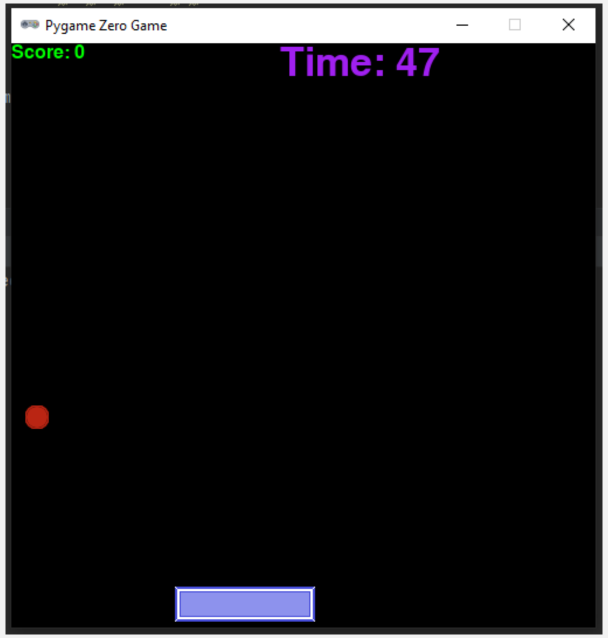
Python Programming
Project 1
Ping Pong Game Level 2.

Presented by Advaspire Team

What we are making

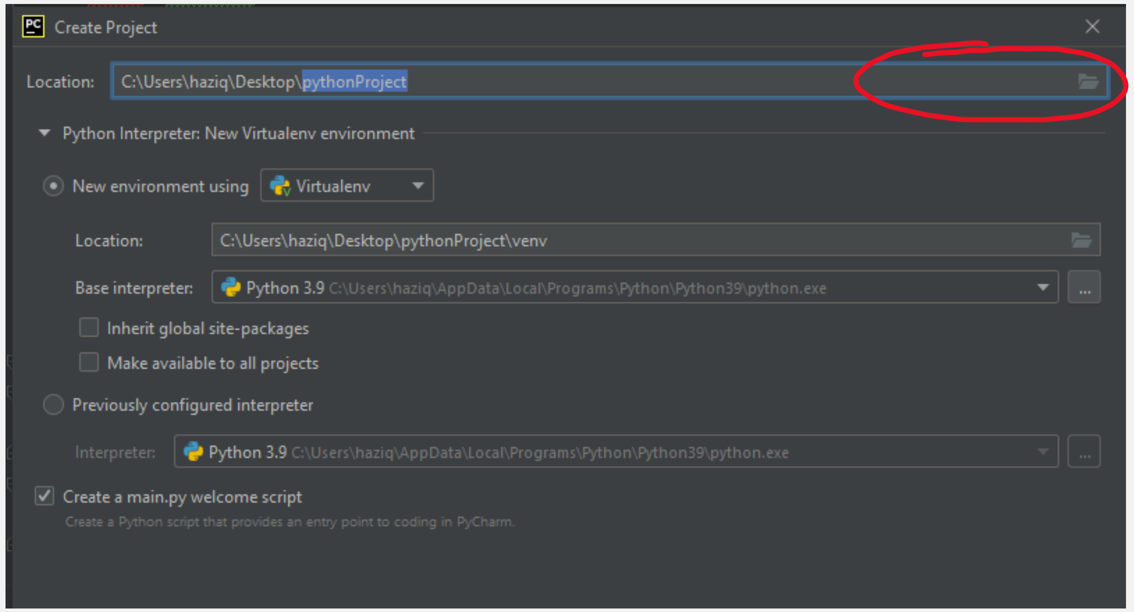
In this lesson, we are going to make a single player ping pong game. Previously, you have learned how the velocity works using ping pong. This time, we are going to make a proper ping pong game. In this project, we are going to use some sprites and sound effect to make it more immersive.

By the end of this lesson, you should have a proper ping pong game.



Setting Up

Create a new project. This time, I recommend to create it at you desktop for easy access. First create a folder for Pyzero projects at desktop, then when creating project make sure to choose that folder.



