Algonquin College Logo

# SCHOOL OF ADVANCED TECHNOLOGY

### ICT - Applications & Programming

### Computer Engineering Technology – Computing Science

A close-up of a computer screen

Description automatically generated with low confidence

A11

Game Interface

Team:

Andrew Lorimer - Id:041056170 Solomon Thangthong – id: 041023691

Game Proposal - Battleship

|  |  |
| --- | --- |
| **Part**  **1** | **GUI Definition** |

**EXPLANATION**

*The purpose of this assignment is to define the elements of the GUI application to be used in your game implementation.*

* ***Example (Prof. suggestion)****: A screenshot of a game

  Description automatically generated with medium confidence*
* ***Note****: The professor interface is also a proposal. It means that your own implementation can be different. What does matter is that the game functionality will be respected.*
  1. **Defining the Functionalities**

**Main Behavior**

*Include the list of functionalities that you imagine to have in the game (they can be from Swing or JavaFX).*

Allow users to create a game with a computer, selecting the game dimensions before starting game. User can randomly place their ships or manually orient their ships on the board before the game begins. After all boats are placed on the grid, players take turns attacking each other by selecting a square on the grid. Audio will play indicating a hit or miss in addition to the square being colored to indicate a hit or miss**1**. The game ends when all boats for either the player or adversary are eliminated. After this the user has an option to reset the game.

**Functionalities and Behaviors**

*Check, for example, on-line examples (*[*https://www.battleshiponline.org/*](https://www.battleshiponline.org/)*):*

My version of the battleship games aims to make the GUI as simple as possible for users so that anyone could use it without a tutorial. The elements present are:

1. Two boards containing the current players board on the left and the adversary with a hidden (blank) board on the right.
2. A language selector for either French or English
3. A Game/History log displaying all plays including hits, misses and ships sunk
4. The option to randomly create a board or manually place ships at the game start, allow user to change orientation of ships.
5. Option to change dimensions of board.
6. A reset button to restart the game.
7. Play button to start the game.
8. Life indicator, showing the percentage of life left, in addition to a battleship display showing which ships are sunk and which are alive.

A screenshot of a video game

Description automatically generated

*What are the behaviors and functionalities that you will provide? How these elements are related with functionalities.*

***Example****: Answer these questions:*

* *Who are the actors (who can design / play the game)?*

The actors in this game are one player, one computer and an automatic feature. The player and computer take turns making attacks on each other’s side of the board. Alongside, both have options to randomize their board and more specifically, players can choose to design their ship layout. The player can reset games, change the language, change board dimensions and click the battleship icon to view information about the developers.

* *What are the preconditions (requirements) for some functionalities?*

The preconditions include choosing a board dimension size which determines how many ships either player will have to place on each side. By default, the dimension size is two, but the user can change the size before starting the game. Lastly, the player and machine must place all ships before the round begins.

* *And the post-conditions / results?*

Users can select squares to attack on the adversary board. Each attack can either be a hit or a miss. The game must end with one winner and allow the user to reset the game. While the game is active, a health bar will be displayed for both players, counting down from 100% to 0%, and the menu bar will show the game log and a time counter.

**Languages**

The game will come in English and French versions. I have selected these languages because I know them best and would appeal to Canadians who wish to play the game as these are the nation’s two official languages.

**Details**

*Drawn the UC (Use-Case) diagram (ex: in an image from Paint / Visio / Powerpoint slide, or any sketch tool), describing:*

* *Manual / automatic features (ex: user selections / time features);*
* *Relationships between actors / functionalities.*
  1. **Template Solution**

**UC Diagram**

A diagram of a computer player

Description automatically generated with low confidence

Image Generate with Lucidchart**2**

**Actors table**

|  |  |
| --- | --- |
| **Actors** |  |
| Player | This actor represents a user selecting where they can place their ships and where they select to attack on the board resulting in either a hit or a miss  This actor also interacts with starting the game, resetting the game, changing board dimensions, and clicking on the about info section |
| Computer Player | This actor is a computer player which randomly selects squares on the users side of the board, resulting in either a hit or a miss. Computer players always have their ships randomly placed on the board. |
| Automatic Features | This actor will take care of background tasks such as health bar, clock counter, and the game log. |

**UC table**

|  |  |
| --- | --- |
| **Use Cases** |  |
| Random Layout | This use case describes how the board can be randomly generated |
| Selection Hitbox | This use case describes how the player or computer can select a square on the board, resulting in a hit or miss |
| About Info | This use case describes how the player can |
| Ship Layout Design | This use case describes how the player can manually place their ships on the board and change their orientation. |
| Change Language | This use case describes how the player can change the language of the game to either French or English |
| Board Size | This use case describes how the player can modify the dimensions of the board before playing a game |
| Reset | The use case is to describe how the player will reset the game after finishing a round, or alternatively, resetting a game at any point after its started. |
| Play | This use case describes how the user can start the game |
| Health Life | This use case describes both the user’s and the adversary’s life as a percentage |
| Game Log | This use case describes both the user’s and the adversary’s past moves in the form of a gamelog |

**Basic cycle**

*Create a brief description about how your game can be used.*

***Example****: If you have to design the solution to be saved and played later, how are the stems. Most importantly, how someone can play the* ***Battleship****.*

Battleship will be played by first selecting the dimension of the board. The user has the option to select the size of the board or leave it as default. Once the board dimension is defined, the number of ships either player can have will be determined. The player will have the ability to manually orient and place the ships on the board or randomly place them. Once the ships are all on the board, the user and computer will take turn selecting squares to attack. After each attack the square will show green indicating a miss or red, indicating a hit. The game ends when all ships on one side are eliminated, resulting in either a computer or user win. The user then has the option to reset the board and start again.

**FINAL SUGGESTIONS**

*Here some ideas to think about your language....*

* *Try to create a game whose execution can be very intuitive (easy to be played).*
* *Remember that this game will be in fact implemented only in the next assignment.*

**References**

1 ) “Battleship Game - Play Battleship Online,” Battleship Online. https://www.battleshiponline.org/

2 ) Lucid, “Intelligent Diagramming | Lucidchart,” *Lucidchart*. https://www.lucidchart.com/pages/

Algonquin College

Summer, 2023