SOMERVILLE SCHOOL , GREATER NOIDA

SESSION : 2020-21

COMPUTER SCIENCE PROJECT

TITLE : SPACE WARS (GAME)

SUBMITTED BY :
NAME : BHAVIKA CHOUDHARY
CLASS/SEC : 12-B

SOMERVILLE SCHOOL , GREATER NOIDA
SESSION : 2020-21

CERTIFICATE

This is to certify that the project entitled “Space Wars” is a record of the bona fide work carried out by BHAVIKA CHOUDHARY, of class XII-B. In partial fulfilment of the requirements in Computer Science prescribed by CBSE for AISSCE 2020-2021 in Somerville School, Greater Noida.

Teacher In-charge
(Mrs. Rachna Agarwal)

Examiner’s Signature

DECLARATION

I hereby declare that the project work entitled “Space Wars”, submitted to SOMERVILLE SCHOOL, GREATER NOIDA for the subject of COMPUTER SCIENCE, under the guidance of Mrs. Rachna Agarwal, is a record of original work done by me in the year 2020-21.

I further declare that this project record or any part of this has not been submitted elsewhere for any other class.

Bhavika Choudhary
Class: XII-B

ACKNOWLEDGEMENT

I would like to express a deep sense of thanks and gratitude to my project guide Mrs. Rachna Agarwal for guiding me immensely through the course of the project. She always evinced keen interest in my work. Her constructive advice and constant motivation have been responsible for the successful completion of this project.

My sincere thanks goes to Dr. Mary Thomas, Our principal, for her co-ordination in extending every possible support for the completion of this project.

I also thank my parents for their motivation and support. I must thank my classmates for their timely help and support for compilation of this project.

Last but not the least, I would like to thank all those who had helped directly or indirectly towards the completion of this project.

BHAVIKA CHOUDHARY
Class : XII-B

INDEX

ABOUT THE PROJECT

This project focusses on the use of modules, importing them in various other modules, learning about the Python Turtle module as well as the connectivity of Python with RDBMS (Relational Database Management System) MySQL.

This project is basically an interactive game named “Space Wars” in which the user can know how to play the game by clicking on “HOW TO PLAY” option in the menu screen. To start the game, one has to simply click on start. After the game has started, it follows an infinite loop which is broken when collision takes place between enemy and player. To kill enemies, player can shoot bullets. Moreover, there are exciting sound effects, which make the game even more interesting. Scores are updated as soon as the player shoots a bullet and it strikes the enemy. One can even compare his/her score from the highscore once the game is ended as the player’s current score as well as highscore is displayed on the screen with the “GAME OVER” text. To quit the game, one can simply exit to main menu and click on “QUIT” button.

This game is extremely interactive and interesting especially for young children and teens. It can elevate one’s mood too !!

WHAT ARE MODULES

Modules refer to a file containing Python statements and definitions.

A file containing Python code, for example: example.py, is called a module, and its module name would be example.

We use modules to break down large programs into small manageable and organized files. Furthermore, modules provide reusability of code.

We can define our most used functions in a module and import it, instead of copying their definitions into different programs.

Types of Modules :

1. Built-in : The Python interpreter has a number of built-in functions. They are loaded automatically as the interpreter starts and are always available. For example, print() and input() for I/O, number conversion functions int(), float(), complex(), data type conversions list(), tuple(), set(), etc. Built-in modules are written in C and integrated with the Python interpreter.
2. User-defined : Functions within a user-defined module can be called from other scripts. A module is a Python script containing related function definitions and Python statements. Every Python script is a module, but we refer to it as a module particularly when we are importing it to use its functions in another script.

MODULES USED

Built-In Modules

1. math module:

This module provides access to the mathematical functions defined by the C standard. These functions cannot be used with complex numbers; use the functions of the same name from the cmath module if you require support for complex numbers. The distinction between functions which support complex numbers and those which don’t is made since most users do not want to learn quite as much mathematics as required to understand complex numbers. Receiving an exception instead of a complex result allows earlier detection of the unexpected complex number used as a parameter, so that the programmer can determine how and why it was generated in the first place. Functions include math.ceil(x), math.sqrt(x), math.pow(a,x), etc.

2. random module:

Python defines a set of functions that are used to generate or manipulate random numbers. This particular type of functions are used in a lot of games, lotteries or any application requiring random number generation.

Some of the functions used are random.randint(), random.randrange(), etc.

