Project Report 2019 - 2020

MANAV RACHNA INTERNATIONAL SCHOOL, NOIDA

By: Vinayak Verma & Dev Babbar

Playground of education

MANAV RACHNA INTERNATIONAL SCHOOL

NOIDA

Project Report

On

Monster Strike

FOR

CBSE 2020 Examination

{As a part of Computer Science (New) Course-083}

Submitted By:-

Vinayak Verma & Dev Babbar

XII – Newton

Roll No. 15 & 4

Under The Guidance Of

Ms Rakhi Garg

Certificate

This is to certify that the Project/Dissertation entitled _____ is a bonafide work done by _____ and _____of class XII Session 2019-20 in partial fulfillment of CBSE's AISSCE Examination 2020 and has been carried out under my direct supervision and guidance. This report or a similar report on the topic has not been submitted for any other examination and does not form a part of any other course undergone by the candidate.

_____ _____

Signature of Students Ms. Rakhi Garg

Faculty

Acknowledgement

We gratefully acknowledge the valuable guidance and support given to me by my teacher Ms. Rakhi Garg in successfully completing my project report. This project gave me a clear insight into real world business situation and gave me ample opportunities to apply me theoretical knowledge to practical situations. We would also like to thank our classmates for their suggestion and support in completing this project.

Index

Introduction

This project aims at remaking an 1980’s game named ‘Alien Invasion’. In today’s world people are forget how cool or soothing games were before.

Nowadays games are basically made in order to make more money by making an in game store or adding DLC’s to the game and to play them you buy them. Which sometimes leads people in pirating a game in order to play it.

Objective And Scope Of The Project

Our game will indulge elderly people and help them relive their childhood memories.

Youngsters will also be attracted to this game to experience how the world of games started.

Problem Definition And Analysis

Elderly people are finding less and less things to do in their time. So basically this game is gonna give them something to do in their free time.

System Implementation

System Requirements

- **Main Processor:** i7-8750H or above
- **Hard Disk:** 250 GB or greater
- **RAM:** 8 GB RAM or greater
- **Operating System:** Windows (10 Home / Professional)
- **Language:** Python

Salient Features

- You can compare your score with all-time top 10 players on EARTH.
- You can save your game score.
- You can pause the game
- There is audio in the game

Code Of Project

Main File(Window.py)

```
import sys
import pygame
from pygame.sprite import Group
from settings import Settings
from ship import Ship
import game_func as gf
from game_stats import GameStats
from button import Button
from score import Score
def run_game():
    #Initialize pygame
    pygame.init()
    #Loading a music file
    pygame.mixer.music.load('Music/Music.mp3')
    #Volume(Decreased volume)
    pygame.mixer.music.set_volume(0.70)
    #-1 repeats it infinite times
    pygame.mixer.music.play(-1)
    #Settings() is a defined class
    Invad_set = Settings()
    #Setting up the screen
    screen = pygame.display.set_mode((Invad_set.width,
Invad_set.height), pygame.RESIZABLE) #pygame.FULLSCREEN

    #Top window name
    pygame.display.set_caption("Kevin The Cube Distroyer")

    #(Changed "start" to "  Click to start")
    play_b = Button(Invad_set,screen,"Click to start")
    pause_b = Button(Invad_set,screen,"CONTINUE")
    stats = GameStats(Invad_set)
    sb= Score(Invad_set, screen, stats)
    ship = Ship(Invad_set, screen)
    bullets = Group()
    aliens= Group()
```

```
    gf.create_fleet(Invad_set, screen, ship, aliens)
    while True:
        gf.check_events(Invad_set, screen,stats,sb,play_b
,ship,aliens,bullets,pause_b)
        if stats.game_active:
            ship.update()
            bullets.update()
            gf.update_bullets(Invad_set,
screen,stats,sb,ship,aliens, bullets)

gf.update_aliens(Invad_set,stats,sb,screen,ship,aliens,bullets)
        gf.update_screen(Invad_set, screen, stats, sb, ship,
aliens, bullets, play_b)



run_game()
```

