NEW PRINCE SHRI BHAVANI SENIOR SECONDARY SCHOOL



Computer science

Investigatory PROJECT

SPACE GAME IN PYTHON

**DONE BY:** -

**GRADE:**12 **SEC:** ’B’

**ROLL NO:** 3

ACKNOWLEDGEMENT

I would like to acknowledge great debt I owe to my respective teachers GEETHA PAVANI & SRIDEVI mam, who took this responsibility to provide help and support to complete this project with all the necessary information. I am also indebted to our principal VEENA ILANGO mam for all the enormous support provided. Without their involvement, this project cannot be achieved.

Finally, I want to thank my team members who undertook different roles to make the project enlightened.

Certificate

This is to certify that \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, student of class XII “B” of NEW PRINCE SHRI BHAVANI SENIOR SECONDARY SCHOOL has done his computer science investigatory project on the topic \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ under my supervision. He has given his best to complete this project.

I certify this project up to my expectations and as per the guidelines of CBSE.

Internal examiner External examiner

Principal

INDex

1. AIM OF THE PROJECT
2. INTRODUCTION
3. REQUIREMENTS
4. CODE IN PYTHON
5. OUTPUT
6. CONCLUSIONS
7. BIBILIOGRAPHY

Aim of project:

To create a adventurous game in python using required modules and establishing a interface with MYSQL to display required data to view it or to take input. The python program to successfully play a game using control is achieved.

Introduction

In the project different modules have been used to successfully create a interface, create a gaming environment desirably and many more.

Pygame, random, sql.connector are some of the major modules in the program given below.

One can easily find distinguished feature of the game as it is designed user friendly. Furthermore, controls and login, scores of account are mentioned to replay the game with the existing account in the database stored.

REQUIREMENTS

|  |  |
| --- | --- |
| **Processor (CPU)** | Intel Core i3 (sixth generation or newer) or equivalent |
| **Operating**  **System** | Microsoft windows 7 x32bit  Keyboard, mouse |
| **Memory** | 4 GB RAM |
| **Storage** | 256 GB internal storage drive |
|  |  |
| **Other** | 802.11ac 2.4/5 GHz wireless adapter |

CODING IN PYTHON

import pygame

import random

import mysql.connector

#SQL CONNECTIONS

my\_db = mysql.connector.connect(

host="localhost",

user="root",

password=" ",#databases's password

database=" "#new or existing database

)

mycs = my\_db.cursor()

user\_name = ""

password = ""

High\_score = 0

ID = 0

mode = "HOME"

def login():

while True:

user = input("Enter the user name: ")

pass\_ = input("Enter the password: ")

mycs.execute("SELECT \* FROM Space WHERE user = %s AND pass = %s", (user, pass\_))

try:

myrs = mycs.fetchone()

global user\_name, password, High\_score, ID

ID, user\_name, password, High\_score = myrs

break

except TypeError:

print("There is no user with this tag!!!, Press C to create a new user", end=" ")

inp = input().upper()

if inp == "C":

while True:

new\_user = input("Enter the new user name: ")

new\_pass\_ = input("Enter the new password: ")

mycs.execute("Insert into Space(user, pass) values(%s, %s)", (new\_user, new\_pass\_))

my\_db.commit()

if mycs.rowcount == 1:

print("Your record has been added!!, PLEASE DON'T FORGET YOUR USERNAME IS %s AND PASSWORD IS %s"

% (new\_user, new\_pass\_))

break

login()

while True:

print("""\*\*TASKS\*\*

'start' ---> To Start

'leaderboard' ---> To See Leaderboard

'change' ---> Change username or password

'score' ---> To see your score""")

choice = input("Enter what you want to perform: ")

if choice.lower() in "start":

break

if choice.lower() in "leaderboard":

mycs.execute("SELECT user, highscore FROM space ORDER BY highscore DESC")

leaderboard = mycs.fetchmany(5)

print("TOP \t USER\t\t HIGHSCORE")

for num in range(5):

record = leaderboard[num]

user\_, high\_S = record

print(num, "\t", user\_, "\t\t", high\_S)

if choice.lower() in "change":

what = input("What do you want to change USERNAME('1'), PASSWORD('2'): ")

if what == 1:

verify = input("Enter your password for confirmation!!")

