**VIETNAM NATIONAL UNIVERSITY OF HO CHI MINH CITY**

**THE INTERNATIONAL UNIVERSITY**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**





**ALGORITHMS & DATA STRUCTURES**

**PROJECT MID-TERM REPORT**

**GAME PROGRAMMING**

**MINESWEEPER**

**Advisor:**

Prof. Tran Thanh Tung

Prof. Nguyen Quang Phu

**Team Member:**

**Lê Thanh Phương Nam ITITWE19025**

**Đặng Khải Dương ITITWE19010**

**Đinh Bình Thanh Thông ITITWE19027**

Ho Chi Minh city, VietNam

Year 2022

**Outline**

[**Contribution**](#_9rhpal2300s8) **2**

[**I/PROJECT DESCRIPTION**](#_lvysb8ghm8rc) **2**

[**Overview**](#_jegbrosciae1) **2**

[**Development History**](#_dl8kjhmddaid) **2**

[**Gameplay**](#_z5p3lmoli34f) **3**

[**Scope of Work**](#_liu7jbmq905v) **3**

[**Conclusion**](#_9rtui5562vyv) **3**

[**II/PROGRAMING LANGUAGE**](#_1lk09pbt3jxd) **4**

[**III/ USE CASE DIAGRAM**](#_8vs92rvzaaoo) **5**

[**IV/ ORGANIZATION SCHEDULE**](#_deztniqda6ot) **6**

[**Gantt Chart**](#_if2o0mogo8du) **6**

[**Figure 2. Gantt Chart**](#_23ckvvd) **6**

[**Work Breakdown Structure**](#_ihv636) **7**

[**Figure 3. Work breakdown structure**](#_32hioqz) **7**

[**Schedule and milestone**](#_1hmsyys) **8**

[**Table 1. Schedule and milestone**](#_odsstosev5k3) **12**

[**V/ RELATED TO DSA**](#_xu6xrv7wuh5v) **12**

[**VI/ CLASS DIAGRAM DESIGN**](#_i0rmw98z0swn) **17**

[**VII/ IMPLEMENTATION**](#_jz4xcbu0x1l0) **17**

[**Gameplay Function**](#_ltdq5vfp244c) **17**

[**Number Function**](#_dklz23dsxahq) **19**

[**Flag Function**](#_4w6abkjdlnv9) **20**

[**Reset Game Function**](#_rvtxeem7chcq) **20**

[**Save Game Function**](#_w7rxdi5y3qso) **20**

[**Open Existing Game Function**](#_e17xsgoprhx2) **22**

[**Undo Function**](#_qggy0zwewvdm) **25**

[**VIII/ TESTCASE**](#_5zo80ph72qw) **27**

[**Table 2. Test case 1**](#_uwj81fnlsthe) **28**

[**Table 3. Test case 2**](#_n86w9xtmwpg1) **29**

[**Table 4. Test case 3**](#_2kuc6hs1fdx) **30**

[**Table 5. Test case 4**](#_uv3wx4ecrdnu) **31**

[**Table 6. Test case 5**](#_y4aeey9ks2jm) **32**

[**Table 7. Test case 6**](#_4rk2xnmhibd4) **33**

[**Table 8. Test case 7**](#_fi4wyhusisr) **34**

[**Table 9. Test case 8**](#_7uq9nr8ukxhn) **35**

[**Table 10. Test case 9**](#_jnp1joj7ks21) **36**

[**Table 11. Test case 10**](#_eaerfbdo5ap4) **36**

[**IX/ GLOSSARY**](#_1jh6pl5cesyr) **37**

[**References**](#_f6rxjtji4e1q) **37**

# **Contribution**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Lê Thanh Phương Nam | Đặng Khải Dương | Đinh Bình Thanh Thông |
| **Percent** | 1/3 | 1/3 | 1/3 |

**GITHUB Link :** [*https://github.com/maplepold/Minesweeper*](https://github.com/maplepold/Minesweeper)

# I/PROJECT DESCRIPTION

## **Overview**

Minesweeper is a single-player puzzle video game. The goal of the game is to clear a rectangular board containing hidden "mines" or bombs without detonating any of them, with the use of hints about the number of surrounding mines in each area. The game dates back to the 1960s, and it has been adapted for a variety of modern computing platforms. It has numerous variants and offshoots

## **Development History**

**The earliest ancestor of Minesweeper** was Jerimac Ratliff's Cube. The basic gameplay style became a popular segment of the puzzle video game genre during the 1980s. Relentless Logic (or RLogic for short) was available for MS-DOS as early as 1985.

**Minesweeper for Windows**

After first appearing in a Microsoft game bundle in 1990, a version of the game known as Microsoft Minesweeper was published as a standard feature of Windows 3.1 in 1992.

The game's popularity and exposure skyrocketed, with tech bloggers and journalists dubbing it "iconic," "renowned," and even "the most successful game ever." Versions of the game were included in regular Windows installations until Windows Vista in 2007, but with Windows 8 (2012) and later, it must be downloaded as an app from the Microsoft Store.

## **Gameplay**

In Minesweeper, mines (that resemble naval mines in the classic theme) are scattered throughout a board, which is divided into cells. An unopened cell is empty and clickable, whereas an opened cell is visible. Right-clicking on a cell will flag it, resulting in the appearance of a flag on it. Flagged cells are still considered unopened, and a player can open them by clicking on them. A Minesweeper game begins when the player clicks the first cell on a board with all cells closed.

During the game, the player uses the information provided by the opened cells to deduce which cells are safe to open. The player is also given the mine count, which is the number of remaining mines on the board.

## **Scope of Work**

**Data structure:**

Learn and practice problem-solving techniques for Data search, processing speed, and a large number of requests

**Regarding the algorithm:**

To solve difficulties in future applications, one must comprehend and use the algorithms associated with the aforementioned data structure.

## **Conclusion**

Minesweeper is a typical game to be able to make good use of the platforms learned from DSA. From this project, we have gained the necessary amount of knowledge to be able to continue studying subjects related to the major, along with experience from creating a game or an application that helps users with data management and new projects in the future.

# **II/PROGRAMING LANGUAGE**

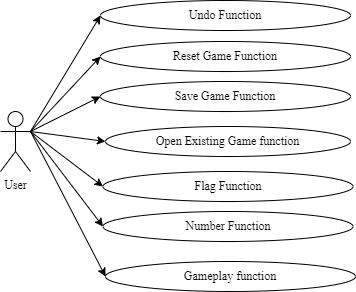
What is Java?

****

Java is an object-oriented programming language with a high level of abstraction and as few implementation dependencies as possible. It is a general-purpose programming language designed to allow programmers to write once and run anywhere (WORA), which means that compiled Java code can run on any platform that supports Java without requiring a recompilation. Java applications are often compiled to bytecode, which may execute on any Java virtual machine (JVM), regardless of the computer architecture. Java's syntax is comparable to those of C and C++, but it offers fewer low-level features. Traditional compiled languages lack dynamic capabilities (such as reflection and runtime code modification), which the Java runtime enables. According to GitHub, as of 2019, Java was one of the most popular programming languages in use, especially for client-server web applications, with 9 million developers client-server.