Settings File(settings.py)

```
import pygame
class Settings():

    def __init__(self):
        #Dimensions of screen
        self.width=1200
        self.height=700

        self.bg_color=(0,0,0)
        #Space background
        self.bg_image = pygame.image.load("img/BG.jpg")

        self.ship_limit = 2

        self.bullet_width = 5
        self.bullet_height = 15
        self.bullet_color= (227,207,87)

        #Drop speed of aliens
        self.fleet_drop_speed = 20

        #Change in speed as round increases-Multiplied
        self.speed_up = 1.3
        #Change in score per kill as round increases
        self.score_s= 50

        self.int_change_set()

        self.s_active = True

        self.pau_b = False

    #Starting speed(Increased space_ship speed)
    def int_change_set(self):
        self.ship_speed = 6
        self.bullet_speed = 4
        self.alien_speed_factor = 2.5
        #Doubt
        self.fleet_dir = 1
        #Points per kill in level 1
        self.alien_p = 100

    #Change in speed as level changes(Spaceship, bullet speed
stays same throughout)
    def inc_speed(self):
        self.alien_speed_factor *= self.speed_up
```

```
        #Points increase as level increase(Now adds by 50 per
round not multiply by 1.5)
        self.alien_p = int(self.alien_p + self.score_s)
```

Ship File(ship.py)

```
import pygame
from pygame.sprite import Sprite
class Ship(Sprite):

    def __init__(self,Invad_set,screen):

        super(Ship,self).__init__()
        self.screen = screen
        self.Invad_set = Invad_set
        self.image = pygame.image.load('img/ship.bmp')
        self.rect = self.image.get_rect()
        self.screen_rect = screen.get_rect()
        self.rect.centerx = self.screen_rect.centerx
        self.rect.bottom = self.screen_rect.bottom
        self.center = float(self.rect.centerx)
        self.moving_right =False
        self.moving_left = False
        '''self.moving_up = False
        self.moving_down = False'''
    def update(self):
        if self.moving_right and self.rect.right <
self.screen_rect.right:
            self.center += self.Invad_set.ship_speed
        elif self.moving_left and self.rect.left > 0:
            self.center -= self.Invad_set.ship_speed
        '''elif self.moving_up == True:
            self.rect.centery -=1
        elif self.moving_down == True:
            self.rect.centery +=1'''
        self.rect.centerx = self.center

    def blitme(self):
            self.screen.blit(self.image, self.rect)
```

