The MagPi ESSENTIALS

[CHAPTER EIGHT]
MINECRAFT SPLAT

Create an exciting two-player game in Minecraft: Pi, inspired by Nintendo's hit game game Splatoon…

Below Can you escape the Lava Trap?

You can play Minecraft: Pi Edition in multiplayer mode when two or more Raspberry Pis on the same network join the same world. In this guide, we use this technique to create a simple versus game that's similar to Nintendo's Splatoon, which sees two teams trying to paint the game area in their team colours.

The objective of our game is very similar: to splat (turn to your team colour) as many blocks as possible for your side, while the opposing team will also be splatting blocks and claiming your splats for themselves. You will earn points for each block that's still your colour at the end of the game, and the player with the most splats wins!

MINECRAFT SPLAT IS SPLIT INTO 5 PARTS:

01 Create the framework for the program and make sure your code runs.

02 Build the pitch that will appear when the game starts and be the splat battleground.

03 Splat blocks by hitting them with a sword.

04 Game over and displaying the winner.

05 Making a better game.

CREATE THE PROGRAM

Open Python 2 from the Programming menu. The Python shell will appear; when it does, create a new program using File>New Window. It's also a good idea to save your program now, using File>Save.

Import the Python modules you'll need:

```python
from mcpi.minecraft import Minecraft
from mcpi import block
from time import sleep, time
from random import getrandbits
```

You'll need a constant to hold the colour each team will use; it's the colour of the wool block that will be used when a player splats a block. Create a list which holds two values: 13 for green and 14 for red.

```python
TEAMCOLS = [13,14]
```

Create the definition for two functions, which you will complete later in this tutorial:

```python
def buildPitch(mc, pos):
    pass
def splatBlock(mc, x, y, z, team):
    pass
```

You'll need a list to hold the points each team has scored. The first element will be team 1's score and the second team 2's – they should both be set to 0:

```python
points = [0,0]
```

Create the connection to Minecraft and post a message to the screen:

```python
mc = Minecraft.create()
mc.postToChat("Minecraft Splat")
```

At this point, you can run your program and if everything is set up, you should see the 'Minecraft Splat' message posted to the screen.

Now start up Minecraft: Pi Edition. Create a new game and then run your program by clicking Run>Run Module.

BUILD THE PITCH

The game needs a pitch where the action can take place; it's a glass 'room' with two glass walls running down the middle.
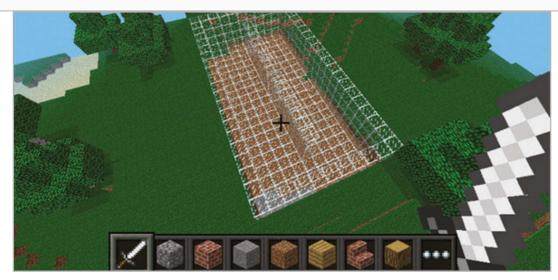
Find the **buildPitch** function in your program:

```python
def buildPitch(mc, pos):
    pass
```

The Minecraft connection, **mc**, and a position, **pos**, where the pitch should be built, should be passed to the function.

Delete the **pass** statement and replace it with the following code, which will create a cube of glass blocks. Then create a cube of air inside it before building the central walls of glass:

```python
def buildPitch(mc, pos):
    # glass cube
    mc.setBlocks(pos.x - 5, pos.y - 1, pos.z - 10,
            pos.x + 5, pos.y + 3, pos.z + 10,
            block.GLASS.id)
    # hollow it out
    mc.setBlocks(pos.x - 4, pos.y, pos.z - 9,
            pos.x + 4, pos.y + 3, pos.z + 9,
            block.AIR.id)
```

```
# add 2 walls down the middle
mc.setBlocks(pos.x, pos.y, pos.z - 7,
        pos.x, pos.y + 3, pos.z - 1,
        block.GLASS.id)
mc.setBlocks(pos.x, pos.y, pos.z + 1,
        pos.x, pos.y + 3, pos.z + 7,
        block.GLASS.id)
```
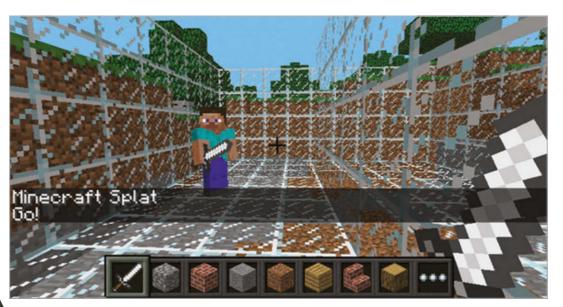
The **buildPitch** function now needs to be called from your program. Add the following code to the end of the program to get the player's position and call the function:

```
pos = mc.player.getTilePos()
buildPitch(mc, pos)
```

