***Magic Realm (1979)***

**Team 6**

**Couturier, Stefan**

**Lam, Ryan**

**Sayyeau, Christopher**

**COMP 3004**

**Date Due: April 8th, 2015**

#LEAVE orange TILL END

**Table of Contents**

1. **Introduction**
   1. **Motivation**
   2. **Terminology**
2. **Game Rules**
3. **Requirements**
   1. **Functional Requirements**
   2. **Assumptions**
4. **Use Cases**
   1. **Use Case Diagram**
   2. **Use Cases**
   3. **Responsibilities**
   4. **Bound and Unbounded Use Case Maps**
5. **Design Decisions**
   1. **Decisions**
   2. **Structural Model (Magic Realm)**
   3. **Structural Model (Bots)**
6. **Object Specifications**
7. **Interaction Diagrams**

**1 - Introduction**

Magic Realm was first realeased in 1979 by Avalon Hill. It was designed as a fantasy adventure board game by Richard Hamblen. It is a very complex role-playing war game that can be played from 1-16 players over the course of several hours. The rules were rereleased in a second edition in 1986 and a third edition was released by fans after the company went out of business in 1998.

* 1. **- Motivation**

This report will briefly cover our interpretation and implementation of this board game. It will outline the functional and non-functional requirements of the software, as well as its use cases and the assumptions made throughout its construction.

The goal of the project was to turn Magic Realm into a networked computer game. The official 3rd edition rules were followed as closely as possible to create the software.

* 1. **- Terminology**

A number of key terms are used throughout this report, which will be listed alphabetically in this section. Definitions are included, however if the term is covered in the Game Rules it will be covered in its own section (2).

|  |  |
| --- | --- |
| Term | Definition |
| Chit  Example 2 | This is a token with symbols on it representing some tangible object in the game universe.  Example 2 |

SEE EXAMPLE

1. **- Game Rules**

The official third edition game rules for Magic Realm.

We will copy in all pages that we reference later

1. **– Requirements**

This Section contains what features must be implemented. Each requirement has its own identifier and a short description. The source of traceability may either be from the official game rules, assumptions, other requirements, or team decisions.

* 1. **– Functional Requirements**

Functional requirements define what behavior and functionality the software must have. They have been categorized below, based on different game states or areas of functionality. All of the functional requirements specified in the CorrectionGrid file provided are represented below.