Game Function File (game_func.py)

```
import sys

import pygame

from bullet import Bullet

from alien import Alien

import time

from button import Button

from settings import __init__


def check_keydown_events(event,Invad_set,screen ,stats, sb,play_b,
ship,aliens, bullets,pause_b):

    aaa=Invad_set.bullet_speed

    bbb=Invad_set.alien_speed_factor

    if event.key == pygame.K_RIGHT:

        #If right key is pressed ship moves right

        ship.moving_right = True

    elif event.key == pygame.K_LEFT:

        #If left key is pressed ship moves left

        ship.moving_left = True

    elif event.key == pygame.K_SPACE:

        #If spacebar is pressed ship shoots out a bullet

        new_bullet = Bullet(Invad_set,screen, ship)

        bullets.add(new_bullet)

        #Sound effect

        effect = pygame.mixer.Sound('Music/GunSound.wav')

        effect.play()

    elif event.key == pygame.K_ESCAPE:

        #Game is deactivated and closed after esc

        Invad_set.s_active = False
```

```
        #Pygame ends after game is closed

        pygame.quit() #sys.exit()

    elif event.key == pygame.K_p:

        Invad_set.bullet_speed=0

        Invad_set.alien_speed_factor=0

    elif event.key==pygame.K_r:

        Invad_set.bullet_speed=4

        Invad_set.alien_speed_factor=2.5

        #pause_e(event,Invad_set,screen ,stats, sb,play_b,
ship,aliens, bullets,pause_b)

'''def pause_e(event,Invad_set,screen ,stats, sb,play_b,
ship,aliens, bullets,pause_b):

    Invad_set.pau_b = True

    if Invad_set.pau_b == True:

        pygame.mouse.set_visible(True)

        pause_b.draw_b()

    pygame.display.flip()'''


def check_keyup_events(event,Invad_set,screen ,stats, sb,play_b,
ship,aliens, bullets,pause_b):

    if event.key == pygame.K_RIGHT:

        ship.moving_right = False

    elif event.key == pygame.K_LEFT:

        ship.moving_left = False

    elif event.key == pygame.K_p:

        pygame.K_RIGHT = False

        pygame.K_LEFT = False

        pygame.K_SPACE = False

    elif event.key==pygame.K_r:

        pygame.K_RIGHT = True
```

```
        pygame.K_LEFT = True

        pygame.K_SPACE = True

        check_events(Invad_set,screen ,stats, sb,play_b,
ship,aliens, bullets,pause_b)




def check_events(Invad_set,screen ,stats, sb,play_b, ship,aliens,
bullets,pause_b):

    if Invad_set.s_active == True:

        '''if event.type == pygame.QUIT:

            sys.exit()'''

        for event in pygame.event.get():


            if event.type == pygame.KEYDOWN:

                check_keydown_events(event,Invad_set,screen
,stats, sb,play_b, ship,aliens, bullets,pause_b)

            elif event.type == pygame.KEYUP:

                check_keyup_events(event,Invad_set,screen ,stats,
sb,play_b, ship,aliens, bullets,pause_b)

            elif event.type == pygame.MOUSEBUTTONDOWN:

                mouse_x,mouse_y = pygame.mouse.get_pos()

                check_play_b(Invad_set,screen,stats,sb,
play_b,ship, aliens, bullets, mouse_x, mouse_y)

            elif event.type == pygame.MOUSEBUTTONDOWN:

                mouse_x,mouse_y = pygame.mouse.get_pos()

        '''elif event.key == pygame.K_UP:

                ship.moving_up = True

        elif event.key == pygame.K_DOWN:

                ship.moving_down = True'''

    else:

        pass
```

```
def check_play_b(Invad_set,screen,stats,sb, play_b,ship, aliens,
bullets, mouse_x, mouse_y):
    b_clicked = play_b.rect.collidepoint(mouse_x, mouse_y)
    if b_clicked and stats.game_active == False:
        Invad_set.int_change_set()
        pygame.mouse.set_visible(False)
        stats.reset_stats()
        stats.game_active = True
        sb.F_score()
        sb.F_high_score()
        sb.level_f()
        sb.prep_ship()



        aliens.empty()
        bullets.empty()


        create_fleet(Invad_set, screen, ship, aliens)
        ship.center
    elif b_clicked and stats.game_active == True:
        pass


def check_pause_b(Invad_set,screen,stats,sb, play_b,ship, aliens,
bullets, mouse_x, mouse_y):
    b_clicked = play_b.rect.collidepoint(mouse_x, mouse_y)
    if b_clicked and stats.game_active == False:
        pygame.mouse.set_visible(False)
