Software Requirements Specification

for

<Sub-Dolphin Machine Cannon>

Version 2.0 approved

Prepared by <Team [SDMC]>

< Team [SDMC]>

<10/21/2016>

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. User and Product Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 3

2.4 Operating Environment 4

2.5 Design and Implementation Constraints 4

2.6 User Documentation 4

3. External Interface Requirements 4

3.1 User Interfaces 5

3.2 Hardware Interfaces 7

3.3 Software Interfaces 7

3.4 Communications Interfaces 7

4. System Features 7

4.1 Game Interface Requirements 7

5. Other Nonfunctional Requirements 8

5.1 Performance Requirements 8

5.2 Software Quality Attributes 8

Appendix A: Glossary 9

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Stephen Harb | 10/20 | First entry | 1.0 |
| Stephen Harb | 10/20 | Addition of PDR sections into SRS | 1.1 |
| All | 10/20 | First addition of functional requirements | 1.2 |
| All | 10/21 | Changes to sections 3,4 | 1.3 |
| All | 10/21 | Additions to sections 2,3,4,5 | 1.4 |
| All | 10/21 | Addition of User class Flowcharts | 1.5 |
| Daniel LaBorde, Mason Meredith | 10/21 | Addition to Sections 3,5 | 1.6 |
| Chris Gore, Stephen Harb | 10/21 | Section Rewrites 1,3,5 | 1.7 |
| Mason Meredith | 10/21 | Addition of Menu Diagram | 1.8 |
| Stephen Harb | 10/21 | Document Edits | 1.9 |
| Daniel LaBorde | 10/21 | Further Edits | 1.10 |
| Chris Gore | 10/21 | Final Document Editing | 2.0 |

# Introduction

## Purpose

*Team [SDCM]* shall develop a tactical game based for the Windows pc. This project shall undergo a full project life-cycle, including gameplay testing. The game shall consist of player elements (Dolphins) and conflict elements (Sharks) that involve high-strategy and tactical decision making in a turn-based environment. Players shall be able to make decisions which effect gameplay in such a way that proper decisions shall lead to victory in the game. The game may include both computer vs player and player vs player gameplay, but not necessarily both. The game shall be called Sub-Dolphin Machine Cannon, or SDMC, which shares an acronym with the team name SDMC or Software Develop Machine Creation.

## Document Conventions

The following conventions were taken when fabricating this document; requirements are inherited from objectives, and design scope is inherited by methodology. The figures are organized around the user case, and take into account each separate option and attribute of potential use.

## Intended Audience and Reading Suggestions

The following document is intended for reading for team members, LSU CSC students and LSU CSC staff. The document is organized around user experience development. For those who wish to reach a shallow but wide coverage of the program the section 2 is the most effective. For more in-depth understanding of the game including many specific components the section 3 is most effective. For navigational information on the game section 4 is effect. For largely miscellaneous information the 5th section is useful.

## Product Scope

Sub-Dolphin Machine is based on turn based tactical full-awareness games in the vein of chess and Square Enix’s *Final Fantasy Tactics Advance* as well as other games within the genre. The games main artistic contribution to the genre is the addition of armed dolphins. The game will involve deep turn based strategy and player advancement as well as the ability to control the in-game avatars, which will be dolphins with cannons on them. The player will progress through different levels of the game, level up, and unlock new items for customization along the way.

## References

For technical reference; refer to Team [SDMC] Preliminary Design Review and Team [SDMC] Contract

# User and Product Description

## Product Perspective

This product is a member of the turn-based tactical game genre. The main selling features will include the games high accessibility and the low skill floor for intro players, with a growing skill cap for play mastery. Individual units will be unique and feature an appealing art style. Both PC and Smartphones are both planned as game platforms for the tactical game.Proof-of-Concept and demos shall be produced for the PC and potentially exported to a mobile platform afterwards.

