Minecraft Pi Edition

What is it?

Minecraft Pi Edition is a cut down version of the full Minecraft for the Raspberry Pi. Not only that but it has a programming interface which allows you to manipulate your world with a few lines of code. No more place block, move, place block, move for you!

Table of Contents

Getting Started
Task 1 - Just Another Brick in the Wall
Task 2 - Beam Me Up Scotty
Task 3 - I'm Walking On Sunshine (woah-oh)
Project 1 - Treasure Hunt
Project 2 - Piano

Minecraft Pi Edition

Let's have a Block Party!

Step 1 - Power up and log on (probably done already) 1. If it's not already running plug the micro-USB (phone charger) lead into the connector on the board 2. If prompted for a login use ***pi*** as the user and ***raspberry*** as the password, don't panic; you won't see the password as you type, just press the Enter key when you're done 3. Type ***startx*** and press Enter 4. The "magic" will start to happen and eventually you'll get to a desktop with a giant raspberry on it. Congratulations, Step 1 complete!	**Step 2 - Setup Minecraft** 1. Double click the ***LXTerminal*** icon - a small black console window will appear 2. Create a folder for your code - we'll use the current time so we don't clash with others: a. At the prompt type ***mkdir*** a space and the time, so ***mkdir 1030*** or ***mkdir 1451***. Don't forget to press Enter b. Now type ***cd*** and the time you just entered (i.e. ***cd 1030***) and press Enter c. Finally, type the following, then press Enter **cp -r ~/mcpi/api/python/* .** *(that's '/*' space '.')* 3. You should now be ready to code
Step 3 - Open IDLE (aka getting your Geek on) 1. Double click on the IDLE icon on the desktop 2. From the *File* menu select *New Window* - a blank white window will appear 3. Aaaand breathe - you're ready to code	**Step 4 - Start Minecraft** 1. Open a new ***LXTerminal*** window and type the following **cd ~/mcpi** **./minecraft-pi** 2. Minecraft should start, now create a new world To release the mouse and keyboard back to the desktop press the TAB key (usually left hand side, above the Caps Lock)

There's a lot of steps to start you off. You should now have Minecraft running in a window and two windows from IDLE. If not, check you've done all four steps before asking for help! It's probably worth moving the Minecraft window into one corner and move/resize the IDLE windows so it all fits on screen. The Minecraft window will always sit over the top of everything else!

Go for a wander

If this is your first time in Minecraft you'll need to have a quick run down of the basics:

'W' = forward,
'A' = left,
'S' = back,
'D' = right,
space = jump,
shift = duck,
'E' = inventory
Esc = back to menu

mouse = look around
left mouse = place block,
right mouse = destroy block

If you double tap the space bar (space - space quickly) your character will fly! At this point
space = fly up
shift = fly down

A double tap of the space bar again returns you to 'normal' mode.

Once you're familiar with your surroundings, let's crack on.

Minecraft Pi Edition

Task 1 - Just Another Brick in the Wall

To start off with, let's get the basics down and build something. Make sure you're not in the water then enter the following:

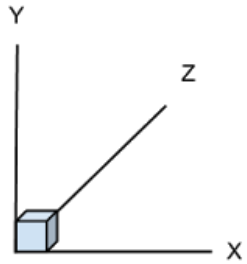
`import mcpi.minecraft as minecraft`	The Minecraft for Pi comes with it's own set of commands, this tells Python we want to use them.
`mc = minecraft.Minecraft.create()`	This creates as a Minecraft object called `mc` which we can then perform actions on in the Minecraft world

`x = <check Minecraft>`
`y = <check Minecraft>`
`z = <check Minecraft>`
	If you look in the top left of your Minecraft window you will see 3 numbers (i.e. *10, 25.5, 9*), these represent your x, y, and z coordinates in the world. Put those numbers in here. Add 10 to the y coordinate before you do. **Write them down here, we'll need them in Task 2!**

`mc.setBlock(x, y, z, 46)`	This creates a block type of 46 at those coordinates.

Before we get too far let's check it works. *File - Save* (or CTRL+S) and give it a filename of ***task1.py***. Now it's saved press F5; go into Minecraft and look up! Now you know what block number 46 is :-) Go back the the code (press TAB to get out of Minecraft), change the block number and run it again to see what happens.

What about changing the x, y and/or z? The image below might help. When you're done lets move on



Task 1 - continued

We can place blocks one at a time, but that's not much better than in the game itself (although probably quicker!). Let's build a box, first find a flat area in the game so we have a clear area to build on. Then enter the following code:

`import mcpi.minecraft as minecraft`
`mc = minecraft.Minecraft.create()`
 	
`player = mc.player.getTilePos()`
`x = player.x + 5`
`y = player.y`
`z = player.z + 5`
	This time we're going to use `player.getTilePos()` as this will tell us where our character is currently standing.