        Invad_set.pau_b == False
```

```
    elif b_clicked and stats.game_active == True:
        pass


def update_screen(Invad_set, screen,stats,sb, ship,aliens,
bullets,play_b):
    screen.blit(Invad_set.bg_image,[0,0])
    for bullet in bullets.sprites():
        bullet.draw_bullet()
    ship.blitme()
    aliens.draw(screen)
    sb.show_score()


    if stats.game_active != True:
        play_b.draw_b()
    pygame.display.flip()



def update_bullets(Invad_set, screen,stats,sb,ship,aliens,
bullets):


    bullets.update()
    for bullet in bullets.copy():
            if bullet.rect.bottom ==0:
                bullets.remove(bullet)
    check_bul_all_coll(Invad_set, screen ,
stats,sb,ship,aliens,bullets)


def check_bul_all_coll(Invad_set, screen , stats,
sb,ship,aliens,bullets):
    coll= pygame.sprite.groupcollide(bullets,aliens,True,True)
```

```
    for aliens in coll.values():
        stats.score += Invad_set.alien_p * len(aliens)
        sb.F_score()
    check_high_s(stats,sb)
    if len(aliens) == 0:

        effect = pygame.mixer.Sound('Music/LevelUp.wav')
        effect.play()
        bullets.empty()
        Invad_set.inc_speed()

        stats.level +=1
        sb.level_f()
        time.sleep(2)
        create_fleet(Invad_set, screen, ship, aliens)

def create_fleet(Invad_set, screen,ship, aliens):
    alien = Alien(Invad_set, screen)
    number_aliens_x = get_number_alien_x(Invad_set,
alien.rect.width)
    number_rows = get_number_rows(Invad_set, ship.rect.height,
alien.rect.height)



    for row_number in range(number_rows):
        for alien_number in range(number_aliens_x):
            create_alien(Invad_set, screen, aliens, alien_number,
row_number)

def get_number_alien_x(Invad_set, alien_width):
```

```
    available_space_x = Invad_set.width - (2*alien_width)

    number_aliens_x= int(available_space_x / (1*alien_width))

    return number_aliens_x


def create_alien(Invad_set, screen, aliens, alien_number,
row_number):

    alien = Alien(Invad_set, screen)

    alien_width = alien.rect.width

    alien.x = alien_width + (1 * alien_width * alien_number)

    alien.rect.x = alien.x

    alien.rect.y = alien.rect.height + (1 * alien.rect.height *
row_number)

    aliens.add(alien)


def get_number_rows(Invad_set, ship_height, alien_height):

    available_space_y = (Invad_set.height - (1* alien_height) -
ship_height)

    number_rows = int(available_space_y  / (2* alien_height))

    return number_rows


def check_fleet_edges(Invad_set, aliens):

    for alien in aliens.sprites():

        if alien.check_edges(Invad_set):

            change_fleet_dirc(Invad_set, aliens)

            break


def change_fleet_dirc(Invad_set, aliens):

    for alien in aliens.sprites():

        alien.rect.y += Invad_set.fleet_drop_speed

    Invad_set.fleet_dir *= -1
```

```
def ship_hit(Invad_set,stats,sb,screen,ship,aliens,bullets):


    if stats.ships_left >0:
        effect = pygame.mixer.Sound('Music/LostLife.wav')
        effect.play()
        stats.ships_left -= 1
        sb.prep_ship()
        aliens.empty()
        bullets.empty()
        create_fleet(Invad_set, screen,ship,aliens)
        ship.center
        time.sleep(2)
    else:
        effect = pygame.mixer.Sound('Music/GameOver.wav')
        effect.play()
        sb.high_score_board()
        time.sleep(10)


        stats.game_active = False
        pygame.mouse.set_visible(True)



def check_aliens_bottom(Invad_set,stats,sb,screen,
ship,aliens,bullets):
    if Invad_set.s_active == True:
        screen_rect = screen.get_rect()
        for alien in aliens.sprites():
            if alien.rect.bottom >= screen_rect.bottom:
```

```
ship_hit(Invad_set,stats,sb,screen,ship,aliens,bullets)
                break
    else:
        pass

def update_aliens(Invad_set,stats,sb,screen,ship,aliens,bullets):
    check_fleet_edges(Invad_set, aliens)
    aliens.update()
    if pygame.sprite.spritecollideany(ship,aliens):
        ship_hit(Invad_set,stats,sb,screen,ship,aliens,bullets)

check_aliens_bottom(Invad_set,stats,sb,screen,ship,aliens,bullets)

def check_high_s(stats,sb):
    if stats.score> stats.high_score:
        stats.high_score = stats.score
        sb.F_high_score()
```

