technotes Documentation

Release 1

Tom JIANG

April 01, 2015

CHAPTER

ONE

KID'S ACTIVITY

1.1 Minecraft Pi Edition

1.1.1 Basic commands

W	move forward
S	move backward
A	move left
D	move right
E	show inventory of blocks
1-8	select items in the quick bar
Space / Ctrl + Space	jump (ascend in fly-mode)
Shift / Ctrl + Shift	sneak (descend in fly-mode)
ESC	pause / menu
left mouse	destroy blocks
right mouse	place blocks
double **Space**	fly / fall
Tab	release mouse

1.1.2 List of python programs

Short-cuts

Ctrl + S	save
F5	run

Display the player's position

```
from mcpi import minecraft

mc = minecraft.Minecraft.create()

x,y,z = mc.player.getTilePos()
mc.postToChat("x="+str(x)+", y="+str(y)+", z="+str(z))
```

Teleport (change the player's position)

In the following program, the player will be teleported 100 higher.

```
from mcpi import minecraft

mc = minecraft.Minecraft.create()

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

Build a huge block of activated TNTs

When you click one TNT, there will be an explosion around that block of TNTs.

```
from mcpi import minecraft

mc = minecraft.Minecraft.create()

x,y,z = mc.player.getTilePos()

tnt = 46
activated = 1
mc.setBlocks(x+1,y+1,z+1,x+5,y+5,z+5,tnt,activated)
```

Put a flower on the path

We will leave a flower when we are on a block of grass. Otherwise we will change the beneath block to a grass block.

```
from mcpi import minecraft
from time import sleep

mc = minecraft.Minecraft.create()

grass = 2
flower = 38
while True:
    x,y,z = mc.player.getTilePos()
    block_beneath = mc.getBlock(x,y-1,z)
    if block_beneath == grass:
        mc.setBlock(x,y,z,flower)
    else:
        mc.setBlock(x,y-1,z,grass)
    sleep(0.1)
```

Clear space with input size

We will clear space for a given **size**. To do so, we will build a cube of **size** x **size** x **size** blocks, filled with the AIR block.

```
from mcpi import minecraft, block

mc = minecraft.Minecraft.create()

x,y,z = mc.player.getTilePos()
size = int(raw_input("size of area to clear? "))
if size > 0:
    mc.setBlocks(x,y,z,x+size,y+size,z+size,block.AIR.id)
```

Challenge: Change a little the above program so that the player is in the middle of the cleared space (and also dig down a few blocks).

Build a house, then a street

```
from mcpi import minecraft, block

mc = minecraft.Minecraft.create()
SIZE = 20

def house():
    midx = x + SIZE/2
    midy = y + SIZE/2
    mc.setBlocks(     x,        y,   z,   x+SIZE,   y+SIZE,   z+SIZE,block.COBBLESTONE.id)
    mc.setBlocks(   x+1,      y+1,z+1,x+SIZE-1,y+SIZE-1,z+SIZE-1,         block.AIR.id)
    # left window
    mc.setBlocks(   x+3,y+SIZE-3,   z,   midx-3,   midy+3,        z,       block.GLASS.id)
    # right window
    mc.setBlocks(midx+3,y+SIZE-3,   z,x+SIZE-3,   midy+3,        z,       block.GLASS.id)
    # door
    mc.setBlocks(midx-3,        y,   z,   midx+3,    midy,        z,  block.DOOR_WOOD.id)
    mc.setBlocks(     x,  y+SIZE,   z,   x+SIZE,   y+SIZE,   z+SIZE,        block.SNOW.id)
    mc.setBlocks(   x+1,      y+1,z+1,x+SIZE-1,     y+1,z+SIZE-1,        block.WOOL.id,7)

x,y,z = mc.player.getTilePos()

# build a house
house()

# build a street
for h in range(5):
    house()
    x = x+SIZE
```

1.2 Pygame

1.2.1 List of pygame programs

Draw a circle

