**Raspberry Pi Activity 5**

*Learning Objective:* Make changes in Minecraft Pi using Python code

Minecraft is a computer game that allows you to build a computer world using virtual building blocks. This lesson plan assumes that Minecraft Pi is already installed on the Raspbian operating system.

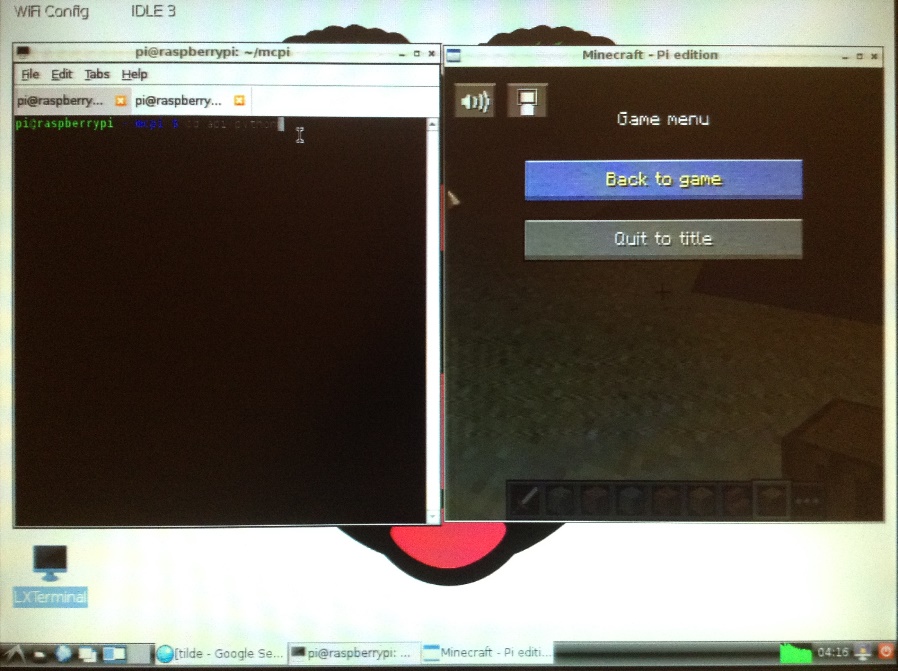
***Task 1.***Write a message in Minecraft. Open up the LXTerminal by double-clicking the icon on the desktop.

To run Minecraft Pi, type these lines in the LXTerminal:  
  
**cd mcpi**

**./minecraft-pi**  
  
The cd command changes directories or folders from the command line. You use it here to move from **/home/pi** into the directory **/home/pi/mcpi** where the **minecraft-pi** game is located. Run the command **./minecraft-pi** to start the game.

When Minecraft Pi has loaded, click on Start Game and select a world from the list that is displayed or click Create New to enter build mode.

Navigate back to the LXTerminal window with your mouse, and open a new tab by clicking File 🡪New Tab. You need an open tab so that you can type commands into LXTerminal while Minecraft Pi is running. Remember to use the tab button to switch between the Minecraft Pi terminal and the LXTerminal.



To make sure that you are in the right directory to begin writing your code, type this command in your LXTerminal tab window:

**cd api/python**

To create your first Minecraft Pi program, open the nano text editor by typing this command:

**nano testmcpi.py**

This opens a text editor from the command line.

Type the following code into the nano text editor file:  
 **import mcpi.minecraft as minecraft**

Here you are importing a module that you need in your program – in this case, the minecraft module. Now type the following command:

**mc=minecraft.Minecraft.create()**

This line connects your program to Minecraft and allows you to start interacting with it. (You must have Minecraft running and be in a game for the program to work).

Next, create a message string using the following code:

**msg="Hello Minecraft World!"**

Then type the following message to post your message to the Minecraft chat window:

**mc.postToChat(msg)**

Press CTRL + X to exit, and press Y (for yes) to save what you have written. The message File name to write: testmcpi.py will appear. Press Enter to confirm that you want to write to this file. You will then be returned to the command line in LXTerminal.

(In a new tab) Now run in your script by typing the following line in the LXTerminal tab window:

**python testmcpi.py**

You will see your message displayed in the Minecraft game window open on your screen.  
***Task 2.***For this task you will find out the current coordinates of your player and transport him to a different location by changing the coordinates.  
  
(In a new LXTerminal window) Type the following commands to create a new nano text editor file called location.py:

**cd mcpi/api/python  
nano location.py**  
  
(In the new text file called location.py) Import the modules that you will need in this program by typing the following code:

**import mcpi.minecraft as minecraft  
import time**  
  
**mc = minecraft.Minecraft.create()  
  
time.sleep(1)  
pos = mc.player.getPos()**  
  
The last line gives you the position of your Minecraft player. Next, you want to display that information in the Minecraft window so that you can see it in the game. To do that, type the following code:

**mc.postToChat("You are located x=" +str(pos.x) + ", y=" +str(pos.y) +", z=" +st$**

pos.x gives the x coordinate, pos.y gives the y coordinate, and pos.z gives the z coordinate.