if verify == password:

print("You are allowed to change the Username!")

user\_ = input("Enter new Username: ")

mycs.execute(f"UPDATE space SET user = {user\_}")

mycs.commit()

print("Username Changed")

if what == 2:

verify = input("Enter your password for confirmation!!")

if verify == password:

print("You are allowed to change the password!")

pass\_ = input("Enter new password: ")

mycs.execute(f"UPDATE space SET pass = {pass\_}")

mycs.commit()

print("Password Changed")

if choice.lower() in "score":

print("----- YOUR PRESENT HIGH SCORE IS ------")

print(f" {High\_score} ")

print(""""

ENJOY

THE

GAME

!!!!!""")

#PYGAME PART

"""HOME SCREEN"""

"""GAME SCREEN"""

"""DISPLAYS"""

pygame.init()

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

pygame.display.set\_caption(user\_name.title() + "'s Space Adventure")

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

pygame.display.set\_icon(Icon)

background = pygame.image.load('background.jpg')

"""CHARACTERS"""

#player

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

playerX = 370

playerY = 500

playerChangeX = 0

#score

score = 0

scoreType = {"small": 1, "mid": 2, "big": 3}

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

textX = 10

textY = 10

show = True

#gameover

game\_over\_font = pygame.font.Font('freesansbold.ttf', 64)

#high score

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

#alien

alienType = []

alienHp = []

alienImg = []

alienX = []

alienY = []

alienChangeX = []

alienChangeY = []

alienState = []

totalNoOfAliens = 6

alienToBeShown = 1

for i in range(totalNoOfAliens):

limits = random.randint(1, 30)

if limits < 20:

alienImg.append(pygame.image.load('alien1.png'))

alienType.append("small")

alienHp.append(1)

elif limits < 28:

alienImg.append(pygame.image.load('alien2.png'))

alienType.append("mid")

alienHp.append(2)

elif limits <= 30:

alienImg.append(pygame.image.load('alien3.png'))

alienType.append("big")

alienHp.append(3)

alienX.append(random.randint(0, 720))

alienY.append(random.randint(10, 50))

alienChangeX.append(1)

alienChangeY.append(30)

alienState.append("alive")

#laser

laserImg = pygame.image.load('laser.png')

laserX = playerX

laserY = 480

laserChangeY = 2

laserState = "ready"

"""Functions"""

def game\_over():

global show, score, High\_score

if score >= High\_score:

High\_score = score

over\_text = game\_over\_font.render("GAME OVER", True, (230, 0, 115))

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

high\_score\_text = highScore.render("ALL TIME HIGH SCORE : " + str(High\_score), True, (230, 0, 115))

screen.blit(high\_score\_text, (230, 350))

show = False

def show\_score(x, y):

global score, show

out\_score = font.render("SCORE: " + str(score), show, (255, 255, 255)) # (R, G, B) <-- colour

screen.blit(out\_score, (x, y))

def shortest\_distance(x1, y1, x2, y2):

if laserState == "fire":

distance = (((x2-x1)\*\*2) + ((y2-y1)\*\*2))\*\*0.5

if distance <= 25:

return True

else:

return False

def alien(img, x, y):

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

def player(x, y):

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

def fire\_laser(x, y):

screen.blit(laserImg, (x, y + 10))

global laserState

laserState = "fire"

global laserX

laserX = x

"""Mainloop"""

Repeat = True

while Repeat:

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

if mode == "HOME":

for event in pygame.event.get():

if event.type == pygame.QUIT:

Repeat = False

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

Hover\_text1 = hoverText.render("HOVER OVER THE GREEN AREA", True, (230, 0, 115))

