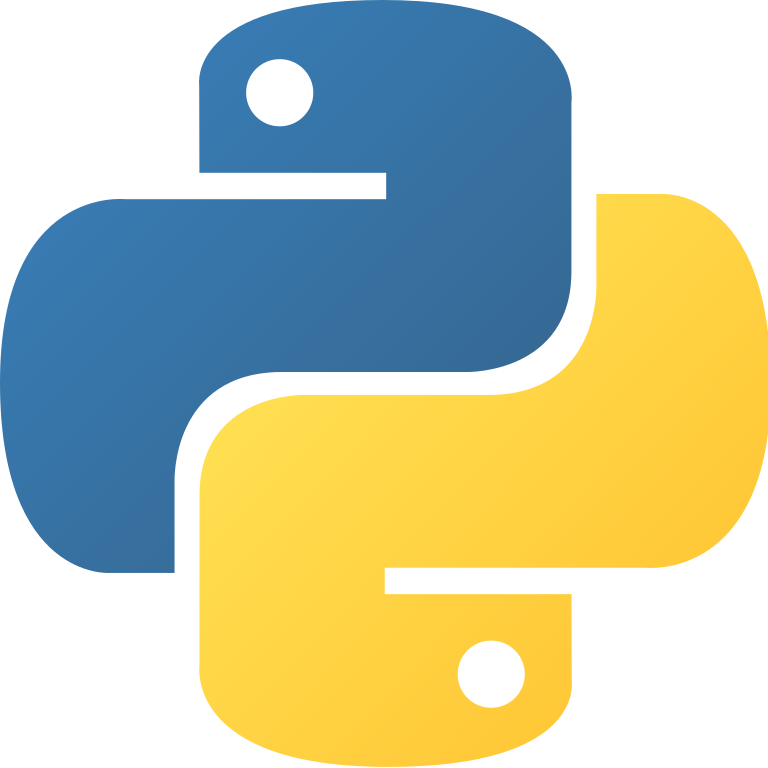
PYTHON

PROJECT

**TOPIC:** Space Shooter game



ABHISHEK PATIL

SYIT C

6375

**MAHATMA EDUCATION SOCIETY’’S**

**PILLAI’S COLLEGE OF ARTS, COMMERECE & SCIENCE**

(AUTONOMOUS INSTITUTION AFFILATED TO THE UNIVERSITY OF MUMBAI)

DR K.M. VASUDEVAN PILLAI CAMPUS, SECTOR 16, NEW PANVEL 410206.

(NAAC ACCREDITED ‘A’ GRADE)

**CERTIFICATE**

THIS IS TO CERTIFY THAT MS/MR         ABHISHEK NARAYAN PATIL\_\_\_\_\_\_

ROLL NO         6375     , DIV    C     , CLASS     SYBSCIT                    HAS WORKED AND DULY COMPLETED HER/HIS PROJECT WORK FOR THE BACHELOR IN SCIENCE

(INFORMATION TECHNOLOGY) UNDER THE FACULTY OF SCIENCE IN THE SUBJECT ’OF

   PYTHON PROGRAMMING                AND PROJECT ENTITLED

SPACE SHOOTER GAME

I FURTHER CERTIFY THAT THE ENTIRE WORK HAS BEEN DONE BY THE LEARNER AND SUCCESSFULLY COMPLETED HIS PROJECT WORK FOR SEMESTER-III YEAR 2020-2021.

**ASST. PROF. SANJANA RAMESH BHANGALE**

INTRODUCTION

## AIM

To program a game using python and display its output.

## WHAT IS PYTHON?

**Python** is an interpreted, high-level and general-purpose programming language. Created by **Guido van Rossum** and first released in **1991**, Python's design philosophy emphasizes code readability with its notable use of significant whitespace.



## LIBRARY USED AND WHAT DOES IT DO?

The **pygame library** is an open-source **module** for the **Python** programming language specifically intended to help you make games and other multimedia applications. Built on top of the highly portable SDL (Simple Direct Media Layer) development **library**, **pygame** can run across many platforms and operating systems.