`x2 = x + 10`
`y2 = y + 5`
`z2 = z + 8`
 	
`mc.setBlocks(x, y, z, x2, y2, z2, 4)`
	x2, y2, and z2 are the coordinates of the opposite corner of our box

`mc.setBlocks(x + 1, y + 1, z + 1, x2 - 1, y2 - 1, z2 - 1, 0)`	There's no deleteBlock command. Instead we just set it to 0 = Air

Save and run; hopefully a building just appeared in front of you. Using the right mouse button break a hole in the wall and see inside. Now see if you can modify the code to build one with a window a window (glass is block type 20) and fill it with water (block 8)

Task 2 - Beam Me Up Scotty!

Not only can we place blocks but we can also move ourselves from place to place

`import time`
`import mcpi.minecraft as minecraft`
`mc = minecraft.Minecraft.create()`
 	
`player = mc.player.getTilePos()`
`x = player.x`
`y = player.y + 100`
`z = player.z`
	Note we're importing `time`, you'll see why in a second

`mc.postToChat('Look down...')`
`time.sleep(5)`
`mc.postToChat('Geronimo!')`
	`postToChat` allows us to display messages on screen
	`sleep` tells the program to wait for a period of time, in seconds

`mc.player.setPos(x, y, z)`	"It's not flying, just falling with style" - Buzz Lightyear

Save this as ***task2.py*** and run it. I hope you had a parachute!

Now change x, y, and z to the position you used in the very first task for that block of TNT (you did write them down didn't you….). Make sure you increase the y value by 1 more so we don't appear IN the TNT block. You can change the messages if you like, but no need.

Assuming that all worked, you have successfully transported yourself a short distance across the Minecraft world. How about always being able to return home? Let's go make ourselves a personal transporter.

Either use the coordinates of the TNT block, or (if you know where it is) go find one of the buildings you made earlier, walk inside a make a note of the coordinates. Let's get coding...

`import time`
`import mcpi.minecraft as minecraft`
`import mcpi.block as block`
`mc = minecraft.Minecraft.create()`
	Now we're importing `block` as well. This allows us to refer to the block by name rather than its number

`xHome = <your x coord.>`
`yHome = <your y coord.>`
`zHome = <your z coord.>`
 	
`while True:`
	The while statement allows us to loop round the same piece of code over and over again. This is very useful if you want to keep performing a task, like checking where the player is standing….

`    player = mc.player.getTilePos()` `    blockType = mc.getBlock(player.x, player.y - 1, player.z)`	`getBlock` allows us to identify the block that exists at specific coordinates; i.e the one below us
`    if blockType == block.MELON.id:` `        mc.player.setPos(xHome, yHome, zHome)`	Here we compare the block we are standing on with the `MELON` block. If they match we move our player
`    time.sleep(0.1)`	This adds a little delay so we don't check too quickly

Save this as ***teleport.py*** and run it. Nothing will happen at the start, so move away from your "home" and the place a melon block. To do this:

- press 'E' on the keyboard
- use the W,A,S,D keys to move to the melon
    - the melon is one up and three left from the bottom right corner
- press Enter
- use the right mouse button to place the block

Step on the block and POP! you're back at your home, move off and try again. When you're done testing/playing, switch the the *Python Shell* window and press CTRL+C to stop the program.

Task 3 - I'm Walking On Sunshine (woah-oh)

Up to now we've placed blocks, built things, and moved our player around the Minecraft world. Now let's track our movements and stop ourselves from falling.

`import time`
`import mcpi.minecraft as minecraft`
`import mcpi.block as block`
`mc = minecraft.Minecraft.create()`	

`while True:`
`    player = mc.player.getTilePos()`
`    x = player.x`
`    y = player.y - 1`
`    z = player.z`
`    blockType = mc.getBlock(x, y, z)`
	`getBlock` allows us to identify the block that exists at specific coordinates; i.e the one below us

`    if blockType == block.AIR.id:` `        mc.player.setBlock(x, y, z, block.WOOL.id, 4)`	Now we check for `AIR` beneath the player and change it to `WOOL` (`4` denotes yellow)
`    time.sleep(0.05)`	This adds a little delay so we don't check too quickly

Save the code as ***bridge.py***, run it, then go take a walk off of a hill side. Occasionally you'll find it drops you down a level, this is just a limitation of the code due to the sleep function. Now see if you can make a bridge of `ICE` on `WATER`?

Achievement Unlocked!

Assuming everything up to this point has gone according to plan good job, you've learnt a bit of Python, some basic coding (`if` and `while`), and hacked Minecraft! Below are two more projects to try out, but if you've had an idea of what to code next then go nuts; I look forward to seeing Gilwell Park built in Minecraft!!

