**Space Shooter Video Game using Pygame**

**Mini-project Report**

Submitted by

1. Aryan Gupta-202001103041

2. Abhishek Ingle-202001103049

To the Faculty of Engineering of

Mahatma Gandhi Mission University, Aurangabad

**Under Guidance of**

Ms. Shweta Janardan Patil



**Department of Computer Science & Engineering**

**MGM’s Jawaharlal Nehru Engineering College, Aurangabad**

**Academic Year 2021-22 Part-1**

ACKNOWLEDGEMENT

We would like to express our special thanks of gratitude to our teacher

**Mrs. Shweta Janardan Patil**, who gave us the golden opportunity to do this wonderful

project on Python Programming.

We were able to learn so many new things in python programming, and get an idea of various advanced topics related to Python programming, which we are sure will be really helpful in our journey of B. Tech CSE, we are really thankful for them.

Abhishek Ingle

&

Aryan Gupta

SY-CSE-1

**INDEX**

|  |  |
| --- | --- |
| **Title** | **Page No** |
| Acknowledgement | 02 |
| Abstract | 04 |
| Introduction | 05 |
| Specification | 06 |
| Implementation with output screenshot | 08-13 |
| Result | 13 |
| Conclusion | 14 |
| References | 15 |

**ABSTRACT**

**TITLE: OPENCV FACIAL RECOGNITION**

**Group members:**

* 1. Aryan Gupta-202001103041

2. Abhishek Ingle-202001103049

**Objectives:**

* Make a game clone of popular 2d game, space invaders.
* Help gamer practice their hand-eye co-ordination.
* To have fun

**Created for:**

Anyone who is fond of gaming

**Functionality:**

* Game where player controls a 2d space ship
* Enemy ships approaches us from the top and we have to shoot them to amass score.

**Topics included:**

* Basic game development logic
* Game design principles
* Python module: PyGame

**­**

**INTRODUCTION**

Children learn from participating in games and develop their **social** and cognitive skills. Adults too have immense benefits from participating in games. The fun that comes with them is enough to improve your mood and make you more productive. You will also fight lifestyle diseases if you are active in games.

Studies have shown that **some video games can boost mood and make for better heart rhythms**—a sign that they may also help relieve stress. The correlation (not causality) between video games and stress has been reflected in numerous unrelated studies, which is why video games have been used in therapy for over a decade.

**The most notable positive effects of gaming include:**

* Improved cognitive abilities.
* Improved problem-solving skills and logic.
* Increased hand-to-eye coordination.
* Greater multi-tasking ability.
* Faster and more accurate decision-making.
* Enhanced prosocial behaviors.
* Better eyesight (attention to detail)

**SPECIFICATION**

**System Specs:**

|  |  |
| --- | --- |
| Brand | ‎HP |
| Resolution | ‎HD (1366 X 768) |
| Processor Brand | ‎AMD |
| Processor Type | ‎Ryzen 5 3500U |
| Processor Speed | ‎3.7 GHz |
| Processor Count | ‎1 |
| RAM Size | ‎8 GB |
| Memory Technology | ‎DDR4 |
| Maximum Memory Supported | ‎16 GB |
| Memory Clock Speed | ‎2400 MHz |
| Hard Drive Size | ‎1 TB |
| Graphics Coprocessor | ‎INTEL |
| Graphics Chipset Brand | ‎AMD |
| Graphics Card Description | ‎Integrated |
| Graphics RAM Type | ‎Shared |
| Graphics Card Interface | ‎Integrated |
| Number of USB 2.0 Ports | ‎1 |
| Number of USB 3.0 Ports | ‎2 |
| Operating System | ‎Windows 10 Home |
| Average Battery Life (in hours) | ‎10.3 Hours |
| Are Batteries Included | ‎Yes |
| Lithium Battery Energy Content | ‎4.9-Watt Hours |
| Lithium Battery Weight | ‎0.85 Grams |
| Number Of Lithium-Ion Cells | ‎2 |
| Included Components | ‎LAPTOP, MANUAL, ADAPTERS AND POWER CABLE |
| Manufacturer | ‎hp |
| Item Weight | ‎1 kg 500 g |

