Requirements for Missouri

An Offline PC Battleship Game

Initial Draft

Authors

Chen Shih, Giwa Haheeb Rilwan, and Phan Thanh-Long

Table of Contents

1. Introduction

This requirements specification discusses the requirements of the battleship game (Missouri) as implemented. We are designing and creating a software that plays like the classic board game Battleship. The game “Missouri” serves as a case study for exploring the basics of software design and programming in C language. Firstly, the choice of programming a game of Battleship, allows one to gain experience with both the concepts of the C programming language, the details and methods of data structures. Secondly, it also allows for experimenting with user interface design, without the risk of getting bogged down in user interface details, compared to designing more elaborate software systems.

The game of Battleship (Missouri) normally features one or two players playing against each other, each having one game board, divided into squares of equal size. Each game board typically has 10 squares both horizontally and vertically. Players will be able to play a match against another opponent. Players can also choose to play with the computer. Once a match is made, the players will place their ships on a field, and the two players will then take turns guessing and attacking the positions of their opponent's ships. The first one who sinks all the opponents ships is the winner of the match.

1.1 Purpose

This document defines the requirements of an Offline PC Battleship Game. It is created to complete a semester project in computer science class. The intended audience of this SRS document are professors, fellow students in the class, game players, and anyone who is interested in this game.

1.2 Scope

“Missouri” has been developed to provide a classic battleship game on PC for one or two player playing at the same time.

1.3 Definitions, acronyms, and abbreviations

None

1.4 References

[1] IEEE Software Requirement Specification Std. 830-1998

1.5 Overview

Section 1 discusses the purpose and scope of the software. Section 2 describes the overall functionalities and constraints of the software and user characteristics. Section 3 details all the requirements needed to design the software.

---

2. Overall description

This section provides an overview of the Offline PC Battleship Game. It describes the context of this game and the intended functionalities. Besides, information regarding User Classes and their Characteristics is given. Furthermore, the implementation constraints, assumptions, and dependencies are also discussed in detail.

2.1 Product Perspective

As shown in Figure 1, players interact with the game using a personal computer which serves as a graphical and physical interface.

Figure1 : Gaming environment and end-to-end interactions

2.2 Product Functions

In this subsection, major functionalities of the game are demonstrated in a use case diagram as shown in Figure 2. Further details will be provided in section 3.

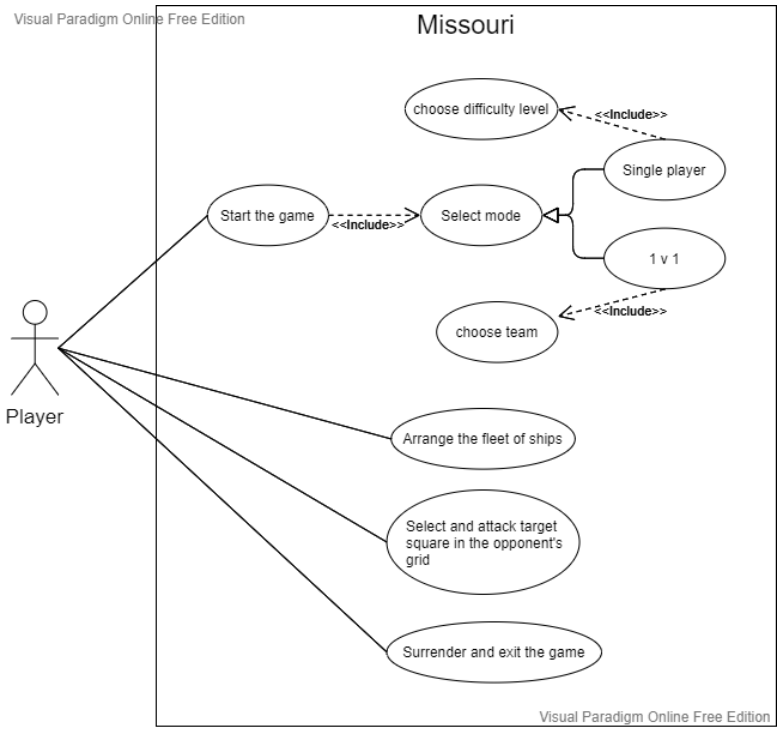


Figure 2: Major use case of Missouri

2.3 User Classes and Characteristics

The following types of user have been identified based on different characteristics:

UG01: Beginner Player

No or only little knowledge or prior experience in battleship games or , however he or she is able to understand and follow the provided instructions and tutorials in the game.

UG02: Intermediate Player

Knowing the rules of the game fully , and applying the basic strategies is not a big issue. Can beat Easy Mode proficiently, however Hard Mode is still quite a challenge.

UG03: Veteran Player

Well acquainted with basic strategies as well as advanced tactics, he or she may have developed a unique mindset toward the game. Victory comes easily in Hard Mode.

