import pygame

from player import Player

from enemies import \*

import tkinter

from tkinter import messagebox

SCREEN\_WIDTH = 800

SCREEN\_HEIGHT = 576

# Define some colors

BLACK = (0,0,0)

WHITE = (255,255,255)

BLUE = (0,0,255)

RED = (255,0,0)

class Game(object):

def \_init\_(self):

self.font = pygame.font.Font(None,40)

self.about = False

self.game\_over = True

# Create the variable for the score

self.score = 0

# Create the font for displaying the score on the screen

self.font = pygame.font.Font(None,35)

# Create the menu of the game

self.menu = Menu(("Start","About","Exit"),font\_color = WHITE,font\_size=60)

# Create the player

self.player = Player(32,128,"player.png")

# Create the blocks that will set the paths where the player can go

self.horizontal\_blocks = pygame.sprite.Group()

self.vertical\_blocks = pygame.sprite.Group()

# Create a group for the dots on the screen

self.dots\_group = pygame.sprite.Group()

# Set the enviroment:

for i,row in enumerate(enviroment()):

for j,item in enumerate(row):

if item == 1:

self.horizontal\_blocks.add(Block(j\*32+8,i\*32+8,BLACK,16,16))

elif item == 2:

self.vertical\_blocks.add(Block(j\*32+8,i\*32+8,BLACK,16,16))

# Create the enemies

self.enemies = pygame.sprite.Group()

self.enemies.add(Slime(288,96,0,2))

self.enemies.add(Slime(288,320,0,-2))

self.enemies.add(Slime(544,128,0,2))

self.enemies.add(Slime(32,224,0,2))

self.enemies.add(Slime(160,64,2,0))

self.enemies.add(Slime(448,64,-2,0))

self.enemies.add(Slime(640,448,2,0))

self.enemies.add(Slime(448,320,2,0))

# Add the dots inside the game

for i, row in enumerate(enviroment()):

for j, item in enumerate(row):

if item != 0:

self.dots\_group.add(Ellipse(j\*32+12,i\*32+12,WHITE,8,8))

# Load the sound effects

self.pacman\_sound = pygame.mixer.Sound("pacman\_sound.ogg")

self.game\_over\_sound = pygame.mixer.Sound("game\_over\_sound.ogg")

def process\_events(self):

for event in pygame.event.get(): # User did something

if event.type == pygame.QUIT: # If user clicked close

return True

self.menu.event\_handler(event)

if event.type == pygame.KEYDOWN:

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

if self.game\_over and not self.about:

if self.menu.state == 0:

self.\_init\_()

self.game\_over = False

elif self.menu.state == 1:

self.about = True

elif self.menu.state == 2:

return True

elif event.key == pygame.K\_RIGHT:

self.player.move\_right()

elif event.key == pygame.K\_LEFT:

self.player.move\_left()

elif event.key == pygame.K\_UP:

self.player.move\_up()

elif event.key == pygame.K\_DOWN:

self.player.move\_down()

elif event.key == pygame.K\_ESCAPE:

self.game\_over = True

self.about = False

elif event.type == pygame.KEYUP:

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

self.player.stop\_move\_right()

elif event.key == pygame.K\_LEFT:

self.player.stop\_move\_left()

elif event.key == pygame.K\_UP:

self.player.stop\_move\_up()

elif event.key == pygame.K\_DOWN:

self.player.stop\_move\_down()

elif event.type == pygame.MOUSEBUTTONDOWN:

self.player.explosion = True

return False

def run\_logic(self):

if not self.game\_over:

self.player.update(self.horizontal\_blocks,self.vertical\_blocks)

block\_hit\_list = pygame.sprite.spritecollide(self.player,self.dots\_group,True)

if len(block\_hit\_list):

self.pacman\_sound.play()

self.score += 1

block\_hit\_list = pygame.sprite.spritecollide(self.player,self.enemies,True)

if len(block\_hit\_list):

self.player.explosion = True

self.game\_over\_sound.play()

self.game\_over = self.player.game\_over

self.enemies.update(self.horizontal\_blocks,self.vertical\_blocks)

def display\_frame(self,screen):

screen.fill(BLACK)

if self.game\_over:

if self.about:

self.display\_message(screen,"It is an arcade Game")

else:

self.menu.display\_frame(screen)

else:

self.horizontal\_blocks.draw(screen)

self.vertical\_blocks.draw(screen)

draw\_enviroment(screen)

self.dots\_group.draw(screen)

self.enemies.draw(screen)

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

text = self.font.render("Score: " + str(self.score),True,GREEN)

pygame.display.flip()

def display\_message(self,screen,message,color=(255,0,0)):

label = self.font.render(message,True,color)

width = label.get\_width()

height = label.get\_height()

# Determine the position of the label

posX = (SCREEN\_WIDTH /2) - (width /2)

posY = (SCREEN\_HEIGHT /2) - (height /2)

# Draw the label onto the screen

screen.blit(label,(posX,posY))

class Menu(object):

state = 0

def \_init\_(self,items,font\_color=(0,0,0),select\_color=(255,0,0),ttf\_font=None,font\_size=25):

self.font\_color = font\_color

self.select\_color = select\_color

self.items = items

self.font = pygame.font.Font(ttf\_font,font\_size)

def display\_frame(self,screen):

for index, item in enumerate(self.items):

if self.state == index:

label = self.font.render(item,True,self.select\_color)

else:

label = self.font.render(item,True,self.font\_color)

width = label.get\_width()

height = label.get\_height()

posX = (SCREEN\_WIDTH /2) - (width /2)

t\_h = len(self.items) \* height

posY = (SCREEN\_HEIGHT /2) - (t\_h /2) + (index \* height)

screen.blit(label,(posX,posY))

def event\_handler(self,event):

if event.type == pygame.KEYDOWN:

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

if self.state:

self.state -= 1

elif event.key == pygame.K\_DOWN:

if self.state &lt: len(self.items) -1

self.state += 1