VIETNAM NATIONAL UNIVERSITY OF HO CHI MINH CITY

THE INTERNATIONAL UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

A blue and white logo

Description automatically generated with low confidence

MINESWEEPER GAME

**GROUP’S MEMBERS**

**Đặng Khải Dương – ITITWE19010**

**Đinh Bình Thanh Thông – ITITWE19027**

**Nguyễn Tấn Tài – ITITWE19023**

**Hồ Tiến Đạt – ITCSIU21047**

Ho Chi Minh City, Vietnam

Year 2022-2023

Contents

[**I/PROJECT DESCRIPTION** 3](#_Toc136382048)

[**Overview** 4](#_Toc136382049)

[**Development History** 4](#_Toc136382050)

[**Gameplay** 4](#_Toc136382051)

[**Scope of Work** 4](#_Toc136382052)

[**Conclusion** 4](#_Toc136382053)

[**II/PROGRAMING LANGUAGE** 3](#_Toc136382054)

[**III/ USE CASE DIAGRAM** 3](#_Toc136382055)

[**IV/ ORGANIZATION SCHEDULE** 3](#_Toc136382056)

[**1.** **Gantt Chart** 4](#_Toc136382057)

[**2.** **Work Breakdown Structure** 4](#_Toc136382058)

[**3.** **Schedule and milestone** 4](#_Toc136382059)

[***Table 1 .* Schedule and milestone** 4](#_Toc136382060)

[**V/ RELATED TO DSA** 3](#_Toc136382061)

[**VI/ CLASS DIAGRAM DESIGN** 3](#_Toc136382062)

[**VII/ IMPLEMENTATION** 3](#_Toc136382063)

[**1. Gameplay Function** 4](#_Toc136382064)

[**2. Number Function** 4](#_Toc136382065)

[**3. Flag Function** 4](#_Toc136382066)

[**4. Reset Game Function** 4](#_Toc136382067)

[**5. Save Game Function** 4](#_Toc136382068)

[**6. Open Existing Game Function** 4](#_Toc136382069)

[**VIII/ TESTCASE** 3](#_Toc136382070)

[**Table 2. Test case 1** 5](#_Toc136382071)

[**Table 3. Test case 2** 6](#_Toc136382072)

[**Table 4. Test case 3** 7](#_Toc136382073)

[**Table 5. Test case 4** 8](#_Toc136382074)

[**Table 6. Test case 5** 9](#_Toc136382075)

[**Table 7. Test case 6** 10](#_Toc136382076)

[**Table 8. Test case 7** 11](#_Toc136382077)

[**IX/ GLOSSARY** 11](#_Toc136382078)

[**References** 11](#_Toc136382079)

**Contribution**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | **Khải Dương** | **Thanh Thông** | **Tấn Tài** | **Tiến Đạt** |
| PERCENTAGE | 30% | 30% | 20% | 20% |

**Github**

<https://github.com/maplepold/DSA_Project>

# **I/PROJECT DESCRIPTION**

## **Overview**

Minesweeper is a game usually played by one individual. The rule is very simple, you play on a rectangular board . Your job is to locate a fixed number of "mines"which are placed at random positions. But the tricky part here is the player has to finish each level as fast as possible by choosing the "safe" squares. At the same time, they have to stay away from the “mines” square as well. Simple interface friendly to players but still Minesweeper creates a huge impact as well as triggering people's interest while playing it. There are some variants of versions of Minesweeper nowadays with more updates in each level to make the game more special and suitable to the next generation.

## **Development History**

Minesweeper is a classic puzzle game that has been included in all versions of Microsoft Windows since Windows 3.1. It was initially developed by Microsoft programmer Curt Johnson in 1989 and was released as part of the Microsoft Entertainment Pack 1 for Windows 3.1. The game has undergone numerous changes and enhancements since its inception, with later versions featuring improved graphics and sound effects. Additionally, new game modes, such as "Advanced Mode" in Windows 7 and "Adventure mode" in Windows 8 and 10, have been introduced to increase the game's complexity and add a storyline. Despite these modifications, the gameplay of Minesweeper has remained constant, making it a popular game among Windows users for over 30 years. The game's success has led to its porting to other platforms, such as mobile devices and gaming consoles.

## **Gameplay**

In the puzzle video game Minesweeper, mines are dispersed into cells that can be in one of three states: unopened, opened, or flagged. A mined cell can be opened by a player, but doing so results in a loss and the game is over. Unopened cells that have been flagged by the player serve as potential mine locations. A flag is used to indicate the presence of a mine in that location. In a procedure known as chording, all nearby, unopened, non-flagged cells are opened.

During the game, the player deduces which cells are safe to open based on the information offered by the opened cells. The mine count, or the amount of remaining mines on the board, is also displayed to the player.

## **Scope of Work**

Minesweeper is a game project created when the Team uses an algorithm and data structure to store the state of each tile on the board, including whether or not the cell contains mines.

1 .The team will need to develop algorithms that generate game boards of different sizes and with different numbers of mines.

2. They interact with the game board, where to place the mines and win or lose the game.

3. When a player clicks on a tile on the game board, algorithms will display that tile, taking into account the existence of surrounding mines and whether the tile is a mine.

4. We require algorithms that display the tiles on the game board as the player clicks them, taking into account the presence of mines in the next tile as well as whether or not the player is mine. They are not.

5.When the game is done, either because the player has uncovered all the tiles without mines or because they clicked on the tile containing the mines.

6.We needed to make the game operate smoothly and interface with the gaming boat as easily as possible.

## **Conclusion**

Millions of people of all ages have enjoyed playing the game minesweeper. Many variations exist single-player logical that are not only well-known and contemporary but are also setting a trend in day-to-day life. To store each tile on the board for the game, algorithms and data structures are needed. The group also had to develop an algorithm that, when a player clicks a tile on the game board, displays all of the mines on that tile as well as whether or not the player is currently on one. A successful game needs a streamlined UI and easy-to-use functionality.

# **II/PROGRAMING LANGUAGE**

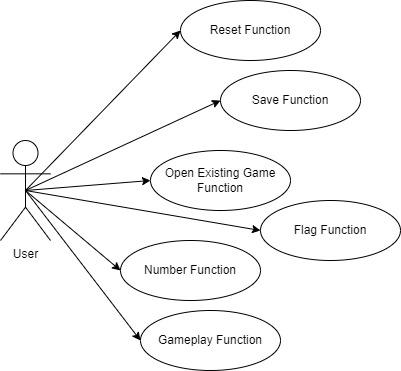
What is Java?

****

In order to have as few implementation dependencies as possible, Java is a high-level, class-based, object-oriented programming language. Programmers can write code in this general-purpose language once and have it run on any computer. Typically, Java applications are translated into bytecode that can be executed on any Java virtual machine (JVM). Java's syntax is comparable to that of C and C++, but it offers fewer low-level facilities. In contrast to conventional compiled languages, the Java runtime offers dynamic capabilities.

