**COMP1206 MathDoku Instructions**

**Guide:** This document will help us run and use your application during marking. Please complete the sections below. You may want to include screenshots if this helps explain the functionality. For most sections, 1-2 sentences are probably sufficient.

If you did not implement a particular part, please write “not implemented” in the relevant section.

These instructions are not assessed directly, but they will help ensure that we do not miss any important features of your application.

|  |
| --- |
| **Installing and Running the Application (Part 1)**  *Copy and paste the contents of your README.txt file below.* |
| MathDoku - Thomas Hoad (30914515)  Guide: 1) Make sure you have a system with a working version of Java and JavaFX in order to run the program. 2) Run command line from folder containing Java class files. 3) Run Main class (this will differ on the version of Java - see below.) 4) An application should appear that displays a GUI. To load a grid, you must load a game.  To run from command line (Java 8 and including JavaFX):  javac Main.java  java Main  To run from command line (Java 9+ or excluding JavaFX):  javac --module-path="directory of lib folder for javafx" --add-modules=ALL-MODULE-PATH Main.java  java --module-path="directory of lib folder for javafx" --add-modules=ALL-MODULE-PATH Main |
| **Starting a Game (Optional – Part 1)**  *If any additional steps are needed to start a game, briefly describe them here.* |
| Once you run the main class, the application should appear. To run the game, you must either select a text file game by pressing “Load from File” or you can enter in a game with raw text by pressing the “Load from Text” button. |
| **Cell Completion (Part 3)**  *Describe how to enter and clear cell values by keyboard and by mouse.* |
| By Keyboard:  The enter a number into a cell with a keyboard, click on a tile you wish to select, then press a number button on the keyboard.  By Mouse: To enter a number into a cell with the mouse, click on a tile you wish to select, then press a number button on the application below the grid. |
| **Can your application handle - and ÷ cages with more than two cells? (Part 4)** |
| Yes |
| **Mistake Detection (Part 4)**  *Describe how to enable mistake detection in your application.* |
| Along the top bar, there is a checkbox to enable checking. Whilst this is selected, the grid will be checked. An incorrect row, column and grid will be highlighted red, whilst a correct one is displayed normally. A selected tile will remain white with the thick red border. |
| **Win Detection / Animation (Parts 4 & 8)**  *Describe how the application notifies the player when the game is won (including any animations you have implemented for Part 8).* |
| The game will constantly check for a win in the background. When the player has won, they will receive an alert to say so. In addition to this, the application will flash multicoloured for a short period of time. |
| **Clearing (Part 5)**  *Describe how to clear the board.* |
| To clear the grid, simply press the “Clear Grid” button. |
| **Undo/Redo (Part 5)**  *Describe how to undo / redo actions.* |
| To undo the last move, simply press the “Undo” button.  To redo the last move, simply press the “Redo” button.  If an action cannot be undone or redone, then the irrelevant button will be greyed out. |
| **Loading Files (Part 6)**  *Describe how to load puzzles both from file and through text input. Also mention any limitations in what puzzles you can load (if any), e.g., up to a certain size if smaller than 8x8.* |
| From File:  Click the “Load from File” button and select your game file from the file chooser window.  From Text:  “Click the “Load from Text” button and enter the game text into the text area. |
| **Font Sizes (Part 7)**  *Describe how to change font sizes* |
| To change the font size, use the drop-down menu from the top bar, and select a font size. The default size will be medium. |
| **Solver (Part 9)**  *Describe how to solve a puzzle, how to get a hint and any limitations there might be (e.g., up to what size you can solve reliably and within <1 min). Also mention where we can find your code for solving the puzzle (which files and lines)?* |
| Solve puzzle:  The solver should work when you press the “Solve” button, however it is very basic.  Limitations (optional):  The solver can fill in single cage tiles, and has a very basic iterative method of reducing possible values. It can solve very small and some very easy games but it is not a reliable method currently. |
| **Random Game Generator (Part 10)**  *Describe how to generate a random game, including what options the player can select. Also specify where we can find your code for generating the puzzle (which files and lines)? Where in the code do you ensure there is only one solution (which file and lines)?* |
| This extension was not attempted. |
| **Additional Information (Optional)**  Any other information that may be useful for us to know. |
|  |