Game Stats File(game_stats.py)

```
class GameStats():

    def __init__(self,Invad_set):

        self.Invad_set = Invad_set

        self.reset_stats()

        self.game_active = False


        self.high_score= 1000000


    def reset_stats(self):

        self.ships_left = self.Invad_set.ship_limit

        self.score = 0

        self.level = 1
```

Button File(button.py)

```
import pygame.font

class Button():

    def __init__(self, Invad_set, screen, msg):

        self.screen = screen

        #Get the rectangular area of the surface

        self.screen_rect = screen.get_rect()


        #(Increased width)

        self.width, self.height =250,50

        #Background color(Changed)

        self.b_color = (0,0,0)

        #Text color

        self.t_color = (255,255,255)

        #Font

        self.font = pygame.font.SysFont(None,48)


        self.rect = pygame.Rect(0,0,self.width,self.height)

        #Center the button

        self.rect.center = self.screen_rect.center


        self.prep_msg(msg)


    #Makes sure text appears in the centre, also turns it into a
rendered image

    def prep_msg(self,msg):

        self.msg_image = self.font.render(msg, True, self.t_color,
self.b_color)

        self.msg_image_rect = self.msg_image.get_rect()
```

```
        self.msg_image_rect.center = self.rect.center


    #Draw blank button and then draw image
    def draw_b(self):
        self.screen.fill(self.b_color, self.rect)
        self.screen.blit(self.msg_image, self.msg_image_rect)
```

Scoreboard File(score.py)

```
import pygame

from pygame.sprite import Group

from ship import Ship

import mysql.connector as mycon

class Score():

    def __init__(self, Invad_set, screen, stats):

        self.screen = screen

        self.screen_rect = screen.get_rect()

        self.Invad_set = Invad_set

        self.stats = stats



        self.t_color =(47, 255, 0)

        self.bg_color=(0,0,0)

        self.font = pygame.font.SysFont("Times New Roman", 50)
#comicsansms


        self.F_score()

        self.F_high_score()

        self.level_f()

        self.prep_ship()


    def F_score(self):

        self.r_score = int(round(self.stats.score, -1))

        self.scores =  "{:,}".format(self.r_score)

        self.score_image = self.font.render(self.scores, True,
self.t_color, self.bg_color)

        self.score_rect = self.score_image.get_rect()
```

```
        self.score_rect.right = self.screen_rect.right - 20

        self.score_rect.top = 20


    def show_score(self):

        self.screen.blit(self.score_image, self.score_rect)

        self.screen.blit(self.high_score_image,
self.high_score_rect)

        self.screen.blit(self.level_image, self.level_rect)


        self.d_image =
self.font.render("P:Pause",True,self.t_color, self.bg_color)

        self.d_rect = self.d_image.get_rect()

        self.d_rect.left = self.screen_rect.left +20

        self.d_rect.top = 550

        self.r_image =
self.font.render("R:Resume",True,self.t_color, self.bg_color)

        self.r_rect = self.r_image.get_rect()

        self.r_rect.left = self.d_rect.left

        self.r_rect.bottom = self.d_rect.bottom + 70


        self.screen.blit(self.d_image, self.d_rect)

        self.screen.blit(self.r_image, self.r_rect)

        for i in range(self.stats.ships_left + 1):

            self.screen.blit(self.sh, self.ship_rect)



    def F_high_score(self):

        high_score  = int(round(self.stats.high_score, -1))

        high_score_s = "{:,}".format(high_score)

        self.high_score_image = self.font.render(high_score_s,
True, self.t_color, self.bg_color)
```

```
        self.high_score_rect = self.high_score_image.get_rect()
        self.high_score_rect.centerx = self.screen_rect.centerx
        self.high_score_rect.top = self.score_rect.top
       # self.high_score_board()


    def level_f(self):
        self.level_image =
self.font.render(str(self.stats.level),True, self.t_color,
self.bg_color)
        self.level_rect = self.level_image.get_rect()
        self.level_rect.right = self.score_rect.right
        self.level_rect.top = self.score_rect.bottom + 10


    def prep_ship(self):
        self.ships = Group()
        for ship_number in range(self.stats.ships_left):
            self.ship = Ship(self.Invad_set, self.screen)
            self.sh = pygame.image.load('img/ship.bmp')
            self.ship_rect = self.sh.get_rect()
            self.ship_wid = self.sh.get_width()
            self.ship_rect.left = self.screen_rect.left
            self.ship_rect.top = self.score_rect.top
            self.ship.rect_x = 10 + (ship_number * self.ship_wid)
            self.ship.rect_y = 10
            self.ships.add(self.ship)


    def high_score_board(self):
        mydb = mycon.connect(host = "localhost",user =
"root",passwd = "ROOTLAP",
                             database = "kevin_the_cube")
```

```
        cur = mydb.cursor()
        h_sc = cur.fetchone()
        cur.execute("select * from High_Score")
        h_sc = cur.fetchall()
        b=str(input("Enter Your Name: "))
        a=0
        print(h_sc)
        c=self.r_score
        for row in h_sc:
            if row[-1] >= c:
                a=row[0]
                print("check 1")
                continue
            elif row[-1] < c:
                a= row[0]
                print("check 1.1")
                break
        print(4)
        if a!=0:
            s = "INSERT INTO High_Score(Rank_,Name_,Score)
VALUES({},'{}',{})".format(a,b,c)
            aa= "update High_Score set Rank_ =Rank_ + {} where
Rank_ < {};".format(1,a)
            cur.execute(aa)
            mydb.commit()
            cur.execute(s)
            mydb.commit()
            print("check 2.1")


        else:
```

```
            s = "INSERT INTO High_Score(Rank_,Name_,Score)
VALUES({},'{}',{})".format(a+1,b,c)

            cur.execute(s)

            mydb.commit()

            print("check 2.2")

        cur.execute("select * from High_Score")

        h_sc = cur.fetchall()

        print("check 3")

        for row in h_sc:

            print(row)

        mydb.close()
```