Before the game starts, you should also include a delay to let the players get ready, and a message to let them know the game has started:

```
sleep(3)
mc.postToChat("Go!")
```

Run the program. You should see the pitch appear around your player and the message to 'Go!'.

SPLATTING BLOCKS

The blocks of the pitch's walls and floor can be splatted by hitting them (right-clicking) with a sword – when you splat a glass block, it'll turn it into a wool block of your team's colour; splatting a block belonging to the opposition will turn it back to glass.

You earn points for each block splatted with your team's colour, and the opposition will lose a point for each block you turn back to glass.

Find the **splatBlock** function in your program:

```
def splatBlock(mc, x, y, z, team):
    pass
```

Change the function so that it splats the block at the position **x**, **y**, **z** for **team**, which are variables passed to the function. When executed, the function will return the number of points scored for each team.

Delete the **pass** statement and create a list which will hold the points scored for each team:

```
def splatBlock(mc, x, y, z, team):
    pointsScored = [0,0]
```

The variable **team**, which is passed into **splatBlock**, will hold either a 0 or 1, depending on which team splatted the block. Use this value to create a variable to hold the other team:
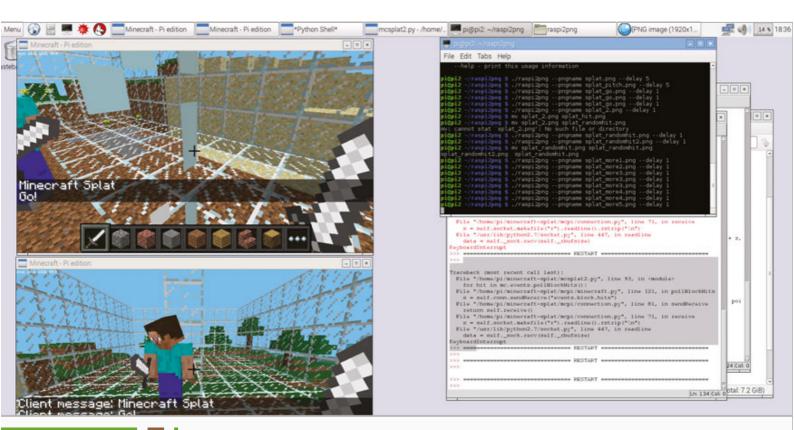
```
        otherTeam = 1 - team
```

Check to see if the block that was hit was a glass block; if it was, turn it into a wool block of the team's colour, and increase the team's score by 1:

```
blockHit = mc.getBlockWithData(x, y, z)
    if blockHit.id == block.GLASS.id:
        mc.setBlock(x, y, z, block.WOOL.id, TEAMCOLS[team])
        pointsScored[team] += 1
```

If the block isn't glass, check to see if it's a wool block of the other team's colour, before turning it back to glass and decreasing the other team's score:

```
elif blockHit.id == block.WOOL.id:
    if blockHit.data == TEAMCOLS[otherTeam]:
        mc.setBlock(x, y, z, block.GLASS.id)
            pointsScored[otherTeam] -= 1
```

The last step in the **splatBlock** function is to return the number of points scored:

```
return pointsScored
```

Now that the **splatBlock** function is complete, you need to add to the code at the bottom of your program which will start the game.

