**Software Requirements Models**

**“Spaceships: Battle for the Frontier”**

**Prepared for Mr. Jörg Kienzle**

Computer Science Department

**McGill University**

By

**The Dangling Pointers**

Sundaram Sankarapapa

Tina Latif

Avery Schmavery

Andrew Fogarty

Emir Aydin

**1. Introduction**

* 1. **Purpose**

The purpose of this document is to provide a general understanding between the requirements and the final product for the video game **Spaceships : Battle for the Frontier.**  We shall present the use cases in a manner that shall hopefully allow all parties to be comfortable and content with the product to be developed and submitted before the end of the Winter 2014 semester at McGill University.

We attempt to create a general consensus among all major parties involved in the development of this product, including, any Teaching Assistants, Professors, and Students involved in the development and / or marking of this software project.

* 1. **Scope of the Document and System**

This software to be developed is a Computer Video Game to be developed for either the Windows Operating System or Macintosh Operating System.

This document will contain and discuss all parts of the requirements elicited in the **Battleships v1.2** document available on the course webpage. The software will involve the interaction of multiple players *across two different machines* over an online server that will connect the players. The players will be able to store usage information, including but not limited to, wins and losses.

This document includes a : **Use Case Model, Environment Model, Concept Model, Operation Model, and Protocol Model.** These five (5) models will interact to present a general and thorough description of the project to be developed.

* 1. **Definitions, Acronyms, and Abbreviations**

**Player :**

When we discuss the players, we are describing the actual players who are interacting with the system by playing the game. They are distinguished from other important actors in that they have simple goals of *joining the game, playing the game, and exiting the system.*

*Current Player:*

The player who has the active turn.

*Other Player:*

The player who is waiting for the Current Player to finish their turn.

**System :**

This refers to the actual game to be developed. It encapsulates all the game logic (i.e. anything regarding playing the game), and all structural and software logic (i.e. how the system creates and interface between the players and how it develops and supports the network allowing this interaction).

**Server:**

This refers to a specific portion of the Game or System attributed to allowing two players simultaneous action to a game interface. We differentiate because, though the System encapsulates the System, it may arise that a distinction must be made between the two.

**Game:**

Another portion that is encapsulated by the System but needs to be distinguished, the Game refers to the portion of the system that deals with Game Logic. When a Player plays the Game, we refer to the player interacting with the Game Logic portion of the system with the implementation and Server logic generally abstracted out.

**Environment**

Anything not directly included within the overreaches of the System (i.e. those not included within the system boundary). This includes any external entities like Players who have some goal that they wish the system to satisfy. Typically the environment is presented an interface (dependent on the individual actor) that outlines the possible actions they are able to take on the system. The system will encapsulate and abstract the rest.

**SBF**

Spaceships: Battle for the Frontier

* 1. **References**
  2. **Overview**

The remainder of this document contains all the information required by the Requirements Models Milestone.

**Section 2:** Contains the Use Case Models and Diagrams. This is a general elicitation of the major ways the System interacts with the environment and *vice-versa.* Section also includes all possible actions a user can take while playing **SBF.**

FILL OUT AS PROGRESS

**2. Use Case Model and Elicitation**

**2.1 Introduction to Section**

The following section outlines the major interactions between the **system** and the **environment** while playing **SBF**.

This section will include all major actions demanded by the Requirements Milestone, however it will not represent each action the **player** can take the **system**, in an effort to preserve brevity. As such, only major goals are presented including by not limited to: Players logging in, Players reviewing their information, Players choosing an opponent, Players playing the game, Players quitting the system, etc.

