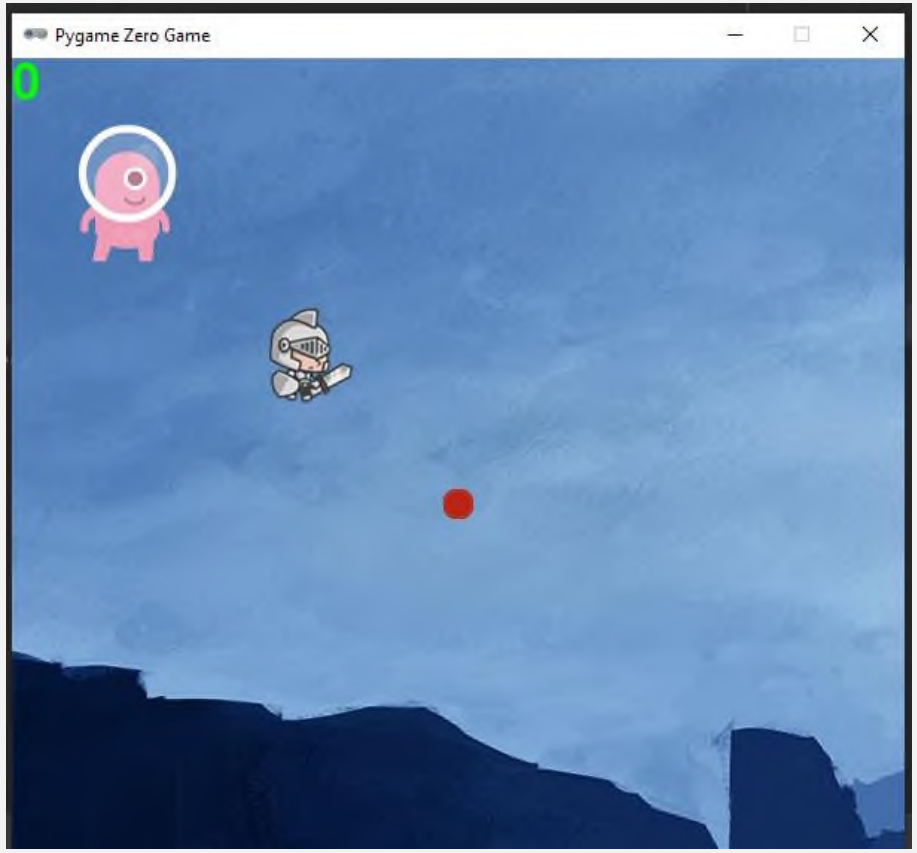
Python Programming
Game Project
Chase Game.

