# Critical Self-Evaluation

The main challenges I faced was working out how I would position ships on a grid for the player and the computer and saving and loading the game from a file.

I think the strengths of my program are the positioning ships, the fire shot for the players and the way I did the grids.

The weaknesses of my program are it has a lot of parameters being passed around, I may have been able to reduce them. The computer has no intelligent decision making behind choosing grid locations. the outputs to the console could have probably been made more readable and easier to understand. Certain methods are probably quiet inefficient.

# Updated Design

## Statement of requirements

Design and Develop a game of Battleships with two game modes. One game mode where the user can play versus the computer and the other game mode where two players can play against each other. The aim of the game is to place your ships on the grid then be the first to successfully shoot down all the opponents ships.

### Assumptions of user

The game is aimed at gamers of an age who know how to operate a PC, so around age 10 and up.

### Inputs and outputs

The Inputs of the game are using the mouse and keyboard shortcuts to select the options within the game menus. Other inputs will be the grid positions to position ships and where the players wishes to fire on the grid.

The outputs of the game are displaying an updated grid of missed and hit shots, scoreboard for missed and hit shots. The game should display a game over or victory message.

## Requirements

### Functional Requirements

R1 – The system shall have a menu to select all the options.  
R1.1 The system shall have a start new game option, which will prompt the user with a choice of game mode.  
R1.1.1 – The player shall have the ability to select an option to play versus the computer.  
R1.1.2 – The player shall have the ability to select an option to play versus player versus player.  
R1.2 – There should be a How to play option to read the instructions and rules from a text document.  
R1.3 – The system shall have a load game option to open a previously saved state of the game from a file.  
R1.3.1 – The system should display an error when no save game file is found.  
R1.4 – The system shall have an exit game option which closes the game.

R2 – The system shall have an in-game screen.  
R2.1 – The system shall have a grid size of 10 by 10

R3 – The system should have a battleship fleet  
R3.1 – The standard fleet should have 1 battleship taking up 5 grid spaces  
R3.2 – The standard fleet should have 2 cruisers taking up 4 grid spaces  
R3.3 – The standard fleet should have 3 destroyers taking up 3 grid spaces  
R3.4 – The standard fleet should have 3 submarines taking up 2 grid spaces

R4 – The system shall randomly place the computer’s ships on the grid.

R5 – The players shall have the option to position his ships on the grid manually.  
R5.1 – The players should not have the ability to overlap ship positions on the grid.  
R5.2 – The players should be able to place their ships horizontally or vertically on the grid.  
R5.4 – The players shall have an option to randomly place their ships on the grid.

R6 – The game shall let the players and computer take independent turns one after the other.

R7 – The players shall have the ability to choose a grid box to fire on the opponent’s grid.  
R7.2 – The players should not have the ability to choose the same grid box more than once.  
R7.2 – The game should notify the players if it was a hit or a miss.

R8 – The computer shall choose a random grid to fire on the player.  
R8.1 – The game should notify the player if it was a hit or a miss

R9 – the game should update the player on the number of misses and hits of the player and opponents  
R9.1 – the game should update the users grid with selected grid position that have either missed or hit

R10 – The game shall end when all the ships on one side have been sunk.  
R10.1 – The game should display a victory message when a player has won.  
R10.2 – The game should display a game over message when a player has lost.

R11 – The player shall have the option to save their game state to a file that can be loaded later.

R12 – The system should have a return to menu option to let the player leave their current game

### Non-functional Requirements

NFR1 – The system shall be compatible with the current version of Windows 10

NFR2 – The system shall not break the Copyright, Designs and Patents Act 1988.

NFR3 – The system shall be in English.

NFR4 – The computer should complete their ship position phase within 1 second.

NFR5 – The computer should complete their turn within 0.5 seconds.

NFR6 – The computer should update the scoreboard within 0.2 seconds after shot fired.

NFR7 – The system should load the game within 1 second.

