Software Requirements Specification

for

Project E

Version 1.0

Prepared by CA314 students

27/10/2011

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This document provides all of the requirements for the On-line Board Game Family: Noughts & Crosses and Draughts.

1.2 Definitions, Acronyms and Abbreviations

OBGF – an acronym for the name of the application – “On-line Board Game Family”

Video game – a game played on a computer

Game session – interactions that happen within the system in the space of time between when two players first connect to a server to play together; and either one or both disconnect.

1.3 Intended Audience and Reading Suggestions

This document is intended to be read by software developers who wish to maintain or extend the application for new board games or port the application to new systems. Non-technical users of the application should instead read the readMe.txt file supplied with the game.

1.4 Project Scope

This document covers the requirements for release 1.0 of OBGF. Mention will be made to possible improvements that may be made in future versions of the software. The purpose of this is to allow to easily extend the system in the future to include other, similar board games.

1.5 References

The Software Requirements Template that was used to create this document was made available through – http://www.processimpact.com/ and can be found at this address - http://www.processimpact.com/process_assets/srs_template.doc/

The rules of Tic-Tac-Toe - http://www.exploratorium.edu/brain_explorer/tictactoe.html

The rules of draughts according to the 1991 EDA - http://draughts.ic.cz/rules.php

Dublin City University module CA314 Object Oriented Analysis and Design - https://www.dcu.ie/registry/module_contents.php?function=2&subcode=CA314

The idea for the system was provided by Dr Liam Tuohey – http://www.computing.dcu.ie/~ltuohey/index.htm - Dr Liam Tuohey’s *homepage*

2. Overall Description

2.1 Product Perspective

OBGF is a brand new self contained product that was developed in 2011 as part of the course work for CA314. The idea for the product was provided by the lecturer for the course – Dr Liam Tuohey on his website.

2.2 Product Features

OBGF will allow two users to connect and play a selected board game over the internet or a similar network. Originally, the choice of games will be limited to two, with a possibility of more in future versions of the product. The application will run and be controlled through a text-based system like a terminal window (depending on the platform the name name may vary). To play together two users will have to connect to the same server, which will handle the interaction. The product will allow the users to specify their own usernames for the game session.

2.3 User Classes and Characteristics

Average player – a person who does not play board games regularly and does not have extensive knowledge on the rules of the games or the terminal commands that are used to control this application. May require to consult the readMe.txt before playing.

Application developer – may use the system to test it or to see how it works before introducing further extensions. Has extensive knowledge of the source code and may have adequate knowledge of the board games.May need to read both this document and the readMe.txt for additional design-related details.

Board-game specialist – a person with extensive knowledge of board games, who is unaware of how the application was built. This is the favoured user class and the product was design to accommodate it. Such person may need to consult the readMe.txt document before using the product for the first time, but isn't likely to find any use for it afterwards.

2.4 Operating Environment

The software is originally designed to run under the Linux family of operating systems. A network connection will be required to run the application and play wit other people.

2.5 Design and Implementation Constraints

The strict deadline is the biggest constraint that was placed on the development team. No other limitations were identified in the current version of the document.

2.6 User Documentation

readMe.txt - a text file that will describe the rules of the various board games included in the package and introduce the various text commands that the user may enter. A .pdf version of the file with helpful pictures may also be included with version 1.0, but depending on the time constraints may be delayed until a later release.

2.7 Assumptions and Dependencies

It is assumed that all team members will be healthy enough during the development process to work towards satisfying all of the requirements.

3. System Features

3.1 Connect to a remote server

3.1.1 Description and Priority

This feature will allow a user to make a connection to a server by entering its address after being asked for it by the application.

PRIORITY: High

3.1.2 Functional Requirements

REQ-1.1: The application must have access to a network – it cannot be blocked by a firewall.

REQ-1.2: The application must have a working server-client system in place that allows for a two way communication.

REQ-1.3: If the user enters an incorrect or inacessible address, he will be informed about it and asked to re-enter it.

3.2 Choose a game

3.2.1 Description and Priority

This feature will allow the user to pick from a list of games using text-commands.

PRIORITY: High

3.2.2 Functional Requirements

REQ-2.1: The games that are given as options must be complete and included in the system.

REQ-2.2: If the user enters an incorrect input, he will be asked to re-enter his preference.

3.3 Make a move

3.3.1 Description and Priority

This feature will allow the user to enter an “input” that the game will then handle. An “input” can be any action allowed by the game, e.g. moving a pawn.

PRIORITY: High

3.3.2 Functional Requirements

REQ-3.1: The system must be able to handle and detect inorrect input – allowed vs. disallawed actions based on the rules of the games. If the input is deemed incorrect, the user will be asked to re-enter it.

REQ-3.2: The user must be connected to a server and be in a game session with another player make a move.

3.4 Disconnect form the server

3.4.1 Description and Priority

This feature will allow the user to disconnect from a server and effectively end the game session.

PRIORITY: High

3.4.2 Functional Requirements

REQ-4.1: The system must correctly handle the request to disconnect – the opposing player must be informed that he won the game and the player who chose to leave should be correctly disconnected form the server, allowing another player to join the server.

REQ-4.2: The user needs to have an established connection to a server to carry out this action.

Primary Class List

class Client

```
Attributes:
    String name;
Methods:
    void act();
    void getMessage();
    boolean join();
```

class Server

```
Attributes:
    Client client1;
    Client client2;
    Game gameSession;
Methods:
    void acceptConnection();
    void closeConnection();
    void sendMessage(String, Client);
```

class Game

```
Attributes:
    Board board;
Methods:
    int checkGame();
    void endGame();
    void startGame();
```

class Board

```
Attributes:
    int noPieces;
    Piece [] [] pieceArray;
    int size;
Methods:
    int size();
```

class Piece

```
Attributes:
    Sting name;
    String type;
Methods:
    void move();
    boolean validMove();
```

Class Diagram