Bullet file(bullet.py)

```
import pygame

from pygame.sprite import Sprite


class Bullet(Sprite):

    def __init__(self,Invad_set,screen,ship):

        super(Bullet, self).__init__()

        self.screen = screen


        self.rect = pygame.Rect(0,0,Invad_set.bullet_width,
Invad_set.bullet_height)

        #Bullet comes from centre of the ship

        self.rect.centerx = ship.rect.centerx

        self.rect.top = ship.rect.top


        self.y = float(self.rect.y)


        self.color = Invad_set.bullet_color

        self.speed = Invad_set.bullet_speed

    def update(self):

        self.y -=self.speed

        self.rect.y = self.y

    def draw_bullet(self):

        pygame.draw.rect(self.screen, self.color, self.rect)
```

Alien File(alien.py)

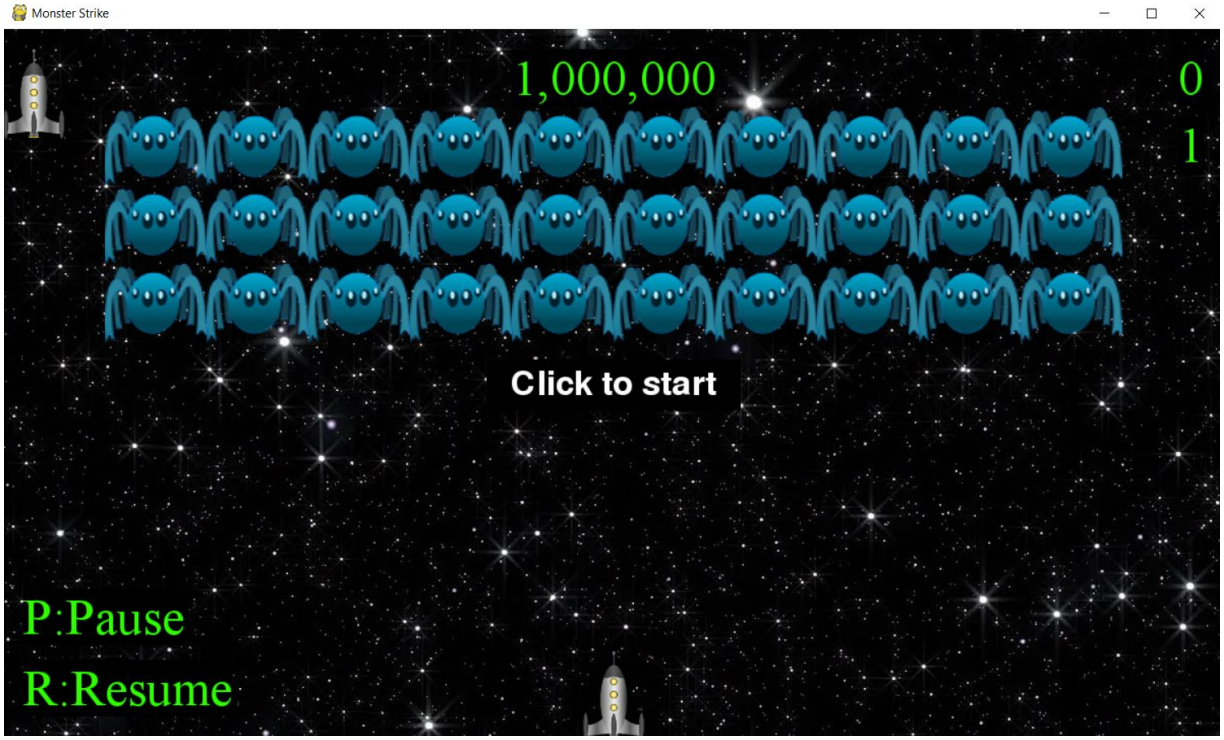
```
import pygame
#Sprite is a pygame module with basic game object classes
from pygame.sprite import Sprite
class Alien(Sprite):