3. turtle module:

Turtle is a special feathers of Python. Using Turtle, we can easily draw in a drawing board. First we import the turtle module. Then create a window, next we create turtle object and using turtle method we can draw in the drawing board. Some of the most commonly used functions are turtle.Turtle(), turtle.Screen(), bgcolor(), title(), forward(), backward(), left(), right(), setposition(), shape(), speed(), color(), penup(), pendown(), pensize(), turtle.listen(), turtle.onkeypress(), turtle.onscreenclick(), etc.

4. playsound module:

The playsound module is a cross platform module that can play audio files. This doesn't have any dependencies, it can be simply installed with pip in your virtualenv and run. Functions used are playsound(), etc.

5. mysql.connector module:

MySQL is a Relational Database Management System (RDBMS) whereas the structured Query Language (SQL) is the language used for handling the RDBMS using commands i.e Creating, Inserting, Updating and Deleting the data from the databases. A connector is employed when we have to use MySQL with other programming languages. The work of mysql-connector is to provide access to MySQL Driver to the required language. Thus, it generates a connection between the programming language and the MySQL Server. Functions used are connect(), <connection_object>.cursor(), <cursor>.execute(), <connection_object>.commit(), <connection_object>.rollback(), etc.

User- Defined Modules (with Code)

1. menu_screen.py :

In this module, the menu screen is prepared with the help of turtle module. It has 3 options: “Start”, “How to Play”, & “Quit Game”. By clicking on each of them, one can start playing, open the instruction screen to know how to play the game or exit the game and close the game screen.

It has a function named main_menu() which, when called executes and produces the menu screen.

CODE:

```
import turtle
import instruction_screen
import start_game
def main_menu():
    #main screen
    wn=turtle.Screen()
    wn.title("SPACE WARS")
    wn.bgpic(" C:/Users/pranjal/Desktop/Space Wars/Space Art Wallpaper-
    761517.gif”)

    #title of the game
    main_text=turtle.Turtle()
    main_text.hideturtle()
    main_text.penup()
    main_text.color("white")
    main_text.setposition(-150,250)
    main_text.write("Space Wars", move=False,align="Left", font=("Dead
    Kansas”,40))
```

```
#drawing and writing the menu screen options
 pen1=turtle.Turtle()
 pen1.hideturtle()
 pen1.penup()
 pen1.setpos(-150,0)
 pen1.speed(0)
 pen1.color("white")
 pen1.pendown()
 pen1.fd(300)
 pen1.lt(90)
 pen1.fd(100)
 pen1.lt(90)
 pen1.fd(300)
 pen1.lt(90)
 pen1.fd(100)

 pen1.penup()
 pen1.lt(90)
 pen1.setpos(-150,-120)
 pen1.speed(0)
 pen1.pendown()
 pen1.fd(300)
 pen1.lt(90)
 pen1.fd(100)
 pen1.lt(90)
 pen1.fd(300)
 pen1.lt(90)
 pen1.fd(100)
```

```
pen1.penup()
pen1.lt(90)
pen1.setpos(-150,-240)
pen1.speed(0)
pen1.pendown()
pen1.fd(300)
pen1.lt(90)
pen1.fd(100)
pen1.lt(90)
pen1.fd(300)
pen1.lt(90)
pen1.fd(100)

#writing buttons
pen2=turtle.Turtle()
pen2.hideturtle()
pen2.color("white")
pen2.penup()
pen2.setpos(-50,35)
pen2.pendown()
pen2.write("START",font=("Permanent Marker",20,"bold"))


pen2.penup()
pen2.setpos(-100,-85)
pen2.pendown()
pen2.write("HOW TO PLAY",font=("Permanent Marker",20,"bold"))

pen2.penup()
pen2.setpos(-90,-205)
pen2.pendown()
pen2.write("QUIT GAME",font=("Permanent Marker",20,"bold"))
```

```
    #defining functions of the buttons
    def start_btn(x,y):
        if -150<x<150 and 0<y<100:
            start_game.start()

    def ins_btn(x,y):
        if -150<x<150 and -120<y<-20:
            instruction_screen.ins_screen()
            def exit_to_menu_btn(x,y):
                if 100<x<350 and -300<y<-210:
                    wn.clear()
                    main_menu()
            turtle.onscreenclick(exit_to_menu_btn,1,add=True)
            turtle.listen()



    def quit_btn(x,y):
        if -150<x<150 and -240<y<-140:
            turtle.bye()

    #binding functions to left mouse click
    turtle.onscreenclick(start_btn,1,add=True)
    turtle.onscreenclick(ins_btn,1,add=True)
    turtle.onscreenclick(quit_btn,1,add=True)
    turtle.listen()

main_menu()
```

2. instruction_screen.py :

In this module, instruction screen is prepared with the help of turtle module. Here, we get to know how to play the game and what are its controls. It also has an “Exit to Main Menu” button to go back to menu screen.