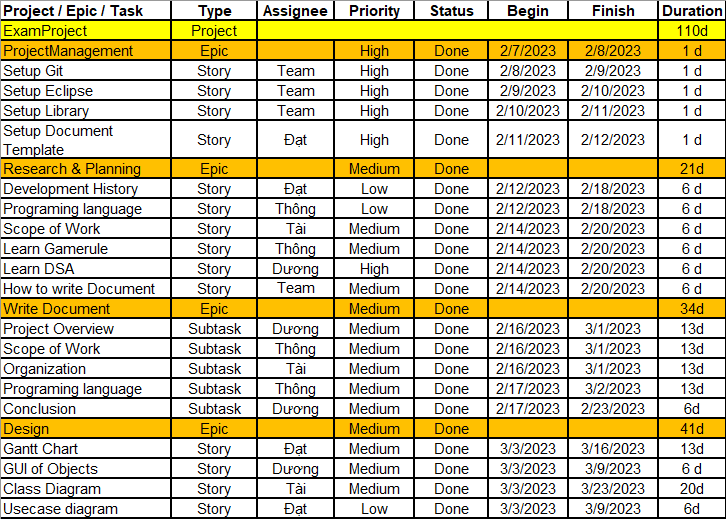
# **III/ USE CASE DIAGRAM**

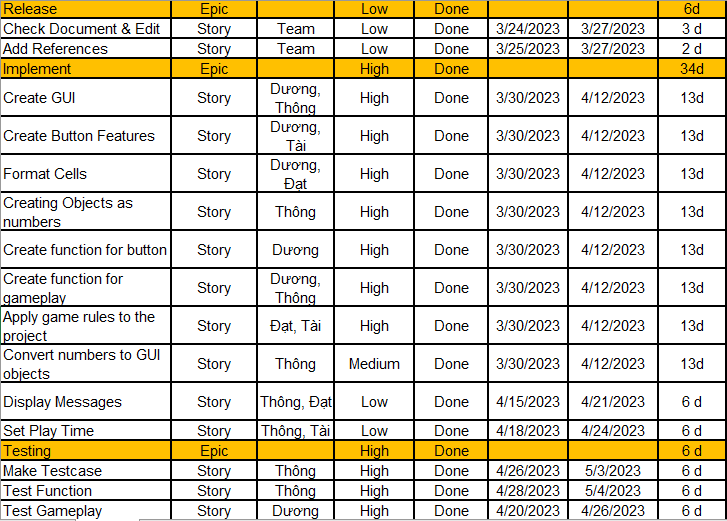
Use Case diagram is to represent the functions that can be performed by the user.

