# Python Programming

## *Project 1*

# Ping Pong Game Level 3.

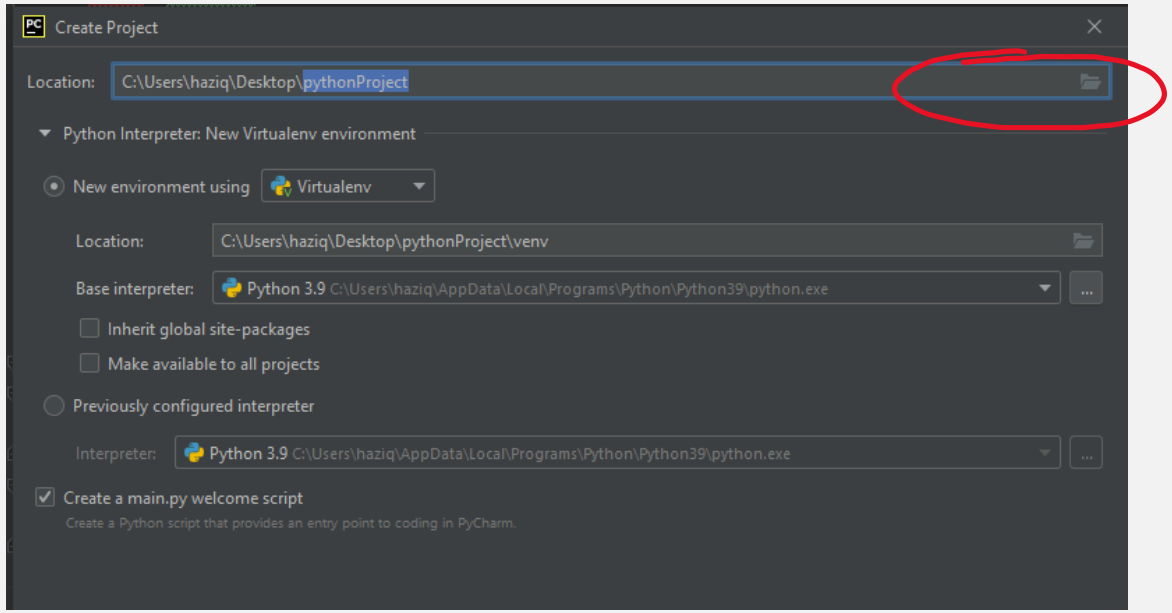
Presented by Advaspire Team

# What we are making

In this lesson, we are going to make a multi player ping pong game. Previously, you have learned how to implement the features to make a single player game. In this project, we are going to add more features that will work well with multiplayer games.

# Setting Up

Create a new project. This time, I recommend to create it at your desktop for easy access. Do the same thing we did last time.



# Setting Up Pyzero Games

Like usual when using the pyzero, make sure to import it at the start and also put pgzrun.go() at end of the code.

```
import pgzrun

WIDTH = 600
HEIGHT = 400

pgzrun.go()
```

Then, we need to set up the game window. Do the following. We are going for 600x400. You are free to make it bigger but keep in mind that the coordinates for sprites will be different depending on your screen size.

# Download Assets

Use the assets that you downloaded for the previous lesson. If you move everything inside your last project, just copy and paste the assets into your new project folder.

# Variables

As this is a multiplayer game, we are going to create 2 bats using one sprite file. To do it, just create 2 variables and put a number after it like in the example. The rest can just follow the example.

The Mid variable is just for indicating where is the middle of the screen. You can leave it if you want. ScoreLeft Right is for keeping track of the scores for each player.

```
Bat1 = Actor('pongbar')
Bat1.pos = (0,200)
Bat2 = Actor('pongbar')
Bat2.pos =(600,200)
BG = Actor('background')
orb = Actor('pongball')
orb.pos = (300,200)
Mid = Rect((300,0),(10,400))
vx =5
vy =5
ScoreLeft =0
ScoreRight=0
time =30
Game_Over= False
Game_End =False
```

# Draw the Sprites

In the draw function, start with an if: . After you have finished the code, change the if to else. The reason why is that we are going to add the game over and end screen after we go further and you can't use the else if there is no if: . Draw every variables that involves a sprite. The mid can be removed if you didn't use it in the variable.

```
else:
    screen.clear()
    BG.draw()
    screen.draw.filled_rect(Mid,'blue')
    Bat1.draw()
    Bat2.draw()
    orb.draw()
    time_string = str(round(time))
    screen.draw.text('Time: '+time_string,(230,0),color='Purple',fontsize=50)
    screen.draw.text("Score: "+ str(ScoreLeft),(0,0),color='Green')
    screen.draw.text('Score: '+ str(ScoreRight),(525,0),color='Green')
```

The text code is for showing the current scores and time. If the position is a bit out of place, you can either change the fontsize or the coordinate.

# Update Function

In the update function, add the followings after the global function. Make sure you add everything so it can work properly.

The rest will be the same as the previous lesson.

```
def update(dt):
    global time,vx,vy,ScoreRight,ScoreLeft,Game_Over,Game_End
    time -=dt
    orb.x +=vx
    orb.y +=vy
```

# Ball Movement

```
#ball movement/velocity
if orb.right >WIDTH or orb.left<0:
    vx = -vx
    sounds.pongwall.play()
if orb.colliderect(Bat1) or orb.colliderect(Bat2):
    vx = -vx
    sounds.paddlepong.play()
if orb.bottom> HEIGHT or orb.top<0:
    vy =-vy
    sounds.pongwall.play()
```

As this is almost similar to the last game, it will have the same principle. The ball will bounce of the wall and also the bat. It will also play a different sfx each time it hits.