    def __init__(self, Invad_set, screen):
        super(Alien,self).__init__()
        self.screen = screen
        self.Invad_set = Invad_set
        #Changed alien image
        self.image = pygame.image.load('img/alien.bmp')
        #Doubt
        self.rect = self.image.get_rect()
        self.rectx = self.rect.width
        self.recty = self.rect.height
        self.x = float(self.rect.x)
    def blitme(self):
        #Doubt
        self.screen.blit(self.image,self.rect)

    def check_edges(self,Invad_set):
        if Invad_set.s_active == True:
            screen_rect = self.screen.get_rect()
            if self.rect.right>= screen_rect.right:
                return True
            elif self.rect.left <= 0:
                return True
        else:
            pass
```

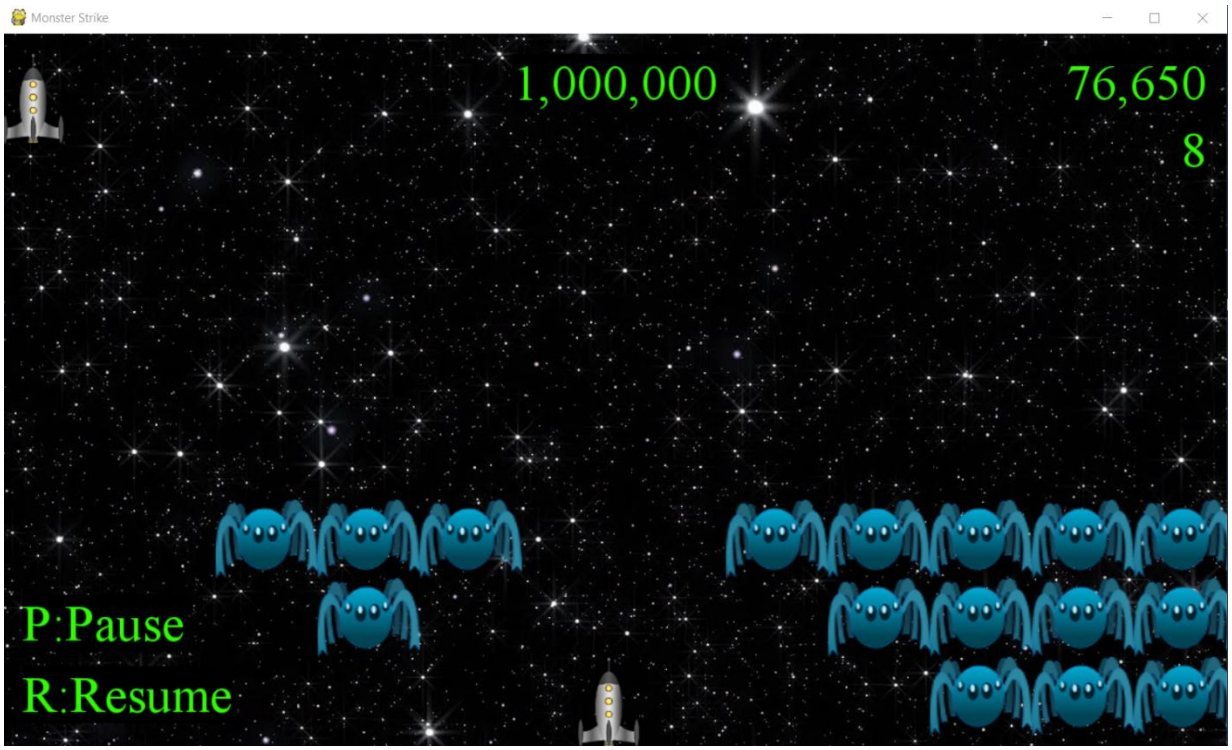
```
    def update(self):
        self.x += (self.Invad_set.alien_speed_factor *
self.Invad_set.fleet_dir)
        self.rect.x = self.x
```

