Deepsea Survival

Software Requirements Specification

v1.0

6 March 2017

Sean Coates
Designer/Coder
Prepared for
CMPS 3350—Software Engineering
Instructor: Gordon Griesel
Spring 2017
## REVISION HISTORY

<table>
<thead>
<tr>
<th>Date</th>
<th>Description</th>
<th>Author</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 March 2017</td>
<td>V1.0</td>
<td>Sean Coates</td>
<td>Added all projected and necessary items to the specifications.</td>
</tr>
</tbody>
</table>

## DOCUMENT APPROVAL

The following Software Requirements Specification has been accepted and approved by the following:

<table>
<thead>
<tr>
<th>Signature</th>
<th>Printed Name</th>
<th>Title</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sean Coates</td>
<td>Designer/Coder</td>
<td>6 March 2017</td>
</tr>
</tbody>
</table>
Table of Contents

REVISION HISTORY ..................................................................................................................... III
DOCUMENT APPROVAL ............................................................................................................. III

1. INTRODUCTION .................................................................................................................. 1
   1.1 PURPOSE ............................................................................................................................ 1
   1.2 SCOPE ................................................................................................................................. 1
   1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS .......................................................... 1
   1.4 REFERENCES ....................................................................................................................... 1
   1.5 OVERVIEW ........................................................................................................................ 1

2. GENERAL DESCRIPTION ....................................................................................................... 2
   2.1 PRODUCT PERSPECTIVE .................................................................................................. 2
   2.2 PRODUCT FUNCTIONS ...................................................................................................... 2
   2.3 USER CHARACTERISTICS ............................................................................................... 2
   2.4 GENERAL CONSTRAINTS .................................................................................................. 2
   2.5 ASSUMPTIONS AND DEPENDENCIES ............................................................................. 2

3. SPECIFIC REQUIREMENTS ................................................................................................... 2
   3.1 EXTERNAL INTERFACE REQUIREMENTS ................................................................. 2
      3.1.1 User Interfaces ............................................................................................................ 2
      3.1.2 Hardware Interfaces .................................................................................................. 2
      3.1.3 Software Interfaces .................................................................................................. 3
      3.1.4 Communications Interfaces ..................................................................................... 3
   3.2 FUNCTIONAL REQUIREMENTS ....................................................................................... 3
      3.2.1 Menu .......................................................................................................................... 4
      3.2.2 GUI .............................................................................................................................. 5
   3.3 USE CASES ....................................................................................................................... 5
      3.3.1 Use Case #1 MENU .................................................................................................. 5
      3.3.2 Use Case #2 MOVEMENT ........................................................................................ 6
   3.4 CLASSES / OBJECTS ......................................................................................................... 6
3.4.1 Class 1 FISH ........................................................................................................6

3.5 Non-Functional Requirements .................................................................................6

3.5.1 Performance ........................................................................................................6

3.5.2 Reliability ...........................................................................................................7

3.5.3 Availability .........................................................................................................7

3.5.4 Security ..............................................................................................................7

3.5.5 Maintainability ..................................................................................................7

3.5.6 Portability .........................................................................................................7

3.6 Design Constraints ................................................................................................7

3.7 Logical Database Requirements .............................................................................7

4. ANALYSIS MODELS ..............................................................................................7

4.1 Sequence Diagrams ...............................................................................................8

4.2 Data Flow Diagrams (DFD) ..................................................................................8
1. INTRODUCTION

1.1 PURPOSE

The purpose of this document is to give an outline of specifications that are to be developed throughout the course of the game development. Deepsea Survival will be marketed to all Windows and Linux based systems with the target audience of children and teens ages 12+.

1.2 SCOPE

This subsection should:

(1) Deepsea Survival (PC Game)

(2) Allow users to play as a multitude of marine animals and adventure in the dangerous ocean

(3) The application will allow for users to control a particular marine animal until they reach an objective or fail through different means.

(a) The game will have an approximate game time of roughly 10-20 minutes before the objective will be reachable. Sub-objectives such as survival will play a constant role in the game's advancement.

(b) Failures include but are not limited to starvation, bleeding, being eaten, oxygen depletion and capture

1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

For the purposes of this documentation the following abbreviations and acronyms may come up.

DSS(DeepSea Survival) Refers to the game itself

GUI(Graphical User Interface) Refers to the on screen menus and buttons that allow the user to interact with the game

1.4 REFERENCES

Currently no references or outside sources have been used although documentation will surely be placed here as development continues.

1.5 OVERVIEW

This SRS will cover a broad and detailed perspective of the overall specifications needed for developing the game application. Included are general description of functions, menus, classes, and objects that will be utilized within our game project.
2. GENERAL DESCRIPTION

2.1 GAME PERSPECTIVE

DSS is a game much like many 2d side scrolllers available today. An appropriate example of the layout and attitude of the game can be characterized by the popular iOS game Jaws Revenge.

2.2 PRODUCT FUNCTIONS

DSS will users a once in a lifetime experience of controlling small to large marine animals in an all-out effort of survival from the environment, animals, humans and time itself.