Create Project

Location: C:\Users\haziq\Desktop\pythonProject

Python Interpreter: New Virtualenv environment

[✓ New environment using] Virtualenv

Location: C:\Users\haziq\Desktop\pythonProject\venv

Base interpreter: Python 3.9 C:\Users\haziq\AppData\Local\Programs\Python\Python39\python.exe

☐ Inherit global site-packages

☐ Make available to all projects

Previously configured interpreter

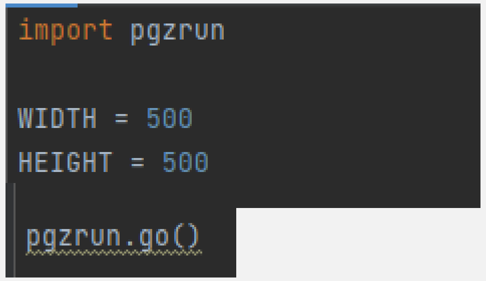
Interpreter: Python 3.9 C:\Users\haziq\AppData\Local\Programs\Python\Python39\python.exe

☑ Create a main.py welcome script

Create a Python script that provides an entry point to coding in PyCharm.

Setting Up Pyzero Games

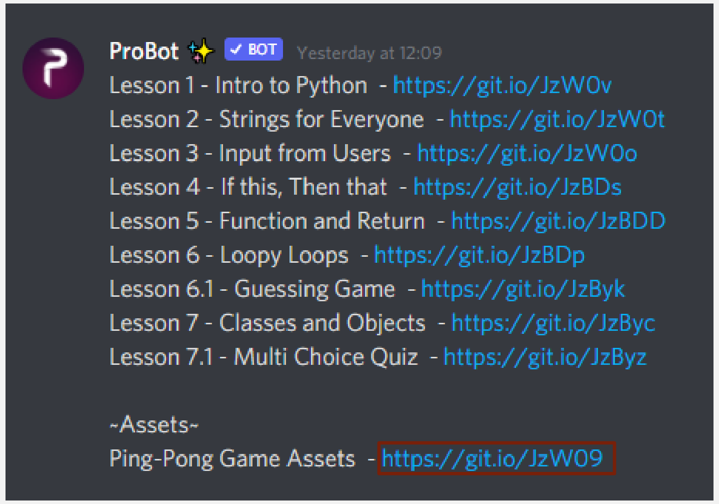
Like usual when using the pyzero, make sure to import it at the start and also put pgzrun.go() at end of the code.


```
import pgzrun

WIDTH = 500
HEIGHT = 500

pgzrun.go()
```

Then, we need to set up the game window. Do the following. We are going for 500x500. You are free to make it bigger but keep in mind that the coordinates for sprites will be different depending on your screen size.

