Use Python to code new creations in Minecraft
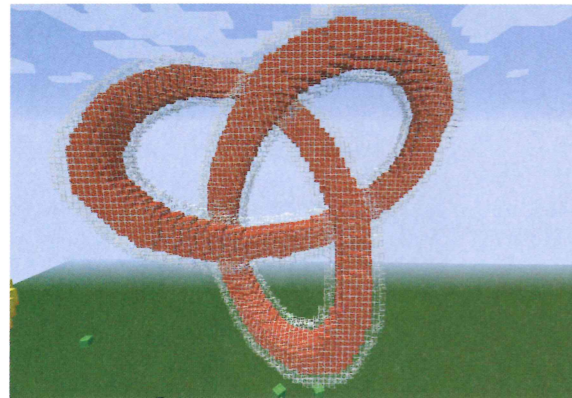
Tap directly into Minecraft with Python and produce fantastic creations using Forge mod

Calvin Robinson teaches Computer Science at GCSE and A-Level and works as a computing consultant in Education, helping schools across London deliver a high-standard computing curriculum.
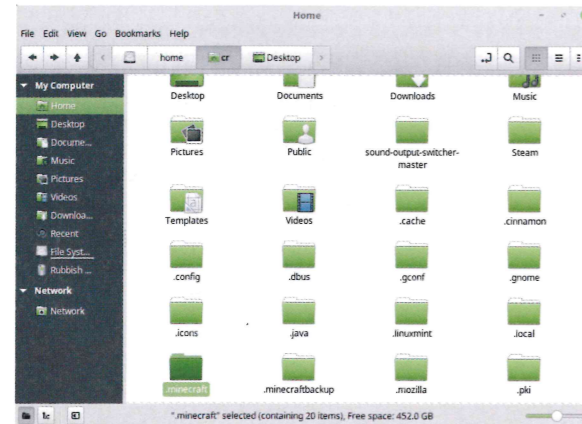

Sometimes, Minecraft can seem far more than just a game. It's an incredibly creative tool and with the use of Redstone and Command Blocks you can produce some amazing worlds. We're taking things a step further by enabling you to plug Python code directly into Minecraft. What you do with it is completely up to your imagination! MCPiPy was developed by 'fleap' and 'bluepillRabbit' of https://mcpipy.wordpress.com, to connect MineCraft Pi Edition with Python on the Raspberry Pi, using open APIs.

However, with the use of Forge we have put together a package that enables the use of Python in retail Minecraft. We're using Raspberry Jam developed by Alexander Pruss, a Forge mod for Minecraft which implements most of the Raspberry Juice/Pi API.

Right Backup your original .minecraft dir and copy over the modded install from McPiFoMo

**Across** You'll need to run Minecraft 1.8 with the pre-installed Forge profile

What you'll need

- Minecraft
  http://www.mojang.com/games
- Python
  https://www.python.org
- McPiFoMo
  http://rogerthat.co.uk/McPiFoMo.rar

Extract the new **.minecraft** directory from McPiFoMo directly into your **home** directory. If you have worlds you'd like to carry over, copy the **Saves** directory from your backup **.minecraft** directory into the new one with `cp -r ~/.minecraft-backup/Saves ~/.minecraft/`

01 Replace your .minecraft directory
Backup **.minecraft** in your **home** directory. If you're using a GUI, you may need to press CTRL+H to view the hidden directories. In terminal `mv ~/.minecraft ~/minecraft-backup` should suffice.
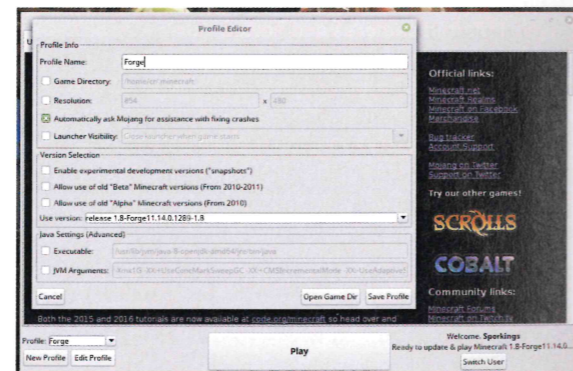
02 Launch Minecraft in Forge mode
Launch Minecraft as you normally would, but after logging in, select the Forge profile. This should load Minecraft 1.8 with Forge 11.14. You can play around with the latest version of Minecraft and download and install an updated Forge if you wish, but these are the versions we've found most compatible with Raspberry Jam.

