import pygame  
import random  
  
level = 1  
play\_game = True  
game\_over = False  
  
fps = 30  
width = 750  
height = 700  
alien\_cooldown = 500  
last\_alien\_shoot = pygame.time.get\_ticks()  
rows = level \* 2 + 3  
  
pygame.mixer.init()  
pygame.init()  
screen = pygame.display.set\_mode((width, height))  
pygame.display.set\_caption('GALAXIAN')  
clock = pygame.time.Clock()  
  
background = pygame.image.load('images/background1\_7.jpg')  
img = pygame.transform.scale(background, (750, 800))  
spaceship = pygame.image.load('images/spaceship.png')  
img1 = pygame.transform.scale(spaceship, (60, 50))  
alien = pygame.image.load('images/alien(3).png')  
img2 = pygame.transform.scale(alien, (50, 40))  
lazer1 = pygame.image.load('images/lazer1.png')  
img3 = pygame.transform.scale(lazer1, (20, 20))  
shield = pygame.image.load('images/solar\_panels.png')  
img4 = pygame.transform.scale(shield, (10, 10))  
lazer2 = pygame.image.load('images/lazer2.png')  
img5 = pygame.transform.scale(lazer2, (20, 20))  
boss = pygame.image.load('images/boss.png')  
img6 = pygame.transform.scale(boss, (200, 100))  
  
'''pygame.mixer.music.load('sounds/background\_music.mp3')  
pygame.mixer.music.play(-1)  
pygame.mixer.music.set\_volume(0.20)  
lazer\_sound = pygame.mixer.Sound('sounds/lazer\_sound.mp3')  
lazer\_sound.set\_volume(0.25)  
explosion1\_sound = pygame.mixer.Sound('sounds/explosion1\_sound.mp3')  
explosion1\_sound.set\_volume(0.15)  
alienbullet\_sound = pygame.mixer.Sound('sounds/aliens\_bullet\_sound.mp3')  
alienbullet\_sound.set\_volume(0.15)'''  
  
  
class Spaceship(pygame.sprite.Sprite):  
 def \_\_init\_\_(self):  
 pygame.sprite.Sprite.\_\_init\_\_(self)  
 self.image = img1  
 self.rect = self.image.get\_rect()  
 self.rect.centerx = int(width / 2)  
 self.rect.bottom = height - 50  
 self.initial\_health = 20  
 self.remaining\_health = 20  
  
 def update(self):  
 global play\_game  
 speed = 20  
 key = pygame.key.get\_pressed()  
 if key[pygame.K\_RIGHT]:  
 self.rect.x += speed  
 elif key[pygame.K\_LEFT]:  
 self.rect.x -= speed  
 if self.rect.right > width:  
 self.rect.right = width  
 elif self.rect.left < 0:  
 self.rect.left = 0  
  
 pygame.draw.rect(screen, (255, 0, 0), (self.rect.x, (self.rect.bottom + 5), self.rect.width, 10))  
 if self.remaining\_health > 0:  
 pygame.draw.rect(screen, (0, 255, 0), (  
 self.rect.x, (self.rect.bottom + 5),  
 int((self.rect.width) \* (self.remaining\_health / self.initial\_health)),  
 10))  
 if pygame.sprite.spritecollide(self, alien\_bullet\_group, True):  
 self.remaining\_health -= 1  
 if self.remaining\_health == 0:  
 self.kill()  
  
 def shoot(self):  
 lazer = Lazer(self.rect.centerx, self.rect.top)  
 if len(spaceship\_group) > 0 and len(lazer\_group) <= 5:  
 lazer\_group.add(lazer)  
  
class Lazer(pygame.sprite.Sprite):  
 def \_\_init\_\_(self, x, y):  
 pygame.sprite.Sprite.\_\_init\_\_(self)  
 self.image = img3  
 self.rect = self.image.get\_rect()  
 self.rect.centerx = x  
 self.rect.bottom = y  
 self.speedy = -15  
  
 def update(self):  
 self.rect.y += self.speedy  
 if self.rect.bottom < 0:  
 self.kill()  
 if pygame.sprite.spritecollide(self, aliens\_group, True):  
 self.kill()  
 explosion = Explosion(self.rect.x, self.rect.top)  
 explosion\_group.add(explosion)  
 '''explosion1\_sound.play()'''  
 if pygame.sprite.spritecollide(self, shields\_group, True):  
 self.kill()  
 if pygame.sprite.spritecollide(self, boss\_group, False):  
 self.kill()  
 if len(aliens\_group) == 0:  
 boss.remaining\_health -= 1  
 explosion = Explosion(self.rect.x, self.rect.top)  
 explosion\_group.add(explosion)  
 '''explosion1\_sound.play()'''  
  
class Aliens(pygame.sprite.Sprite):  
 def \_\_init\_\_(self, x, y):  
 pygame.sprite.Sprite.\_\_init\_\_(self)  
 self.image = img2  
 self.rect = self.image.get\_rect()  
 self.rect.center = [x, y]  
 self.move\_direction = 1  
 self.move\_counter = 0  
  