It has a function named ins_screen() which, when called produces the instruction screen.

CODE:

```
import turtle
def ins_screen():
    wn=turtle.Screen()
    wn.clear()
    wn.title("SPACE WARS")
    wn.bgpic("C:/Users/pranjal/Desktop/Space Wars/Space Art
     Wallpaper-761517.gif")

    main_text=turtle.Turtle()
    main_text.hideturtle()
    main_text.penup()
    main_text.color("white")
    main_text.setposition(-150,250)
    main_text.write("Space Wars", move=False,align="Left", font=
    (“Dead Kansas”,40))
```

```
pen=turtle.Turtle()
pen.speed(0)
pen.color("white")
pen.hideturtle()
pen.penup()
pen.setpos(-350,100)
pen.pendown()
pen.write("INSTRUCTIONS TO PLAY:",move=True,font=("Dead
Kansas",20))

pen.penup()
pen.setpos(-350,50)
pen.pendown()
pen.write("~ To move the player upwards, use UP ARROW
 KEY.”,font=(“Arial”,15))

pen.penup()
pen.setpos(-350,0)
pen.pendown()
pen.write("~ To move the player downwards, use DOWN ARROW
KEY.”,font=(“Arial”,15))

pen.penup()
pen.setpos(-350,-50)
pen.pendown()
pen.write("~ To move the player towards left, use LEFT ARROW
KEY.”font=(“Arial”,15))
```

```
 pen.penup()
pen.setpos(-350,-100)
pen.pendown()
pen.write("~ To move the player towards right, use RIGHT ARROW
KEY.”font=(“Arial”,15))
pen.penup()
pen.setpos(-350,-150)
pen.pendown()
pen.write("~ To shoot the bullet, use SPACEBAR.",font=("Arial",15))
pen.penup()
pen.setpos(-350,-200)
pen.pendown()
pen.write("~ Each hit on enemy counts as 10 points.”font=(“Arial”,15))
pen.penup()
pen.color("white")
pen.setpos(100,-300)    #Creating EXIT TO MAIN MENU BUTTON
pen.pendown()
pen.fd(250)
pen.lt(90)
pen.fd(50)
pen.lt(90)
pen.fd(250)
pen.lt(90)
pen.fd(50)
pen.lt(90)
pen.penup()
pen.setpos(130,-290)
pen.pendown()
pen.write("Exit to Main Menu",font=("Permanent Marker",17))
```

3. start_game.py :

It is the main program of the game. To play the game, one can run this program.

It has a function named start(), which, when called produce the main game screen and one can then start playing.