```
import pygame

width,height = 640,480
radius = 100
fill = 1

pygame.init()
window = pygame.display.set_mode((width,height))
window.fill(pygame.Color(255,255,255)) # white

while True:
    pygame.draw.circle(window,
                       pygame.Color(255,0,0), # red
                       (width/2,height/2),
```

```
                        radius,
                        fill)
    pygame.display.update()
    if pygame.QUIT in [e.type for e in pygame.event.get()]:
        break
```

Draw circles based on mouse move / position

```
import pygame
from pygame.locals import *

width,height = 640,640
radius = 0
fill = 1
mouseX,mouseY = 0,0

pygame.init()
window = pygame.display.set_mode((width,height))
window.fill(pygame.Color(255,255,255)) # white
fps = pygame.time.Clock() # FPS = Frame Per Second

while True:  # one frame per loop
    for event in pygame.event.get():
        if event.type == MOUSEMOTION:
            mouseX,mouseY = event.pos
        if event.type == MOUSEBUTTONDOWN: # mouse click
            window.fill(pygame.Color(255,255,255)) # clear screen
        radius = (abs(width/2 - mouseX) + abs(height/2 - mouseY))/2 + 1
        pygame.draw.circle(window,
                           pygame.Color(255,0,0), # red
                           (mouseX,mouseY),
                           radius,
                           fill)
    pygame.display.update()
    if pygame.QUIT in [e.type for e in pygame.event.get()]:
        break
    fps.tick(30) # wait so that frame rate is 30 fps
```

1.3 Scratch

CHAPTER

TWO

HARDWARE

2.1 Raspberry Pi

2.1.1 Default settings

login	**pi**
password	**raspberry**
hostname	**raspberrypi**
keyboard	UK

2.1.2 Basic commands

Config

```
$ sudo raspi-config
```

Start X server

```
$ startx
```

Reboot

```
$ sudo reboot
```

Shutdown

```
$ sudo shutdown -h now
```

Change datetime

```
$ sudo date --set="Sun Nov 18 1:55:16 EDT 2012"
```

Update

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

2.1.3 Information

Check OS version

```
$ cat /proc/version
```

Check board version

```
$ cat /proc/cpuinfo
```

Display network interface and associated IP addresses

```
$ ifconfig
```

2.1.4 Short-cuts

Ctrl + C	kill currently running program
Ctrl + D	exit shell
Ctrl + A	move cursor to the beginning of the line
Ctrl + E	move cursor to the end of the line
Ctrl + Alt + Backspace	[optional] terminate the X server

2.1.5 Setup Keyboard

The default keyboard is UK. Let's change it to AU keyboard.

The trick is that Australia is not listed in the country list for the keyboard, we need to setup a US keyboard instead.

Change the keyboard config

```
$ sudo vi /etc/default/keyboard
```

```
XKBMODEL ="pc105"
XKBLAYOUT="us"
XKBVARIANT=""
XKBOPTIONS=""

BACKSPACE="guess"
```

Then run the following commands and reboot

```
$ sudo setxkbmap -layout us
$ sudo udevadm trigger --subsysstem-match=input --action=change
```

2.1.6 Utilities / Softwares

raspi-config tool

```
$ sudo apt-get install raspi-config
```

Minecraft

```
$ sudo apt-get install minecraft-pi
```

Screenshot : scrot

```
$ sudo apt-get install scrot
```

Mercurial

```
$ sudo apt-get install mercurial
```

2.2 Arduino

CHAPTER

THREE

SYSTEM

3.1 Linux

3.2 Windows

3.2.1 Connect to Internet via Ethernet cable (from PC/laptop)

Control Panel –> Network and Internet –> Network Connections

Ctrl + select local and wireless connections, right click **Bridge Connections**

CHAPTER

FOUR

EDITOR

4.1 VIM (Vi IMproved)

4.1.1 Basic commands

Read only (use :wq! to force the modification)

```
$ vim -R file
```

Running shell commands

```
!command
```

e.g. **!ls** will launch **ls**

if you wants to go directly to shell without quitting from VI editor you can go by executing **!sh / !bash / !ksh** from VI and then come back to VI editor by just executing command **exit** from shell. for Cygwin, **!bash** and **exit** seems to be the best choice

Launch VIM from command line

