|  |
| --- |
| Drexel University |
| Acceptance Test Plan |
| DragonChess |

|  |
| --- |
| Group 4 |

Table of Contents

[Revision History 3](#_Toc223883229)

[1 Introduction 4](#_Toc223883230)

[1.1 Background 4](#_Toc223883231)

[1.2 References 4](#_Toc223883232)

[1.3 Definitions, Acronyms & Abbreviations 4](#_Toc223883233)

[2 Test Approach and Constraints 4](#_Toc223883234)

[3 Test Assumptions and Exclusions 4](#_Toc223883235)

[4 Entry and Exit Criteria 6](#_Toc223883236)

[5 Testing Participants 7](#_Toc223883237)

6 Testing Project Test Cases…………………………………………………………………………………………………………………….8

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Change | Version |
| Dan DeCamillo | Mar 18 2009 | Created | 1.0 |

# 1 Introduction

## 1.1 Background

This document provides the plan in which the testing of the DragonChess system as required for the acceptance test plan will illustrate. DragonChess is a lightweight portable network chess game.

## 1.2 References

Software Requirements Specification for DragonChess

## 1.3 Definitions, Acronyms & Abbreviations

## Software Requirements Specification - A document which describes the behavior of a system.

## Functional Requirements - internal workings of the software.

## Unit Tests – validates individual parts of the code.

## Integration Test - tests how all the different modules of a software system fit and work together with each other.

## System Test - System Evaluates the system’s compliance with its specified requirements

# 2 Test Approach and Constraints

2.1 Test Objective

The test plan will prompt the user to evaluate and test the chess game to see whether it complies with the specifications that were given in the software requirements document.

2.2 Test Structure

The Acceptance Test plan will consist mostly of unit tests that were designed to test the components of the system. There will be some integration and system testing but for the most part the testing will be unit based.

It is important to note that the execution of the unit tests and other tests are pertinent to the display of this testing information to the end user and client.

# 3 Test Assumptions and Exclusions

3.1 Assumptions

It is assumed that all tests have been completed before the action of using the Acceptance Test Plan. This included system functionality, usability, and proper documentation.

3.3 Exclusions

It is assumed that everything that is not addressed here has been addressed through a different medium not included in this documentation. This means non-functional requirements, and the structure of the source code is not tested.

# 4 Entry and Exit Criteria

4.1 Entry Criteria

The Acceptance Test Plan can be started when the following are complete:

• DragonChess has been tested through Unit Tests, Integration Test and System Test.

• The testing environment is sufficient to the end client or user.

• All documentation is up to date

• The newest version of DragonChess has been set up to use.

• The tester must know how to play chess.

4.2 Exit Criteria

Any of the following listed actions should stop the test plan.

• Requirements tested without any unexpected results. (Success)

• One of the tests fail. (Failure)

• The client and designer decide the test is over or does not meet requirements. (Failure)

# 5 Testing Participants

5.1 Roles and Responsibilities

Testing Leader: Dan DeCamillo

Client Representative: The person who issued the requirements.

Tester: A person to execute the actual tests.

5.2 Training Requirements

All peoples involved in this testing should be familiar with both the game of chess, as well as the program itself in its functionality.

5.3 Problem Reporting

All problems that arise from this Acceptance Test Plan will be documented by the Testing Leader and then will be addressed by the project head to make corrections to the software pending a team meeting.

5.4 Progress Reporting

This Acceptance Test Plan is only to be executed one time for the end client and will be finished by the Testing Leader and the results will be given to the project leader.

# 6 Testing Project Test Cases

6.1 Introduction

The Test cases will be listed below. Only unit tests will be listed. The format will be as follows.

Name – Name of Test

Precondition – Things that are needed to run the test.

Actions – what is tested.

Post Conditions – what is the result of the test.

6.2 Test Cases

6.2.1 Log In

|  |  |
| --- | --- |
| Precondition | Tester has a computer with an internet connection and the DragonChess source code. |
| Action | The Tester enters their name and color. |
| Post Condition | The game waits for the other player or the game starts. |

6.2.2 Start Game

|  |  |
| --- | --- |
| Precondition | Tester has a computer with an internet connection and the DragonChess source code. Tester is possible to host a game on a specific socket. |
| Action | Enters Name and clicks host game. |
| Post Condition | User waits for the other player to connect. |

6.2.3 Connect to Game

|  |  |
| --- | --- |
| Precondition | Tester has a computer with an internet connection and the DragonChess source code. Tester is permitted to connect to a game on a socket. |
| Action | Enter name and press connect to game and input IP address. |
| Post Condition | The tester is connected to a game and the game starts. |

6.2.4 Move a Piece

|  |  |
| --- | --- |
| Precondition | Tester has a computer with an internet connection and the DragonChess source code. The tester is logged in and the game has started. |
| Action | Move any piece by entering the correct coordinates. |
| Post Condition | The piece moves in accordance to the game logic. |

6.2.5 Win Game

|  |  |
| --- | --- |
| Precondition | Tester has a computer with an internet connection and the DragonChess source code. The tester is logged in and the game has started. |
| Action | The tester moves a piece to achieve checkmate status in the game logic. |
| Post Condition | The tester wins. |

6.2.6 Save Game

|  |  |
| --- | --- |
| Precondition | Tester has a computer with an internet connection and the DragonChess source code. The tester is logged in and the game has started. |
| Action | The tester will choose from the menu to save a game. |
| Post Condition | The proper save game file will be created in the PGN format for later use. |

6.2.7 Load Game

|  |  |
| --- | --- |
| Precondition | Tester has a computer with an internet connection and the DragonChess source code. The tester and another player are logged in and they want to load a game. |
| Action | The player with the PGN save file clicks load and selects the file. |
| Post Condition | The game is loaded to the saved position. |

6.2.8 Run Unit Tests

|  |  |
| --- | --- |
| Precondition | Tester has the DragonChess source code and a testing platform. |
| Action | Run the JUnit Tests |
| Post Condition | All tests should pass. |

6.2.9 End Game

|  |  |
| --- | --- |
| Precondition | Tester has a computer with an internet connection and the DragonChess source code. The tester is logged in and the game has started. |
| Action | The tester exits the game. |
| Post Condition | The other user is prompted to end the game and the game exits. |

6.2.10 Draw

|  |  |
| --- | --- |
| Precondition | Tester has a computer with an internet connection and the DragonChess source code. The tester is logged in and the game has started. |
| Action | The tester decides to lay down his King, which is a forfeit. He selects draw from the menu. |
| Post Condition | The other player is prompted to accept the draw and if he does the game ends. |