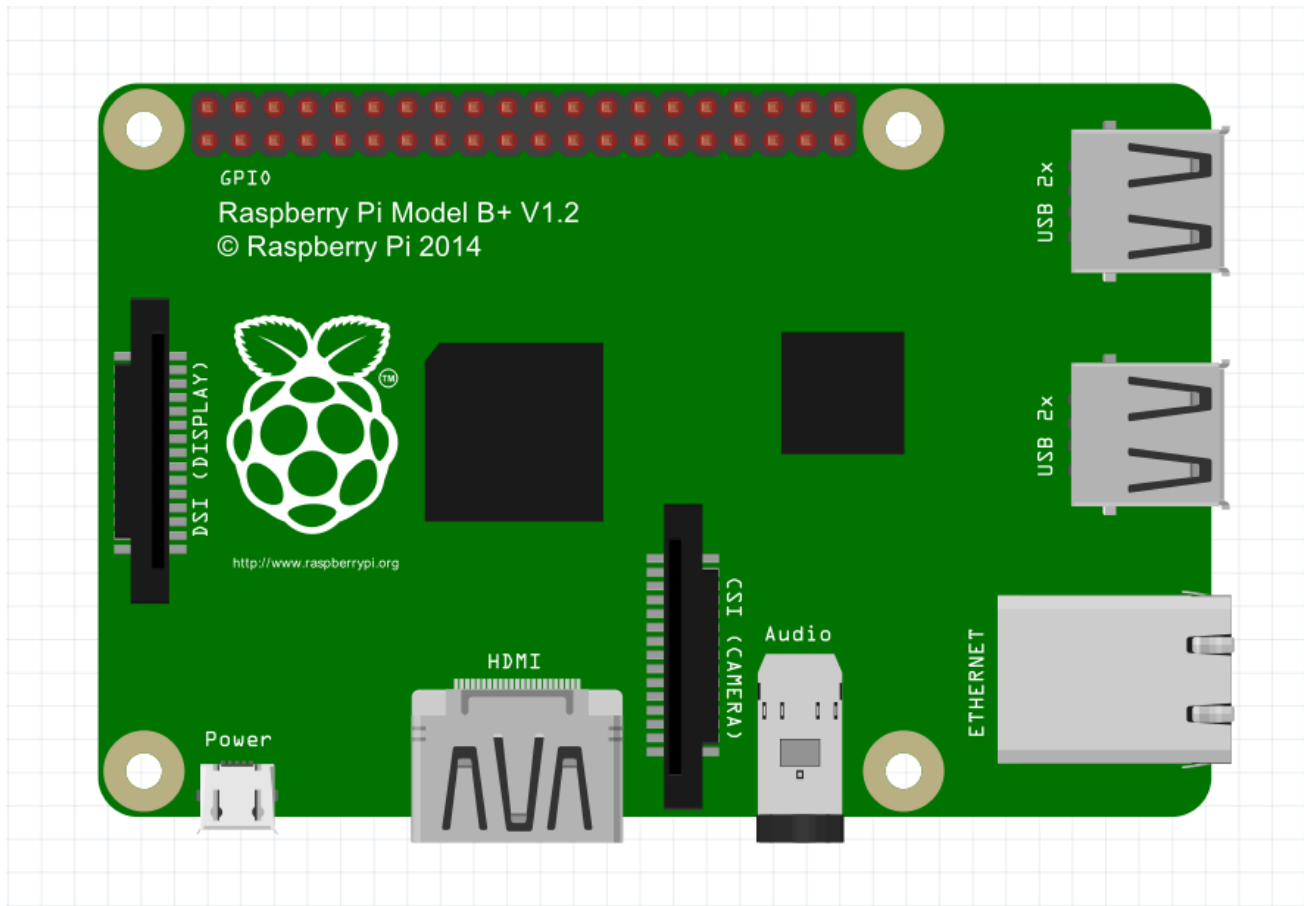




uOttawa and Carleton 2015

# Tutorial Time with a Raspberry Pi



# Activity 1: Raspberry Pi & Camera Module

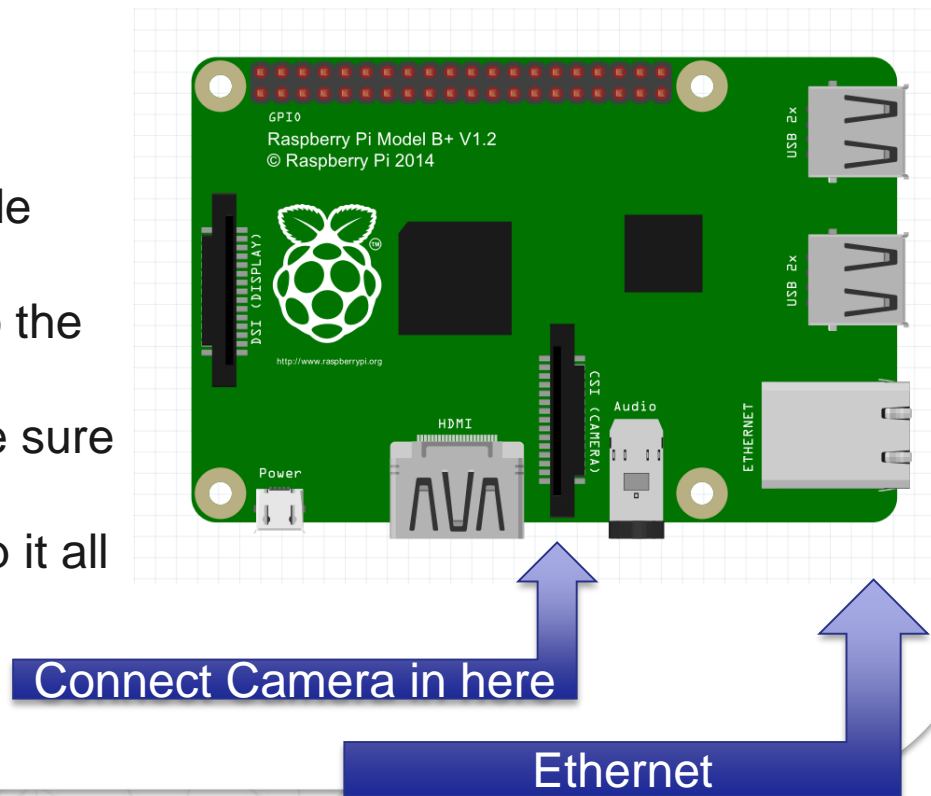
**Before you start:** Please use **caution** when handling the Camera Module with the Raspberry Pi. Cameras are static sensitive. Earth yourself before handling the hardware: hold the USB port on the Pi to ground yourself to the board.

# Activity 1: Raspberry Pi & Camera Module

## Step 1 – Attach the Camera to the board

While the board is **off**

- On the Pi, slide the two side tabs on the connector up.
- Insert the ribbon cable into the slot, blue side towards the Ethernet connection, make sure it is in straight!
- Press down the tabs to clip it all in place.



# Activity 1: Raspberry Pi & Camera Module

## Step 2 – Set up the Camera Software

- First you might have to enable the camera support
- In command line write (without the “>”, which means code you can copy on the command line):  
> sudo raspi-config
- Use your mouse to move to the camera option, select enable and follow instructions to reboot your pi.

# Activity 1: Raspberry Pi & Camera Module

## Step 3 – Test the System

- Pick up the camera, it's time to test it!
- In command line write the following, and get ready to say cheese:  
> `raspistill -v -o test.jpg`

The display should show a 5 sec preview from the camera and then take a picture.

- *Raspistill* is the program used to take pictures. *test* is the name of the picture you just took. The pi saved it as a jpeg to your SD card!

# Activity 1: Raspberry Pi & Camera Module

## Step 4 – Picture time!

- You can use Raspistill for taking pictures and Raspivid to take video!
- You can also use Python to interface with the camera (this is what we will do now)

## Step 5 – Picture time with Python!

- First we must get the python-picamera library

```
>sudo apt-get update
```

```
>sudo apt-get install python-picamera
```

## Activity 1: Raspberry Pi & Camera Module

- Next, open a Python script and enter the following code (no need to write the comments, remember #this is a comment)

```
#make library available to the script
import picamera
#Now create an instance of the PiCamera Class
camera = picamera.PiCamera()
#Take a picture, named img1
camera.capture('img1.jpg')
```

- And that's all you need to take a picture!