Database Structure

```
create table High_Score(Rank_ int,Name_ varchar(30) unique, Score
int not null,primary key(Rank_));

insert into High_Score values(1,'Gogeta',100000000);

insert into High_Score values(2,'Ronin',99999000);

insert into High_Score values(3,'Ninja',5000000);

insert into High_Score values(4,'Fresh',4000050);

insert into High_Score values(5,'VV',2990950);

insert into High_Score values(6,'Beta',900950);

insert into High_Score values(7,'Golu',850000);

insert into High_Score values(8,'Police',600000);

insert into High_Score values(9,'Batman',500000);

insert into High_Score values(10,'Jiren',400950);

#drop table High_Score;

select * from High_Score;
```

OUTPUT


Monster Strike
1,000,000
0
1
Click to start
P:Pause
R:Resume


Monster Strike
1,000,000
76,650
8
P:Pause
R:Resume

Monster Strike	15-12-2019 13:05	File folder

__pycache__	15-12-2019 18:02	File folder	
img	13-12-2019 23:08	File folder	
misc	13-12-2019 23:08	File folder	
Music	13-12-2019 23:08	File folder	
alien.py	13-12-2019 23:08	Python File	2 KB
bullet.py	13-12-2019 23:08	Python File	1 KB
button.py	13-12-2019 23:08	Python File	2 KB
game_func.py	15-12-2019 14:48	Python File	9 KB
game_stats.py	13-12-2019 23:08	Python File	1 KB
score.py	15-12-2019 15:55	Python File	5 KB
ScoreBoard.sql	13-12-2019 23:08	SQL Text File	1 KB
settings.py	15-12-2019 15:05	Python File	2 KB
ship.py	15-12-2019 14:52	Python File	2 KB
Window.py	15-12-2019 18:02	Python File	2 KB

alien.bmp	12-06-2019 18:06	BMP File	31 KB
BG.jpg	30-06-2019 20:03	JPG File	161 KB
cube.bmp	30-06-2019 20:28	BMP File	14 KB
hero.bmp	04-07-2019 15:47	BMP File	9 KB
logo bg.bmp	30-06-2019 20:12	BMP File	8,101 KB
maxresdefault.bmp	30-06-2019 15:04	BMP File	391 KB
ship.bmp	12-06-2019 13:29	BMP File	21 KB
ship.gif	30-06-2019 19:17	GIF File	6 KB

Background Music ....
GameOver.wav
Gun Sound.mp3
GunSound.wav
LevelUp.wav
LostLife.wav
Music.mp3

Scope Of Improvement

- We can add HEALTH bar for the ship instead of 1 hit death.
- We can also develop a better Pause Menu.
- We can also add a feature where the user can chose his/her desired hero from the give choices.

Bibliography

- https://www.pygame.org/docs/
- Phyton Programming by Dr.R.Nageswara
- Python GUI Programming by Burkhard A. Meier