# **III/ USE CASE DIAGRAM**

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

**  
Figure 1 . Usecase Diagram**

# 

# 

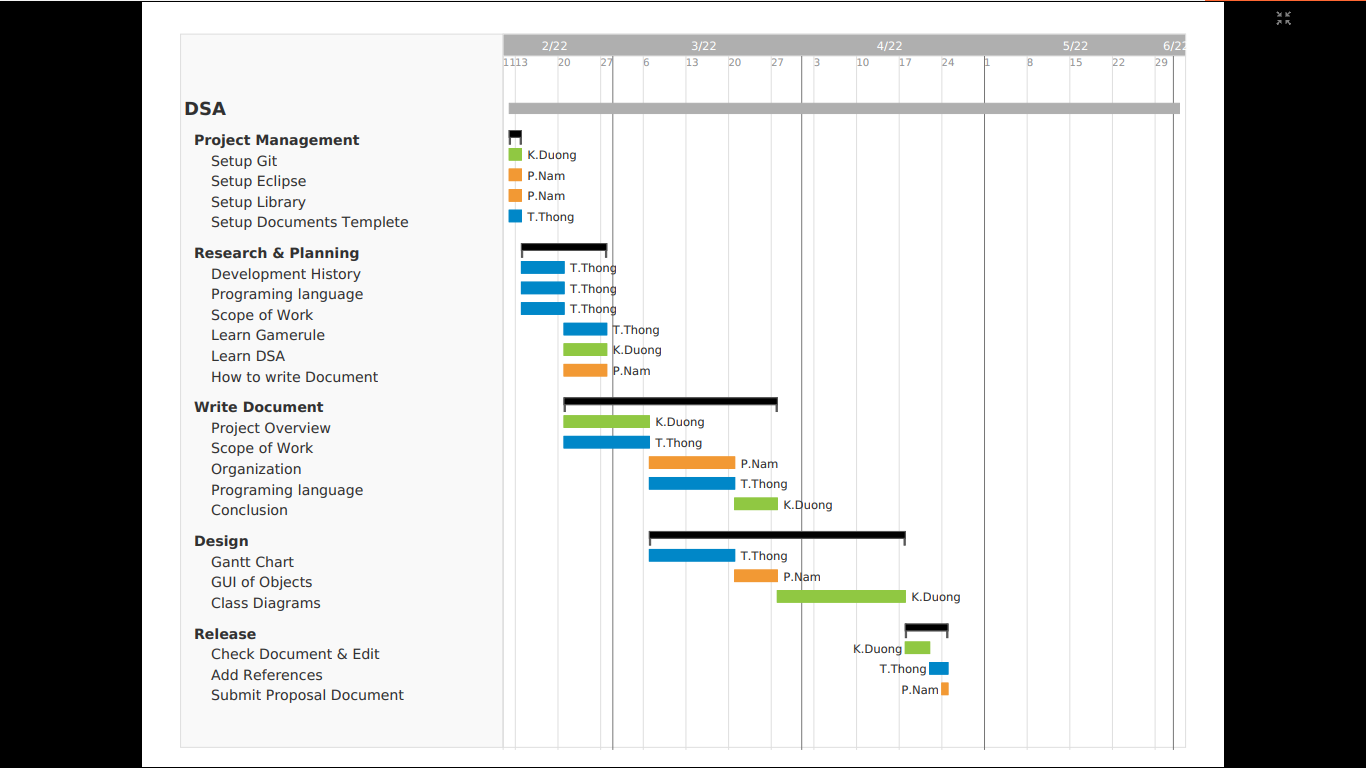
# 

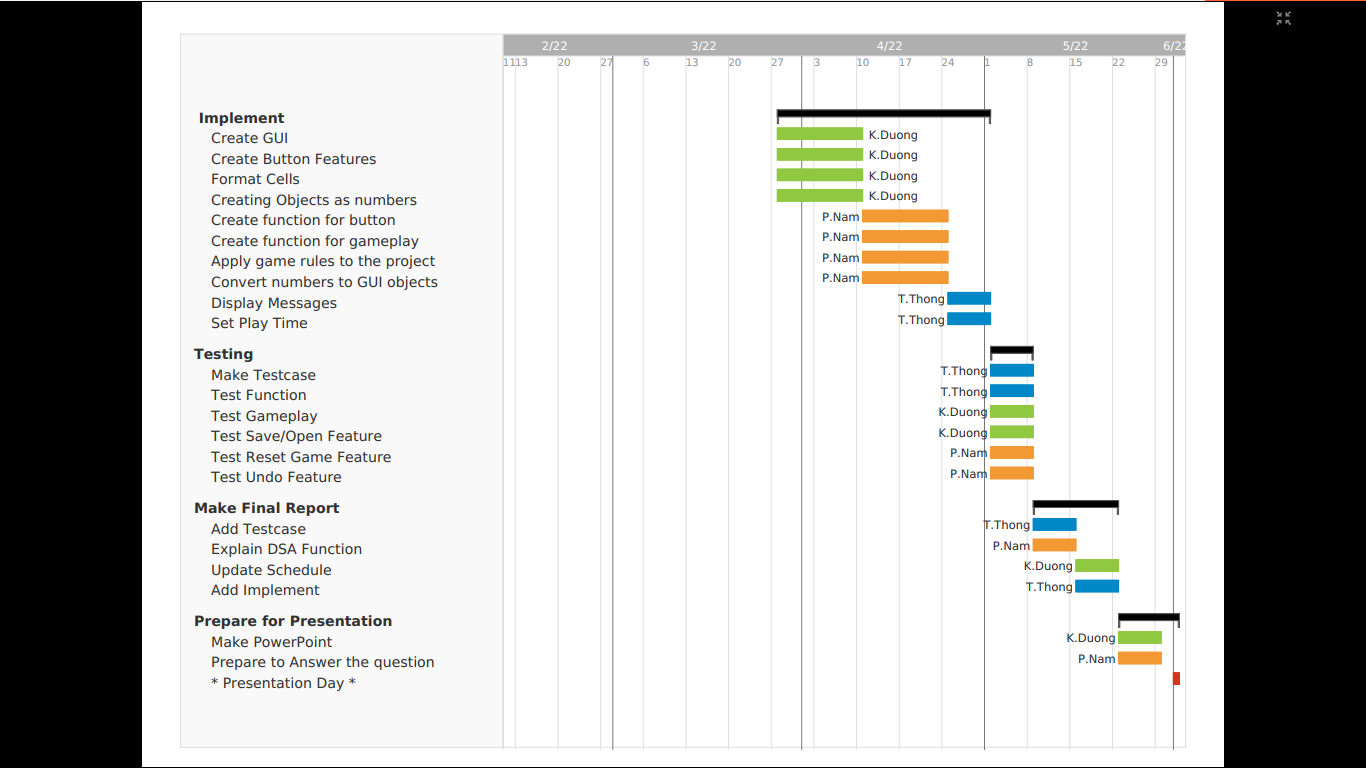
# 

# 

# **IV/ ORGANIZATION SCHEDULE**

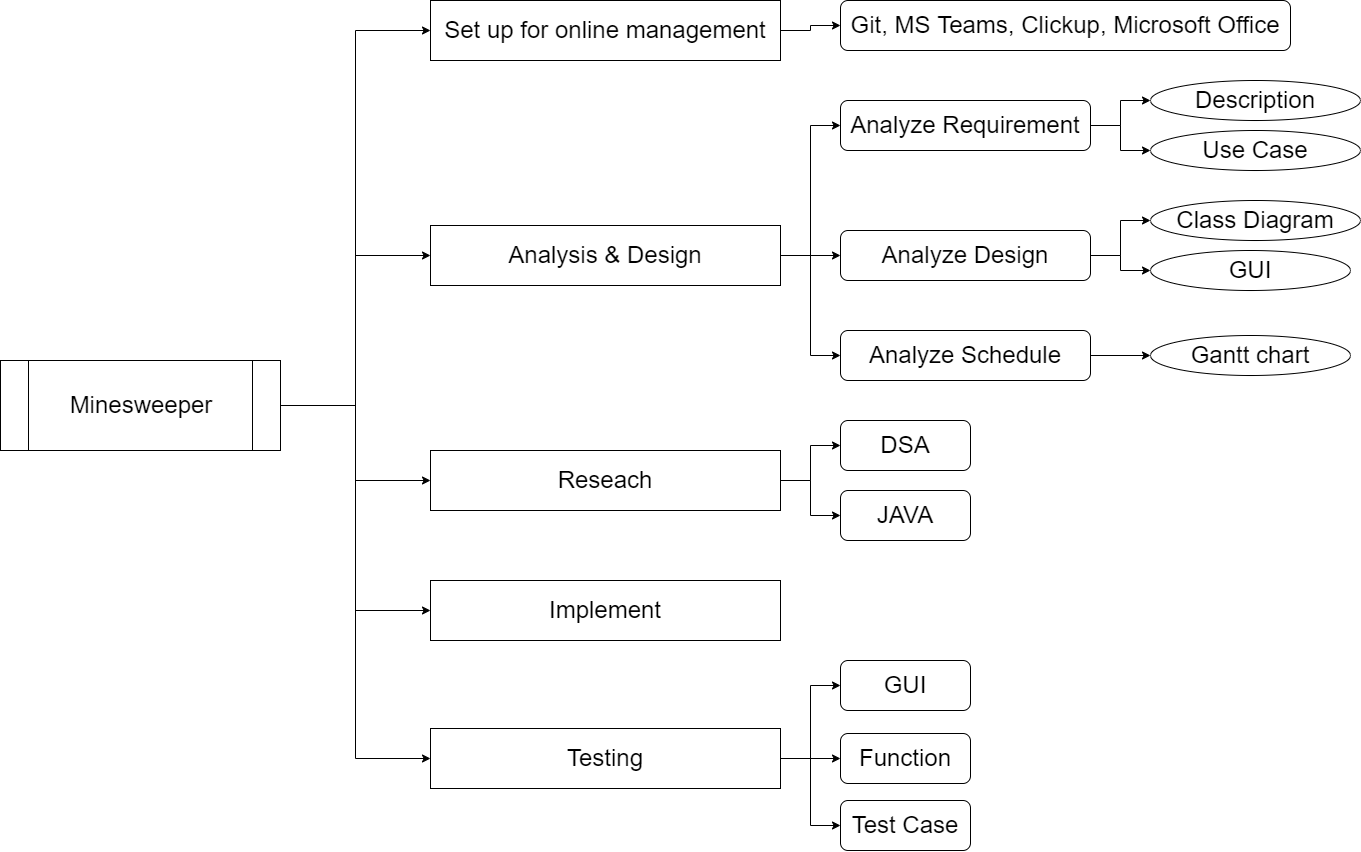
## **Gantt Chart**





### **Figure 2 . Gantt Chart**

## **Work Breakdown Structure**

******

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

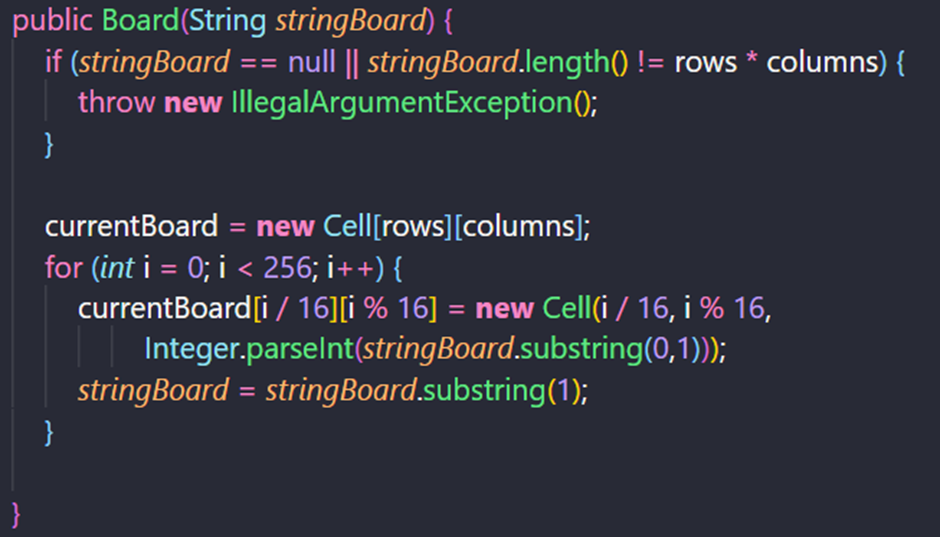
## **Schedule and milestone**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Project / Epic / Task** | **Type** | **Assignee** | **Priority** | **Status** | **Start** | **Finish** | **Days** |
| ExamProject | Project |  |  |  | 11/2/2022 | 1/6/2022 | 110d |
| ProjectManagement | Epic |  | High | Done | 12/2/2022 | 13/2/2022 | 1 d |
| Setup Git | Story | Dương | High | Done | 12/2/2022 | 13/2/2022 | 1 d |
| Setup Eclipse | Story | Nam | High | Done | 12/2/2022 | 13/2/2022 | 1 d |
| Setup Library | Story | Nam | High | Done | 12/2/2022 | 13/2/2022 | 1 d |
| Setup Document Template | Story | Thông | High | Done | 12/2/2022 | 13/2/2022 | 1 d |
| Research & Planning | Epic |  | Medium | Done | 14/2/2022 | 7/3/2022 | 21d |
| Development History | Story | Thông | Low | Done | 14/2/2022 | 20/2/2022 | 6 d |
| Programing language | Story | Thông | Low | Done | 14/2/2022 | 20/2/2022 | 6 d |
| Scope of Work | Story | Thông | Medium | Done | 14/2/2022 | 20/2/2022 | 6 d |
| Learn Gamerule | Story | Thông | Medium | Done | 21/2/2022 | 27/2/2022 | 6 d |
| Learn DSA | Story | Dương | High | Done | 21/2/2022 | 27/2/2022 | 6 d |
| How to write Document | Story | Nam | Medium | Done | 21/2/2022 | 27/2/2022 | 6 d |
| Write Document | Epic |  | Medium | Done | 21/2/2022 | 27/3/2022 | 34d |
| Project Overview | Subtask | Dương | Medium | Done | 21/2/2022 | 6/3/2022 | 13d |
| Scope of Work | Subtask | Thông | Medium | Done | 21/2/2022 | 6/3/2022 | 13d |
| Organization | Subtask | Nam | Medium | Done | 7/3/2022 | 20/3/2022 | 13d |
| Programing language | Subtask | Thông | Medium | Done | 7/3/2022 | 20/3/2022 | 13d |
| Conclusion | Subtask | Dương | Medium | Done | 21/3/2022 | 27/3/2022 | 6d |
| Design | Epic |  | Medium | Done | 7/3/2022 | 17/4/2022 | 41d |
| Gantt Chart | Story | Thông | Medium | Done | 7/3/2022 | 20/3/2022 | 13d |
| GUI of Objects | Story | Nam | Medium | Done | 21/3/2022 | 27/3/2022 | 6 d |
| Class Diagram | Story | Dương | Medium | Done | 28/3/2022 | 17/4/2022 | 20d |
| Usecase diagram | Story | Nam | Low | Done | 28/3/2022 | 3/4/2022 | 6d |
| Workbench diagram | Story | Dương | Low | Done | 28/3/2022 | 3/4/2022 | 6d |