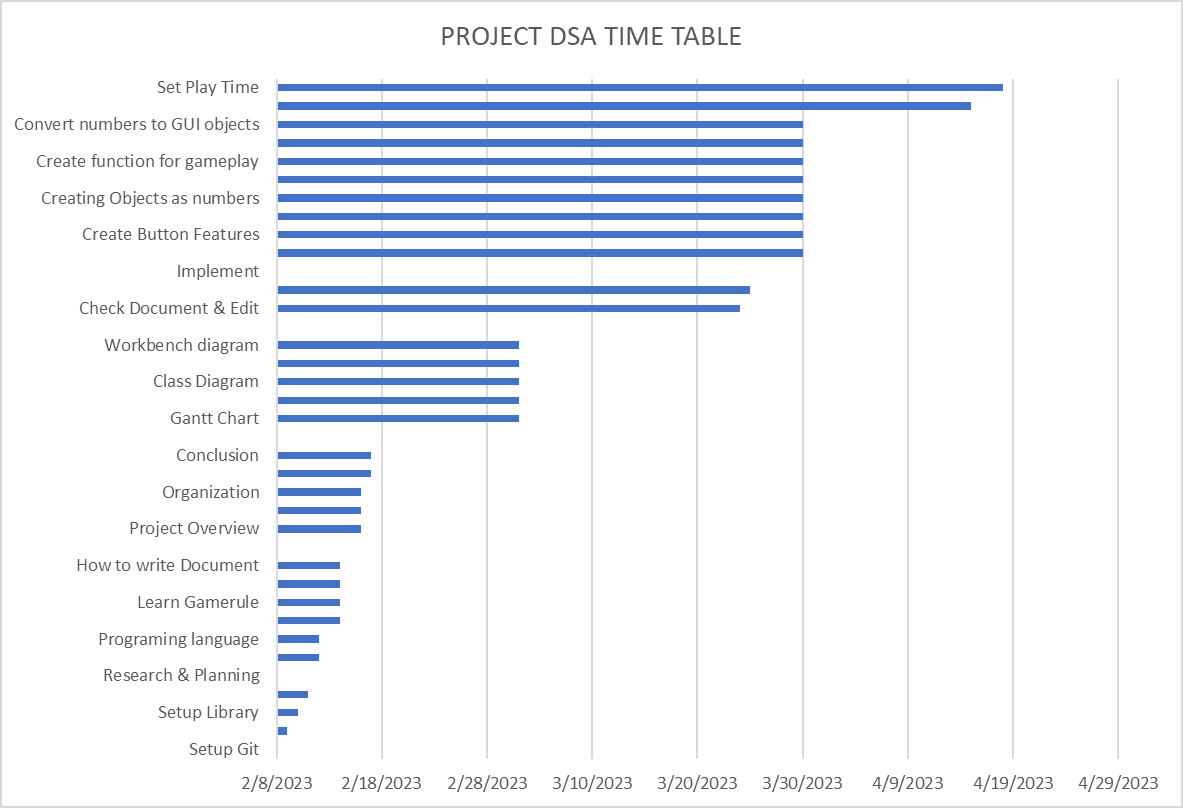
**  
Figure 1 . Use Case Diagram**

# **IV/ ORGANIZATION SCHEDULE**

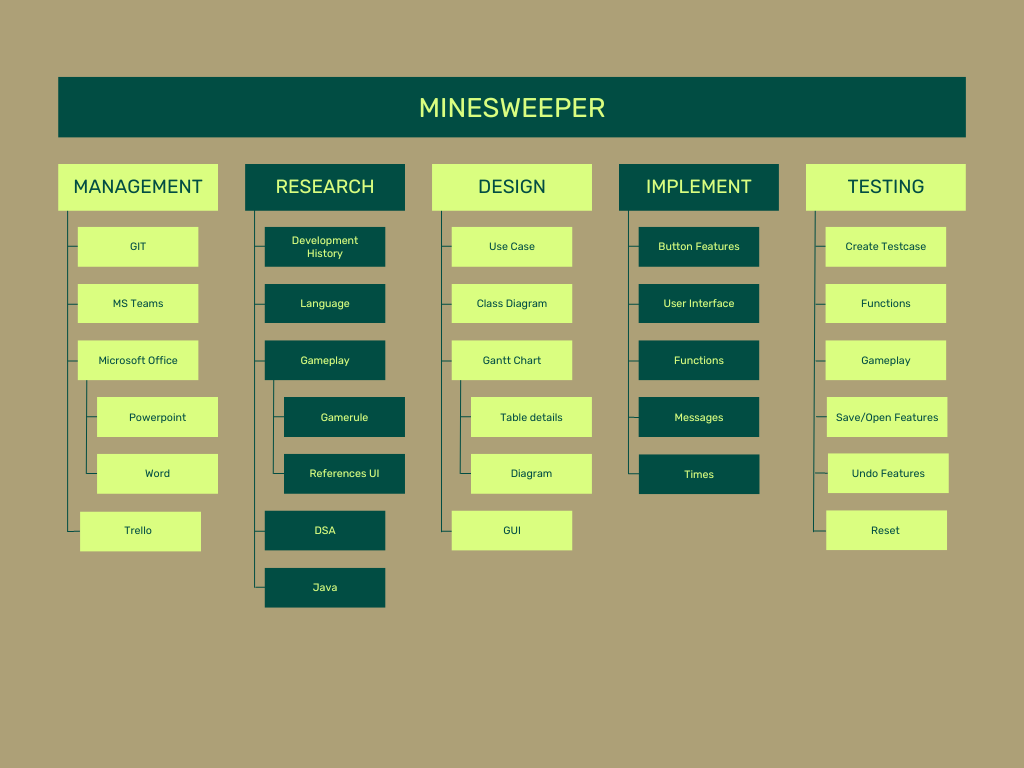
## **Gantt Chart**







## **Work Breakdown Structure**

****Figure 3 . Work breakdown structure**

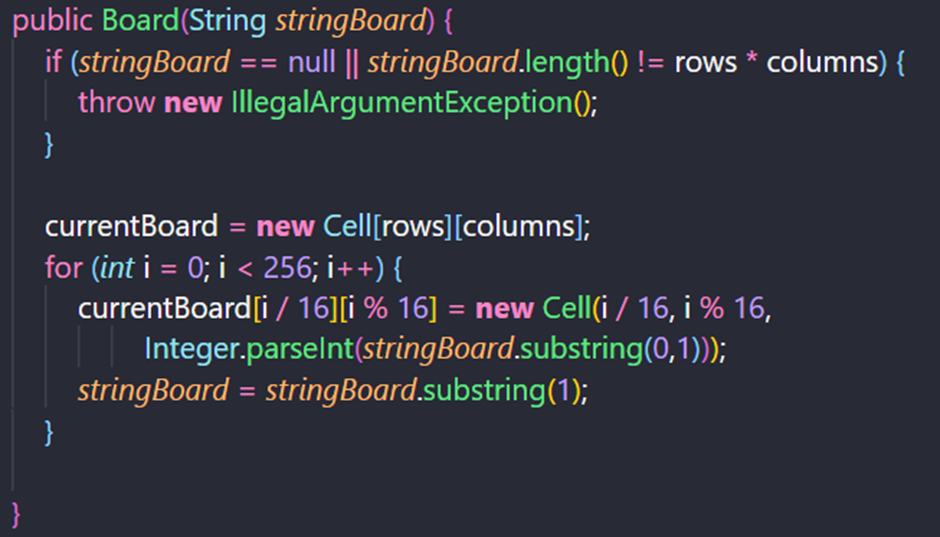
## **Schedule and milestone**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Project / Epic / Task** | **Type** | **Assignee** | **Priority** | **Status** | **Estimate Times** |
| ExamProject | Project |  |  |  | 110d |
| ProjectManagement | Epic |  | High | Done | 1 d |
| Setup Git | Story | Team | High | Done | 1 d |
| Setup VSCode | Story | Team | High | Done | 1 d |
| Setup Library | Story | Team | High | Done | 1 d |
| Setup Document Template | Story | Đạt | High | Done | 1 d |
| Research & Planning | Epic |  | Medium | Done | 21d |
| Development History | Story | Đạt | Low | Done | 6 d |
| Programing language | Story | Thông | Low | Done | 6 d |
| Scope of Work | Story | Tài | Medium | Done | 6 d |
| Learn Gamerule | Story | Thông | Medium | Done | 6 d |
| Learn DSA | Story | Dương | High | Done | 6 d |
| How to write Document | Story | Team | Medium | Done | 6 d |
| Write Document | Epic |  | Medium | Done | 34d |
| Project Overview | Subtask | Dương | Medium | Done | 13d |
| Scope of Work | Subtask | Thông | Medium | Done | 13d |
| Organization | Subtask | Tài | Medium | Done | 13d |
| Programing language | Subtask | Thông | Medium | Done | 13d |
| Conclusion | Subtask | Dương | Medium | Done | 6d |
| Design | Epic |  | Medium | Done | 41d |
| Gantt Chart | Story | Đạt | Medium | Done | 13d |
| GUI of Objects | Story | Dương | Medium | Done | 6 d |
| Class Diagram | Story | Tài | Medium | Done | 20d |
| Usecase diagram | Story | Đạt | Low | Done | 6d |
| Workbench diagram | Story | Dương | Low | Done | 6d |
| Release | Epic |  | Low | Done | 6d |
| Check Document & Edit | Story | Team | Low | Done | 3 d |
| Add References | Story | Team | Low | Done | 2 d |
| Implement | Epic |  | High | Done | 34d |
| Create GUI | Story | Dương, Thông | High | Done | 13d |
| Create Button Features | Story | Dương, Tài | High | Done | 13d |
| Format Cells | Story | Dương, Đạt | High | Done | 13d |
| Creating Objects as numbers | Story | Thông | High | Done | 13d |
| Create function for button | Story | Dương | High | Done | 13d |
| Create function for gameplay | Story | Dương, Thông | High | Done | 13d |
| Apply game rules to the project | Story | Đạt, Tài | High | Done | 13d |
| Convert numbers to GUI objects | Story | Thông | Medium | Done | 13d |
| Display Messages | Story | Thông, Đạt | Low | Done | 6 d |
| Set Play Time | Story | Thông, Tài | Low | Done | 6 d |
| Testing | Epic |  | High | Done | 6 d |
| Make Testcase | Story | Thông | High | Done | 6 d |
| Test Function | Story | Thông | High | Done | 6 d |
| Test Gameplay | Story | Dương | High | Done | 6 d |
| Test Save/Open Feature | Story | Dương | Medium | Done | 6 d |
| Test Reset Game Feature | Story | Dương | Medium | Done | 6 d |
| Make Final Report | Epic |  | Low | Done | 13d |
| Add Testcase | Story | Thông | Low | Done | 6 d |
| Explain DSA Function | Story | Dương | Medium | Done | 6 d |
| Update Schedule | Story | Đạt | Low | Done | 6 d |
| Add Implement | Story | Tài | Low | Done | 6 d |
| Prepare for Presentation | Epic |  | Low | Done | 6 d |
| Make PowerPoint | Story | Team | Low | Done | 6 d |
| Prepare to Answer the question | Story | Team | Low | Done | 6 d |
| Presentation | Epic |  | High | To Do | 0 d |

