Space Invader Game

OOPCGL – Mini Project

(Semester - III)

Name of Candidates

- Poonam Chapke (21113)
- Vedant Aher (21103)

SECOND YEAR ENGINEERING

Society for Computer Technology and Research's

PUNE INSTITUTE OF COMPUTER TECHNOLOGY

DHANKAWADI, PUNE – 43

A.Y. 2022-23

========== Second Year Engineering PICT, PUNE-43

Acknowledgement

We take this opportunity to thank all the individuals connected with this mini-project for their useful direction, help and timely support which helped us to complete the project in specific amount of time. We are heartily thankful to our college PICT for giving us this opportunity. We would also like to express great gratitude to our Head of Computer Department Ms. Geetanjali Kale and Parag Jambhulkar Sir for his all-important support, motivation, guidance and helpful suggestions all over the project work.

Aim: To design and implement game using open-source tools.

Objectives: Make use of maximum features of Object-Oriented Programming.

Technologies Used: Python game development library 'PyGame.'

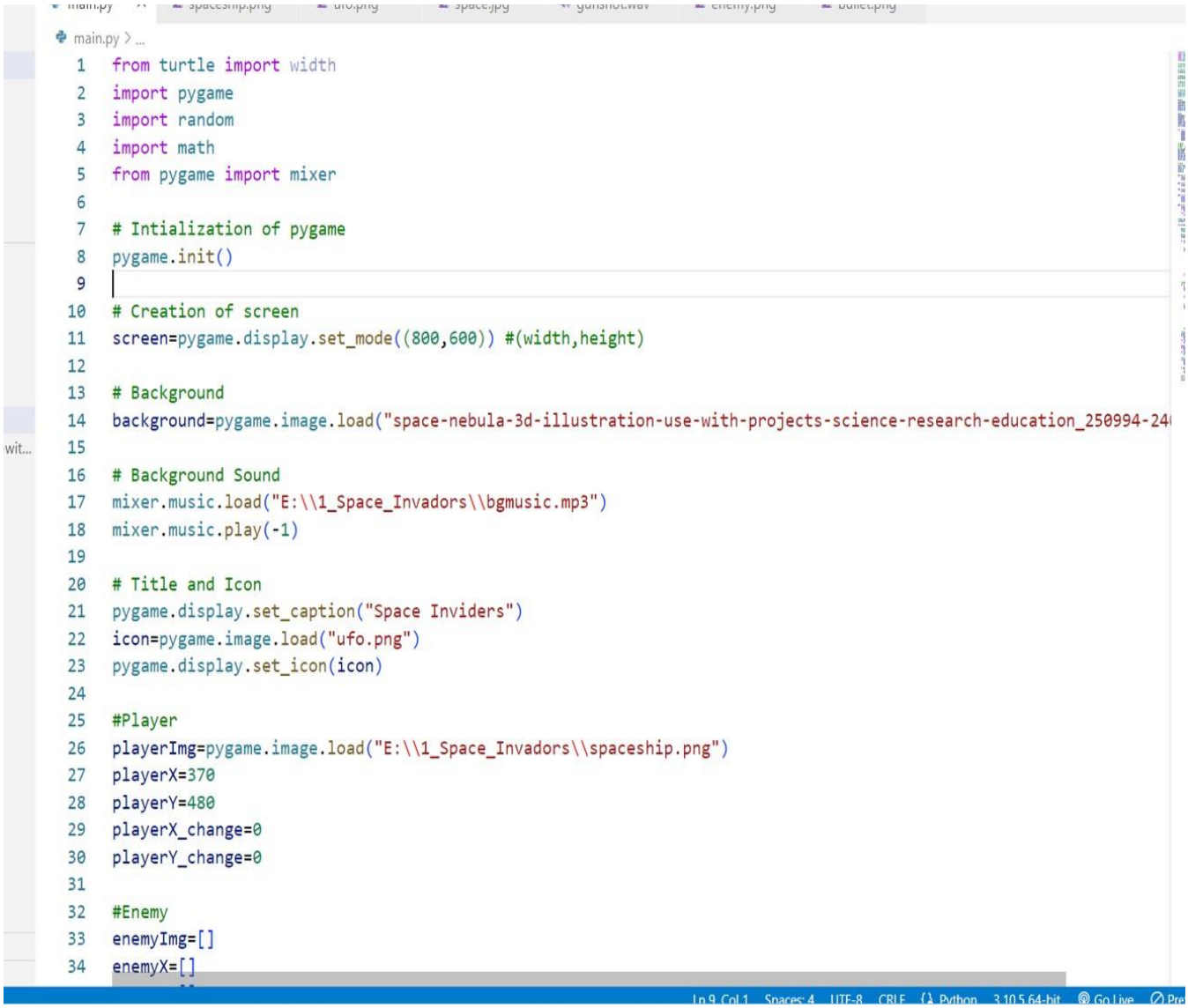
Brief Description:

In this mini project we have developed a 'Space Invaders Game' by using PyGame game development library i.e. PyGame.

Functions Used:

1. **pygame.display:** pygame module to control the display window and screen.
2. **pygame.mixer:** For loading and playing songs.
3. **pygame.event:** For interacting with events and queues.
4. **pygame.font:** Loading and rendering true type fonts.
5. **pygame.image:** Loading saving and transferring of surfaces.

Code and Output Snippets:



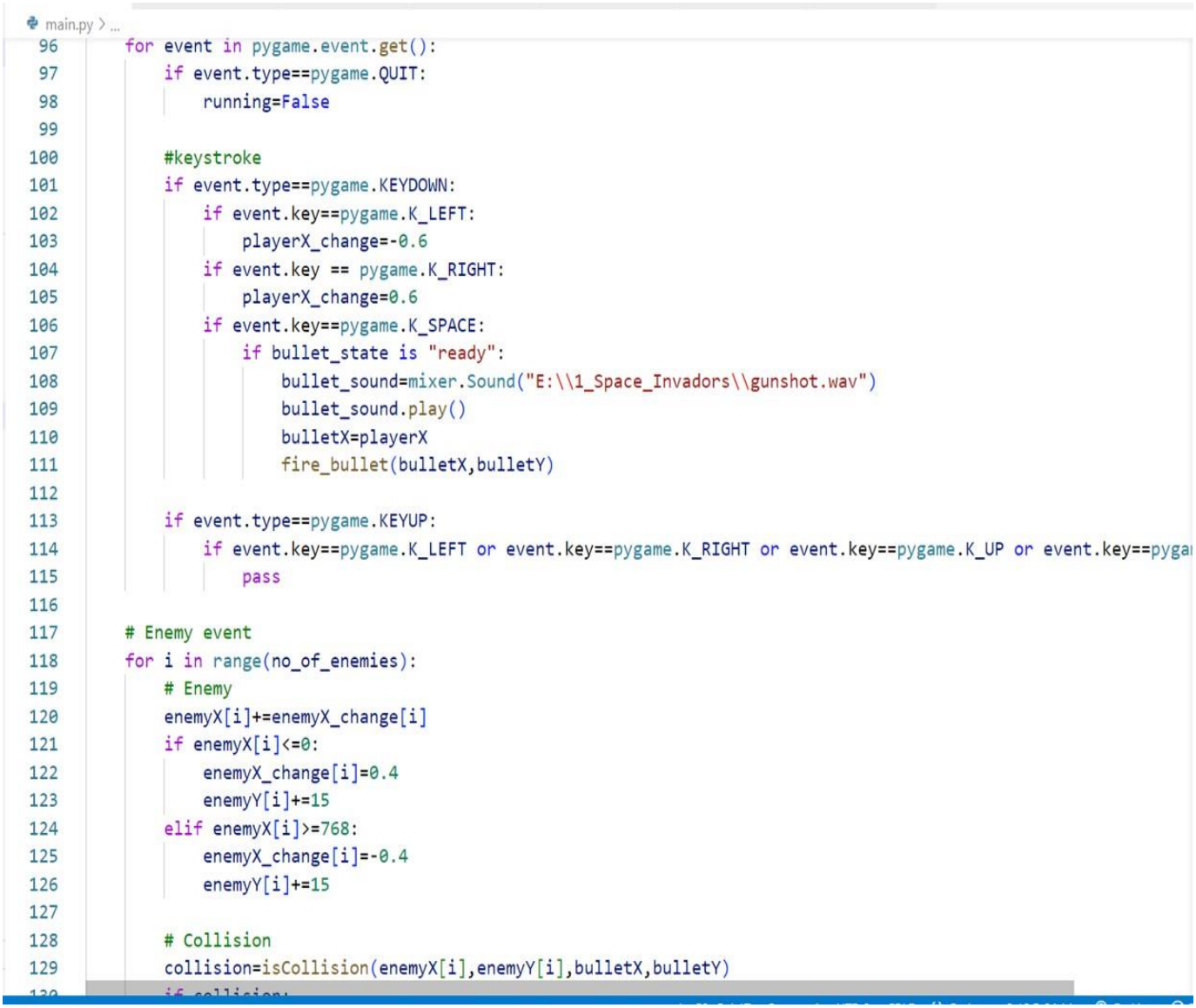
```python
from turtle import width
import pygame
import random
import math
from pygame import mixer

# Intialization of pygame
pygame.init()

# Creation of screen
screen=pygame.display.set_mode((800,600)) #(width,height)

# Background
background=pygame.image.load("space-nebula-3d-illustration-use-with-projects-science-research-education_250994-24

# Background Sound
mixer.music.load("E:\\1_Space_Invadors\\bgmusic.mp3")
mixer.music.play(-1)

# Title and Icon
pygame.display.set_caption("Space Inviders")
icon=pygame.image.load("ufo.png")
pygame.display.set_icon(icon)

#Player
playerImg=pygame.image.load("E:\\1_Space_Invadors\\spaceship.png")
playerX=370
playerY=480
playerX_change=0
playerY_change=0

#Enemy
enemyImg=[]
enemyX=[]
```

