

*“Think. Work. Serve.”*

Semester Project

Department: Electrical/ Computer Engineering

Course: Adv. Programming

Name: Omer Abdel

Title: Space Invader

Date: 16/11/2021

----------------------------------------------------------------------------------------------------

Evaluation:

|  |  |
| --- | --- |
| Sections | Points |
| A. Introduction (20) |  |
| B. Equipment (20) |  |
| C. Procedure (30) |  |
| D. Results and Discussion (30) |  |
| Bonus |  |

Overall:

Comments:

Table of Contents

[Introduction 3](#_Toc88088338)

[Code Explanation 3](#_Toc88088339)

[Libraries Used 3](#_Toc88088340)

[Initializing pygame root window 3](#_Toc88088341)

[Setting Up Player Variables 4](#_Toc88088342)

[Creating Enemies 4](#_Toc88088343)

[Creating Bullet Image 4](#_Toc88088344)

[Score Board 4](#_Toc88088345)

[Main Game Loop 5](#_Toc88088346)

[Important Functions 6](#_Toc88088347)

[EXECUTION 7](#_Toc88088348)

[CONCLUSION 8](#_Toc88088349)

**Space Invader**

**Using: pygame Python**

## **Introduction**

In this task, I have studied and implemented the pygame module to create a basic game which has objects on its screen which can be controlled via keyboard keys by the user. Objects are vector images and control of images is done by pygame module. Moving images work like game players and objects.

## **Code Explanation**

In this section I will go through the lines of my code to show how this game will work.

## **Libraries Used**

import math

import random

import pygame

from pygame import mixer

## **Initializing pygame root window**

# Intialize the pygame

pygame.init()

# create the screen

screen = pygame.display.set\_mode((800, 600))

# Background

background = pygame.image.load('bg.png')

# Sound

mixer.music.load("background.wav")

mixer.music.play(-1)

# Caption and Icon

pygame.display.set\_caption("Space Invader")

icon = pygame.image.load('ufo.png')

pygame.display.set\_icon(icon)

## **Setting Up Player Variables**

# Player

playerImg = pygame.image.load('player.png')

playerX = 370

playerY = 480

playerX\_change = 0

## **Creating Enemies**

# Enemy

enemyImg = []

enemyX = []

enemyY = []

enemyX\_change = []

enemyY\_change = []

num\_of\_enemies = 6

for i in range(num\_of\_enemies):

enemyImg.append(pygame.image.load('enemy.png'))

enemyX.append(random.randint(0, 736))

enemyY.append(random.randint(50, 150))

enemyX\_change.append(4)

enemyY\_change.append(40)

## **Creating Bullet Image**

# Bullet

# Ready - You can't see the bullet on the screen

# Fire - The bullet is currently moving

bulletImg = pygame.image.load('bullet.png')

bulletX = 0

bulletY = 480

bulletX\_change = 0

bulletY\_change = 10

bullet\_state = "ready"

## **Score Board**

# Score

score\_value = 0

font = pygame.font.Font('freesansbold.ttf', 32)

def show\_score(x, y):

score = font.render("Score : " + str(score\_value), True, (255, 255, 255))

screen.blit(score, (x, y))

## **Main Game Loop**

# Game Loop

running = True

while running:

# RGB = Red, Green, Blue

screen.fill((0, 0, 0))

# Background Image

screen.blit(background, (0, 0))

for event in pygame.event.get():

if event.type == pygame.QUIT:

running = False

# if keystroke is pressed check whether its right or left

if event.type == pygame.KEYDOWN:

if event.key == pygame.K\_LEFT:

playerX\_change = -5

if event.key == pygame.K\_RIGHT:

playerX\_change = 5

if event.key == pygame.K\_SPACE:

if bullet\_state is "ready":

bulletSound = mixer.Sound("laser.wav")

bulletSound.play()

# Get the current x cordinate of the spaceship

bulletX = playerX

fire\_bullet(bulletX, bulletY)

if event.type == pygame.KEYUP:

if event.key == pygame.K\_LEFT or event.key == pygame.K\_RIGHT:

playerX\_change = 0

# 5 = 5 + -0.1 -> 5 = 5 - 0.1

# 5 = 5 + 0.1

playerX += playerX\_change

if playerX <= 0:

playerX = 0

elif playerX >= 736:

playerX = 736

# Enemy Movement

for i in range(num\_of\_enemies):

# Game Over

if enemyY[i] > 440:

for j in range(num\_of\_enemies):

enemyY[j] = 2000

game\_over\_text()

break

enemyX[i] += enemyX\_change[i]

if enemyX[i] <= 0:

enemyX\_change[i] = 4

enemyY[i] += enemyY\_change[i]

elif enemyX[i] >= 736:

enemyX\_change[i] = -4

enemyY[i] += enemyY\_change[i]

