**N-Queens Visualizer Project Report**

**Overview**

The N-Queens Visualizer is a Java application that visualizes the process of solving the N-Queens problem. The N-Queens problem is a classic combinatorial problem where the goal is to place N chess queens on an N×N chessboard such that no two queens threaten each other. This means no two queens can share the same row, column, or diagonal.

**Project Components**

**1. NQueensVisualizer Class**

This class extends JPanel and serves as the main visual component of the application. It handles the board's graphical representation and the logic for placing the queens.

**Key Variables**

* size: The size of the board (N×N).
* board: A 2D array representing the board, where 1 indicates the presence of a queen, and 0 indicates an empty cell.
* CELL\_SIZE: The size of each cell in pixels.
* PADDING: The padding around the board to center it within the window.
* SLEEP\_TIME: The delay in milliseconds between moves to visualize the process.

**2. Constructor**

A black background with colorful text

Description automatically generated

The constructor initializes the size of the board and the 2D array.

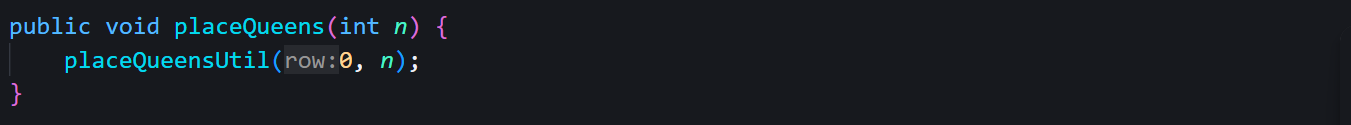
3. **paintComponent Method**

A computer screen shot of code

Description automatically generated

This method is overridden to draw the board. It colors the cells in a checkerboard pattern and places black circles to represent the queens.

**4. placeQueens Method**

This method initiates the recursive process to place the queens. It starts a new thread to avoid blocking the main GUI thread.

5. **placeQueensUtil Method**

A screen shot of a computer code

Description automatically generated

This recursive method attempts to place queens row by row. It checks if placing a queen in a particular column is safe and continues recursively. If placing a queen leads to a solution, it returns true; otherwise, it backtracks and tries the next column.

6. **isSafe Method**

A computer code with many colorful text

Description automatically generated with medium confidence

This method checks if placing a queen at board[row][col] is safe by ensuring no other queens threaten this position.

7. **Main Method**

A screen shot of a computer code

Description automatically generated

The main method sets up the GUI. It prompts the user for the board size and number of queens, creates a JFrame to hold the NQueensVisualizer panel, and starts the process of placing the queens.

**Execution Flow**

1. **Initialization**: The user inputs the board size and the number of queens. The application initializes the board and GUI components.
2. **Placing Queens**: The placeQueens method starts the recursive process in a new thread.
3. **Recursive Backtracking**: The placeQueensUtil method attempts to place queens row by row, using the isSafe method to check for valid positions. If a conflict arises, it backtracks and tries a different column.
4. **Visualization**: The paintComponent method is called to update the GUI after each move, with a delay to visualize the process.

**Conclusion**

The N-Queens Visualizer provides an interactive way to understand and visualize the backtracking algorithm used to solve the N-Queens problem. By integrating threading and proper UI updates, it ensures a smooth and informative user experience