Battleship Game using PyGame in Python

Karan Kaushal (101118071)

1. Problem Statement:

Battleship is a shooting game in which the player i.e. spaceship which controls the space from bottom of the screen by firing the laser bullets towards UFO's which comes from the top side. Here the UFO's are trying to reach our player at the bottom and player will fire the laser bullets to stop them. As a result, whenever the bullet hits the enemy i.e. UFO player will get a score for the same. Also, the game gets over when the enemies reaches bottom of the screen where the player is standing.

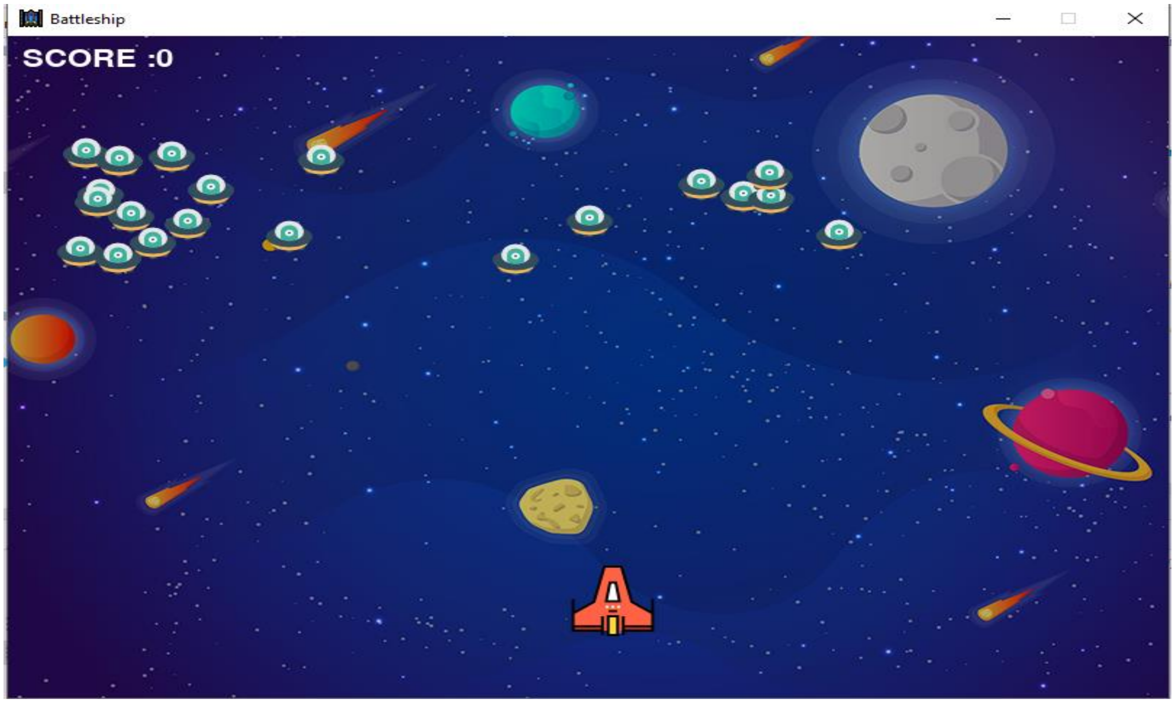


Figure1: Main Playing Screen

2. Problem Analysis: In order to develop this game, I have made use of PyGame library in python.

Now in order to develop this using PyGame I have worked on it as bellow:

A. *Creating Main Window*: Now in order to run this game we need main window first where we will see our output so in order to that I gave done this using pygame and functions. Also, in order to keep window open I have called the same window in while loop in the main function. All the background images seen on the game screen are coming from Images class being define in the code.

```
window = pygame.display.set_mode((800, 600)) ---- Window Dimensions
self.title = pygame.display.set_caption("Battleship")  # Game Title
self.icon = pygame.image.load('icon.png')  # Game Icon
self.background = pygame.image.load('background.png')  # Background Image
pygame.display.set_icon(self.icon)
pygame.display.set_caption("Battleship")
```

B. *Adding Enemies and Player*: Now in order to add player and enemies I have downloaded the images online from various website as PNG and placed in a same file where I have placed by code and then called those images using pycharm functions in python.