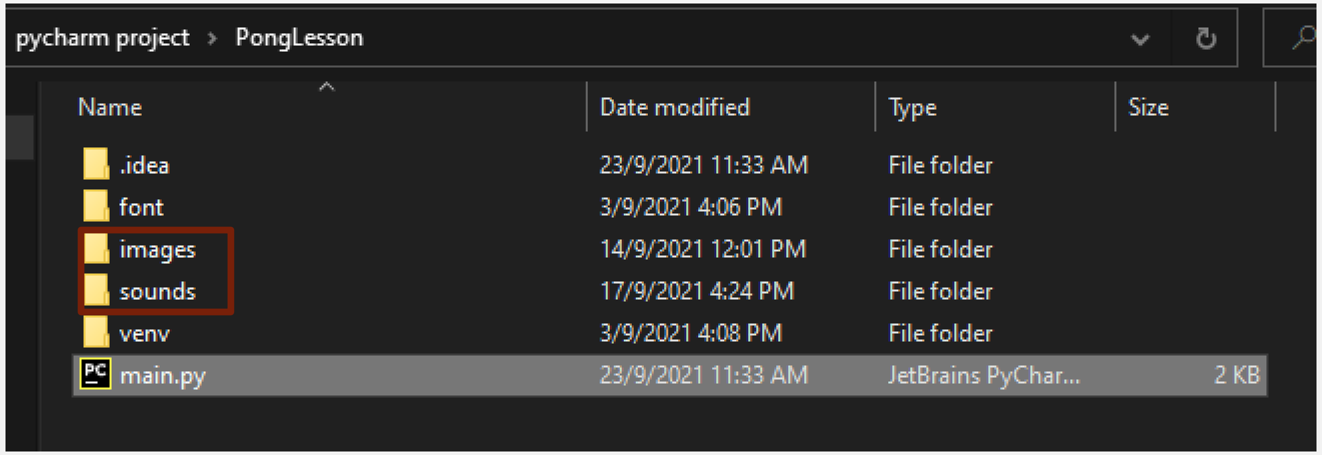
Download Assets

Before we go even further, I need you to go to our discord server and get the assets for the pingpong game. Once downloaded, move the sprites and sound into your project folder. Inside the folder, create 2 folder called images and sounds. Make sure to spell it exactly as shown. Then transfer the assets according to their type.



ProBot BOT Yesterday at 12:09
Lesson 1 - Intro to Python - https://git.io/JzW0v
Lesson 2 - Strings for Everyone - https://git.io/JzW0t
Lesson 3 - Input from Users - https://git.io/JzW0o
Lesson 4 - If this, Then that - https://git.io/JzBDs
Lesson 5 - Function and Return - https://git.io/JzBDD
Lesson 6 - Loopy Loops - https://git.io/JzBDp
Lesson 6.1 - Guessing Game - https://git.io/JzByk
Lesson 7 - Classes and Objects - https://git.io/JzByc
Lesson 7.1 - Multi Choice Quiz - https://git.io/JzByz

~Assets~
Ping-Pong Game Assets - https://git.io/JzW09

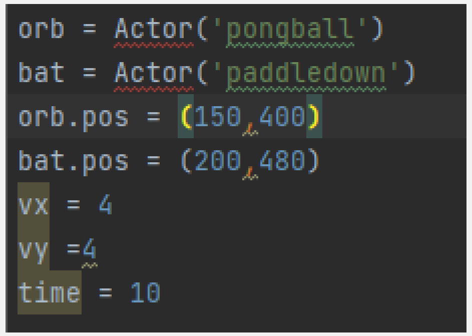


pycharm project › PongLesson

Name	Date modified	Type	Size
.idea	23/9/2021 11:33 AM	File folder	
font	3/9/2021 4:06 PM	File folder	
images	14/9/2021 12:01 PM	File folder	
sounds	17/9/2021 4:24 PM	File folder	
venv	3/9/2021 4:08 PM	File folder	
main.py	23/9/2021 11:33 AM	JetBrains PyChar...	2 KB

Variables

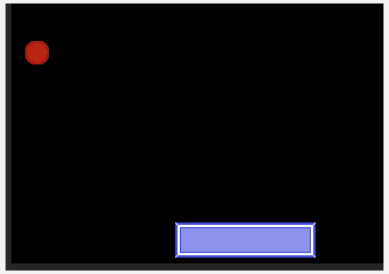
Create the following variables:
orb = Actor(filename)
bat = Actor(filename)
orb.pos = (150,400)
bat.pos = (200,480)
vx = 4
vy = 4
time = 10

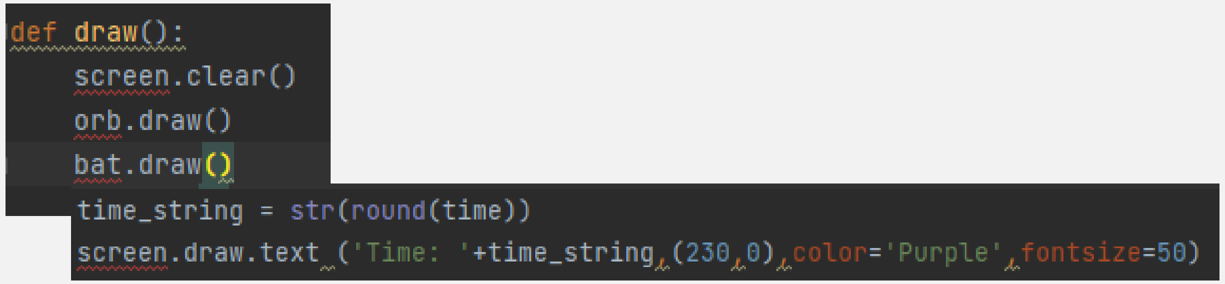


```
orb = Actor('pongball')
bat = Actor('paddledown')
orb.pos = (150,400)
bat.pos = (200,480)
vx = 4
vy =4
time = 10
```

All this variable will be the foundation for our game. Most of the value can be change based on what you want.