main.py > ...

```
33  enemyImg=[]
34  enemyX=[]
35  enemyY=[]
36  enemyX_change=[]
37  enemyY_change=[]
38  no_of_enemies=5
39
40  for i in range(no_of_enemies):
41      enemyImg.append(pygame.image.load("E:\\1_Space_Invadors\\enemy.png"))
42      enemyX.append(random.randint(0,735))
43      enemyY.append(random.randint(0,150))
44      enemyX_change.append(0.8)
45      enemyY_change.append(0.8)
46
47  # Bullet
48  # Ready : you can't see the bullet on the screen
49  # Fire : the bullet is currently moving
50  bulletImg=pygame.image.load("E:\\1_Space_Invadors\\bullet.png")
51  bulletX=0
52  bulletY=480
53  bulletX_change=0
54  bulletY_change=1
55  bullet_state="ready"
56
57  # Score
58  score_value=0
59  font = pygame.font.Font("freesansbold.ttf",32)
60  textX=10
61  textY=10
62
63  def show_score(x,y):
64      score=font.render("Score : "+str(score_value),True,(255,255,255))
65      screen.blit(score,(x,y))
66
```

main.py > ...

```
63  def show_score(x,y):
64      score=font.render("Score : "+str(score_value),True,(255,255,255))
65      screen.blit(score,(x,y))
66
67  def player(x,y):
68      screen.blit(playerImg,(x,y))
69
70  def enemy(x,y,i):
71      screen.blit(enemyImg[i],(x,y))
72
73  def fire_bullet(x,y):
74      global bullet_state
75      bullet_state="fire"
76      screen.blit(bulletImg,(x+16,y+10))
77
78  def isCollision(enemyX,enemyY,bulletX,bulletY):
79      distance=math.sqrt((math.pow(enemyX-bulletX,2))+(math.pow(enemyY-bulletY,2)))
80      if distance < 27:
81          return True
82      else:
83          return False
84
85  # Game loop
86  running=True
87
88  while(running):
89
90      #RGB-Red,Green,Blue
91      screen.fill((0,0,0))
92
93      #Background
94      screen.blit(background,(0,0))
95
96      for event in pygame.event.get():
```



```
main.py > ...
96      for event in pygame.event.get():
97          if event.type==pygame.QUIT:
98              running=False
99
100         #keystroke
101         if event.type==pygame.KEYDOWN:
102             if event.key==pygame.K_LEFT:
103                 playerX_change=-0.6