**2.2 List and Descriptions of Major Actors**

*Figure 2.1 – Actor Description Table*

|  |  |  |
| --- | --- | --- |
| **Actor** | **Description** | **Multiplicity** |
| Player  (Primary) | The primary players of the game. The actors who have goals of logging in, playing, and logging out of the game. | 2 Players can be playing a game at any 1 time.  \* Players can be interacting with the system. |
| Server  (Secondary) | Jointly part of the system, but jointly an external entity that will allow the System to interact the multiple players. Goal of connecting the Player Actors over an online connection. | 1 Server will be active that will interact the various other components. |

**2.3 List of all Interactions to be specified by specific Use Cases**

*Figure 2.2 – Use Case Table of Contents*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name Id** | **Description** | **Details** | **Related Cases** | **Figure Number** |
| **Play SBF [S1]** | The overall summary interaction of one player as he navigates the system to login, play and exit the game. | A broad understanding of the major decisions a player takes when interacting with **SBF.** | **All** |  |
| **Login [U2]** | The specific user goal of logging into the system. | Requires User to have Login Credentials. If not, depends on Sign-Up | **Sub1** |  |
| **Sign Up[U3]** | The player wishes to create a new account. | Accessed through the Login Screen, given player does not have valid credentials. | **Sub2** |  |
| **Review Credentials[U4]** | The User reviews her game credentials, including all stats. | Accessible to all user including new users. New user credentials are blank. | **Sub3** |  |
| **Get Matched[U5]** | The User selects another player from a list to play against in the game. | Two players must agree to play against each other. This requires notification of both players. |  |  |
| **Review Game Conditions [U6]** | The two users review the game decisions. Deciding on an appropriate map. | Two users are provided a map, and both must agree on it.  The players also agree on the amount of gold they can spend. |  |  |
| **Play Game [S2]** | The summary interaction of the two players playing the actual game. | Differentiated by the Login phase and the exit phases.  Contains the brunt of the interactions. |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name Id** | **Description** | **Details** | **Related Cases** | **Figure Number** |
| **Select Ships[U7]** | The two players select the ships they are taking into battle. | The players select their ships using the gold amount they agreed on. |  |  |
| **Place Ships[U8]** | The Users place their ships on the battlefield, according to game rules. | Placement of ships is asynchronous. Players merely alert the system when they are done. |  |  |
| **Start Game [U9]** | The two players agree to start the game. | This is based on the joint decisions they have made in U6 | **U6** |  |
| **Play Turn**  **[U10]** | The player plays their turn. | The player has a host of actions they can take while playing the game. **See figure 2.3** | **Figure 2.3**  **Sub2** |  |
| **Review Credentials**  **[Sub1]** | The system checks the Username and password | Three attempts before giving user option to create new account. |  |  |
| **Store State Details**  **[Sub2]** | The system stores some State detail. | Include: User/Pass, game details, stats. |  |  |
| **Retrieve State Details [Sub3]** | The system retrieves some State details. | Requests server for specific information needed to accomplish some other goal. |  |  |

*Figure 2.3 – In Game Actions Players can take.*

|  |  |  |
| --- | --- | --- |
| **Action** | **Description** | **Must be Current Player?** |
| Shoot (Torpedo or Cannon) | The player shoots one of his projectiles. The System informs player of allowed choices. Current player is notified if hit, other player is notified that shot was taken. | Yes, only the Current player can shoot. |
| Deploy/ Retrieve Mine | The Current player uses one of his mine ships to deploy or retrieve a mine in the surrounding area. System does *not* notify other player that mine was deployed/ | Yes, only the current player can deploy and retrieve mines. |
| Activate / Deactivate Sonar | The current player uses one of her radar ships in order to assess the map. | Yes, only the current player can activate or deactivate the sonar. |
| Save / Load Game State | Both the Current and the Other Player agree to save or load the game state. | **No,** both players must together decide whether they want to save or load the game at any time. |
| Repair Ship | The Current Player navigates their ship to the base, and selects the repair option. Allowing their ship to be repaired over time. | Yes, only the Current Player may be able to activate reparations at any time. |
| Send Message | The players send a text message to one another. | No, either player can send and receive messages at any time. |
| Review Game Rules | Either player decides to review the game rules or the possible decisions they are able to make. | No, either player can review the game rules at any given time. |