import pygame

from pygame import mixer

from pygame.locals import \*

import random

pygame.mixer.pre\_init(44100, -16, 2, 512)

mixer.init()

pygame.init()

*#define fps*

clock = pygame.time.Clock()

fps = 60

screen\_width = 600

screen\_height = 800

screen = pygame.display.set\_mode((screen\_width, screen\_height))

pygame.display.set\_caption('Space Duck')

*#define fonts*

font30 = pygame.font.SysFont('Constantia', 30)

font40 = pygame.font.SysFont('Constantia', 40)

*#load sounds*

explosion\_fx = pygame.mixer.Sound("img/splash.wav")

explosion\_fx.set\_volume(0.25)

explosion2\_fx = pygame.mixer.Sound("img/splash2.wav")

explosion2\_fx.set\_volume(0.25)

laser\_fx = pygame.mixer.Sound("img/duck.wav")

laser\_fx.set\_volume(0.25)

*#define game variables*

rows = 5

cols = 5

alien\_cooldown = 1000*#bullet cooldown in milliseconds*

last\_alien\_shot = pygame.time.get\_ticks()

countdown = 3

last\_count = pygame.time.get\_ticks()

game\_over = 0*#0 is no game over, 1 means player has won, -1 means player has lost*

*#define colours*

red = (255, 0, 0)

green = (0, 255, 0)

white = (255, 255, 255)

*#load image*

bg = pygame.image.load("img/bg.png")

**def** draw\_bg():

  screen.blit(bg, (0, 0))

*#define function for creating text*

**def** draw\_text(text, font, text\_col, x, y):

  img = font.render(text, True, text\_col)

  screen.blit(img, (x, y))

*#create spaceship class*

**class** Spaceship(pygame.sprite.Sprite):

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

    pygame.sprite.Sprite.\_\_init\_\_(self)

    self.image = pygame.image.load("img/duck.png")

    self.rect = self.image.get\_rect()

    self.rect.center = [x, y]

    self.health\_start = health

    self.health\_remaining = health

    self.last\_shot = pygame.time.get\_ticks()

**def** update(self):

*#set movement speed*

    speed = 8

*#set a cooldown variable*

    cooldown = 500 *#milliseconds*

    game\_over = 0

*#get key press*

    key = pygame.key.get\_pressed()

    if key[pygame.K\_LEFT] and self.rect.left > 0:

      self.rect.x -= speed

    if key[pygame.K\_RIGHT] and self.rect.right < screen\_width:

      self.rect.x += speed

*#record current time*

    time\_now = pygame.time.get\_ticks()

*#shoot*

    if key[pygame.K\_SPACE] and time\_now - self.last\_shot > cooldown:

      laser\_fx.play()

      bullet = Bullets(self.rect.centerx, self.rect.top)

      bullet\_group.add(bullet)

      self.last\_shot = time\_now

*#update mask*

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

*#draw health bar*

    pygame.draw.rect(screen, red, (self.rect.x, (self.rect.bottom + 10), self.rect.width, 15))

    if self.health\_remaining > 0:

      pygame.draw.rect(screen, green, (self.rect.x, (self.rect.bottom + 10), int(self.rect.width \* (self.health\_remaining / self.health\_start)), 15))

    elif self.health\_remaining <= 0:

      explosion = Explosion(self.rect.centerx, self.rect.centery, 3)

      explosion\_group.add(explosion)

      self.kill()

      game\_over = -1

    return game\_over

*#create Bullets class*

**class** Bullets(pygame.sprite.Sprite):

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

    pygame.sprite.Sprite.\_\_init\_\_(self)

    self.image = pygame.image.load("img/gbullet.png")

    self.rect = self.image.get\_rect()

    self.rect.center = [x, y]

**def** update(self):

    self.rect.y -= 5

    if self.rect.bottom < 0:

      self.kill()

    if pygame.sprite.spritecollide(self, alien\_group, True):

      self.kill()

      explosion\_fx.play()

      explosion = Explosion(self.rect.centerx, self.rect.centery, 2)

      explosion\_group.add(explosion)

*#create Aliens class*

**class** Aliens(pygame.sprite.Sprite):

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

    pygame.sprite.Sprite.\_\_init\_\_(self)

    self.image = pygame.image.load("img/alien" + str(random.randint(1, 5)) + ".png")

    self.rect = self.image.get\_rect()

    self.rect.center = [x, y]

    self.move\_counter = 0

    self.move\_direction = 1

**def** update(self):

    self.rect.x += self.move\_direction

    self.move\_counter += 1

    if abs(self.move\_counter) > 75:

      self.move\_direction \*= -1

      self.move\_counter \*= self.move\_direction

*#create Alien Bullets class*

**class** Alien\_Bullets(pygame.sprite.Sprite):

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

    pygame.sprite.Sprite.\_\_init\_\_(self)

    self.image = pygame.image.load("img/bbullet.png")

    self.rect = self.image.get\_rect()

    self.rect.center = [x, y]

