import pygame.font

class Button():

    def \_\_init\_\_(*self*, *screen*, *msg*):

        """Initialize button attributes."""

*self*.width, *self*.height = 200, 50

*self*.screen = *screen*

*self*.screen\_rect = *screen*.get\_rect()

        # Set the dimensions and properties of the button. self.width, self.height = 200, 50

*self*.button\_color = (0, 0, 0)

*self*.text\_color = (255, 255, 255)

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

        # Build the button's rect object and center it.

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

*self*.rect.center = *self*.screen\_rect.center

        # The button message needs to be prepped only once.

*self*.prep\_msg(*msg*)

    def prep\_msg(*self*, *msg*):

        """Turn msg into a rendered image and center text on the button."""

*self*.msg\_image = *self*.font.render(*msg*, True, *self*.text\_color,

*self*.button\_color)

*self*.msg\_image\_rect = *self*.msg\_image.get\_rect()

*self*.msg\_image\_rect.center = *self*.rect.center

    def draw\_button(*self*):

        # Draw blank button and then draw message.

*self*.screen.fill(*self*.button\_color, *self*.rect)

*self*.screen.blit(*self*.msg\_image, *self*.msg\_image\_rect)

import pygame

class Background\_pause(*pygame*.*sprite*.*Sprite*):

    def \_\_init\_\_(*self*, *image\_file*, *location*):

        pygame.sprite.Sprite.\_\_init\_\_(*self*)  #call Sprite initializer

*self*.image = pygame.image.load(*image\_file*)

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

*self*.rect.left = *location*

*self*.rect.top = *location*

class Background\_game(*pygame*.*sprite*.*Sprite*):

    def \_\_init\_\_(*self*, *image\_file*, *location*):

        pygame.sprite.Sprite.\_\_init\_\_(*self*)  #call Sprite initializer

*self*.image = pygame.image.load(*image\_file*)

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

*self*.rect.left = *location*

*self*.rect.top = *location*

import pygame

import random

class Fruit:

    def \_\_init\_\_(*self*, *ai\_settings*, *screen*):

*self*.ai\_settings = *ai\_settings*

*self*.screen = *screen*

*self*.image = pygame.image.load('images/apple.bmp')

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

*self*.rect.x =  random.randint(50, *ai\_settings*.screen\_width-50)

*self*.rect.y =  random.randint(30, *ai\_settings*.screen\_height-30)

*self*.x = float(*self*.rect.x)

*self*.y = float(*self*.rect.y)

    def blitme(*self*):

        """Draw the apple at its current location."""

*self*.screen.blit(*self*.image, *self*.rect)

import pygame

import time

import sys

from pygame import mixer

import game\_functions as gf

class Snake:

    def \_\_init\_\_(*self*, *ai\_settings*):

*self*.x = *ai\_settings*.screen\_width/2

*self*.y = *ai\_settings*.screen\_height/2

*self*.x\_change = 0

*self*.y\_change = 0

*self*.rect = pygame.Rect(*self*.x, *self*.y, *ai\_settings*.snake\_width,

*ai\_settings*.snake\_height)

*self*.snake\_list = []

*self*.length\_snake = 1

        #self.apple\_sound1 = mixer.Sound("music/mary\_sound1.wav")

    def move\_snake(*self*, *ai\_settings*):

        for event in pygame.event.get():

            if event.type == pygame.QUIT:

                sys.exit()

            if event.type == pygame.KEYDOWN:

                if event.key == pygame.K\_RIGHT and *self*.x\_change >= 0: #Move the snake only if it is not moving in the opposite direction

                    #self.apple\_sound1.play()

*self*.x\_change = *ai\_settings*.snake\_speed

*self*.y\_change = 0

                elif event.key == pygame.K\_LEFT and *self*.x\_change <= 0:

                    #apple\_sound = mixer.Sound("music/mary\_sound1.wav")

                    #self.apple\_sound1.play()

*self*.x\_change = -*ai\_settings*.snake\_speed

*self*.y\_change = 0

                elif event.key == pygame.K\_UP and *self*.y\_change <= 0:

*self*.y\_change = -*ai\_settings*.snake\_speed

*self*.x\_change = 0

                    #apple\_sound = mixer.Sound("music/mary\_sound1.wav")

