**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. | 1 Player interacts with the system. They play as if they are playing against the system, where in reality the “system” is another player doing the same thing. |
| Screen  (Secondary) | The screen where the program is displayed. The screen is the main hardware interface between the player and the game. | 1 there is one active per player. |
| Mouse / Keyboard  (Secondary) | These are the operational tools of the player. They are what the player does to navigate the screen (i.e. the game interface). | 1 Set of mouse and keyboard per player. |

**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 Statistics[Sub2]** | The User wishes to review game statistics for Player. | Player can review statistics for themselves, or other players. |  |  |
| **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. |  |  |
| **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** |  |
| **Validate**  **Credentials**  **[Sub1]** | The system checks the Username and password | Three attempts before giving user option to create new account. |  |  |

*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. |

**2.4 Use Case Model**

Play SBF Use Case [S1]

**Use Case:** Play Spaceships: Battle for the Frontier

**Scope:** SBF

**Level:** Summary

**Intention in Context:** The player wishes to receive some enjoyment by playing *Spaceships: Battle for the Frontier*, hoping also to win the game.

**Multiplicity:** One player interacts with his system interface at any given time. Multiple players will be playing on multiple interfaces at any given time.

**Primary Actor:** Player

**Secondary Actors:** Mouse / Keyboard and Screen

**Pre-Condition:** Player has valid program downloaded and installed on an appropriate operating system.

**Main Success Scenario:**

1. *Player* logs into[U2] *System*.
2. *Player* gets matched [U5] to an opponent from list provided by *System*.
3. *Player* plays a game[S2] of **SBF**.

*Steps 2-3 can be repeated indefinitely.*

1. *Player* exits from *System*

**E**xtensions

1a. *Player* does not have an account. Sign Up[U3]

3a. *Player* wishes to save game state. Proceed to (4). Use Case ends in Success.

Login Use Case [U2]

**Use Case:** Login to System.

**Scope:** SBF

**Level:** User Goal

**Intention in Context:** The player wishes to retrieve gameplay details, credentials, and identity, from a previously created account.

**Multiplicity:** Multiple players can attempt to retrieve their credentials simultaneously.

**Primary Actor:** Player.

**Secondary Actor:** Mouse / Keyboard and Screen.

**Pre-Condition:** The user has already created an account with the system – specifically the server.

**Post-Condition:** The player will be logged into the System identified by their unique identification name provided in login.

**Main Success Scenario:**

1. *System* displays user login page to *User*
2. *Player* provides System with login details.
3. *System* validates user credentials [Sub2] .
4. *System* notifies Player of valid login details.

**Extensions**

2a. Player does not have login details. Player Signup[U3]

(3) a. System determines incorrect login. User is notified. Use case starts again.

Sign Up Use Case [U3]

**Use Case:** Create a new account with the game.

**Scope:** SBF

**Level:** User Goal

**Intention in Context:** The player does not have an account with the System. Wishes to create a new account.

**Multiplicity:** Multiple players can and must create new accounts.

**Primary Actor:** Player

**Secondary Actors:** Mouse / Keyboard and Screen.

**Pre-Condition:** The Player does not have a valid account with the system. Wishes to play SBF.

**Post-Condition:** The System will store and recognize details for a new player.

**Main Success Scenario:**

1. *Player* provides *System* with required credentials.
2. *System* validates user credentials [Sub2]
3. *System* alerts *Player* that credentials are validated.
4. *Player* Logs into[U2] *System.*

**Extensions:**

2a. User does not provide full and appropriate credentials. Case resumes at 1.

3a. *System* determines some major error. Use case ends in failure.

Review Statistics [Sub2]

**Use Case:** Review statistics and player performance.

**Scope:** SBF

**Level:** Sub-Function of Get Matched.

**Intention in Context:** The player wishes to review the game statistics of some other Player actor (even the himself).

**Multiplicity:** Multiple users can review their statistics simultaneously.

**Primary Actor:** Player

**Secondary Actor:** Mouse / Keyboard and Screen.

**Pre-Condition:** User has successfully logged into the system.

**Main Success Scenario:**

***Steps 1-2 can be repeated in sequence.***

1. *Player* notifies *System* about a specific player statistic to review.
2. *System* presents appropriate game statistic to *Player*.
3. Player finishes reviewing statistics, returns to Get Matched [U5]

Get Matched [U5]

**Use Case:** Match players in order to start a game.

**Scope:** SBF

**Level:** User Goal

**Intention in Context:** The player is ready to start a game. She wants to find an appropriate partner to engage in a game of SBF.