| Release | Epic |  | Low | Done | 18/4/2022 | 24/4/2022 | 6d |
| Check Document & Edit | Story | Dương | Low | Done | 18/4/2022 | 21/4/2022 | 3 d |
| Add References | Story | Thông | Low | Done | 22/4/2022 | 24/4/2022 | 2 d |
| Submit Proposal Document | Story | Nam | Low | Done | 24/4/2022 | 24/4/2022 | 0 d |
| Implement | Epic |  | High | Done | 28/3/2022 | 1/5/2022 | 34d |
| Create GUI | Story | Dương | High | Done | 28/3/2022 | 10/4/2022 | 13d |
| Create Button Features | Story | Dương | High | Done | 28/3/2022 | 10/4/2022 | 13d |
| Format Cells | Story | Dương | High | Done | 28/3/2022 | 10/4/2022 | 13d |
| Creating Objects as numbers | Story | Dương | High | Done | 28/3/2022 | 10/4/2022 | 13d |
| Create function for button | Story | Nam | High | Done | 11/4/2022 | 24/4/2022 | 13d |
| Create function for gameplay | Story | Nam | High | Done | 11/4/2022 | 24/4/2022 | 13d |
| Apply game rules to the project | Story | Nam | High | Done | 11/4/2022 | 24/4/2022 | 13d |
| Convert numbers to GUI objects | Story | Nam | Medium | Done | 11/4/2022 | 24/4/2022 | 13d |
| Display Messages | Story | Thông | Low | Done | 25/4/2022 | 1/5/2022 | 6 d |
| Set Play Time | Story | Thông | Low | Done | 25/4/2022 | 1/5/2022 | 6 d |
| Testing | Epic |  | High | Done | 2/5/2022 | 8/5/2022 | 6 d |
| Make Testcase | Story | Thông | High | Done | 2/5/2022 | 8/5/2022 | 6 d |
| Test Function | Story | Thông | High | Done | 2/5/2022 | 8/5/2022 | 6 d |
| Test Gameplay | Story | Dương | High | Done | 2/5/2022 | 8/5/2022 | 6 d |
| Test Save/Open Feature | Story | Dương | Medium | Done | 2/5/2022 | 8/5/2022 | 6 d |
| Test Reset Game Feature | Story | Nam | Medium | Done | 2/5/2022 | 8/5/2022 | 6 d |
| Test Undo Feature | Story | Nam | High | Done | 2/5/2022 | 8/5/2022 | 6 d |
| Make Final Report | Epic |  | Low | Done | 9/5/2022 | 22/5/2022 | 13d |
| Add Testcase | Story | Thông | Low | Done | 9/5/2022 | 15/5/2022 | 6 d |
| Explain DSA Function | Story | Nam | Medium | Done | 9/5/2022 | 15/5/2022 | 6 d |
| Update Schedule | Story | Dương | Low | Done | 16/5/2022 | 22/5/2022 | 6 d |
| Add Implement | Story | Thông | Low | Done | 16/5/2022 | 22/5/2022 | 6 d |
| Prepare for Presentation | Epic |  | Low | Done | 23/5/2022 | 29/5/2022 | 6 d |
| Make PowerPoint | Story | Dương | Low | Done | 23/5/2022 | 29/5/2022 | 6 d |
| Prepare to Answer the question | Story | Nam | Low | Done | 23/5/2022 | 29/5/2022 | 6 d |
| Presentation | Epic |  | High | To Do | 1/6/2022 | 1/6/2022 | 0 d |

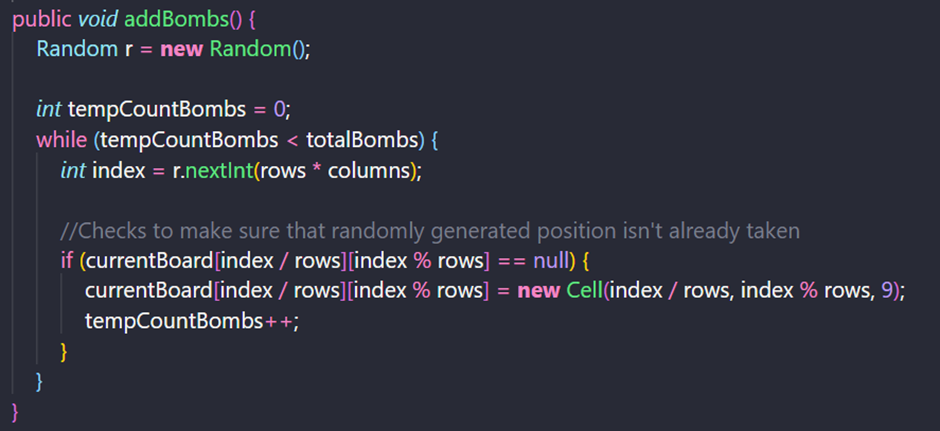
### ***Table 1 .* Schedule and milestone**