Draw the Sprites





```
def draw():
    screen.clear()
    orb.draw()
    bat.draw()
    time_string = str(round(time))
    screen.draw.text ('Time: '+time_string,(230,0),color='Purple',fontsize=50)
```

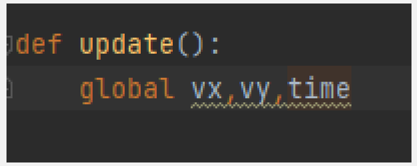
Now we need to create a function called draw. Every graphics that is show on game screen will be programmed here. Once you have the function, start the code with screen.clear() . This is so each time we run it, the game will clear everything on screen. Then the rest is as shown.

If you are using a background image, make sure the bg is after the screen clear. This is because the order of the code will affect which sprites appear first. If you put the bg after the orb, then the orb will be hidden behind the bg.

Update Function

For each movement or activities inside the game, we need to code it inside the update function.

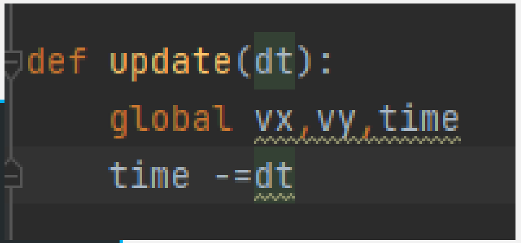
First, we need to insert the global statement. The global statement will carry over any variables or data the entire round. It usually used for things that needs to be constantly updating itself.



```
def update():
    global vx,vy,time
```

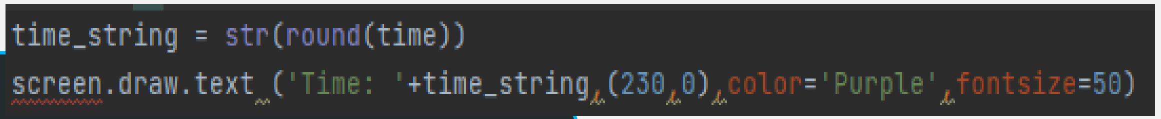
To use it, just type global then the variables that will used in our game that requires movement or update in it. Each time you add a new variable that will affect the game mechanics, make sure to also add it to the global or it wont work.

Timer



```
def update(dt):
    global vx,vy,time
    time -=dt
```

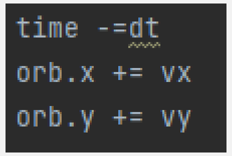
Timer in Pyzero is a bit different than most. To use it, first you need to insert the value dt inside the bracket for update. Then, after the global stuff, type time -= dt so it will decrease for each second. If you want it to increase, change the – to +.



```
time_string = str(round(time))
screen.draw.text ('Time: '+time_string,(230,0),color='Purple',fontsize=50)
```

One other thing, the timer will show the seconds along with decimal points. So that is why in the draw function, we code the time_string, so the time shown is only in whole number, not decimal.

Ping Pong Ball Physics

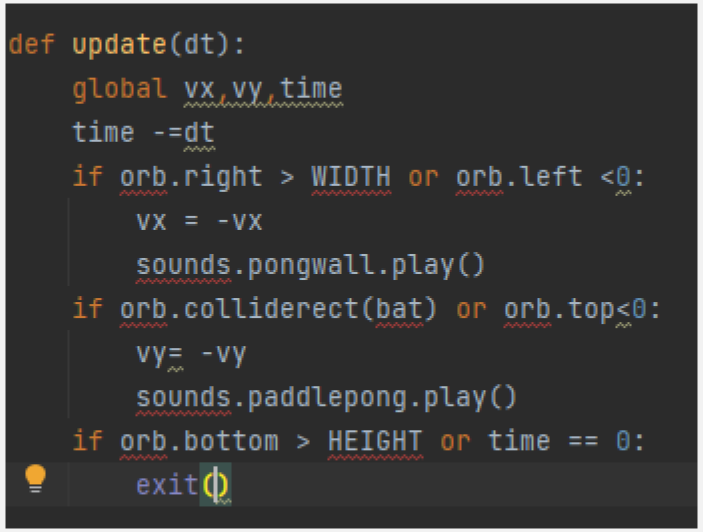


```
time -=dt
orb.x += vx
orb.y += vy
```

For this part, its quite simple as its similar to the previous pong game you learned.