You'll find out how many players are in the game, create a loop which will continue until the end of the game, and call **splatBlock** each time a block is hit.

Get a list of players currently in the game and the time the game started, and store them in variables:

```
players = mc.getPlayerEntityIds()
start = time()
```

Set the variable **gameOver** to False before creating a **while** loop, which will continue until **gameOver** is set to True when the game finishes:

```
gameOver = False
while not gameOver:
```

Use **pollBlockHits()** to find out if any blocks have been hit, before looping through each 'hit' with a **for** loop:

```
blockHits = mc.events.pollBlockHits()
    for hit in blockHits:
```

Every player in *Minecraft* has an entity ID and these are held in the **players** list you created earlier. The player's position in the list will determine what team they are on: even = team 1, odd = team 2. Use the **players** list and the entity ID of the player who hit the block to work out what team they are on:
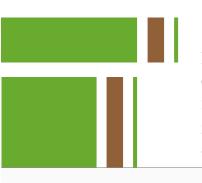
```
team = players.index(hit.entityId) % 2
```

Call the **splatBlock** function, passing the position of the block which was hit and the team who hit it, and add the points scored to the total points for the team:

```
pointsScored = splatBlock(mc,
    hit.pos.x, hit.pos.y, hit.pos.z, team)
    points[0] += pointsScored[0]
    points[1] += pointsScored[1]
```

Run your program and, as before, the pitch should appear around your player. Now, however, hitting blocks (right-clicking while holding a sword) should turn the blocks to coloured wool. You could even get a friend to join your game and test turning your opponent's blocks back to glass.

As you haven't created the code to end the game, the program will continue forever. You can use **CTRL+C** or click Shell>Restart Shell in the Python shell to end the program.

GAME OVER

Each match is 30 seconds long and the game is over when the time runs out. Under the **while** loop, you need to check whether the time now minus the time the game started is greater than 30 seconds. Once the game is over, you should post the team's points to the chat window, along with the winner:

```
if time() - start > 30:
    gameOver = True
    mc.postToChat("Game Over")
    mc.postToChat(
"Green Team = " + str(points[0]))
    mc.postToChat(
"Red Team = " + str(points[1]))
    if points[0] > points[1]:
        mc.postToChat("Green Team wins")
      else:
        mc.postToChat("Red Team wins")
```

Find a friend with a Raspberry Pi, challenge them to a game of Minecraft Splat, and run your program.

MAKING A BETTER SPLAT

The splat made at the moment is less of a splat and more of a blob. If you want to take the program further, in the next section you will use randomisation to splatter the blocks around the block that was hit as well.

After your code to splat the block, loop through each of the blocks around the one which was hit:

```
for hit in blockHits:
    team = players.index(hit.entityId) % 2

        pointsScored = splatBlock(
            mc, hit.pos.x, hit.pos.y, hit.pos.z, team)

        points[0] += pointsScored[0]
        points[1] += pointsScored[1]

        for x in [-1, 0, 1]:
            for y in [-1, 0, 1]:
                for z in [-1, 0, 1]:
```

Using the code **getrandbits(1)**, you can randomly generate a 1 or 0, giving a 50/50 chance of it being 1. If it is, splat the block for the team and add the points to the total:

```
        if getrandbits(1) == 1:
            pointsScored = splatBlock(mc,
                    hit.pos.x + x,
                    hit.pos.y + y,
                    hit.pos.z + z,
                    team)
            points[0] += pointsScored[0]
            points[1] += pointsScored[1]
```

Run your program again. Now, each time you splat a block, it should randomly splatter the blocks around it too.

This is just one improvement you can make to the game; the only limit is your imagination. How will you take it forward and make it your own?

The code for Minecraft Splat is on GitHub at **magpi.cc/29qpm3r**.