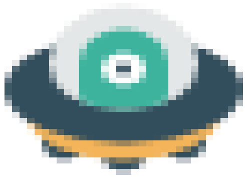


Figure2. Enemy (UFO)

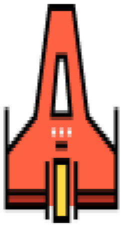


Figure 3. Player (Spaceship)

C. *Creating Multiple Enemies:* After the enemy image in being loaded on the screen we need multiple enemies moving around the game window so in order to show that I have created an empty list for enemies first and then appended it accordingly. Also in order to show movement at random places random library of python is being used.

```
enemy_image_list = []
enemyX = []
enemyY = []
enemyX_change = []
enemyY_change = []
num_of_enemies = 20
for i in range(num_of_enemies):  # Loop for showing random enemies on screen
    enemy_image_list.append(pygame.image.load("ufo.png"))
    enemyX.append(random.randint(0, 735))
    enemyY.append(random.randint(50, 150))
    enemyX_change.append(4)
    enemyY_change.append(40)
```

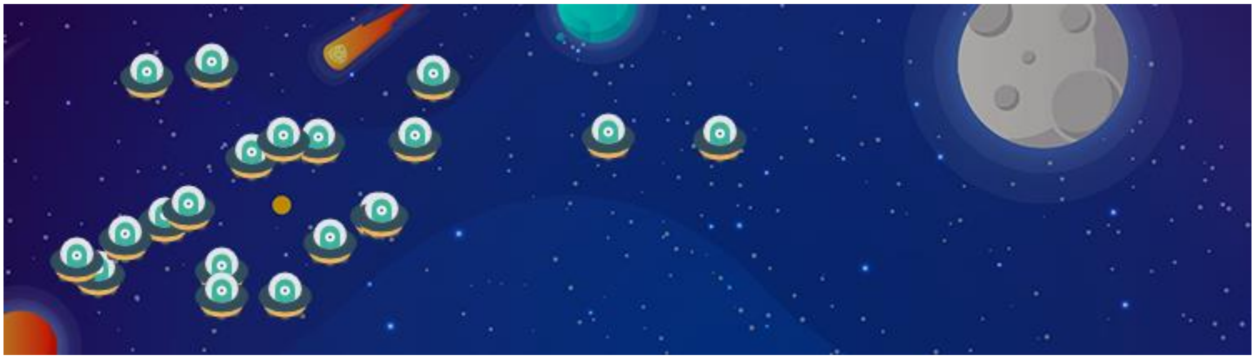


Figure 4: Multiple Enemies on Screen

D. *Boundary for Enemy and player:* Now in order to define the boundary for player and enemies so that they don’t go off from the window from right, left side I have defined the X and Y coordinates for their movement under in main function of the code.

```
root2.shipX += root2.shipX_change
if root2.shipX <= 0:  # creating boundary for ship on screen
    root2.shipX = 0
elif root2.shipX >= 736:
    root2.shipX = 736
```

E. *Bullet and Enemy Collision:* The enemy is killed when it got hit by bullet being fired by player from bottom. So in order to make exact collision I have made use of math library in python. Also the collision is calculated using the below formula:

$$d(P, Q) = sqrt\{p(x2 - x1)2 + (y2 - y1)2\}$$

F. *Adding score and Finishing:* After finishing all the movements and boundary arrangements I have added score option at the top left of the screen which will increment accordingly as player shoots the enemy. Now in order to end the game I have defined the enemy movement dimension as in when the enemy reached the bottom of the screen i.e. near to player the Game will get over showing Game Over message on the screen.

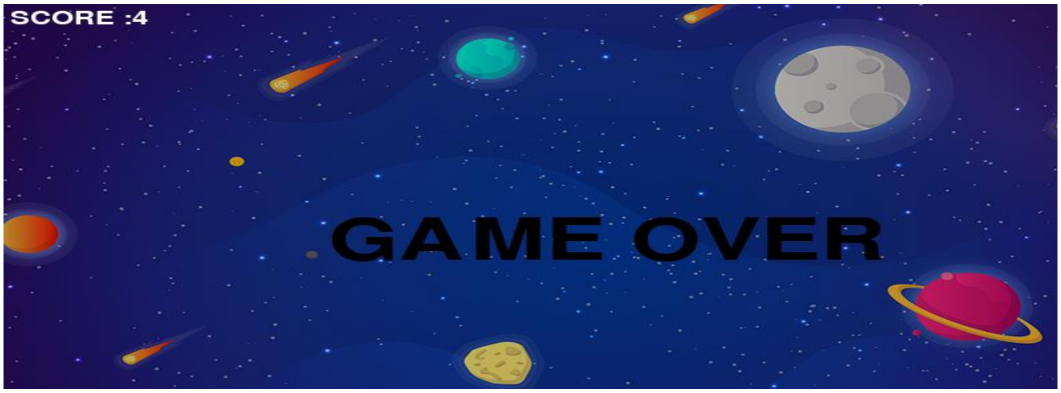


Figure 5: Game Over Screen

3. **Specifications:** As this game is being developed using PyGame library in python so I have used pygame function of controlling the player, bullets.

