USER MANUAL

Installing and Running the Application

1. Install the NetBeans Integrated Development Environment and the latest version of the Java Development Kit (JDK).
2. Download the attached ZIP file and unzip it using file management software of your choice.
3. In the NetBeans IDE, click on File → Open Project, and then click on the unzipped folder.
4. Right click on the project in the project panel on the left, then click on “Run”. Alternatively, press F6 to run the project.

Reference Manual

- Accessing the Score Record
- Clearing Tiles
- Flagging Tiles
- Saving and Quitting

Accessing the Score Record

To access the score record, click on the settings icon in the top right corner, click on “Score Record”, and the score record will be opened in a new window, including (in order from fastest to slowest) a record of all scores, and the local date and time when the respective score was recorded.

Clearing Tiles

LEFT CLICK a tile to clear it. If the tile is covering a mine, the game ends.

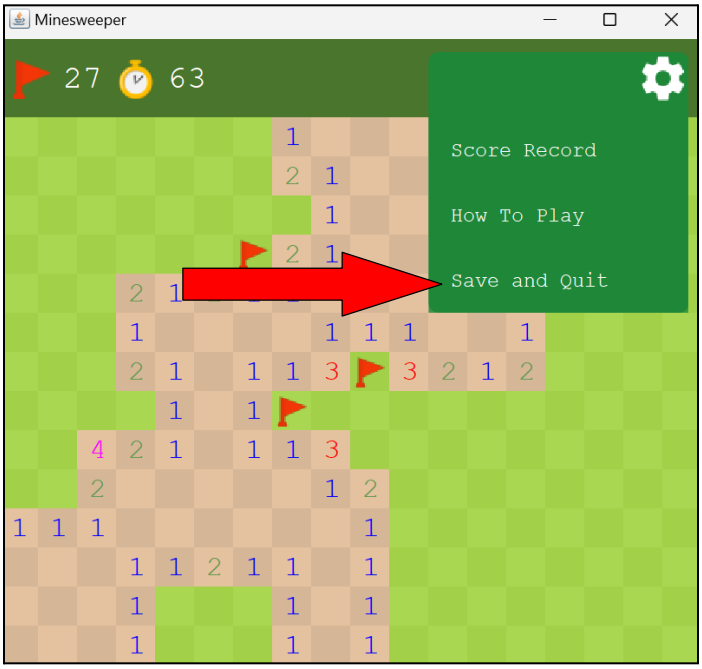
Flagging Tiles

RIGHT CLICK a tile to place a flag on it. RIGHT CLICK the tile again to remove the flag. You can place as many flags as there are in your inventory, which can be seen in the top left corner of the application window.

Saving and Quitting

Click on the settings icon in the top right corner.

Then, click on “Save and Quit”.

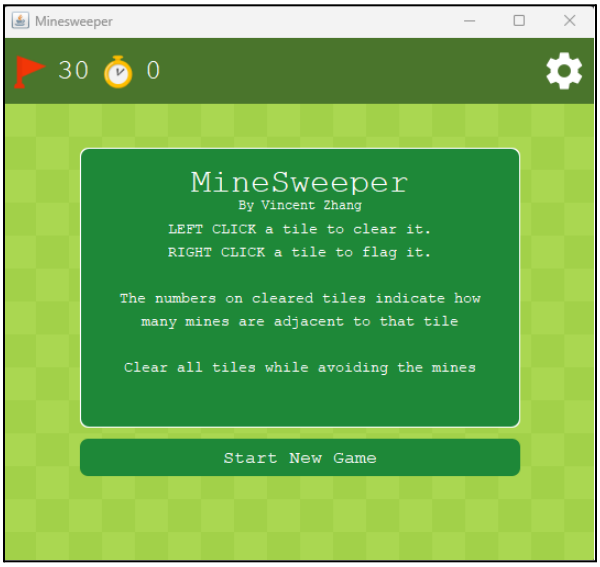


The application will close. When you reopen the application, you will receive a prompt to continue with the previous round. Click “Continue” to continue from the previous save file, or click “Start New Game” to start a new game. (NOTE: clicking “Start New Game” will delete the previous save file).

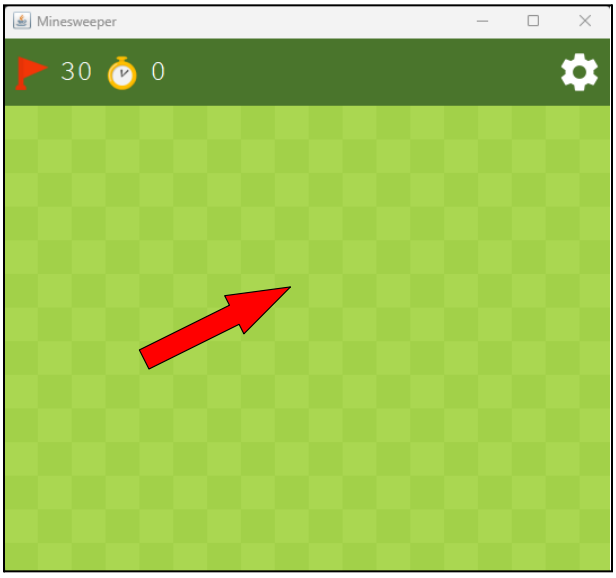
If you close the application using the “X” icon in the upper right corner, the current round will not be saved and you will have to start a new round.

