SYSTEMS AND SOFTWARE REQUIREMENTS SPECIFICATION (SSRS) FOR

Dots & Boxes Game



Version 1.0

2/27/2013

Prepared for:

IT 484   
Minnesota State University - Mankato

Prepared by:

Russell Walters, Chisomo Banda, Lindiwe Hove, Matt Ruppert Dot and Boxes Game Application SSRS

**RECORD OF CHANGES**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Change number | Date completed | Location of change (e.g., page or figure #) | **A M D** | Brief description of change | Approved by (initials) | Date Approved |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

\***A** - ADDED **M** - MODIFIED **D** - DELETED

TABLE OF CONTENTS

Contents

[1.1 IDENTIFICATION 4](#_Toc350113394)

[1.2 PURPOSE 4](#_Toc350113395)

[1.3 SCOPE 4](#_Toc350113396)

[1.4 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS 5](#_Toc350113397)

[1.5 REFERENCES 5](#_Toc350113398)

[1.6 OVERVIEW AND RESTRICTIONS 5](#_Toc350113399)

[1.7 PRODUCT PERSPECTIVE 7](#_Toc350113400)

[1.8 PRODUCT FUNCTIONS 7](#_Toc350113401)

[1.9 USER CHARACTERISTICS 7](#_Toc350113402)

[1.10 CONSTRAINTS 7](#_Toc350113403)

[1.11 ASSUMPTIONS AND DEPENDENCIES 7](#_Toc350113404)

[1.12 SYSTEM LEVEL (NON-FUNCTIONAL) REQUIREMENTS 8](#_Toc350113405)

[1.12.1 Site dependencies 8](#_Toc350113406)

[1.12.2 Safety, security and privacy requirements 8](#_Toc350113407)

[1.12.3 Performance requirements 8](#_Toc350113408)

[1.12.4 System and software quality 8](#_Toc350113409)

[1.12.5 Packaging and delivery requirements 8](#_Toc350113410)

[1.12.6 Personnel-related requirements 8](#_Toc350113411)

[1.12.7 Training-related requirements 8](#_Toc350113412)

[1.12.8 Logistics-related requirements 8](#_Toc350113413)

[1.12.9 Other requirements 8](#_Toc350113414)

[1.12.10 Precedence and criticality of requirements 9](#_Toc350113415)

[1.13 EXTERNAL INTERFACE REQUIREMENTS 10](#_Toc350113416)

[1.13.1 Hardware Interfaces 10](#_Toc350113417)

[1.13.2 Software Interfaces 10](#_Toc350113418)

[1.13.3 User Interfaces 10](#_Toc350113419)

[1.13.4 Other Communication Interfaces 10](#_Toc350113420)

[1.14 SYSTEM FEATURES 11](#_Toc350113421)

[1.14.1 Use Case Diagrams 11](#_Toc350113422)

[1.14.2 Use Case Specifications 12](#_Toc350113423)

[1.14.3 System feature F2: Human versus Human play 13](#_Toc350113424)

[1.14.4 System feature F3: Human versus Computer play 14](#_Toc350113425)

[1.14.5 System feature F6: Line colored for each player 15](#_Toc350113426)

[1.14.6 System feature F7: Completed boxes filled-in with player color code 15](#_Toc350113427)

[1.14.7 System feature F8: Winner determined and final score displayed 16](#_Toc350113428)

[1.14.8 System feature F9: When game play is over, the player can choose to play again 16](#_Toc350113429)

[1.14.9 System feature F10: Ability to quit the game at any time 17](#_Toc350113430)

[1.14.10 System feature F12: Human versus Computer play will allow an easy mode 17](#_Toc350113431)

[1.14.11 System feature F13: Human versus Computer play will allow an intermediate mode 18](#_Toc350113432)

[1.14.12 System feature F1: Start the Game 20](#_Toc350113433)

[1.14.13 System feature F5: Color coding for each player 21](#_Toc350113434)

[1.14.14 System feature F4: Real-time score during play 22](#_Toc350113435)

[1.14.15 System feature F11: Ability to restart the game at any time 24](#_Toc350113436)

[1.14.16 REQUIREMENTS TRACEABILITY 25](#_Toc350113437)

1. Introduction

## IDENTIFICATION

The software system being considered for development is referred to as Dots & Boxes. The customer providing specifications for the system is Professor Schilling. The ultimate customer, or end-user, of the system will be avid and casual gamers. This is a new project effort, so the version under development is version 1.0.

