import simplegui

import random

WIDTH = 600

HEIGHT = 400

BALL\_RADIUS = 20

PAD\_WIDTH = 8

PAD\_HEIGHT = 80

HALF\_PAD\_WIDTH = PAD\_WIDTH / 2

HALF\_PAD\_HEIGHT = PAD\_HEIGHT / 2

LEFT = False

RIGHT = True

ball\_pos = [WIDTH/2, HEIGHT/2]

ball\_vel = [ 0, 0]

paddle1\_pos = [HALF\_PAD\_WIDTH , (HEIGHT - PAD\_HEIGHT) /2 ]

paddle2\_pos = [WIDTH - HALF\_PAD\_WIDTH, (HEIGHT - PAD\_HEIGHT) /2 ]

paddle1\_vel = 0

paddle2\_vel = 0

score1 = 0

score2 = 0

win\_scores = 5

win\_scores\_not = 5

time\_start = True

#開啟新局，分數、球拍、球，回到原始位置

def new\_game():

global paddle1\_pos, paddle2\_pos, paddle1\_vel, paddle2\_vel, ball\_pos, ball\_vel

global score1, score2

paddle1\_pos = [HALF\_PAD\_WIDTH , (HEIGHT - PAD\_HEIGHT) /2 ]

paddle2\_pos = [WIDTH - HALF\_PAD\_WIDTH, (HEIGHT - PAD\_HEIGHT) /2 ]

score1 = 0

score2 = 0

ball\_vel = [0, 0]

direction = random.randrange(0, 2)

spawn\_ball(direction)

#暫停，原地再開始

def Start():

global ball\_pos, ball\_vel, score1, score2

global time\_start

if time\_start :

#球的位置

ball\_pos[0] = ball\_pos[0] + ball\_vel[0]

ball\_pos[1] = ball\_pos[1] + ball\_vel[1]

timer.start()

else:

timer.stop()

time\_start = True

def Stop():

global time\_start

if timer.is\_running():

time\_start = False

#設定贏家分數

def input\_win\_scores(input\_win\_scores):

global win\_scores,win\_scores\_not

try:

int(input\_win\_scores) <= 0

except:

win\_scores\_not = input\_win\_scores

else:

if int(input\_win\_scores) <= 0:

win\_scores\_not = input\_win\_scores

else:

win\_scores = int(input\_win\_scores)

win\_scores\_not = int(input\_win\_scores)

new\_game()

Start()

#贏家分數警語

def warning():

global win\_scores\_not

try:

int(win\_scores\_not) <= 0

except:

return "Please enter an positive integer in winner scores."

else:

if int(win\_scores\_not) <= 0:

return "Please enter an positive integer in winner scores."

else:

return ""

#使球回到中心，並隨機選擇向右上或左下

def spawn\_ball(direction):

global ball\_pos, ball\_vel

ball\_pos = [WIDTH/2, HEIGHT/2]

if direction == True:

ball\_vel[0] = random.randrange(100, 200)/100.0

ball\_vel[1] = random.randrange(100, 200)/100.0

else:

ball\_vel[0] = -random.randrange(100, 200)/100.0

ball\_vel[1] = random.randrange(100, 200)/100.0

#球速

def speed\_up():

global ball\_vel

ball\_vel[0] = ball\_vel[0]\*1.25

ball\_vel[1] = ball\_vel[1]\*1.15

def draw(canvas):

global score1, score2, paddle1\_pos, paddle2\_pos, ball\_pos, ball\_vel, paddle1\_vel, paddle2\_vel

global win\_scores, win\_scores\_not, time\_start

#開始公布語

if ball\_vel == [ 0, 0] and time\_start :

canvas.draw\_text("Press 'n' to start a new game.", (WIDTH/5, HEIGHT/3), 30, "White")

canvas.draw\_text("Press 'space' to stop the game.", (WIDTH/5, HEIGHT/2.5), 30, "White")

#暫停公布語

if timer.is\_running() != True and ball\_pos != [0, 0] and ball\_pos != [WIDTH/2, HEIGHT/2] :

canvas.draw\_text("Stop", (WIDTH/2.2, HEIGHT/2), 30, "White")

canvas.draw\_text("Press 'space' to restart the game.", (WIDTH/3.2, HEIGHT/1.8), 20, "White")

#贏家分數和贏家分數警語

canvas.draw\_text(str(win\_scores), (WIDTH / 3 , 50), 20, "White")

canvas.draw\_text(str(win\_scores), (WIDTH / 1.2 , 50), 20, "White")

canvas.draw\_text("/" , (WIDTH / 3.18 , 50), 20, "White")

canvas.draw\_text("/" , (WIDTH / 1.23 , 50), 20, "White")

canvas.draw\_text(warning() , (WIDTH / 5 , HEIGHT/1.5), 20, "White")

#畫球桌

canvas.draw\_line([WIDTH / 2, 0],[WIDTH / 2, HEIGHT], 1, "White") #中線

#球的位置

canvas.draw\_circle(ball\_pos, BALL\_RADIUS, 1, "White", "White")

#球拍的位置，球拍不能超過畫布

if paddle1\_pos[1] < 0 :

paddle1\_pos[1] = 0

elif paddle1\_pos[1]+PAD\_HEIGHT > HEIGHT:

paddle1\_pos[1] = (HEIGHT - PAD\_HEIGHT)

else:

paddle1\_pos[1] = paddle1\_pos[1] + paddle1\_vel

if paddle2\_pos[1] < 0 :

paddle2\_pos[1] = 0

elif paddle2\_pos[1]+PAD\_HEIGHT > HEIGHT:

paddle2\_pos[1] = (HEIGHT - PAD\_HEIGHT)

else:

paddle2\_pos[1] = paddle2\_pos[1] + paddle2\_vel

canvas.draw\_line(paddle1\_pos, [HALF\_PAD\_WIDTH , paddle1\_pos[1]+PAD\_HEIGHT], PAD\_WIDTH, "White")

canvas.draw\_line(paddle2\_pos, [WIDTH - HALF\_PAD\_WIDTH, paddle2\_pos[1]+PAD\_HEIGHT], PAD\_WIDTH, "White")