## Product Functions

The tactical game will feature turned based selection features with a built-in menu. The selection features will be overplayed on a tactical game board it unites that are either player or computer controlled. These units will have different types and stats that will drive the main conflicts of the game. Player elements are mainly driven by tactical game decisions. Combat elements will be driven by tactical choices and inter-player strategy. Artistically this game is allowing for non-standard expression of artistic ability for players. There are currently no concept art or concept media. Artistic influences are mostly 2D Japanese strategy games, including; Final Fantasy Tactics Advance, Fire Emblem, Shining Force, etc. Music is intended to be acquired for the game

## https://documents.lucidchart.com/documents/116f5b1b-1821-44fe-97ff-db2a6925898a/pages/0_0?a=410&x=9&y=50&w=1122&h=660&store=1&accept=image%2F*&auth=LCA%2073b30320210cc2471f10d3fde15a9e0df1c40a82-ts%3D1477089688User Classes and Characteristics

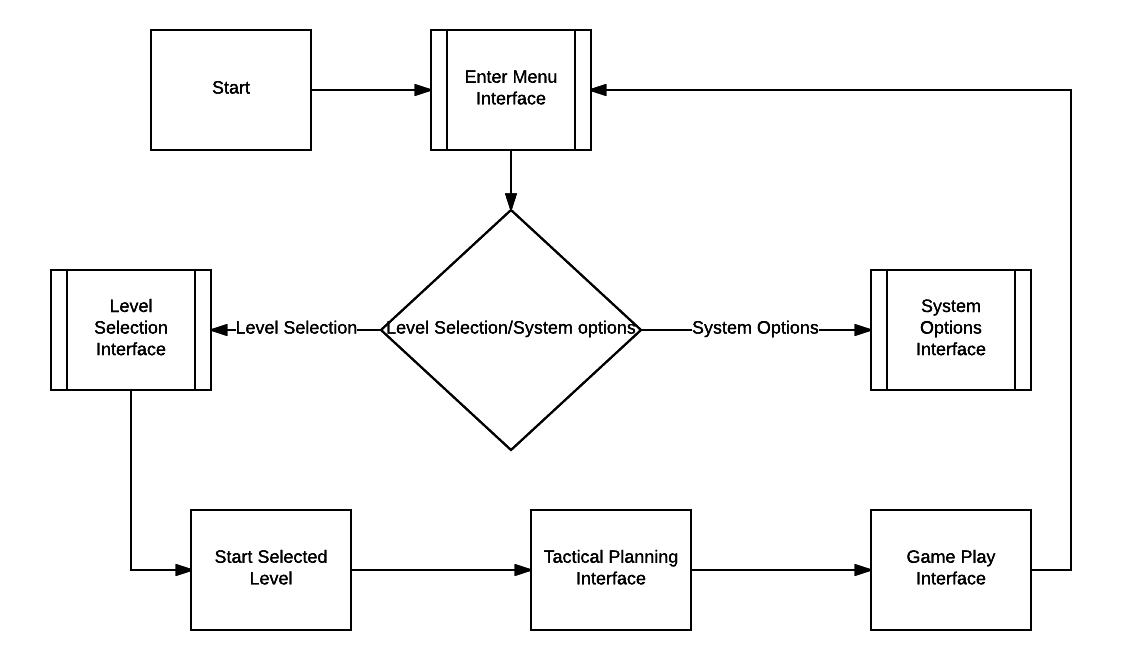


Figure 1. Logical flowchart for user interface. The User classes are based on the interaction of the user with the system, the user makes selections that bring them into their game or give them the ability to control certain aspects of the game

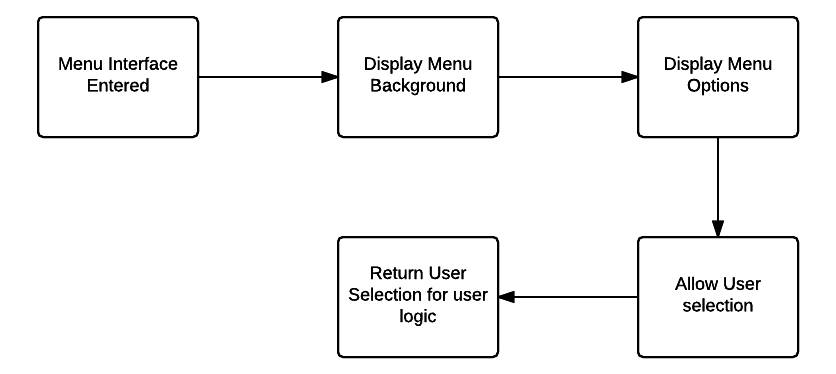
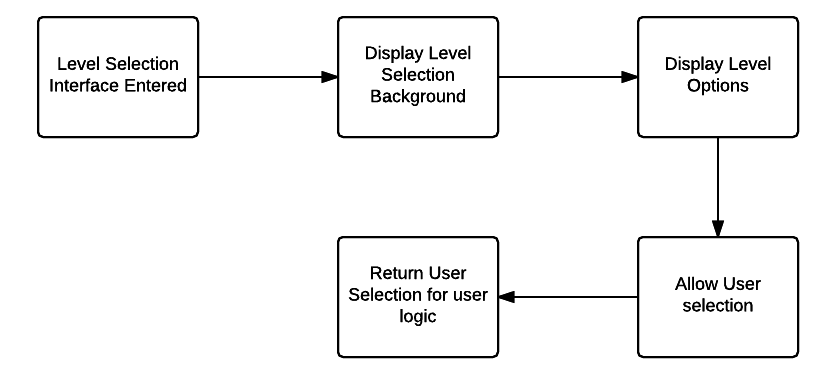
Figure 2: Menu Interface Logic, outlines the actions taken by the menu interface

