BATTLE-SHIP GAME

Pena Benafa
pena.benafa@stud.hshl.de
Patrick Stephen
stephen-eteng.patrick@stud.hshl.de
Computer Programming and Software Engineering
Summer Semester 2021

July 2021

Contents

1 Game Objectives

The digital Battleship game is a strategy type guessing game for two players (player vs player or player vs computer). The game is played on the computer grids board. The ships are placed strategically on the board manually. The locations of each player's ships are concealed. Players alternate turns when missed, but will continue to play if any ship is hit. Both players can see the board of each other. The toggling will be automatic, and X will be placed on any location the player misses. The core objective of the battleship game is to destroy the opposing player's ship, and the first player to destroy his opponent ship is the winner. [4] The figure below shows a design of a player's board. while the game is in progress. The boxes marked with O are the ships found and destroyed, while the boxes marked with X show the player miss fired.

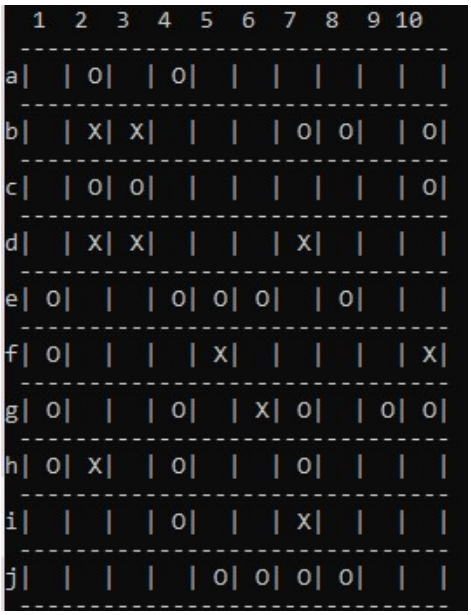

Figure 1: A player's game board during game play

2 Employed Tools and Technology

The battleship game was designed using the C programming language on a laptop. The software used in writing the C language is CodeBlock. To achieve the goal of this project, several online tools were used. At first, because there was no initial knowledge of what a battleship game is, the use of YouTube to watch an in-depth explanatory of how to play a battleship game [2] was a. great tool. Another tool used was Miro [1]. The online Miro white card dashboard was used for our brainwriting (brain-coding). The third tools used are the lecture slides from Professor Dr Tim Schattkowsky and Professor Dr Georg Birkenheuer. The fourth tool used is lucidchart [3]. Lucidchart is an online intelligent diagramming tool to draw UML diagrams. The UML diagram drew helped us make a better decision before coding.

3 Game Architecture (module structure)

The figure below is an architecture of the battleship game. When the game starts, an audio sound will be played. This audio sound indicates the game is working; also, welcome graphics will be displayed. The next is user input, which will lead to the game's logic, then the game graphics and output displayed.

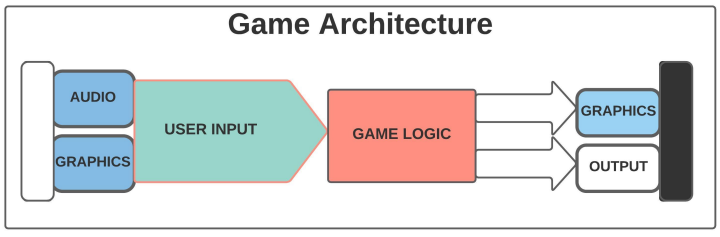

Figure 2: The Game Architecture

4 Major Design Decisions

There are several decisions taken while designing the game. One of the decision is a simple user interface. Whenever the user is at any submenu, there is an option to go back to the previous menu instead of closing the game and starting all over. Another decision taken in designing the game is to place the ships manually. This is because the game is played on one PC, an opponent can memorize the placement of the other user. This will kill the fun of the game. For the sake of bragging right, whenever anyone plays against his friend, we have included a last played status menu that shows if the previous game played is a win/lose or a draw game. This idea is shown in the figure below.