2.4 Constraints

The system shall be implemented using C programming language as instructed for the semester project, and only works on Windows 10. Windows 10 is the most popular operating system for end users, given the time constraint and limited resources, it's not feasible to develop other versions for other operating systems. Another reason is that we developers all use Windows 10, therefore it is easier to develop it in the Windows 10 environment.

2.5 Assumptions and Dependencies

None

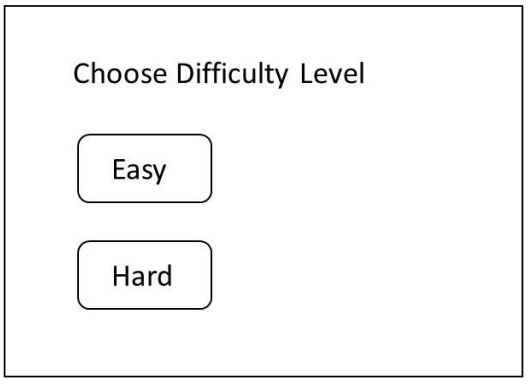
---

3. Specific Requirements

3.1.1 User Interface Requirements

The interface provided to the user should be a very user-friendly one and it should provide an optional interactive help for the player.

As shown in Figures 3, 4 and 5, players are enabled to start the game and select the game mode. Also choose the difficult level of the game.



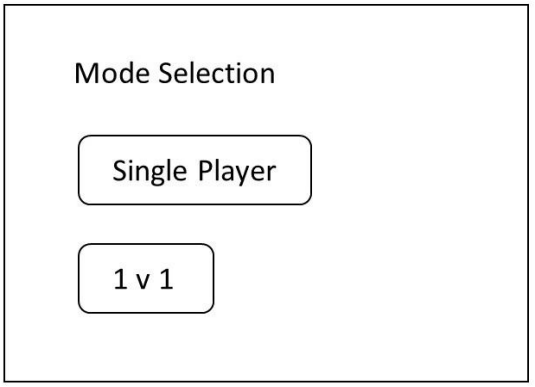


Figure 3: Start button

Figure 4: Mode selection

Figure 5: Choose difficulty level in Single player mode

In the setup phase, the player selects the ships and places them in desired position (either vertically or horizontally) as shown in Figure 6. The program will remind users if there are misplacements of vessels by showing message boxes(See Figure 7).

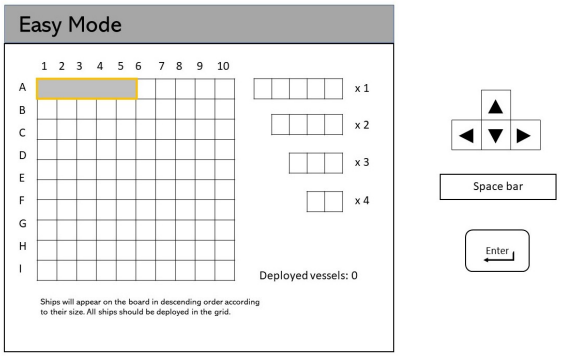


Figure 6: deployment of vessels

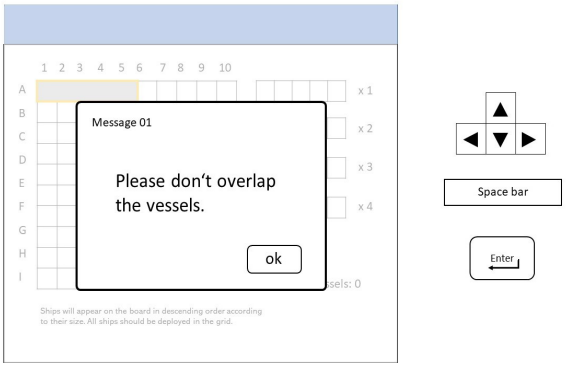


Figure 7: Message box 01

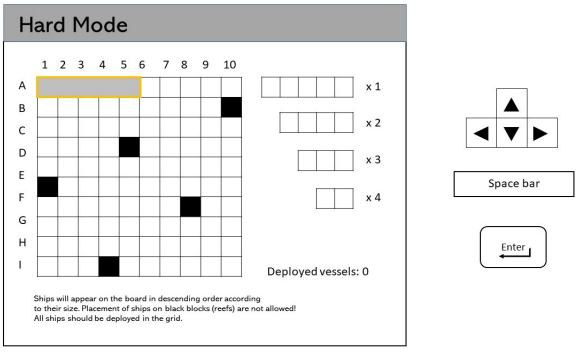


Figure 8: Deployment of vessels in "Hard" mode

And In each turn of the battle, the player first sees the damage done by the opponent and then gets to fire a shot by selecting the target position and pressing the button labeled "Attack". (See Figure 9) By clicking the “Quit” button, the user can leave the game, as shown in Figure 10.