Playing the Game

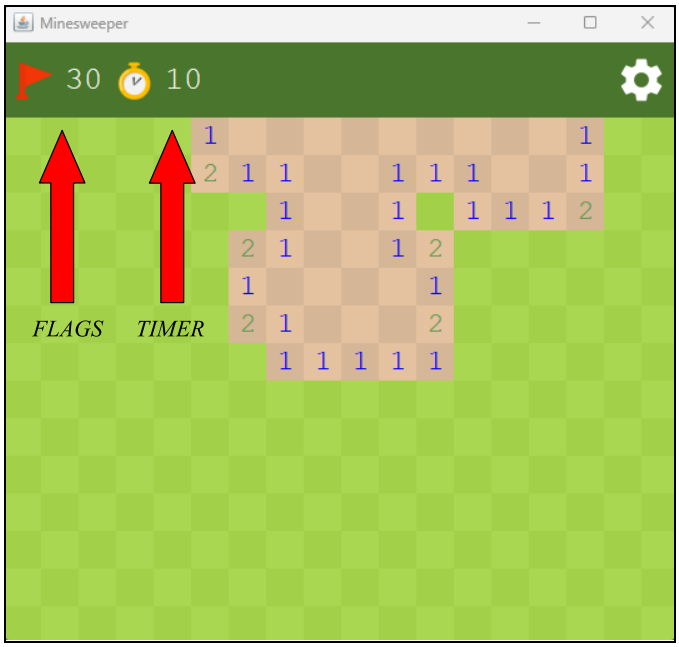
Upon first opening the application, you will be greeted by the welcome screen, shown below:



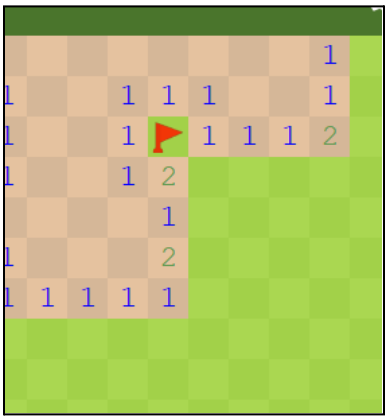
As you may have observed, there are brief instructions on how to play the game. However, since they are so brief, we will endeavour to explain them in greater depth in this section. Please click on the “Start New Game” button to proceed.



Once the welcome screen has disappeared, LEFT CLICK on any one of the tiles to start the game.



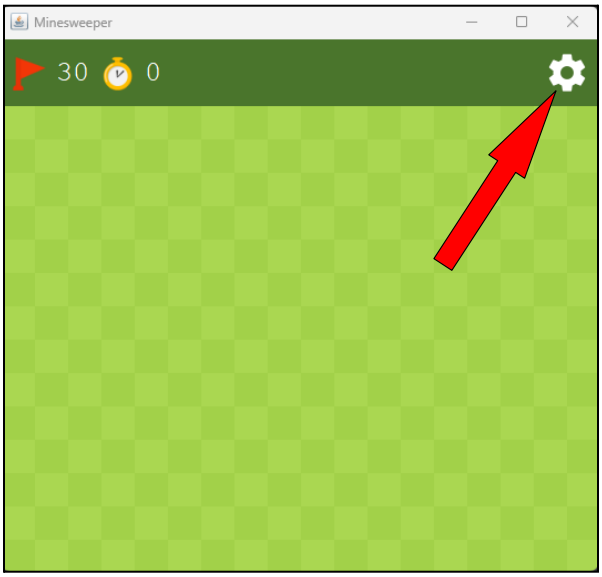
A portion of tiles will become “cleared”, and the timer (see above) will begin counting upwards from zero.



The numbers on uncovered tiles indicate how many “mines” are *adjacent* to that tile. Tiles are *adjacent* to one another when they share at least one corner. If an uncovered tile does not have a number on it, there are no mines adjacent to that tile.

Once you are certain that you have discovered a mine, RIGHT CLICK the tile to place a flag on it (see above). Similarly, LEFT CLICK a tile to clear it if you are certain that it is not covering a mine. The goal of the game is to clear all tiles that do not have mines under them as fast as possible.

If you clear all tiles successfully, you will receive a notification indicating the total time elapsed, which will be recorded in a text file that can be accessed in any of the following ways:



1. Click on the settings icon in the top right corner. Click on “Score Record”. A text file will be opened, with a record of all your successful attempts in order of fastest to slowest, including the dates and times upon which they were completed. (NOTE: if your operating system is not Windows, this feature may not work as expected).

2. Alternatively, you can manually open the score record by finding “record.txt” in the application’s folder. This will work regardless of the operating system.

However, if you clear a tile with a mine underneath it, the game will end and all mines will be revealed. The save file will be wiped as well.

Code Spotlight

This section outlines some of the key programming techniques deployed in the development of the MineSweeper application.

1. Multi-dimensional parallel arrays: The game board data as a two-dimensional integer array (generated from Java's built-in Random class) communicates with a parallel Graphical User Interface game board array to accurately portray information to the user.

2. Object Orientation: the program is divided into six Java classes, which represent button objects, JFrame objects, and Java Swing component objects, which can be easily created, modified, and destroyed using their respective class methods.

3. Anonymous Functions and Classes: Lambda expressions and anonymous class declarations are used in multiple locations for convenience and readability.

4. Algorithms:
    a. Recursive Algorithms: the method used to uncover tiles calls itself recursively, depending on the states of adjacent tiles.

    b. Searching Algorithms: a linear search algorithm is used in multiple locations to determine the attributes of tiles in the gameboard.

    c. Sorting Algorithms: the Quick Sort algorithm is used to sort the score record.

5. External File Reading and Writing: text files are used to store application data and score records.