## ABOUT THIS PROJECT.

This project is created using python library called pygame. This project is titled space shooter. In this project there is a space ship which shoots incoming enemies. Also score and lives are displayed. Player has health bar which is displayed in a green rectangle.

Program

import pygame

import os

import time

import random

pygame.font.init()

WIDTH, HEIGHT = 400, 600

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

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

# Load images

PURPLE\_SPACE\_SHIP = pygame.transform.scale(pygame.image.load(

    os.path.join("assets", "purple\_ufo.png")), (45, 45))

BLUE\_SPACE\_SHIP = pygame.transform.scale(pygame.image.load(

    os.path.join("assets", "blue\_ufo.png")), (45, 45))

RED\_SPACE\_SHIP = pygame.transform.scale(pygame.image.load(

    os.path.join("assets", "red\_ufo.png")), (45, 45))

# Player player

HERO\_SPACE\_SHIP = pygame.transform.scale(pygame.image.load(

    os.path.join("assets", "spaceship\_green.png")), (64, 64))

HERO\_SPACE\_SHIP\_SMALL = pygame.transform.scale(pygame.image.load(

    os.path.join("assets", "spaceship\_green.png")), (32, 32))

# Lasers

ENEMY\_LASER = pygame.transform.scale(pygame.image.load(

    os.path.join("assets", "enemy-layzer.png")), (8, 8))

HERO\_LASER = pygame.transform.scale(pygame.image.load(

    os.path.join("assets", "hero-layzer.png")), (16, 16))

# Background

BG = pygame.transform.scale(pygame.image.load(

    os.path.join("assets", "bg.png")), (WIDTH, HEIGHT))

# space sprites

PURPLE\_PLANET = pygame.transform.scale(pygame.image.load(

    os.path.join("assets", "planet\_purple.png")), (64, 64))

PURPLE\_PLANET\_SMALL = pygame.transform.scale(pygame.image.load(

    os.path.join("assets", "planet\_purple.png")), (32, 32))

ORANGE\_PLANET = pygame.transform.scale(pygame.image.load(

    os.path.join("assets", "planet\_orange.png")), (100, 100))

ORANGE\_PLANET\_SMALL = pygame.transform.scale(pygame.image.load(

    os.path.join("assets", "planet\_orange.png")), (80, 80))

GREY\_METEOR = pygame.transform.scale(pygame.image.load(

    os.path.join("assets", "meteor\_grey.png")), (16, 16))

SMALL\_GREY\_METEOR = pygame.transform.scale(pygame.image.load(

    os.path.join("assets", "meteor\_grey.png")), (32, 32))

# explosion

EXPLOSION = pygame.transform.scale(pygame.image.load(

    os.path.join("assets", "explosionzoom.png")), (64, 64))

# buttons

BUT\_W = pygame.transform.scale(pygame.image.load(

    os.path.join("assets", "but\_W2.png")), (32, 32))

BUT\_A = pygame.transform.scale(pygame.image.load(

    os.path.join("assets", "but\_A2.png")), (32, 32))

BUT\_S = pygame.transform.scale(pygame.image.load(

    os.path.join("assets", "but\_S2.png")), (32, 32))

BUT\_D = pygame.transform.scale(pygame.image.load(

    os.path.join("assets", "but\_D2.png")), (32, 32))

BUT\_SPACE = pygame.transform.scale(pygame.image.load(

    os.path.join("assets", "but\_space2\_not.png")), (120, 90))

# title

TITLE = pygame.image.load(

    os.path.join("assets", "TITLE.png"))

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 = 30

    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.explosion\_img = None

        self.lasers = []

        self.cool\_down\_counter = 0

        self.planet\_img = None

    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+13, self.y, self.laser\_img)

            laser2 = Laser(self.x+42, self.y+5, self.laser\_img)

            self.lasers.append(laser)

            self.lasers.append(laser2)