Inputs: This part of game is handled by keyboard where I have used left and right arrow keys for the player movement on screen.

PRESS LEFT KEY: Player Moves Towards LEFT

PRESS RIGHT KEY: Player Moves Towards RIGHT

SPACEBAR: PRESS To Fire Bullets from spaceship

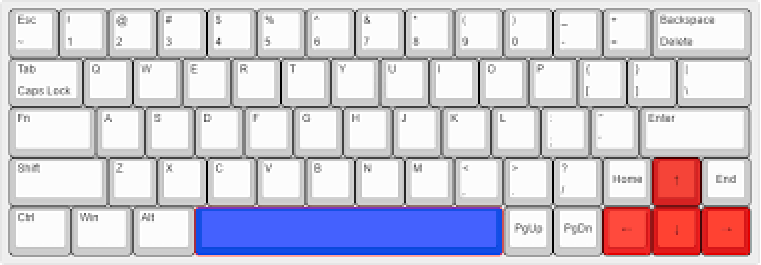
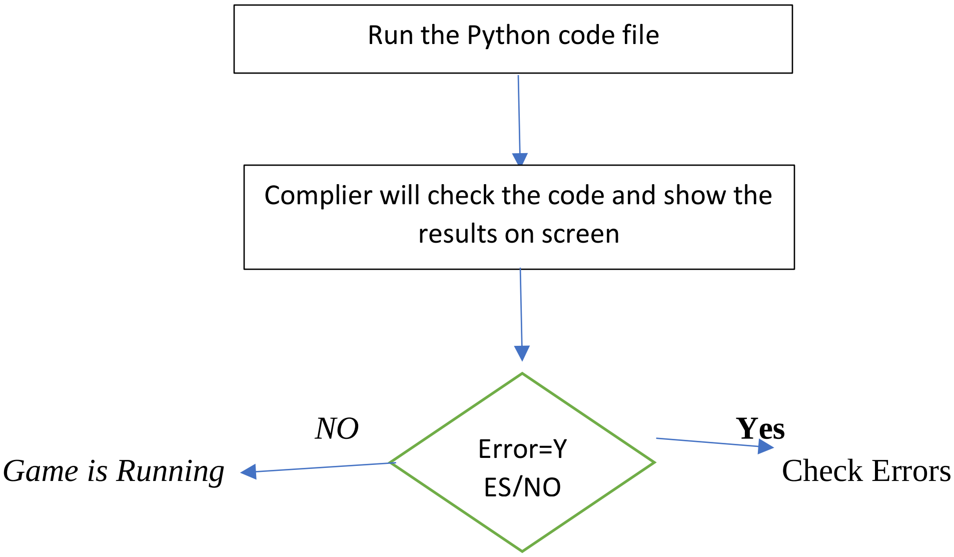
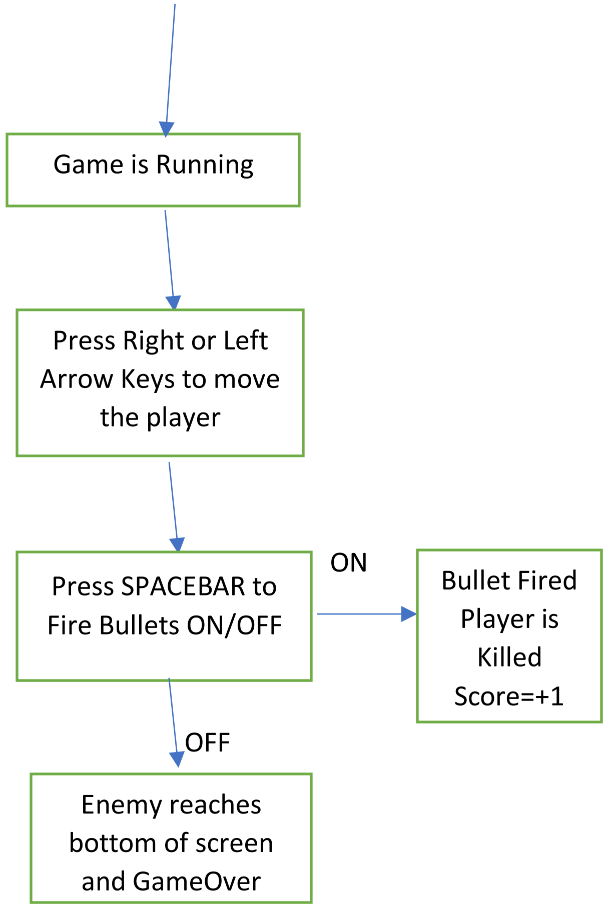