## PURPOSE

The purpose of the system under development is to create a fun and challenging one or two person strategy game. While the system will be used by gamers, this document is intended to be read and understood by MSU IT software designers and coders. The document will also be vetted or approved by Professor Schilling.

## SCOPE

The purpose of this project is to create a software system to play the game of Dots and Boxes. The original *Dots and Boxes* is a paper and pencil game that we propose to migrate to a computer form. Given a square or rectangular array of dots, two players take turns joining two adjacent dots with a horizontal or vertical line. When such a move adds the fourth side of a box, the player who did the deed claims the box (marking it as hers in some fashion) and must take an extra turn. A player who can complete a box is not obliged to do so. The game ends when all the boxes are taken. The player who closed more boxes.

There are two versions of the game:

* The first version will allow two humans to play against each other.
* The second version will allow one human to play against the software game.

It will be available to the general public with no written or express warranty.

## DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

|  |  |
| --- | --- |
| **Term or Acronym** | **Definition** |
| Alpha test | Limited release(s) to selected, outside testers |
| Beta test | Limited release(s) to cooperating customers wanting early access to developing systems |
| Final test | aka, Acceptance test, release of full functionality to customer for approval |
| DFD | Data Flow Diagram |
| SDD | Software Design Document, aka SDS, Software Design Specification |
| SRS | Software Requirements Specification |
| SSRS | System and Software Requirements Specification |
| DAB | Dots & Boxes |
| PLAY Framework | PLAY framework is a framework used to create applications in Java |
| Processing Language | Open source programming language and environment for people who want to create images, animations, and interactions |
|  |  |

## REFERENCES

Professor Schilling, Susan, “Dots and Boxes Project Description”, DB-SE Project.doc, February 2013.

Professor Schilling, Susan, “Project schedule for IT 484/584, Spring 2013”, Project schedule Sp2013, February 2013.

Dots and Boxes implementation available on the web at the following URL: <http://www.math.ucla.edu/~tom/Games/dots&boxes.html>

The lightweight Play Framework for Java and Scala development available at the following URL: <http://www.playframework.com/>

The Open Source Processing Language web site available at the following URL: <http://www.processing.org/>

The Open Source Processing Language for the Web – Processing.JS web site available at the following URL: <http://processingjs.org/>

## OVERVIEW AND RESTRICTIONS

This document is for limited release only to MSU IT personnel working on the project and Professor Schilling.

Section 2 of this document describes the system under development from a holistic point of view. Functions, characteristics, constraints, assumptions, dependencies, and overall requirements are defined from the system-level perspective.

Section 3 of this document describes the specific requirements of the system being developed. Interfaces, features, and specific requirements are enumerated and described to a degree sufficient for a knowledgeable designer or coder to begin crafting an architectural solution to the proposed system.

Section 4 provides the requirements traceability information for the project. Each feature of the system is indexed by the SSRS requirement number and linked to its SDD and test references.

Sections 5 and up are appendices including original information and communications used to create this document.OVERALL DESCRIPTION

## PRODUCT PERSPECTIVE

## PRODUCT FUNCTIONS

|  |  |
| --- | --- |
| **Feature ID** | **Feature** |
| F1 | Starting the game |
| F2 | Human versus Human play |
| F3 | Human versus Computer play |
| F4 | Real-time score during play |
| F5 | Color coding for each player |
| F6 | Line colored for each player |
| F7 | Completed boxes filled-in with player color code |
| F8 | Winner determined and final score displayed |
| F9 | When game play is over, the player can choose to play again |
| F10 | Ability to quite the game at any time |
| F11 | Ability to restart the game at any time |
| F12 | Human versus Computer play will allow an easy mode |
| F13 | Human versus Computer play will allow an intermediate mode |

## USER CHARACTERISTICS

Dots will be available to all users regardless of skill level. Requirements for use of the game are an understanding of the game and beginner computer skills.

