Snake Game

Design and projected by:

Muhammad Asghar Ali (44000)
Fahad Ali Akbar (44314)
Dilawer Khalid (49085)
Umar Azeem (51984)

Subject: DAA
Submited to: Mr. Usman Sharif

The game loop runs continuously and is responsible for updating the game state. It checks for collisions, updates positions, handles input, and manages the score. Here's a closer look at each operation within the loop and its time complexity:

1. **Screen Update**:
   - wn.update() is called at the start of each loop iteration. This function updates the graphics on the screen and is generally considered O(1) because it does not depend on the number of segments or other game elements.
2. **Collision with Borders**:
   - The conditions:

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

     This check involves four comparisons to see if the snake's head has gone beyond the screen boundaries. Each comparison takes constant time, so this entire operation is O(1).

3. **Collision with Food**:
   - The condition:

     python
     Copy code
     if head.distance(food) < 20:

     Here, the distance() function calculates the distance between the snake's head and the food. This involves some arithmetic operations (subtraction, squaring, and taking a square root) which all run in constant time. Thus, this check is also O(1).

4. **Collision with Body Segments**:
   - The most complex part of the loop is checking if the head collides with any of the body segments:

     python
     Copy code
     for segment in segments:
       if segment.distance(head) < 20:

- Here, the loop iterates over the segments list. If n is the number of segments, this loop will run n times. Each iteration involves a constant-time distance check, leading to a total complexity of O(n)O(n)O(n) for this operation.

5. **Movement of the Snake**:
   - The movement of the snake's segments also involves a loop:

     ```
     python
     Copy code
     for index in range(len(segments) - 1, 0, -1):
         x = segments[index - 1].xcor()
         y = segments[index - 1].ycor()
         segments[index].goto(x, y)
     ```

   - This loop again iterates over the segments, moving each segment to the position of the segment in front of it. This operation runs in O(n)O(n)O(n) time since it processes all n segments.
6. **Updating Segment 0 Position**:
   - After moving the segments, the position of segment 0 is updated to match the head:

     ```
     python
     Copy code
     if len(segments) > 0:
         x = head.xcor()
         y = head.ycor()
         segments[0].goto(x, y)
     ```

   - This operation checks the length of segments (constant time, O(1)O(1)O(1)) and updates segment 0's position (also O(1)O(1)O(1)). So, this is still O(1)O(1)O(1).
7. **Resetting After Collision**:
   - If a collision is detected, resetting the snake's position and clearing segments involves:

     ```
     python
     Copy code
     head.goto(0, 0)
     head.direction = "stop"
     for segment in segments:
         segment.goto(1000, 1000)
     segments.clear()
     ```

   - Setting the head position and stopping direction are O(1)O(1)O(1) operations. The loop that moves each segment off-screen is O(n)O(n)O(n), and clearing the list is O(1)O(1)O(1). Therefore, resetting after a collision takes O(n)O(n)O(n) in the worst case.

Total Time Complexity for One Iteration of the Game Loop

Now, let's summarize the time complexities of each component within a single iteration of the game loop:

- Screen update: O(1)
- Border collision check: O(1)
- Food collision check: O(1)
- Body collision check: O(n)
- Moving segments: O(n)
- Updating segment 0 position: O(1)
- Resetting after collision: O(n) (in the worst case)

Combining all these operations, the total time complexity for one iteration of the game loop is:

$O(1)+O(1)+O(1)+O(n)+O(n)+O(1)+O(n)=O(n)$

Thus, we conclude that the overall time complexity for each iteration of the main game loop is O(n)O(n)O(n), where n is the number of segments in the snake. This is primarily due to the body collision check and the movement of the segments. As the snake grows longer (more segments), the time taken for these operations increases linearly.