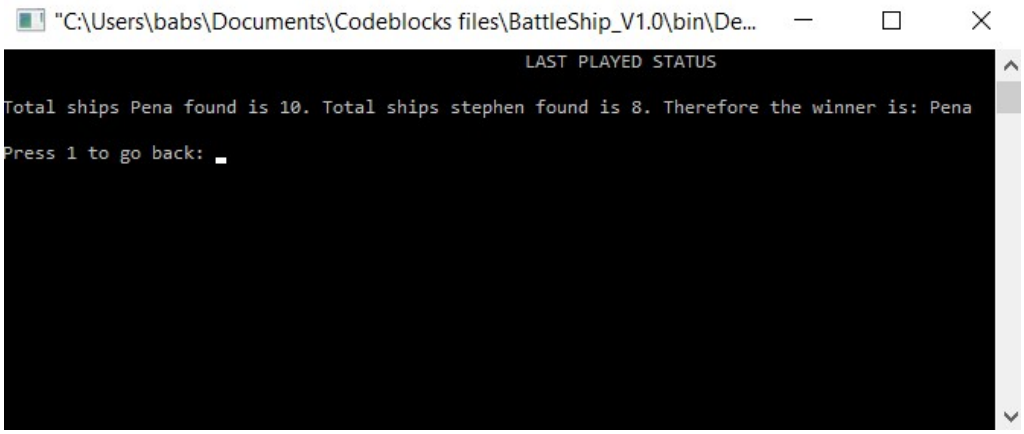

Figure 3: Last Played Status

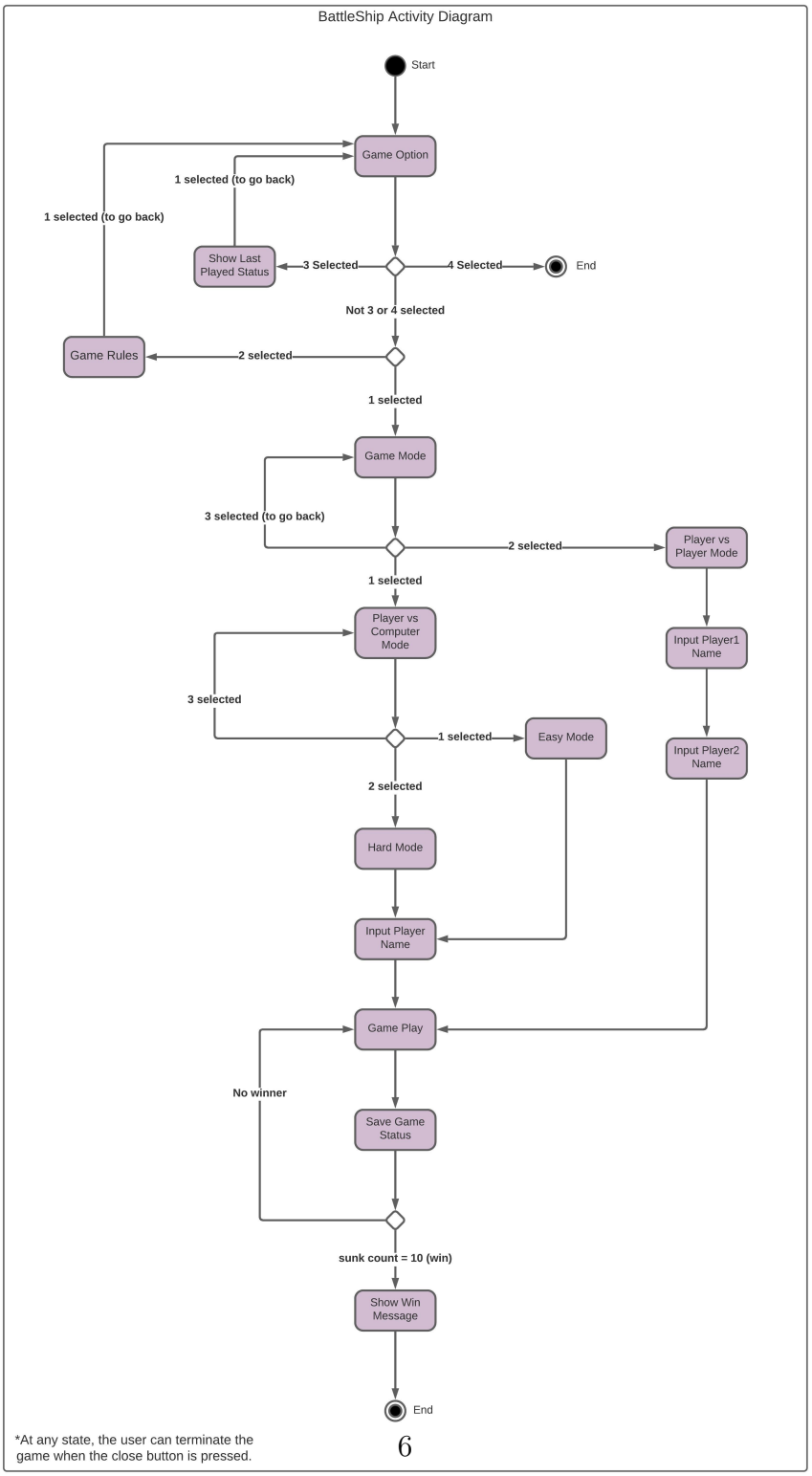

Figure 4: Battle-Ship Activity Diagram

5 Game Implementation

The battleship game runs in a sequential flow. This can be seen in figure 4. In general, when the game is started, the user has several options to select from. At any submenu, the user can go to the previous menu. The design of the game enables player vs player mode and player vs computer mode. If player vs computer is selected, there is also an option to choose an easy mode or hard mode. For easy mode, the computer guesses the positions of the ships randomly, while in hard mode, the computer will correctly sink seven ships of its opponent; thereafter, the play will be random. Whenever a ship has been sunk, there is a getstatus