**Multiplicity:** Multiple players will be attempting to find partners at any given time.

**Primary Actor:** Player

**Secondary Actor:** Player

**Pre-Condition:** Each individual player has appropriately logged into the System.

**Main Success Scenario:**

1. *Player* notifies *System* that she wants to select a new opponent.
2. *System* responds by presenting *Player* a list of possible opponents.
3. *Player* notifies *System* of a player she wishes to challenges.
4. *System* responds confirmation that other Player has accepted challenge.
5. *Player* plays a game [S2] s.

**Extensions**

2a. *System* determines there are no other *players* logged in, Use case ends in Failure.

3a. Player Reviews Statistics about other player to determine compatibility. Use case resumes per step 2.

4a. *Other Player* does not accept invitation to play game. Use case resumes per step 2.

4ai. *Player* cannot send more then 2 requests to *Other Player* in a span of 1 hour.

Play Game Use Case [S2]

**Use Case:** A new game of SBF is started.

**Scope:** SBF

**Level:** Summary Level

**Intention in Context:** The player has been matched up with another player. He wishes to engage in a new game of SBF.

**Actor:** *Player*

**Secondary Actor:** *Other Player*

**Pre-Condition:** System has matched up two Players to engage in a game. Both players have agreed to face each other.

**Post-Condition:** Game statistics for the *Player* will be updated, including rankings.

**Main Success Scenario:**

*At this point, the Players may also decide to Load a game. This scenario means use case begins at step 9.*

1. *System* notifies player of Game Conditions [U6].
2. *Player* sends confirmation of Game Conditions[U6] to *System*.
3. *System* presents ship selection page to *User*.
4. *Player* notifies *System* which Ships he will take into Battle [U7],
5. *System* confirms validity of Ship Selection to Player.
6. *System* asks *User* where ships should be placed.
7. *Player* notifies *System* where she would like to place her ships[U8] initially before game start.
8. *Player* alerts *System* she is ready to Start game[U9]*.*

*Step 7 is repeated until one player wins the game.*

1. *Player* takes a turn [U10]
2. *System* alerts *Player* of end of game.

**Extensions**

2a. *Player* updates game conditions. Use case begins at 1.

3a. *System* determines invalid selection of ships. Use case begins at 3

7a.  *System* determines invalidity of ship placement. Use case begins again at 6.

9a. *User* saves game state. System exists. Use Case ends in success.

Game Conditions Use Case [U6]

**Use Case:** Review game terms and conditions.

**Scope:** Play Game

**Level:** User Goal.

**Intention in Context:** The *Player* is presented with a list of game terms. These terms include the Map and the amount of money the players have to buy ships for that game. The player reviews these conditions and makes any appropriate alterations.

**Multiplicity:** 2 Players are interacting with the System simultaneously. One player at a time gets updated game conditions.

**Primary Actor:** Player

**Secondary Actor:** Other Player.

**Pre-Condition:** Both players have been properly matched up, and have agreed to start a game.

**Post-Condition:** The System has registered the appropriate conditions to produce for the specific game instance.

**Main Success Scenario**

*Steps 1 and 2 are repeated until a unanimous decision is made by both players.*

1. *Player* is notified of game conditions by System.
2. *Player* reviews and sends new set of conditions to System.
3. *System* returns confirmation of game condition to *Player*.

Select Ships Use Case [U7]

**Use Case:** Selection of ships for battle.

**Scope:** Play Game

**Level:** User Goal.

**Intention in Context:**  The *Player* selects ships from a roster of available brigs according to a declining balance decided upon in the **Game Conditions.**  These are the ships the player will be bringing into battle.

**Multiplicity**: 2 player will be selecting their ships in any one given game instance at the same time.

**Primary Actor:** Player

**Main Success Scenario:**

1. *System* notifies *player* of valid and appropriate ships.
2. *Player* selects ships based on balance and alerts *System* of choice.
3. *Player* alerts system that all valid choices are made. Move to Deploy Ships