Hover\_text2 = hoverText.render("TO PLAY THE GAME", True, (230, 0, 115))

screen.blit(Hover\_text1, (150, 0))

screen.blit(Hover\_text2, (200, 40))

mouse = pygame.mouse.get\_pos()

pygame.draw.rect(screen, (0, 230, 0), (350, 275, 100, 50))

if 450 >= mouse[0] >= 350 and 325 >= mouse[1] >= 275:

pygame.draw.rect(screen, (0, 255, 0), (350, 275, 100, 50))

mode = "GAME"

elif mode == "GAME":

for event in pygame.event.get():

if event.type == pygame.QUIT:

Repeat = False

if event.type == pygame.KEYDOWN:

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

playerChangeX = -1.1

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

playerChangeX = 1.1

if event.key == pygame.K\_SPACE and laserState == "ready":

fire\_laser(playerX, laserY)

if event.type == pygame.KEYUP:

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

playerChangeX = 0

#bullet change

if laserState == "fire":

fire\_laser(laserX, laserY)

laserY -= laserChangeY

for i in range(alienToBeShown):

if shortest\_distance(alienX[i], alienY[i], laserX, laserY):

if alienHp[i] == 1:

alienState[i] = "dead"

alienX[i] = random.randint(0, 720)

alienY[i] = random.randint(10, 50)

limits = random.randint(1, 30)

if limits < 20:

alienImg[i] = pygame.image.load('alien1.png')

alienType[i] = "small"

alienHp[i] = 1

elif limits < 28:

alienImg[i] = pygame.image.load('alien2.png')

alienType[i] = "mid"

alienHp[i] = 2

elif limits <= 30:

alienImg[i] = pygame.image.load('alien3.png')

alienType[i] = "big"

alienHp[i] = 3

alienChangeX[i] = 1

alien(alienImg[i], alienX[i], alienY[i])

laserY = 10

score += scoreType[alienType[i]]

else:

laserY = 10

alienHp[i] -= 1

if laserY <= 10:

laserState = "ready"

laserY = 480

#alien change

for i in range(alienToBeShown):

if alienX[i] >= 730:

alienX[i] = 730

alienChangeX[i] = -1.3

alienY[i] += alienChangeY[i]

if alienX[i] <= 0:

alienX[i] = 0

alienChangeX[i] = 1.3

alienY[i] += alienChangeY[i]

#boundaries

if playerX >= 730:

playerX = 730

if playerX <= 0:

playerX = 0

"""Movements"""

playerX += playerChangeX

player(playerX, playerY)

show\_score(textX, textY)

for i in range(alienToBeShown):

if alienY[i] > 460:

for j in range(alienToBeShown):

alienY[j] = 2000 # to move it out of screen

game\_over()

break

for i in range(alienToBeShown):

alienX[i] += alienChangeX[i]

alien(alienImg[i], alienX[i], alienY[i])

if score <= 10:

alienToBeShown = 1

elif score <= 20:

alienToBeShown = 2

elif score <= 30:

alienToBeShown = 3

laserChangeY = 3

for i in range(alienToBeShown):

if alienChangeX[i] == -1.3:

alienChangeX[i] = -1.6

elif alienChangeX[i] == 1.3:

alienChangeX[i] = 1.6

elif score <= 40:

alienToBeShown = 4

elif score <= 50:

alienToBeShown = 5

laserChangeY = 5

for i in range(alienToBeShown):

if alienChangeX[i] == -1.6:

alienChangeX[i] = -2

elif alienChangeX[i] == 1.6:

alienChangeX[i] = 2

elif score >= 60:

alienToBeShown = 6

laserChangeY = 8

for i in range(alienToBeShown):

if alienChangeX[i] == -2:

alienChangeX[i] = -2.4

elif alienChangeX[i] == 2:

alienChangeX[i] = 2.4

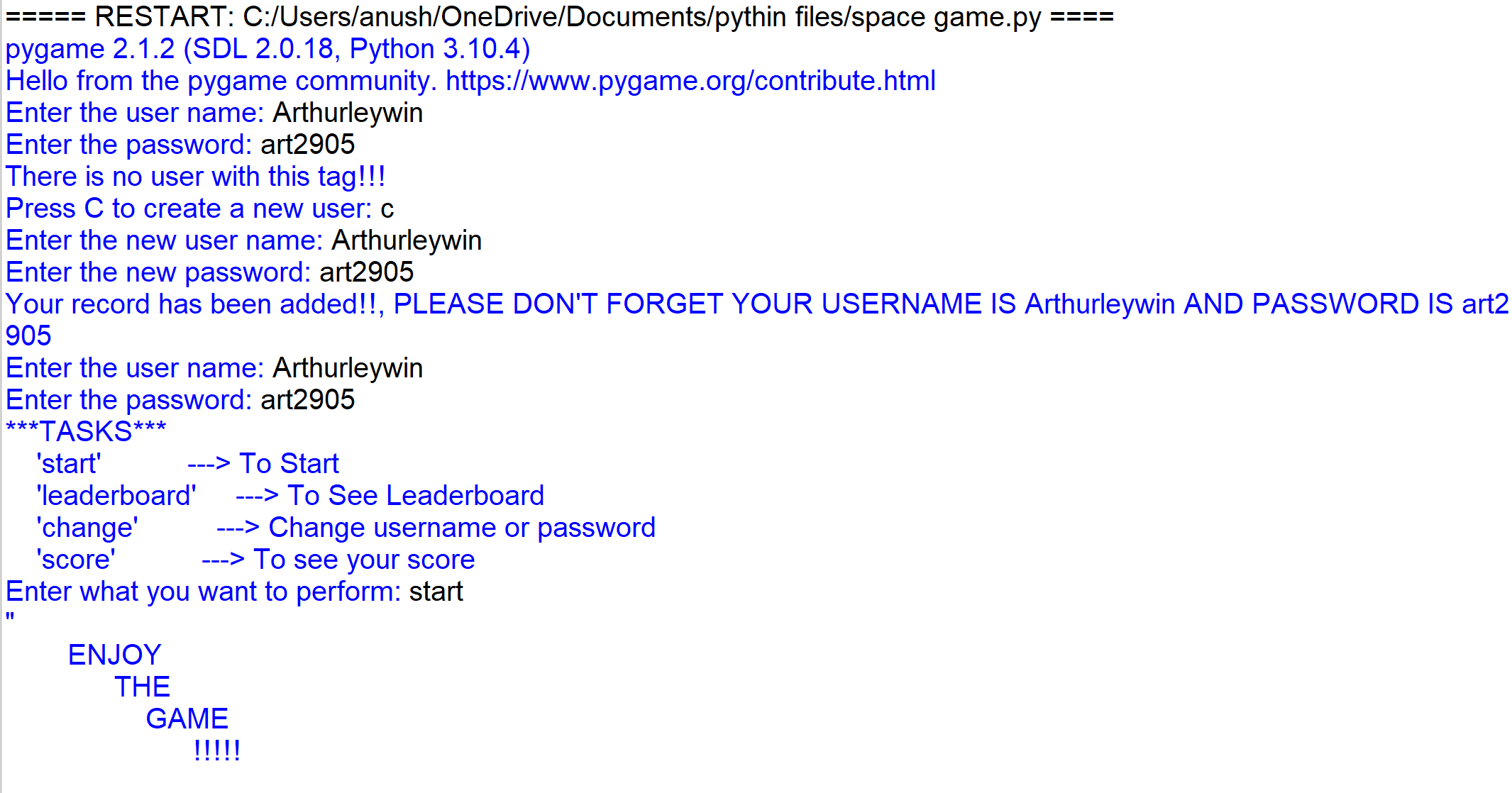
pygame.display.update()