Presented by Advaspire Team

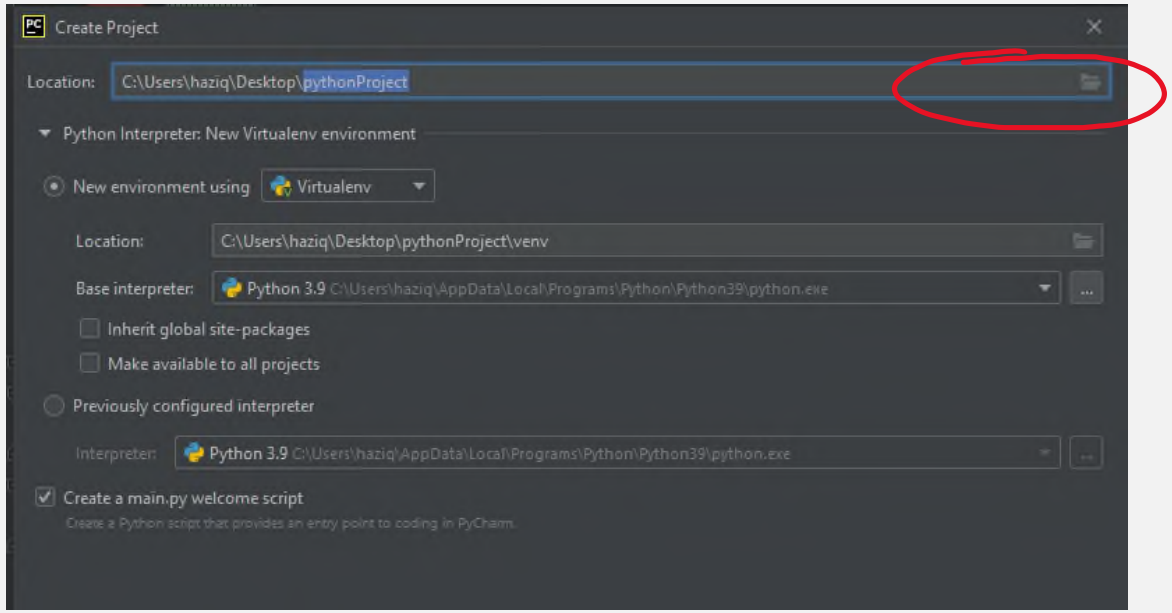
What we are making

In this lesson, we are going to make a chase game. There will be two character, one is the player and the other is the enemy. It is quite similar to how the pac-man game works.



Setting Up

You can use existing project and create a new file inside or you can create a new project.



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. Also import random and if you want to, pygame.

```
import pgzrun

WIDTH = 600
HEIGHT = 600

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

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

To start, make sure to import everything that is shown in the example. Next, create the sprites variable that will be used in this game. Make sure to also have their position in.

Then set a score variable and the time. Score will be 0 and the time can be changed accordingly. For the testing stage, I suggest to set it at 10 seconds so it won't take long to see if the code is working or not.

```
import pgzrun
import random
import pygame
WIDTH = 600
HEIGHT = 600

bg = Actor('background')
player = Actor ('player')
player.pos = (200,200)
enemy = Actor('alien')
coin = Actor ('coinsilver')
coin.pos = (300,300)
score = 0
time= 15
```

Draw the Sprites

```
def draw():
        pygame.display.set_caption('Chase Game')
        screen.clear()
        bg.draw()
        player.draw()
        enemy.draw()
        coin.draw()
```

In the draw function, lets start with naming our game using one of the pygame function. By doing so you will see the name you set as the program name, and not just pygame zero game.

The rest is like usual, just follow the order of codes here. Most of the variable needs to be converted to string first so that is why there are some there.

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. Global here is used to keep track of any variables that will keep on updating while the game is running. That is why the score and time is there. If you didn't put them there, it will return an error.

```
def update(delta):
    global score, time
    time = time - delta
```

On the second line, it will be for the time tracker. To have time in your game, you will need to put the delta or dt inside the update perimeter and also in the time variable.
In the variable, it will be time – delta for decreasing the seconds. If you want to make it increment, make sure to change to 0 in the time variable at the start of our code. Then use + delta.

Player Movement

1

```
if keyboard.d:
    player.x += 4
if keyboard.a:
    player.x -= 4
if keyboard.w:
    player.y -= 4
if keyboard.s:
    player.y += 4
```

2

```
if player.x > WIDTH:
    player.x =0
if player.x <0:
    player.x = WIDTH
if player.y < 0:
    player.y = HEIGHT
if player.y >HEIGHT:
    player.y = 0
```

Now, lets start program the sprites control and movements. This one will be a bit long so make sure to focus and use the # to categorize between control.

In the example 1, it is the usual control mechanic that we use so I will not explain about it.

For example 2, it's the condition for when the player sprites move over the screen border. Whenever it move over, it will appear again at the opposite site of the direction you move it to.

Enemy Movement

As the game name suggest, the enemy will chase the player around the game. The code for it are as follow. The mechanic behind it is that each time the x or y axis for the player is changed, the enemy axis will follow it.

