**Name or Email Id:** saisujayvs@gmail.com

**Task Title:** Snake Game Development

**Task Description:** Develop a classic Snake game using Java.

**Steps Taken:**

1. **Set up the game board:** Created a grid representing the game board using a 2D boolean array. This array tracks whether each cell on the grid is occupied by the snake or not.
2. **Created the Snake class:** Defined a class to represent the snake with properties such as length, direction, and body segments. The Snake class includes methods to move the snake, check for collisions with walls and itself, and grow the snake when it eats food.
3. **Created the Food class:** Defined a class to represent the food. The Food class ensures that food appears at random positions on the game board where the snake is not present.
4. **Implemented game logic:** Developed the main game loop that controls the flow of the game. This loop handles user input, updates the positions of the snake and food, checks for collisions, and manages the game state.
5. **Handled user input:** Allowed the player to control the direction of the snake using arrow keys. KeyListener interface is used to capture user input and change the direction of the snake accordingly.
6. **Displayed the game:** Utilized Java Swing to create a graphical user interface for the game. The SnakeGame class extends JPanel and overrides the paintComponent method to draw the game board, snake, and food on the screen. The repaint method is called to update the graphics during each iteration of the game loop.
7. **Handled collisions:** Implemented collision detection to check for collisions between the snake, food, and game boundaries. If the snake collides with itself or the game boundaries, the game ends.
8. **Added scoring:** Maintained the player's score based on the length of the snake and the number of food items eaten. The score is incremented each time the snake eats food.
9. **Implemented game over:** Displayed a game over message when the game ends, showing the player's score. The player can choose to restart the game or exit.

**Challenges Faced:**

* Implementing smooth movement for the snake in all four directions without causing glitches or delays.
* Ensuring that food spawns at different positions each time and does not overlap with the snake's body.

**Solutions Implemented:**

* Implemented a robust game logic to handle snake movement smoothly and prevent collisions with itself or the boundaries.
* Utilized random number generation to spawn food at random positions on the game board, ensuring it does not overlap with the snake.

**Learnings:**

* Gained a deeper understanding of game development principles, including game loops, collision detection, and user input handling.
* Enhanced proficiency in using Java Swing for creating graphical user interfaces and rendering game graphics.

**Project Update:**  
The Snake game development is now complete, offering players a classic gaming experience. Players can control the snake, eat food to grow longer, and track their score as they progress. The game ends if the snake collides with itself or the boundaries, allowing players to restart and continue playing if desired.