                    #apple\_sound.play()

                elif event.key == pygame.K\_DOWN and *self*.y\_change >= 0:

                    #apple\_sound = mixer.Sound("music/mary\_sound1.wav")

                    #apple\_sound.play()

*self*.y\_change = *ai\_settings*.snake\_speed

*self*.x\_change = 0

*self*.rect.x += *self*.x\_change

*self*.rect.y += *self*.y\_change

*self*.x = float(*self*.rect.x)

*self*.y = float(*self*.rect.y)

    def in\_game(*self*, *ai\_settings*, *screen*):

        difficulty = "hard"

        if difficulty == "hard":

            #self.x += self.x\_change

            if *self*.rect.x >= *ai\_settings*.screen\_width - 10:

                #self.rect.x = 0

*self*.reset(*ai\_settings*, *screen*)

            elif *self*.rect.x < 0 - 5:

                #self.rect.x = ai\_settings.screen\_width

*self*.reset(*ai\_settings*, *screen*)

            #self.y += self.y\_change

            if *self*.rect.y >= *ai\_settings*.screen\_height:

                #self.rect.y = 0

*self*.reset(*ai\_settings*, *screen*)

            elif *self*.rect.y < 0 - 5:

                #self.rect.y = ai\_settings.screen\_height

*self*.reset(*ai\_settings*, *screen*)

    def draw\_snake(*self*, *screen*, *ai\_settings*):

        for i in range(len(*self*.snake\_list)):

            pygame.draw.rect(*screen*, *ai\_settings*.snake\_color,

            [*self*.snake\_list[i][0], *self*.snake\_list[i][1], *ai\_settings*.snake\_width,

*ai\_settings*.snake\_height])

            # x = snake\_list[i][0]

            # y = snake\_list[i][1]

    def snake\_update(*self*, *ai\_settings*, *screen*):

*self*.snake\_head = []

*self*.snake\_head.append(*self*.x)

*self*.snake\_head.append(*self*.y)

*self*.snake\_list.append(*self*.snake\_head)

        if len(*self*.snake\_list) > *self*.length\_snake:

            del *self*.snake\_list[0]

        for x in *self*.snake\_list[:-1]:

            if x == *self*.snake\_head:

*self*.reset(*ai\_settings*, *screen*)

    def reset(*self*, *ai\_settings*, *screen*):

*ai\_settings*.game\_over = True

*ai\_settings*.score = 0

        gf.message("Game Over!", *ai\_settings*.message\_color,

*ai\_settings*, *screen*, 2.5, 2.5)

        pygame.display.update()

*self*.x\_change = 0

*self*.y\_change = 0

*self*.rect.x = *ai\_settings*.screen\_width/2

*self*.rect.y = *ai\_settings*.screen\_height/2

*self*.x = float(*self*.rect.x)

*self*.y = float(*self*.rect.y)

*self*.length\_snake = 1

*self*.snake\_list.clear()

        mixer.music.pause()

        game\_over\_sound = mixer.Sound("music/game\_over.wav")

        game\_over\_sound.play()

        time.sleep(2)

        mixer.music.unpause()

import pygame

class Settings():

    """A class to store all settings for Alien Invasion."""

    def \_\_init\_\_(*self*):

        """Initialize the game's settings."""

        # Screen settings

*self*.screen\_width = 600

*self*.screen\_height = 600

*self*.bg\_color = (126,200,80) #Green

        # Snake settings

*self*.snake\_color = 0, 0, 0 #Black

*self*.snake\_width = 15

*self*.snake\_height = 15

*self*.snake\_speed = 10

        #Score settings

*self*.score = 0

*self*.score\_font = pygame.font.SysFont("comicsansms", 20)

*self*.score\_color = (255, 255, 255) #White

*self*.message\_style = pygame.font.SysFont("bahnschrift", 35)

*self*.message\_color = (255, 0, 0)

        #Menu settings

*self*.before\_color = (255, 255, 255)

*self*.game\_over = True

import random

import pygame

from pygame import mixer

import sys

def snake\_and\_fruit(*snake*, *fruit*, *ai\_settings*):

    #What happends when tha snake eats the fruit

    if *snake*.rect.colliderect(*fruit*.rect):