Figure 3: Level Selection Interface, outlines the actions taken by the level selection interface

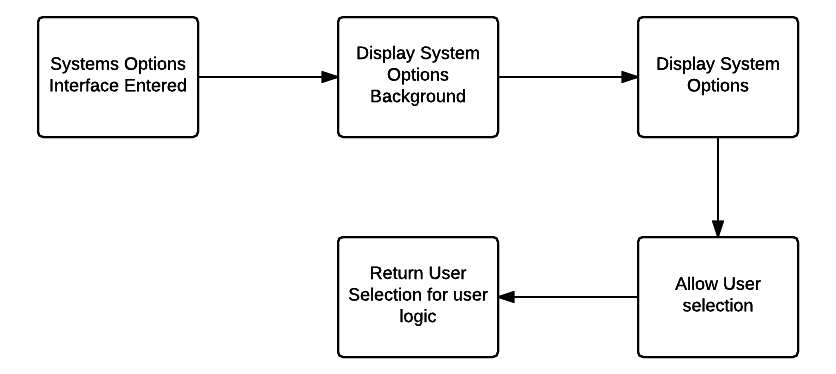


Figure 4: Systems Options Interface, outlines the actions taken by the systems options interface

## Operating Environment

Currently multiple game-design engines are being looked at right now as options. Unity, Unreal, SDL2, QT, Libgdx, Allegro are all being considered, each providing different benefits. Team members are experienced in Java, as such environments that support Java, and engines using Java will be treated preferentially. The game requires an in-depth GUI and a multi-layered menu system.

## Design and Implementation Constraints

The time constraint on the product development prevents full implementation of all potential features. Code maintenance is a large factor in determining the success of this project, thus a heavily modulated system is required for the development. An agile development model is best for the development of this game.

## Assumptions and Dependencies

Team [SDMC] plans to use the Unity development engine to develop the GUI, the assumption is that the team will be able to use the engine for free and have limitless access to the software. However, if this were to change, development of the GUI would become much more difficult, and would require a change to the development environment for the GUI.

# External Interface Requirements

## User Interfaces

The following user interfaces are planned; menu/selection interface, tactical planning interface, play interface, and system options interface.

Each interface will work as such:

3.1.1 Menu Interface:

This Interface will include the start menu. The start menu will allow the player to either enter the main game, or shut down the program. This menu also allows you to go into either the campaign or a simple random battle.

3.1.2 Tactical Planning Interface:

This interface will be the available after a campaign or random battle has been selected. The main function of this interface is to select difficulty level, player units (either dolphins or sharks), or a saved game.

3.1.3 Play Interface:

The player interface is the interface that the user will use the most. This interface will present itself as a square based map. The player will have units on the map, and will have to choose which squares to move said units. Each unit has its own special features and abilities. This is where the strategy in the game comes into play.

3.1.4 Systems Options Interface:

This menu is where the player will either save the game or exit the game to the menu. This menu will thusly have three basic features “exit to main menu”, “exit the current menu”, or “save game.”

3.1.5 The Following GUI standards will be developed for:

Menu Interface

The main menu will require three basic “buttons.” The first button will be a link to shut down the game. The second button allows the user to view the tactical planning interface (campaign). The last button leads to the same tactical planning interface but it is (skirmish mode).