**IMPLEMENTATION & OUTPUT**

CODE:

import pygame

import os

import time

import random

pygame.font.init()

WIDTH, HEIGHT = 800, 700

WIN = pygame.display.set\_mode((WIDTH, HEIGHT))

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

# Load images

RED\_SPACE\_SHIP = pygame.image.load(os.path.join("assets", "pixel\_ship\_red\_small.png"))

GREEN\_SPACE\_SHIP = pygame.image.load(os.path.join("assets", "pixel\_ship\_green\_small.png"))

BLUE\_SPACE\_SHIP = pygame.image.load(os.path.join("assets", "pixel\_ship\_blue\_small.png"))

# Player player

YELLOW\_SPACE\_SHIP = pygame.image.load(os.path.join("assets", "pixel\_ship\_yellow.png"))

# Lasers

RED\_LASER = pygame.image.load(os.path.join("assets", "pixel\_laser\_red.png"))

GREEN\_LASER = pygame.image.load(os.path.join("assets", "pixel\_laser\_green.png"))

BLUE\_LASER = pygame.image.load(os.path.join("assets", "pixel\_laser\_blue.png"))

YELLOW\_LASER = pygame.image.load(os.path.join("assets", "pixel\_laser\_yellow.png"))

# Background

BG = pygame.transform.scale(pygame.image.load(os.path.join("assets", "background-black.png")), (WIDTH, HEIGHT))

class Laser:

def \_\_init\_\_(self, x, y, img):

self.x = x

self.y = y

self.img = img

self.mask = pygame.mask.from\_surface(self.img)

def draw(self, window):

window.blit(self.img, (self.x, self.y))

def move(self, vel):

self.y += vel

def off\_screen(self, height):

return not(self.y <= height and self.y >= 0)

def collision(self, obj):

return collide(self, obj)

class Ship:

COOLDOWN = 1

def \_\_init\_\_(self, x, y, health=100):

self.x = x

self.y = y

self.health = health

self.ship\_img = None

self.laser\_img = None

self.lasers = []

self.cool\_down\_counter = 0

def draw(self, window):

window.blit(self.ship\_img, (self.x, self.y))

for laser in self.lasers:

laser.draw(window)

def move\_lasers(self, vel, obj):

self.cooldown()

for laser in self.lasers:

laser.move(vel)

if laser.off\_screen(HEIGHT):

self.lasers.remove(laser)

elif laser.collision(obj):

obj.health -= 10

self.lasers.remove(laser)

def cooldown(self):

if self.cool\_down\_counter >= self.COOLDOWN:

self.cool\_down\_counter = 0

elif self.cool\_down\_counter > 0:

self.cool\_down\_counter += 1

def shoot(self):

if self.cool\_down\_counter == 0:

laser = Laser(self.x, self.y, self.laser\_img)

self.lasers.append(laser)

self.cool\_down\_counter = 1

def get\_width(self):

return self.ship\_img.get\_width()

def get\_height(self):

return self.ship\_img.get\_height()

class Player(Ship):

def \_\_init\_\_(self, x, y, health=100):

super().\_\_init\_\_(x, y, health)

self.ship\_img = YELLOW\_SPACE\_SHIP

self.laser\_img = YELLOW\_LASER

self.mask = pygame.mask.from\_surface(self.ship\_img)

self.max\_health = health

def move\_lasers(self, vel, objs):

self.cooldown()

for laser in self.lasers:

laser.move(vel)

if laser.off\_screen(HEIGHT):

self.lasers.remove(laser)

else:

for obj in objs:

if laser.collision(obj):

objs.remove(obj)

if laser in self.lasers:

self.lasers.remove(laser)

def draw(self, window):

super().draw(window)

self.healthbar(window)

def healthbar(self, window):

pygame.draw.rect(window, (255,0,0), (self.x, self.y + self.ship\_img.get\_height() + 10, self.ship\_img.get\_width(), 10))

pygame.draw.rect(window, (0,255,0), (self.x, self.y + self.ship\_img.get\_height() + 10, self.ship\_img.get\_width() \* (self.health/self.max\_health), 10))

class Enemy(Ship):