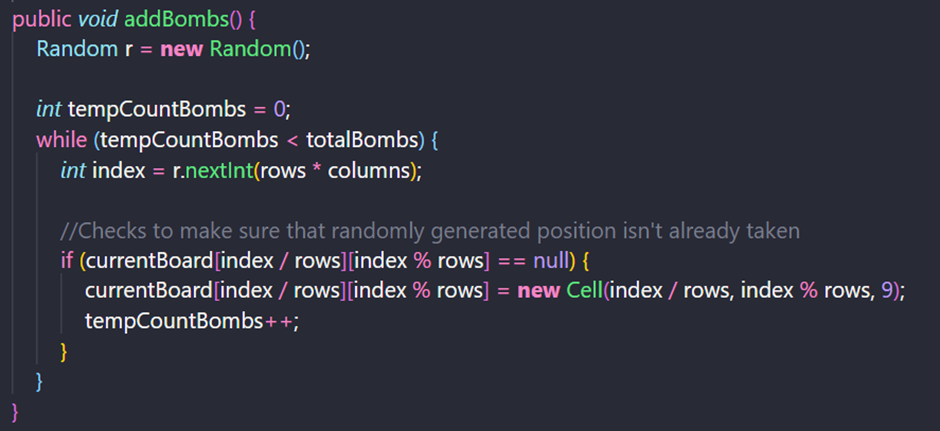
### ***Table 1 .* Schedule and milestone**

# **V/ RELATED TO DSA**

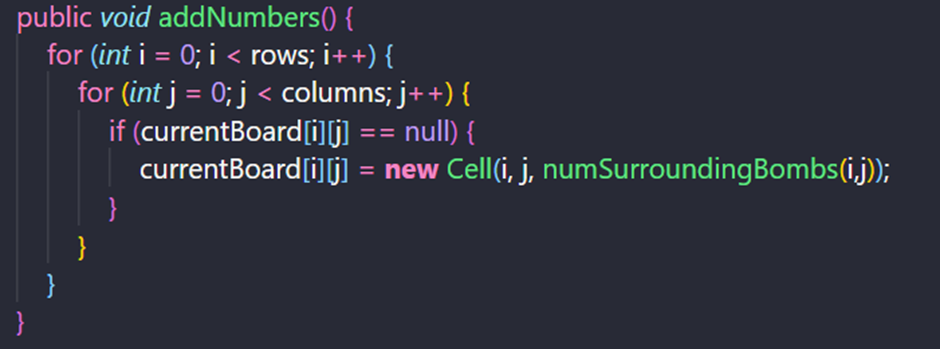
1. **Overloaded constructor. Takes in a string representation of a board state**