CODE:

```
import turtle
import math
import random
from playsound import playsound
import menu_screen
import mysql.connector as ms

mycon=ms.connect(host="localhost",user="root",passwd="bhavika",database="players")
mycursor=mycon.cursor()

bulletstate="ready"
def start():
     #background
    sc=turtle.Screen()
    sc.clear()
    sc.bgcolor("black")
    sc.title("SPACE WARS")
    sc.bgpic("C:/Users/pranjal/Desktop/Space Wars/bgpic.gif")
    #sc.setup(width=1.0,height=1.0)
```

```
#drawing border
proj=turtle.Turtle()
proj.speed(0)
proj.color("yellow")
proj.penup()
proj.setposition(-350,-250)
proj.pendown()
proj.pensize(5)
proj.fd(700)
proj.lt(90)
proj.fd(560)
proj.lt(90)
proj.fd(700)
proj.lt(90)
proj.fd(560)
proj.hideturtle()

#creating player
player=turtle.Turtle()
player.color("red")
player.shape("C:/Users/pranjal/Desktop/Space Wars/player.gif")
player.turtlesize(1.5)
player.penup()
player.speed(10)
player.setposition(0,-220)
player.setheading(90)

playerspeed=20
```

```
#creating bullet
bullet=turtle.Turtle()
bullet.color("yellow")
bullet.shape("triangle")
bullet.penup()
bullet.speed(0)
bullet.setheading(90)
bullet.shapesize(0.5,0.5)
bullet.hideturtle()
bullet.setpos(0,-1200)
bulletspeed=50

#creating enemies
number_of_enemies=5
enemies=[]

for i in range(number_of_enemies):
    enemies.append(turtle.Turtle())

for enemy in enemies:
    enemy.color("green")
    enemy.shape("C:/Users/pranjal/Desktop/Space Wars/enemy.gif")
    enemy.penup()
    x=random.randint(-200,200)
    y=random.randint(150,250)
    enemy.setposition(x,y)
enemyspeed=10
```

```
#Scores
 score="0"
 s=0

 score_pen=turtle.Turtle()
 score_pen.speed(0)
 score_pen.penup()
 score_pen.color("white")
 score_pen.setposition(-350,-300)
 write_score="SCORE:"+score
 score_pen.pendown()

 score_pen.write(“write_score”,move=False,align=“Left”,font=(“Arial”, 20,”bold”))

 score_pen.hideturtle()

 #Game over text
 game_over=turtle.Turtle()
 game_over.penup()
 game_over.color("white")
 game_over.setposition(0,0)
 game_over.hideturtle()
 #exit to main menu button
 pen=turtle.Turtle()
 pen.hideturtle()
 pen.color("white")
 pen.penup()
 pen.setpos(100,-320)
 pen.pendown()
 pen.fd(250)
 pen.lt(90)
 pen.fd(50)
 pen.lt(90)
 pen.fd(250)
 pen.lt(90)
 pen.fd(50)
 pen.lt(90)

 pen.penup()
 pen.setpos(130,-310)

 pen.pendown()
 pen.write("Exit to Main Menu",font=("Permanent Marker",17))

#defining functions
#moving player left, right, up and down
 def move_left():
     x=player.xcor()
     if x>-300:
         x=x-playerspeed
         player.setx(x)
```

```
def move_right():
        x=player.xcor()
        if x<300:
            x=x+playerspeed
            player.setx(x)

    def move_up():
        y=player.ycor()
        if y<280:
            y=y+playerspeed
            player.sety(y)

    def move_down():
        y=player.ycor()
        if y>-220:
            y=y-playerspeed
            player.sety(y)


    #firing bullets
    def fire_bullet():
        global bulletstate
        if bulletstate=="ready":
            bulletstate="fire"
            playsound("C:/Users/pranjal/Desktop/Space Wars/How To
                              Recreate The STAR WARS Laser Sound_00 02 25-00
                              02 27 – oDownloader.mp3”,False)
        x=player.xcor()
        y=player.ycor()+10
        bullet.setposition(x,y)
        bullet.showturtle()
        bulletstate="ready“
 #collision checking
def collision_happens(a,b):
    distance=math.sqrt(math.pow((a.xcor()-b.xcor()),2)+math.pow((a.ycor()
                                   -b.ycor()),2))
   if distance<60:
        return True
    else:
        return False



#keyboard bindings
turtle.listen()
turtle.onkeypress(move_left,"Left")
turtle.onkeypress(move_right,"Right")
turtle.onkeypress(move_up,"Up")
turtle.onkeypress(move_down,"Down")
turtle.onkey(fire_bullet,"space")



def exit_to_menu_btn(x,y):
    if 130<x<380 and -310<y<-260:
        sc.clear()
        menu_screen.main_menu()
turtle.onscreenclick(exit_to_menu_btn,1,add=True)
turtle.listen()
```