## CONSTRAINTS

There have been no imposed constraints from our client. The application is not critical so its availability is not a concern.

## ASSUMPTIONS AND DEPENDENCIES

Assumptions are that if the user is playing a two player game, both users are seated at the same computer and alternate turns. Games are not saved and closing a browser will quit the current game.

## SYSTEM LEVEL (NON-FUNCTIONAL) REQUIREMENTS

### Site dependencies

Site dependencies are that the user has an Operating System installed, internet access, and a web browser with HTML 5 support. The user is expected to run a computer with a mouse. Keyboard input is not accepted.

### Safety, security and privacy requirements

Dots & Boxes game does not ask for or save the users’ personal information.

### Performance requirements

The maximum level of players in a game is two. The amount of information is minimal with the majority of input being user mouse clicks. The placement of a line should respond within less than 3 seconds. There is no limit to the number of times the game can be played.

### System and software quality

At a minimum, the game will be required to support two modes of play. The computer strategy play will depend on the difficulty selected by the user. The software design will utilize MVC architecture for ease of maintenance. The system architecture also has built in support for automated testing and HTML5 will be used for portability and flexibility. The game is intuitive for the user.

### Packaging and delivery requirements

The executable system and all associated documentation (i.e., SSRS, SDD, code listing, test plan (data and results), and user manual) will be delivered to the customer on CD’s and/or via email, as specified by the customer at time of delivery. Although document “drops” will occur throughout the system development process, the final, edited version of the above documents will accompany the final, accepted version of the executable system.

### Personnel-related requirements

The system under development has no special personnel-related characteristics.

### Training-related requirements

No training materials or expectations are tied to this project other than the limited help screens built into the software and the accompanying user manual.

### Logistics-related requirements

Dots & Boxes has no support or warranty and will not be distributed.

### Other requirements

All requirements have been covered in previous sections.

### Precedence and criticality of requirements

The order of precedence is building the two players version of the game first, followed by the player versus computer option. In the player versus computer mode, the “easy” computer strategy play will be built first, before the intermediate computer strategy mode.SPECIFIC REQUIREMENTS

## EXTERNAL INTERFACE REQUIREMENTS

### Hardware Interfaces

Players must use a computer with an operating system and a browser with Internet access.

### Software Interfaces

The browser must be HTML5 compatible.

### User Interfaces

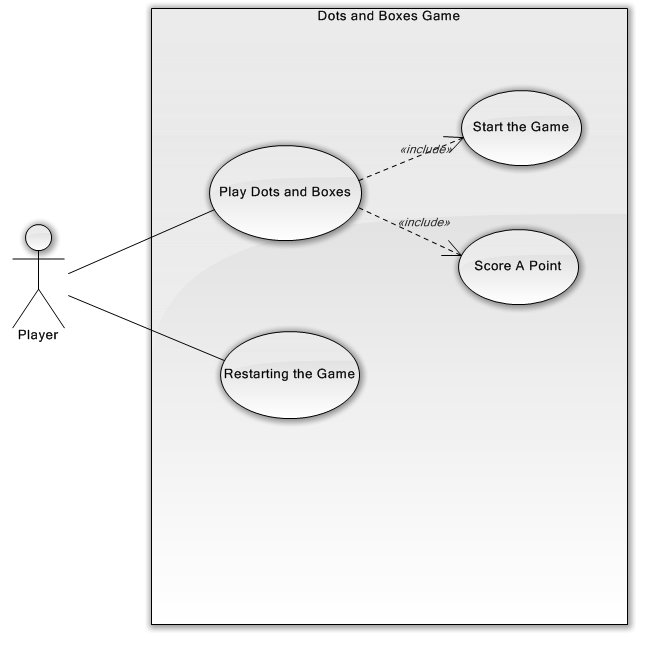
Gameplay requires the use of a mouse. Keyboard support is not offered.

### Other Communication Interfaces

None.