# Activity 1: Raspberry Pi & Camera Module

Options for Python PiCamera you can play around with

- Horizontal and Vertical Flip

```
camera.hflip = True #apply a horizontal flip  
camera.vflip = True #apply a vertical flip
```

- Preview your pictures on the camera feed on screen.  
This will overlay your Python session by default.

```
camera.start_preview()  
Camera.stop_preview()
```

Use `stop_preview` to remove the overlay and restore the display.  
Use `Ctrl+D` to end the Python session if needed.

# Activity 1: Raspberry Pi & Camera Module

Options for Python PiCamera you can play around with

- Camera settings: you can change other camera configurations by editing their values, for example:

```
camera.brightness = 70
```

This will change the brightness setting from its default 50 to 70. Values are between 0 to 100.

Other settings you can play with:

```
camera.sharpness  
camera.contrast  
camera.saturation  
camera.rotation
```

## Activity 2:

# MINECRAFT PI EDITION



## Activity 2: Minecraft Pi Edition



Minecraft Pi

### Step 1 – Get Picraft on your pi

- If you already have the Minecraft Application on your Rasbian desktop, you can skip to Step 2.

From your command line:

```
>sudo apt-get update  
>sudo apt-get install minecraft-pi
```

This command installs Minecraft Pi and the Python libraries you will need.

## Activity 2: Minecraft Pi Edition



Minecraft Pi

### Step 2 – Run Picraft

- Once you boot up Raspbian, open a terminal window and enter:  
> minecraft-pi

This will load Minecraft. You will notice the loading times and game might be running on the slow side, this is normal as the Pi isn't a fast machine.

- When Picraft is loaded, click on **Start Game**, followed by **Create new**.

## Activity 2: Minecraft Pi Edition



Minecraft Pi

### Step 3 – Explore Picraft

Action	Reaction
Move mouse	Look around
Right click	Use Item/Place blocks
Left click	Attack/Destroy blocks
A, S, D, W	Move around
Space	Jump
Double space	Fly/Fall
Esc	Pause/open game menu
E	Open inventory
Tab	Toggle the mouse cursor

## Activity 2: Minecraft Pi Edition



Minecraft Pi

### Step 4 – Programming interface with Python

- Time to bring some code into the game!
  1. Bring your focus away from Minecraft (Press Tab)
  2. Open IDLE (Not IDLE3) and move the two windows side by side if you wish
  3. Create a new file in IDLE (File> New Window) and save it (File>Save). You can save this file where ever you like, we suggest you create a new folder in your home folder.

You can name your file whatever you like, I suggest calling it “Helloworld.py”

Make sure it ends with “.py”

## Activity 2: Minecraft Pi Edition



Minecraft Pi

### Step 4 – Programming interface with Python

4. Your first Hello World Mod in Picraft (write this in your helloworld.py)  
(Again, no need to write the comments)

```
from mcpi import minecraft
#this imports the Python Minecraft library
mc = minecraft.Minecraft.create()
#this creates a connection to the game
mc.postToChat("HelloWorld")
#Can you guess what this does?
```



## Activity 2: Minecraft Pi Edition



Minecraft Pi

### Step 4 – Programming interface with Python

5. Save your file(`Ctrl + S`) and run with `F5`. When your code runs, you should see something on your screen in the game.

### Step 5 – Find your location using Python Shell

- To find your location, in a python shell, type the following commands directly to the Python window. Don't forget to hit `enter` after each line:

```
from mcpi import minecraft
mc = minecraft.Minecraft.create()
pos = mc.player.getPos()
```

The variable `pos` now contains your location

## Activity 2: Minecraft Pi Edition



Minecraft Pi

### Step 5 – Find your location using Python Shell (cont.)

- Another way to find your current location is to specific set of coordinates with  $x$ ,  $y$ ,  $z$ . This is your exact location in 3 dimensions. In your Python Shell once more, type:

```
x, y, z = mc.player.getPos()
```

- Now  $x$ ,  $y$ , and  $z$  contain each part of your position.
- $x$  and  $z$  are the walking directions (forward/back and left/right)
- $y$  is up/down

## Activity 2: Minecraft Pi Edition



Minecraft Pi

### Step 6 – Teleport!

- Now you can specify a particular location to teleport to

```
x, y, z = mc.player.getPos()  
mc.player.setPos(x, y+100, z)
```

- This will transport your player to 100 spaces in the air. This will mean you'll teleport to the middle of the sky and fall straight down to where you started.
- Try teleporting to somewhere else!

