Minecraft Hide and Seek

This game comes thanks to Stuff About Code (www.stuffaboutcode.com).

The concept is really simple; A diamond block is hidden at a random location in the Minecraft world and you have to find it and stand next to it in the quickest time. You are helped along the way by the game telling you whether you are getting "warmer" or "colder" and how far you are from the block.

1 *Setting up Minecraft*

Before doing anything else, lets make sure that we have Minecraft working. If it is already set up there should be a directory called **mcpi** in your home directory. If there isn't, you will need to get the package called **minecraft-pi-0.1.1.tar.gz** and unpack it. When you have it, use this to unpack it:

```
pi@raspberry~ $ tar -xzvf minecraft-pi-0.1.1.tar.gz
```

You should now have the **mcpi** directory and you can run minecraft with:

```
pi@raspberry~ $ ./mcpi/minecraft-pi &
```

When you have done this start a new game and then go back to the terminal window.

As well as the Minecraft program itself, there is a *module* for Python (a special set of functions) that allow you to interact with the Minecraft game. We need to put this somewhere where Python can find it. First, create a new directory to put your code in:

```
pi@raspberry~ $ mkdir mc-code
```

Then copy the module into this new directory

```
pi@raspberry~ $ cp -r mcpi/api/python/mcpi mc-code/minecraft
```

'cp' is the command to copy a file from one place to another and the '-r' tells it to copy a whole directory.

In Windows, files inside directories are written like **mcpi\api**. In Linux, the '/' character is used instead of '\' but otherwise it is exactly the same, so **mcpi/api/python/mcpi** is just a long line of directories inside each other.

Change into the new mc-code directory and make sure that it the minecraft directory inside it.

```
pi@raspberry~ $ cd mc-code
pi@raspberry~/mc-code $ ls
```

Make sure that you can see the **minecraft** directory – this will mean that when we have lines starting with 'import minecraft' in the code then Python will know where to find it.

If the minecraft directory is there then we can start editing the Hide and Seek game.

```
pi@raspberry~/mc-code $ nano hideandseek.py
```

2 Where am I?

The first thing we need to do is find where the player is.

```
# www.stuffaboutcode.com
# Raspberry Pi, Minecraft – hide and seek

# import the minecraft.py module from the minecraft directory
import minecraft.minecraft as minecraft
# import the time module, so delays can be used
import time

if __name__ == “__main__”:

    # Connect to minecraft by creating the minecraft object
    # - minecraft needs to be running and in a game
    mc = minecraft.Minecraft.create()

    # Post a message to the minecraft chat window
    mc.postToChat(“Hi, Minecraft Hide & Seek”)

    time.sleep(2)

    # Find the player's position
    playerPos = mc.player.getPos()
    print playerPos
```

Now save the file and see what happens. If you haven't done so already, start Minecraft and start a game. Then try:

```
pi@raspberry~/mc-code $ python hideandseek.py
```

This should print a message to the Minecraft game and then print the position of the player to the screen. *Challenge: Can you get it to print the players position to the Minecraft window instead?*

3 Hiding the diamond

The next job is to find a position for the hidden block and put a diamond block there. We need to use two new modules, one for choosing random numbers and one for working with the minecraft blocks. This is done with two new 'import' lines.

To choose the random position we use the `randrange` function from the 'random' module. This function chooses a random number between the two numbers we put in the brackets. You may have noticed that the player position, printed out in the last section, was written in long decimal numbers. For choosing a random position we want to use whole numbers so we are going to create a function called `roundVec3` that changes the numbers in the players position to whole numbers.

Finally, we are going to place the diamond block in the position we have chosen. This is done with the `setBlock` function.

Remember, the **bold** text shows you what new code needs to be added.

```
# www.stuffaboutcode.com
# Raspberry Pi, Minecraft – hide and seek

# import the minecraft.py module from the minecraft directory
import minecraft.minecraft as minecraft
# import minecraft block module
import minecraft.block as block
# import the time module, so delays can be used
import time
# import random module to create random numbers
import random

# function to round players float positions to integer position
def roundVec3(vec3):
    return minecraft.Vec3(int(vec3.x), int(vec3.y), int(vec3.z))

if __name__ == “__main__”:

...

    # Find the player's position
    playerPos = mc.player.getPos()
    print playerPos

    # create random position within 50 blocks from the player
    # our hidden block will go there
    randomBlockPos = roundVec3(playerPos)
    randomBlockPos.x = random.randrange( randomBlockPos.x – 50,
                                         randomBlockPos.x + 50 )
    randomBlockPos.y = random.randrange( randomBlockPos.y – 5,
                                         randomBlockPos.y + 5 )
    randomBlockPos.z = random.randrange( randomBlockPos.z – 50,
                                         randomBlockPos.z + 50 )
    print randomBlockPos

    # Create hidden diamond block
    mc.setBlock(randomBlockPos.x,
                randomBlockPos.y,
                randomBlockPos.z,
                block.DIAMOND_BLOCK)
    mc.postToChat(“A diamond has been hidden – go find!”)
```

When you run this you should see the position of the player displayed as three decimal numbers (with a decimal point) and then the position that has been chosen for the hidden block as three whole numbers. In computing, whole numbers like 1, 2 and 3 are called *integers* and decimal numbers like 1.3, 2.9 and 3.14159 are called *floating point numbers* or *floats*.

4 *Help me find it!*

At the moment, our code is going to hide the diamond but then will not help us find it. Now we will use a loop to get it to tell the player how close the diamond is. For this we need a function that calculates the distance from the player to the diamond. Fortunately, a clever mathematician called Pythagoras found out how to do this a long time ago and we will use his formula in the function called 'distanceBetweenPoints'.

