



Hackpack Anthology

Volume II

V.O.1



Contents

Hackpack Anthology	1
Computer Science Skills competencies checklist.....	2
Hack 1: Around the world with Python by @ncscomputing world code by @damienmoney	8
Hack 2: 'Gui Zero Tweeting Minecraft Selfie Machine'	13
by Many including mostly Ferran Febregas, Martin O'Hanlon bits on top by @ncscomputing.....	13

Computer Science Skills competencies checklist

Based on NC and @Craigarghs brilliant Python programming with Minecraft book checklist(Reproduced with his kind permission :))

Syntax

Variables		Changing Variables	
Data types		Statements	
Integers		Whitespace and Tabs	
		Single-line comments	
Floats		Multi-line Comments	
Boolean			

Maths Operations

Expressions and statements		Exponentials	
Maths operators		Modulo	
Addition		Operator order	
Subtraction		Interchanging variables and values	
Multiplication		Short hand operators	
Division			

String and console output

Strings		Placeholders	
---------	--	--------------	--

Substrings		Console Input	
String functions -len() -lower() -upper() -str()		Date and Time	
Print			
Concatenation			

Comparators and Control Flow

Comparators including:		Greater than or equal to (>=)	
-Equal to (==)		Boolean Operators	
-Not equal to (!=)		If statements	
-Less than(<)		Else statements	
-Less than or equal to (<=)		Elif statements	
-Greater than (>)			

Functions

Creating and Calling Functions		Built in functions including:	
Returning a value		-max	
Arguments		-min	
Modules		-abs	
Importing modules		-type	

Lists and Dictionaries

Creating lists		Removing items	
Accessing index positions		For loop	
Adding items		Sorting a list	
List length		Combining lists	
Slicing		Defining a dictionary	
Searching a list		Changing / adding items in a dictionary	
Inserting an item		Deleting items in a dictionary	

Functions and Lists

Lists as arguments		Splitting a string into a list	
Modifying every list item		Multi-dimension lists	

Range Function		Joining two lists	
Converting a list into a string		Undefined number of lists	

Loops

While Loops		Strings as lists	
Boolean Operators with While Loops		Looping dictionaries	
Infinite Loops		Indexes and for loops	
Break		Zippping two lists	
While/else		For /else loops	
For Loops		For / else break	

File Input and Output

Opening a file		Reading a line	
Writing and closing a file		Automatically closing a file	
Reading a file		Closed attribute	

Classes and Object Orientated Programming

Creating classes		Creating objects	
------------------	--	------------------	--

U-init-Q		Accessing attributes	
Arguments		Class scope	
Creating methods		Overriding methods and attributes	
Multiple objects		Referencing superclass methods in a subclass	
Inheritance			

Programmer competencies

Reusing code		Working in team	
Decomposing a problem		Sharing knowledge	
Problem solving		Testing	
Persevering when a program doesn't work		Peer review and constructive feedback	
Systems thinking understanding how parts of a program relate to each other		Choosing the correct technology for the solution.	
Communication with others		Requirements analysis (understanding and prioritising the different needs of the system)	

Hack 1: Around the world with Python by @ncscomputing world code by @damienmoney

Acknowledgements

For the project we use a couple of tools that we want to reference and whose authors we want to give credit to.

Damian Mooney created the script which we will use to draw and create the world map. This is used in his ISS Minecraft real time tracker project which you can see [here](#).

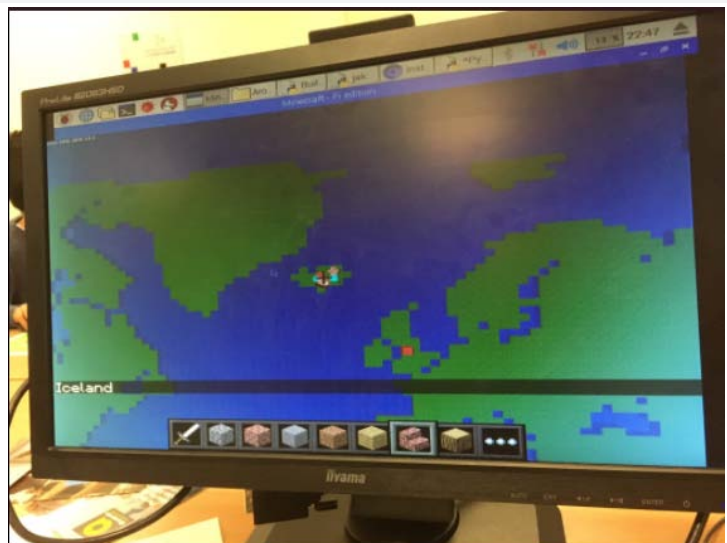
Also want to thank the [author](#) of the raspi2png software I used to capture the screenshots. I followed the tutorial created by Martin O'Hanlon, is available [here](#) you can also use [this guide](#) from Les Pounder.