## Activity 2: Minecraft Pi Edition



Minecraft Pi

### Step 7 – Set Blocks

- You can place a single block at a given set of coordinates with `mc.setBlock()`:

```
x, y, z = mc.player.getPos()  
mc.setBlock(x+1, y, a, 1)
```

- A stone block should appear beside where you're standing. If not, use the mouse to look around, it will be near your original position (x+1).
- The four arguments you gave the method are the `x`, `y`, `z` coordinates plus the block id.

## Activity 2: Minecraft Pi Edition



Minecraft Pi

### Step 7 – Set Blocks (cont.)

- You can use other blocks by changing the id:

id	Block
1	Stone
2	Grass
8	Water
17	Wood
38	Flower
51	Fire

- You can find more id's here <http://minecraft.gamepedia.com/Blocks>

## Activity 2: Minecraft Pi Edition



Minecraft Pi

### Step 7 – Set Blocks (cont.)

- Another way to get blocks is to retrieve their ID through the Python Library, or to store it directly into a variable if you know it:

```
from mcpi import block
dirt = block.Dirt.id
mc.setBlock(x,y,z, dirt)

stone = 1;
mc.setBlock(x+2,y,z, stone)
```

## Activity 2: Minecraft Pi Edition



Minecraft Pi

### Step 8 – Set Multiple Blocks at once

- As well as setting a single block with `setBlock`, you can fill a volume of space in one go with `setBlocks`:

```
stone = 1  
x,y,z = mc.player.getPos()  
mc.setBlocks(x+1,y+1,z+1, x+11, y+11, z+11, stone)
```

## Activity 3: Dropping blocks as you walk in Minecraft Pi Edition

- Now you know how to drop blocks, let's use our moving location to drop blocks when you walk.
- The following code will drop flower behind you wherever you walk:

```
from mcpi import minecraft
from time import sleep

mc.minecraft.Minecraft.create()

flower = 38

while True:
    x, y, z = mc.player.getPos()
    mc.setBlock(x,y,z, flower)
    sleep(0.1)
```



## Activity 3: Dropping blocks as you walk in Minecraft Pi Edition

- Since we used a `while True` loop this will go on forever. To stop it, hit, `Ctrl + C` in the Python window.
- What if we only wanted to drop flowers when the player walks on grass? We can use `getBlock` to find out what type a block is:

```
x, y, z = mc.player.getPos()
this_block = mc.getBlock(x, y-1, z) #block standing on
print(this_block)
```

## Activity 3: Dropping blocks as you walk in Minecraft Pi Edition

- We can use an if statement to choose whether or not we plant a flower:

```
grass = 2
flower = 38
while True
    x, y, z = mc.player.getPos()
    block = mc.getBlock(x, y-1, z)

    if block == grass:
        mc.setBlock(x,y,z flower)
    sleep(0.1)
```

## Activity 3: Dropping blocks as you walk in Minecraft Pi Edition

- Another alternative: turn the tile we're standing on, into grass if it isn't already:

```
if block == grass
    mc.setBlock(x, y, z, flower)
else:
    mc.setBlock(x, y-1, z, grass)
```

- Now we can walk forward and if we walk on grass, we'll leave a flower behind. If it's not grass, it turns to grass. If you walk back onto it, we will now leave a flower.
- What else can you do with this setting blocks technique?

## Other activities with the Raspberry Pi

- The Turing Test (Coding with Python)  
<http://www.raspberrypi.org/learning/turing-test-lessons/>
- Exercises to learn with the Pi  
<http://www.raspberrypi.org/resources/learn/>
- Projects to make with the Pi  
<http://www.raspberrypi.org/resources/make/>

Thank You!

# ONWiE Members Hosting Go Code Girl



# Thank You to Our Sponsors!

## *Patrons*



Partners in Powerful Communities



**Carleton**  
UNIVERSITY

Faculty of  
**Engineering  
and Design**

## *Sponsors*



**UNIVERSITY OF WATERLOO**  
FACULTY OF ENGINEERING