generated. The getstatus txt file generated will show how many ships has been sunk by a player. In figure 5 below shows the activity diagram to set difficulty mode when playing against a computer. The logic is the use of switch case. In player vs computer mode, if the user enters 1, the mode is set to easy mode and if 2 is entered, the game is set to hard mode while 3 return the submenu to the previous menu.

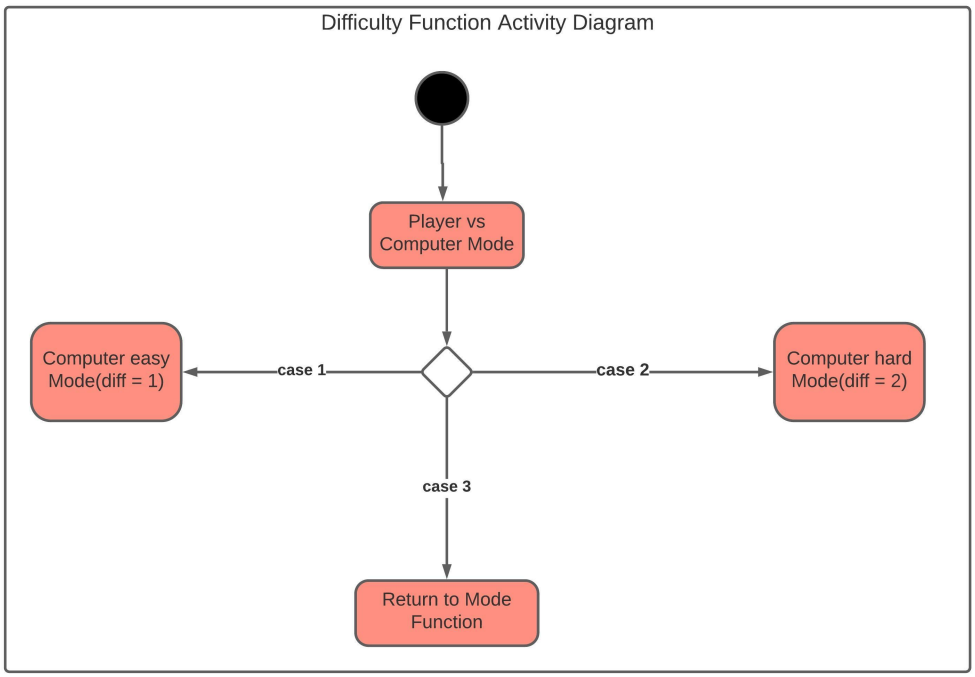

Figure 5: Activity Diagram for Difficulty Function

Figure 6 below is the activity diagram that shows how the game play function works. When in play mode, it will always be the player's turn to play at first and when there is a miss, the game switches to the next player (payer2 or computer) otherwise the player keeps playing until all ships are sunk.

