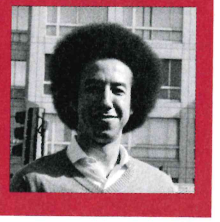
Use Python code to create pixel art in Minecraft

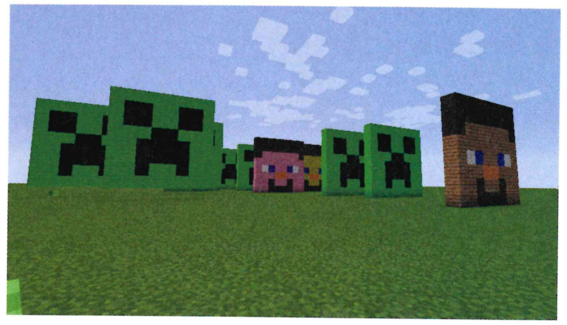
Continuing our series on coding in Minecraft, this issue we'll show you how to create automated pixel art



Calvin Robinson is head of computing and network manager at an all-through state school. Calvin also consults with schools all over London, helping provide high-quality teaching and learning in computing.

Resources

- **Minecraft** www.mojang.com/games/
- **Wireshark** www.wireshark.org
- **Python** www.python.org
- **McPiFoMo** http://rogerthat.co.uk/McPiFoMo.rar
- **Block IDs** http://minecraft-ids.grahamedgecombe.com



Creating pixel art in Minecraft can be quite therapeutic, but placing the blocks down one-by-one can also be incredibly time-consuming. However, by using Python and a handy little 'for' loop or two, we can create scripts to automatically build prefabricated pixel art directly into our Minecraft worlds.

If you're using Minecraft Pi edition on a Raspberry Pi, no additional software is necessary. We've also put together a number of tools to ensure this hack works on Linux, with a retail version of Minecraft. Therefore as a prerequisite, we assume you've installed McPiFoMo from our previous three issues. McPiFoMo includes MCPiPy by 'fleap' and 'bluepillRabbit' of MCPiPy.com, and Raspberry Jam developed by Alexander Pruss.

In this tutorial we're going to use Python to create an array that'll build blocks in rows and columns, to produce some pixel art. You'll need to keep your block ID list to hand, so you can select a range of blocks matching the colours you desire. We're going to be using wool for the choice here, but you're welcome to use any blocks that meet your purpose.

All Python scripts should be saved in the directory **~/home/.minecraft/mcpipy/**, regardless of whether you're running Minecraft Pi edition or retail Linux Minecraft. Be sure to run Minecraft with the Forge 1.8 profile that's included in our McPiFoMo package.

01 Setting up the code

Here's the main code listing...

```
from mcpi import minecraft
mc = minecraft.Minecraft.create()

#Wool is blockID 35
#Green wool is blockID 35:5 (type 5)
woolBlock = 35
woolBlockGreenType = 5
#Black wool is blockID 15
woolBlockBlackType = 15

pixelArt = []

pos = mc.player.getTilePos()
x = pos.x
y = pos.y + 7
#8 blocks above ground)
z = pos.z

for row in range(len(pixelArt)):
    for pixel in range(len(pixelArt[row])):
        if pixelArt[row][pixel] == 0:
            mc.setBlock(x, y - row, z +
pixel, woolBlock, woolBlockBlackType)
        elif pixelArt[row][pixel] == 1:
            mc.setBlock(x, y - row, z +
pixel, woolBlock, woolBlockGreenType)
```

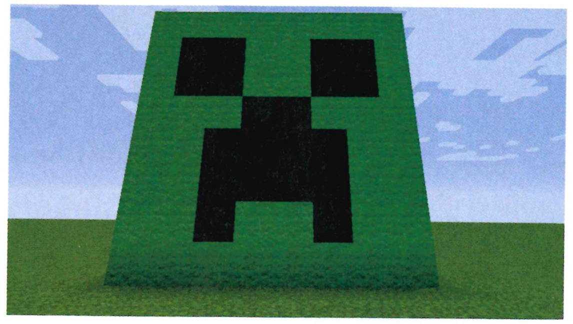
02 Dissecting the code – Block types

There are several points of interest in the above code. Firstly, you'll notice we're using two variables per block. To set our block ID to 35:5, which is green wool, we use `woolBlock 35` and `woolBlockGreenType` 5. We'll do this for all the blocks that we add to our pixel art. If there's no block 'type', simply use 0.

In our `for` loop, we call `woolBlock`, `woolBlockGreenType`. If you're regularly using blocks with no type (such as `Diamond: 57`), you can drop the type variable altogether.