Explanation:
If orb.bottom>height or orb.top<0: This is so whenever the ball is less than zero at the top in pixel coordinate, it will bounce off. When the ball is at the bottom, it will become more than the screen, so it will bounce.

# Game Over/ Game End

In this game, we are going to set the condition to trigger the game_over and game_end to true. Game_over is for when the time reach 0. While Game_end is for when the player reach a specific score. The score can be anything you like, as long as you think it can be achieved within the time limit. For testing though, I recommend to set the winning score to 2.

```
if time <0:
    Game_Over =True
if ScoreLeft == 10:
    Game_End =True
if ScoreRight == 10:
    Game_End = True
```

# Scoring

The score will increase by one each time the ball hit the left or right wall. When it hit the right wall, ScoreLeft will go up by 1 and vice versa for the next one.

```
#scoring point
if orb.right >WIDTH:
    ScoreLeft +=1
    sounds.scorepong.play()
if orb.left <0:
    ScoreRight +=1
    sounds.scorepong.play()
```

# Controlling the Paddle/Bat

For controlling the bat, in the same update function, type down the code shown. Because this is a multiplayer game, you will need to set up 2 method of control. One is using WASD and other is the arrow keys. Make sure you use the correct bat.

```
#bat1 and 2 control
if (keyboard.w):
    Bat1.y -=4
if (keyboard.s):
    Bat1.y +=4
if (keyboard.Up):
    Bat2.y -=4
if (keyboard.Down):
    Bat2.y +=4
```

# Function for Player Win

In this function, we are going to setup a condition for when a player wins. When called, it will write the Player 1/2 wins depending on Score. We will use this function inside the draw section after this.

```
def WinScore():
    if ScoreLeft>ScoreRight:
        screen.draw.text('Player 1 Win', (340, 350), color=('white'), fontsize=30,fontname='ari')

    if ScoreRight>ScoreLeft:
        screen.draw.text('Player 2 Win', (340, 350), color=('White'), fontsize=30,fontname='ari')
```

Note: Just copy until the fontsize, no need until the fontname.

# Game Over/End Graphic

Inside the Draw function,start with the game_over. It will draw the game over text, the color and fontsize can be changed if it doesn't fit the display. Once done, we are going to call the Winscore function. So when times up, the text will appear on the screen and also tell us who won. The game_end is also the same. Make sure the if and elif is aligned with the else:

```
if Game_Over:
    screen.draw.text('Game Over', (360, 300), color=('white'), fontsize=50)
    WinScore()
elif Game_End:
    screen.draw.text('Game Ended', (360, 300), color=('white'), fontsize=50)
    WinScore()
```

The Winscore kind of save some space and not make things a bit complex.

# Complete Game Code

```
import pgzrun

WIDTH = 600
HEIGHT = 400

Bat1 = Actor('pongbar')
Bat1.pos = (0,200)
Bat2 = Actor('pongbar')
Bat2.pos =(600,200)
BG = Actor('background')
orb = Actor('pongball')
orb.pos = (300,200)
Mid = Rect((300,0),(10,400))
vx =5
vy =5
ScoreLeft =0
ScoreRight=0
time =5
Game_Over= False
Game_End =False

def draw():
    if Game_Over:
        screen.draw.text('Game Over', (360, 300), color=('white'), fontsize=50)
        WinScore()
    elif Game_End:
        screen.draw.text('Game Ended', (360, 300), color=('white'), fontsize=50)
        WinScore()
    else:
        screen.clear()
        BG.draw()
        screen.draw.filled_rect(Mid,'blue')
        Bat1.draw()
        Bat2.draw()
        orb.draw()
        time_string = str(round(time))
        screen.draw.text('Time: '+time_string,(230,0),color='Purple',fontsize=50)
        screen.draw.text("Score: "+ str(ScoreLeft),(0,0),color='Green')
        screen.draw.text('Score: '+ str(ScoreRight),(525,0),color='Green')
```

```
def update(dt):
    global time,vx,vy,ScoreRight,ScoreLeft,Game_Over,Game_End
    time -=dt
    orb.x +=vx
    orb.y +=vy
    #ball movement/velocity
    if orb.right >WIDTH or orb.left<0:
        vx = -vx
        sounds.pongwall.play()
    if orb.colliderect(Bat1) or orb.colliderect(Bat2):
        vx = -vx
        sounds.paddlepong.play()
    if orb.bottom> HEIGHT or orb.top<0:
        vy =-vy
        sounds.pongwall.play()
    if time <0:
        Game_Over =True
    if ScoreLeft == 10:
        Game_End =True
    if ScoreRight == 10:
        Game_End = True
    #scoring point
    if orb.right >WIDTH:
        ScoreLeft +=1
        sounds.scorepong.play()
    if orb.left <0:
        ScoreRight +=1
        sounds.scorepong.play()
    #bat1 and 2 control
    if (keyboard.w):
        Bat1.y -=4
    if (keyboard.s):
        Bat1.y +=4
    if (keyboard.Up):
        Bat2.y -=4
    if (keyboard.Down):
        Bat2.y +=4
```

```
def WinScore():
    if ScoreLeft>ScoreRight:
        screen.draw.text('Player 1 Win', (340, 350), color=('white'), fontsize=50)
    if ScoreRight>ScoreLeft:
        screen.draw.text('Player 2 Win', (340, 350), color=('White'), fontsize=50)
pgzrun.go()
```

# Complete + Improvement tips.

By the end of this, you should have a functioning Ping Pong game that you can play with someone.
However, you may notice that it got some ‘bugs’ in it. It won’t affect your game to much but I will try to find some ways to debug it.

If you found a way to solve the problem, do tell me how.

You can direct message your teacher and ask your question through Slack Robotene Community or arrange a One-to-One Consultation with your teacher.

# Any Questions?

# Thank you :)