import turtle

import time

import random

delay = 0.1

score = 0

highScore = 0

segments = []

# Set up the screen

wn = turtle.Screen()

wn.setup(width=600, height=600)

wn.bgcolor("blue")

wn.tracer(0) # So that screen doesn't update automatically

# Snake head

head = turtle.Turtle()

head.speed(0)

head.shape("square")

head.color("black")

head.penup()

head.goto(0,0)

head.direction = "stop"

# Snake food

food = turtle.Turtle()

food.speed(0)

food.shape("circle")

food.color("red")

food.penup()

food.goto(0,100)

# score text

scoreTxt = turtle.Turtle()

scoreTxt.speed(0)

scoreTxt.shape("square")

scoreTxt.color("white")

scoreTxt.penup()

scoreTxt.hideturtle()

scoreTxt.goto(0, 260)

scoreTxt.write("Score: 0 High Score: 0", align="center", font=("Courier", 24, "normal"))

# Direction Functions

def go\_up():

if head.direction != "down":

head.direction = "up"

def go\_down():

if head.direction != "up":

head.direction = "down"

def go\_left():

if head.direction != "right":

head.direction = "left"

def go\_right():

if head.direction != "left":

head.direction = "right"

# move Function

def move():

if head.direction == "up":

y = head.ycor()

head.sety(y + 20)

if head.direction == "down":

y = head.ycor()

head.sety(y - 20)

if head.direction == "left":

x = head.xcor()

head.setx(x - 20)

if head.direction == "right":

x = head.xcor()

head.setx(x + 20)

def restart():

time.sleep(1)

head.goto(0,0)

head.direction = "stop"

for segment in segments:

segment.goto(1000,1000)

segments[:] = []

score = 0

delay = 0.1

scoreTxt.clear()

scoreTxt.write("Score: {} High Score: {}".format(score, highScore), align="center", font=("Courier", 24, "normal"))

# Keyboard controls

wn.listen()

wn.onkey(go\_up, "Up")

wn.onkey(go\_down, "Down")

wn.onkey(go\_left, "Left")

wn.onkey(go\_right, "Right")

# Main game loop

while True:

wn.update()

#Detect collision with borders

if head.xcor() < -290 or head.xcor() > 290 or head.ycor() > 290 or head.ycor() < -290:

restart()

#Detect collision with Food

if head.distance(food) < 20:

# relocate food

x = random.randint(-290, 290)

y = random.randint(-290, 290)

food.goto(x,y)

#add new segment

new\_segment = turtle.Turtle()

new\_segment.speed(0)

new\_segment.shape("square")

new\_segment.color("grey")

new\_segment.penup()

segments.append(new\_segment)

delay -= 0.002

score += 10

# update highscore

if score > highScore:

highScore = score

scoreTxt.clear()

scoreTxt.write("Score: {} High Score: {}".format(score, highScore), align="center", font=("Courier", 24, "normal"))

# move segments

for index in range(len(segments)-1, 0, -1):

x = segments[index-1].xcor()

y = segments[index-1].ycor()

segments[index].goto(x,y)

# move segment 0 to where the head is

if len(segments) > 0:

x = head.xcor()

y = head.ycor()

segments[0].goto(x,y)

move()

# detect collision with body segments

for segment in segments:

if segment.distance(head) < 20:

restart()

time.sleep(delay)