3.1.6 Tactical Planning Interface

This interface will have the options to pick either the dolphins or sharks in a drop down menu. These two sides each have certain boons to consider including attributes like attack, defense, or resources. Additionally, the tactical planning interface also allows the user to download previous games into a saved slot or overwrite the one saved slot. This means that there are two buttons regarding this function one for a new game, and one for a continued game. Finally, if it is a new game then the difficulty is re-set to either easy or hard with an additional two buttons.

3.1.7 Play Interface

This interface will need a select function for each unit in play. Each unit will need a picture, and the select will cause a set of red squares to open for each moveable space. This unit will also have a small button at the bottom of the screen that will open the System Options Interface. When next to an enemy unit the friendly will turn to a new picture. By selecting this unit, the user can either attack (changing the color of the unit background), or defend (changing the color of the unit background).

3.1.7 System Options Interface:

This interface is a simpler interface that will require only an open menu with an “x” button at the top of the menu, a “save” button, and a “return to game” button

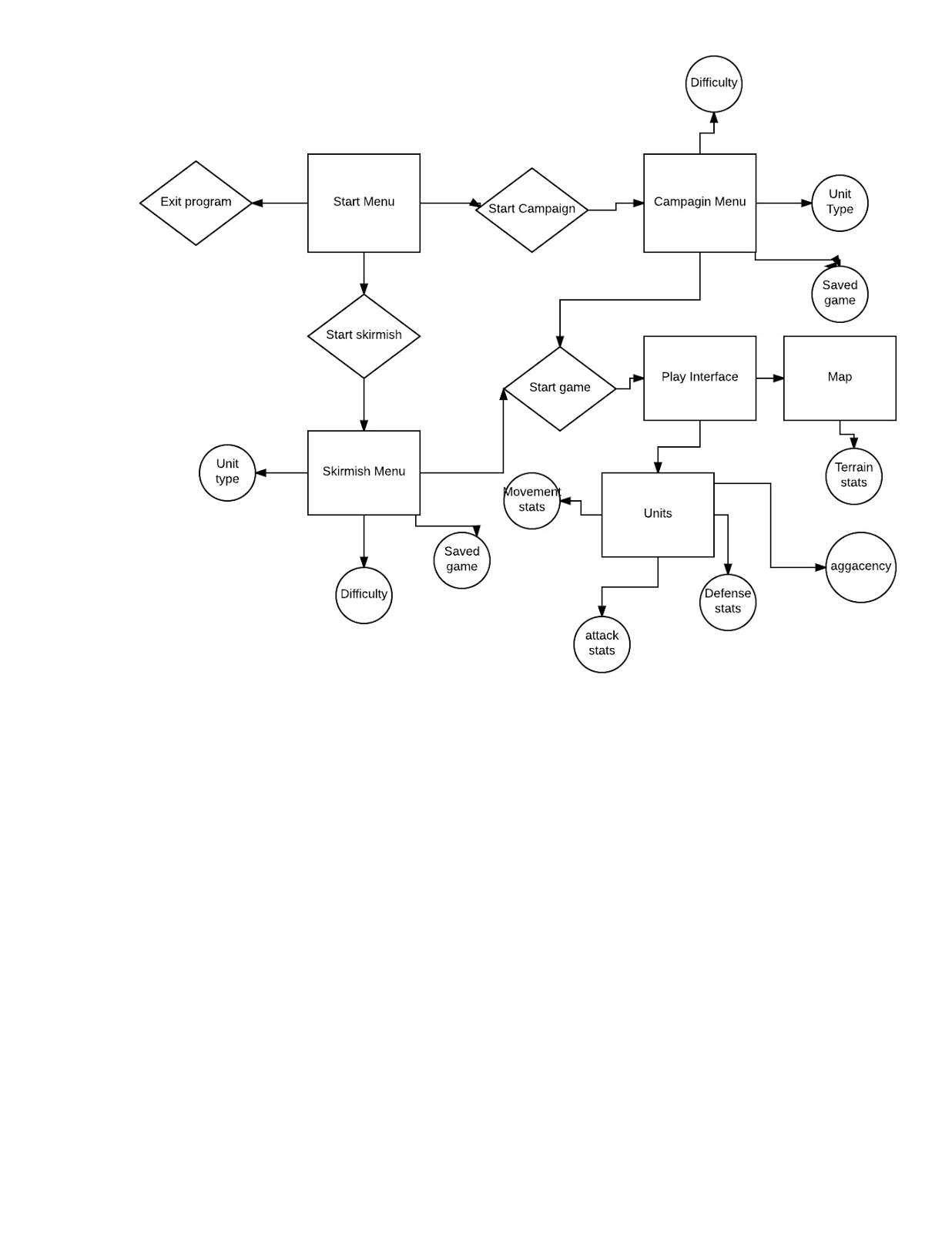


Figure 5: System Menu Use Case Chart, outlines the attributes and system interfaces of the user menus.