Building the project

The text here was written by the Kano Blog author (David Ortiz) who used the tutorial I sent in. You can access here: <https://kano.me/blog/tour-the-world-with-minecraft-and-python/>

Here's the list of steps you need to follow to make it happen:

- Create a folder on your desktop to store the files. We will call it "Around the world in Minecraft" for example.
- Download the world map data file from [here](#). This file contains the blueprint for the Minecraft map. This will automatically create a world map in Minecraft so we can focus on the fun bit of travelling around it.
- Save it to the folder you created above. Now you need to download Damien's python script which draws the world based on the blueprint downloaded in the previous step. For this project I have very slightly tweaked it so the link is from my Github repository [here](#).



- Now we have the map blueprint and a script that actually paints it on Minecraft. Looking good so far. Next step is to create a script to teleport Steve and control his position.
- Open up the Python shell.

Create a new document and type in the following code:
Complete Code:

“Written by : @ncscomputing /@warksraspijam”

```
from mcpi import minecraft as minecraft
from mcpi import block as block
from datetime import datetime
import time
import random
import BuildWorldDM as bw

WoolList [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]

EngX = 1.0
EngZ = 50.0

UsaX = 53.8
UsaZ = 39.7

CanadaX = 59.0
CanadaZ = 61.3

IceLandX = 10.0
IceLandZ = 61.1

mc = minecraft.Minecraft.create()

bw.Build()

time.sleep(9)
```

```
def Teleport(x,z,Country):
    mc.player.setPos(x,20,z)
    mc.camera.setFollow()
    mc.setBlock(x,1,z,35,random.choice(WoolList))
    mc.postToChat(Country)
```

```
while True
```

```
    Teleport(EngX,EngZ,"England")
```

```
    time.sleep(8)
```

```
    Teleport(UsaX,UsaZ,"USA")
```

```
    time.sleep(8)
```

```
    Teleport(CanadaX,CanadaZ,"Canada")
```

```
    time.sleep(8)
```

```
    Teleport(IceLandX,IceLandZ,"Iceland")
```

```
    time.sleep(8)
```

```
123456789101112131415161718192021222324252627282930313233343536373839404142434
```

4

- Now you have created your code you will need to save it.
- Open up Minecraft and create a new world and press f5 to work through any errors.
- Your program should now be teleporting around the 4 countries ☺

Explanation of the code

The code first imports the required libraries into the system. Is essential that you import Damien's python script with the correct name of the file. E.g: If the file name is BuildWorldDM.py you would import BuildWorldDM

```
from mcpi import minecraft as minecraft
from mcpi import block as block
from datetime import datetime
import time
import random
import BuildWorldDM as bw
```

After that variables are created. They are the latitude and longitude coordinates of the following countries USA, Eng land, Iceland and Canada, that we use as Minecraft X and Z axis positions.

The list is used to store the different colours of wool. You could use random.randint() function but that is just personal preference.

```
WoolList = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]
```

```
EngX = 1.0
```

```
EngZ = 50.0
```

```
UsaX = 53.8
```

```
UsaZ = 39.7
```

```
CanadaX = 59.0
```

```
CanadaZ = 61.3
```

```
IceLandX = 10.0
```

```
IceLandZ = 61.1
```

After that, we connect to the game so that we can manipulate it. The build world library function 'build' is called which builds the map of the world. A delay of 9 seconds is added to give the Pi enough time to build the world

A function called 'Teleport' is created that allows us to efficiently move the player and drop a coloured block on that country and then send the new country name as a message to the chat.