## SYSTEM FEATURES

### Use Case Diagrams



### Use Case Specifications

|  |  |  |
| --- | --- | --- |
| Header | | |
| Use Case ID | | UC1 |
| Use Case Version | | V.1.0 |
| Status | |  |
| Release | | Release ID |
| Author | | Team |
| Body | | |
| Title | Play Dots and Boxes | |
| Actors | Player | |
| Normal Flow | 1. Player starts the game. 2. (steps 2 and 3 repeat until all line locations have been selected) Players alternate selecting a location to place a horizontal or vertical line. 3. Game validates the location is available and draws the line indicating which player owns the line 4. Game determines and announces winner with the highest score. 5. Game prompts the players to play again. 6. Player chooses to quit the game. 7. Game closes Player’s browser. | |
| Alternate Flows | | |
|  | A1. Game fails to start:   1. Player makes note of error and waits to try again later.   A2 – A3. Player elects to restart the game:   1. Execute use case restarting the game.   A2 – A3. Player elects to quit the game:   1. Game closes Player’s browser.   A3a. Selected line location encloses a square:   1. Execute use case score a point. 2. Player scoring a point makes another move.   A3b. Game determines no more line locations available:   1. Game determines and announces winner with the highest score.   A6. Player elects to play again:   1. Execute use case restarting the game. | |

### System feature F2: Human versus Human play

#### Introduction/Purpose of this feature

Two players can play the game against each other.

#### Input/Output sequence for this feature

A Player will select to play human versus human mode of game play and then players will alternate taking turns placing a line on the grid. If a player places the fourth line forming a square, the square will be filled-in with the player’s color code, their score will be incremented by 1 point and the player will take another turn. This game play will continue until all lines have been taken. The player with the most squares will win the game.

#### Design constraints of this feature

In 2 player mode, both players must be playing from the same computer.

A mouse is required to place a line.

#### Performance requirements of this feature

None.

#### Detailed functional requirements of this feature

##### Functional requirement FR7

Player1 will start the game by selecting a line location to place a vertical or horizontal line.

##### Functional requirement FR8

The system will determine if the line location is available.

##### Functional requirement FR8.1

If the line location is already taken, the system will not allow the player to place a line at that location.

##### Functional requirement FR10

If the line segment does not represent the fourth side of a square, play will alternate to the other player’s move.

##### Functional requirement FR11

If a line would represent the fourth side of a square, a player is not required to select that line location for their move.

### System feature F3: Human versus Computer play

#### Introduction/Purpose of this feature

This mode of game play will allow a single player play the game against the computer.

#### Input/Output sequence for this feature

A Player will select to play human versus computer mode, then the level of difficulty of easy or intermediate. The player and the computer will alternate taking turns placing a line on the grid. If a player places the fourth line forming a square, the square will be filled-in with the player’s color code, their score will be incremented by 1 point and the player will take another turn. This game play will continue until all lines have been taken. The player with the most squares will win the game.

#### Design constraints of this feature

The human player is required to use a mouse to place a line.

#### Performance requirements of this feature

None.

#### Detailed functional requirements of this feature

##### Functional requirement FR12

The human player will always start with the first move.

### System feature F6: Line colored for each player

#### Introduction/Purpose of this feature

When a player places a line on the grid, it will be color coded to indicate which player owns the line.

#### Input/Output sequence for this feature

A Player will select an available location to place a line by clicking the mouse. The system will place a line at the location with it color coded for that player.

#### Design constraints of this feature

The human player is required to use a mouse to place a line.

#### Performance requirements of this feature

None.

#### Detailed functional requirements of this feature

##### Functional requirement FR9

The system will present a line in the color code of the player.

### System feature F7: Completed boxes filled-in with player color code

#### Introduction/Purpose of this feature

If the line segment represents the fourth side of a square, the system will fill in the square in the color code of the player.

#### Input/Output sequence for this feature

A Player will select an available location to place a line by clicking the mouse. The system will place a line at the location with it color coded for that player. If the line represents the fourth line of a square, the square will be filled-in with the player’s color code.

#### Design constraints of this feature

The human player is required to use a mouse to place a line.

#### Performance requirements of this feature

None.

#### Detailed functional requirements of this feature

##### Functional requirement FR17

If the line segment represents the fourth side of a square, the system will fill in the square in the color code of the player.

### System feature F8: Winner determined and final score displayed

#### Introduction/Purpose of this feature

When the last line location is selected by a player, the system will update the player’s score, and determine a winner.