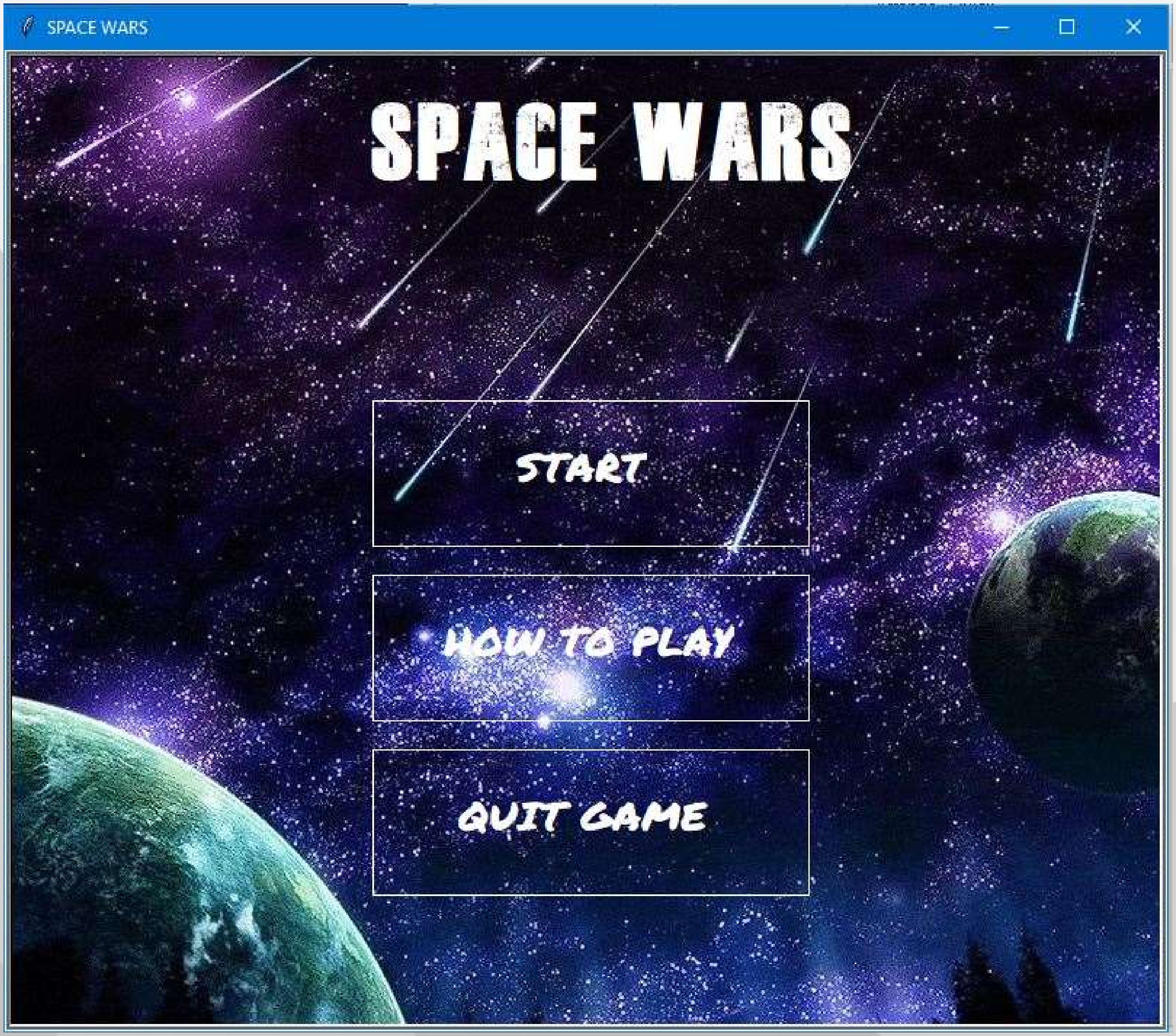
```
#main loop of the game
while True:
    for enemy in enemies:
        x=enemy.xcor()
        x+=enemyspeed
        enemy.setx(x)

        #moving the enemy
        if enemy.xcor()>300 or enemy.xcor()<-300:
            for i in enemies:
                y=i.ycor()
                y-=40
                i.sety(y)
            enemyspeed*=-1

        if enemy.ycor()<-220:
            enemy.hideturtle()
            x=random.randint(-200,200)
            y=random.randint(150,250)
            enemy.setposition(x,y)
            enemy.showturtle()


        #collision checking between enemy & bullet
        if collision_happens(bullet,enemy):
            bullet.hideturtle()
            bulletstate="ready"
            bullet.setposition(-3000,1200)

            playsound("C:/Users/pranjal/Desktop/Space Wars/Star Wars
                                  explosion sound effects part 3_00 00 00-00 00 01
                                  - oDownloader.mp3”,False)
```

```
       enemy.hideturtle()
        x=random.randint(-200,200)
        y=random.randint(150,250)
        enemy.setposition(x,y)
        enemy.showturtle()

        #updating score
        s=eval(score)
        s=s+10
        score=str(s)
        write_score="SCORE:"+score
        score_pen.clear()
        score_pen.write(write_score,move=False,align=“Left”,font=
                                    (“Arial”,20,”bold”))
        score_pen.hideturtle()


#moving bullet
y=bullet.ycor()
y+=bulletspeed
bullet.sety(y)

#border checking bullet
if bullet.ycor()>330:
    bullet.hideturtle()
    bulletstate="ready"
```

