

## Team 4 Project Outline: Sudoku Solving Algorithm

### Project Description:

This application is capable of solving sudoku puzzles through the use of back-tracking. To use the program you can type in the name of the sample txt file that is provided, or your own sudoku txt file. To use your own txt file just read the sudoku puzzle from left in right typing the numbers that are there and typing dots for empty spaces. (ie 5....1.78...523.9. is two rows of a puzzle).

After putting in a valid file the program will ask for how often an iteration of the puzzle will be displayed. So the higher the number the quicker the puzzle will solve, but the less test numbers will be shown in the GUI. After the number of iterations have been chosen the program will run and display the test numbers that are being replaced until the correct answer has been found.

After the correct solution has been found the program will stop and display the correct solution until the user closes the application.

### What we learned:

- We learned a lot about back-tracking algorithms and how they can be used in programs.
- We expanded what we knew about GUI input and output.
- Further cemented our understanding of multiple class object oriented programming.

### Problems we encountered:

- We had no background knowledge on back-tracking so we had to research the best way to go about solving that issue.

- It was difficult finding the best way to take the item objects and grouping them into columns, rows, and especially difficult to group them into boxes, so we spent a large portion of time on those algorithms.
- We had limited experience working with GUI's so we had to find out the best way to go about tasks such as changing font sizes, and size of the window in the program.
- We were able to achieve pretty much all of the original goals of the project, but we were able to expand on some of our ideas such as being able to show the board as its being solved.
- We were able to finish the project effectively, and were able to learn along the way, so in retrospect we do not have anything that we would change.

### Breakdown of Tasks:

Logan Newsom: Logan came up with the algorithm that we used to make the Box objects, and he wrote the documentation for the Box, Row, Column, and Item classes. Logan also helped come up with ideas for how to do the Sudoku class.

Jacob Williams: Jacob came up with lots of ideas to help create the item classes, and the overall layout of the project. He wrote a large portion of the project proposal as well. Was instrumental in helping solve many of the problems that came up when creating the program.

Parker Billinger: Parker helped brainstorm and assist on many of the classes in the project, and some of the algorithms used for the rows and columns. He also wrote the entirety of the project outline.

Trey Etzel: Trey was the overall team leader. He did a large portion of the writing when it came to code, and did lots of the brainstorming when it came to the project. He was also responsible for the backtracking algorithm, lots of the GUI code, and the general layout for most of the classes. Trey also helped contribute to and write parts of the project proposal.