03 Dissecting the code – Positioning

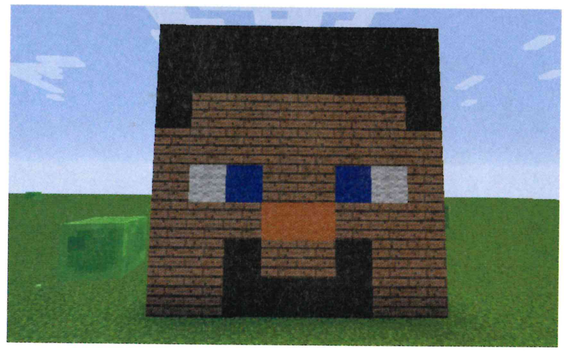
You'll also notice the position of our blocks is +7 on the y-axis. That's because we'll be creating pixel art of eight blocks high, and we want to make sure it appears upward from ground level.



```
pos = mc.player.getTilePos()
x = pos.x
y = pos.y + 7
z = pos.z
```

Our blocks are built in rows:

```
for row in range(len(pixelArt)):
    for pixel in range(len(pixelArt[row])):
        if pixelArt[row][pixel] == 0:
            mc.setBlock(x, y - row, z +
pixel, woolBlock, woolBlockBlackType)
        elif pixelArt[row][pixel] == 1:
            mc.setBlock(x, y - row, z +
pixel, woolBlock, woolBlockGreenType)
```



04 Drawing a 2D creeper

Now, to create your first piece of pixel art you'll need to create a list of 1s and 0s in the `pixelArt` list:

```
pixelArt = [
[1, 1, 1, 1, 1, 1, 1, 1],
[1, 0, 0, 1, 1, 0, 0, 1],
[1, 0, 0, 1, 1, 0, 0, 1],
[1, 1, 1, 0, 0, 1, 1, 1],
[1, 1, 0, 0, 0, 0, 1, 1],
[1, 1, 0, 0, 0, 0, 1, 1],
[1, 1, 0, 1, 1, 0, 1, 1],
[1, 1, 1, 1, 1, 1, 1, 1]]
```

If you squint, you can make out a Creeper head in the 1s and 0s.

05 Creating a Steve head – adding more block type variables

Obviously a Creeper is just black and green, but if you're looking to create something more advanced, you'll need to add more colour blocks by adding more variables with block IDs and types. Here are the colours we'd need to create a Steve head:

```
#Wool is blockID 35
#Orange wool is type 1
woolBlock = 35
woolBlockOrangeType = 1
#Blue wool is type 11
woolBlockBlueType = 11
#Brown wool is type 12
woolBlockBrownType = 12
#White wool is type 0
woolBlockWhiteType = 0

#Pink wool is type 6, but double jungle
wood slabs look good (125:3)
woodBlockPink = 125
woodBlockPinkType = 3
```

06 Creating a Steve head – extending your for loops

We'll also need to add these new variables to our **for** loop below, so that they're added to each row. We can do this as many times as we like:

```
for row in range(len(pixelArt)):
    for pixel in range(len(pixelArt[row])):
        if pixelArt[row][pixel] == 0:
            mc.setBlock(x, y - row, z +
pixel, woolBlock, woolBlockOrangeType)
        elif pixelArt[row][pixel] == 1:
            mc.setBlock(x, y - row, z +
pixel, woolBlock, woolBlockBlueType)
        elif pixelArt[row][pixel] == 2:
            mc.setBlock(x, y - row, z +
pixel, woolBlock, woolBlockBrownType)
        elif pixelArt[row][pixel] == 3:
            mc.setBlock(x, y - row, z +
pixel, woolBlock, woolBlockWhiteType)
        elif pixelArt[row][pixel] == 4:
            mc.setBlock(x, y - row, z +
pixel, woodBlockPink, woodBlockPinkType)
```

07 Running your code

Save your **pixelart.py** file in **~/home/.minecraft**, open Minecraft with the Forge 1.8 profile, create/load a world and type `/python pixelart` to spawn your creation. You can do this as many times as you like to build a new copy of your prefabricated pixel art. ■

■ Finding new pixel art

If you're stuck for ideas for your pixel art creation, or you need a little assistance, try Googling for pixel art templates. There are plenty of free templates available, with grid-lines and countable boxes. You can mirror these by counting the rows/columns and adjusting your Python list to match. Our Creeper and Steve heads were 8×8, but you could easily adjust this to 12×12 by adding four more lines of code and appending four more numbers to each row.

We'd love to see your 2D/3D Minecraft pixel art creations. Pop us a tweet with your best screenshot **@linuxusermag** and let us know which distro you're running Minecraft on.