import tkinter  
import random  
  
ROWS = 25  
COLS = 25  
TILE\_SIZE = 25  
  
WINDOW\_WIDTH = TILE\_SIZE \* COLS  
WINDOW\_HEIGHT = TILE\_SIZE \* ROWS  
  
class Tile:  
 def \_\_init\_\_(self, x, y):  
 self.x = x  
 self.y = y  
  
# GAME WINDOW  
window = tkinter.Tk()  
window.title("Snake Game")  
window.resizable(False, False)  
  
canvas = tkinter.Canvas(window, bg="black", width=WINDOW\_WIDTH, height=WINDOW\_HEIGHT, borderwidth=0, highlightthickness=0)  
canvas.pack()  
window.update()  
  
# center window  
screen\_width = window.winfo\_screenwidth()  
screen\_height = window.winfo\_screenheight()  
window\_x = int((screen\_width / 2) - (WINDOW\_WIDTH / 2))  
window\_y = int((screen\_height / 2) - (WINDOW\_HEIGHT / 2))  
window.geometry(f"{WINDOW\_WIDTH}x{WINDOW\_HEIGHT}+{window\_x}+{window\_y}")  
  
# game state  
snake = Tile(5 \* TILE\_SIZE, 5 \* TILE\_SIZE)  
snake\_body = []  
food = Tile(random.randint(0, COLS - 1) \* TILE\_SIZE, random.randint(0, ROWS - 1) \* TILE\_SIZE)  
velocityX = 1  
velocityY = 0  
game\_over = False  
score = 0  
  
def change\_direction(e):  
 global velocityX, velocityY  
 if game\_over:  
 return  
 if e.keysym == "Up" and velocityY != 1:  
 velocityX = 0  
 velocityY = -1  
 elif e.keysym == "Down" and velocityY != -1:  
 velocityX = 0  
 velocityY = 1  
 elif e.keysym == "Left" and velocityX != 1:  
 velocityX = -1  
 velocityY = 0  
 elif e.keysym == "Right" and velocityX != -1:  
 velocityX = 1  
 velocityY = 0  
  
def move():  
 global snake, food, snake\_body, game\_over, score  
  
 if game\_over:  
 return  
  
 # Add current head to body before moving  
 if velocityX != 0 or velocityY != 0:  
 snake\_body.insert(0, Tile(snake.x, snake.y))  
  
 # Move snake head  
 snake.x += velocityX \* TILE\_SIZE  
 snake.y += velocityY \* TILE\_SIZE  
  
 # Check collision with walls  
 if snake.x < 0 or snake.x >= WINDOW\_WIDTH or snake.y < 0 or snake.y >= WINDOW\_HEIGHT:  
 game\_over = True  
 return  
  
 # Check collision with self  
 for tile in snake\_body:  
 if snake.x == tile.x and snake.y == tile.y:  
 game\_over = True  
 return  
  
 # Check food collision  
 if snake.x == food.x and snake.y == food.y:  
 score += 1  
 food.x = random.randint(0, COLS - 1) \* TILE\_SIZE  
 food.y = random.randint(0, ROWS - 1) \* TILE\_SIZE  
 # Don't remove tail → snake grows  
 else:  
 if snake\_body:  
 snake\_body.pop() # Remove last tail piece to keep length constant  
  
def draw():  
 move()  
 canvas.delete("all")  
  
 # Draw food  
 canvas.create\_rectangle(food.x, food.y, food.x + TILE\_SIZE, food.y + TILE\_SIZE, fill="red")  
  
 # Draw snake head  
 canvas.create\_rectangle(snake.x, snake.y, snake.x + TILE\_SIZE, snake.y + TILE\_SIZE, fill="lime green")  
  
 # Draw snake body  
 for tile in snake\_body:  
 canvas.create\_rectangle(tile.x, tile.y, tile.x + TILE\_SIZE, tile.y + TILE\_SIZE, fill="green")  
  
 # Draw score or game over  
 if game\_over:  
 canvas.create\_text(WINDOW\_WIDTH / 2, WINDOW\_HEIGHT / 2, font="Arial 20", text=f"Game Over! Score: {score}", fill="white")  
 else:  
 canvas.create\_text(50, 10, font="Arial 10", text=f"Score: {score}", fill="white")  
  
 window.after(100, draw)  
  
# Start the game  
draw()  
window.bind("<KeyRelease>", change\_direction)  
window.mainloop()