# Brandon Quinlan (bmq4) and Caitlin Stanton (cs968)

# ECE5725, Lab 2, Due 3/7

import os

import pygame

from pygame.locals import \*

import time

import RPi.GPIO as GPIO

# Initialize variables

starttime = time.time()

code\_running = True

game\_running = False

paused = False

framerate = 40 #framerate value for fast/slow buttons

GPIO.setmode(GPIO.BCM)

GPIO.setup(27,GPIO.IN,pull\_up\_down=GPIO.PUD\_UP)

clock = pygame.time.Clock()

# Display on TFT, use touchscreen

os.putenv('SDL\_VIDEODRIVER','fbcon')

os.putenv('SDL\_FBDEV','/dev/fb1')

os.putenv('SDL\_MOUSEDRV','TSLIB')

os.putenv('SDL\_MOUSEDEV','/dev/input/touchscreen')

# Initialize display constants

pygame.init()

pygame.mouse.set\_visible(False)

WHITE = 255,255,255

BLACK = 0,0,0

size = width, height = 320,240

speed = [5,5]

speed2 = [7,7]

screen = pygame.display.set\_mode((320,240))

ball = pygame.image.load('magic\_ball.png')

ball2 = pygame.image.load('ledlightblue.png')

ball\_rect = ball.get\_rect(center=(50,50))

ball2\_rect = ball2.get\_rect(center=(100,100))

my\_font = pygame.font.Font(None,30)

my\_buttons = {'start':(40,180),'quit':(280,180)}

game\_buttons = {'pause':(40,180),'fast':(130,180),'slow':(210,180), \

'back':(280,180)}

screen.fill(BLACK)

hit\_text = ""

# Continue looping until quit button or timeout

while code\_running:

clock.tick(framerate)

screen.fill(BLACK)

# balls are displayed and not paused - need to update location and speed

if (game\_running):

if(not paused):

speed = [5,5]

speed2 = [7,7]

ball\_rect = ball\_rect.move(speed)

ball2\_rect = ball2\_rect.move(speed2)

if ball\_rect.left < 0 or ball\_rect.right > width:

speed[0] = -speed[0]

if ball\_rect.top < 0 or ball\_rect.bottom > height:

speed[1] = -speed[1]

if ball2\_rect.left < 0 or ball2\_rect.right > width:

speed2[0] = -speed2[0]

if ball2\_rect.top < 0 or ball2\_rect.bottom > height:

speed2[1] = -speed2[1]

# check if balls collide with each other

if ball\_rect.colliderect(ball2\_rect):

speed[0] = -speed[0]

speed[1] = -speed[1]

speed2[0] = -speed2[0]

speed2[1] = -speed[1]

if(paused):

speed11[0]=speed[0]

speed[0]=0

speed11[1]= speed[1]

speed[1]=0

speed21[0]=speed2[0]

speed2[0]=0

speed21[1]= speed2[1]

speed2[1]=0

# ensures buttons will be displayed/checked whether game is paused/unpaused

if (game\_running):

# pause,fast,slow,and back buttons displayed

for my\_text, text\_pos in game\_buttons.items():

text\_surface = my\_font.render(my\_text,True,WHITE)

rect = text\_surface.get\_rect(center=text\_pos)

screen.blit(text\_surface,rect)

# balls displayed (in new location if unpaused)

screen.blit(ball,ball\_rect)

screen.blit(ball2,ball2\_rect)

pygame.display.flip()

# Check for button presses

for event in pygame.event.get():

if (event.type is MOUSEBUTTONDOWN):

pos = pygame.mouse.get\_pos()

elif (event.type is MOUSEBUTTONUP):

pos = pygame.mouse.get\_pos()

x,y = pos

if y > 160:

if x < 60:

print 'pause pressed'

paused = not paused

elif x > 110 and x < 150:

print 'fast pressed'

framerate = framerate\*1.1

elif x > 190 and x < 230:

print 'slow pressed'

framerate = framerate/0.8

elif x > 260:

print 'back pressed'

game\_running = False

code\_running = True

paused = False

# Balls not displayed - start menu

else:

# start, quit buttons displayed

for my\_text, text\_pos in my\_buttons.items():

text\_surface = my\_font.render(my\_text,True,WHITE)

rect = text\_surface.get\_rect(center=text\_pos)

screen.blit(text\_surface,rect)

# Display "hit"

if (hit\_text != ""):

text\_surface = my\_font.render(hit\_text,True,WHITE)

rect = text\_surface.get\_rect(center=(100,100))

screen.blit(text\_surface,rect)

# Check start screen button presses

for event in pygame.event.get():

if (event.type is MOUSEBUTTONDOWN):

pos = pygame.mouse.get\_pos()

elif (event.type is MOUSEBUTTONUP):

pos = pygame.mouse.get\_pos()

x,y = pos

if y > 160:

if x < 60:

print 'start pressed'

game\_running = True

code\_running = True

elif x > 260:

print 'quit pressed'

game\_running = False

code\_running = False

else:

hit\_text="Hit at " + str(pos[0]) + "," + str(pos[1])

print hit\_text

pygame.display.flip()

# Check quit conditions

if not GPIO.input(27):

code\_running = False

now = time.time()

elapsed = now - starttime

if elapsed >= 30:

code\_running = False