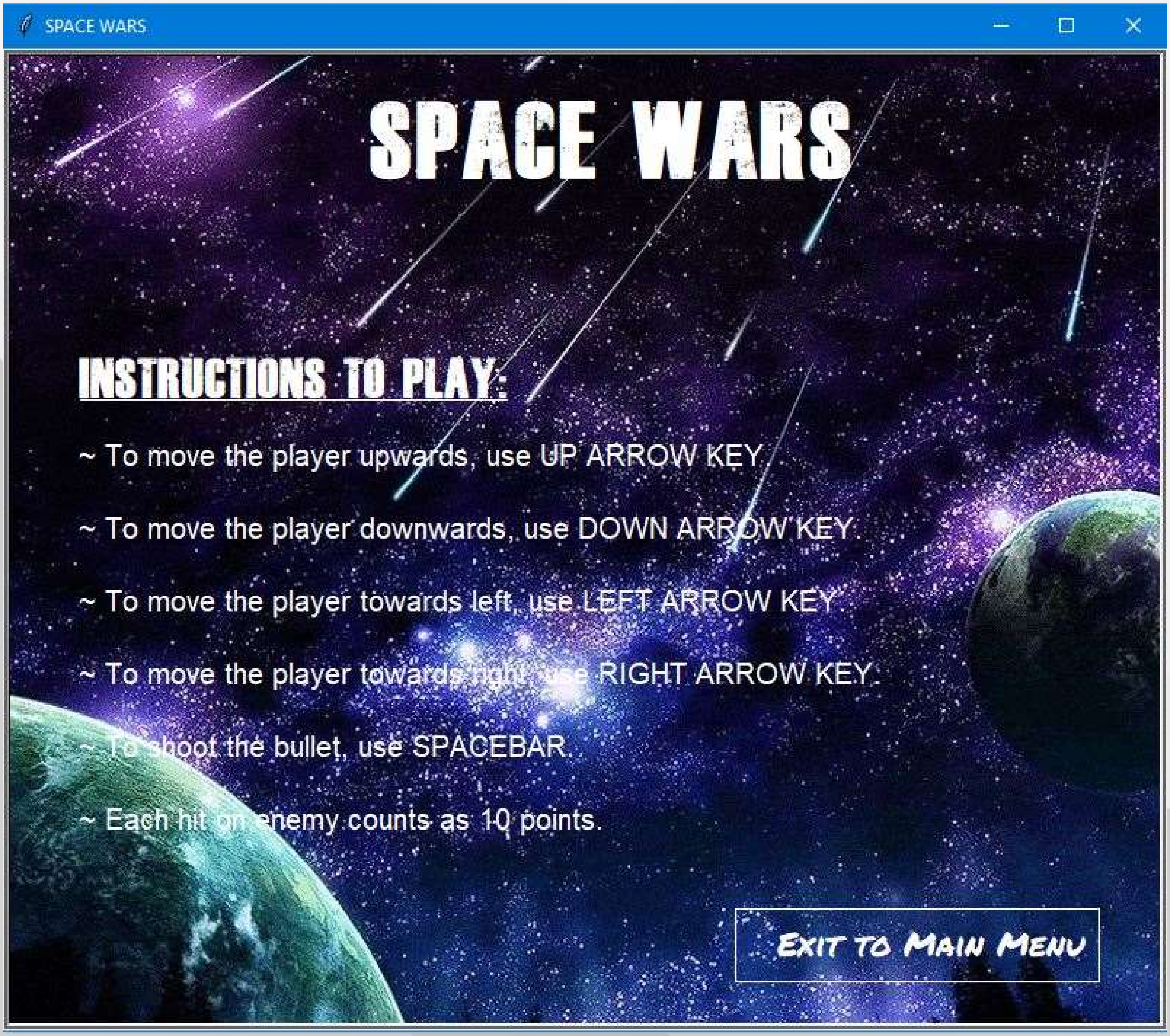
```
#collision checking between player & enemy
if collision_happens(player,enemy):
    playsound("C:/Users/pranjal/Desktop/Space Wars/27_5s_to_2_
                    327_5s_Game_Over_Sound_Effects_All_Soun.mp3”,
                     False)
    enemy.hideturtle()
    player.hideturtle()
    game_over.write("GAME OVER”,move=False,align=“centre”,font=
                              (“Arial”,40,”bold”))
    game_over.hideturtle()
    game_over.penup()
    mycursor.execute("insert into data values({})".format(s))
    mycon.commit()
    mycursor.execute("select max(High_score) from data")
    x=mycursor.fetchall()
    game_over.setpos(-100,-50)
    game_over.pendown()
    game_over.write("Your Score: “+str(s),move=False,align=“Left”,
                              font=(“Arial”,25,”bold”))
    game_over.penup()
    game_over.setpos(-100,-100)
    game_over.pendown()
    game_over.write("Highscore  :  “+str(x[0][0]),move=False,align=
                              “Left”, font=(“Arial”,25,”bold”))
     break
```

OUTPUT SCREENS

MENU SCREEN (Main Screen)



INSTRUCTIONS SCREEN (when "HOW TO PLAY" button is clicked)



SPACE WARS

INSTRUCTIONS TO PLAY:

~ To move the player upwards, use UP ARROW KEY.