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

```
bw.Build()
```

```
time.sleep(9)
```

```
def Teleport(x,z,Country):
```

```
mc.player.setPos(x,20,z)
mc.camera.setFollow()
mc.setBlock(x,1,z,35,random.choice(WoolList))
mc.postToChat(Country)
```

1234567891011

And to finalize, we start a permanent loop. This allows the program to run until you kill the python script.

The function Teleport is reused for each country name and X and X coordinates passed through as parameters.

```
while True:
```

```
    Teleport(EngX,EngZ,"England")
```

```
    time.sleep(8)
```

```
    Teleport(UsaX,UsaZ,"USA")
```

```
    time.sleep(8)
```

```
    Teleport(CanadaX,CanadaZ,"Canada")
```

```
    time.sleep(8)
```

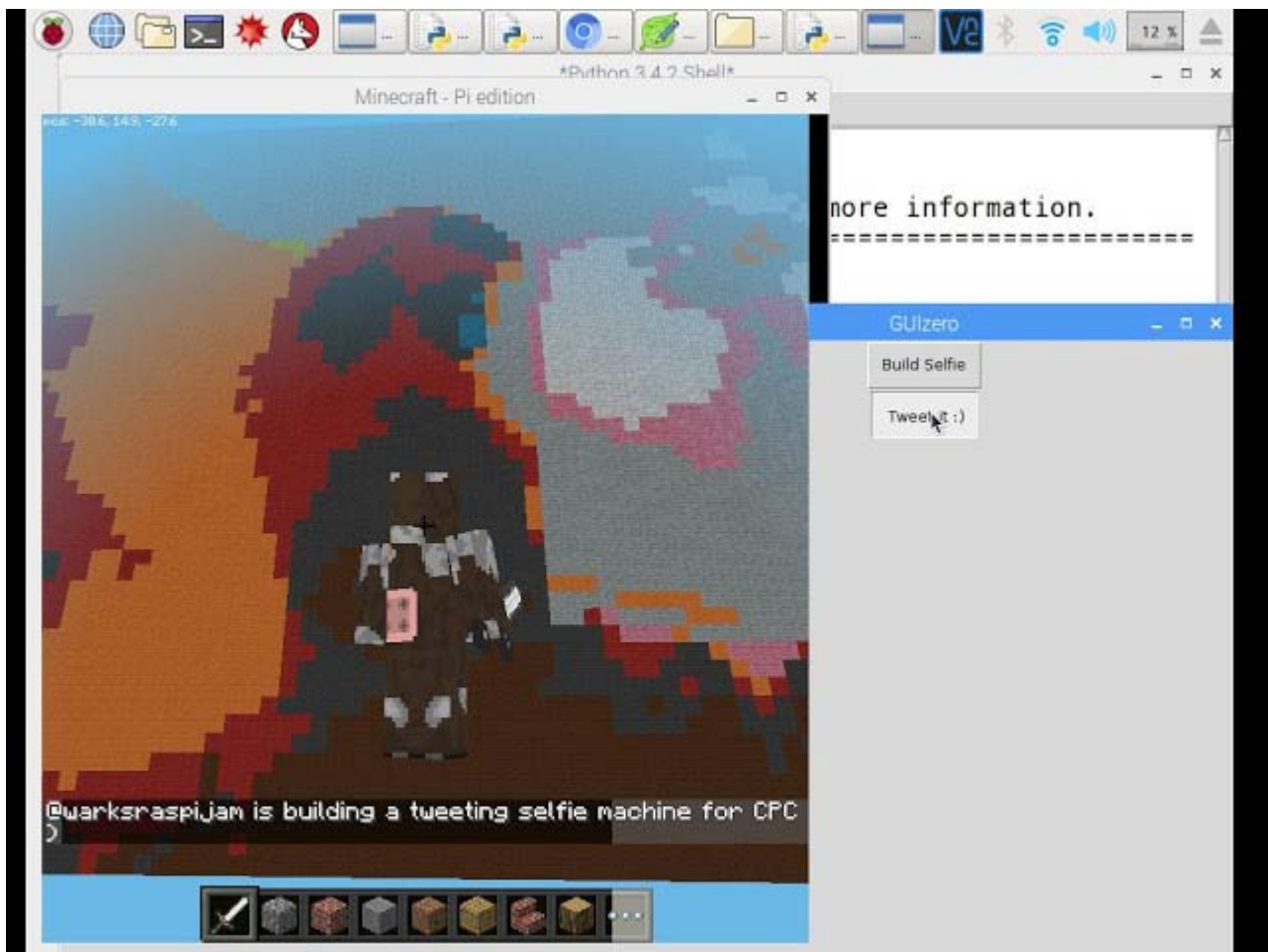
```
    Teleport(IceLandX,IceLandZ,"Iceland")
```

```
        time.sleep(8)
```