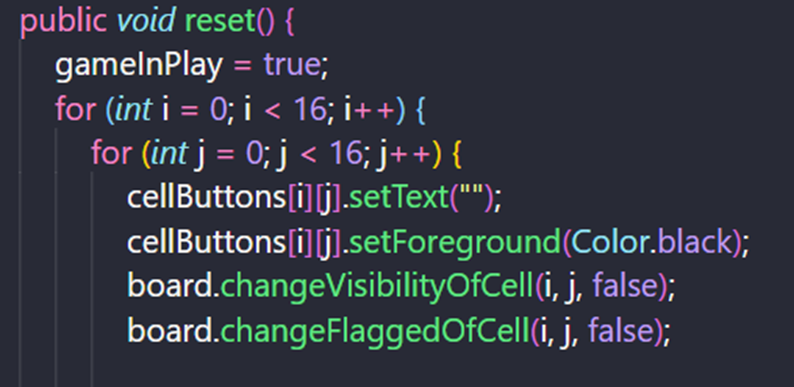
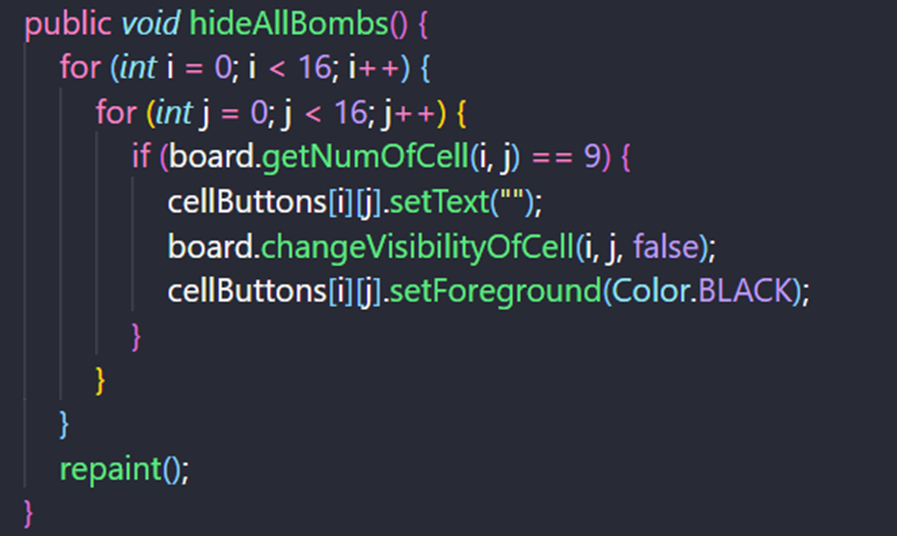
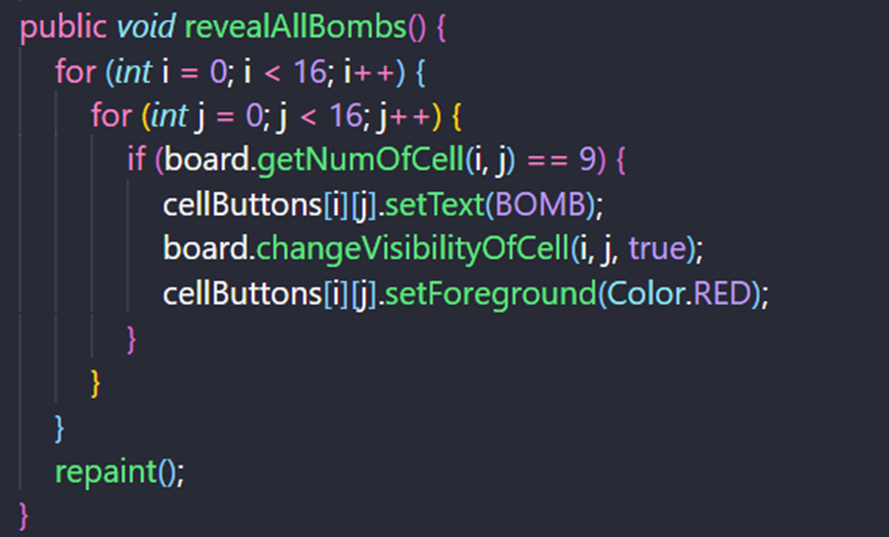
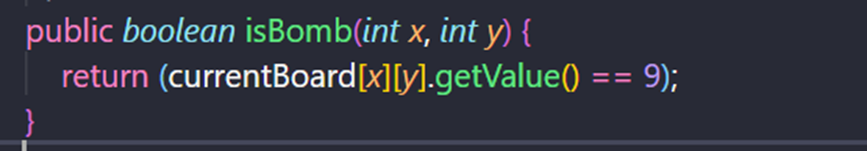
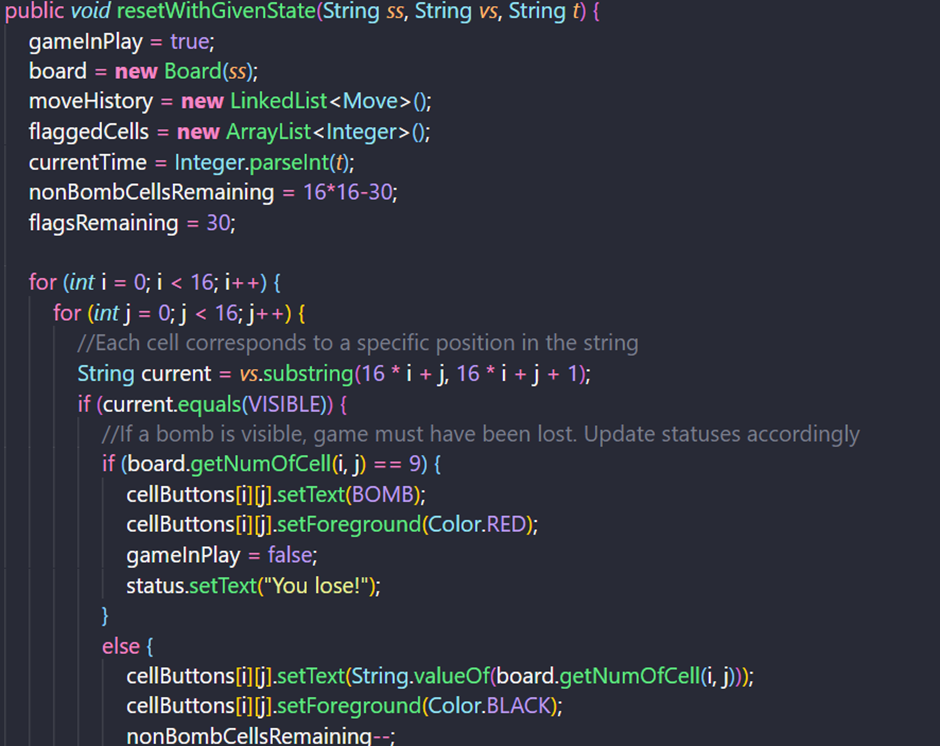
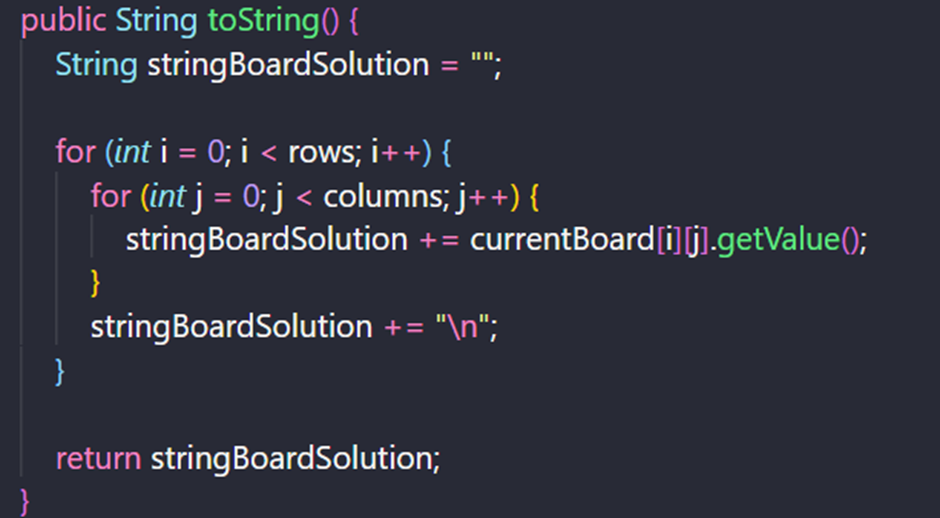
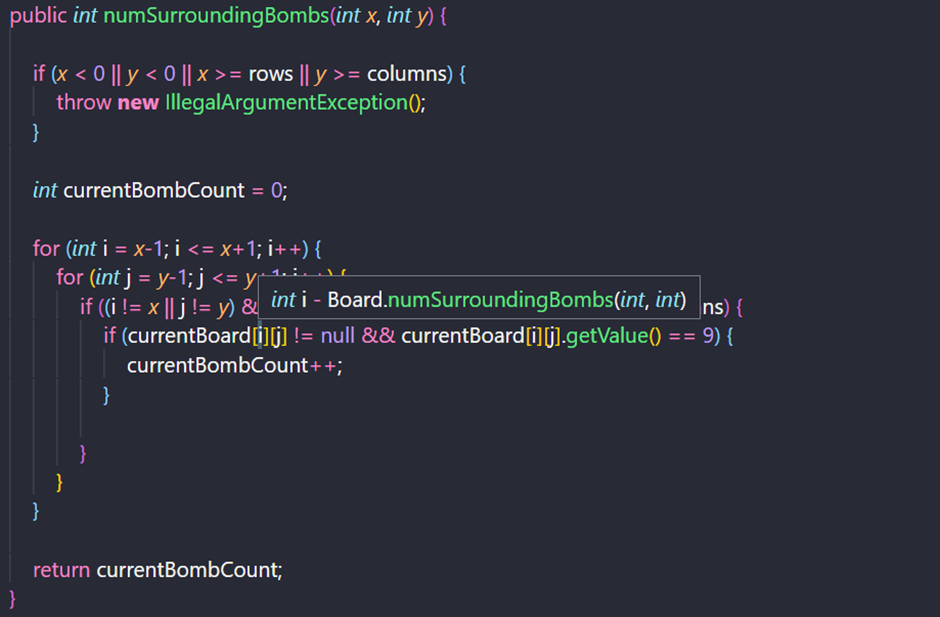
****