            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 = HERO\_SPACE\_SHIP

        self.laser\_img = HERO\_LASER

        self.explosion\_img = EXPLOSION

        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, ENEMY\_LASER),

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

        "purple": (PURPLE\_SPACE\_SHIP, ENEMY\_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+10, self.y, self.laser\_img)

            self.lasers.append(laser)

            self.cool\_down\_counter = 1

class Planet(Ship):

    COLOR\_MAP = {

        "small\_orange\_planet": (ORANGE\_PLANET\_SMALL),

        "orange": (ORANGE\_PLANET),

        "purple": (PURPLE\_PLANET),

        "small\_purple\_planet": (PURPLE\_PLANET\_SMALL),

        "meteor": (GREY\_METEOR),

        "small\_meteor": (SMALL\_GREY\_METEOR)

    }

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

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

        self.ship\_img = self.COLOR\_MAP[color\_planet]

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

    def move(self, vel):

        self.y += vel

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 = 100

    level = 0

    lives = 3

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

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

    clock = pygame.time.Clock()

    enemies = []

    wave\_length = 5

    enemy\_vel = 1

    planets = []

    wave\_length\_planets = 2

    planet\_vel = 0.5

    laser\_vel = 5

    player = Player(200, 330)

    player\_vel = 5

    lost = False

    lost\_count = 0

    def redraw\_window():

        WIN.blit(BG, (0, 0))

        WIN.blit(HERO\_SPACE\_SHIP\_SMALL, (10, 10))

        lives\_label = main\_font.render(

            f"x {lives}", 1, (255, 255, 255))

        level\_label = main\_font.render(

            f"Level: {level}", 1, (255, 255, 255))

        WIN.blit(lives\_label, (HERO\_SPACE\_SHIP\_SMALL.get\_width()+10, 10))

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

        for planet in planets:

            planet.draw(WIN)

        player.draw(WIN)

        for enemy in enemies:

            enemy.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

        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()+30 < HEIGHT:  # down

            player.y += player\_vel

        if keys[pygame.K\_SPACE]:

            player.shoot()

        if len(enemies) == 0:

            level += 1

            wave\_length += 2

            for i in range(wave\_length):

                enemy = Enemy(random.randrange(

                    50, WIDTH-100), random.randrange(-1150, -100), random.choice(["red", "blue", "purple"])

                )

                enemies.append(enemy)

        for enemy in enemies[:]:

            enemy.move(enemy\_vel)

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

            if random.randrange(0, 2\*240) == 1:

                enemy.shoot()

            if collide(enemy, player):

                player.health -= 10

                lives -= 1

                enemies.remove(enemy)

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

                enemies.remove(enemy)

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

        if len(planets) == 0:

            wave\_length\_planets = 3

            for i in range(wave\_length\_planets):

                planet = Planet(random.randrange(

                    50, WIDTH-100), random.randrange(-1150, -100), random.choice(["meteor", "small\_meteor", "orange", "small\_orange\_planet", "purple", "small\_purple\_planet"])

                )

                planets.append(planet)

        for planet in planets[:]:

            planet.move(planet\_vel)

            if planet.y + planet.get\_height() > HEIGHT+100:

                planets.remove(planet)

        for event in pygame.event.get():

            if event.type == pygame.QUIT:

                quit()

def main\_menu():

    title\_font = pygame.font.SysFont("comicsans", 20)

    run = True

    while run:

        WIN.blit(BG, (0, 0))

        title\_label = title\_font.render(

            "Press the mouse to begin...", 1, (255, 255, 255))

        title\_label3 = title\_font.render(

            "SPACE - Shoot", 1, (255, 255, 255))

        WIN.blit(TITLE, (-7, 0))

        WIN.blit(BUT\_W, (130, 368))

        WIN.blit(BUT\_A, (98, 400))

        WIN.blit(BUT\_S, (130, 400))

        WIN.blit(BUT\_D, (162, 400))

        WIN.blit(BUT\_SPACE, (200, 370))

        WIN.blit(title\_label, (WIDTH/2 - title\_label.get\_width()/2, 470))

      #  WIN.blit(title\_label3, (WIDTH/2 - title\_label.get\_width()/2, 290))

        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()

OUTPUT

## MAIN MENU