## Hardware Interfaces

The main intended support device for this game is the PC, thus no specific hardware interactions will be taken into account. The game shall run as an executable, with 2D graphics, so no special software or hardware will be needed to run the game.

## Software Interfaces

## This game program will be developed on Unity and specifically designed for Windows systems. The program is a game, and thusly will require art to some degree. However, the game will only need simple art though certain aspects may require artistic programs.

# System Features

## Game Interface Requirements

4.1.1 The game interface has the highest design priority. The interface will involve selected movements and player involvement as well as the tactical structure for the game. The game interface will include the menu interface, the play interface, the tactical planning interface and the system options interface. These interfaces will be the main way that the player interacts with the game. The highest risk factor is in this interface, as an error in the game interface will prevent the game from being played, and must be tested exhaustively.

4.1.2 User action sequence, menu selection->level selection->tactical planning interface->play interface, the menu selection can go into level selection or system options interface which will allow a player to control the set system options.

4.1.3 Functional objectives:

1. -User shall be able to select game play or system options from a menu
   1. -User shall be able to enter a system options menu is system options is selected
   2. -User shall be able to control game lighting from the system options menu
   3. -User shall be able to control game volume from the system options menu
2. -User shall be able to select a game play level if game play is selected
   1. -Once a level is selected user shall be moved to a tactical planning menu
   2. -The user shall then have the option to control certain starting conditions
   3. -User shall be able to start the level at any time from the tactical planning menu
   4. -Once a user starts a level, the game will start on the user turn
   5. -The user shall be able to quit a game at anytime
   6. -The game shall end once the end conditions are met
3. -Levels that have been completed shall be noted as completed in the level selection menu
4. -The system shall include a tested error check system
5. -Invalid inputs into the system shall return no response

4.1.4 Functional Requirements

From Objective 1.

REQ-1: A user start menu shall be designed, developed and tested.

REQ-2: A user system options menu shall be designed, developed and tested.

REQ-3: A user level selection interface shall be designed, developed and tested.

REQ-4: An option for controlling game lighting shall be designed, developed and tested.

REQ-5: An option for controlling game volume shall be designed, developed and tested.

From Objective 2.

REQ-1: A series of levels of play shall be designed, developed and tested.

REQ-2: A tactical planning interface shall be designed, developed and tested.

REQ-3: In-game options shall be designed around the player experience

From Objective 3.

REQ-1: A log of completed levels and unlocked play options shall be recorded

REQ-2: An interface so players can see their unlocked play options shall be available

From Objective 4.

REQ-1: A series of error types shall be implemented for testing as well as play

REQ-2: Errors shall be recorded and used for testing

REQ-3: Each element of the game shall be tested and redesigned based on the error logs

From Objective 5.

REQ-1: Any key pressed that does not have a mapped return shall have no response for the game.

# Other Nonfunctional Requirements

## Performance Requirements

PR-1: The system shall allow for two users to have access to the game at one time, depending on game type (single/multiplayer).

PR-2: The system shall have a time determined through testing to load a new or saved game.

PR-3: The system shall have a time determined through testing to open interfaces and display messages, after the user has interacted with system.

PR-4 Enemy units (AI) response time to the user’s actions will need to be within a consistent amount of time, which will be determined during testing.

## Quality Attributes

Robustness: In multiplayer, if users lose connection to each other. The system shall save the game and enable the users to reconnect to the game, once connection has been restored.

Portability: The game will not require internet connection, allowing for more portability with less gameplay constraints.

Usability: The games UI’s will be very easy to navigate, and the actual gameplay will be very straightforward. Yet, it will still allow for very strategic gameplay.

Availability: Users with the proper devices to access the game will have access at all times.

Appendix A: Glossary

**Team [SDMC]: Team Software Develop Machine Creation**

**SDMC: Sub-Dolphin Machine Cannon**