import pygame

import random

import sys

import time

import os

# Initialize pygame

pygame.init()

pygame.mixer.init()

# Constants

WIDTH, HEIGHT = 600, 600

CELL\_SIZE = 20

GRID\_WIDTH = WIDTH // CELL\_SIZE

GRID\_HEIGHT = HEIGHT // CELL\_SIZE

# Colors

WHITE = (255, 255, 255)

GREEN = (0, 200, 0)

RED = (200, 0, 0)

BLACK = (0, 0, 0)

YELLOW = (255, 255, 0)

# Setup

win = pygame.display.set\_mode((WIDTH, HEIGHT))

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

clock = pygame.time.Clock()

font = pygame.font.SysFont("Arial", 24)

# Universal path to assets folder

ASSETS = os.path.join(os.path.dirname(\_\_file\_\_), "assets")

# Sounds

eat\_sound = pygame.mixer.Sound(os.path.join(ASSETS, "coin.mp3"))

crash\_sound = pygame.mixer.Sound(os.path.join(ASSETS, "crash.mp3"))

# Snake and food setup

snake = [(5, 5)]

direction = (1, 0)

score = 0

speed = 10

game\_over = False

# Food values and timers

food\_weights = [1, 2, 3]

food = (random.randint(0, GRID\_WIDTH - 1), random.randint(0, GRID\_HEIGHT - 1))

food\_value = random.choice(food\_weights)

food\_timer = time.time()

FOOD\_LIFETIME = 5 # seconds

def draw\_grid():

win.fill(GREEN)

# Draw food

pygame.draw.rect(win, RED, (food[0]\*CELL\_SIZE, food[1]\*CELL\_SIZE, CELL\_SIZE, CELL\_SIZE))

food\_text = font.render(str(food\_value), True, BLACK)

win.blit(food\_text, (food[0]\*CELL\_SIZE + 5, food[1]\*CELL\_SIZE))

# Draw snake

for segment in snake:

pygame.draw.rect(win, BLACK, (segment[0]\*CELL\_SIZE, segment[1]\*CELL\_SIZE, CELL\_SIZE, CELL\_SIZE))

# Draw score

score\_text = font.render(f"Score: {score}", True, WHITE)

win.blit(score\_text, (10, 10))

pygame.display.update()

def reset\_game():

global snake, direction, food, food\_value, score, speed, game\_over, food\_timer

snake = [(5, 5)]

direction = (1, 0)

food = (random.randint(0, GRID\_WIDTH - 1), random.randint(0, GRID\_HEIGHT - 1))

food\_value = random.choice(food\_weights)

food\_timer = time.time()

score = 0

speed = 10

game\_over = False

def update\_snake():

global food, food\_value, score, speed, game\_over, food\_timer

head = snake[0]

new\_head = (head[0] + direction[0], head[1] + direction[1])

# Wall collision

if new\_head[0] < 0 or new\_head[0] >= GRID\_WIDTH or new\_head[1] < 0 or new\_head[1] >= GRID\_HEIGHT:

crash\_sound.play()

game\_over = True

return

# Self collision

if new\_head in snake:

crash\_sound.play()

game\_over = True

return

# Move

snake.insert(0, new\_head)

# Eat food

if new\_head == food:

eat\_sound.play()

score += food\_value

food = (random.randint(0, GRID\_WIDTH - 1), random.randint(0, GRID\_HEIGHT - 1))

food\_value = random.choice(food\_weights)

food\_timer = time.time()

if score % 5 == 0:

speed += 2

else:

snake.pop()

# Food timeout

if time.time() - food\_timer > FOOD\_LIFETIME:

food = (random.randint(0, GRID\_WIDTH - 1), random.randint(0, GRID\_HEIGHT - 1))

food\_value = random.choice(food\_weights)

food\_timer = time.time()

# Game loop

while True:

clock.tick(speed)

if not game\_over:

for event in pygame.event.get():

if event.type == pygame.QUIT:

pygame.quit()

sys.exit()

elif event.type == pygame.KEYDOWN:

if event.key == pygame.K\_UP and direction != (0, 1):

direction = (0, -1)

elif event.key == pygame.K\_DOWN and direction != (0, -1):

direction = (0, 1)

elif event.key == pygame.K\_LEFT and direction != (1, 0):

direction = (-1, 0)

elif event.key == pygame.K\_RIGHT and direction != (-1, 0):

direction = (1, 0)

update\_snake()

draw\_grid()

else:

# Game over screen

win.fill(BLACK)

over\_text = font.render("Game Over! Press R to Restart", True, YELLOW)

score\_text = font.render(f"Final Score: {score}", True, WHITE)

win.blit(over\_text, (WIDTH // 2 - over\_text.get\_width() // 2, HEIGHT // 2 - 20))

win.blit(score\_text, (WIDTH // 2 - score\_text.get\_width() // 2, HEIGHT // 2 + 20))

pygame.display.update()

# Wait for restart

for event in pygame.event.get():

if event.type == pygame.QUIT:

pygame.quit()

sys.exit()

if event.type == pygame.KEYDOWN and event.key == pygame.K\_r:

reset\_game()