**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.* |
| In "coursework" folder there is the MathDoku.java file;  In order to compile and run the application from command line, the following 2 commands are needed:  \*\*you also need to know the path to the javafx sdk lib folder; in my case it is "D:\javafx-sdk-11.0.2\lib"  //For compilation  javac --module-path=D:\javafx-sdk-11.0.2\lib --add-modules=ALL-MODULE-PATH MathDoku.java  //For running the application  java --module-path=D:\javafx-sdk-11.0.2\lib --add-modules=ALL-MODULE-PATH MathDoku |
| **Starting a Game (Optional – Part 1)**  *If any additional steps are needed to start a game, briefly describe them here.* |
| A default grid of 6x6 will be displayed when the game starts. |
| **Cell Completion (Part 3)**  *Describe how to enter and clear cell values by keyboard and by mouse.* |
| By Keyboard:  Click on the desired cell and enter a number from the keyboard (you are allowed digits as great as the size of the board). To clear a cell click on the number displayed in the cell and type the backspace key.  By Mouse:  Click on the desired cell and then use the digit buttons on the top right to enter a value (you are allowed digits as great as the size of the board). To clear a cell click on the number displayed in the cell and then click the backspace button from the top right. |
| **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.* |
| Click on the toggle button “MISTAKES”. |
| **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).* |
| An alert of type information is shown when all the cells are correctly completed. An animation of type RotateTransition is being used. |
| **Clearing (Part 5)**  *Describe how to clear the board.* |
| Click on the button “CLEAR” and then “OK” on the Confirmation alert. |
| **Undo/Redo (Part 5)**  *Describe how to undo / redo actions.* |
| Undo is possible only for the cells which have been entered at least a value by clicking on the “UNDO” button, otherwise it is disabled.  Redo is possible only for the cells where UNDO has been called. Redo is actioned by clicking the “REDO” button. |
| **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 FILE” button and then select a text file with appropriate content from the computer.  From Text:  Click the “LOAD INPUT” button and then you are shown a dialog box with a TextArea where one could write the appropriate content and then click “OK”.  Limitations (optional): |
| **Font Sizes (Part 7)**  *Describe how to change font sizes* |
| Click on one of “SMALL”, “MEDIUM”, or “LARGE” buttons to apply the appropriate font size to the values and targets shown in cells. |
| **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)?* |
| NOT IMPLEMENTED |
| **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)?* |
| NOT IMPLEMENTED |
| **Additional Information (Optional)**  Any other information that may be useful for us to know. |
|  |