First, we need to set the ball or orb movement. Remember that we are using the velocity for it so after the time -=dt , type orb.x +=vx and orb.y += vy. This will make the orb move.

Gameplay



```
def update(dt):
    global vx,vy,time
    time -=dt
    if orb.right > WIDTH or orb.left <0:
        vx = -vx
        sounds.pongwall.play()
    if orb.colliderect(bat) or orb.top<0:
        vy= -vy
        sounds.paddlepong.play()
    if orb.bottom > HEIGHT or time == 0:
        exit()
```

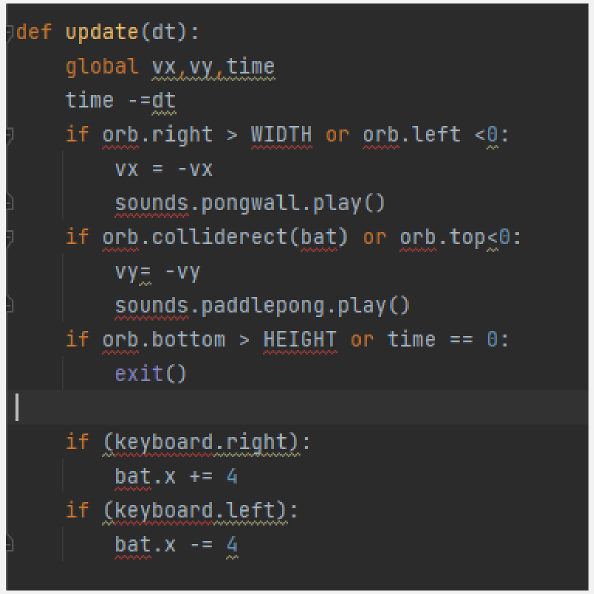
For the next part, we are going to program so that when the ball hits the wall, it will bounce back to the other side. So there will be around 3 set of code for each direction.

The first line will be when the ball touch the wall on left and right. It’s the same like in previous class, just this time we are going to add a sound effect whenever the ball hit the wall. If you want to use other sfx, make sure the format is in wav.

The second line is the same, whenever the ball hit the bat/paddle, it will bounce back the ball and play an sfx. The last line here will be the game over situation for our game. Whenever the ball touch the bottom screen or the time reach 0, the game will exit itself.

Controlling the Paddle/Bat

For controlling the bat, in the same update function, type down the code shown. If you prefer using (wasd) key, just change the right and left with the letter a and d. Also, the bracket is there in case you want to play the game with someone else. Without the bracket, the bats can’t move simultaneously (in case you have 2 bats).