|  |  |  |
| --- | --- | --- |
| **ID** | **Functional Requirements** | **Traceability** |
| FR-1 | Display correct titles of the board. | Location in game rules elsewhere in this document |
| FR-2 | Board is scrollable for each client. | NEED TO FILL |
| FR-3 | Networking capabilities. Capable of having multiple players. The actions of one player are represented in each window of his opponents. |  |
| FR-4 | Initial character selection offers 2 or more distinct characters. |  |
| FR-5 | Cheat Mode allows user to assign 4 types of Map Chits( V, W, M, C) to specific tiles. |  |
| FR-6 | Can give the 5 Map Chits to Lost Castle. |  |
| FR-7 | Can give the 5 Map Chits to Lost City. |  |
| FR-8 | Can manually place Lost Castle. |  |
| FR-9 | Can manually place Lost City. |  |
| FR-10 | Sound and Warning chits are correctly displayed to clients. |  |
| FR-11 | Support choosing starting location for characters. |  |
| FR-12 | Support of moving within the tile. |  |
| FR-13 | Support of moving across tiles. |  |
| FR-14 | Support Captain’s extra phase when he is in a dwelling. |  |
| FR-15 | Support Amazon’s extra Move Phase. |  |
| FR-16 | Support the choice of dice roll results in cheat mode. |  |
| FR-17 | Support hiding successfully (1-5). |  |
| FR-18 | Support hiding unsuccessfully (6). |  |
| FR-19 | Selection of fight anr move counters for combat. Then resolve combat. |  |
| FR-20 | Can choose shield direction. |  |
| FR-21 | Support MultiRounds of combat between 2 characters. Resulting in a death or no death. |  |
| FR-22 | Support a 2 effort limit per round of combat. |  |
| FR-23 | Support random order of player turns. |  |
| FR-24 | Support Dwarf only having 2 phases a day. |  |
| FR-25 | Support Black Knight |  |
| FR-26 | Support the recording of players phases and then implementing them after all players have finished recording. |  |
| FR-27 | Support the viewing of objects in clearing. |  |
| FR-28 | Support the alerting of weapons. |  |
| FR-29 | Support using diferent stats of an alerted weapon. |  |
| FR-30 | Support combat with unalerted weapons. |  |
| FR-31 | Support fatiguing counters. |  |
| FR-32 | Support wounding counters. |  |
| FR-33 | Support resting fatigued counters. |  |
| FR-34 | Support resting wounded counters. |  |
| FR-35 | Support locating of treasure. |  |
| FR-36 | Support looting of treasure. |  |
| FR-37 | Support Great treasures. |  |
| FR-38 | Support specification of treasure’s notoriety and fame. |  |
| FR-39 | Player’s current gold, fame, notoriety updated regularly. |  |
| FR-40 | Support cheat mode for searching tables. |  |
| FR-41 | Support locating of secret paths and passages. |  |
| FR-42 | Support of Cloak of Mist. |  |
| FR-43 | Support of Magic Spectacles. |  |
| FR-44 | Support Cave phase restriction. |  |
| FR-45 | Support of random dice rolls in normal mode. |  |
| FR-46 | Support monsters prowling to other clearings. |  |
| FR-47 | Support missile attacks. |  |
| FR-48 | Support viewing of clearing values. |  |
| FR-49 | Support monsters. |  |
| FR-50 | Support character death. Creating pile of his belongings. |  |
| FR-51 | Support Looting of his belongings |  |
| FR-52 | Support victory point selection. |  |
| FR-53 | Support victory point calculation. |  |

* 1. **– Assumptions**

While building the software many assumptions were made about the rules so that the rules could be captured the software requirements. Many of the requirements above will trace to these items. The following table will identify them along with their justification.

|  |  |  |
| --- | --- | --- |
| **ID** | **Assumption** | **Justification** |
| A-1 | TABLE SEE EXAMPLE |  |
| A-2 |  |  |
| A-3 |  |  |

1. **– Use Cases**

A use case covers a scenario. It details the path of events needed for this scenario.

* 1. **– Use Case Diagram**

DIAGRAM AND EXPLANATION

* 1. **– Use Cases**

Each use case is detailed in its own table, which describes its sequence of events.

TABLES SEE EXAMPLEs

* 1. **– Responsibilities**

These are created from the use cases above; they will be used in the use case maps in section 4.4 and 4.5. The reference for the respective use case is on the right.

TABLE SEE EXAMPLE

**4.4 – Bound and Unbounded Use Case Maps**

The bound maps correspond to the unbounded ones directly below them; while the unbounded maps correspond to section 4.2. The Triggering and resulting events are labeled in the tables following them.

SEE Example

1. **– Design Decisions**

This section will cover the design decisions that were taken with respect to classes and objects chosen for the system. The UML diagram will be in section 5.2.

* 1. **– Decisions**

SEE EXAMPLE

* 1. **– Structural Model (Magic Realm)**

SEE EXAMPLE

* 1. **– Structural Model (Bots)**

This diagram involves the implementation of artificial players in the software. ??!!PRETTY SURE WE DON”T USE THESE

SEE EXAMPLE

1. **– Object Specifications**

SEE TABLES, IN EXAMPLE

1. **– Interaction Diagrams**

Below are the UML Unteraction Diagrams. Each correspond’s to the previous bound use case maps in section 4.5.

SEE TABLES, IN EXAMPLE