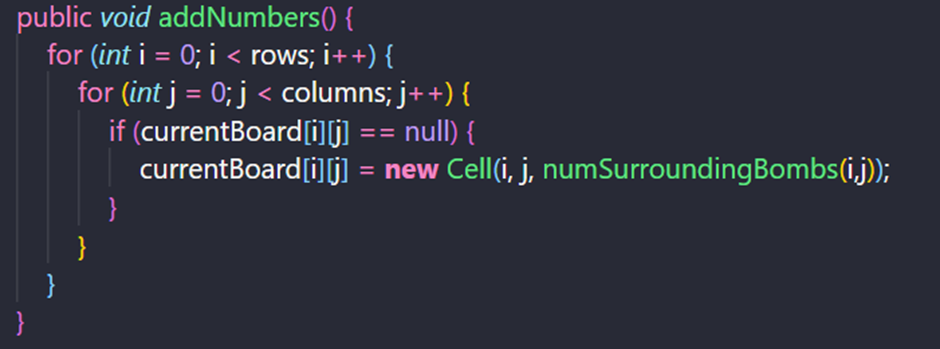
# **V/ RELATED TO DSA**

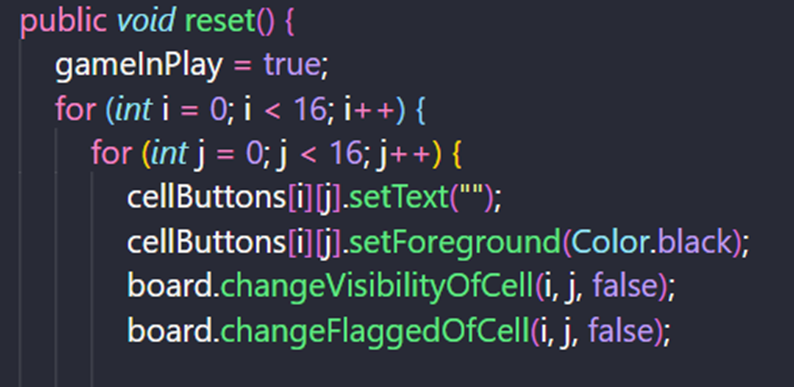
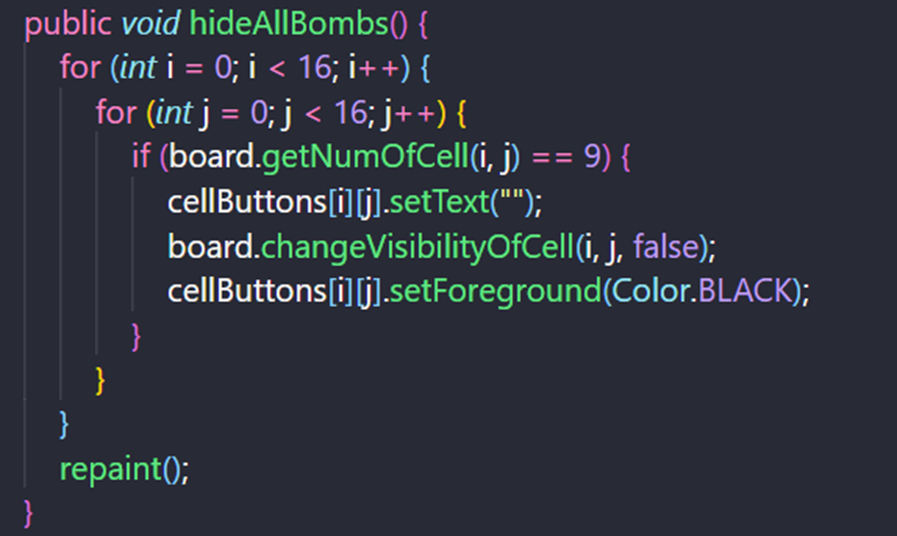
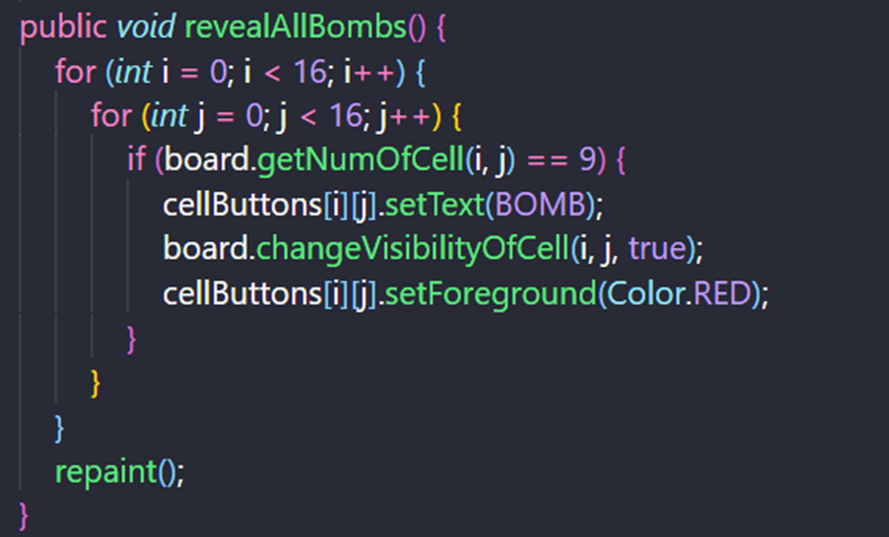
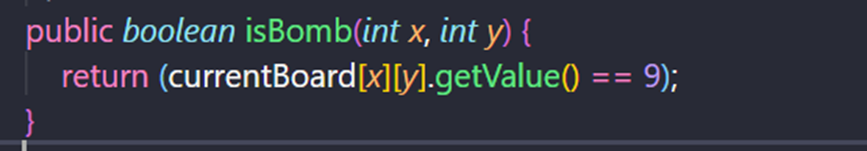
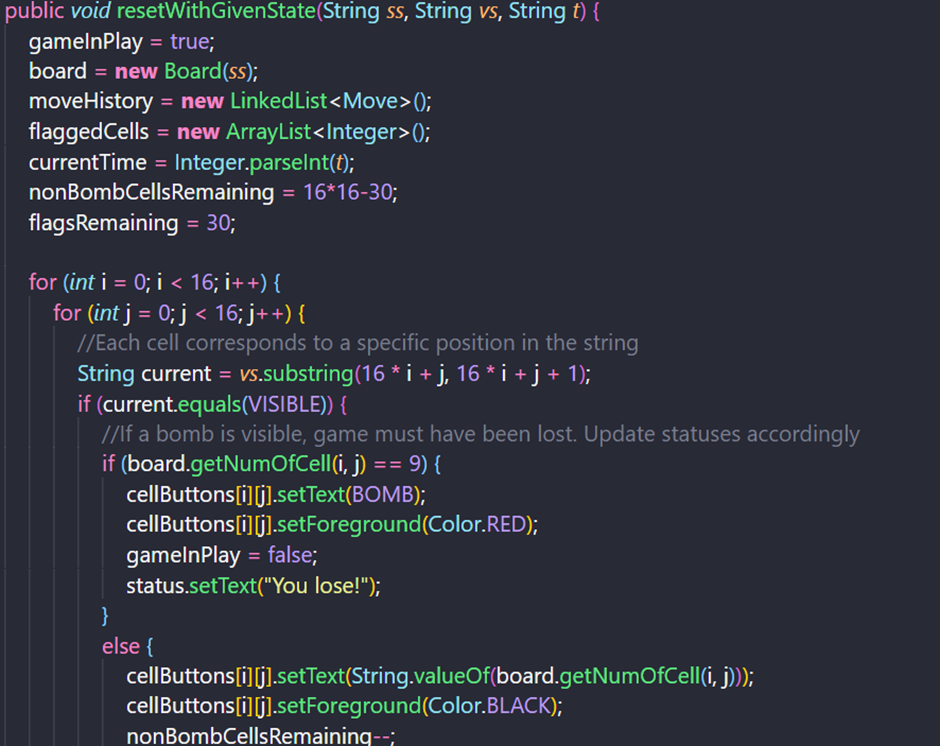
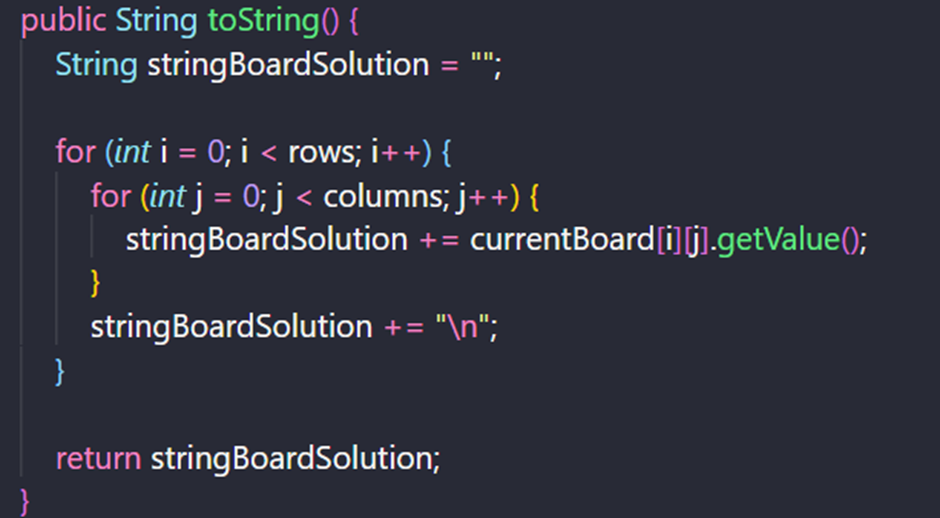
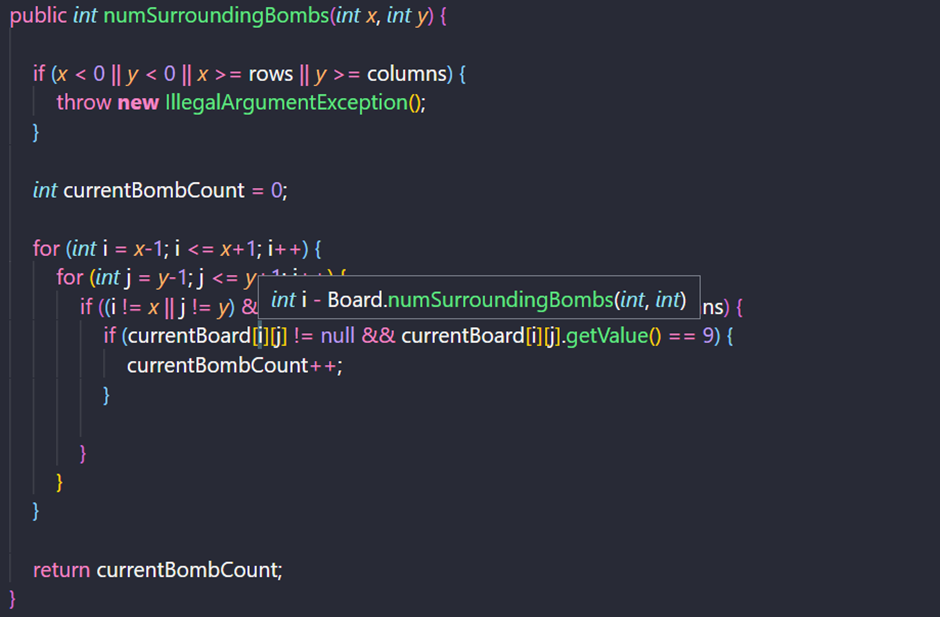
1. Overloaded constructor. Takes in a string representation of a board state