104             if event.key == pygame.K_RIGHT:
105                 playerX_change=0.6
106             if event.key==pygame.K_SPACE:
107                 if bullet_state is "ready":
108                     bullet_sound=mixer.Sound("E:\\1_Space_Invadors\\gunshot.wav")
109                     bullet_sound.play()
110                     bulletX=playerX
111                     fire_bullet(bulletX,bulletY)
112
113         if event.type==pygame.KEYUP:
114             if event.key==pygame.K_LEFT or event.key==pygame.K_RIGHT or event.key==pygame.K_UP or event.key==pyga
115                 pass
116
117     # Enemy event
118     for i in range(no_of_enemies):
119         # Enemy
120         enemyX[i]+=enemyX_change[i]
121         if enemyX[i]<=0:
122             enemyX_change[i]=0.4
123             enemyY[i]+=15
124         elif enemyX[i]>=768:
125             enemyX_change[i]=-0.4
126             enemyY[i]+=15
127
128         # Collision
129         collision=isCollision(enemyX[i],enemyY[i],bulletX,bulletY)
130         if collision:
```

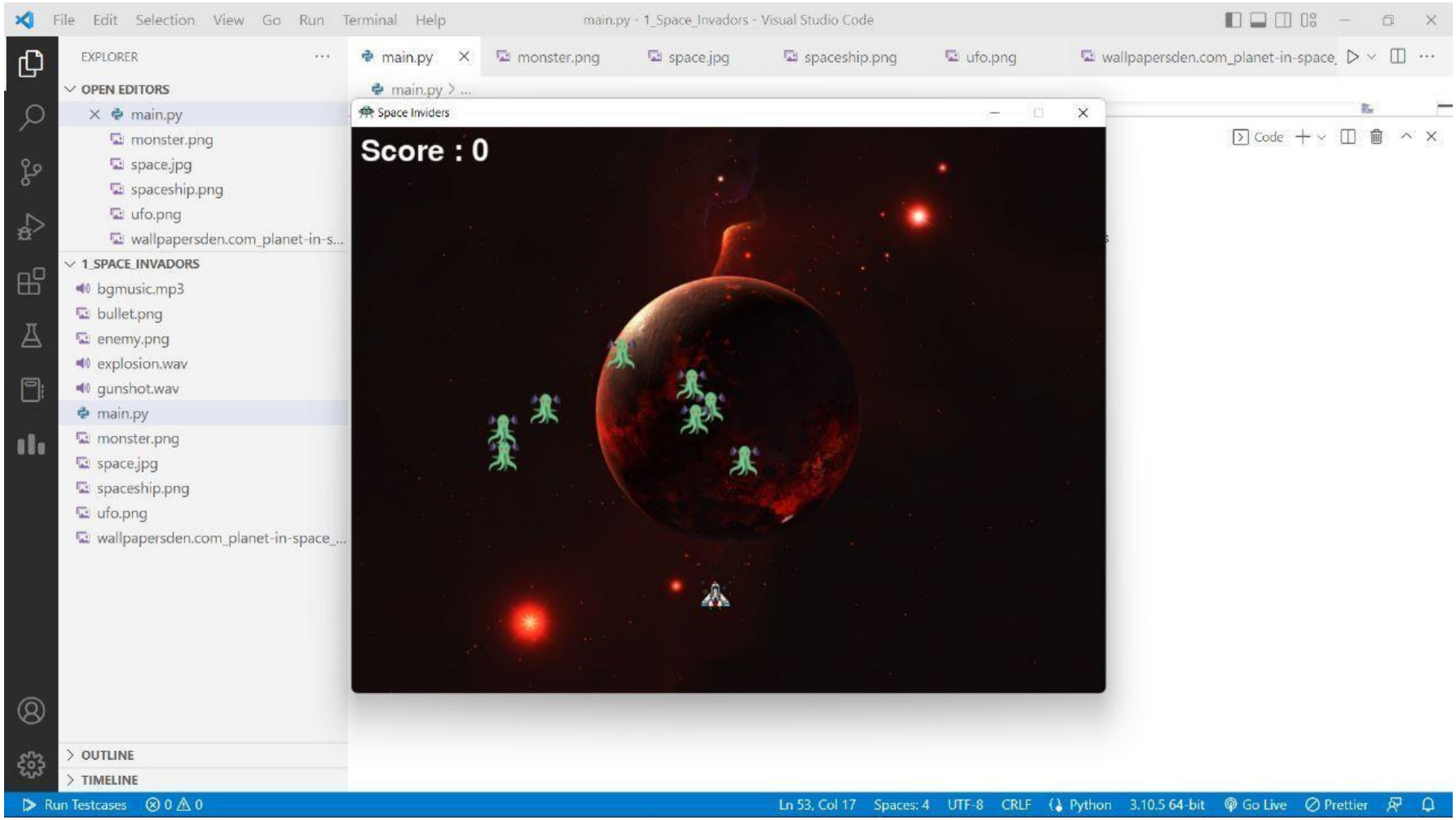
main.py > ...

```
128         # Collision
129         collision=isCollision(enemyX[i],enemyY[i],bulletX,bulletY)
130         if collision:
131             explosion_sound=mixer.Sound("E:\\1_Space_Invadors\\explosion.wav")
132             explosion_sound.play()
133             bulletY=480
134             bullet_state="ready"
135             score_value+=1
136             print(score_value)
137             enemyX[i]=random.randint(0,735)
138             enemyY[i]=random.randint(50,150)
139 
140         enemy(enemyX[i],enemyY[i],i)
141 
142     # Player
143     playerX+=playerX_change
144     if playerX<=0:
145         playerX=0
146     elif playerX>=736:
147         playerX=736
148 
149     playerY+=playerY_change
150     if playerY<=0:
151         playerY=0
152     elif playerY>=536:
153         playerY=536
154 
155     # Bullet Movement
156     if bulletY<=0:
157         bulletY=480
158         bullet_state="ready"
159 
160     if bullet_state is "fire":
161         fire_bullet(bulletX,bulletY)
```