You'll know you're running the correct profile when you see the version numbers in the bottom left corner of the window. Create a new super flat world in singleplayer creative mode and you're ready to begin coding. We've included a single 'Flat' world pre-installed with the McPiFoMo package.
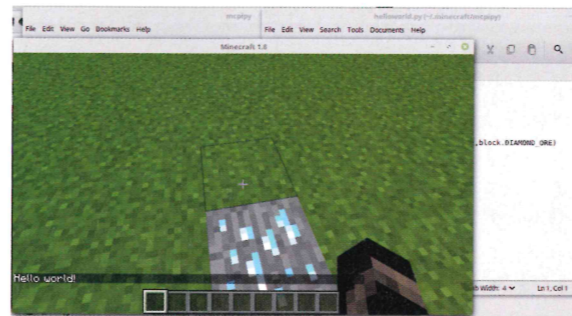
03 Hello World – Implement chat commands
Using your favourite text editor, you'll need to create a new **helloworld.py** file and save it in **~/.minecraft/mcpipy** directory:
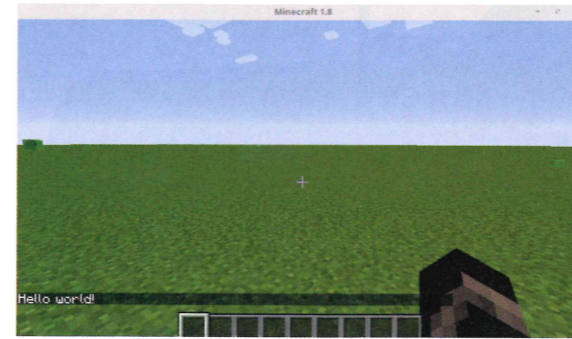

```
from mc import *
mc = Minecraft()
mc.postToChat("Hello world!")
```

Return to Minecraft and type `/python helloworld` Minecraft will now run your python script, which should result in a chat command saying Hello world!

04 Create blocks
Now, by using setBlock() and getPos() commands we can place blocks into the world relative to our player's position. Try adding the following two lines to your **helloworld** script:

```
playerPos = mc.player.getPos()
mc.setBlock(playerPos.x,playerPos.y-1,playerPos.z,DIAMOND_ORE)
```

Then run `/python helloworld` again in Minecraft. You'll see the chat message again, but this time if you look down to the ground below your player character, you'll see a diamond block has also been placed at your feet. You can try replacing `DIAMOND_ORE` with any other Minecraft block ID (i.e. DIRT/GRASS).

05 Mmm, doughnuts
One of the pre-fab scripts that you will find in the MCPiPy collection is the doughnut:

```
from mc import *

def draw_donut(mcx,mcy,mcz,R,r,mcblock):
    for x in range(-R-r,R+r):
        for y in range(-R-r,R+r):
            xy_dist = sqrt(x**2 + y**2)
            if (xy_dist > 0):
                ringx = x / xy_dist * R #
nearest point on major ring
                ringy = y / xy_dist * R
```
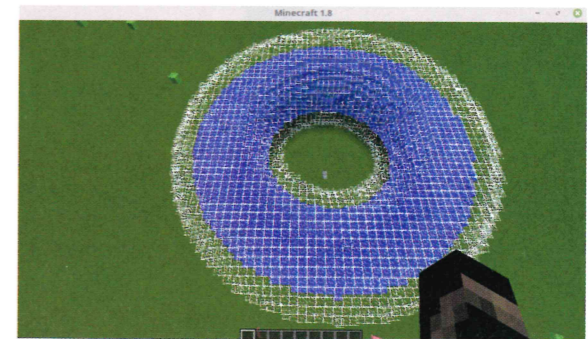
Left You can create water-filled glass doughnuts with a simple for loop


```
                ring_dist_sq = (x-ringx)**2 +
(y-ringy)**2

                for z in range(-R-r,R+r):
                    if (ring_dist_sq + z**2 <=
r**2):
                        mc.setBlock(mcx+x, mcy+z,
mcz+y, mcblock)

mc = Minecraft()

playerPos = mc.player.getPos()

draw_donut(playerPos.x, playerPos.y + 9,
playerPos.z, 18, 9, GLASS)
mc.postToChat("Glass donut done")
draw_donut(playerPos.x, playerPos.y + 9,
playerPos.z, 18, 6, WATER_STATIONARY)
mc.postToChat("Water donut done")
```

By changing the block ID from WATER_STATIONARY you can fill the doughnut with any object type. Try filling the glass with lava. Then try changing outer shell from glass to TNT.

06 Common errors
If you get a 'Script not found' error, this probably means that you don't have the mod scripts installed in your Minecraft directory. Check that you've replaced **.minecraft** with the one from McPiFoMo.

If you receive a 'Cannot run program "python"' error, your game cannot locate Python. Ensure you've got the latest version of Python installed, and that it's installed in Path. In the Bash shell type `export ATH="$PATH:/usr/local/bin/python"` to check. Should you come into any problems with memory leakage or infinite loops, you can stop a script that's running by just typing `/python`. ∎

CANNOT FIND SCRIPT
If you see red text stating 'Cannot find script', check the name and location of your PY file. All of your Python scripts should be in **~/.minecraft/mcpipy**, for Python and Minecraft to be able to locate them. You don't need to append '.py' to the end of your run command, just be sure you're using the exact name of your Python script file. `/python doughnut` will work just as well as `/python doughnut.py`, so long as **doughnut.py** is stored in **~/.minecraft/mcpipy**. If you can't see this directory, remember to un-hide your files (CTRL+H).