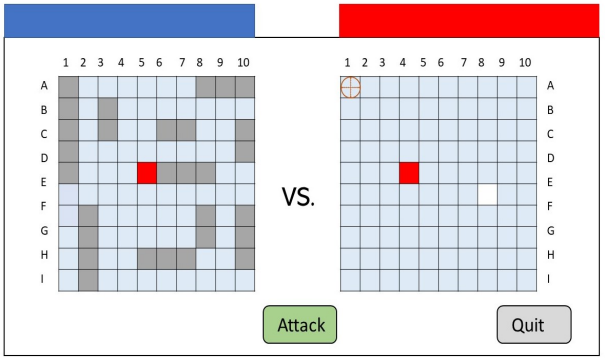


Figure 9. Fire a shot

Figure 10. Quit game

The program announces the result when the match is over, and the user can choose to return to the Home Page or exit the program. (See Figure 11)

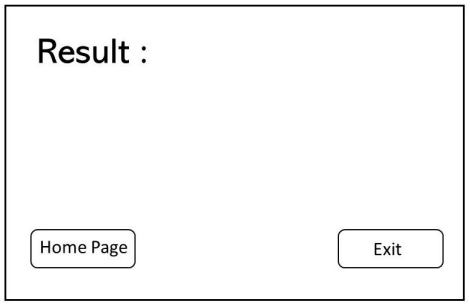


Figure 11. Game Result

3.1.2 Hardware interface Requirement.

There are various hardware components with which the game is required to interact. Various hardware interface requirements that need to be fulfilled for successful functioning of the software are as follows:

- The controller for the game is a keyboard and mouse.
- A monitor is required for display.
- PC with adequate processing power and memory to execute the program.

3.1.3 Software interface Requirement.

The program is developed with an intention to execute in the Windows 10 environment.

3.2 Functional Requirements

FN01:

Users are enabled to do basic control using mouse and keyboard.

Description:

Players can move the cursor and click on buttons as well as enter commands in the game.

FN02:

Users are enabled to start the game by clicking on the start button.

Description:

Move the cursor to the start button and click on it using the mouse.

FN03:

Users are enabled to enter the game.

Description:

Move the cursor to the start button and click on it using the mouse.

FN04:

Users are enabled to select game mode .

Description:

Move the cursor to the button “Single Player” or “1v1” and click on it to select the desired mode.

FN05:

Users are enabled to select one from 2 different difficulty levels in single player mode.

Description:

Move the cursor to the button “Difficult” or “Easy” and click on it to select the difficulty level.

FN07:

Users are enabled to deploy their fleet of ships.

Description:

The fleet is composed of 1 Aircraft Carrier(size of 5 blocks), 2 Battle vessels(size of 4 blocks), 3 patrol vessels(size of 3 blocks), and 4 small supply ships(size of 2 blocks). The ship will appear in the top left corner of the grid in a descending order. The Users are enabled to move the ship using arrow keys. The rotation is achieved by pressing the spacebar, whenever the spacebar is pressed, the vessel will rotate 90 degrees clockwise. Once the location is selected, the user can press Enter key to lock the vessel.

When the user chooses “Hard” in single player mode, 5 black squares are added to the grid, which represent reefs under the water. Placement of vessels on those blocks are not allowed.

FN08:

Users are enabled to select a target block in the opponent’s grid and launch an attack on it.

Description:

In the beginning of each round, a block-sized reticle will appear in the top left corner of the opponent's grid. The user can move the reticle to change target blocks by using arrow keys. Clicking on the “Attack” button can see if the selected block is a part of a hidden ship. The block becomes red when it hits the opponent's ship, otherwise the block will be filled with white color.

FN09:

Users are enabled to respond to the message box.

Description:

Overlapping ships are not allowed. Whenever ships overlap, a message box will pop up to alert the user. Clicking on a button labeled “ok” will close the message box, and the newly added ship will return to the spawn point.

In the “Hard” mode, if the user places the ship in a black area, a message box will pop up as a reminder. Clicking on the “ok” button will close the message box, and the newly added ship will return to the spawn point.

FN10:

Users are enabled to quit the game.

Description: If the user clicks on the “Quit” button, there will be a message screen “Are you sure you want to quit the game?”. Click “Yes” will bring the user back to the Home Page, while clicking “No” will close the message window and return to the game.

FN11:

Announce the result, and the user is enabled to return to Home Page or Exit the game.

Description: Once the match is over, a message box will pop out to announce the results. Subsequently users can choose to return to Homepage to exit the game by clicking on the buttons.

3.3 Additional Non-Functional Requirements

NF01: Security

Description: The program is protected, any unauthorized modifications and access are not allowed.

NF02: Bug-free

Description: The program is able to perform all the intended functions, preventing critical bugs from breaking the game.

NF03: Exploit-free

Description: The program minimizes the amount of loopholes which players can exploit for cheating and violating the rules.

NF04: Efficiency

Description: The program utilizes processing power efficiently.