 def update(self):  
 self.rect.x += self.move\_direction  
 self.move\_counter += 5  
 if abs(self.move\_counter) > 500:  
 self.move\_direction \*= -1  
 self.move\_counter \*= self.move\_direction  
 self.rect.y += 5  
  
 def collide(self):  
 if pygame.sprite.spritecollide(self, shields\_group, True):  
 self.kill()  
  
class Explosion(pygame.sprite.Sprite):  
 def \_\_init\_\_(self, x, y):  
 pygame.sprite.Sprite.\_\_init\_\_(self)  
 self.explosion = []  
 for i in range(1, 6):  
 img = pygame.image.load(f'images/exp{i}.png')  
 img = pygame.transform.scale(img, (i \* 5, i \* 5))  
 self.explosion.append(img)  
 self.index = 0  
 self.image = self.explosion[self.index]  
 self.rect = self.image.get\_rect()  
 self.rect.center = [x, y]  
 self.counter = 0  
  
 def update(self):  
 explosion\_speed = 3  
 self.counter += 1  
 if self.counter >= explosion\_speed and self.index < len(self.explosion) - 1:  
 self.counter = 0  
 self.index += 1  
 self.image = self.explosion[self.index]  
 if self.index >= len(self.explosion) - 1 and self.counter >= explosion\_speed:  
 self.kill()  
  
class Alien\_Bullet(pygame.sprite.Sprite):  
 def \_\_init\_\_(self, x, y):  
 pygame.sprite.Sprite.\_\_init\_\_(self)  
 self.image = img5  
 self.rect = self.image.get\_rect()  
 self.rect.center = [x, y]  
 self.speedy = 15  
  
 def update(self):  
 self.rect.y += self.speedy  
 if self.rect.top > height:  
 self.kill()  
 if pygame.sprite.spritecollide(self, spaceship\_group, False):  
 self.kill()  
 spaceship.remaining\_health -= 1  
 if pygame.sprite.spritecollide(self, shields\_group, True):  
 self.kill()  
  
  
class Shields(pygame.sprite.Sprite):  
 def \_\_init\_\_(self):  
 pygame.sprite.Sprite.\_\_init\_\_(self)  
 self.image = img4  
 self.rect = self.image.get\_rect()  
  
 def update(self):  
 if pygame.sprite.spritecollide(self, aliens\_group, False):  
 self.kill()  
  
class Boss(pygame.sprite.Sprite):  
 def \_\_init\_\_(self):  
 pygame.sprite.Sprite.\_\_init\_\_(self)  
 self.image = img6  
 self.rect = self.image.get\_rect()  
 if level == 5:  
 self.rect.center = [325, 150]  
 self.health = 10  
 self.remaining\_health = 10  
 self.move\_direction = 1  
 self.move\_counter = 0  
  
  
 def update(self):  
 pygame.draw.rect(screen, (255, 0, 0), (self.rect.x, (self.rect.bottom-200), self.rect.width, 10))  
 if self.remaining\_health > 0:  
 pygame.draw.rect(screen, (0, 255, 0), (self.rect.x, (self.rect.bottom -200),int((self.rect.width) \* ( self.remaining\_health / self.health)),10))  
 if self.remaining\_health <= 0:  
 self.kill()  
 self.rect.x += 7\*self.move\_direction  
 self.move\_counter += 7  
 if abs(self.move\_counter) > 550:  
 self.move\_direction \*= -1  
 self.move\_counter \*= self.move\_direction  
  
  
class Boss\_Bullets(pygame.sprite.Sprite):  
 def \_\_init\_\_(self, x, y):  
 pygame.sprite.Sprite.\_\_init\_\_(self)  
 self.image = img5  
 self.rect = self.image.get\_rect()  
 self.rect.center = [x, y]  
 self.speedy = 20  
  
 def update(self):  
 self.rect.y += self.speedy  
 if self.rect.top > height:  
 self.kill()  
 if pygame.sprite.spritecollide(self, spaceship\_group, False):  
 self.kill()  
 spaceship.remaining\_health -= 1  
  
spaceship\_group = pygame.sprite.Group()  
spaceship = Spaceship()  
spaceship\_group.add(spaceship)  
lazer\_group = pygame.sprite.Group()  
aliens\_group = pygame.sprite.Group()  
shields\_group = pygame.sprite.Group()  
alien\_bullet\_group = pygame.sprite.Group()  
explosion\_group = pygame.sprite.Group()  
boss\_group = pygame.sprite.Group()  
boss\_bullets\_group = pygame.sprite.Group()  
boss = Boss()  
boss\_group.add(boss)  
  
def create\_aliens(a, x, y):  
 for row in range(1, a):  
 for column in range(1, 11):  
 alien = Aliens((x + (50 \* column)), (y + (50 \* row)))  
 aliens\_group.add(alien)  
  
  
def start():  
 global play\_game, last\_alien\_shoot, level  
 play\_game1 = play\_game  
 while play\_game1:  
 clock.tick(fps)  
 screen.fill((0, 0, 0))  
 screen.blit(img, (0, 0))  
  