****

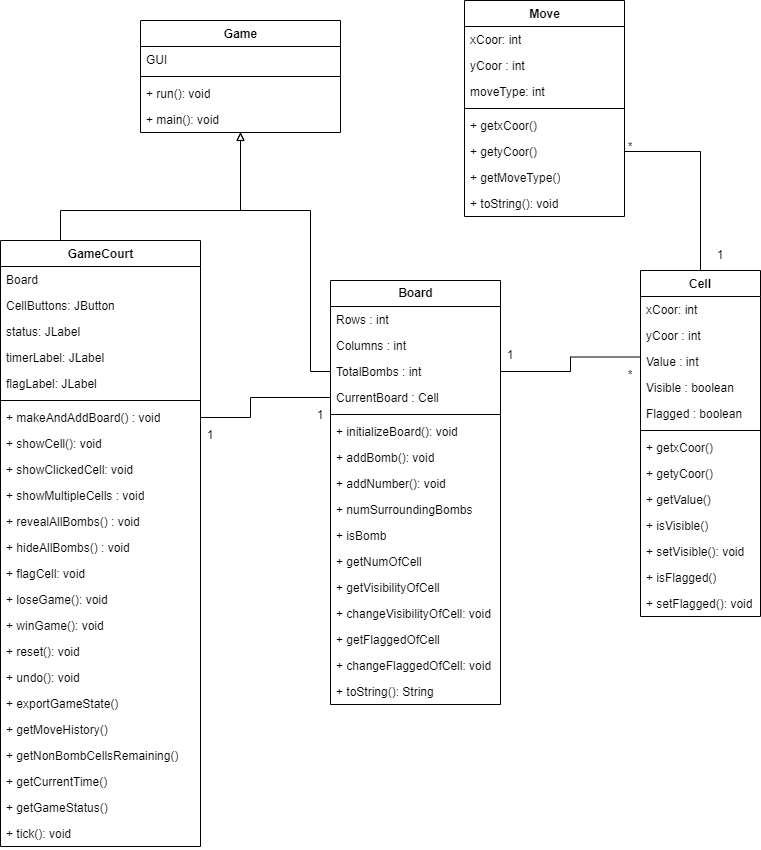
1. Uses a random number generator to determine placements of bombs.

****

**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**

**Figure 4 . ClassDiagram**

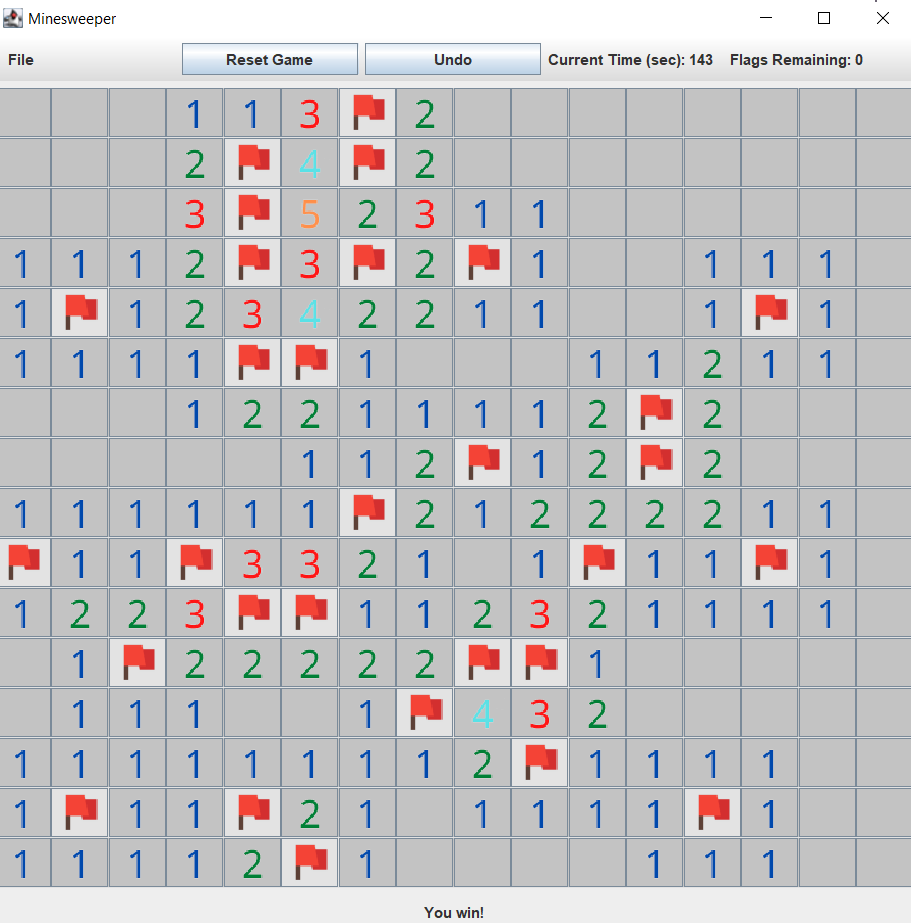
# **VII/ IMPLEMENTATION**

## **Gameplay Function**

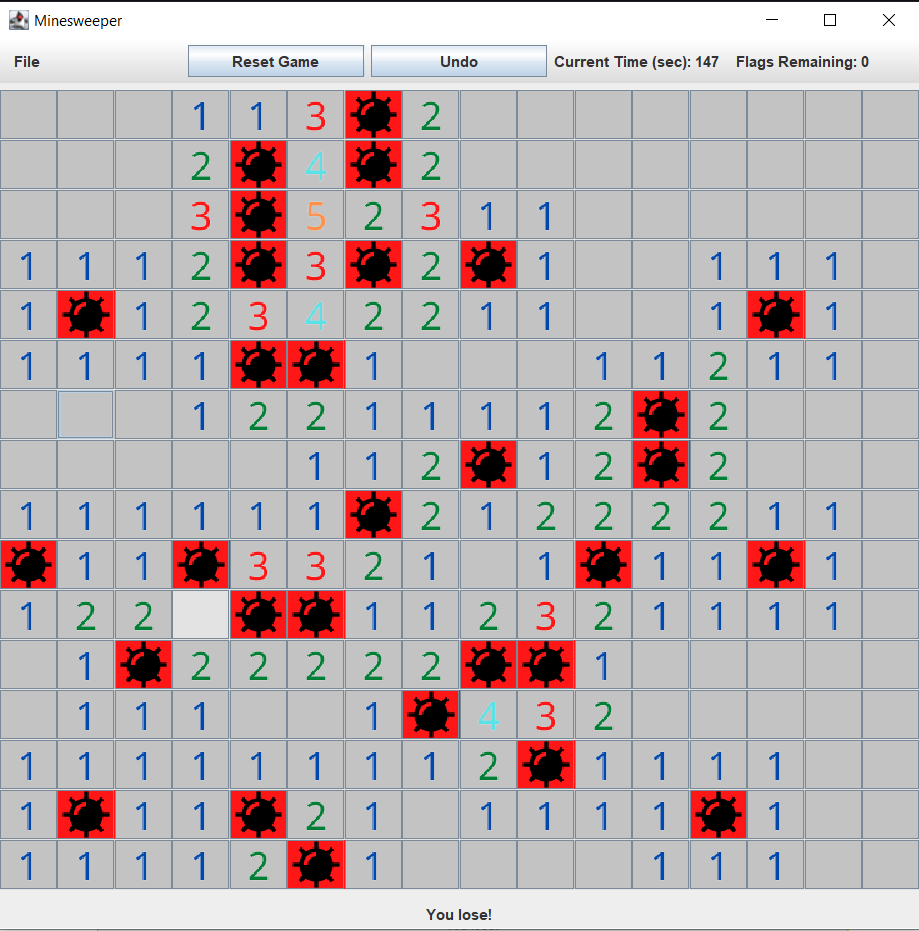
When the user runs the code, the game interface will default that you have started the game and the watch will calculate how long it took the user to complete a game turn.