# Collision

collision = isCollision(enemyX[i], enemyY[i], bulletX, bulletY)

if collision:

explosionSound = mixer.Sound("explosion.wav")

explosionSound.play()

bulletY = 480

bullet\_state = "ready"

score\_value += 1

enemyX[i] = random.randint(0, 736)

enemyY[i] = random.randint(50, 150)

enemy(enemyX[i], enemyY[i], i)

# Bullet Movement

if bulletY <= 0:

bulletY = 480

bullet\_state = "ready"

if bullet\_state is "fire":

fire\_bullet(bulletX, bulletY)

bulletY -= bulletY\_change

player(playerX, playerY)

show\_score(textX, testY)

pygame.display.update()

## **Important Functions**

def game\_over\_text():

over\_text = over\_font.render("GAME OVER", True, (255, 255, 255))

screen.blit(over\_text, (200, 250))

def player(x, y):

screen.blit(playerImg, (x, y))

def enemy(x, y, i):

screen.blit(enemyImg[i], (x, y))

def fire\_bullet(x, y):

global bullet\_state

bullet\_state = "fire"

screen.blit(bulletImg, (x + 16, y + 10))

def isCollision(enemyX, enemyY, bulletX, bulletY):

distance = math.sqrt(math.pow(enemyX - bulletX, 2) + (math.pow(enemyY - bulletY, 2)))

if distance < 27:

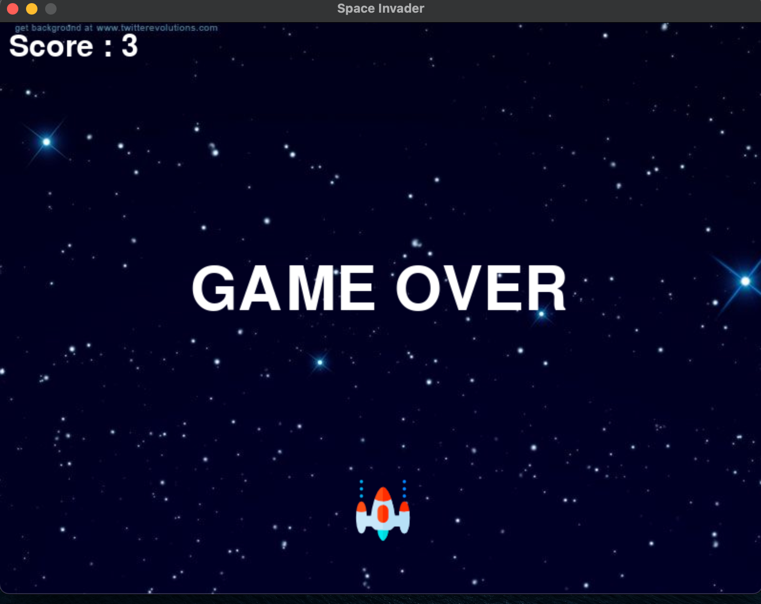
return True

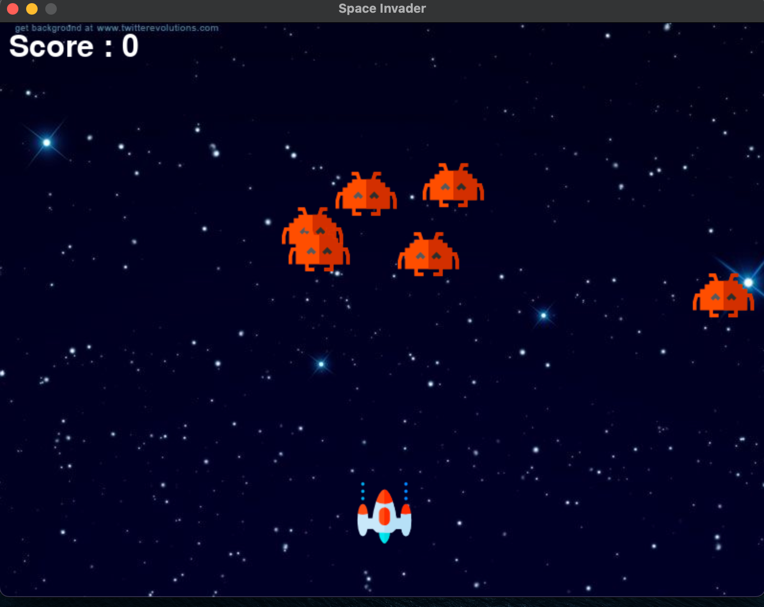
else:

return False

## **EXECUTION**

Each screen shot below shows game interface at different position. The user player is







## **CONCLUSION**

With the above simulations, I have learnt how to debug a Python code and how to write every own code. I faced number of problems during this task when I used pygame module to build a game for the first time ever. Figuring out how to size my window and objects was a difficulty I had to overcome during this project. It frustrated me out a lot because my game would crash due to the size of the objects.

All the game simulation screenshots are attached above with python code.