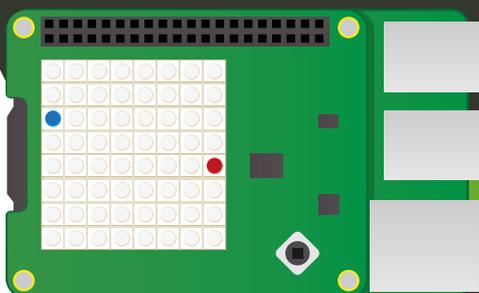


[CHAPTER THREE] WALKING WITH STEVE

Tired of using your fingers to tap keys to move in Minecraft?
Then why not use your wrist instead, and take advantage
of the awesome power of the Sense HAT?



YOU'LL NEED A
SENSE HAT FOR
THIS PROJECT!

[MINECRAFT & SENSE HAT]

The Sense HAT is an add-on board which attaches to the Pi's GPIO pins. It has lots of sensors, such as the humidity sensor and the magnetometer. The sensor we'll be using in this project is the accelerometer. It also has an 8x8 LED matrix.

One of the cool things about the console edition of *Minecraft* is that you can use a controller instead of a keyboard. The Pi edition might sometimes seem a little basic, but you can make your game more like the console edition by deploying your Sense HAT as a tiltable controller, instead of using a keyboard. If you don't know which way to tilt it, the arrows appearing on the LED matrix will help you.

The first thing you need to do is to install all the necessary modules. An obvious one is the Sense HAT library: if you have Raspbian Jessie, this comes bundled with it, but if not, you can install it by typing:

```
sudo pip install sense_hat
```

You'll also need another Python module, which in turn requires the Xlib library:

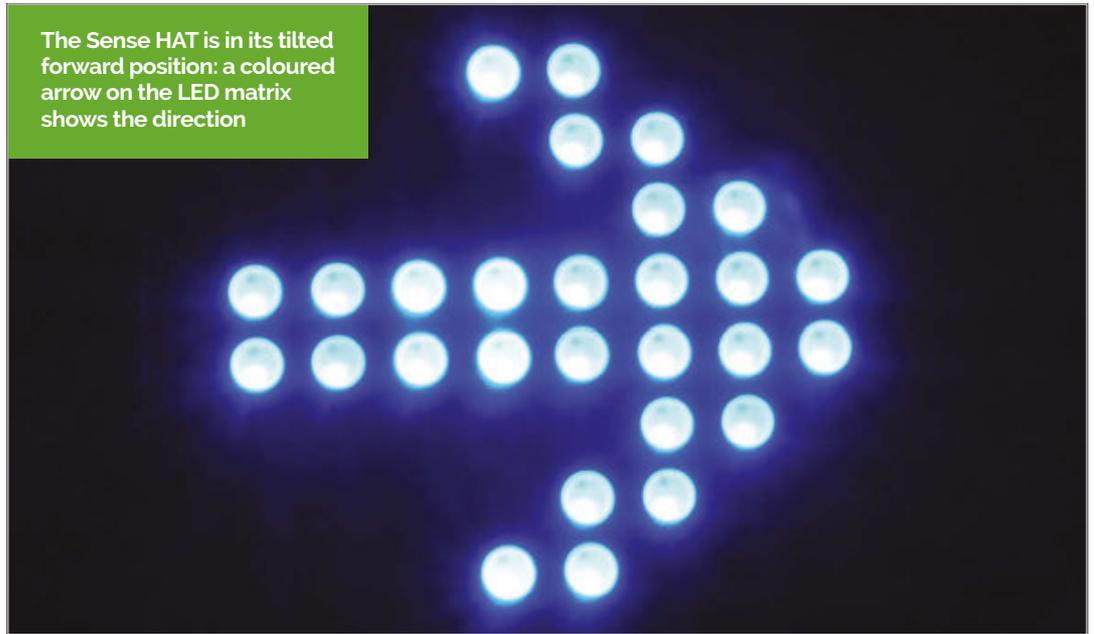
```
sudo apt-get install python-xlib
sudo pip install pyautogui
```

Using the **pyautogui** functions, you can simulate keys as if they were actually pressed. This is how you make Steve walk around his blocky world.

