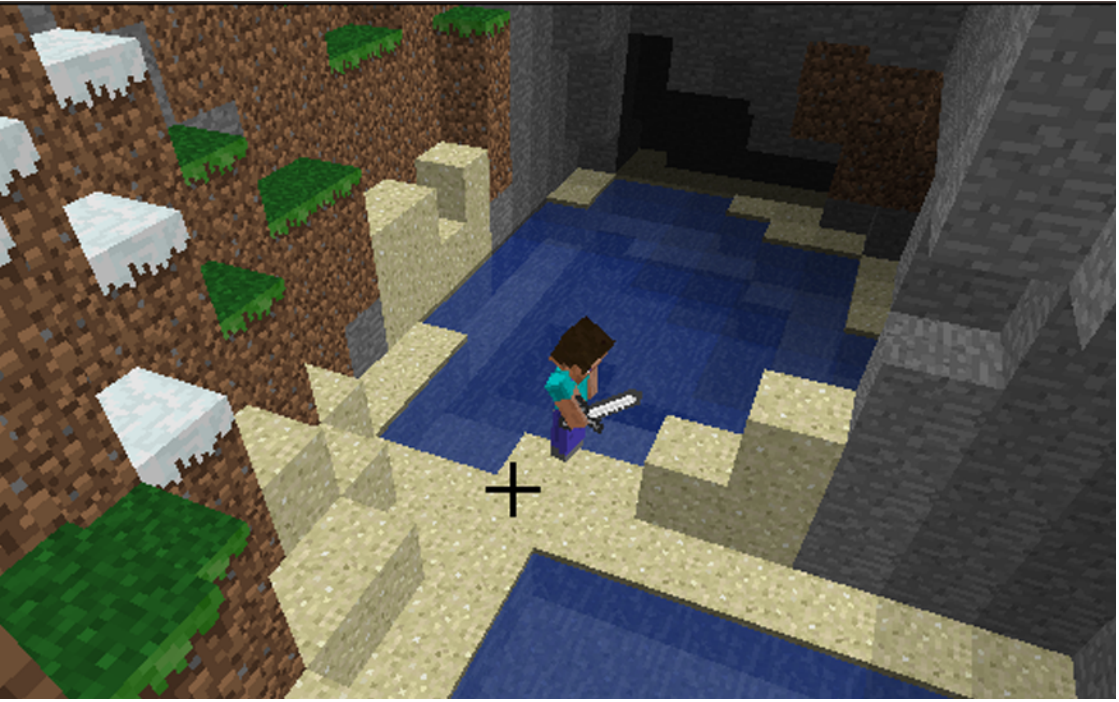


6. Block Fighter

Using the Minecraft Pi API you can create your own mini-games. This means you can build upon the existing game to create your own ideas quickly without having to start from scratch.

In this guide we'll learn how to use the `for` loop to create a mini-game that gives the player points for the different blocks they hit/right-click with a sword. By using `time` in the Python program the player has a time limit in which they must hit the blocks.

Try adapting the code when you're done. For example you could make your sword place rare blocks.

Code

Import the API and wait a bit

As usual we import the API and connect to the game.

Import the API and wait a bit

We wait for 60 seconds in order to give the player time to hit some blocks with a sword. The `points` variable will store the player's score.

Record the block hits

The `pollBlockHits()` function returns a `list` of block hits. A `list` contains several a values that can be stored using a single variable. This means you can store several values without needing a separate variable for each.

Loop through every block hit

A `for` loop is a type of loop that repeats for each item in a `list`. In this case it loops through each item in the `hits` list and stores each value in the `hit` variable, one at a time. The code inside the loop gets the co-ordinates that the player hit and finds the block type at those co-ordinates. The value of the block type is then added to the `points` variable.

Post the points to the chat

Once the `for` loop has ended we display the number of points that the player scored to chat in the Minecraft game.

```
1  import mcpi.minecraft as minecraft
2  mc = minecraft.Minecraft.create()
```

```
3  import time
4  time.sleep(60)
5  points = 0
```

```
6  hits = mc.events.pollBlockHits()
```

```
7  for hit in hits:
8      x = hit.pos.x
9      y = hit.pos.y
10     z = hit.pos.z
11     points = points + mc.getBlock(x, y, z)
```

```
12 mc.postToChat("You got " + str(points) + "
       ↪ points.")
```

What you've learned

lists A `list` is a datatype that can store several values. Think of it like a shopping list, you have a number of items in an order. The `hits` variable is an example of a list in our program as it stores all of the block hits that the player made in 60 seconds.

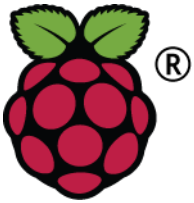
pollBlockHits() The `pollBlockHits()` function returns a list of all of the block hits that the player has made with a sword since the start of the program. The list contains co-ordinates of the blocks that the player has hit. Only right-clicks with a sword will be returned. The co-ordinates are accessed on lines 9–11.

for loops A `for` loop will repeat a section of code a number of times. In the example the `for` loop will repeat for every item in the `hits` list. A `for` loop stores the value of each item in a list in a variable that changes each time it runs. For example in our code the `hit` will hold the value of the current item in the `hits` list each time its run. This variable can be used in the body of the loop.

Extensions

Here are some suggestions to extend your code and make it do different things. Even better if you come up with your own ideas.

- Make a magic wand. Rearrange the code so that every time the player hits a block with their sword, it is magically transformed into a rare block type.



For further exercises check out Python Programming with Minecraft Pi, the book available as a free pdf from www.arghbox.wordpress.com