**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.* |
| Change the path to where the code is stored.  javac --module-path [PATH TO JAVAFX LIB FILE] --add-modules=ALL-MODULE-PATH MathDoku.java  java --module-path [PATH TO JAVAFX LIB FILE] --add-modules=ALL-MODULE-PATH MathDoku  Example: |
| **Starting a Game (Optional – Part 1)**  *If any additional steps are needed to start a game, briefly describe them here.* |
| Press “LOAD” on top left corner of the MATHDOKU window. A choice dialog box will open, prompting you to choose a method of loading the game. Go on the choice box, click on the desired method and press OK.  If you click on “Load from a file”, you will be asked to choose the file via filechooser.  If you click on “Load from text input”, a new window, having a text area where you can input how to load the game, will appear. Format for the text input loading is the same as for the text file method. A prompt is shown in the text area to give more details.  Press OK when done and the cages will appear. Press Cancel to go back to the original MATHDOKU window. |
| **Cell Completion (Part 3)**  *Describe how to enter and clear cell values by keyboard and by mouse.* |
| By Keyboard:  Click on the desired cell. Enter numbers using the number keys and delete using the backspace key.  By Mouse:  Click on the desired cell. A keypad is implemented on the right side of the MATHDOKU window. It contains numbers and the backspace button. Focus will not change. The keypad outputs only numbers that are relevant to the current grid size. |
| **Can your application handle - and ÷ cages with more than two cells? (Part 4)** |
| Yes / ~~No~~ (delete as appropriate) |
| **Mistake Detection (Part 4)**  *Describe how to enable mistake detection in your application.* |
| Click on “CHECK” button on the left side of the MATHDOKU window. The correct cages will be highlighted in green. The incorrect cages will be highlighted in red. Columns and rows that contain duplicates will be highlighted in red.  It also lights up in red for cells that are empty to make the user aware that they have not completed the grid. |
| **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).* |
| Win detection is automatic when the user finishes the grid, that is all cells are full. Winning animation consists of the cells in the grid to light up in a palette motion. |
| **Clearing (Part 5)**  *Describe how to clear the board.* |
| Press “CLEAR” button. All moves will be cleared, and the grid will be empty. |
| **Undo/Redo (Part 5)**  *Describe how to undo / redo actions.* |
| “UNDO” and “REDO” buttons on the top. They will be disabled according to the situation, that is you cannot undo at the start and redo unless you undo something. |
| **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: If you click on “Load from a file”, you will be asked to choose the file via FileChooser. Press OK and the desired grid and cages will appear.  From Text: If you click on “Load from text input”, a new window, having a text area where you can input how to load the game, will appear. Format for the text input loading is the same as for the Fext file method. A prompt is shown in the text area to give more details.  Limitations (optional): |
| **Font Sizes (Part 7)**  *Describe how to change font sizes* |
| Click on the ChoiceBox “Medium” which is the initial font size that the textfield and label are at. In the ChoiceBox, there is a small and large font size which will change the text size. |
| **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: Press “SOLVE” button and if the grid has a unique solution, it will be put in the cells at the correct positions.  Get hint: Click “HINT” button to get a brief part of the solution for 2.5s at a random available space in the textfields.  Limitations (optional): Grids 6x6 and above take too much time to solve. Can solve grids up to 5x5 in much less time.  Files / lines for solver: Lines 975 – 1240 |
| **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)?* |
| Generate puzzle (including options): Press “LOAD”. Select “Random Game Generator” from the ChoiceBox of methods. Select a grid size and difficulty from the next dialogbox and press “Generate”.  Files / lines for generator: Lines 324-424 & 1242 - 1967  File / lines to ensure there is only one solution: Implementation of solver ensures it |
| **Additional Information (Optional)**  Any other information that may be useful for us to know. |
|  |