~ To move the player downwards, use DOWN ARROW KEY.

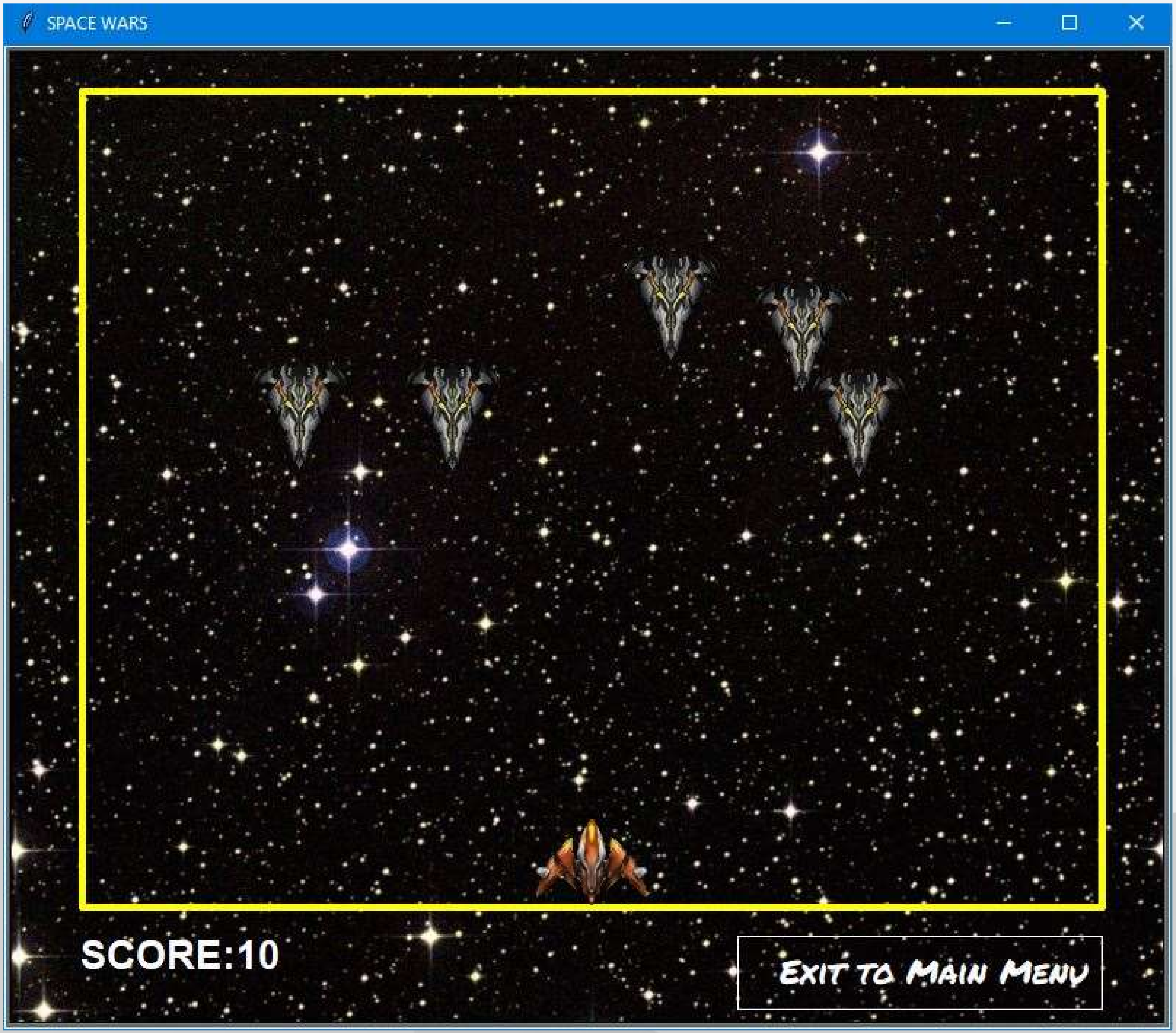
~ To move the player towards left, use LEFT ARROW KEY.

~ To move the player towards right, use RIGHT ARROW KEY.

