Python PiCamera Worksheet

Wilmslow CoderDojo

January 2015

Stephen Blythe

@shblythe piecesofpi.co.uk

These worksheets were based, with permission, on some excellent work by Carl Monk (@ForToffee) at fortoffee.org.uk

Introduction

- Using these worksheets, you are going to learn how to use the Python language to control the Raspberry Pi Camera
- If you've only used Scratch before, you will find Python a little different, as you have to type everything you want the Pi to do, rather than the "drag and drop" style of Scratch
- Remember to check carefully what you've typed if anything goes wrong

Getting Started: 1, 2, 3

1) If it isn't already started, power on your Raspberry Pi

2) Once you can see the desktop, click the menu and look for the "terminal" icon, it usually looks like a picture of a screen

3) Create a directory to work in, perhaps use your name, and then choose that directory

You are now ready to code!

Smile please!

- Now we're going to put your picture on the screen, so make sure the camera is pointing at you and you're smiling!
- Let's use a text editor called nano to create a new file. Type

```
$ nano smile.py
```

- nano will open with an empty file
- Type in the code you see to the right

```
import picamera
import time

camera=picamera.PiCamera()

try:
    camera.start_preview()
    time.sleep(5)
    camera.stop_preview()
finally:
    camera.close()
```

For these lines which have spaces at the beginning, press TAB before you type to make the space

- Once you've typed in the code, exit and save by pressing Ctrl and X together, then press Y and then ENTER
- Then to run the program, type

```
$ python smile.py
```

What did we just do?

```
import picamera
import time

camera=picamera.PiCamera()

try:

    camera.start_preview()
    time.sleep(5)
    camera.stop_preview()

finally:
    camera.close()
```

Tells python we want to use the "picamera" and "time" modules

Create a PiCamera "object" which we can work with

Tells python to try to run the next bit of code, but if it goes wrong, run what comes after "finally" even if it goes wrong

Show the camera preview on the screen, wait 5 seconds, then turn off the preview

Tidy up after ourselves! If we don't and something goes wrong, the camera preview could be stuck on the screen

Selfie time!

- Now we're going to take your picture, so you might want to give your face a quick wipe!
- Edit the file again with nano
- Remember how?

```
$ nano smile.py
```

```
import picamera
import time

camera=picamera.PiCamera()

try:
    camera.start_preview()
    time.sleep(5)
    camera.capture('selfie.jpg')
    camera.stop_preview()
finally:
    camera.close()
```

- nano will open with the code you entered before
- Add the line in red in the appropriate place – it captures a file of the image in the camera, your face, and stores it in a file called 'selfie.jpg'
- Once you've typed in the code, exit and save by pressing Ctrl and X together, then press Y and then ENTER
- Then to run the program, type

```
$ python smile.py
```

Look what you did!

- Now, I suppose you want to check that file to see if it really saved your picture?
- Type:

```
$ gpicview selfie.jpg
```

- Wow! Did it work?

Special effects?

- So far so good, we can see what the camera can see, and we can save it to a file to take a picture
- But wait, there's more!
- Open the file again with nano (if you don't remember how, look back to the previous sheets)
- Add the line in red in the appropriate place – and change the filename, also shown in red, so that we don't overwrite your first picture

```
import picamera
import time

camera=picamera.PiCamera()

try:
    camera.start_preview()
    camera.image_effect='sketch'
    time.sleep(5)
    camera.capture('sketch.jpg')
    camera.stop_preview()
finally:
    camera.close()
```

- Save the code (remember how?)
- Then run it using python again
- You can use gpicview again to see your picture

What? More? Really?

- There are lots more effects available:
  - sketch, posterise, gpen, colorbalance, film, pastel, emboss, denoise, negative, hatch, colorswap, colorpoint, saturation, blur, watercolor, cartoon, none, washedout, solarize, oilpaint
- Try them all! What do they do?
- Which is your favourite?