#球的位置，球撞擊到上下邊緣會反彈

if ball\_pos[1] <= BALL\_RADIUS:

ball\_vel[1] = - ball\_vel[1]

speed\_up()

if HEIGHT - ball\_pos[1] <= BALL\_RADIUS:

ball\_vel[1] = - ball\_vel[1]

speed\_up()

#球撞擊到球拍會反彈

if paddle1\_pos[1] <= ball\_pos[1] <= paddle1\_pos[1]+ PAD\_HEIGHT and ball\_pos[0] < HALF\_PAD\_WIDTH + BALL\_RADIUS:

ball\_vel[0] = - ball\_vel[0]

speed\_up()

if paddle2\_pos[1] <= ball\_pos[1] <= paddle2\_pos[1]+ PAD\_HEIGHT and ball\_pos[0] > WIDTH-(PAD\_WIDTH + BALL\_RADIUS):

ball\_vel[0] = - ball\_vel[0]

speed\_up()

#球撞擊到球拍邊緣會反彈

((BALL\_RADIUS\*\*2)/2)\*\*0.5

ditance\_up = ((ball\_pos[0] - (paddle1\_pos[0] - HALF\_PAD\_WIDTH))\*\*2 + (ball\_pos[1] - paddle1\_pos[1])\*\*2)\*\*0.5

ditance\_low = ((ball\_pos[0] - (paddle1\_pos[0] - HALF\_PAD\_WIDTH))\*\*2 + (ball\_pos[1] - paddle1\_pos[1]-PAD\_HEIGHT)\*\*2)\*\*0.5

if ditance\_up <= BALL\_RADIUS:

ball\_vel[0] = - ball\_vel[0]

speed\_up()

if ditance\_low <= BALL\_RADIUS:

ball\_vel[0] = - ball\_vel[0]

speed\_up()

#球撞擊到左右畫布邊緣會回到原點，且會使對方得分

if ball\_pos[0] - BALL\_RADIUS <= 0:

spawn\_ball(RIGHT)

score2 = score2 +1

if ball\_pos[0] + BALL\_RADIUS >= WIDTH:

spawn\_ball(LEFT)

score1 = score1 +1

canvas.draw\_text(str(score1), (WIDTH / 4 , 50), 50, "White")

canvas.draw\_text(str(score2), (3\* WIDTH / 4, 50), 50, "White")

#結束公布語

if score1 >= win\_scores :

canvas.draw\_text("Left side player win!", (WIDTH/3.5, HEIGHT/3), 30, "White")

canvas.draw\_text("Press 'n' to start a new game.", (WIDTH/5, HEIGHT/2.5), 30, "White")

ball\_pos = [WIDTH/2, HEIGHT/2]

Stop()

elif score2 >= win\_scores:

canvas.draw\_text("Right side player win!" , (WIDTH/3.5, HEIGHT/3), 30, "White")

canvas.draw\_text("Press 'n' to start a new game.", (WIDTH/5, HEIGHT/2.5), 30, "White")

ball\_pos = [WIDTH/2, HEIGHT/2]

Stop()

#設定玩家鍵盤

def keydown(key):

global paddle1\_vel, paddle2\_vel, time\_start

if key == simplegui.KEY\_MAP["space"]:

Stop()

if key == simplegui.KEY\_MAP["space"] and time\_start == True:

Start()

if key == simplegui.KEY\_MAP["n"]:

Start()

new\_game()

if key == simplegui.KEY\_MAP["w"]:

paddle1\_vel = -3

if key == simplegui.KEY\_MAP["s"]:

paddle1\_vel = 3

if key == simplegui.KEY\_MAP["up"]:

paddle2\_vel = -3

if key == simplegui.KEY\_MAP["down"]:

paddle2\_vel = 3

def keyup(key):

global paddle1\_vel, paddle2\_vel

if key == simplegui.KEY\_MAP["w"]:

paddle1\_vel = 0

if key == simplegui.KEY\_MAP["s"]:

paddle1\_vel = 0

if key == simplegui.KEY\_MAP["up"]:

paddle2\_vel = 0

if key == simplegui.KEY\_MAP["down"]:

paddle2\_vel = 0

frame = simplegui.create\_frame("Pong", WIDTH, HEIGHT)

frame.set\_draw\_handler(draw)

timer = simplegui.create\_timer(15, Start)

frame.set\_keydown\_handler(keydown)

frame.set\_keyup\_handler(keyup)

frame.add\_label("Right side player", 200)

frame.add\_label("Up = up", 200)

frame.add\_label("Down = down", 200)

frame.add\_label(" ", 200)

frame.add\_label("Left side player", 200)

frame.add\_label("Up = w", 200)

frame.add\_label("Down = s", 200)

frame.add\_label(" ", 200)

frame.add\_label("new game = n", 200)

frame.add\_label("stop/start = space", 200)

frame.add\_label(" ", 200)

frame.add\_label("Set an winner scores.", 200)

frame.add\_label("Press 'Enter' to input.", 200)

frame.add\_input("(Original setting is 5 points.)", input\_win\_scores, 100)

frame.start()