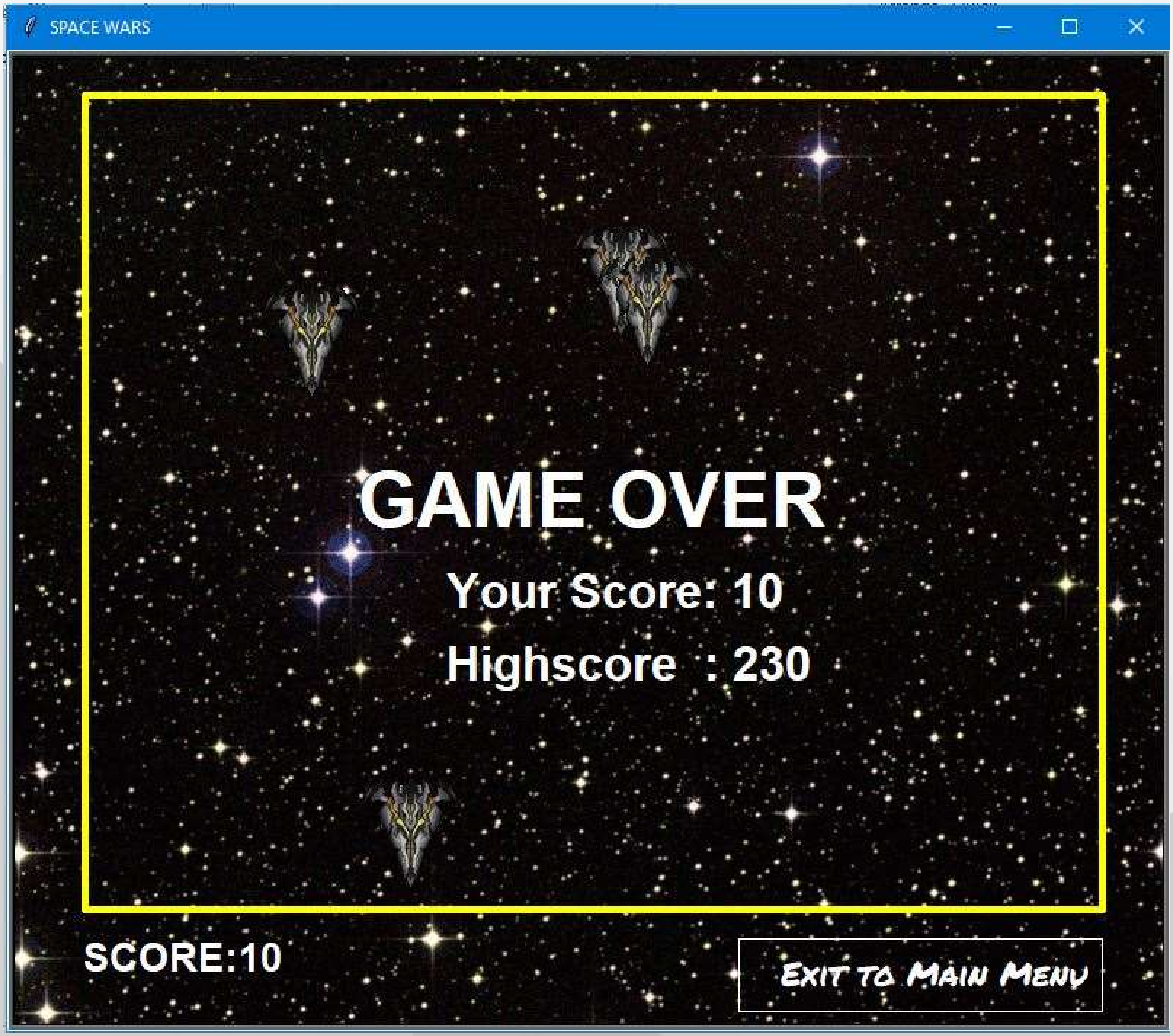
~ To shoot the bullet, use SPACEBAR.

~ Each hit on enemy counts as 10 points.

MAIN GAME SCREEN (when “START” button is clicked)



GAME OVER SCREEN



GAME OVER

Your Score: 10

Highscore : 230

SCORE:10

BIBLIOGRAPHY

1. www.TokyoEdTech, youtube.com
2. www.Programmers' Colloqouy, youtube.com
3. www.Zamzar.com, for file conversion
4. www.Resizeimage.net, Online image resizer
5. www.Kindpng.com, for pictures
6. www.Pixabay.com, for pictures
7. www.oDownloader.com, for sound file conversion
8. www.Geeksforgeeks.com
9. www.Tutorialspoint.com
10. www.Stackoverflow.com
11. www.python.org
12. Textbook for CBSE Class XII Computer Science with Python by Preeti Arora