COLOR\_MAP = {

"red": (RED\_SPACE\_SHIP, RED\_LASER),

"green": (GREEN\_SPACE\_SHIP, GREEN\_LASER),

"blue": (BLUE\_SPACE\_SHIP, BLUE\_LASER)

}

def \_\_init\_\_(self, x, y, color, health=100):

super().\_\_init\_\_(x, y, health)

self.ship\_img, self.laser\_img = self.COLOR\_MAP[color]

self.mask = pygame.mask.from\_surface(self.ship\_img)

def move(self, vel):

self.y += vel

def shoot(self):

if self.cool\_down\_counter == 0:

laser = Laser(self.x-20, self.y, self.laser\_img)

self.lasers.append(laser)

self.cool\_down\_counter = 1

def collide(obj1, obj2):

offset\_x = obj2.x - obj1.x

offset\_y = obj2.y - obj1.y

return obj1.mask.overlap(obj2.mask, (offset\_x, offset\_y)) != None

def main():

run = True

FPS = 60

level = 0

lives = 5

main\_font = pygame.font.SysFont("calibri", 50)

lost\_font = pygame.font.SysFont("calibri", 60)

enemies = []

wave\_length = 5

enemy\_vel = 1

player\_vel = 5

laser\_vel = 5

player = Player(300, 630)

clock = pygame.time.Clock()

lost = False

lost\_count = 0

def redraw\_window():

WIN.blit(BG, (0,0))

# draw text

lives\_label = main\_font.render(f"Lives: {lives}", 1, (255,255,255))

level\_label = main\_font.render(f"Level: {level}", 1, (255,255,255))

WIN.blit(lives\_label, (10, 10))

WIN.blit(level\_label, (WIDTH - level\_label.get\_width() - 10, 10))

for enemy in enemies:

enemy.draw(WIN)

player.draw(WIN)

if lost:

lost\_label = lost\_font.render("You Lost!!", 1, (255,255,255))

WIN.blit(lost\_label, (WIDTH/2 - lost\_label.get\_width()/2, 350))

pygame.display.update()

while run:

clock.tick(FPS)

redraw\_window()

if lives <= 0 or player.health <= 0:

lost = True

lost\_count += 1

if lost:

if lost\_count > FPS \* 3:

run = False

else:

continue

if len(enemies) == 0:

level += 1

wave\_length += 5

for i in range(wave\_length):

enemy = Enemy(random.randrange(50, WIDTH-100), random.randrange(-1500, -100), random.choice(["red", "blue", "green"]))

enemies.append(enemy)

for event in pygame.event.get():

if event.type == pygame.QUIT:

quit()

keys = pygame.key.get\_pressed()

if keys[pygame.K\_a] and player.x - player\_vel > 0: # left

player.x -= player\_vel

if keys[pygame.K\_d] and player.x + player\_vel + player.get\_width() < WIDTH: # right

player.x += player\_vel

if keys[pygame.K\_w] and player.y - player\_vel > 0: # up

player.y -= player\_vel

if keys[pygame.K\_s] and player.y + player\_vel + player.get\_height() + 15 < HEIGHT: # down

player.y += player\_vel

if keys[pygame.K\_SPACE]:

player.shoot()

for enemy in enemies[:]:

enemy.move(enemy\_vel)

enemy.move\_lasers(laser\_vel, player)

if random.randrange(0, 2\*60) == 1:

enemy.shoot()

if collide(enemy, player):

player.health -= 10

enemies.remove(enemy)

elif enemy.y + enemy.get\_height() > HEIGHT:

lives -= 1

enemies.remove(enemy)

player.move\_lasers(-laser\_vel, enemies)

def main\_menu():

title\_font = pygame.font.SysFont("calibri", 70)

run = True

while run:

WIN.blit(BG, (0,0))

title\_label = title\_font.render("Press the mouse to begin...", 1, (255,255,255))

WIN.blit(title\_label, (WIDTH/2 - title\_label.get\_width()/2, 350))

pygame.display.update()

for event in pygame.event.get():

if event.type == pygame.QUIT:

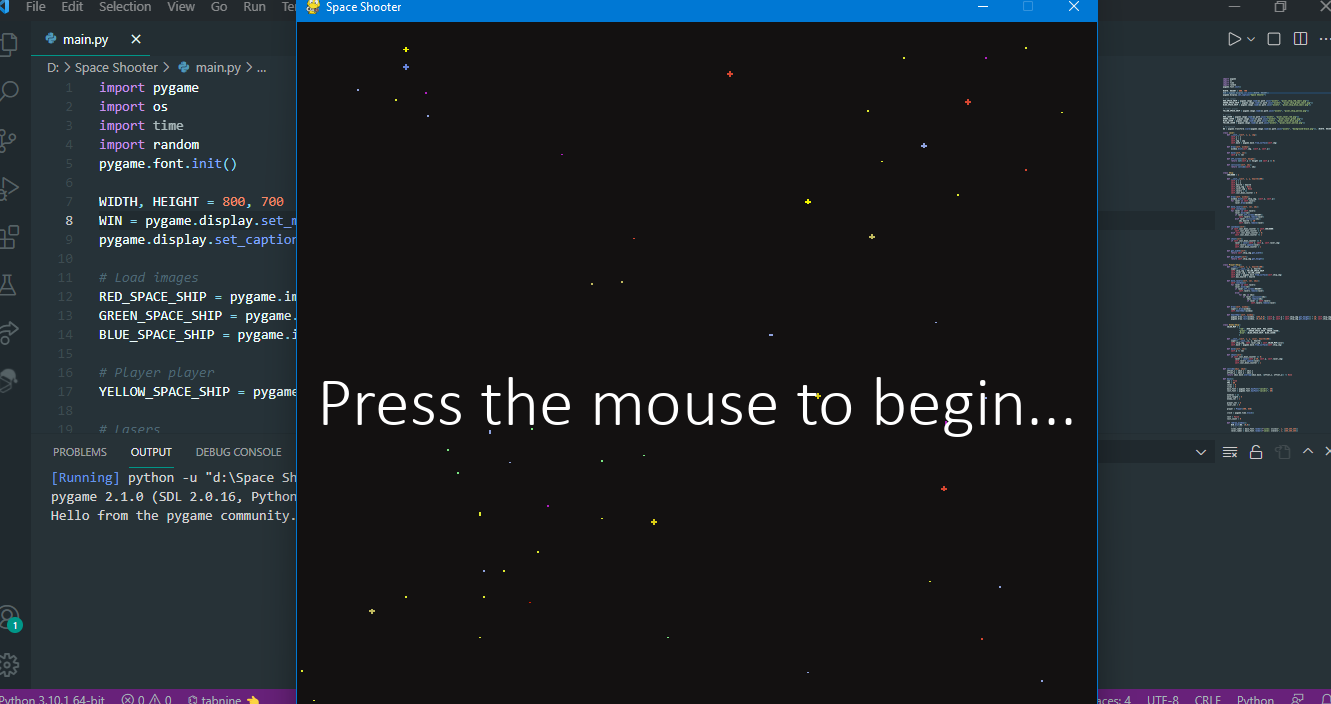
run = False

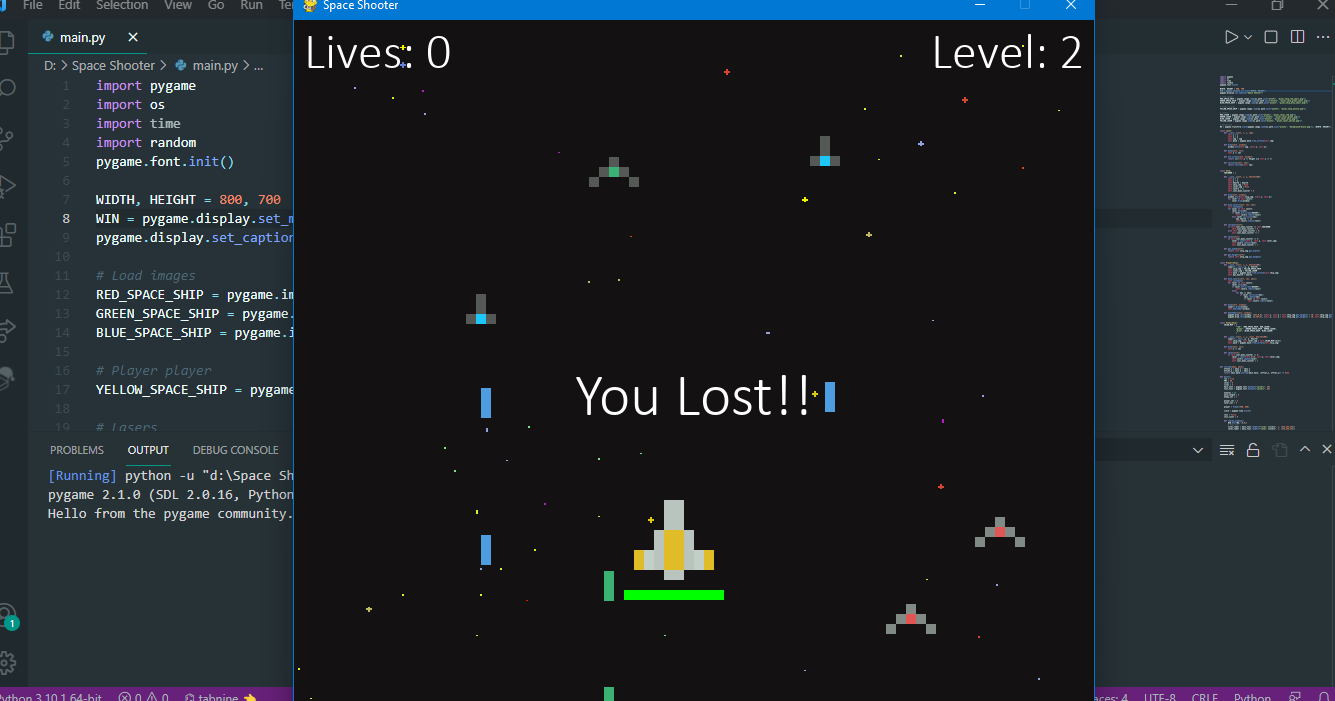
if event.type == pygame.MOUSEBUTTONDOWN:

main()

pygame.quit()

main\_menu()





**RESULT**

This program is a space shooter video game that utilizes the pygame library. Use of other modules like os, random and time is also present in this program. So, the basic design of the game is in the following way. Player controls a spaceship with WASD keys as a generic controller. Enemies are smaller spaceships which come in waves and the frequency of the waves increase with the level. The objective of the game is to shoot the enemies and prevent them from crossing the screen. Because if they do, then player will lose a life. Player has a total of 5 lives. Player also has a health bar and if they are hit by incoming lasers then they will lose some of their health. If by any chance, health or lives hit 0 then the player will lose the game. The main goal is to reach the maximum number of levels without losing their health or lives.

For the implementation of this design, we first must specify the size of the window. After that next step is to load all the resources like images of all the various coloured enemy ships, player spaceship, lasers of the spaceships and the background of the game. Then the most crucial part must be written into the code and that is game mechanics. For reusability and flexibility of code, object-oriented programming is used. For example, a spaceship class will become the parent of the player and enemy child classes using an oops concept called inheritance. After that the main function must be initialized.

**CONCLUSION**

Because of our daily stress filled day to day life people have grown lethargic and unenthusiastic. Our goal was to create a game which targets people who can be entertained by high intensity, shooting gameplay. This game not only increases Improved cognitive abilities it also improves problem-solving skills and logic, it Increases hand-to-eye coordination, Greater multi-tasking ability, Faster and more accurate decision-making, Enhanced prosocial behaviors, better eyesight (attention to detail).

The correlation (not causality) between video games and stress has been reflected in numerous unrelated studies, which is why video games have been used in therapy for over a decade.

Video games are a form of media that is often associated with negative health consequences. However, when games are played in moderation and with mindfulness, they are a viable source of stress relief as well as a catalyst for mental health improvement and development of social skills.

In conclusion, this game was made for players to relax and enjoy a shooting game where they can see how long they can last against the incoming barrage of lasers from the enemy ships.

**REFERENCES**

1. https://en.wikipedia.org wiki › Shooter\_game
2. <https://www.youtube.com/watch?v=Q-__8Xw9KTM>
3. <https://www.pygame.org/docs/>
4. <https://www.pygame.org/wiki/tutorials>