****

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.

****

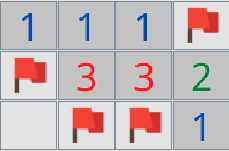
## **Number Function**

**People left-click on any box in the game interface to appear numbers locating the location of bombs.**

****

## **Flag Function**

**The user right-clicks to create a flag marking the location of the bombs.**

****

**Here we leave the default 30 flags equivalent to 30 bombs hidden in the game.**

****

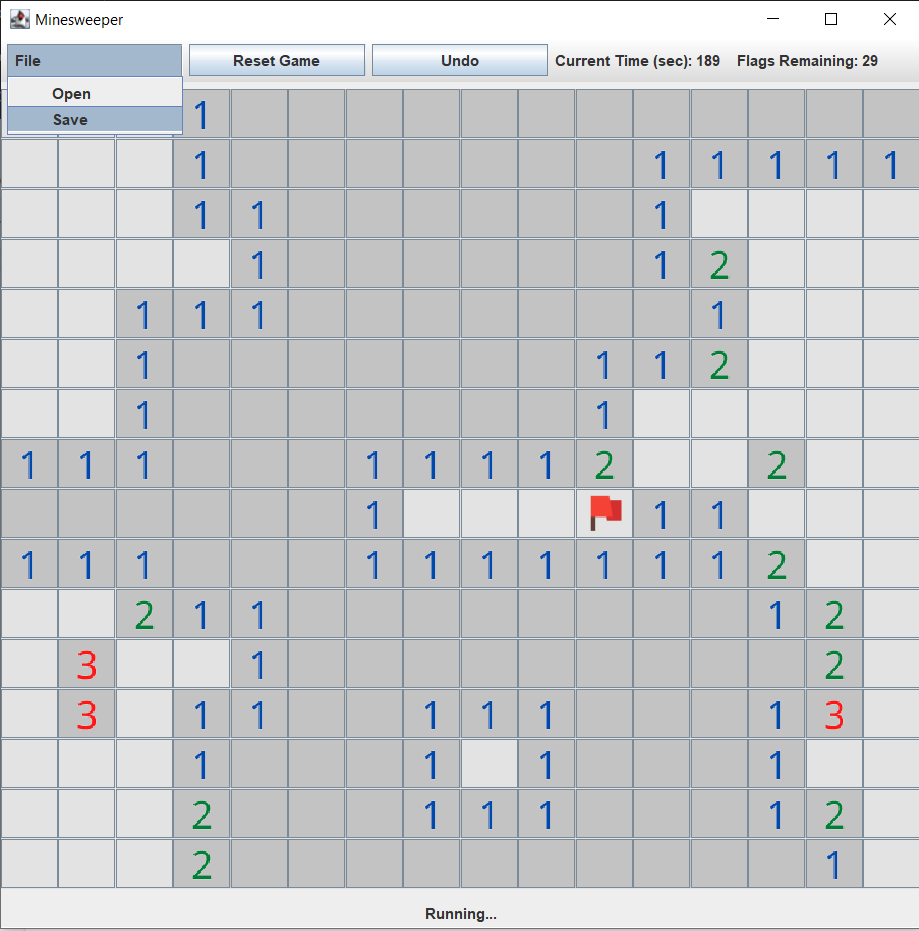
## **Reset Game Function**

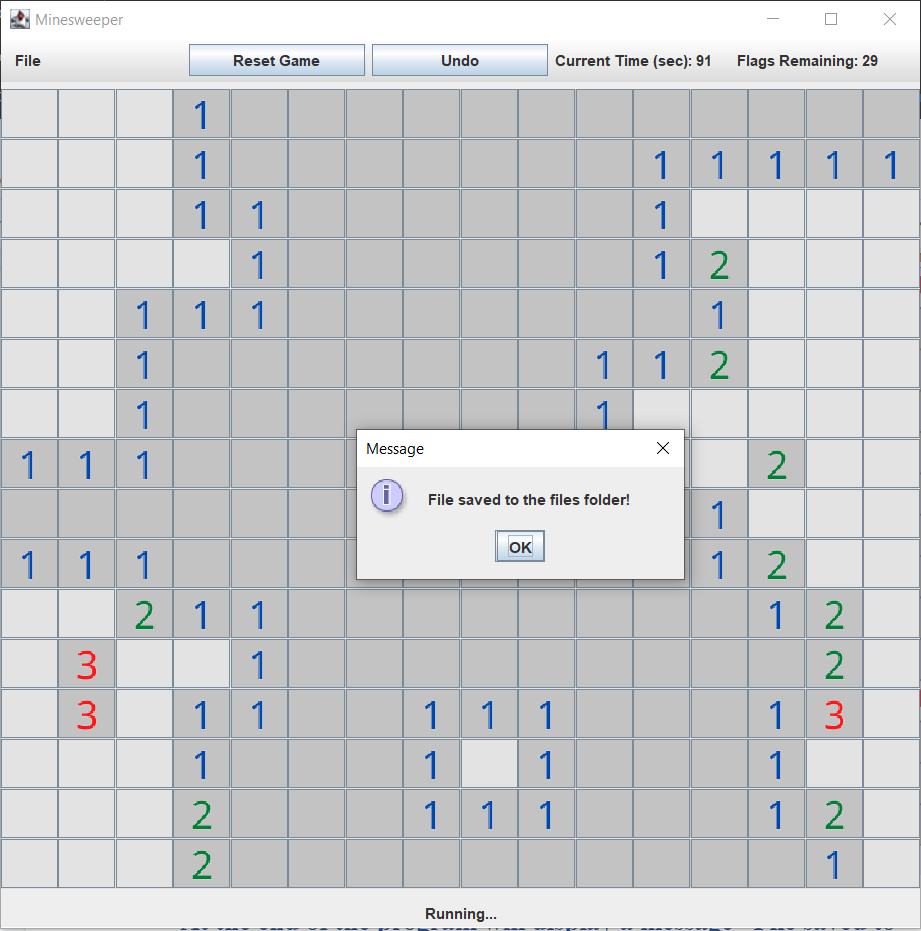
**When the user wants to play a new game again, they can press the "Reset Game" button to return to the game start interface.**

****

## **Save Game Function**

**When the user wants to save the data of the unfinished game for the next time to play, he can click the "Save" button in the “File” placed on the game control bar.**