1. **Uses a random number generator to determine placements of bombs.**

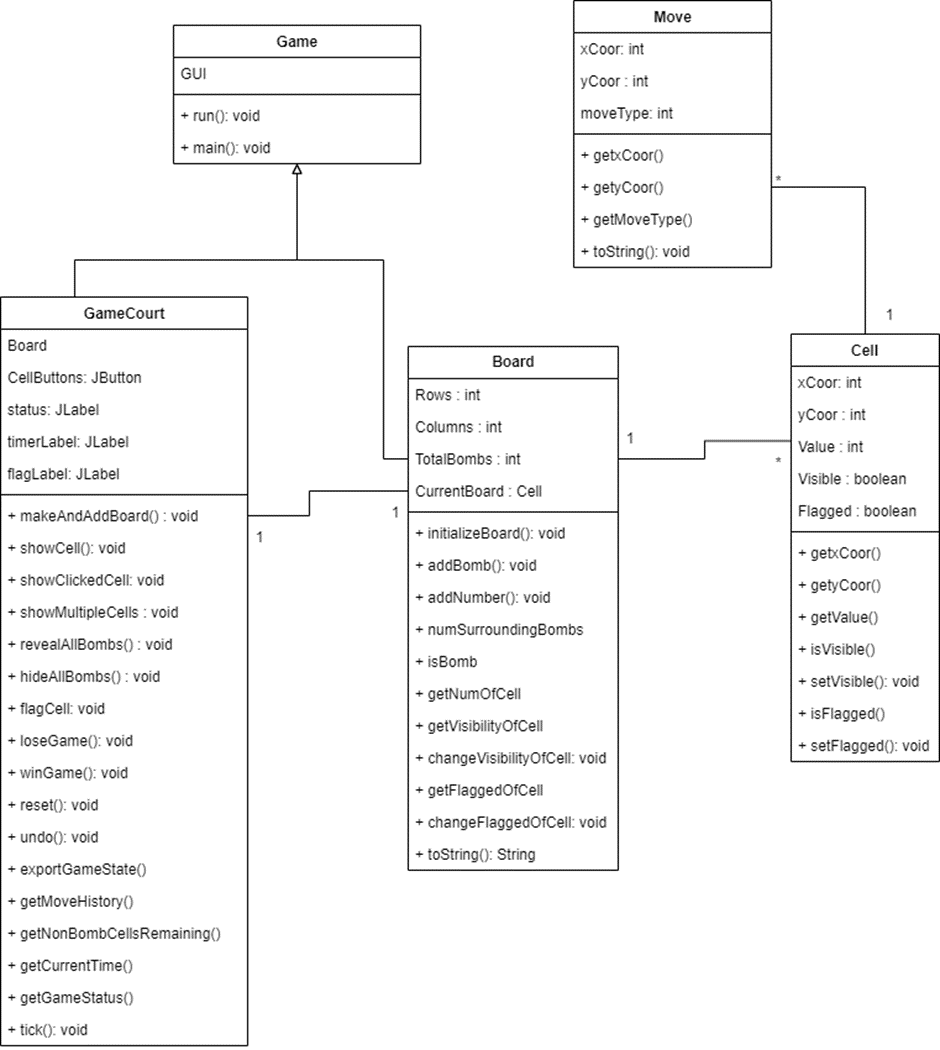
****

**3**. **After the bombs are in place, numbers are added. Only provides numerical count for cells that aren't bombs.**



**4. Accounts for corner and edge cells**

# **VI/ CLASS DIAGRAM DESIGN**

****

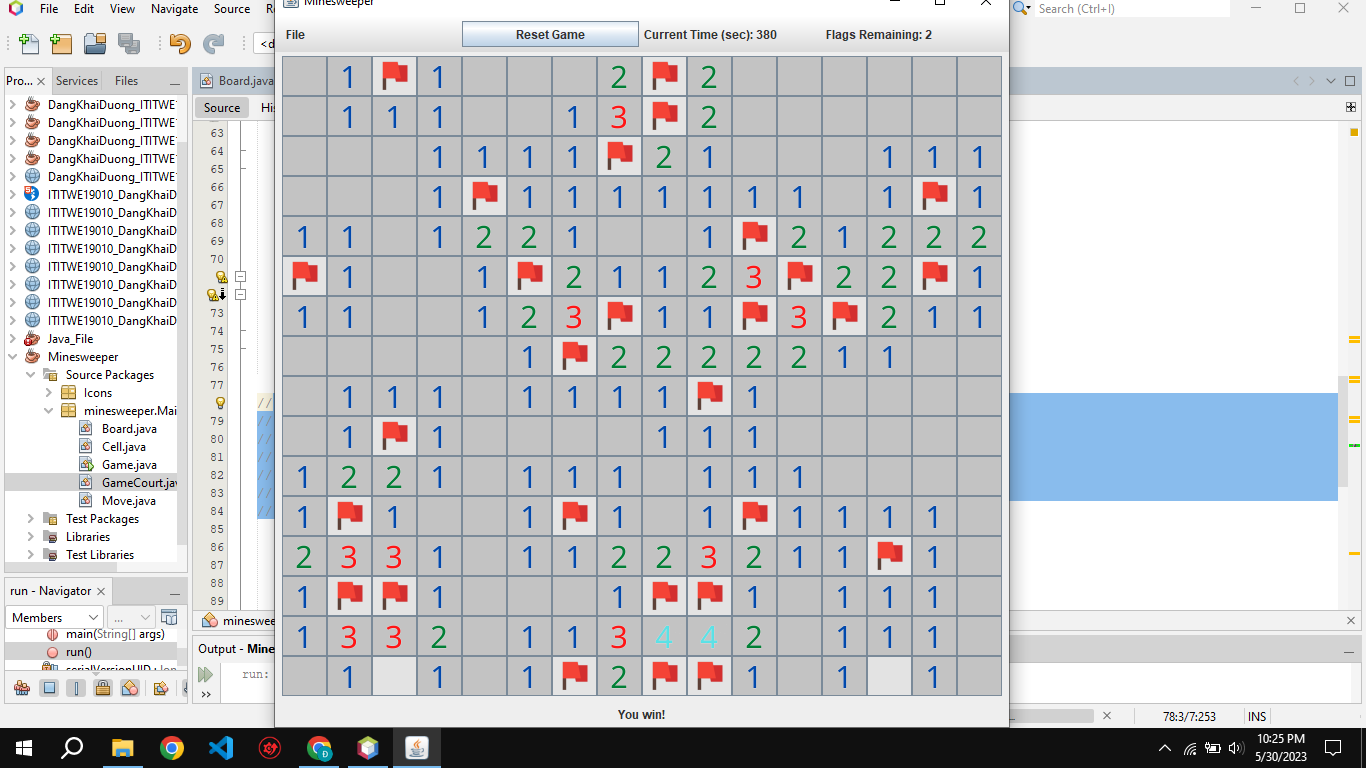
# **VII/ IMPLEMENTATION**

## **1. Gameplay Function**

When the user runs the code, the game interface will default that you have started the game and the watch will calculate how long it took the user to complete a game turn.

****

When the user wins the game, the system will show the winning message to the user

****

When the user presses the correct location of the hidden bomb, it means they have lost the game and receive a notification of defeat and the bombs will explode.

****

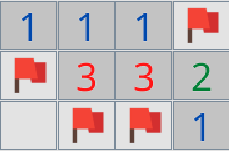
## **2. Number Function**

When the player clicks anywhere on the game interface, numbers appear showing the location of bombs.

****

## **3. Flag Function**

The player right-clicks to place a flag marking a bomb's location.

****

By default, there are 30 flags corresponding to 30 hidden bombs in the game.