Instead of pressing keys, we'll use the Sense HAT's accelerometer to find out which direction the HAT is being tilted. Each time we measure, we get values representing the acceleration intensity of the x, y, and z axes (in Gs). These are sometimes called roll, pitch, and yaw, like on an aeroplane or a spaceship. We only need the x and y axes, as the z axis is rotation and we're not using that.

[LIGHT THE WAY]

The arrows displayed on the LED matrix while you're moving look really cool, but are actually easy to generate. Using Python, we make a 64-element list containing the arrow shape - you can customise this to make your own shape of arrow - and then simply display it using a different RGB colour value and with a different rotation. The Sense HAT API makes this whole process very simple.



How to use it

Download or type up the code from the listing (right) into IDLE, then press **F5** to run it. Make sure Minecraft is running and you've entered a world when you run it, otherwise lots of errors will appear! If you tilt the Sense HAT forward, the **pyautogui** module will trigger a W key and move Steve forward; make sure your mouse is clicked in the Minecraft window when this happens, otherwise it will just generate a 'w' in the Python shell. The same thing happens when you tilt it backwards, but it will generate an S. If you tilt it to the side, it will generate a D or an A, depending on which direction you've tilted. You still need to use the mouse to look around, and the E key to open your inventory. So, the idea is for you to get all the items you need in your hotbar, then hold the mouse in one hand and the Sense HAT in the other.

Using Minecraft

When the program is running, you'll only be able to walk (or fly) around using the keys when the Sense HAT is in the level position (all the LEDs will be red). Even then, you'll only move if you tap the key repeatedly instead of holding it down. So you're better off sticking to the Sense HAT!

Once you have written and understood this program, you could try to improve it by making the Sense HAT's joystick open your inventory or whatever else you can think of!

WalkingWithSteve.py

Download
magpi.cc/
1Qr7wjw

```

01. from mcpi.minecraft import Minecraft
02. import pyautogui as pag
03. import time
04. from sense_hat import SenseHat
05. sh = SenseHat()
06.
07. # unpresses all the keys
08. def unpress():
09.     for key in ['s','w','a','d']:
10.         pag.keyUp(key)
11.
12. # presses the correct key
13. def move(direction):
14.     unpress()
15.     pag.keyDown(direction)
16.
17. # the arrow
18. def displayArrow(c,rot):
19.     arrow = [
20.         e,e,e,c,c,e,e,e,
21.         e,e,c,c,c,c,e,e,
22.         e,c,c,c,c,c,c,e,
23.         c,c,e,c,c,e,c,c,
24.         c,e,e,c,c,e,e,c,
25.         e,e,e,c,c,e,e,e,
26.         e,e,e,c,c,e,e,e,
27.         e,e,e,c,c,e,e,e]
28.     sh.set_rotation(rot)
29.     sh.set_pixels(arrow)
30.
31. # define the colours
32. r = [255,0,0]
33. e = [0,0,0]
34. g = [0,255,0]

```

```
35. b = [0,0,255]
36. stop = [ # the stop sign
37. r,r,r,r,r,r,r,r,
38. r,r,r,r,r,r,r,r,
39. r,r,r,r,r,r,r,r,
40. r,r,r,r,r,r,r,r,
41. r,r,r,r,r,r,r,r,
42. r,r,r,r,r,r,r,r,
43. r,r,r,r,r,r,r,r,
44. r,r,r,r,r,r,r,r]
45.
46. mot = 'SSSS'
47. while True: # main loop
48.     x, y, z = sh.get_accelerometer_raw().values()
49.     x = round(x, 0)
50.     y = round(y, 0)
51.     if x == -1 and abs(y) == 0 and mot != 'rrrr':
52.         displayArrow(b,0)
53.         move('d') # right
54.         mot = 'rrrr'
55.     elif x == 1 and abs(y) == 0 and mot != 'llll':
56.         displayArrow(b,180)
57.         move('a') # left
58.         mot = 'llll'
59.     elif y == -1 and mot != 'wwww':
60.         displayArrow(g,270)
61.         move('w') # fwd
62.         mot = 'wwww'
63.     elif y == 1 and mot != 'bbbb':
64.         displayArrow(g,90)
65.         move('s') # back
66.         mot = 'bbbb'
67.     elif abs(x) == 0 and abs(y) == 0:
68.         unpress() # stop
69.         sh.set_pixels(stop)
70.         mot = 'SSSS'
```