Introduction to Computer Science using JavaII

Final Project Instructions

N-Queens Solution Algoritm.

Antonio Recalde

1) Description

The “8-Queens” game is a chess variant in which the final objective of the player is to place 8 Chess Queens on the board in such a way that no Queens attack each other.
For this to happen, no Queen may occupy the same row, column or diagonal with another Queen.
A standard chess board has 8 rows and 8 columns, totaling 64 squares. In this board configuration, there are 92 different possible solutions.

This program finds all such solutions, calculates the time it took to complete this task, and provides each solution to the user via a GUI.

2) Execution.

This program can be executed in two different ways:

a) In the command prompt, through “Board.java.”
b) Using Javax.swing, through “NQueens.java.”

In the event that the program were to be executed via “Board.java,” the text spanning from line 90 through line 105 should be uncommented in order to receive some information feedback from the program.

Granted a the GUI implementation were to be executed, the user must simply compile and run the file “NQueens.java.”

The GUI implementation is the official final product produced for the purpose of satisfying the Final Project requirements.

3) Features:

This program features:

- A GUI presentation of all solutions. In this GUI, the user may interact with the program via three buttons.
    a) A “Start” button, which starts the algorithm analysis.
    b) A “Previous” button, which displays the previous solution in the solutions array.

c) A “Next” button, which displays the next available solution.

- A command prompt execution, that displays features not present in the GUI. E.g., The current values of the attacked[] array, and tried[] array.

4) Expansion:

This program’s most natural expansion would be to enable a board in which the User may try to play/solve the 8-Queens game on his own. Additionally, the GUI may acquire more buttons that perform functions such as:

- Show attacked squares.
- Enable the use of other pieces instead of Queens.
- Change the size of the board.
- Change the colors of the board.
- Enable a timer to measure how long the user is taking to solve the problem.

5) Reactions:

This Final Project made me realize that having good ideas is just a valuable as being able to put those ideas into code. It is far easier to simply do the assignments, where you are restricted to solving programming problems in a certain way. However, as you are given more freedom about what to do, the task automatically becomes a more difficult, as now you must not only use logic, but also creativity.

One of the things that were the most difficult for me was measuring the complexity of the task. However, once completed I realized that I am a lot (or a little) better at this, than I thought I was.

I consider that this course achieved its objectives. It was the perfect introductory course for me.