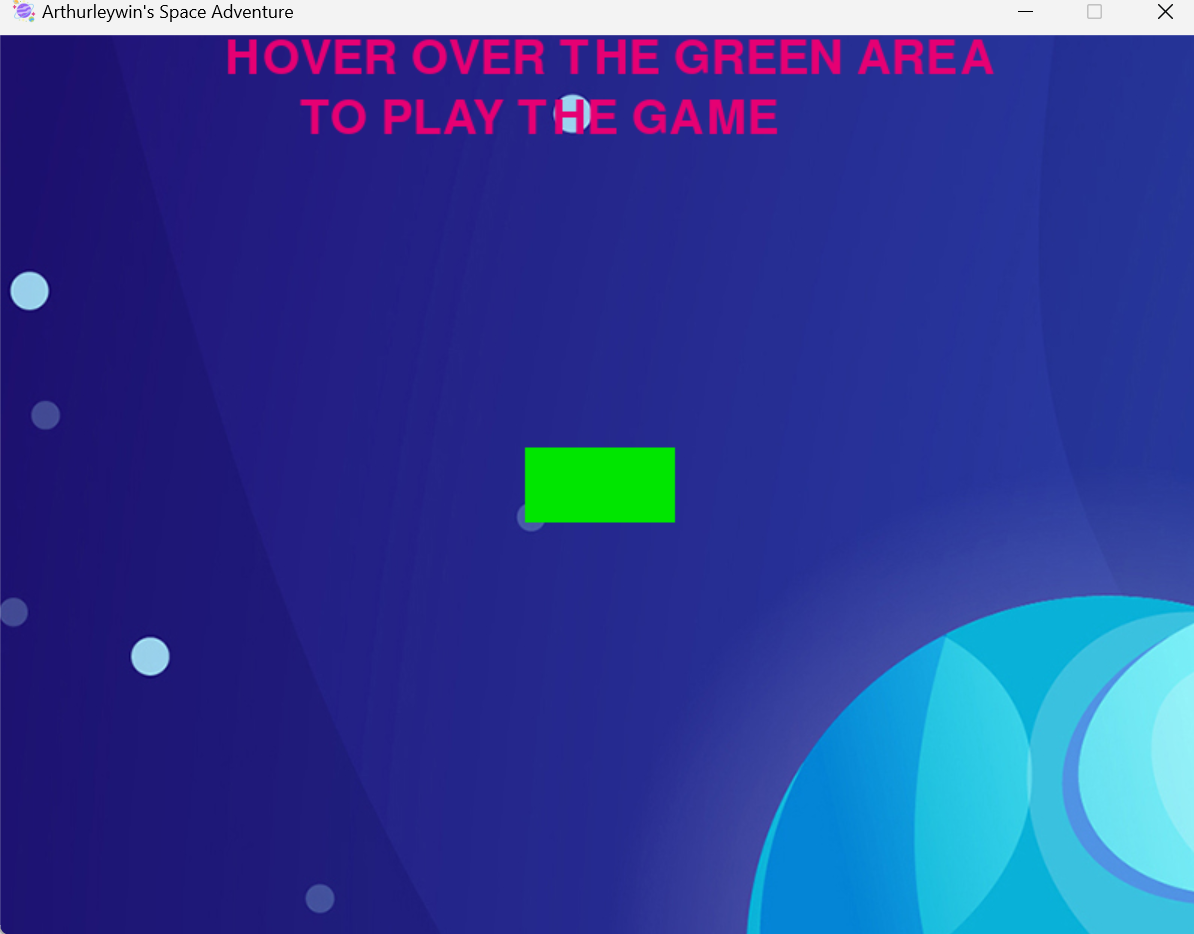
#final updation of sql

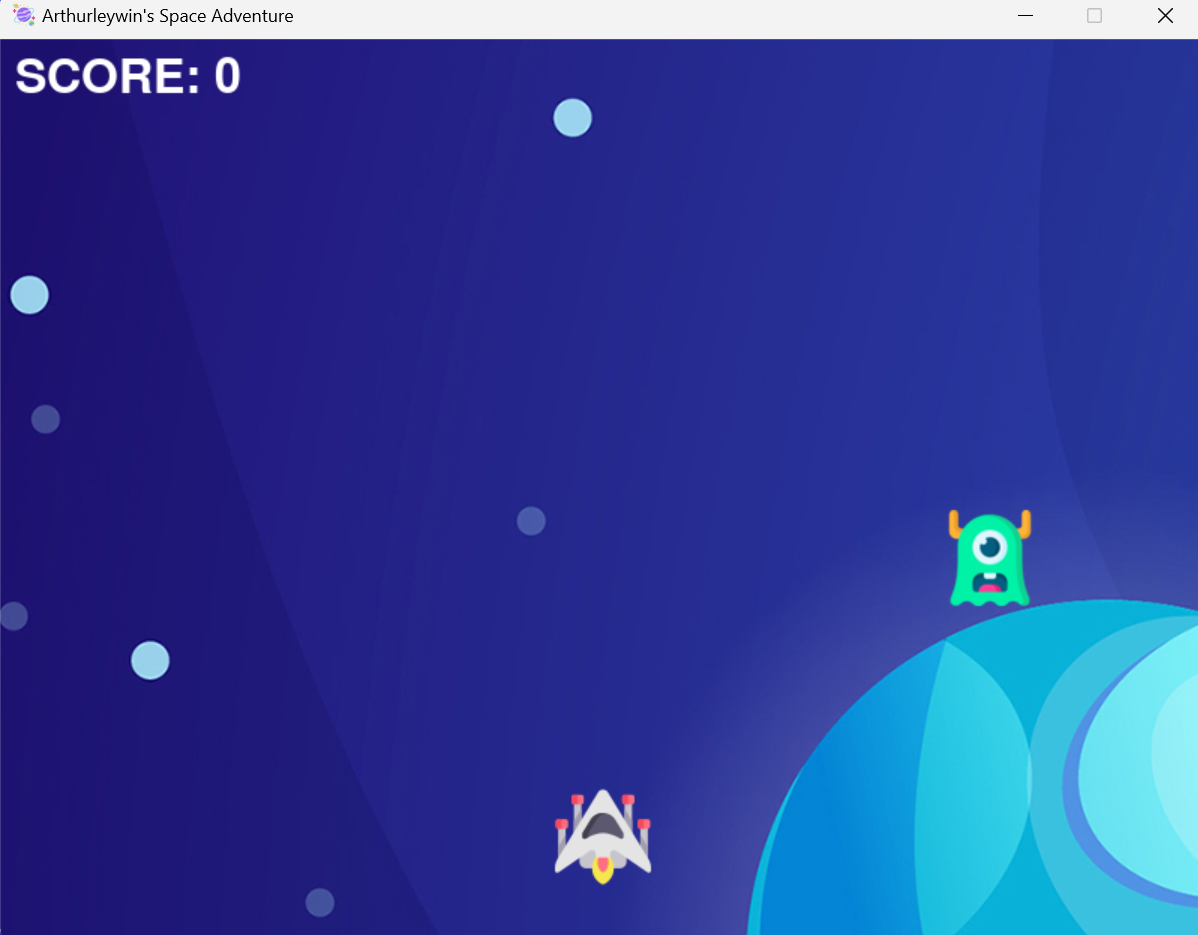
mycs.execute("UPDATE SPACE SET highscore = %s WHERE id = %s" % (High\_score, ID))

my\_db.commit()

my\_db.close()

OUTPUt





CONCLUSIONS

Thus the python and MYSQL interfaced program to run a pygame has been excecuted and the output is verified.

Bibliography

* [www.geeksforgeeks.org](http://www.geeksforgeeks.org)
* Sumita Arora textbook class 12
* [www.umass.edu](http://www.umass.edu)
* en.wikipedia.org