

## 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.*

To compile and run the program from command line you have to have the Coursework.java and the cssstylesheet.css in the same folder and then enter the following commands where you have to change the module path according to the javafx folder

```
javac --module-path=/home/student/javafx-sdk-11.0.2/lib --add-modules=ALL-MODULE-PATH Coursework.java
```

```
java --module-path=/home/student/javafx-sdk-11.0.2/lib --add-modules=ALL-MODULE-PATH Coursework
```

When you run the application the game will start with the basic example.

I noticed that sometimes when you run it from the command line the “÷” sign is not well rendered because of the UTF-8 encoding and it is replaced with “Ä•”.

### Starting a Game (Optional – Part 1)

*If any additional steps are needed to start a game, briefly describe them here.*

### Cell Completion (Part 3)

*Describe how to enter and clear cell values by keyboard and by mouse.*

By Keyboard: To enter a value in a cell you have to click it and either use the keyboard to type the number or click on backspace to delete it

By Mouse: To enter a value in a cell you have to click it to select it and then use the mouse and press the button of the number you want to enter from the numpad on the right or click on the last button, the backspace to delete it

**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.*

To enable mistake detection in my application you have to click on the “Show Mistakes” button, the one coloured in purple gradient. After you clicked it you can see the mistakes as you enter the numbers. If the number is higher than it is permitted the number will be coloured RED. If there are the same 2 numbers on a column or row the whole column or row will be highlighted with RED border. If a cell is entirely completed but the result is not the one wanted, the entire numbers in that cell will be coloured in RED and the sign operation and result in the top corner will be coloured RED.

**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).*

When a player wins the game a message “YOU WON” will appear to notify the player that they won. The animation that I chose to implement is that the message “YOU WON” will change the colours repeatedly.

**Clearing (Part 5)**

*Describe how to clear the board.*

To clear the board you have to click the button “Clear Board” coloured in red gradient. An alert will appear where the player is informed that this action is irreversible. If they click confirm the board will be cleared, if they click cancel this action will be cancelled and they can continue playing

**Undo/Redo (Part 5)**

*Describe how to undo / redo actions.*

To Undo or Redo an action you have to click on the cell you want to Undo or Redo the input from. Every cell has its own Undo and Redo, this means that every cell will be able to be changed to the previous or the most recent change.

**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: To load a puzzle from a file you have to click the button "Load From File" coloured in light blue. After that the user will have a way of choosing the file they want to load the puzzle from and if the input is correct, then the game will load, otherwise the user will be notified with the problems of the file, this can include: File can be empty, File can have less cells than needed, File can have more cells than needed, The cells that are in the same cage have to be neighbours.

From Text: To load a puzzle from a file you have to click the button "Load From Text" coloured in blue. After that a text input will be shown to the player where they can write the puzzle and click on "Add Game" and if the input is correct, then the game will load, otherwise the user will be notified with the problems of the file, this can include: File can be empty, File can have less cells than needed, File can have more cells than needed, The cells that are in the same cage have to be neighbours.

Limitations (optional): There are no limitations, but I created the cell to look better for the sizes from 2x2 to 8x8

### **Font Sizes (Part 7)**

*Describe how to change font sizes*

To change the font size you have to click on one of the three buttons at the bottom of the game "Small", "Medium" or "Large". This will change the font used in the cells of the game