Name: Prince David Sanam

Project Name: Space Invaders

Game Type: Arcade

This game involves a player (Spaceship) and multiples enemies. We have written the code in such a way that, the player must blast the enemy to survive and play the game as long as possible.

First, we have created a game screen and changed the background color. Then we’ve added the image of the player at the bottom of the screen. Now, we’ve added the enemy image at the top of the screen. We’ve then added the bullet image to the code. As per the code, the player moves in X-axis and the enemy moves in Y-axis downwards and the bullet moves in Y-axis in upwards direction.

When the range(distance) between the bullet and the enemy is less than the desired distance, the enemy explodes, and the score will be increased by 1 point. So, this way, we’ve added multiple enemies to increase the complexity of the game. When the enemy enter the range of the spaceship, since the player cannot shoot the enemy, the game must end. We’ve also written code to display the score and game over text. To improve the gaming experience, we’ve added background music and bullet firing, enemy explosion sounds.

Thanks to Dr. Huabo Lu for this opportunity.

Python code:

import pygame  
import math  
import random  
from pygame import mixer  
  
# Initialize the Pygame  
pygame.init()  
# Create the screen  
screen = pygame.display.set\_mode((800,600))  
  
# Background  
background = pygame.image.load('background.jpg')  
  
#Background Music  
mixer.music.load('background.wav')  
mixer.music.play(-1)  
  
# Title and Icon  
pygame.display.set\_caption("Space Invaders")  
icon = pygame.image.load('space-game.png')  
pygame.display.set\_icon(icon)  
  
# Player  
playerimg = pygame.image.load('player.png')  
playerX=370  
playerY=480  
playerX\_change = 0  
  
# Enemy  
enemyimg = []  
enemyX= []  
enemyY= []  
enemyX\_change = []  
enemyY\_change = []  
num\_of\_enemies = 6  
  
for i in range(num\_of\_enemies):  
 enemyimg.append(pygame.image.load('enemy.png'))  
 enemyX.append(random.randint(0,800))  
 enemyY.append(random.randint(0,150))  
 enemyX\_change.append(0.75)  
 enemyY\_change.append(64)  
  
  
# Bullet  
# Bullet state:  
# Ready - You can't see the bullet on screen  
# Fire - The bullet is currently moving  
  
bulletimg = pygame.image.load('bullet.png')  
bulletX= 0  
bulletY= 480  
bulletX\_change = 0  
bulletY\_change = 1.5  
bullet\_state = "ready"  
  
#score  
score\_value = 0  
font = pygame.font.Font('freesansbold.ttf',32)  
  
textX = 10  
textY = 10  
  
#Game Over text  
  
over\_font = pygame.font.Font('freesansbold.ttf',64)  
  
  
def show\_score(x,y):  
 score = font.render("Score: " + str(score\_value), True, (255, 255, 255))  
 screen.blit(score, (x,y))  
  
def game\_over\_text():  
 over\_text = over\_font.render("GAME OVER", True, (255,255,255))  
 screen.blit(over\_text, (200, 250))  
  
def player(x,y):  
 screen.blit(playerimg,(x,y))  
  
def enemy(x,y,i):  
 screen.blit(enemyimg[i],(x,y))  
  
def fire\_bullet(x,y):  
 global bullet\_state  
 bullet\_state = "fire"  
 screen.blit(bulletimg,(x+16,y+10))  
  
def isCollision(enemyX, enemyY, bulletX, bulletY):  
 #distance = math.sqrt((math.pow((enemyX - bulletX), 2)) + (math.pow((enemyY - bulletY), 2)))  
 distance = math.sqrt(math.pow(enemyX - bulletX, 2) + (math.pow(enemyY - bulletY, 2)))  
 # Rest of your collision detection logic  
  
 if distance < 27:  
 return True  
 else:  
 return False  
  
#Game Loop  
running = True  
while running:  
 #RGB - Red, Green, Blue(0-255)  
 screen.fill((150,200,245,))  
 #Background Image  
 screen.blit(background, (0,0))  
  
 for event in pygame.event.get():  
 if event.type == pygame.QUIT:  
 running = False  
  
 # If keystroke is presses, check whether its right or left  
 if event.type == pygame.KEYDOWN:  
 if event.key == pygame.K\_LEFT:  
 playerX\_change = -0.8  
 if event.key == pygame.K\_RIGHT:  
 playerX\_change = 0.8  
 if event.key == pygame.K\_SPACE:  
 if bullet\_state == "ready":  
 bullet\_Sound = mixer.Sound('laser.wav')  
 bullet\_Sound.play()  
 # Get the current X coordinate of bullet  
 bulletX = playerX  
 fire\_bullet(bulletX,bulletY)  
 if event.type == pygame.KEYUP:  
 if event.key == pygame.K\_LEFT or event.key == pygame.K\_RIGHT:  
 playerX\_change = 0  
  
  
 # Checking for the boundary of player  
 playerX += playerX\_change  
  
 if playerX <= 0:  
 playerX = 0  
 elif playerX >= 736:  
 playerX = 736  
  
 # Enemy movement  
 for i in range(num\_of\_enemies):  
  
 #Game Over  
 if enemyY[i] > 440:  
 for j in range(num\_of\_enemies):  
 enemyY[j] = 2000  
 game\_over\_text()  
 break  
  
 enemyX[i] += enemyX\_change[i]  
 if enemyX[i]<= 0:  
 enemyX\_change[i] = 0.75  
 enemyY[i] += enemyY\_change[i]  
 elif enemyX[i] >= 736:  
 enemyX\_change[i] = -0.75  
 enemyY[i] += enemyY\_change[i]  
 # Collision  
 collision = isCollision(enemyX[i], enemyY[i], bulletX, bulletY)  
 if collision:  
 explosion\_Sound = mixer.Sound('explosion.wav')  
 explosion\_Sound.play()  
 bulletY = 480  
 bullet\_state = "ready"  
 score\_value += 1  
 enemyX[i] = random.randint(0,735)  
 enemyY[i] = random.randint(50,150)  
 enemy(enemyX[i],enemyY[i],i)  
  
 # Bullet Movement  
 if bulletY <=0:  
 bulletY = 480  
 bullet\_state = "ready"  
  
 if bullet\_state == "fire":  
 fire\_bullet(bulletX,bulletY)  
 bulletY -= bulletY\_change  
  
 # Collision  
 collision = isCollision(enemyX[i], enemyY[i], bulletX, bulletY)  
 if collision:  
 bulletY = 480  
 bullet\_state = "ready"  
 score\_value += 1  
 enemyX = random.randint(0,735)  
 enemyY = random.randint(50,150)  
 player(playerX,playerY)  
 show\_score(textX, textY)  
 pygame.display.update()