Figure 6: Game Controls

4. **Design:** As per the requirements of code I have made us of OOPS concept wherein I have divided the code into classes and functions. Below Flowchart show how the game is running as an example:

Figure7: Game Flow Chart




Game is Running
Press Right or Left Arrow Keys to move the player
Press SPACEBAR to Fire Bullets ON/OFF
ON
Bullet Fired Player is Killed Score=+1
OFF
Enemy reaches bottom of screen and GameOver

Class Diagram of working code: As per the code is concerned, I have divided python code in to four classes and used these classes in main function where all the main working is taking place. Below is the class diagram for the same:

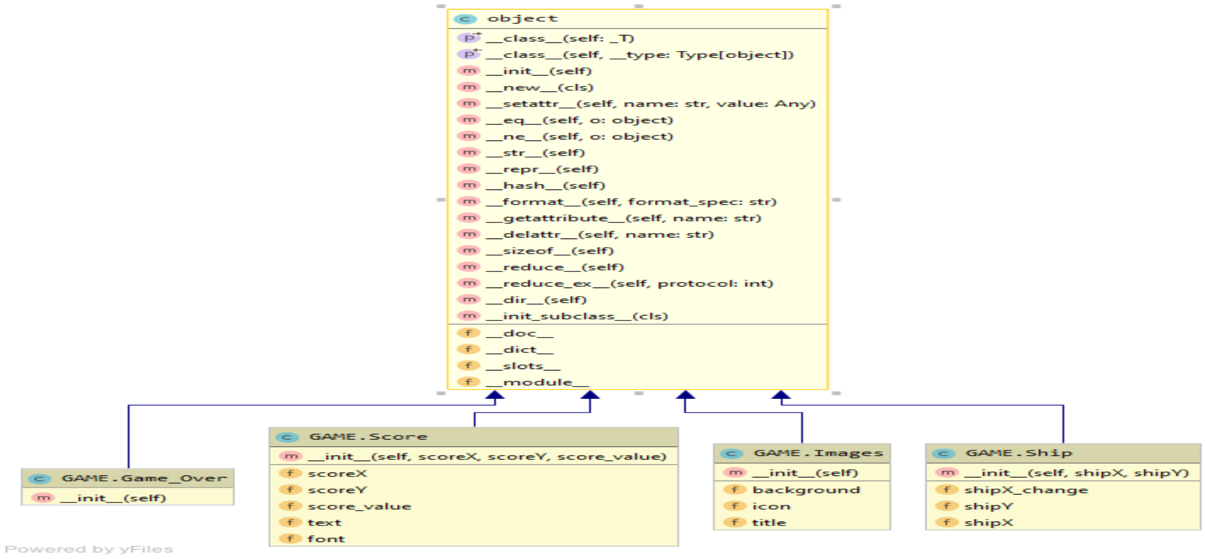


Figure 8: Class Diagram

5. Testing: Testing for this game has been done on 64 bit Windows 10 Laptop powered with AMD processor and 8 GB RAM, In order to run the code, make sure all the images, music files and code should be in one folder. Also all the code is written in PyCharm IDE.

I. *Code is running first time when score=0:* When we first run the python initially the player score is zero and you can see enemies moving on screen.

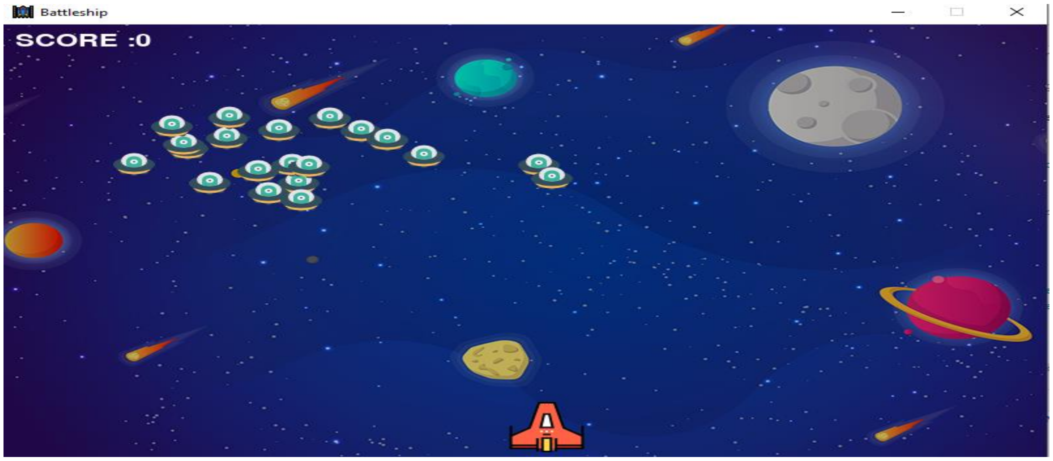


Figure 9: First Time Screen

II. *When Player fires the bullet Score=1:* Now when player fires the bullet and enemy is killed then score will increase by one,