Hack 2: 'Gui Zero Tweeting Minecraft Selfie Machine'

by Many including mostly Ferran Febregas, Martin O'Hanlon bits on top by @ncscomputing

Picture of what happens



Code

Martin O'Hanlon
www.stuffaboutcode.com
Minecraft Selfie Camera

```

# Minecraft Picture Rendering Script By Ferran Fabregas (ferri.fc@gmail.com)
from guizero import *
import subprocess

import picamera
from PIL import Image
import math
from mcpi.minecraft import Minecraft
from time import sleep, time

mc=Minecraft.create()

import sys
from twython import Twython

#once you have created your own Twitter app put in your info below

consumer_key = "

consumer_secret = "

access_token = "

access_token_secret = "

api = Twython(consumer_key,consumer_secret,access_token,access_token_secret)

def Tweet():
    pos = mc.player.getTilePos()
    block = mc.getBlock(pos.x,pos.y,pos.z)
    mc.postToChat("@warksraspijam is building a tweeting selfie
machine for CPC :)")
    msg = "LMF CPC tweeting selfie test 1 "
    a=subprocess.check_output('./raspi2png -d 3 -p
"myscreenshot.png"',shell=True)
    photo = open('myscreenshot.png', 'rb')
    response = api.upload_media(media=photo)
    api.update_status(status = msg, media_ids=[response['media_id']])

def takePicture(filename):
    with picamera.PiCamera() as camera:
        camera.start_preview(alpha=192)
        sleep(1)
        camera.capture(filename)
        camera.stop_preview()

def colormap(pixel):
    white=(221,221,221)
    orange=(219,125,62)
    magenta=(179,80,188)

```

```

lightblue=(107,138,201)
yellow=(177,166,39)
lime=(65,174,56)
pink=(208,132,153)
gray=(64,64,64)
lightgray=(154,161,161)
cyan=(46,110,137)
purple=(126,61,181)
blue=(46,56,141)
brown=(79,50,31)
green=(53,70,27)
red=(150,52,48)
black=(25,22,22)

```

```

colors=(white,orange,magenta,lightblue,yellow,lime,pink,gray,lightgray,cyan,purple,blue,brown,green,red,black)

```

```

thecolor=0
finalresult=256*256*256
for idx,color in enumerate(colors):
    result=math.fabs(color[0]-pixel[0])+math.fabs(color[1]-pixel[1])+math.fabs(color[2]-pixel[2])
    if result<finalresult:
        finalresult=result
        thecolor=idx
return thecolor

```

```

def buildMCImage(mc, filename, pos):

```

```

    MAXY = 60
    im = Image.open(filename)

    #resize image file
    if im.size[1] > MAXY:
        ratio = MAXY / float(im.size[1])
        sizeY = MAXY
        sizeX = int(im.size[0] * ratio)
        size = sizeX, sizeY
        im.thumbnail(size, Image.ANTIALIAS)

```

```

    pixels=im.load()

```

```

    for x in range (-int(im.size[0]/2),int(im.size[0]/2)):
        for y in range (0,(im.size[1])):
            mc.setBlock(pos.x + x, pos.y + sizeY - y, pos.z - 1, 35,
                colormap(pixels[x+(im.size[0]/2) , y]))

            #print "{}.{}.{}".format(x, y, 5)
            sleep(0.005)

```

```

def BuildAndSave():
    #mc=Minecraft.create()

```

```
filename = "/home/pi/" + str(int(time())) + ".png"
mc.postToChat("Taking picture in 1 second")
sleep(1)
takePicture(filename)
mc.postToChat("Building image")
pos=mc.player.getTilePos()
pos.x = pos.x-30
#start in front of you
buildMCImage(mc, filename, pos)
mc.postToChat("Image built")
# break
sleep(0.1)

app = App()
buttonBuildImage = PushButton(app,BuildAndSave,text="Build Selfie")
buttonTweetImage = PushButton(app,Tweet,text="Tweet it :)")
app.display()
```