```
$ vi file.txt                       open and edit file file.txt
$ vi file1.txt file2.txt file3.txt  open several files
$ vi +25 file.txt                   edit from the 25th line
$ vi + file.txt                     edit at the end of file
$ vi +/text file.txt                edit from the first line containing the word test
$ vi -r file.txt                    restore a crashed file
$ view file.txt                     vi in read-only mode
$ vimtutor                          VIM tutorial
```

Saving and quiting commands

```
:w            save the current file (before quit)
:w file.txt   save the modified file with another file name (even if the file was opened in read-only
:wq           save and quit
ZZ            save and quit
:q!           quit without saving
:wq!          save change in the current file opened in read-only mode, and then quit
:w!           save change in the current file opened in read-only mode
```

Checking history and help

```
:history        vim commands history
:help           all helps
:help command   help on one command
```

Recording and replaying commands

Recoding in vim or VI editor can be done by using **q** and the executing recorded comment by using **q@1**

4.1.2 Options

Here are the major VIM editor options

:set nu	This will display line number in front of each line quite useful if you want line by line information.
	You can turn it off by executing **set nonu**. Remember for turning it off put "no" in front of option, like here option is "nu" so for turning it off use "nonu".
:set nonu	removing line number display
:set hlsearch	This will highlight the matching word when we do search in VI editor, quite useful but if you find it annoying or not able to see sometime due to your color scheme you can turn it off by executing **set nohlsearch**.
:set wrap	If your file has contains some long lines and you want them to wrap use this option, if its already on and you just don't want them to wrap use **set nowrap**.
:colorscheme	color scheme is used to change color of VIM editor, my favorite color scheme is murphy so if you want to change color scheme of VI editor you can do by executing **colorscheme murphy**.
:syntax on	syntax can be turn on and off based on your need, if it's on it will display color syntax for .xml, .html and .perl files.
:set ignorecase	This VI editor option allows you do case insensitive search because if it's set VI will not distinguish between two words which are just differ in case.
:set smartcase	Another VI editor option which allows case-sensitive search if the word you are searching contains an uppercase character.

4.1.3 Navigation

Here are some navigating commands

gg	goes to start of file
shift g	goes to end of file
0	goes to beginning of the line
$	goes to end of the line
nG	goes to nth line
:n	another way of going to nth line

4.1.4 Editing

Editing commands

yy	equivalent to cut also called yank
p	paste below line
Shift p	paste above line
dd	deletes the current line
5dd	deletes 5 lines
u	undo last change
Ctrl + R	Re do last change

Copy (or cut) / paste (without strange indent)

1. move the mouse pointer to the beginning of your desired copy text
2. type 'v' (visual) for Visual mode, then using mouse pointer move to the end of selected text
3. type 'y' (yank) for Copy or 'd' (delete) for Cut
4. move to your paste location, then type 'p' (paste)

Tabulation

1. define TAB as 2 spaces

```
:set tabstop=2 shiftwidth=2 expandtab
```

2. replace TAB by 4 spaces

```
:%s/\t/    /g
```

4.1.5 Multi-files, multi-windows

Opening multi-files / another file

```
$ vim file1 file2 file3 ...
```

```
:n        edit next file among multi-files (with respect to the order given in the command line)
:wn       save the modification and edit the next file
:n!       edit the next file without saving the ongoing modification
:e        reload the current file
:e file   load file in the current window
```

Multi-windows

```
:sp(lit) file   split horizontally the window and load file in the splitted window
:vsplit file    split vertically the window and load file in the splitted window
:vs             vertically split window
CTRL + w + w    switch among all (sub-)windows
:q              close the current (sub-)window
```

Since Pythagoras, we know that $a^2 + b^2 = c^2$.

$$\begin{aligned}(a+b)^2 &= (a+b)(a+b) \\ &= a^2 + 2ab + b^2\end{aligned}$$

4.2 Joe

4.3 Nano

CHAPTER

FIVE

PROGRAMMING LANGUAGE

5.1 Shell

5.2 Python