2.3 USER CHARACTERISTICS

Players of the game will have a taste of adventure and the unknown. Those who enjoy progressing and reaching various small and large objectives in a leaderboard style format will be drawn to the game.

2.4 GENERAL CONSTRAINTS

The game should be relatively light on the system in order to provide the most fluid gameplay. GUI elements must be clean, non-obtrusive and simple as to not confuse or o the target audience.

2.5 ASSUMPTIONS AND DEPENDENCIES

This Game will be dependent on Linux Based systems and Windows based systems. A typical computer with keyboard and mouse are assumed to be required by the uses to play the game.

3. SPECIFIC REQUIREMENTS

3.1 EXTERNAL INTERFACE REQUIREMENTS

3.1.1 USER INTERFACES

A Clean Simple Menu with functions to play, set options and close the game. These functions will interact with user input and deliver intended output or changes.

3.1.2 HARDWARE INTERFACES

A Computer is Required that contains a keyboard. Mouse input is optional but recommended to start off the game. This serves as the user end and links up to the application and considered user input.
3.1.3 SOFTWARE INTERFACES

Software must be readily able to dynamically size the game to different screens and formats. Input must also be handled at the highest importance to allow responsive gameplay. Handling of both user input and output will be handled by the software and properly executed to properly run the game.

3.1.4 COMMUNICATIONS INTERFACES

No Communication interfaces are needed as this game will remain standalone and single player. All communication is done solely through the user system and the application.

3.2 FUNCTIONAL REQUIREMENTS

This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.
3.2.1 FUNCTIONAL REQUIREMENT 1 MENU

3.2.1.1 The menu is the very first thing a user will be presented with during startup. It must work in every case and allow the user to easily and quickly start a game and exit a game.

3.2.1.2 Three initial input buttons will be used to minimize screen clutter. First, a ‘play game’ button to easily start a game. Second, an ‘Options’ or ‘Settings’ button to allow the user to change software related settings.

3.2.1.3 The ‘Play Game’ button will change the game state to run and begin the game session. The exit button will exit the game and its current session. Finally, the Options button will bring up pre-defined settings that can be changed before game start.

3.2.1.4 Outputs of the three Menu items will be a change of screens to begin running the game, a secondary menu to show additional changeable settings and the termination and closure of the game.

3.2.1.5 Error Handling will be done during game testing and development. Error handling on release can be submitted to the developer for inspection of bugs and errors.
3.2.2 FUNCTIONAL REQUIREMENT 2 GUI

3.2.2.1 The GUI is the forefront of the playable game. A GUI that functions properly will allow the user to navigate the game with ease with little to no prior knowledge of game functions. A GUI must contain enough information to be helpful but be clean enough to limit screen congestion.

3.3 USE CASES

3.3.1 USE CASE #1 MENU

3.3.1.1 The menu must be always accessible at all times during the gameplay. The button as well as hitting the escape key must always bring up the menu prompt in a stopped game state.

3.3.1.2 The user clicks either escape or the menu button to pause the game and bring up the menu screen in order to restart the game, reconfigure options or exit the game.
3.3.2 USE CASE #2 MOVEMENT

3.3.2.1 Preconditions:
1. The User must have started the game
2. The User has selected an animal
3. The game has initialized all game mechanics and GUI
4. The game is now running

3.3.2.2 when a player presses either the up or down arrow keys/moving mouse on top or below the fish the fish will begin to decrease or increase depth relative to its position. Using the right or left arrows or using the mouse to the left or right of the fish will accelerate the animal in the direction specified.

3.4 CLASSES / OBJECTS

3.4.1 CLASS 1 FISH

3.4.1.1 Attributes of this class will include:
   Name: String
   Health: float
   Hunger: float
   Oxygen: float
   maxdepth: Integer

3.4.1.2 Functions of this class will include HealthChange(), HungerChange(), OxygenChange() and Movement()

3.5 NON-FUNCTIONAL REQUIREMENTS

3.5.1 PERFORMANCE

Performance must be maintained throughout the gameplay. Menu Transitions must be processed in less than 1 second and mathematical calculation must not slow down movement or gameplay.
3.5.2 RELIABILITY

Reliability is another key factor we wish to maintain throughout the game. When an action is performed we not only want that action to be done quickly but we want the action to happen every time. Reliability of events, actions and changing of game states must all be at minimum 98% uptime.

3.5.3 AVAILABILITY

The Application should be available to all PC users on Linux and Windows and be functionally identical.

3.5.4 MAINTAINABILITY

The Game should be written in a manner to easily implement new and improved optimization and additions. Likewise the code should be able to spot and report most errors to developers.

3.5.5 PORTABILITY

The game should be easily portable from one computer to another and with the possibility of transforming it into an iOS or android application with relative ease.

3.6 DESIGN CONSTRAINTS

*Hardware is constrained to computers found on school grounds with limited graphics processors.*

3.7 LOGICAL DATABASE REQUIREMENTS

*A database may be used in the future to keep track of global leaderboards and statistics but all data will be maintained by the game client alone.*

4. ANALYSIS MODELS
4.1 SEQUENCE DIAGRAMS

4.2 DATA FLOW DIAGRAMS (DFD)