## Use Cases

|  |  |  |  |
| --- | --- | --- | --- |
|  | User case title | Player versus computer | Alternatives |
|  | Description | An option that starts the player versus computer game mode. Where the computer and the player place their ships on the grid then take turns firing on each other’s grid until either the players or the computer ships have all been destroyed. |  |
| 1 | User | Select player vs computer option | A1, A2 |
| 2 | System | Switches to game interface |  |
| 3 | System | Computer player positions their ships on the grid |  |
| 4 | System | Ask user to position ships |  |
| 5 | User | Selects a grid position for each ship |  |
| 6 | System | Places the ships on the grid where the user selected | A6 |
| 7 | System | Asks user to confirm ship positions |  |
| 8 | User | Confirms ship positions |  |
| 9 | System | Start players turn | A7 |
| 10 | System | Asks user to select a position on the grid to fire at |  |
| 11 | User | Selects a position on the grid to fire at | A8 |
| 12 | System | Notifies the user whether their shot was a hit or miss |  |
| 13 | System | Updates the scoreboard |  |
| 14 | System | Updates which of the ships have been hit |  |
| 15 | System | End players turn |  |
| 16 | System | Starts computers turn | A7 |
| 17 | System | Computer fires at a random grid excluding previously selected grid positions | A9 |
| 18 | System | Notifies the user whether their shot was a hit or miss |  |
| 19 | System | Updates the scoreboard |  |
| 20 | System | Updates which of the ships have been hit |  |
| 21 | System | End computers turn |  |
| 22 | System | Repeat steps 9 to 22 |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | User case title | Player versus Player | Alternatives |
|  | Description | An option that starts the player versus player game mode. Where two players place their ships on the grid then take turns firing on each other’s grid until one player has had all their ships destroyed. |  |
| 1 | User | Select player vs Player option | A1, A2 |
| 2 | System | Switches to game interface |  |
| 3 | System | Ask player 1 to position ships |  |
| 4 | User | Selects a grid position for each ship |  |
| 5 | System | Places the ships on the grid where the user selected | A6 |
| 6 | System | Asks user to confirm ship positions |  |
| 7 | User | Confirms ship positions |  |
| 8 | System | Ask player 2 to position ships |  |
| 9 | User | Selects a grid position for each ship |  |
| 10 | User | Places the ships on the grid where the user selected | A6 |
| 11 | System | Asks user to confirm ship positions |  |
| 12 | User | Confirms ship positions |  |
| 13 | System | Start player 1 turn | A7 |
| 14 | System | Asks user to select a position on the grid to fire at |  |
| 15 | User | Selects a position on the grid to fire at | A8 |
| 16 | System | Notifies the user whether their shot was a hit or miss |  |
| 17 | System | Updates the scoreboard |  |
| 18 | System | Updates which of the ships have been hit |  |
| 19 | System | End players turn |  |
| 20 | System | Start player 2 turn | A7 |
| 21 | System | Repeat steps 14 to 18 |  |
| 22 | System | End player 2 turn |  |
| 23 | System | Repeat steps 13 to 23 |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | User Case Title | How to play | Alternatives |
|  | Description | An option that opens a text file which will include the instructions on how to use the program and the rules of the game. |  |
| 1 | User | Select how to play option |  |
| 2 | System | Opens file |  |

### Alternatives

|  |  |  |  |
| --- | --- | --- | --- |
|  | User Case Title | A1 – Load saved game | Alternatives |
|  | Description | An option that allows the user to select a previous saved game to load. |  |
| 1 | User | Selects load saved game option | A5 |
| 2 | System | Opens folder with saved files |  |
| 3 | User | Selects file |  |
| 4 | System | opens file into the game interface |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | User Case Title | A2 – Exit Game | Alternatives |
|  | Description | User can leave the game entirely by pressing this button |  |
| 1 | User | Selects exit game option |  |
| 2 | System | Closes the game |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | User Case Title | A3 – Save game | Alternatives |
|  | Description | A feature that allows the user to save game progress to a file which can be load later to resume the game. |  |
| 1 | User | Selects Save game option |  |
| 2 | System | Saves game state to a folder |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | User Case Title | A4 – Return to menu | Alternatives |
|  | Description | A feature that allows the user to quit the current game and return to the main menu screen. |  |
| 1 | User | Selects Return to menu button |  |
| 2 | System | Deletes game progress |  |
| 3 | System | Returns user to main menu |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | User Case Title | A5 – No save game found error | Alternatives |
|  | Description | When no save game files exist |  |
| 1 | System | Display error message |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | User Case Title | A6 – Overlapping ships error | Alternatives |
|  | Description | An error that occurs if the user tries to position a ship onto the grid where a previous ship is already positioned |  |
| 1 | System | Displays error message |  |
| 2 | System | Ask player to position ship again |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | User Case Title | A7 – End game | Alternatives |
|  | Description | System won’t start next players turn or computer turn when all their ships have been sunk and instead will display a game over/win/loss screen. It will also give the user an option to return to menu. |  |
| 1 | System | Displays a game over screen |  |
| 2 | System | Display scoreboard |  |
| 3 | System | Ask the user to return to menu |  |
| 4 | User | Selects return to menu option |  |
| 5 | System | Deletes game progress |  |
| 6 | System | Returns user to main menu |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | User Case Title | A8 – Selected grid position has already been selected | Alternatives |
|  | Description | An error that occurs if a player tries to fire on a grid position that has already been selected. |  |
| 1 | System | Display error message |  |
| 2 | System | Asks user to select a position on the grid to fire at again |  |

### Class Descriptions including Responsibilities, Fields and Methods

Menu *– the* class menu holds different options for the game

Fields:

* Game interface – needed to create instance of game interface

Methods

* process choices
* process game modes choice
* display choices
* close the game
* load game
* read saved files
* Read help file

Game-Interface – the class ‘game interface’ is required to allow the user select in-game options, stores and displays the scoreboard, calls and displays the grid.

Fields:

* Player grid – grid for player
* Player ship grid – grid for player’s ships
* Player 2 grid – grid for player2
* Player 2 ship grid – grid for player2’s ships
* Computer grid – grid for computer
* Computer ship grid – grid for computer ships
* ship names – array to hold names of ships
* ship length – array to hold length of ships

Methods

* saveGame
* pvp
* pvc
* player turn
* computer
* turn
* place ships
* take shot

Grid – the class grid holds information on the playing grid

Fields:

* gridBoxId – stores info on the grid position (A1, D3, F5, etc)
* gridSize – store the size of grid
* hit – stores what grid boxes are hit locations
* miss – stores what grid boxes are miss locations

Methods:

* get gridBoxId
* set gridSize equal to 10x10
* set shipPosition on grid
* get shipPosition on grid
* set hit position on grid
* set miss position on grid
* display grid
* display hits on grid
* display misses on grid
* call ship fleet

# Class diagram

Menu

Load game interface

Call and display grid

Game interface

Grid