#### Input/Output sequence for this feature

A Player will select an available location to place a line by clicking the mouse. The system will place a line at the location with it color coded for that player. If the line represents the last available line location, the system will update the final score and determine a winner.

#### Design constraints of this feature

None.

#### Performance requirements of this feature

Upon the last line location selected, the system will determine who the winner is within 3 seconds or less.

#### Detailed functional requirements of this feature

##### Functional requirement FR18

When all squares have been taken by the players, the game will end.

##### Functional requirement FR19

When all squares have been taken by the players, the system will present the winner and the final score.

### System feature F9: When game play is over, the player can choose to play again

#### Introduction/Purpose of this feature

When game play is over, a player can choose to play again, and the system will present the game mode selection screen.

#### Input/Output sequence for this feature

A player selects to play the game again, the system presents the game mode screen.

#### Design constraints of this feature

None.

#### Performance requirements of this feature

Upon electing to play again, the system will present the game mode selection screen within 3 seconds or less.

#### Detailed functional requirements of this feature

##### Functional requirement FR20

When a game is over, the system will allow the game to be replayed.

### System feature F10: Ability to quit the game at any time

#### Introduction/Purpose of this feature

At any time during game play, a player can choose to quite the game, which will close the browser application.

#### Input/Output sequence for this feature

A Player will select to quit the game and any-time. The system will close the browser application.

#### Design constraints of this feature

None.

#### Performance requirements of this feature

Once a player has elected to quit, the application will close the browser within 3 seconds or less.

#### Detailed functional requirements of this feature

##### Functional requirement FR21

At any time during game play, a Player may choose to quite the game.

### System feature F12: Human versus Computer play will allow an easy mode

#### Introduction/Purpose of this feature

When a player elects to play against the computer, they can choose an easy mode of play where the computer’s strategy is trivial.

#### Input/Output sequence for this feature

A Player selects easy mode of play against the computer, and the computer will use a trivial strategy for selecting it’s line location.

#### Design constraints of this feature

None.

#### Performance requirements of this feature

node

#### Detailed functional requirements of this feature

##### Functional requirement FR13

The human player can choose to play easy mode against the computer.

##### Functional requirement FR13.1

Computer easy mode will look to complete a square by placing the fourth line; otherwise it will randomly select an open line segment.

### System feature F13: Human versus Computer play will allow an intermediate mode

#### Introduction/Purpose of this feature

When a player elects to play against the computer, they can choose an intermediate mode of play where the computer’s strategy will employ offensive and defensive strategies.

#### Input/Output sequence for this feature

A Player selects intermediate mode of play against the computer, and the computer will use a offensive and defensive strategy for selecting it’s line location.

#### Design constraints of this feature

None.

#### Performance requirements of this feature

node

#### Detailed functional requirements of this feature

##### Functional requirement FR14

The human player can choose to play intermediate mode against the computer.

##### Functional requirement FR14.1

Computer intermediate mode will look to complete a square by placing the fourth line; otherwise it will review for the best defensive or offensive play.

|  |  |  |
| --- | --- | --- |
| Header | | |
| Use Case ID | | UC2 |
| Use Case Version | | V.1.0 |
| Status | |  |
| Release | | Release ID |
| Author | | Team |
| Body | | |
| Title | Start the Game | |
| Actors | Player | |
| Normal Flow | 1. Player goes to the game URL in a browser. 2. Game prompts player to select one of:  * 2 Player mode * Player versus Computer – novice * Player versus Computer - intermediate  1. Player selects the desired mode of game play. 2. Game displays the grid of dots and shows the score as zero for each player. 3. Game prompts for player 1 to make a move. | |
| Alternate Flows | | |
|  | A1. Error loading game:   1. Player notes the error, and waits to try again later.   A3. Player exits game by closing browser:   1. Game prompts player if they really want to quit. 2. Player chooses to quit the game. | |

### System feature F1: Start the Game

#### Introduction/Purpose of this feature

A player will initiate the start of the Dots and Boxes game to play.

#### Input/Output sequence for this feature

Player will call up the game URL in their browser, select the game mode, and the system will present the game ready for the 1st player to make a move.