```
if enemy.x < player.x:
    enemy.x = enemy.x + 1
if enemy.x > player.x:
    enemy.x = enemy.x - 1
if enemy.y < player.y:
    enemy.y = enemy.y + 1
if enemy.y > player.y:
    enemy.y = enemy.y - 1
```

Game Mechanics

There are several game mechanics that we are going to add.

The first mechanic we are adding is the collision system and losing. Whenever the enemy touch the player sprite, it will end the game. In an advance version of the game, we can add a Win/Lose screen but that will be for another lesson.

Then, the or operator is so that if either condition is met, it will run the code.

```
if player.colliderect(enemy) or time<= 0:
    exit()
```

Whenever the time reach 0, will exit the game. So for testing stage, it is recommended to set the time variable to around 5 -10 seconds.

Game Mechanics

Remember the coin sprites that we add earlier? We are going to use it for the scoring system for this game. The coin will at first appear at a specific spot each time the game is run. But as each time the player touch it, it will disappear and appear again randomly in the game. The score will also increase in which the increment will be decided by you. If you are unsure, can just follow mine.

```
if coin.colliderect(player):
    coin.x = random.randint(0, WIDTH)
    coin.y = random.randint(0, HEIGHT)
    score = score + 1
```

The random.randint code is to make the system randomize the position for the (x,y). The (0, Width / Height) is to give range for it to randomize.

Displaying Score and Time

To show the score and time left on the game screen, we must first create variable to convert them into strings first. Go to draw (), then after the last line(coin), write the string there. After that, the code is for displaying it on the screen.

```
def draw():
        pygame.display.set_caption('Chase Game')
        screen.clear()
        bg.draw()
        player.draw()
        enemy.draw()
        coin.draw()
        score_string = str(score)
        screen.draw.text('Score : ' + score_string, (0, 0), color='green',fontsize = 50)
        time_string = str(round(time))
        screen.draw.text('Time : '+ time_string, (300, 0), color='green',)
```

The format for the screen.draw.text will be the same most of the time. If the text to small , then just add the fontsize = number to adjust it.

Complete Game Code

```
import pgzrun
import random
import pygame.display

WIDTH = 600
HEIGHT = 600

bg = Actor('background')
player = Actor ('player')
player.pos = (200,200)
enemy = Actor('alien')
coin = Actor ('coinsilver')
coin.pos = (300,300)
score = 0
time= 5
def draw():
    screen.clear()
    pygame.display.set_caption('Chase Game')
    bg.draw()
    player.draw()
    enemy.draw()
    coin.draw()
    score_string = str(score)
    screen.draw.text(score_string, (0, 0), color='green',fontsize = 50)
    time_string = str(round(time))
    screen.draw.text(time_string, (50, 0), color='green')
```

```
def update(delta):
    global score,time
    time = time - delta
    if time <= 0:
        exit()
    if keyboard.d:
        player.x += 4
    if keyboard.a:
        player.x -= 4
    if keyboard.w:
        player.y -= 4
    if keyboard.s:
        player.y += 4
    if player.x > WIDTH:
        player.x =0
    if player.x <0:
        player.x = WIDTH
    if player.y < 0:
        player.y = HEIGHT
    if player.y >HEIGHT:
        player.y = 0

    if enemy.x < player.x:
        enemy.x = enemy.x + 1
    if enemy.x > player.x:
        enemy.x = enemy.x - 1
    if enemy.y < player.y:
        enemy.y = enemy.y + 1
    if enemy.y > player.y:
        enemy.y = enemy.y - 1
    if player.colliderect(enemy):
        exit()
    if coin.colliderect(player):
        coin.x = random.randint(0, WIDTH)
        coin.y = random.randint(0, HEIGHT)
        score = score + 1
        print("Score:", score)
pgzrun.go()
```

Complete + Improvement tips.

By the end of this, you should have a functioning Chase game.
There are a lot more features that can be added to make it more immersive. You will learn how to do it in Chase game Lv2

If you have some more ideas to add in the game, do let me know.

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 :)