Software Requirements Specification

for

Discworld: Ankh Morpork

Version 1.0 approved

Prepared by

Varun Pattiah, Taiseem Ahmed, Sumit Sarkar, Khurram Shehzad

Concordia University

Feb. 15, 2015

Table of Contents

Table of Contents i

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. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

4. System Features 4

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The purpose of this SRS document is to give detailed description of the functionalities of Discworld Ankh-Morpork build1, which is board game. This document will give an insight into intended features, glimpse of user interface, software and hardware dependencies.

## Document Conventions

This document has some terminologies which a user might not be familiar with. See Appendix for these terms.

## Intended Audience and Reading Suggestions

This document is intended for the developers of the project Discworld Ankh-Morpork. The readers can focus on section 1 and 2 for taking brief overview of project.

## Product Scope

The project Discworld is software version of board game which is a multiplayer game intended for entertainment.

## References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

# Overall Description

## Product Perspective

Discworld Ankh-Morpork is based on playing of cards, placing minions on different areas on board. Two to four players can play this at a time who are given secret personality card which describes winning condition for that player. For build 1 we are not going much into details of the cards, characters, action associated with cards and random events.

## Product Functions

The various features and functionalities of the project are described below.

1. Viewing current status of game board
2. Minions with their colors in different areas of board
3. Buildings and trouble marker presence in each area
4. Demons and trolls existence
5. Status of players inventory
6. No. of minions left
7. No. of buildings currently established and areas cards holding
8. Money in dollars
9. Players cards which are green and brown
10. Loading previously saved games
11. Start new game

## User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

## Operating Environment

The build 1 of the project does not feature real application of the game, hence it will available to be executed on eclipse using JDK.

## Design and Implementation Constraints

The first build does not allow multiple players to play from multiple machine i.e. all players are expected to play on a single system. Lack of time is measure concern in build 1, so we will be integrating real gaming in later versions.

## User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

## Assumptions and Dependencies

Third party jar file “java-json.jar” has been used to parse game state stored in json format.

# External Interface Requirements

## User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Current Status of Board (Name of the game)** | | | | | |
| City Areas(ID) | Minions | Trouble Marker? | Building? | Trolls | Demons |
| The Shades(1) |  |  |  |  |  |
| Dolly Sisters(2) |  |  |  |  |  |
| The Scours(3) |  |  |  |  |  |
| Dimwell(4) |  |  |  |  |  |
| Nap Hill(5) |  |  |  |  |  |
| Longwall(6) |  |  |  |  |  |
| The Hippo(7) |  |  |  |  |  |
| Dragon’s Landing(8) |  |  |  |  |  |
| Isle of Gods(9) |  |  |  |  |  |
| Small Gods(10) |  |  |  |  |  |
| Seven Sleepers(11) |  |  |  |  |  |
| Unreal Estates(12) |  |  |  |  |  |

Select Number of Players b/w 2 and 4 from Dropdown list

Player 1

Player 1

Player 1

New Game

Player 2

Enter the name of players

Player 2

Each players is randomly assigned a color and a personality which is secret

At the start of the game, inventory of a player look like this

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Player 1 Current Inventory** | | | **Personality Card(Not displayed to other players)** | | | |
| Minions | Buildings | City Area Cards | | Money($) | Player Cards (Secret) | |
| 12 | 0 |  | | 12 | Green | Brown |
| 2, 23,4,16,7 |  |

After placing minions in different areas of the board. Game board status update

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Current Status of Board (Name of the game)** | | | | | |
| City Areas(ID) | Minions | Trouble Marker? | Building? | Trolls | Demons |
| The Shades(1) |  |  |  |  |  |
| Dolly Sisters(2) |  |  |  |  |  |
| The Scours(3) |  |  |  |  |  |
| Dimwell(4) |  |  |  |  |  |
| Nap Hill(5) |  |  |  |  |  |
| Longwall(6) |  |  |  |  |  |
| The Hippo(7) |  |  |  |  |  |
| Dragon’s Landing(8) |  |  |  |  |  |
| Isle of Gods(9) |  |  |  |  |  |
| Small Gods(10) |  |  |  |  |  |
| Seven Sleepers(11) |  |  |  |  |  |
| Unreal Estates(12) |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Player 1 Current Inventory** | | | **Personality Card(Not displayed to other players)** | | | |
| Minions | Buildings | City Area Cards | | Money | Player Cards | |
|  |  |  | |  | Green | Brown |
|  |  |

## Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

## Software Interfaces

This project is to be developed in Java (Java Development Kit) using Eclipse IDE(Integrated Development Environment).

## Communications Interfaces

The first build does not any communication feature but we are expecting to include this in future versions.

# System Features

The build 1 is supposed to contain certain core features which includes game launching, game loading and saving the current game for later use.

## Launching a New Game

4.1.1 Description and Priority

Launching a new game will yield to initialize the data structure of the game.

4.1.2 Stimulus/Response Sequences

1. User is prompted to select number of players participating.
2. Players are asked to enter their names.
3. Each player is assigned a color randomly after which a secret personality card is given to them which they follow to win game. This build does not contains action associated with each card.
4. Every player is given a set of minions and buildings as per color to play, along with troll, demons, trouble marker on board.

4.1.3 User Requirements

The players are supposed not to look on system screen for checking the personality of others.

## Loading Status of Game

4.1.1 Description and Priority

This feature will display the current status of a game, which will be used by a player to decide its moves.

4.1.2 Stimulus/Response Sequences

1. User will be asked to select the game for which status is to be loaded on screen.
2. A table will display the current status of the game board which describes which color minions are in which area, whether that area is having a building, troll, demons or trouble marker.
3. A separate table will display the status of inventory of all individuals.
4. The current player is asked to make a valid move.
5. Updating inventory and status of game board on each move.

4.1.3 User Requirements

The players are supposed not to look on system screen for checking the personality of others.

## Saving Current Status of Game\*

4.1.1 Description and Priority

This feature will save the current status of a game into a file for later use.

4.1.2 Stimulus/Response Sequences

4.1.3 User Requirements

The players are supposed not to look on system screen for checking the personality of others.

# Other Nonfunctional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

## Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

# Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>