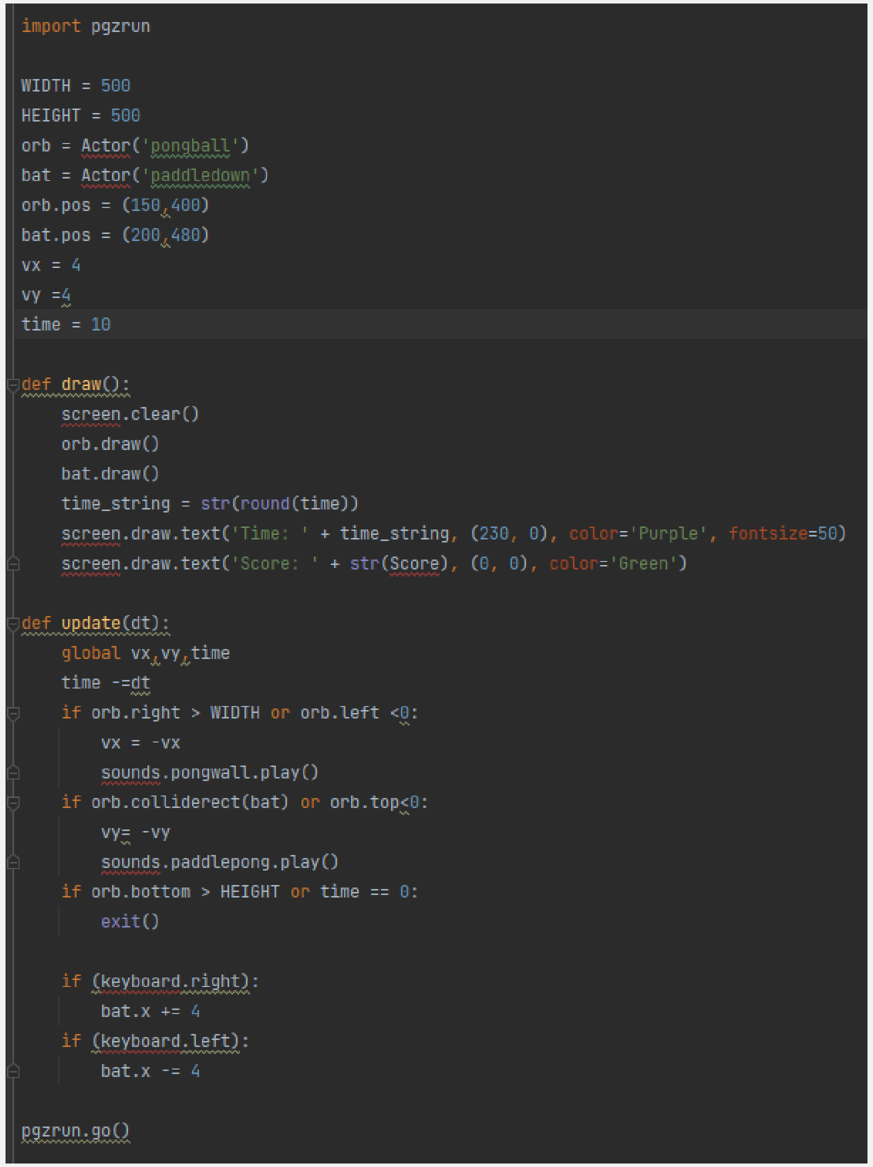


```
def update(dt):
    global vx,vy,time
    time -=dt
    if orb.right > WIDTH or orb.left <0:
        vx = -vx
        sounds.pongwall.play()
    if orb.colliderect(bat) or orb.top<0:
        vy= -vy
        sounds.paddlepong.play()
    if orb.bottom > HEIGHT or time == 0:
        exit()

    if (keyboard.right):
        bat.x += 4
    if (keyboard.left):
        bat.x -= 4
```

Complete + Improvement tips.

By the end of this, you should have a functioning Ping Pong game that you can play around with. There are some more improvement that can be done for the game. Like maybe you don't want the game to exit when the ball touch the bottom or the timer's up. Or even a scoring system.

All of this will be explored in the next lesson, in which we will make the same pong game but with multiplayer functions and some other improvements to the game.



```
import pgzrun

WIDTH = 500
HEIGHT = 500
orb = Actor('pongball')
bat = Actor('paddledown')
orb.pos = (150,400)
bat.pos = (200,480)
vx = 4
vy =4
time = 10

def draw():
    screen.clear()
    orb.draw()
    bat.draw()
    time_string = str(round(time))
    screen.draw.text('Time: ' + time_string, (230, 0), color='Purple', fontsize=50)
    screen.draw.text('Score: ' + str(Score), (0, 0), color='Green')

def update(dt):
    global vx,vy,time
    time -=dt
    if orb.right > WIDTH or orb.left <0:
        vx = -vx
        sounds.pongwall.play()
    if orb.colliderect(bat) or orb.top<0:
        vy= -vy
        sounds.paddlepong.play()
    if orb.bottom > HEIGHT or time == 0:
        exit()

    if (keyboard.right):
        bat.x += 4
    if (keyboard.left):
        bat.x -= 4

pgzrun.go()
```

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

Q&A Any Questions?

Thank you ;)