        #If there is music I lower it and play eating sound

        mixer.music.set\_volume(0.1)

        mixer.music.pause()

        apple\_sound = mixer.Sound("music/apple\_sound4.wav")

        apple\_sound.play()

        #New spawn point for the fruit

*fruit*.rect.x =  random.randint(30, *ai\_settings*.screen\_width-30)

*fruit*.rect.y =  random.randint(30, *ai\_settings*.screen\_height-30)

        for x in *snake*.snake\_list[:-1]:

            while *fruit*.rect.x == x[0] or *fruit*.rect.y == x[1]:

*fruit*.rect.x =  random.randint(30, *ai\_settings*.screen\_width-30)

*fruit*.rect.y =  random.randint(30, *ai\_settings*.screen\_height-30)

*fruit*.x = float(*fruit*.rect.x)

*fruit*.y = float(*fruit*.rect.y)

*snake*.length\_snake += 2  #How fast the snake grows

*ai\_settings*.score += 1   #Increase the score

        mixer.music.unpause()

        mixer.music.set\_volume(0.5)

def score(*ai\_settings*, *screen*):

    #Display score function

    value = *ai\_settings*.score\_font.render("Score: " + str(*ai\_settings*.score), True, *ai\_settings*.score\_color)

*screen*.blit(value, [10, 10])

def message(*msg*, *color*,*ai\_settings*, *screen*,*w* ,*h*):

    #Display message function

    mesg = *ai\_settings*.message\_style.render(*msg*, True, *color*)

*screen*.blit(mesg, [*ai\_settings*.screen\_width / *w*, *ai\_settings*.screen\_height / *h*])

def menu(*ai\_settings*, *screen*, *background\_ps*, *play\_button*):

    while *ai\_settings*.game\_over == True:

        #mixer.music.rewind()

*screen*.fill(*ai\_settings*.before\_color)

*screen*.blit(*background\_ps*.image, *background\_ps*.rect)

*play\_button*.draw\_button()

        #gf.score(ai\_settings, screen)     #Optional Score while in the menu

        pygame.display.update()

        for event in pygame.event.get():

            if event.type == pygame.KEYDOWN:

                if event.key == pygame.K\_q:

                    sys.exit()

                if event.key == pygame.K\_p:

*ai\_settings*.game\_over = False

            elif event.type == pygame.MOUSEBUTTONDOWN:

                mouse\_x, mouse\_y = pygame.mouse.get\_pos()

                if *play\_button*.rect.collidepoint(mouse\_x, mouse\_y):

*ai\_settings*.game\_over = False

            elif event.type == pygame.QUIT:

                sys.exit()

import pygame

import game\_functions as gf

from settings import Settings

from the\_snake import Snake

from the\_fruit import Fruit

from button import Button

from background import Background\_pause

from background import Background\_game

from pygame import mixer

pygame.mixer.pre\_init(44100, -16, 1, 512)

pygame.init()

ai\_settings = Settings()

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

snake = Snake(ai\_settings)

fruit = Fruit(ai\_settings, screen)

clock = pygame.time.Clock()

#background\_game = Background\_game("images/grass\_bg.png", [0][0])

background\_ps = Background\_pause("images/snake\_bg2.png", [0][0])

play\_button = Button(screen, "Play")

pygame.display.set\_caption("Snake Game")

#game\_over = True

mixer.music.load("music/snake\_song.mp3")

mixer.music.play(*loops* = -1, *start* = 5)

mixer.music.set\_volume(0.5)

#pygame.mixer.Channel(0).play(pygame.mixer.Sound("music/snake\_song.mp3"))

#pygame.mixer.Channel(1).play(pygame.mixer.Sound('music\game\_over.mp3'))

while True:

    mixer.music.rewind()

    gf.menu(ai\_settings, screen, background\_ps, play\_button)

    snake.move\_snake(ai\_settings)

    snake.in\_game(ai\_settings, screen)

    screen.fill(ai\_settings.bg\_color)

    #screen.blit(background\_game.image, background\_game.rect)

    gf.score(ai\_settings, screen)

    fruit.blitme()

    snake.snake\_update(ai\_settings,screen)

    snake.draw\_snake(screen, ai\_settings)

    pygame.display.update()

    gf.snake\_and\_fruit(snake, fruit, ai\_settings)

    clock.tick(30)