 time\_now = pygame.time.get\_ticks()  
 if time\_now - last\_alien\_shoot > alien\_cooldown and len(aliens\_group) != 0:  
 attacking\_alien = random.choice(aliens\_group.sprites())  
 alien\_bullet = Alien\_Bullet(attacking\_alien.rect.centerx, attacking\_alien.rect.bottom)  
 alien\_bullet\_group.add(alien\_bullet)  
 last\_alien\_shoot = time\_now  
 '''alienbullet\_sound.play()'''  
  
 if level == 5:  
 if time\_now - last\_alien\_shoot > alien\_cooldown:  
 if len(aliens\_group) == 0:  
 if spaceship.rect.x >= boss.rect.left and spaceship.rect.x <= boss.rect.right:  
 boss\_bullet = Boss\_Bullets(spaceship.rect.x, boss.rect.bottom)  
 boss\_bullets\_group.add(boss\_bullet)  
 '''alienbullet\_sound.play()'''  
 elif spaceship.rect.x <= boss.rect.left:  
 boss\_bullet = Boss\_Bullets(boss.rect.x, boss.rect.bottom)  
 boss\_bullets\_group.add(boss\_bullet)  
 '''alienbullet\_sound.play()'''  
 elif spaceship.rect.x >= boss.rect.right:  
 boss\_bullet = Boss\_Bullets(boss.rect.right, boss.rect.bottom)  
 boss\_bullets\_group.add(boss\_bullet)  
 last\_alien\_shoot = pygame.time.get\_ticks()  
  
 for event in pygame.event.get():  
 if event.type == pygame.QUIT:  
 play\_game = False  
 play\_game1 = False  
 elif event.type == pygame.KEYDOWN:  
 if event.key == pygame.K\_SPACE:  
 spaceship.shoot()  
 '''lazer\_sound.play()'''  
  
 spaceship\_group.draw(screen)  
 lazer\_group.draw(screen)  
 explosion\_group.draw(screen)  
 spaceship\_group.update()  
 lazer\_group.update()  
 explosion\_group.update()  
  
  
 aliens\_group.draw(screen)  
 shields\_group.draw(screen)  
 alien\_bullet\_group.draw(screen)  
 aliens\_group.update()  
 shields\_group.update()  
 alien\_bullet\_group.update()  
 if level == 5:  
 boss\_group.add(boss)  
 boss\_group.draw(screen)  
 boss\_bullets\_group.draw(screen)  
 boss\_group.update()  
 boss\_bullets\_group.update()  
  
 pygame.display.update()  
 if len(aliens\_group) == 0 and level < 5:  
 play\_game1 = False  
 elif len(boss\_group) == 0 and level == 5:  
 play\_game1 = False  
'''  
create\_aliens(5, 80, 30)  
spaceship.remaining\_health = spaceship.initial\_health  
level = 1  
for shield in range(4):  
 for row in range(5):  
 for column in range(10):  
 shields = Shields()  
 shields.rect.x = (50 + (195 \* shield)) + (10 \* column)  
 shields.rect.y = 500 + (10 \* row)  
 shields\_group.add(shields)  
start()  
  
create\_aliens(7, 80, 30)  
spaceship.remaining\_health = spaceship.initial\_health  
fps = 45  
level = 2  
alien\_cooldown = 400  
for shield in range(4):  
 for row in range(3):  
 for column in range(10):  
 shields = Shields()  
 shields.rect.x = (50 + (195 \* shield)) + (10 \* column)  
 shields.rect.y = 500 + (10 \* row)  
 shields\_group.add(shields)  
start()  
  
create\_aliens(9, 80, 30)  
spaceship.remaining\_health = spaceship.initial\_health  
fps = 60  
level = 3  
alien\_cooldown = 300  
for shield in range(4):  
 for row in range(3):  
 for column in range(5):  
 shields = Shields()  
 shields.rect.x = (50 + (195 \* shield)) + (10 \* column)  
 shields.rect.y = 500 + (10 \* row)  
 shields\_group.add(shields)  
start()  
  
create\_aliens(9, 80, 30)  
spaceship.remaining\_health = spaceship.initial\_health  
fps = 60  
level = 4  
alien\_cooldown = 200  
for shield in range(1):  
 for row in range(5):  
 for column in range(20):  
 shields = Shields()  
 shields.rect.x = (290 + (195 \* shield)) + (10 \* column)  
 shields.rect.y = 500 + (10 \* row)  
 shields\_group.add(shields)  
start()  
'''  
level = 5  
create\_aliens(7, 80, 150)  
spaceship.remaining\_health = spaceship.initial\_health  
fps = 60  
alien\_cooldown = 500  
for shield in range(4):  
 for row in range(3):  
 for column in range(10):  
 shields = Shields()  
 shields.rect.x = (50 + (195 \* shield)) + (10 \* column)  
 shields.rect.y = 500 + (10 \* row)  
 shields\_group.add(shields)  
start()  
  
pygame.quit()