main.py > ...

```
149     playerY+=playerY_change
150     if playerY<=0:
151         playerY=0
152     elif playerY>=536:
153         playerY=536
154 
155     # Bullet Movement
156     if bulletY<=0:
157         bulletY=480
158         bullet_state="ready"
159 
160     if bullet_state is "fire":
161         fire_bullet(bulletX,bulletY)
162         bulletY-=bulletY_change
163 
164     player(playerX,playerY)
165     show_score(textX,textY)
166     pygame.display.update()
```

==========


File Edit Selection View Go Run Terminal Help
main.py - 1_Space_Invadors - Visual Studio Code
EXPLORER
OPEN EDITORS
main.py
monster.png
space.jpg
spaceship.png
ufo.png
wallpapersden.com_planet-in-s...
1_SPACE_INVADORS
bgmusic.mp3
bullet.png
enemy.png
explosion.wav
gunshot.wav
main.py
monster.png
space.jpg
spaceship.png
ufo.png
wallpapersden.com_planet-in-space_...
OUTLINE
TIMELINE
main.py
monster.png
space.jpg
spaceship.png
ufo.png
wallpapersden.com_planet-in-space_
Space Inviders
Score : 0
Code
Run Testcases
Ln 53, Col 17
Spaces: 4
UTF-8
CRLF
Python
3.10.5 64-bit
Go Live
Prettier

Conclusion: We learnt basics of PyGame and successfully implemented OOPs concepts.

References:

- Geeks for Geeks.
- Reema Thareja, “Python Programming Using Problem Solving Approach” Oxford University Press