**def** update(self):

    self.rect.y += 2

    if self.rect.top > screen\_height:

      self.kill()

    if pygame.sprite.spritecollide(self, spaceship\_group, False, pygame.sprite.collide\_mask):

      self.kill()

      explosion2\_fx.play()

*#reduce spaceship health*

      spaceship.health\_remaining -= 1

      explosion = Explosion(self.rect.centerx, self.rect.centery, 1)

      explosion\_group.add(explosion)

*#create Explosion class*

**class** Explosion(pygame.sprite.Sprite):

**def** \_\_init\_\_(self, x, y, size):

    pygame.sprite.Sprite.\_\_init\_\_(self)

    self.images = []

    for num in range(1, 6):

      img = pygame.image.load(**f**"img/exp{num}.png")

      if size == 1:

        img = pygame.transform.scale(img, (20, 20))

      if size == 2:

        img = pygame.transform.scale(img, (40, 40))

      if size == 3:

        img = pygame.transform.scale(img, (160, 160))

*#add the image to the list*

      self.images.append(img)

    self.index = 0

    self.image = self.images[self.index]

    self.rect = self.image.get\_rect()

    self.rect.center = [x, y]

    self.counter = 0

**def** update(self):

    explosion\_speed = 3

*#update explosion animation*

    self.counter += 1

    if self.counter >= explosion\_speed and self.index < len(self.images) - 1:

      self.counter = 0

      self.index += 1

      self.image = self.images[self.index]

*#if the animation is complete, delete explosion*

    if self.index >= len(self.images) - 1 and self.counter >= explosion\_speed:

      self.kill()

*#create sprite groups*

spaceship\_group = pygame.sprite.Group()

bullet\_group = pygame.sprite.Group()

alien\_group = pygame.sprite.Group()

alien\_bullet\_group = pygame.sprite.Group()

explosion\_group = pygame.sprite.Group()

**def** create\_aliens():

*#generate aliens*

  for row in range(rows):

    for item in range(cols):

      alien = Aliens(100 + item \* 100, 100 + row \* 70)

      alien\_group.add(alien)

create\_aliens()

*#create player*

spaceship = Spaceship(int(screen\_width / 2), screen\_height - 100, 3)

spaceship\_group.add(spaceship)

run = True

while run:

  clock.tick(fps)

*#draw background*

  draw\_bg()

  if countdown == 0:

*#create random alien bullets*

*#record current time*

    time\_now = pygame.time.get\_ticks()

*#shoot*

    if time\_now - last\_alien\_shot > alien\_cooldown and len(alien\_bullet\_group) < 5 and len(alien\_group) > 0:

      attacking\_alien = random.choice(alien\_group.sprites())

      alien\_bullet = Alien\_Bullets(attacking\_alien.rect.centerx, attacking\_alien.rect.bottom)

      alien\_bullet\_group.add(alien\_bullet)

      last\_alien\_shot = time\_now

*#check if all the aliens have been killed*

    if len(alien\_group) == 0:

      game\_over = 1

    if game\_over == 0:

*#update spaceship*

      game\_over = spaceship.update()

*#update sprite groups*

      bullet\_group.update()

      alien\_group.update()

      alien\_bullet\_group.update()

    else:

      if game\_over == -1:

        draw\_text('GAME OVER!', font40, white, int(screen\_width / 2 - 100), int(screen\_height / 2 + 50))

      if game\_over == 1:

        draw\_text('YOU WIN!', font40, white, int(screen\_width / 2 - 100), int(screen\_height / 2 + 50))

  if countdown > 0:

    draw\_text('GET READY!', font40, white, int(screen\_width / 2 - 110), int(screen\_height / 2 + 50))

    draw\_text(str(countdown), font40, white, int(screen\_width / 2 - 10), int(screen\_height / 2 + 100))

    count\_timer = pygame.time.get\_ticks()

    if count\_timer - last\_count > 1000:

      countdown -= 1

      last\_count = count\_timer

*#update explosion group*

  explosion\_group.update()

*#draw sprite groups*

  spaceship\_group.draw(screen)

  bullet\_group.draw(screen)

  alien\_group.draw(screen)

  alien\_bullet\_group.draw(screen)

  explosion\_group.draw(screen)

*#event handlers*

  for event in pygame.event.get():

    if event.type == pygame.QUIT:

      run = False

  pygame.display.update()

pygame.quit()