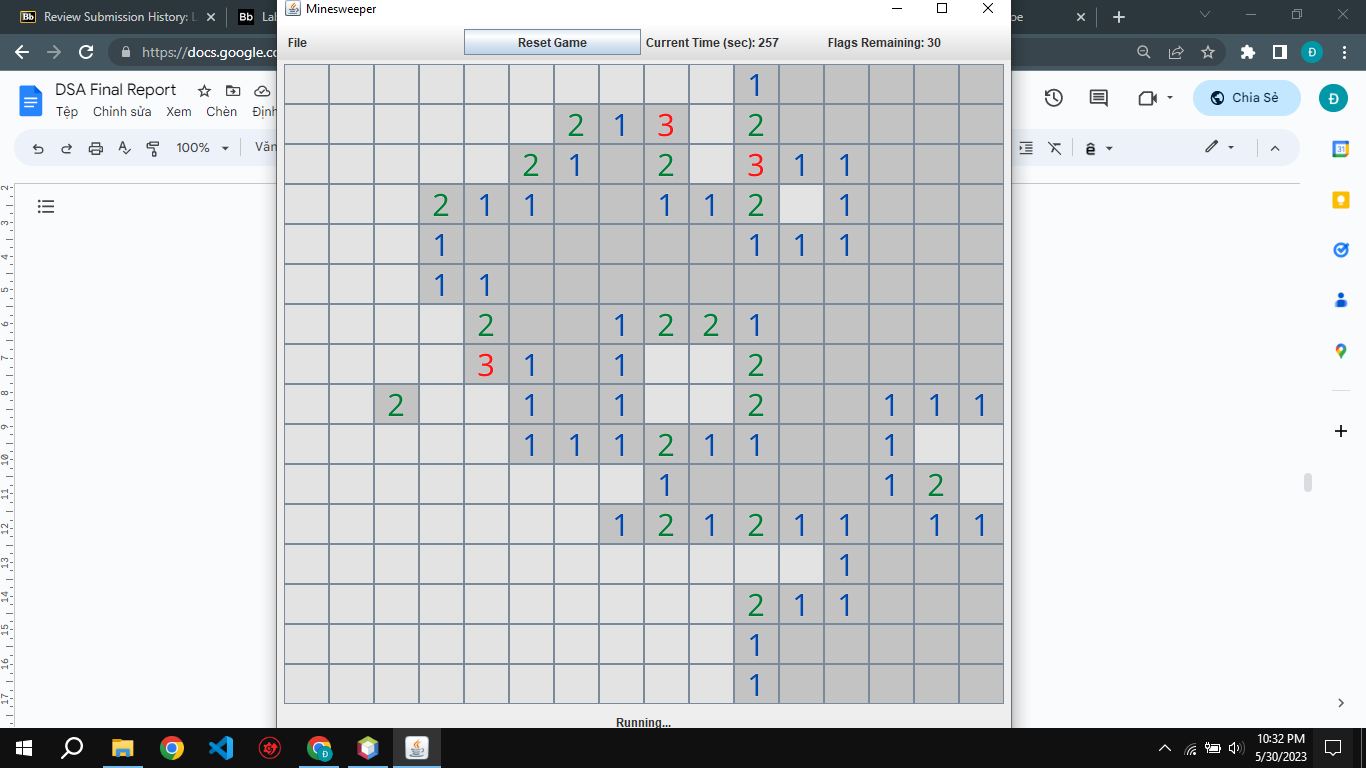
## **4. Reset Game Function**

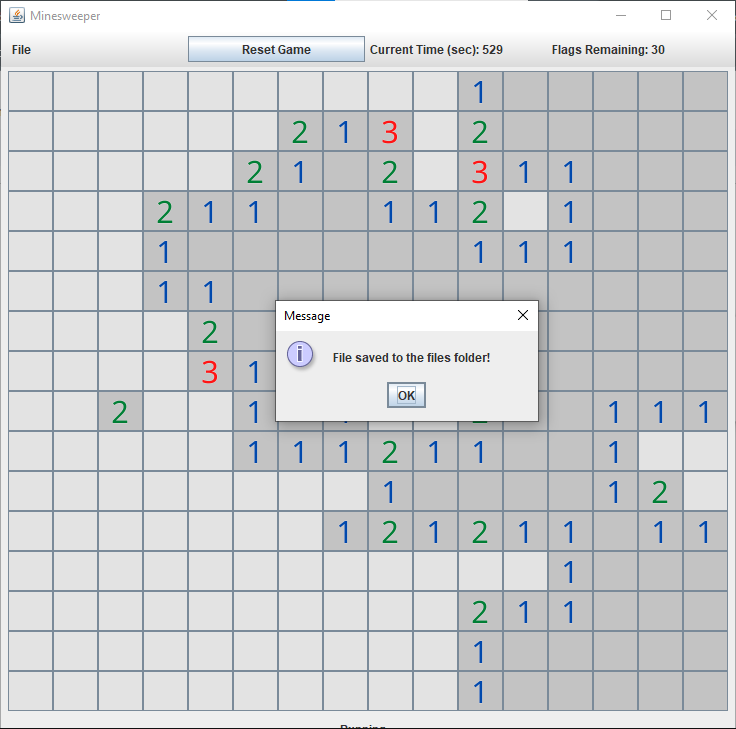
When the player wants to start a new game, they can press the "Reset Game" button to return to the initial game interface.

****

## **5. Save Game Function**

If the player wants to save their unfinished game to continue playing later, they can click the "Save" button in the "File" menu on the game's control bar.

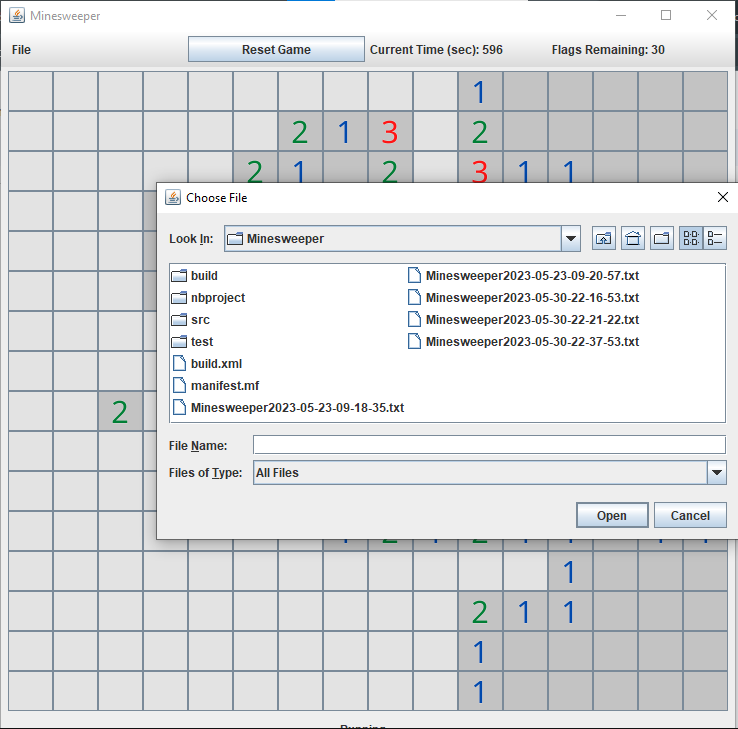
****

****

At the end, a message will display "File saved to files folder" and the .txt file will automatically save.

## **6. Open Existing Game Function**

When the player wants to continue an unfinished, previously saved game, they can press the "Open" button in the "File" menu on the control bar.

****

A window will open for the player to select the location of the saved data file. If the selected file is valid, the game interface will load the previously saved game state so the player can continue where they left off.