#### Design constraints of this feature

In 2 player mode, both players must be playing from the same computer.

A mouse is required to place a line.

#### Performance requirements of this feature

Once a player has made the last game setup selection, the system will present the game ready for the first player to make a move in less than 3 seconds.

#### Detailed functional requirements of this feature

##### Functional requirement FR1

The system will use a 4 X 4 grid for game play.

##### Functional requirement FR2

Upon start-up of the game, the system will prompt for the mode of play.

###### Functional requirement FR2.1

The system will provide a mode of play for 2 human players.

###### Functional requirement FR2.2

The system will provide a mode of play for 1 human versus computer.

##### Functional requirement FR3

Upon start-up of the game, the system will display a grid of dots representing the grid size.

##### Functional requirement FR4

Upon start-up of the game, the system will display the score (0, 0).

##### Functional requirement FR5

Upon start-up of the game, the grid will not show any dots connected with lines.

### System feature F5: Color coding for each player

#### Introduction/Purpose of this feature

Each player will be assigned a color code indicating who owns a line and which player has won a square.

#### Input/Output sequence for this feature

When a player selects a line location, it will be color coded to that players’ color. When a line is placed as the fourth line of a square, the square will be filled-in with the color code for the player.

#### Design constraints of this feature

None

#### Performance requirements of this feature

Once a player has placed a line, the system will present the line in the player’s color code within 2 seconds or less.

If a player places the fourth line of a square, the square will be filled-in with the player’s color code within 2 seconds or less.

#### Detailed functional requirements of this feature

##### Functional requirement FR6

Upon start-up of the game, the system will present the color code for each player.

|  |  |  |
| --- | --- | --- |
| Header | | |
| Use Case ID | | UC3 |
| Use Case Version | | V.1.0 |
| Status | |  |
| Release | | Release ID |
| Author | | Team |
| Body | | |
| Title | Score a Point | |
| Actors | Player | |
| Normal Flow | 1. Game determines the line selection finishes a square. 2. Game marks the square for the current player. 3. Game adds a 1 point to the player’s score. | |
| Alternate Flows | | |
|  | A1. Alternate flow | |

### System feature F4: Real-time score during play

#### Introduction/Purpose of this feature

When a player wins a square, the system will maintain and update the score for the players.

#### Input/Output sequence for this feature

When a player places a line and it is placed as the fourth line of a square, the score for the players will be updated by adding 1 point to that player’s score.

#### Design constraints of this feature

None

#### Performance requirements of this feature

Once a player places the fourth line of a square, the scores for the players will be updated within 2 seconds or less.

#### Detailed functional requirements of this feature

##### Functional requirement FR15

If the line segment represents the fourth side of a square, the system will add one point to the players score.

##### Functional requirement FR16

If the line segment represents the fourth side of a square, the same player will be allowed to make another move by selecting another line location.

|  |  |  |
| --- | --- | --- |
| Header | | |
| Use Case ID | | UC4 |
| Use Case Version | | V.1.0 |
| Status | |  |
| Release | | Release ID |
| Author | | Team |
| Body | | |
| Title | Restarting the Game | |
| Actors | Player | |
| Normal Flow | 1. Game prompts player to confirm restart of game. 2. Player elects to restart the game. 3. Execute use case Start the Game. | |
| Alternate Flows | | |
|  | A2. Player elects not to restart the game:   1. Game leaves current lines and squares on grid 2. Game leaves the score as is. | |

### System feature F11: Ability to restart the game at any time

#### Introduction/Purpose of this feature

A player may choose to restart the game at any-time during game play.

#### Input/Output sequence for this feature

If a player elects to restart the game, the existing game will not be saved, and the application will take the player back to the starting the game screen where they can select which mode to play.

#### Design constraints of this feature

None

#### Performance requirements of this feature

Once a player elects to restart the game, the system will present the mode selection screen within 2 seconds or less.

#### Detailed functional requirements of this feature

##### Functional requirement FR21

At any time during game play, a Player may choose to restart the game.