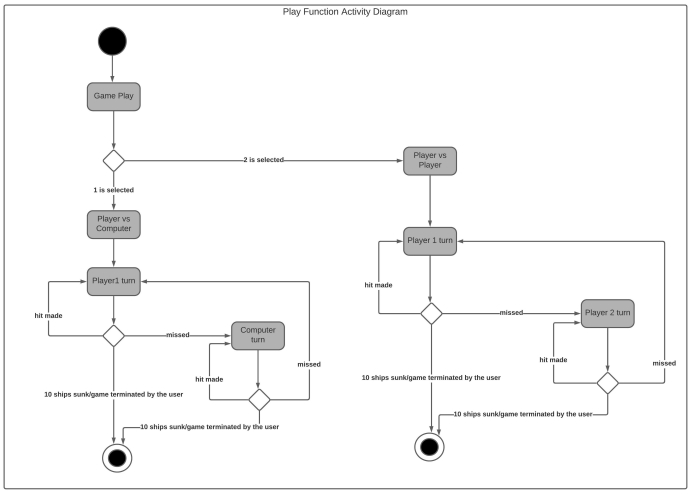

Figure 6: Activity Diagram for Play Function

To check win, a variable called sunkcount1 and sunkcount2 has been initialized for both players. At first, these variable were set to zero. Whenever there is a ship sink, the variable updates its count. The player that gets to sunkcount = 10 wins the game. The logical flow can be seen in figure 7 below.

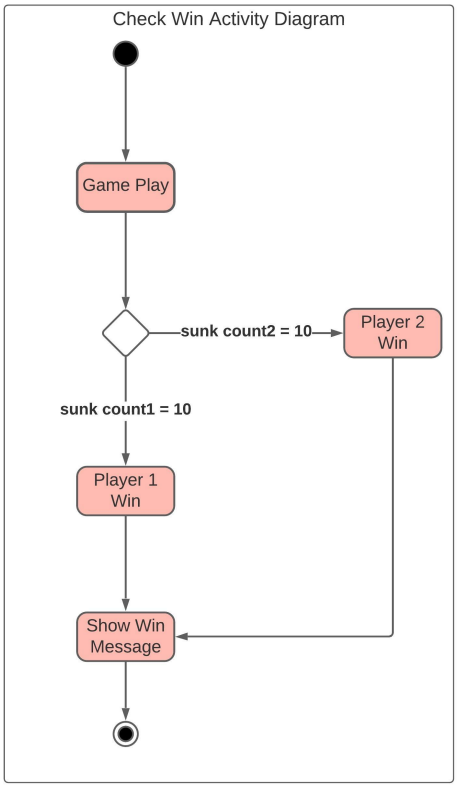

Figure 7: Activity Diagram to Check-Win Function

There are two players who take turn in playing the game. Whenever any players misses a hit, it will be the next player's turn. This feature will occur automatically with the use of toggle function created. The activity diagram in figure 8 shows the flow logic when writing the code. At first, it will be player 1 turn. If there is a hit, player 1 will keep playing until there is a miss.

