan

1.5.5 Constants

pi, e, phi, sqrt(2) (sqrt(1/2)) infinity

even odd prime whole positive negative divisible by

change item: this block generates a programming expression that validates if a variable is a number and then adds the number indicated in the right of the block. if the variable is not a number, the block considers it as a zero



1.5.6 Round

This block allows to round a number or expression

1.5.7 Aggregation functions

block sum, min, max, average, modes, standar deviation, random item

1.5.8 Reminder

1.5.9 Constraint

1.5.10 Random

1.5.11 Random fraction

1.6 Text

1.7 Lists

1.8 Map

1.9 Color

1.10 AdvancedJS

1.11 Variables

1.12 Function

1.13 Examples

1.13.1 MakerSCAD Overview

In this video you can see some quick examples about Makerscad

1.14 Tutorials

Here you will find the videos to learn how to use MakerSCAD!
check out our YouTube Channel to see the latests videos.

1.14.1 Basic Shapes tutorial

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`