We are going to go keep going round the loop until the diamond is found and ever two seconds we will put a message to the screen to say if the player is getting closer (warmer) or further away (colder).

```
...

# import random module to create random numbers
import random
# import math module to use the square root function
import math

# function to round players float positions to integer position
def roundVec3(vec3):
    return minecraft.Vec3(int(vec3.x), int(vec3.y), int(vec3.z))

def distanceBetweenPoints(point1, point2):
    xd = point2.x – point1.x
    yd = point2.y – point1.y
    zd = point2.z – point1.z
    return math.sqrt((xd*xd) + (yd*yd) + (zd*zd))

...

    # Create hidden diamond block
    mc.setBlock(randomBlockPos.x,
                randomBlockPos.y,
                randomBlockPos.z,
                block.DIAMOND_BLOCK)
    mc.postToChat(“A diamond has been hidden – go find!”)

    # Start hide and seek
    seeking = True
    lastPlayerPos = playerPos
    lastDistanceFromBlock \
            = distanceBetweenPoints(randomBlockPos,
                                    lastPlayerPos)
    timeStarted = time.time()
    while ( seeking == True ):
        # Get players position
        playerPos = mc.player.getPos()
        # Has the player moved
        if lastPlayerPos != playerPos:
            distanceFromBlock \
                    = distanceBetweenPoints(randomBlockPos,
                                            playerPos)
            if distanceFromBlock < 2:
                # found it!
                seeking = False
            else:
                if distanceFromBlock < lastDistanceFromBlock:
                    mc.postToChat(“Warmer: “
                                  + str(int(distanceFromBlock))
                                  + “ blocks away”)
                if distanceFromBlock > lastDistanceFromBlock:
                    mc.postToChat(“Colder: “
                                  + str(int(distanceFromBlock))
                                  + “ blocks away”)
            lastDistanceFromBlock = distanceFromBlock
        time.sleep(2)
    timeTaken = time.time() - timeStarted
    mc.postToChat(“Well done - “
                  + str(int(timeTaken))
                  + “ seconds to find the diamond”)

    time.sleep(5)

    mc.postToChat(“www.stuffaboutcode.com”)
```

Phew! That's a lot of code. Try it now to see if it works. There may be mistakes that you need to go and fix and Python will tell you on which lines to find them. Once you have all the code correct, happy hunting!

5 Challenges

- How would you change the diamond block into something else? Hint, try GOLD_BLOCK, FLOWER_YELLOW or GLASS. (There is a list of options at http://www.stuffaboutcode.com/p/minecraft-api-reference.html)
- When you find the diamond block, can you change it into something else (STONE_SLAB, for example) and then hide another diamond?

6 Code listing

Here is the whole code in one place.

```
# www.stuffaboutcode.com
# Raspberry Pi, Minecraft – hide and seek

# import the minecraft.py module from the minecraft directory
import minecraft.minecraft as minecraft
# import minecraft block module
import minecraft.block as block
# import the time module, so delays can be used
import time
# import random module to create random numbers
import random
# import math module to use the square root function
import math

# function to round players float positions to integer position
def roundVec3(vec3):
    return minecraft.Vec3(int(vec3.x), int(vec3.y), int(vec3.z))

def distanceBetweenPoints(point1, point2):
    xd = point2.x – point1.x
    yd = point2.y – point1.y
    zd = point2.z – point1.z
    return math.sqrt((xd*xd) + (yd*yd) + (zd*zd))

if __name__ == “__main__”:

    # Connect to minecraft by creating the minecraft object
    # - minecraft needs to be running and in a game
    mc = minecraft.Minecraft.create()

    # Post a message to the minecraft chat window
    mc.postToChat(“Hi, Minecraft Hide & Seek”)

    time.sleep(2)

    # Find the player's position
    playerPos = mc.player.getPos()
    print playerPos
    # create random position within 50 blocks from the player
    # our hidden block will go there
    randomBlockPos = roundVec3(playerPos)
    randomBlockPos.x = random.randrange( randomBlockPos.x – 50,
                                         randomBlockPos.x + 50 )
    randomBlockPos.y = random.randrange( randomBlockPos.y – 5,
                                         randomBlockPos.y + 5 )
    randomBlockPos.z = random.randrange( randomBlockPos.z – 50,
                                         randomBlockPos.z + 50 )
    print randomBlockPos

    # Create hidden diamond block
    mc.setBlock(randomBlockPos.x,
                randomBlockPos.y,
                randomBlockPos.z,
                block.DIAMOND_BLOCK)
    mc.postToChat(“A diamond has been hidden – go find!”)

    # Start hide and seek
    seeking = True
    lastPlayerPos = playerPos
    lastDistanceFromBlock \
            = distanceBetweenPoints(randomBlockPos,
                                    lastPlayerPos)
    timeStarted = time.time()
    while ( seeking == True ):
        # Get players position
        playerPos = mc.player.getPos()
        # Has the player moved
        if lastPlayerPos != playerPos:
            distanceFromBlock \
                    = distanceBetweenPoints(randomBlockPos,
                                            playerPos)
            if distanceFromBlock < 2:
                # found it!
                seeking = False
            else:
                if distanceFromBlock < lastDistanceFromBlock:
                    mc.postToChat(“Warmer: “
                                  + str(int(distanceFromBlock))
                                  + “ blocks away”)
                if distanceFromBlock > lastDistanceFromBlock:
                    mc.postToChat(“Colder: “
                                  + str(int(distanceFromBlock))
                                  + “ blocks away”)
            lastDistanceFromBlock = distanceFromBlock
        time.sleep(2)
    timeTaken = time.time() - timeStarted
    mc.postToChat(“Well done - “
                  + str(int(timeTaken))
                  + “ seconds to find the diamond”)

    time.sleep(5)

    mc.postToChat(“www.stuffaboutcode.com”)
```