import pygame

import random

# Initialize pygame

pygame.init()

# Game screen

SCREEN\_WIDTH = 800

SCREEN\_HEIGHT = 600

screen = pygame.display.set\_mode((SCREEN\_WIDTH, SCREEN\_HEIGHT))

pygame.display.set\_caption("Side Scroller")

# Colors

WHITE = (255, 255, 255)

BLACK = (0, 0, 0)

RED = (255, 0, 0)

# Game fonts

font = pygame.font.SysFont("Arial", 30)

font\_small = pygame.font.SysFont("Arial", 20)

# Game variables

scroll\_thresh = 200

gravity = 1

# Player variables

current\_level= 1

player\_lives = 3

player\_health = 100

player\_max\_health = 100

projectile\_damage = 25

# Enemy variables

enemy\_health = 100

enemy\_damage = 25

# Levels

level\_1 = [

"XXXXXXXXXXXX",

"X X",

"X X",

"X X",

"X E X",

"X X X",

"X XXXXXX",

"X X",

"XXXXXXXXXXXX"]

level\_2 = [

"XXXXXXXXXXXXXX",

"X X",

"X X",

"X X",

"X E X",

"X X",

"X X",

"X XXXX X",

"X X",

"XXXXXXXXXXXXXX"]

level\_3 = [

"XXXXXXXXXXXXXXXXX",

"X X",

"X X",

"X X",

"X X",

"X X",

"X E X",

"X X",

"X XXXXXXXXX X",

"X X",

"XXXXXXXXXXXXXXXXX"]

levels = [level\_1, level\_2, level\_3]

# Load images

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

player\_img = pygame.image.load("player.png")

projectile\_img = pygame.image.load("projectile.png")

enemy\_img = pygame.image.load("enemy.png")

health\_pickup\_img = pygame.image.load("health.png")

ammo\_pickup\_img = pygame.image.load("ammo.png")

# Sprite classes

class Player(pygame.sprite.Sprite):

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

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

self.image = player\_img

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

self.rect.center = (x, y)

self.health = player\_health

self.max\_health = player\_max\_health

self.direction = 1

self.vel\_y = 0

self.jump = False

self.alive = True

def update(self):

dx = 0

dy = 0

walk\_speed = 10

jump\_speed = 15

key = pygame.key.get\_pressed()

if key[pygame.K\_SPACE] and self.alive:

self.vel\_y = -jump\_speed

self.jump = True

if key[pygame.K\_LEFT]:

dx -= walk\_speed

self.direction = -1

if key[pygame.K\_RIGHT]:

dx += walk\_speed

self.direction = 1

def apply\_gravity(self):

dx=0

dy=0

# Gravity

self.vel\_y += gravity

if self.vel\_y > 10:

self.vel\_y = 10

dy += self.vel\_y