****

# **VIII/ TESTCASE**

Test Case Tables represent the test steps based on the function of Use Case.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Scenario** | Save Game function | | |
| **Test ID** | TC-1 | **Test Case Name** | Save recent game results to file |
| **Designed by** | Dương | **Design Date** | 22/05/2023 |
| **Executed by** | Dương | **Execution Date** | 25/05/2023 |
| **Pre-Conditions** | 1. When the player opens and runs the game, it launches successfully.  2. The "Save" button is displayed correctly on the control bar and works as intended. | **Post-Condition(s)** | 1. The game runs without any bugs or crashes, functioning properly during play. |
| **Test Data/Action** | Click the “Save” button | | |
| **Test Step** | **Expected Result** | **Comments** | **Pass/Fail** |
| 1.The user opens and launches the Minesweeper game application.  2.The user clicks on any of the cells/squares on the game board.  3.The user clicks the "Save" button to save their current game state. | 1. The system displays the initial game interface when the game is started.  2.The system displays any in-game functions, such as revealing cells, placing flags, etc.  3.The system displays a message box stating "File saved to the files folder!" along with an "OK" button. The current game state is saved to the "files" folder.  4.The system returns to the main game interface after the user clicks "OK", allowing the user to continue playing. |  | PASS |

### **Table 2. Test case 1**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Scenario** | Reset Game function | | |
| **Test ID** | TC-2 | **Test Case Name** | Reset to a new game |
| **Designed by** | Dương | **Design Date** | 22/05/2023 |
| **Executed by** | Thông | **Execution Date** | 25/05/2023 |
| **Pre-Conditions** | 1. When the player launches the game, it opens and runs as intended without any issues.  2. The "Reset Game" button is displayed correctly on the control bar and functions properly, allowing the player to restart the game. | **Post-Condition(s)** | 1. The game runs without any bugs or crashes, functioning properly during play. |
| **Test Data/Action** | Click the “Reset Game” button | | |
| **Test Step** | **Expected Result** | **Comments** | **Pass/Fail** |
| 1. User opens the game.  2. User clicks a cell.  3. User clicks "Reset Game". | 1. The system shows the new game interface.  2. The system shows in-game functions.  3. The system clears all data, reverts to the new game interface and resets the timer to 0 seconds. |  | PASS |

### **Table 3. Test case 2**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Scenario** | Open Existing Game function | | |
| **Test ID** | TC-3 | **Test Case Name** | Open the valid file saved |
| **Designed by** | Thông | **Design Date** | 22/05/2023 |
| **Executed by** | Tài | **Execution Date** | 27/05/2023 |
| **Pre-Conditions** | 1. The game launches successfully when the player opens it.  2. The "Open" button is displayed correctly on the control bar and functions properly. | **Post-Condition(s)** | 1. The game runs smoothly without any bugs or crashes.  2. The "Open File" dialog box functions correctly, allowing the user to choose a saved game file. |
| **Test Data/Action** | Click Open and Choose file: “Minesweeper2023-05-23-09-18-35.txt” | | |
| **Test Step** | **Expected Result** | **Comments** | **Pass/Fail** |
| 1. The user opens the Minesweeper game application.  2. The user clicks the "Open" button on the game's control bar.  3. The user selects a valid ".txt" file saved game from the file directory and clicks the "Open" button. | 1. The system displays the new game interface. 2. The system displays the "Open File" dialog box where the user can choose a saved game file. 3. The system loads and displays the saved game data, including the time when the game was saved. |  | PASS |

### **Table 4. Test case 3**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Scenario** | Flag function | | |
| **Test ID** | TC-4 | **Test Case Name** | Flag with right-click |
| **Designed by** | Dương | **Design Date** | 23/05/2023 |
| **Executed by** | Tài | **Execution Date** | 25/05/2023 |
| **Pre-Conditions** | 1. The game launches successfully when the player opens it. | **Post-Condition(s)** | 1. The game runs smoothly without any bugs or crashes. 2. The flagging mechanism - where the player can flag suspected bomb locations - works correctly as intended. |
| **Test Data/Action** | Right-click | | |
| **Test Step** | **Expected Result** | **Comments** | **Pass/Fail** |
| 1. The user opens the Minesweeper game application. 2. The user right-clicks on any cell on the game board. | 1. The system displays the initial game interface. 2. The system displays the flag GUI (graphical user interface) and decrements the remaining flags count by 1 (for example, from 30 to 29). |  | PASS |

### **Table 5. Test case 4**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Scenario** | Number function | | |
| **Test ID** | TC-5 | **Test Case Name** | Number surrounding the bomb |
| **Designed by** | Thông | **Design Date** | 23/05/2023 |
| **Executed by** | Dương | **Execution Date** | 26/05/2023 |
| **Pre-Conditions** | 1. The player opens and runs the game successfully. | **Post-Condition(s)** | 1. No errors when the game running.  2. Do not touch the bomb.  3. The Number GUI works properly |
| **Test Data/Action** | Left-click on Number Cells | | |
| **Test Step** | **Expected Result** | **Comments** | **Pass/Fail** |
| 1. The user opens the Minesweeper game application. 2. The user left-clicks on any cell on the game board. | 1. The system displays the initial game interface. 2. The system displays the number GUI (graphical user interface) showing how many bomb cells are adjacent to the clicked cell. |  | PASS |

### **Table 6. Test case 5**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Scenario** | Gameplay function | | |
| **Test ID** | TC-6 | **Test Case Name** | Successfully Win Game |
| **Designed by** | Dương | **Design Date** | 24/05/2023 |
| **Executed by** | Đạt | **Execution Date** | 27/05/2023 |
| **Pre-Conditions** | 1. The player opens and runs the game successfully. | **Post-Condition(s)** | 1. No errors when the game running.  2. Do not touch the bomb. |
| **Test Data/Action** | Complete all Cells | | |
| **Test Step** | **Expected Result** | **Comments** | **Pass/Fail** |
| 1. User opens game. 2. User clicks cell. 3. User clicks last non-mine cell to win. | 1. Shows start screen. 2. Shows GUIs. 3. Shows "You win!" message. |  | PASS |

### **Table 7. Test case 6**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Scenario** | Gameplay function | | |
| **Test ID** | TC-7 | **Test Case Name** | Touch the Bomb |
| **Designed by** | Thông | **Design Date** | 24/05/2023 |
| **Executed by** | Đạt | **Execution Date** | 27/05/2023 |
| **Pre-Conditions** | 1. The player opens and runs the game successfully. | **Post-Condition(s)** | 1. No errors when the game running.  2. Touch the bomb.  3. The Bomb GUI works properly. |
| **Test Data/Action** | Left-click on Bomb Cells | | |
| **Test Step** | **Expected Result** | **Comments** | **Pass/Fail** |
| 1. User opens game 2. User clicks bomb cell | 1. Shows start screen 2. Shows bomb GUI and "You lose!" message |  | PASS |

### **Table 8. Test case 7**

# **IX/ GLOSSARY**

## **References**

1. [*https://en.wikipedia.org/wiki/Java\_(programming\_language)*](https://en.wikipedia.org/wiki/Java_(programming_language))
2. [*https://www.geeksforgeeks.org/data-structures/*](https://www.geeksforgeeks.org/data-structures/)
3. [*https://minesweepergame.com/strategy/how-to-play-minesweeper.php#:~:text=Minesweeper%20Rules&text=Minesweeper%20is%20a%20game%20where,mine%20you%20lose%20the%20game!*](https://minesweepergame.com/strategy/how-to-play-minesweeper.php#:~:text=Minesweeper%20Rules&text=Minesweeper%20is%20a%20game%20where,mine%20you%20lose%20the%20game!)
4. [*https://en.wikipedia.org/wiki/Minesweeper\_(video\_game)#:~:text=The%20earliest%20ancestor%20of%20Minesweeper,%2C%201983)%2C%20and%20Cube.*](https://en.wikipedia.org/wiki/Minesweeper_(video_game)#:~:text=The%20earliest%20ancestor%20of%20Minesweeper,%2C%201983)%2C%20and%20Cube.)