**Software Design Specification**

Revision – 0.1

Last printed

**Duel Reality**

**Approval Block**

|  |  |  |
| --- | --- | --- |
| **Name** | **Responsibility** | **Initials** |
| Josh Kilgore (Team Lead) | Game Mechanics & AI |  |
| Obi Atueyi | User Interface |  |
| Tom Calloway | Graphics Window |  |
| Ye Tian | Database |  |

**Abstract (Tom)**

The following is a software functional specification document for the Duel Reality personal computer game. The document fully identifies and describes both the high and low level functionality of the software without going into the design details themselves. This document attempts to serve the needs of those looking to understand the functional requirements of the game from both the user and designer perspectives.

**Revision History (Tom)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Revised By** | **Comments** |
|  |  |  |  |
|  |  |  |  |

**Table of Contents (Tom)**

**List of Figures (Tom)**

# 1. Introduction (Obi)

This document describes the functions of the Duel Reality modules, in accordance with its Architectural Specification [2].

Interactions between the user and the game are provided via the User Interface (UI). The UI is the main window that comprises the menu, toolbar, and status bar. It is through this interface that the user sets the desired game play options and receives error messages during battles.

The main window also contains a game view that comprises the map and player unit. This is the Game Graphics and it provides the user a visual representation of the state of the game during a battle.

The Game Mechanics provides the state of the game to the Game Graphics. In response to the user’s turns during battle, the Game Mechanics uses the game play options entered at the UI and interactions with the Artificial Intelligence (AI) to define the state of the game at any time.

The AI is the user’s opponent that adapts to the game level and the user’s units. It retrieves game and user information from the Database and uses this information to model an opponent suitable for the user’s experience level.

The Database stores information provided by the AI and UI. It also fetches information on request by the AI and UI. The Database provides permanent storage of such information for use during future program runs.

## 1.1 Document Purpose (Tom)

The purpose of this document is to provide the functional specifications for the development of the Duel Reality turn based strategy role playing game.

## 1.2 Product Scope (Josh)

This game is meant to be run on a solitary Windows PC. User interactions will take place with a standard mouse and keyboard. Graphics will be displayed on the screen and sound played through the standard sound output device of the computer.

## 1.3 Terminology

Sprite A two-dimensional pre-rendered figure

Unit An individual infantry

Widget An interactive feature pertaining to user interface graphics

## 1.4 Acronyms

AI Artificial Intelligence

AP Action Point

XP Experience Point

GUI Graphical User Interface

UI User Interface

TBS Turn-Based Strategy

# 2.0 Overall Description

//INSERT OVERARCHING DESCRIPTION OF GAME.

## 2.1 Module Description

//BRIEF Description of Module – will be expanded in section 3.0 (what it is, what it consists of, what it does)

2.1.1 UI

2.1.2 Graphics

2.1.3 Game Mechanics

2.1.4 Database

## 2.4 Design Constraints (Josh)

Some of the limiting factors in our design and execution of this project include the time limit of having the project due on a certain date and being limited to C++ for development language.

## 2.5 Assumptions and Dependencies (Josh)

We assume that C++ object oriented programming would be sufficient to do game design. That using QT tools adds value to this process, that time constraints would be   
sufficient to make a working game, and that we were assuming a Windows   
environment for deployment.

2.6 Design Environment and Tools

# 3.0 Software Design

## 3.1 User Interface Classes(Obi)

//DISCUSS CLASSES IN THIS MODULE AND HOW THEY FIT TOGETHER INTERNALLY & WHAT THEY INTERACT WITH FROM OTHER MODULES.

### Description

Although the UI module is, in general, the application display to the user, it will be separate from the game graphics. The UI enables the user to configure game options for each level of the game. The menu options are used to start, load, save or quit a game. During a battle, the user uses the tool-bars to move or perform actions on units. Any invalid moves or actions during battle are indicated on the status bar.

## 3.2 Game Graphics (Tom)

//DISCUSS CLASSES IN THIS MODULE AND HOW THEY FIT TOGETHER INTERNALLY & WHAT THEY INTERACT WITH FROM OTHER MODULES.

### Description (Tom)

The main dialog of the Duel Reality PC game features a game graphics window, which represents the map and units currently in play. As such, the graphics window is the user’s primary feedback regarding the current state of the game. The graphics window will include a background image, two-dimensional sprites to represent units, and a grid representing the discrete locations of the map. The game graphics window will also feature sounds and visual effect to represent significant game events.

## 3.3 Game Mechanics (Josh)

//DISCUSS CLASSES IN THIS MODULE AND HOW THEY FIT TOGETHER INTERNALLY & WHAT THEY INTERACT WITH FROM OTHER MODULES.

### Description

The Game Mechanics module comprises the rules about how the different parts of the program work together to present the player with a full experience. In this section are defined the units, their interaction with the board and the players, the basic rules of the game including movement and actions and the battle process as shown in Figure 2‑2.

## 3.4 AI (Josh)

//DISCUSS CLASSES IN THIS MODULE AND HOW THEY FIT TOGETHER INTERNALLY & WHAT THEY INTERACT WITH FROM OTHER MODULES.

### Description

As shown in Figure 3‑2, the AI will consist of a set of decisions and actions which will mimic the actions of another human opponent for the single Player to play against. The AI will operate under the same constraints as the Player in terms of gameplay, but will not have the ability to automatically upgrade its capabilities. This AI model will be sufficient for the purposes of this game, and can scale easily into a medium complexity.

.

Figure ‑: AI flow

## 3.5 Database (Ye)

//DISCUSS CLASSES IN THIS MODULE AND HOW THEY FIT TOGETHER INTERNALLY & WHAT THEY INTERACT WITH FROM OTHER MODULES.

### Description (Ye)

The game database contains all of the tables and data records, which function like a background support to other modules. It is a collection of the names, parameters, status of all units, and the game content such as maps. The data is retrieved and overwritten in real-time game going and asynchronous backup.

The database module interacts with other modules, but does not affect them. It ***shall*** provide operation function by other modules.

4.0 CLASS MEMBER FUNCTIONS

//EXAMPLE SHOWS DISCRIPTION OF EACH OF THE FUNCTIONS UNDER EACH CLASS ALONG WITH A STATE DIAGRAM OF HOW THE CLASS FLOWS, IF THAT’S USEFUL. SO SOMETHING ALONG THOSE LINES HERE

4.1 UI Class Member Functions

4.2 Graphics Member Functions

4.3 Game Mechanics Member Functions

4.4 Database Member Functions

5.0 TESTING

5.1 Unit Testing

5.2 System Testing

# 6.0 References (Ye)

Team Gold: Josh Kilgore, Obi Atueyi, Thomas Calloway, Ye Tian, "Duel Reality: A Turn-Based Battle Strategy Game", Proposal, 02/19/2010.

Team Gold: Josh Kilgore, Obi Atueyi, Thomas Calloway, Ye Tian, "Duel Reality: A Turn-Based Battle Strategy Game", Software Architecture Specification, 03/20/2010.