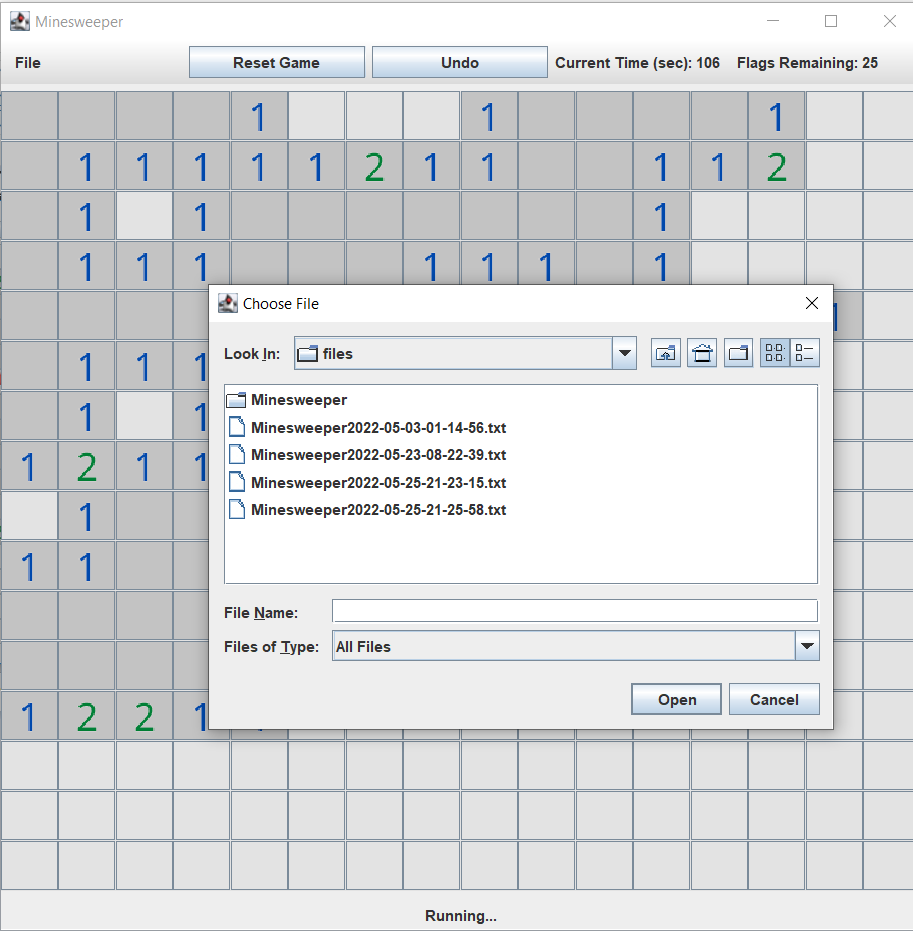
****

****

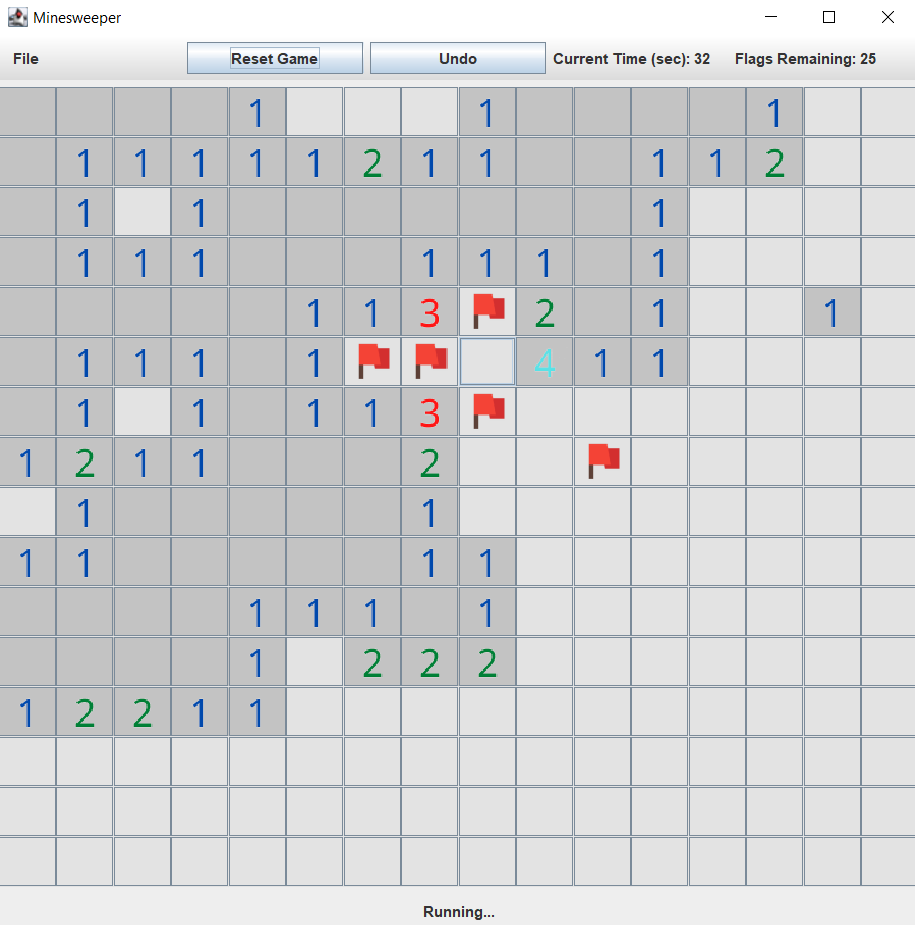
**At the end of the program will display a message “File saved to files folder” and the txt file will auto.**

## **Open Existing Game Function**

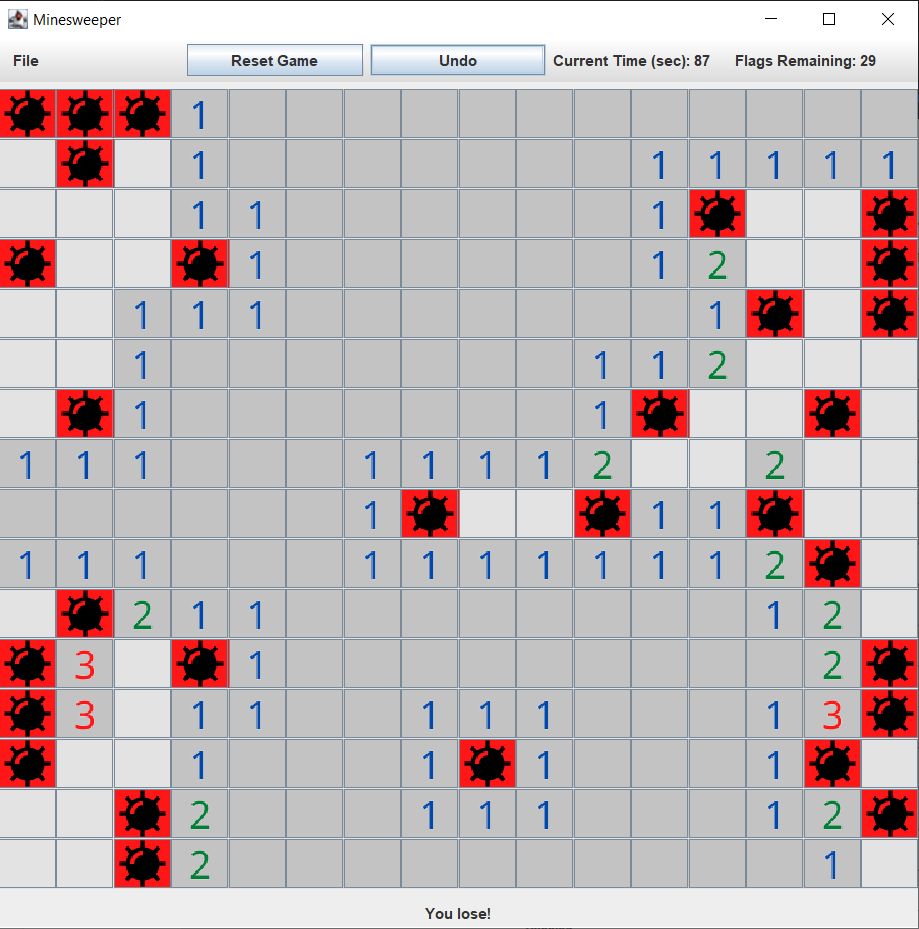
**When the user wants to open previously saved data to continue the unfinished game, he can press the "Open" button in the “File” placed on the game control bar.**

****

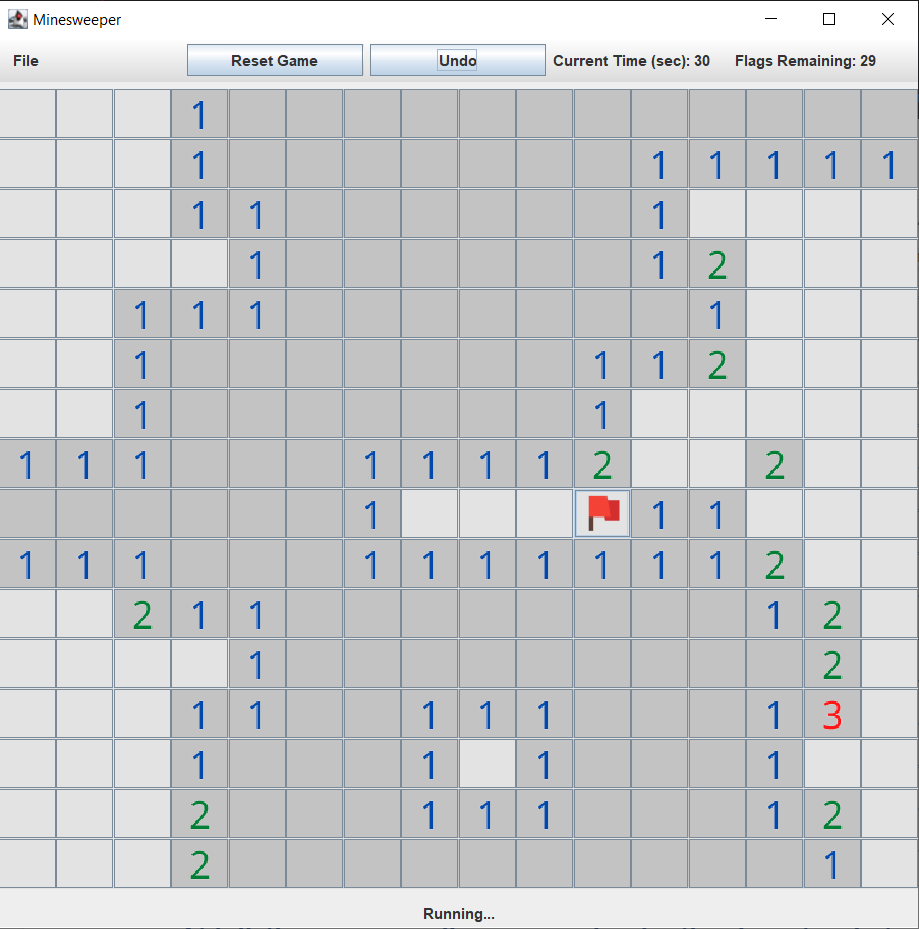
**At full, the program will open a window for the player to select the address where the data file is stored. If the selected file is valid, the game interface will immediately go to the previously saved game and here the user can continue his game.**

****

## **Undo Function**

****

**When a user wants to repeat their previous moves such as marking a flag or accidentally blowing a boom and doesn't want to end the game in a loss, they can press the "undo" button to go back to one any moves.**

