**Members:**

**Nguyễn Anh Tuấn – ITITIU19061**

**Phạm Hồng Đăng – ITITIU19009**

**Nguyễn Gia Phúc – ITITIU19041**

**Final Project: Minesweeper**

Minesweeper is a popular one- player computer game written by Robert Donner and Curt Johnson which was included in Microsoft Windows in 1991. Playing the game requires logical, arithmetic and probabilistic reasoning based on spatial relationships on the board.

1. **The Basic Game**

The minesweeper is a single-player puzzle video game. At the beginning of the game, a 2D grid of identically looking tiles is presented to the player. The number of aimed tiles is known to the player and it determines the difficulty of the games. The goal of the game is to uncover all the tiles which do not contain a mine. Each turn the player can select one of three actions: to mark a tile as a mine, to unmark a tile, and to uncover the tile If a tile hiding a mine is revealed , the player will lose the game. If all the tiles not containing a mine are revealed, the player will win.

Graphical user interface, text, application, chat or text message

Description automatically generatedA picture containing graphical user interface

Description automatically generated

Chart

Description automatically generated

1. **Game Console**

|  |  |
| --- | --- |
| **GAME** | **MINESWEEPER** |
| Basic Features | * Ability to play and replay game. * Randomly generated mines. * Notifices player the number of mines adjacent to a block once it has been selected. * Time counter. |
| Additional  Features | * Buzzer notification when the player wins or loses by pressing the block containing mine. * Create flags to mark doubtful blocks. * Undo features. |

1. **Variation and Offshort**

The game itself opens to many possibilities and, indeed, many variations of it have been already implemented. Here below is a list of different versions of the game.

1. A first alternative is to play not on a rectangular field. Any shape can be used, but tiles are always square.
2. A similar idea is to change the shape of the tile. Triangles or hexagons are the most common alternative shapes. This reduces the maximum number of adjacent tiles.
3. A more challenging option is to allow more mines to be under the same tile. Even if in principle not needed, usually a maximum number of mines allowed to stay together is fixed. Moreover, the player is commonly told about how many tiles contain one mine, how many contain two, and so on.
4. Another rule that can be changed is the number of dimensions of the grid. There exists a 3D version of Minesweeper, which implies the possibility to have up to 26 adjacent mines per tile. Of course, the idea of having to safely walk on a mined field is a bit lost, but you could have to swim underwater in a dangerous sea with invisible poison localised here and there. . .
5. **Implementing the game.**

* The first part of this project is to implement a basic version of the basic game, which is played within the terminal. The idea is to get the grid displayed and to enter the coordinate of the tile we want to reveal. If we neither win nor lose, we get the grid displayed again and we are asked again for new coordinates.
* Next, we built a GUI class to handle functions of the game. In GUI, we drew an user interface with 16x16 blocks and used two-dimension arrays for storing information of a blocks about mine, neighbours, flagged and revealed.

Text

Description automatically generated

* We created a class Random to generate the mines randomly. In this class, two for loops were used to check all of the blocks. Each block has i and j which represent the position of the block. When we have i and j, we use a random function to take a number from 0-100. If this number is less than 15, the mines will be placed in this block. Hence, The number of mines is not the same for each game. And, in this class, we will count the number of mines in 8 adjacent-blocks.

Text

Description automatically generated

Text

Description automatically generated with medium confidence

* Next, when a player clicks a block on the board, the coordinate of it will be returned. It will be difficult for us to handle some features and store information. Hence our task is converting the coordinate to [i][j] which represent the position of blocks in two -dimensional arrays. We created the class “Move” that implements the mouseMotionListener interface to get the coordinates when we move the mouse on the board.

Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generated

* Next task is opening the blocks when the player clicks it. There are three situations:

+ A block contains mine: all of the mines in this game will be displayed, and the player cannot click the remaining blocks on the board.

+ A block contains the number of mines in 8 adjacent-blocks: display the number of mines and set the value of revealed[i][j] = true ( this block has already been reached by the player). We cannot click this block.

Graphical user interface, text

Description automatically generated

+ An empty block: When player clicks an empty block, we will check 8 adjacent- block of this to find the other empty block, call the recursive function for this and set the value of revealed[i][j] (because a blocks can be checked many times). Open an empty block when all of the empty blocks are covered by the non-empty blocks.

Text

Description automatically generated

* One of the additional features is undo features. To handle this feature, we used two stacks to store the position of the block when the player clicks. This feature is dependent on Open class, we call it Close class. We used two dimensional arrays to identify the blocks which are opened from the empty block (closeEmptyRevealed[][]) and non-empty blocks (closeRevealed[][]). Depending on the Open method, when a player clicks the blocks, these blocks will be marked to know what process this block is opened. Similar to the open method, a recursive function is used to find the blocks which are opened from empty-block.

Text

Description automatically generated

* Finally, we will add and set-up some buttons for the game. There are three buttons for this game such as flagger, undo, reset game. Each button will have a boolean variable. If the player clicks on the button, the value of the button will be true. Then, some of the processes will be executed, and the value of the button will change into false automatically.

A picture containing text

Description automatically generated

Graphical user interface, text

Description automatically generated

In conclusion, The minesweeper is a single-player puzzle video game. This game is built by using 2D- painting, knowledge of loops, stacks, and two dimensional arrays, and some tutorials on the Internet. Our group has the basic function of this game and some additional features such as display win/lose, reset game or undo. But, this game has some ineffective solutions (ex: using 2D- array of buttons instead of painting each block) which we will deal with in the future.