**VIETNAM NATIONAL UNIVERSITY**

**INTERNATIONAL UNIVERSITY**

**🙠🙠🙠 ------ 🙢🙢🙢**



**PROJECT**

*Minesweeper*

***Instructor*:** *Trần Thanh Tùng*

***Course****: Data Structure & Algorithm*

**Team member :**

* Lưu Hiển Long - ITITIU19025
* Nguyễn Minh An - ITITIU19070
* Dương Thiên Phúc - ITITIU19184
* Nguyễn Tiến Anh - ITITIU19073

Semester 02, 2020 – 2021

**Table of Contents**

[**I. Intention**](#_1t3h5sf) 3

[**II. Introduction**](#_gjdgxs) 3

[**III. Functionalities , features , data structure and algorithms**](#_30j0zll) 3

[**IV. Materials**](#_2et92p0) 6

[**V. Contributors**](#_3dy6vkm)6

**VI. Reference and Technologies** 6

# **Intention**

The game is made so that team members can clearly understand the knowledge contained in the course, serve the purpose of completing the final project and can apply the techniques in practice.

# **Introduction**

Data Structure and Algorithm is one of the important courses in the curriculum for software engineer students. We have to choose a project topic to understand clearly about the knowledge and serve the needs of the final project, which is “ **Minesweeper** ”. The language used to make all the algorithms and functionalities, including mathematical functions is C#. We use Unity to create a UI for the player , we have implemented extra Undo/Redo functions by using Stack, a 2-D array used to indicate position. In addition, Unity also supports us to build a finished game so that users can easily access them .

# **Functionalities, features, data structure and algorithms:**

1. **Classes :**

* Resource :
* *Board :* The board is the whole gameplay. It receives input of the coordination of x and y, connects the IndexBoard and StateBoard (explained below), processes the logic and then displays the output into the screen. When the game is imported to Unity, this class is separated into 2 classes, UIBoard to receive input and IndexStateConnector as the connector of index and state boards.
* *IndexBoard :* Index board contains the index values of the board (the numbers 1,2,3,4,..., having mines or not having mines)
* *StateBoard :* State Board contains the state of each cell of the board (revealed or not revealed)
* Unity :
* *CameraController :* Allow the player to move the camera, zoom in and zoom out. It also performs automatic camera placement, such as setting the camera on the middle cell of the board at each start of the game.
* *IndexStateConnecter :* It contains and links together indexBoard and stateBoard.
* *MathFunc :* All the customs mathematical functions needed for the game. It works as a library, other pieces of code can not access the constructor, only the functions that are public static.
* *Timer:* Setting the time running for the gameplay ( 0 to 999 )
* *UIBoard :* Handle input of the whole board and provide the functionalities related to the flag-marking feature of the game.
* *UICell :* Handle input of each cell in the board and transfer the information back into the UIBoard to process. This class also connects the UI button to the C# code.
* *WinLoseController :* Setting conditions for the gameplay if the user wins or loses, the game will stop. It also contains the function to reset the game.

1. **Features** :

* *Input* : User can input the height and the width of the board . Moreover , users can input the number of mines that are suitable with the size of the board.
* *Cell Expansion*: When a user clicks on a concealed and empty (having no mines around it) cell, it and its surrounding cells will be revealed.
* *Flag*: Being able to mark a cell with a flag using right-click mouse and remove an existing flag using right-click mouse. If a cell is already revealed and has a flag around it, the player can target that cell to open the remaining surrounding cells (this can cause game loss if a mine is revealed).
* *Undo & Redo*: Players can undo to the previous step and redo to the following step.
* *Time Counter*: There will be a time counter to count from 0 to 999 seconds
* *Reset*: If the player wants to restart the game or if they input the new value for height , width , number of mines.
* *Camera manipulation*: the player can drag the middle mouse to move the camera, as well as scroll the mouse wheel to zoom in and zoom out.

1. **Game Design**:

* Design Pattern - Decorator Design Pattern: Unity provides its GameObjects system. In a game scene, every game object has its core component (Transform component), and other decorative components to add on top. This system shares similarities with the decorator design pattern, and our team has developed the UI using it.
* Game loop design: Our team has designed the flow of events that happens in the game. This helps the game to consume less resources, as well as to make the development of the projects get carried out much more easily. (Figure 1.)

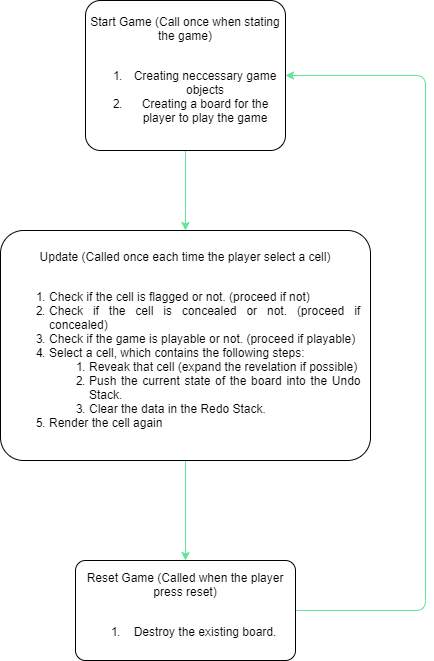


Figure 1. Game loop diagram

1. **Data Structure and algorithm**:
2. **Data structure**
   1. *2-dimensional Array*: Is used as data of the index board, state board and the connector of those 2 classes.
   2. *Stack*: Is used to store previous and following states of the board, which will be used to build the Undo or Redo features.
3. **Algorithm:**
   1. Undo and Redo: The Pop(), Push() and Peek() functions of the Undo and Redo stacks are used by the index-state connector to move back and forth between states of the board.
   2. Cell Expansion: When a concealed empty cell is revealed alongside its surrounding cells within a certain radius. If a cell (except the one originally selected) is revealed, the radius will increase and the process will repeat. This process stops when there are no more cells to open, or the radius reaches the value of the product of height and width of the board. This function is implemented to avoid the usual cursive approach, which is suspected to be more resource-consuming.

# **Materials:**

# Materials :

1. C# is the programming language.
2. Game Engine Unity is used to build the UI.
3. Online images are used for the game.
4. Researching source :
5. W3School provides basic knowledge of C#
6. Unity documentation provides overall knowledge of Unity.
7. Unity forum provides guidance for specific cases

# **Contributors**

1. **Lưu Hiển Long (** ITITIU19025 )

*Github name :* [Chillisaucery](https://github.com/Chillisaucery)

1. **Nguyễn Minh An (** ITITIU19070 )

*Github name :* minhan608

1. **Dương Thiên Phúc** ( ITITIU19184 )

*Github name :* Bill-Fashion

1. **Nguyễn Tiến Anh** ( ITITIU19073)

*Github name :* [C](https://github.com/Chillisaucery)haudeptrai123456

# **References and Technologies**

* **Programming Language:** C#
* **IDE:** VS Code, Visual Studio
* W3School: <https://www.w3schools.com/cs/>
* Game Engine Unity: <https://unity.com/>
* Unity documentation: <https://docs.unity3d.com/2021.2/Documentation/Manual/index.html>
* Unity forum: <https://forum.unity.com/>
* Diagram drawing application: <https://app.diagrams.net/>
* Github repository: <https://github.com/Chillisaucery/Minesweeper_DSA>