Changing your location in minecraft:

Now you can try to change the player’s position in Minecraft. Add the following lines of code to the end of your program:

**time.sleep(2)  
mc.postToChat("I'm flying in the sky!")  
  
time.sleep(5)  
mc.player.setPos(pos.x, pos.y + 60, pos.z)**

When you run this code, the player will suddenly change position. The last line of code here adds 60 only to the y axis (up and down), transporting the player to the middle of the sky.  
  
  
***Task 3.***By using a version of Minecraft that you can manipulate with code, you don’t need to spend much time building a structure - you can simply program it to be built.

In the LXTerminal window, type the following commands to create a new nano text editor file called placeblock.py:

**cd mcpi/api/python  
nano placeblock.py**  
  
You need to import the block module from Minecraft Pi that allows you to use the Minecraft block names. Type the following code inti the nano text editor window:  
  
**import mcpi.minecraft as minecraft  
import mcpi.block as block  
mc = minecraft.Minecraft.create()  
  
mc.setBlock(1, 10, 1, block.WOOD)**  
  
The three numbers in the last line refer to the x, y and z coordinates. This is followed by the block type you wish to use – WOOD. The result in Minecraft Pi will be a single block of wood hanging in the sky above the player.

Press CTRL + X to exit nano, ensuring that you press y and Enter to save your work. Test your code to see what happens.

Challenge: See if you can navigate to a new location, make a note of the coordinates and then place some different blocks using the setBlock command.

***Task 4.***With the addition of one letter, you can place more than one block at a time.

The setBlocks command (note the added s at the end) works in the same way as setBlock. You start with a set of x, y and z coordinates that indicate where in Minecraft you would like the blocks to be placed. Then the second set of x, y and z coordinates represent the number of blocks needed to make the shape that you want to create:

setBlocks (x1, y1, z1, x2, y2, z2, blocktype)  
  
In the LXTerminal window, type the following code to create a new nano text editor file called placeblocks.py:

**cd mcpi/api/python  
nano placeblocks.py**

Import the modules that you will need:

**import mcpi.minecraft as minecraft  
import mcpi.block as block**

Set up the connection to Minecraft and post a message to the screen:

**mc = minecraft.Minecraft.create()**

It makes more sense to place blocks where the player is located, so that you have an empty space for the cube. To do this, you need to get the player’s position using the following code:  
  
**pos = mc.player.getTilePos()**

You can then use setBlocks, adding the coordinates of the player’s position to indicate where you would like the blocks to be placed:

**mc.setBlocks(pos.x, pos.y, pos.z, pos.x+13, pos.y+2, pos.z+5, block.WOOD)**

Try different modifications to the code to see what happens, e.g:   
  
**mc.setBlocks(pos.x + 5, pos.y + 5, pos.z, pos.x + 10, pos.y + 10, pos.z + 10, block.STONE)**

***Task 5.***It can take a long time to move from one side of the Minecraft world to the other. You can speed things up by creating a diamond block transporter to teleport you from one location to another. To do this, you will use setBlock, getPos and setPos.

In the LXTerminal window, type the following code to create a new nano text editor file called teleporter.py:  
  
**cd mcpi/api/python  
nano teleporter.py**

Import the modules that you will need:

**import mcpi.minecraft as minecraft  
import mcpi.block as block  
import time**

Set up the connection to Minecraft and post a message to the screen:

**mc = minecraft.Minecraft.create()  
mc.postToChat("A Teleportation Experience")  
time.sleep(5)**

The time delays are important in this code, so that you can move the player around the Minecraft world before the transporter blocks are placed.

Place a diamond block as the first teleporter location, underneath the player’s position:

**teleporter1 = mc.player.getTilePos()  
mc.setBlock(teleporter1.x, teleporter1.y - 1, teleporter1.z,  
block.DIAMOND\_BLOCK)  
mc.postToChat("1st Teleporter created")  
time.sleep(2)**

Display a message on the screen to tell the player to move and find the second location where he wants to be able to transport to and from:

**mc.postToChat("Find another location in 30 seconds")  
time.sleep(30)**  
  
**teleporter2 = mc.player.getTilePos()  
mc.setBlock(teleporter2.x, teleporter2.y -1, teleporter2.z, block.DIAMOND\_BLOCK)  
mc.postToChat("2nd Teleporter created")  
time.sleep(2)**

Create a while true loop to continually check the player’s position. If the player is positioned on the first transporter diamond block location, her location will be changed to where the second diamond block is located. If the player is positioned on the second teleporter diamond block location, his location will be changed to where the first diamond block is located.

**while True:  
  time.sleep(1)  
  pos = mc.player.getTile.Pos()  
  
  if(pos.x == teleporter1.x) and (pos.y == teleporter1.y) and (pos.z == teleporter1.z):  
        mc.player.setPos(teleporter2.x, teleporter2.y, teleporter2.z)  
  if(pos.x == teleporter.x) and (pos.y == teleporter2.y) and (pos.z == teleporter2.z):  
        mc.player.setPos(teleporter1.x, teleporter1.y, teleporter1.z)**