### REQUIREMENTS TRACEABILITY

| **UC Name** | **Feature Name** | **Req No.** | **Requirement Description** | **Priority** | **SDD** | **Alpha Release** | | Beta Release | | **Final Test** | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case(s)** | **Test Res.** | **Test Case(s)** | **Test Res.** | **Test Case(s)** | **Test Res.** |
| UC1 – Play Dots and Boxes | F2 Human versus Human play | FR7 | Player1 will start the game by selecting a line location to place a vertical or horizontal line. |  |  |  |  |  |  |  |  |
| FR8 | The system will determine if the line location is available. |  |  |  |  |  |  |  |  |
|  | FR8.1 | If the line location is already taken, the system will not allow the player to place a line at that location. |  |  |  |  |  |  |  |  |
| FR10 | If the line segment does not represent the fourth side of a square, play will alternate to the other player’s move. |  |  |  |  |  |  |  |  |
|  | FR11 | If a line would represent the fourth side of a square, a player is not required to select that line location for their move. |  |  |  |  |  |  |  |  |
| F3 Human versus Computer play | FR12 | The human player will always start with the first move. |  |  |  |  |  |  |  |  |
|  | F6Line colored for each player | FR9 | The system will present a line in the color code of the player. |  |  |  |  |  |  |  |  |
|  | F7 Completed boxes filled-in with player color code | FR17 | If the line segment represents the fourth side of a square, the system will fill in the square in the color code of the player. |  |  |  |  |  |  |  |  |
| F8 Winner determined and final score displayed | FR18 | When all squares have been taken by the players, the game will end. |  |  |  |  |  |  |  |  |
| FR19 | When all squares have been taken by the players, the system will present the winner and the final score. |  |  |  |  |  |  |  |  |
| F9 When game play is over, the player can choose to play again | FR20 | When a game is over, the system will allow the game to be replayed. |  |  |  |  |  |  |  |  |
| F10 Ability to quite the game at any time | FR21 | At any time during game play, a Player may choose to quite the game. |  |  |  |  |  |  |  |  |
| F12 Human versus Computer play will allow an easy mode | FR13 | The human player can choose to play easy mode against the computer. |  |  |  |  |  |  |  |  |
| FR13.1 | Computer easy mode will look to complete a square by placing the fourth line; otherwise it will randomly select an open line segment. |  |  |  |  |  |  |  |  |
| F13 Human versus Computer play will allow an intermediate mode | FR14 | The human player can choose to play intermediate mode against the computer. |  |  |  |  |  |  |  |  |
| FR14.1 | Computer intermediate mode will look to complete a square by placing the fourth line; otherwise it will review for the best defensive or offensive play. |  |  |  |  |  |  |  |  |
| UC2 - Start the game | F1 Start the game | FR1 | The system will use a 4 X 4 grid for game play. |  |  |  |  |  |  |  |  |
| FR2 | Upon start-up of the game, the system will prompt for the mode of play. |  |  |  |  |  |  |  |  |
| FR2.1 | The system will provide a mode of play for 2 human players. |  |  |  |  |  |  |  |  |
| FR2.2 | The system will provide a mode of play for 1 human versus computer. |  |  |  |  |  |  |  |  |
| FR3 | Upon start-up of the game, the system will display a grid of dots representing the grid size. |  |  |  |  |  |  |  |  |
| FR4 | Upon start-up of the game, the system will display the score (0, 0). |  |  |  |  |  |  |  |  |
| FR5 | Upon start-up of the game, the grid will not show any dots connected with lines. |  |  |  |  |  |  |  |  |
| F5 Color coding for each player | FR6 | Upon start-up of the game, the system will present the color code for each player. |  |  |  |  |  |  |  |  |
| UC3 - Score a Point | F4 Real-time score during play | FR15 | If the line segment represents the fourth side of a square, the system will add one point to the players score. |  |  |  |  |  |  |  |  |
| FR16 | If the line segment represents the fourth side of a square, the same player will be allowed to make another move by selecting another line location. |  |  |  |  |  |  |  |  |
| UC4 -Restarting the Game | F11 Ability to restart the game at any time | FR22 | At any time during game play, a Player may choose to restart the game. |  |  |  |  |  |  |  |  |

Priorities are: **M**andatory, **L**ow, **H**igh

SDD link is version and page number or function name.

Test cases and results are file names and **P**ass/**F**ail or % passing