

COMP2911 PROJECT

June 2, 2013

Part I

Who is this aimed at?

This program is named for all users (novice, intermediate, expert). It is designed to have ratings which will cater for all users.

The highest difficulty caters for the most experienced sudoku players. The hardest difficulty requires techniques such as X-Wing, Nishio and guessing for it to be solved.

Part II

What sort of features to the application?

In the application, all users are able to play a game of sudoku. The backend will generate a random sudoku puzzle for the user, and the user will be able to interact with the system to place in numbers and remove numbers (only those which they have placed). Our application prevents the user from place a number which exists in the same row, column and square, and when selecting a cell, the row, column and square corresponding to that cell will be highlighted as well.

All users will be able to save and load the game and continue playing where they left off.

The user also can also see how long he has been playing a particular sudoku for. When a user saves the current game, it will also save the time. When the game is loaded again, the timer will continue where it left off.

The user can also request hints, and also see the full solution. Hints are based off the next state in the search we run to solve the sudoku, as there will be a high probability that the search will yield a helpful hint.

There is also a reset functions which will reset the board.

Part III

What sort of features are needed by different categories of users: how can the interface handle seemingly different requirements?

Sudoku is fairly universal, and the only real feature that a user needs is one that enables them to learn the game. Because of this, we have implemented 2 difficulty ratings which will tell you if the number you have inputted in a cell is correct. This is to ensure that one can learn the game at a fast pace (using just the basic techniques). Once one becomes avid at the game, they can try harder difficulties which will remove this feature, and users will be able to play the game with little aid.

Part IV

What sort of helps or hints should the system be able to provide users?

Like above, there are various mechanisms to help the user.

We chose to remove the ability to place invalid moves (moves which violate the game rules) for many reasons. Most sudoku mediums are played on pen and paper, and allow this sort of behavior. We wanted to remove this because this is a fairly trivial matter, and we see this to be a “mistake” rather than a result of a technique. Because of this, we our users not to become frustrating by such a simple mistake, and practice less obvious techniques to improve their general game playing experience.

The highlighting of squares is a similar issue to above. It is simply to remove frustrating from the user.

We also want the user to be able to solve every sudoku puzzle even with assistance. If a user cannot finish a puzzle, it would seem silly for them to just quit and end of a frustrated note. Our hint and solve features remedy this problem.

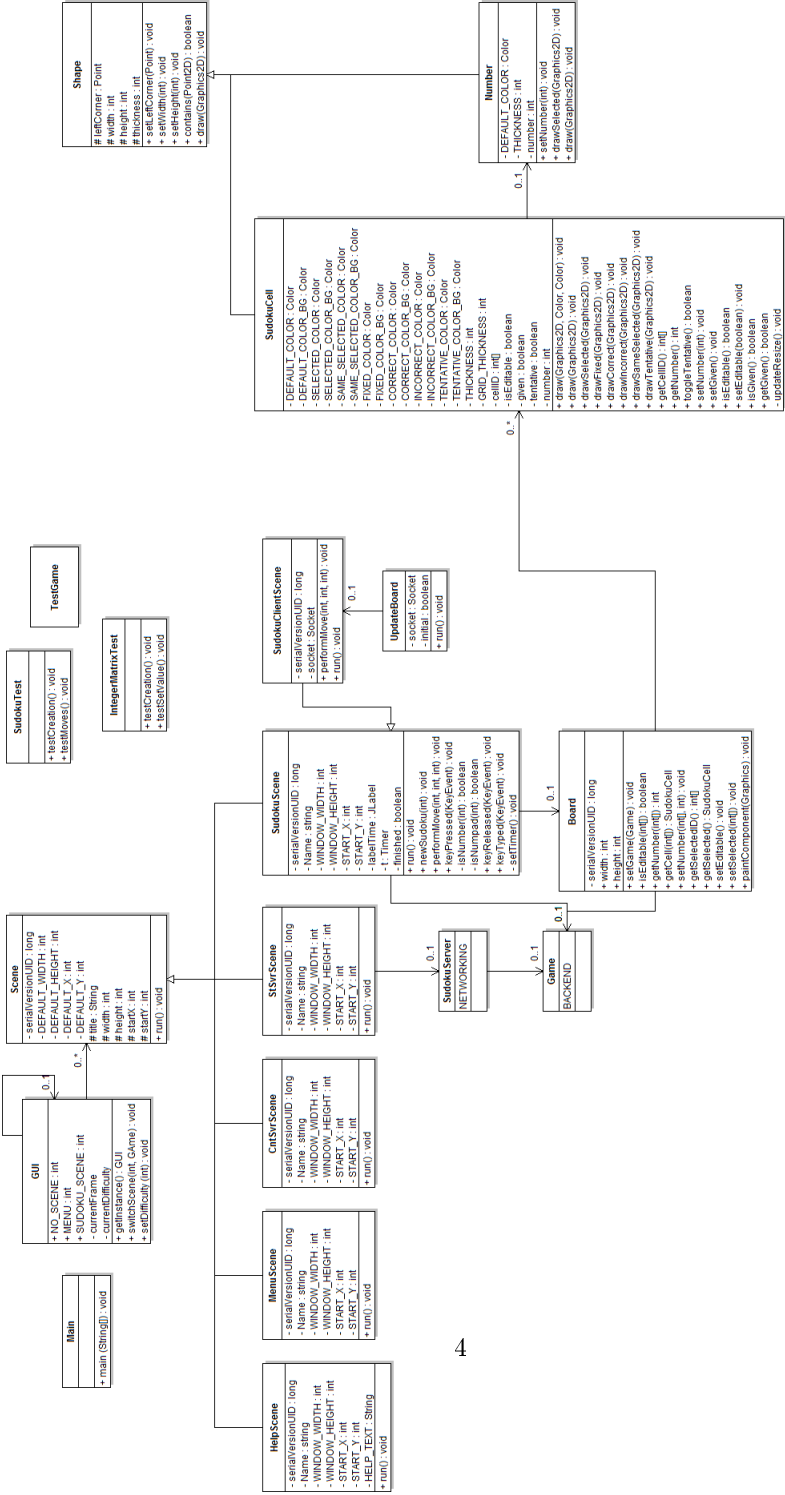
Part V

What sort of platforms with what form factors is the system designed to run on?

Since this is coded in Java, any platform with a JVM will be able to run the application. Ideally this was designed for Linux, MacOS and Windows, but can also be extended to further platforms. Our main scene (the area where one can play a sudoku) has a resize feature that resizes the board based on the dimensions of the window.

One can also play with the mouse or the keyboard (numbers 0-9/keypad/arrow keys). This is designed such that people with a touch screen can also play the sudoku, and it is not limited to a keyboard.

FRONTEND



BACKEND



NETWORKING

