Name ~ ARNAB KUMAR ROY
Email ID~ arnabroy10121998@gmail.com

Project 01: Snake Game

Description

The Snake game is a classic arcade game that has been around since the earliest days of home computing. It has re-emerged in recent years on mobile phones. It was most popular in Nokia Phones in late 90s and early 2000s.

Objectives

The objective of the game is simple: control a snake to eat food (represented by an apple) and grow longer. However, the game becomes challenging as the snake grows and must avoid colliding with its own body or the game boundaries.

Technologies Used

- **Java**: The programming language used for game development.
- **VSCode**: The integrated development environment (IDE) for coding and executing the game.

Method of Working

The Snake game project includes the following features:

1. **Snake Movement**: The player controls the snake using arrow keys (up, down, left, right). The snake moves continuously in the direction specified by the player.
2. **Food Generation**: A Food appears randomly on the game board. The snake must eat the apple to grow longer.
3. **Collision Detection**:
    - If the snake collides with the game boundaries (edges of the screen), the game ends.
    - If the snake collides with its own body, the game ends.
4. **Scoring and Time Calculation**:
    - The player's score increases each time the snake eats an apple.
    - The game keeps track of the time played.

Challenges Faced

1. The implementations of java.awt and java.swing packages inside a nascent java code was tricky and difficult.
2. Implementations of Collision physics in 2D was new for me.
3. Working with GUI interfaces with active keyboard usage and user participations was a bit difficult for me.

Solutions Implemented

1. When designing the Snake game, focus on creating well-organized classes and their interactions.
2. Implement snake movement based on user input (arrow keys).
3. Detect collisions:
    i. If the snake hits the game boundaries, end the game.
    ii. If the snake collides with itself, end the game.
4. Keep track of the player's score (number of apples eaten).

Learnings:

1. Understanding the principles of OOD, including classes, objects, inheritance, and encapsulation.
2. Learning to create well-organized class structures for the Snake game.
3. Learning how to manage interactions between different components of the game.
4. Study how the Snake game works:
    - Snake movement based on user input (arrow keys).
    - Food generation (usually represented by an apple).
    - Collision detection (snake hitting boundaries or itself).
5. Implement these core mechanics step by step.
6. Exploring Java Swing and AWT libraries for creating a user-friendly GUI.
7. Learn about components like panels, buttons, labels, and event listeners.

Project Update

While the current implementation covers the basic functionality of the Snake game, there are several ways to enhance it:

1. Graphics and Animation: Improve the visual appeal by adding graphics, animations, and sound effects.
2. Levels: Implement different levels of difficulty, where the snake moves faster or additional obstacles are introduced.
3. High Scores: Store and display high scores achieved by players.
4. Multiplayer Mode: Add a multiplayer mode where two players can compete simultaneously.