Project 1 - Treasure Hunt

`import time`
`import mcpi.minecraft as minecraft`
`import mcpi.block as block`
`import random`
`import math`
`mc = minecraft.Minecraft.create()`	

`safeBlockFound = False`
`while safeBlockFound == False:`
`    x = random.randint(-100,100)`
`    z = random.randint(-100,100)`
`    y = mc.getHeight(x, z)`
`    topBlock = mc.getBlock(x,y - 1,z)`
`    if topBlock != block.WATER.id:`
`        safeBlockFound = True`
	This piece of code finds a random block in a 200x200 square.
	• `getHeight` finds the highest block at that location
	• Then we check to make sure this is not on water (as that would be silly!)

`mc.setBlock(x, y, z, block.CHEST.id)`	create our treasure chest
`found == False` `while found == False:`	Loop until we find our treasure chest

`    player = mc.player.getTilePos()`
`    xDist = x - player.x`
`    yDist = y - player.y`
`    zDist = z - player.z`
	calculate our distance from the chest in all three planes (x, y, and z)

`    vDist = math.sqrt(xDist**2 + zDist**2)`	Use some basic trigonometry to work out our straight line distance to the chest (Pythagoras was a clever chap)

`    if vDist <= 1:`
`        mc.postToChat('You found it!')`
`        time.sleep(2)`
`        mc.setBlock(x, y, z, block.AIR.id)`
`        found == True`
	If we're standing next to or on the box - we win! We'll make the box disappear for good measure then exit the loop (setting `found = True`)

`    else:`
`        mc.postToChat('Distance: ' + str(math.floor(vDist)))`
`        time.sleep(5)`
	Otherwise show in the message window how we're getting on

`mc.postToChat('Game over')`	All done!

Hint: This might be easier if you're in fly mode

Extension tasks: Once you found the box, regenerate it and start again
Rather than stand next to the chest to win, check if the player has destroyed it (hit with the LEFT mouse button)

Project 2 - Minecraft Piano

Firstly, close the Python windows (the ones you've been coding in and the *Python Shell*). In the terminal window you typed ***idle*** earlier type ***sudo idle***. This puts us into a more 'privileged' position and will allow us to make use of the buzzer on the Pibrella! Create a new code window via *File - New Window* and let's get coding:

`import time`
`import mcpi.minecraft as minecraft`
`import mcpi.block as block`
`import pibrella`
`mc = minecraft.Minecraft.create()`	

`notes = [-4, -2, 0, 2, 3, 5, 7, 8, 10, 12, 14]`
`player = mc.player.getTilePos()`
`x = player.x`
`y = player.y`
`z = player.z`	

`colour = 0`
`for note in notes:`
`  mc.player.setBlocks(x, y, z, x+2, y, z, block.WOOL.id, color)`
	for each `note`, add a `WOOL` block of a new colour that is 2x1x1

`while True:` `  blockHits = mc.events.pollBlockHits()`	`pollBlockHits` returns a list of the blocks that have been hit

`  if blockHits:`
`  for blockHit in blockHits:`
`  xBlock = blockHit.pos.x`
`  yBlock = blockHit.pos.y`
`  zBlock = blockHit.pos.z`
`  blockType = mc.getBlockWithData(xBlock, yBlock, zBlock)`
	for each block that has been hit get some information about with `getBlockWithData`.

`if blockType.id == block.WOOL.id:`
`  note = notes[blockType.data]`
`  pibrella.buzzer.note(note)`
`  time.sleep(0.2)`
`  pibrella.buzzer.off()`
	if the `blockType.id` matches `WOOL` then get its colour (`data`). The colour number matches the note position in our list of `notes`.

`time.sleep(0.1)`	This adds a little delay so we don't check too quickly

Firstly, switch to Minecraft and find yourself a flat piece of land with a bit of space.
Save your code as ***piano.py*** and run it. In Minecraft you should now have a multicolour piano! Go up to a block and use the RIGHT mouse button. The buzzer should sound. Try another, a different note? Play on maestro!

Acknowledgements

Raspberry Pi and the Raspberry Pi logo are trademarks owned by the Raspberry Pi Foundation (http://www.raspberrypi.org)
"Minecraft" is a trademark of Notch Development AB (http://www.minecraft.com & http://pi.minecraft.com)
The tasks are based on work by Martin O'Hanlon (http://www.stuffaboutcode.com) and Craig Richardson (http://arghbox.wordpress.com)
Treasure Hunt based on work by Chris Reynolds (https://github.com/chrisreynolds/treasure-hunt)
Piano based on work by Martin O'Hanlon (http://www.stuffaboutcode.com)