MCSplat.py

```python
01.  # import modules
02.  from mcpi.minecraft import Minecraft
03.  from mcpi import block
04.  from time import sleep, time
05.  from random import getrandbits
06.
07.  TEAMCOLS = [13,14]
08.
09.  def buildPitch(mc, pos):
10.      # create the glass cube playing area
11.      mc.setBlocks(pos.x - 5, pos.y - 1, pos.z - 10,
12.                   pos.x + 5, pos.y + 3, pos.z + 10,
13.                   block.GLASS.id)
14.
15.      # hollow it out
16.      mc.setBlocks(pos.x - 4, pos.y, pos.z - 9,
17.                   pos.x + 4, pos.y + 3, pos.z + 9,
18.                   block.AIR.id)
19.
20.      # add 2 walls down the middle
21.      mc.setBlocks(pos.x, pos.y, pos.z - 7,
22.                   pos.x, pos.y + 3, pos.z - 1,
23.                   block.GLASS.id)
24.
25.      # add 2 walls down the middle
26.      mc.setBlocks(pos.x, pos.y, pos.z + 1,
27.                   pos.x, pos.y + 3, pos.z + 7,
28.                   block.GLASS.id)
29.
30.  def splatBlock(mc, x, y, z, team):
31.
32.      pointsScored = [0,0]
33.
34.      # who is the other team?
35.      otherTeam = 1 - team
36.
37.      # what type of block has been hit?
38.      blockHit = mc.getBlockWithData(x, y, z)
39.      # has a glass block been hit?
40.      if blockHit.id == block.GLASS.id:
```

```
41.          # claim it for the team
42.          mc.setBlock(
43.  x, y, z, block.WOOL.id, TEAMCOLS[team])
44.          # increase the team's score
45.          pointsScored[team] += 1
46.
47.      # was it a wool block?
48.      elif blockHit.id == block.WOOL.id:
49.          # if other team's colour turn it back to GLASS
50.          if blockHit.data == TEAMCOLS[otherTeam]:
51.              mc.setBlock(x, y, z, block.GLASS.id)
52.              # reduce the other team's score
53.              pointsScored[otherTeam] -= 1
54.
55.      return pointsScored
56.
57.  # set up points
58.  points = [0,0]
59.
60.  # create connection to Minecraft
61.  mc = Minecraft.create()
62.
63.  # post the message to the screen
64.  mc.postToChat("Minecraft Splat")
65.
66.  # find out the host player's position
67.  pos = mc.player.getTilePos()
68.
69.  # build the pitch
70.  buildPitch(mc, pos)
71.
72.  sleep(3)
73.
74.  mc.postToChat("Go!")
75.
76.  # get a list of the players
77.  players = mc.getPlayerEntityIds()
78.
79.  start = time()
80.
81.  gameOver = False
82.
```

```
83.     # continue till the end of the game
84.     while not gameOver:
85.
86.         # has a block been hit?
87.         blockHits = mc.events.pollBlockHits()
88.         for hit in blockHits:
89.
90.             # which team was it?
91.             team = players.index(hit.entityId) % 2
92.
93.             pointsScored = splatBlock(
94.                 mc, hit.pos.x, hit.pos.y, hit.pos.z, team)
95.
96.             # update the points
97.             points[0] += pointsScored[0]
98.             points[1] += pointsScored[1]
99.
100.            # splat blocks around it
101.            for x in [-1, 0, 1]:
102.                for y in [-1, 0, 1]:
103.                    for z in [-1, 0, 1]:
104.                        if getrandbits(1) == 1:
105.                            pointsScored = splatBlock(mc,
106.                                                hit.pos.x + x,
107.                                                hit.pos.y + y,
108.                                                hit.pos.z + z,
109.                                                team)
110.
111.                            # update the points
112.                            points[0] += pointsScored[0]
113.                            points[1] += pointsScored[1]
114.
115.        # if the time has run out, set game over
116.        if time() - start > 30:
117.            gameOver = True
118.            mc.postToChat("Game Over")
119.            mc.postToChat("Green Team = " + str(points[0]))
120.            mc.postToChat("Red Team = " + str(points[1]))
121.            if points[0] > points[1]:
122.                mc.postToChat("Green Team wins")
123.            else:
124.                mc.postToChat("Red Team wins")
```