****

# **VIII/ TESTCASE**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Scenario** | Undo function | | |
| **Test ID** | TC-1 | **Test Case Name** | Undo when having no previous moves left |
| **Designed by** | Dương | **Design Date** | 03/05/2022 |
| **Executed by** | Thông | **Execution Date** | 6/05/2022 |
| **Pre-Conditions** | 1. The player opens and runs the game successfully.  2. The “Undo” button shows on the control bar and works properly | **Post-Condition(s)** | 1. No errors when the game running.  2. No previous move left. |
| **Test Data/Action** | Click the “Undo” button | | |
| **Test Step** | **Expected Result** | **Comments** | **Pass/Fail** |
| 1. Users access the Minesweeper game application  2. Users click the “Undo” button on the game control bar.  3. Users click the “Ok” button to continue playing the game. | 1. The system displays New Game Interface  2. The system gives a message box: "No more moves to undo!" with the “Ok” button  3. The system is back to Game Interface. |  | PASS |

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Scenario** | Undo function | | |
| **Test ID** | TC-2 | **Test Case Name** | Undo when having any previous moves |
| **Designed by** | Dương | **Design Date** | 03/05/2022 |
| **Executed by** | Thông | **Execution Date** | 6/05/2022 |
| **Pre-Conditions** | 1. The player opens and runs the game successfully.  2. The “Undo” button shows on the control bar and works properly | **Post-Condition(s)** | 1. No errors when the game running.  2. Having any previous moves. |
| **Test Data/Action** | Play the game and click “Undo” | | |
| **Test Step** | **Expected Result** | **Comments** | **Pass/Fail** |
| 1. Users access the Minesweeper game application  2. Users click on any Cells on Game Board.  3. Users click the “Undo” button on the game control | 1. The system displays New Game Interface  2. Display any game function  3. The system rolled back to 1 previous moveset. |  | PASS |

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Scenario** | Save Game function | | |
| **Test ID** | TC-3 | **Test Case Name** | Save recent game results to file |
| **Designed by** | Nam | **Design Date** | 05/05/2022 |
| **Executed by** | Nam | **Execution Date** | 09/05/2022 |
| **Pre-Conditions** | 1. The player opens and runs the game successfully.  2. The “Save” button shows on the control bar and works properly. | **Post-Condition(s)** | 1. No errors when the game running. |
| **Test Data/Action** | Click the “Save” button | | |
| **Test Step** | **Expected Result** | **Comments** | **Pass/Fail** |
| 1. Users access the Minesweeper game application  2. Users click on any Cells.  3. Users click the “Save” button  4. Users click the “Ok” button to continue playing the game. | 1. The system displays New Game Interface  2. The systems display any Game Function  3. The systems display a message box: “File saved to the files folder!” with the “Ok” button and the recent game result saved to the “files” folder.  4. The system is back to Game Interface. |  | PASS |

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Scenario** | Reset Game function | | |
| **Test ID** | TC-4 | **Test Case Name** | Reset to a new game |
| **Designed by** | Nam | **Design Date** | 05/05/2022 |
| **Executed by** | Nam | **Execution Date** | 9/05/2022 |
| **Pre-Conditions** | 1. The player opens and runs the game successfully.  2. The “Reset Game” button shows on the control bar and works properly. | **Post-Condition(s)** | 1. No errors when the game running. |
| **Test Data/Action** | Click the “Reset Game” button | | |
| **Test Step** | **Expected Result** | **Comments** | **Pass/Fail** |
| 1. Users access the Minesweeper game application  2. Users click on any Cells.  3. Users click the “Reset Game” button | 1. The system displays New Game Interface  2. The systems display any Game Function  3. The system cleans all data, back to New Game Interface, and resets the “Current time” to “0s”. |  | PASS |

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Scenario** | Open Existing Game function | | |
| **Test ID** | TC-5 | **Test Case Name** | Open the valid file saved |
| **Designed by** | Dương | **Design Date** | 05/05/2022 |
| **Executed by** | Nam | **Execution Date** | 10/05/2022 |
| **Pre-Conditions** | 1. The player opens and runs the game successfully.  2. The “Open” button shows on the control bar and works properly | **Post-Condition(s)** | 1. No errors when the game running.  2. The Open File Dialog Form “Choose file” works properly |
| **Test Data/Action** | Click Open and Choose file: “Minesweeper2022-05-21-22-53-10.txt” | | |
| **Test Step** | **Expected Result** | **Comments** | **Pass/Fail** |
| 1. Users access to the Minesweeper game application  2. Users click the “Open” button on the game control bar.  3. Users choose the valid “.txt” file directory and click the “Open” button. | 1. The system displays New Game Interface  2. The system displays Open File Dialog Form “Choose file”.  3. The system displays the saved game data with the saved current time. |  | PASS |

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Scenario** | Open Existing Game function | | |
| **Test ID** | TC-6 | **Test Case Name** | Open the invalid file saved |
| **Designed by** | Thông | **Design Date** | 05/05/2022 |
| **Executed by** | Dương | **Execution Date** | 10/05/2022 |
| **Pre-Conditions** | 1. The player opens and runs the game successfully.  2. The “Open” button shows on the control bar and works properly | **Post-Condition(s)** | 1. No errors when the game running.  2. The Open File Dialog Form “Choose file” works properly |
| **Test Data/Action** | Click Open and Choose file: “xxxxx” | | |
| **Test Step** | **Expected Result** | **Comments** | **Pass/Fail** |
| 1. Users access to the Minesweeper game application  2. Users click the “Open” button on the game control bar.  3. Users choose the invalid file directory and click the “Open” button.  4. Users click the “Ok” button to continue playing the unsaved game. | 1. The system displays New Game Interface  2. The system displays the Open File Dialog Form “Choose file”.  3. The system displays a message box: “Invalid File!” with an “Ok” button.  4. The system is back to the unsaved Game Interface. |  | PASS |

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Scenario** | Flag function | | |
| **Test ID** | TC-7 | **Test Case Name** | Flag with right-click |
| **Designed by** | Dương | **Design Date** | 03/05/2022 |
| **Executed by** | Nam | **Execution Date** | 6/05/2022 |
| **Pre-Conditions** | 1. The player opens and runs the game successfully. | **Post-Condition(s)** | 1. No errors when the game running.  2. The Flag GUI works properly |
| **Test Data/Action** | Right-click | | |
| **Test Step** | **Expected Result** | **Comments** | **Pass/Fail** |
| 1. Users access to the Minesweeper game application  2. Users Right-click on any Cells | 1. The system displays New Game Interface  2. The system displays the flag GUI and the Flags Remaining minus to 1 (ex: 30 to 29) |  | PASS |

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Scenario** | Number function | | |
| **Test ID** | TC-8 | **Test Case Name** | Number surrounding the bomb |
| **Designed by** | Dương | **Design Date** | 03/05/2022 |
| **Executed by** | Nam | **Execution Date** | 6/05/2022 |
| **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. Users access to the Minesweeper game application  2. Users left-click on any Cells | 1. The system displays New Game Interface  2. The system displays the number of GUI surrounding the bomb. |  | PASS |

### **Table 9. Test case 8**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Scenario** | Gameplay function | | |
| **Test ID** | TC-9 | **Test Case Name** | Successfully Win Game |
| **Designed by** | Thông | **Design Date** | 03/05/2022 |
| **Executed by** | Nam | **Execution Date** | 8/05/2022 |
| **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. Users access to the Minesweeper game application  2. Users click on any Cells  3. Users complete truly the Last Cell | 1. The system displays New Game Interface  2. The system displays any GUIs.  3. The system displays the message “You win!” |  | PASS |

### **Table 10. Test case 9**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Scenario** | Gameplay function | | |
| **Test ID** | TC-10 | **Test Case Name** | Touch the Bomb |
| **Designed by** | Thông | **Design Date** | 03/05/2022 |
| **Executed by** | Nam | **Execution Date** | 8/05/2022 |
| **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. Users access to the Minesweeper game application  2. Users left-click on the bomb Cell | 1. The system displays New Game Interface  2. The system displays the bomb GUIs displays a message “You lose!” |  | PASS |

### **Table 11. Test case 10**

# **IX/ GLOSSARY**

## **References**

[*http://web.cse.ohio-state.edu/~bair.41/616/Project/Example\_Document/Req\_Doc\_Example.html#Func*](http://web.cse.ohio-state.edu/~bair.41/616/Project/Example_Document/Req_Doc_Example.html?fbclid=IwAR3shSt7bw0Gq3FhsTUsFZwsULrvM4haGIgHItFDcNujg0rZEHzMe1zDXfQ#Func)

[*https://en.wikipedia.org/wiki/Java\_(programming\_language)*](https://en.wikipedia.org/wiki/Java_(programming_language))

[*https://www.geeksforgeeks.org/data-structures/*](https://www.geeksforgeeks.org/data-structures/)

[*https://github.com/maplepold/Minesweeper*](https://github.com/maplepold/Minesweeper)

[*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!)

[*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.)

[*https://code.tutsplus.com/tutorials/build-a-minesweeper-game-within-200-lines-of-code--active-8578*](https://code.tutsplus.com/tutorials/build-a-minesweeper-game-within-200-lines-of-code--active-8578)