Explanation of snake game

import turtle

import time

import random

* **turtle**: Graphics module to create visuals (snake, food, etc.).
* **time**: To control timing (e.g., delay after collisions).
* **random**: To generate random coordinates for food.

python

CopyEdit

delay = 0.1

* Sets the **initial delay** between game loop updates (controls speed of the snake).

**7–9: Score setup**

python

CopyEdit

score = 0

high\_score = 0

* Tracks the **current score** and **high score**.

**11–15: Set up the screen**

python

CopyEdit

wn = turtle.Screen()

wn.title("Snake Game by @TokyoEdTech")

wn.bgcolor("green")

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

wn.tracer(0)

* Creates the game **window** with a title and green background.
* .setup(...): Sets window size to 600×600 pixels.
* .tracer(0): Turns off automatic screen updates (we’ll manually update it).

**17–23: Create the snake head**

python

CopyEdit

head = turtle.Turtle()

head.speed(0)

head.shape("square")

head.color("black")

head.penup()

head.goto(0,0)

head.direction = "stop"

* **head** is the main snake.
* .penup(): Prevents drawing lines.
* .goto(0,0): Starts in the center.
* .direction: Custom attribute to track movement direction.

**25–30: Create the food**

python

CopyEdit

food = turtle.Turtle()

food.speed(0)

food.shape("circle")

food.color("red")

food.penup()

food.goto(0,100)

* The **food** appears as a red circle at (0,100).

**32: Segment list**

python

CopyEdit

segments = []

* This will store the **snake body parts** as the snake grows.

**34–40: Create the pen (score display)**

python

CopyEdit

pen = turtle.Turtle()

pen.speed(0)

pen.shape("square")

pen.color("white")

pen.penup()

pen.hideturtle()

pen.goto(0, 260)

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

* A hidden turtle that **writes the score** at the top of the screen.

**42–55: Movement functions**

python

CopyEdit

def go\_up():

if head.direction != "down":

head.direction = "up"

* Prevents reversing directly into itself. Similar logic for:

python

CopyEdit

def go\_down(): ...

def go\_left(): ...

def go\_right(): ...

**57–69: Move function**

python

CopyEdit

def move():

if head.direction == "up":

y = head.ycor()

head.sety(y + 20)

* Moves the snake **20 pixels** in the current direction.
* .xcor() and .ycor() get the current position.
* .setx() and .sety() move it.

**71–75: Keyboard bindings**

python

CopyEdit

wn.listen()

wn.onkeypress(go\_up, "w")

...

* Binds keys to movement:
  + w = up
  + s = down
  + a = left
  + d = right

**77: Start of game loop**

python

CopyEdit

while True:

* Runs the game continuously until the user closes it.

**78: Update screen**

python

CopyEdit

wn.update()

* Manually refreshes the screen.

**80–96: Border collision detection**

python

CopyEdit

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

* If the snake hits the **border**, it resets:
  + Pauses for 1 second.
  + Snake moves to center.
  + Snake stops.
  + Clears all segments.
  + Score resets.
  + Score display updates.

**99–115: Collision with food**

python

CopyEdit

if head.distance(food) < 20:

* Checks if the snake is **close enough** to eat the food:
  + Food moves to a random location.
  + A new body segment is added.
  + Snake speeds up slightly.
  + Score increases.
  + If it's a new high score, update it.
  + Update score display.

**117–122: Move snake body segments**

python

CopyEdit

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

* **Moves each segment** to the position of the one in front of it.
  + Done in reverse to prevent overlapping.

**124–127: Move first segment to head**

python

CopyEdit

if len(segments) > 0:

x = head.xcor()

y = head.ycor()

segments[0].goto(x,y)

* Moves the first body segment to where the head was.

**129: Move the head**

python

CopyEdit

move()

* Moves the snake’s head based on the current direction.

**131–147: Collision with self**

python

CopyEdit

for segment in segments:

if segment.distance(head) < 20:

* Checks if the head **hits any body segment**:
  + Reset snake.
  + Hide all segments.
  + Reset score and delay.
  + Update score display.

**149: Sleep between loop cycles**

python

CopyEdit

time.sleep(delay)

* Slows down the loop using current delay.

**151: Keep window open (never reached)**

python

CopyEdit

wn.mainloop()

* Not needed here because while True loop runs forever.
* This would keep the window open if the loop exited.