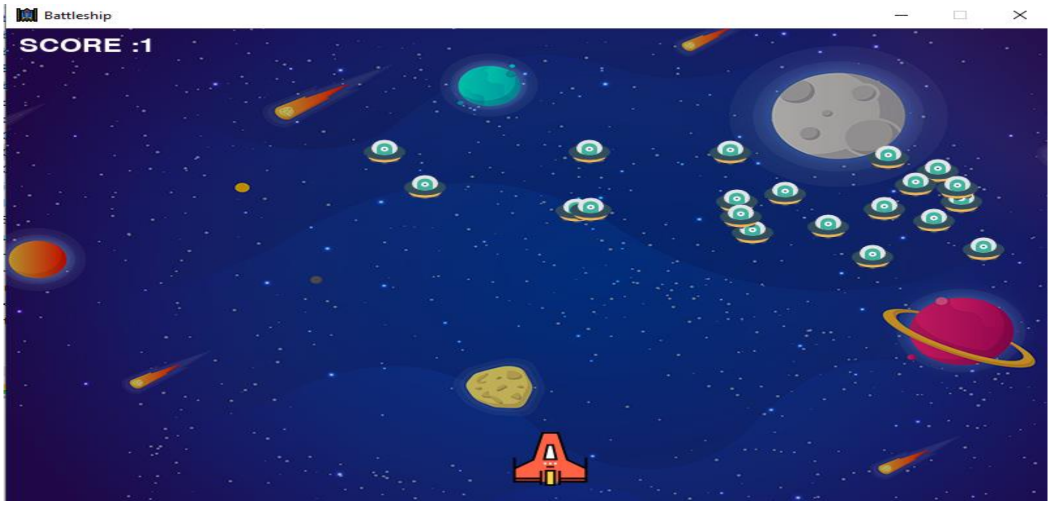


Figure 10: Player Kills One Enemy

III. *Ending Game over:* Now when the enemy reaches the bottom of the screen near to the player game is called Game over as result score is displayed at the top left.

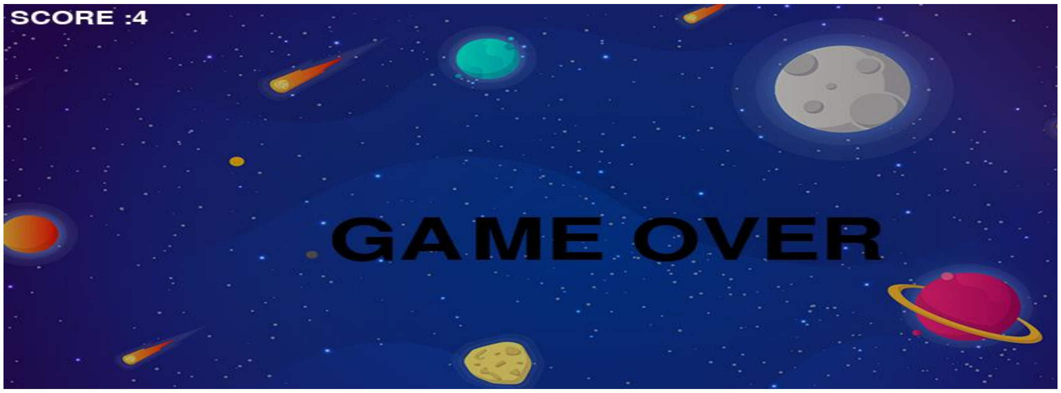


Figure 11: Game Over Screen

6. References:

In order to write the code and for better understanding of pygame I have taken help from below websites:

1. https://www.pygame.org/docs/ : For Pygame Library.

2. https://felgo.com/game-resources/16-sites-featuring-free-game-graphics : For Game Graphics

3. https://www.dl-sounds.com/royalty-free/category/game-film/video-game/ : For Game Music

PLEASE MAKE SURE BEFORE RUNNING THE CODE ALL THE FILES ARE IN SAME FOLDER.