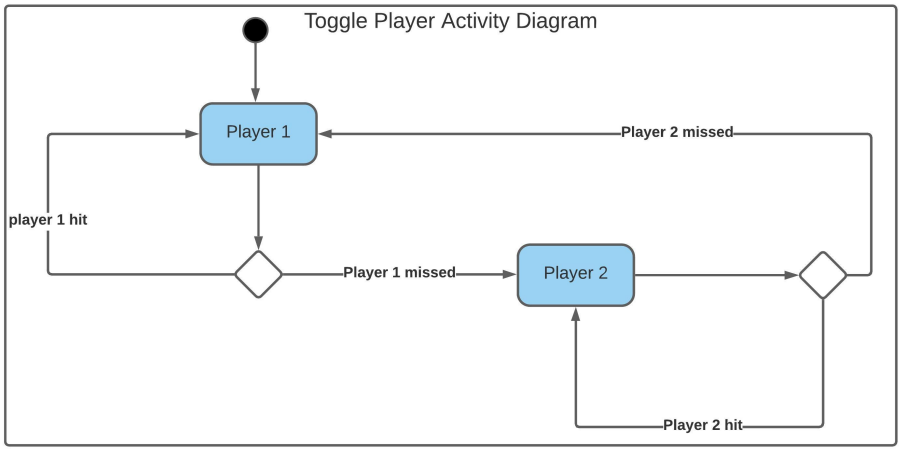

Figure 8: Toggle Player Activity Diagram

The battleship game requires that a player should guess correctly the position of his opponent ship and in that case, the ship will be sunk (destroyed). To achieve this, an activity diagram in figure 9 below shows the flow of how the code function work.

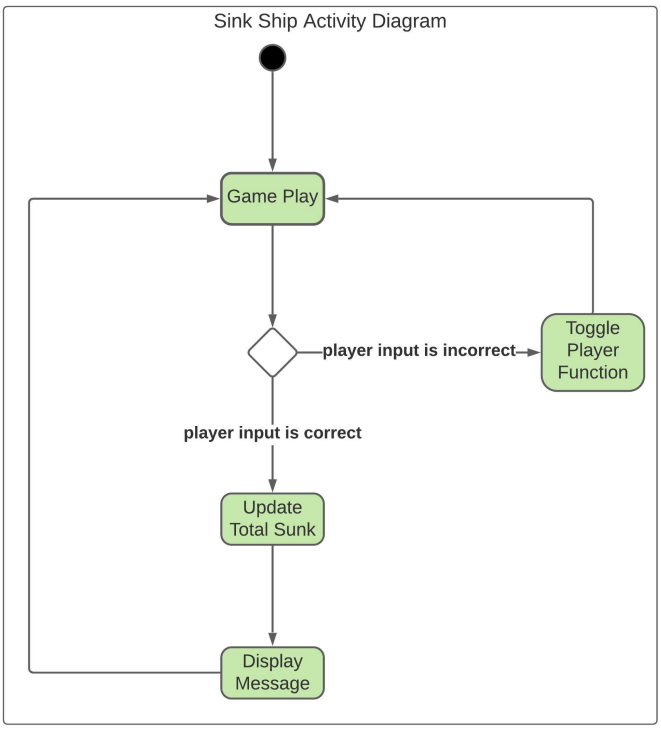

Figure 9: Activity Diagram for Sink-Ship Function

6 Team Organisation

The battleship game was developed and coded by Pena Benafa and Patrick Stephen. To achieve our goal, we met thrice in a week (Monday, Thursday and Friday) using about 4 hours in each day. We worked simultaneously in developing the game code so that there won't be any lapses.

7 Conclusion

There are various battleship game designed to be played on computer, therefore digitalizing the game. The game developed by the NG team has features such as can go back, can open game status, can play a sound. There are others features such as the ability to play online with a user on a different PC as well as a count-down timer to check player unnecessary delay will be added.

References

[1] Miro — online whiteboard for visual collaboration. https://miro.com/app/dashboard/. (Accessed on 05/04/2021).

[2] Gather Together Games. How to play battleship - youtube. https://www.youtube.com/watch?v=4gHJlYLomrs, 2019. (Accessed on 04/30/2021).

[3] Lucidchart. Intelligent diagramming. https://www.lucidchart.com/pages/. (Accessed on 05/05/2021).

[4] Wikipedia. Battleship (game). https://en.wikipedia.org/wiki/Battleship_(game). (Accessed on 07/01/2021).