SE 3XA3: SRS
Ultimate Tic Tac Toe

Team 3
Kunal Shah - shahk24
Pareek Ravi - ravip2

November 16, 2016

Contents

List of Tables

List of Figures

Abstract

This document describes the requirements for Ultimate Tic Tac Toe. The template for the Software Requirements Specification (SRS) is a subset of the Volere template (Robertson and Robertson, 2012).

1 Project Drivers

1.1 The Purpose of the Project

The purpose of this project is to redevelop an existing open source project while following proper documentation and waterfall design methods.

1.2 The Stakeholders

The stakeholders include all parties that would have a vested interest in this projected. This includes, the client Dr Smith, the developers Pareek and Kunal ,all the people who would be interested in playing the game and the teaching assistant Christopher.

1.2.1 The Client

The client of Ultimate Tic Tac Toe is Dr. Smith as he is the project manager and he requested this project. We are creating this game for him to distribute to the customer.

1.2.2 The Customers

The target customer for this game would be anyone who has a device connected to the internet and wants to play a game with a friend. This game is targeted towards young people, but it is open for anyone to play.

1.2.3 Other Stakeholders

Another stakeholder is the teaching assistant, Christopher, as is aiding Dr. Smith in project management during lab and developers during the development process.

1.3 Mandated Constraints

There are 3 constraints of this game.

1. The internet
2. A device that can connect to the internet
3. A friend to play the game with

1.4 Naming Conventions and Terminology

The naming convention for winning an inner game will be called controlling a square. In order to win, a player must get a tic tac toe of controlled squares. If there is a draw in a square, it will be called a dumped square.

1.5 Relevant Facts and Assumptions

We assume that the users have a stable internet connection and are aware of the rules of regular Tic Tac Toe. It is important to note that the game will work better in large screens rather than a small display. We assume that the user is using a device that is either touch enables or has a mouse to click with.

2 Functional Requirements

2.1 The Scope of the Work and the Product

We investigated the original Ultimate tic tac toe game (kennycason, 2010). In this game the developer has poor user interface and no game instructions. It also only supports single device with no AI option. These shortcomings are the features we would like to improve upon.

2.1.1 The Context of the Work

The central task is to create a game of Ultimate Tic Tac Toe that can be played on the internet with a friend. This will require a server to host the game, each user to connect to the game, and ensure a connection with the server. In order for the game to be played, the server will determine which player is to play, get that player's move record it in the virtual board, and then notify the other player of the first player's move. This process of sending the move made back and forth will be the role of the server. The server will also indicate to a player if they have won, lost or if that game has ended in a draw.

2.1.2 Work Partitioning

The development of this game is shared evenly between the two team members. The game logic is developed by both members. The graphical portion is mainly going to be done by Kunal, and the server portion will be done by Pareek.

2.1.3 Individual Product Use Cases

One of a possible use case is when a player might disconnect from their internet connection, but their internet resumes and they wish to reconnect to the game. Another case is if a player wished to change the device they are playing on during an on going game. There could be a case where both players agree that a player should be able to revert back by one move they made, in which case the game will undo the user's last move. The general use case is of course two users playing on their devices without any flaws from start to end.

2.2 Functional Requirements

FR1: The game will display a game board on the user's device

FR2: The user's move will be registered, recorded and reflected on the game board
Fit Criteria: Game board should have the user's character at the cell that they clicked on if it is a valid move

FR3: The game will determine when a player has taken control of a square
Fit Criteria: The array for the inner board will have one of 8 possible sets of three in a row. Three columns, three rows and the two diagonals

FR4: The game will determine when the square has ended in a draw
Fit Criteria: The array for the inner board will not have any of the 8

possible sets of three in a row. Three columns, three rows and the two diagonals

FR5: The game will determine when the full game is won or drawn
Fit Criteria: The array for the inner board will have one of 8 possible sets of three in a row on the full board. Three columns, three rows and the two diagonals

FR6: The game will determine in which square the next player will make their move based on the previous player's move
Fit Criteria: Clicking on any cell on an inner board that is not allowed will not result in any move

3 Non-functional Requirements

3.1 Look and Feel Requirements

LF1: Game should be visually appealing
Fit Criteria: 90% of surveyed users should find the colours used appealing

LF2: The game should have a clean graphical interface
Fit Criteria: 90% of surveyed users should report that there no unnecessary graphics

3.2 Usability and Humanity Requirements

UH1: The game must be easy to learn
Fit Criteria: The average time for a user to fully learn the rules must be no more than 2 minutes after reading the rules

UH2: The game should allow users to interact with the game in multiple ways
Fit Criteria: The click interface must always work and on computers that have touch, that should also always work

UH3: There should a be a button for instructions on the rules of the game
Fit Criteria: 90% of users should state that they used or at least found the rules button

3.3 Performance Requirements

PR1: There should be no delay in reflecting a player's move on the board
Fit Criteria: The maximum time for a move to be reflected on the board is 750 milliseconds

PR2: The game should process if a player has won the game or completed an inner board quickly
Fit Criteria: The maximum time for the board to reflect if a board is complete is 1 second

PR3: The CPU/GPU should not be overtaxed by the game
Fit Criteria: The maximum CPU/GPU usage on any device should be 30%

3.4 Operational and Environmental Requirements

OE1: The browser should not be using too much of the CPU power
Fit Criteria: The maximum browser usage on the device should be 45%

OE2: The browser used needs HTML5
Fit Criteria: Browsers will need to pass the HTML5 test as provided by (NielsLeenher, 2010)

3.5 Maintainability and Support Requirements

MS1: Should the game fail to run, it should be repairable
Fit Criteria: The time to repair it should be no longer than 2 hours

3.6 Security Requirements

SE1: The game should not be collecting any user information or analytics
Fit Criteria: There shall be 0 cookies collected with this game

3.7 Cultural Requirements

CU1: The product shall not be offensive to religious or ethnic groups.
Fit Criteria: A survey should indicate that at least 90% of people did not find the game offensive

CU2: The game user interface should be language ambiguous.
Fit Criteria: All the members of a group of people speaking different languages must be able to play the game

3.8 Legal Requirements

LE1: The game will have a rating suitable for all ages
Fit Criteria: The ESRB rating should be E for Everyone.

3.9 Health and Safety Requirements

HS1: The game should not result in people having epileptic episodes
Fit Criteria: A survey of people who play the game should indicate that none of the people had an epileptic episode as a result of playing the game

4 Project Issues

4.1 Open Issues

One issue we are concerned about is the adaptability of the user interface based on the device. Some issues could arise from touch or mouse input. Screen size of device could cause scaling problems such as having part of the UI cut off. Lastly the user interface might not be the same based on where the game is accessed. This could be caused due to the fact that different browsers render elements differently.

4.2 Off-the-Shelf Solutions

We will be utilizing four off the shelf solutions to assist this project

1. Moore server for web hosting
2. Gitlab for version control and issue tracking
3. Karma unit testing tool for verification
4. JsDoc for documentation generation

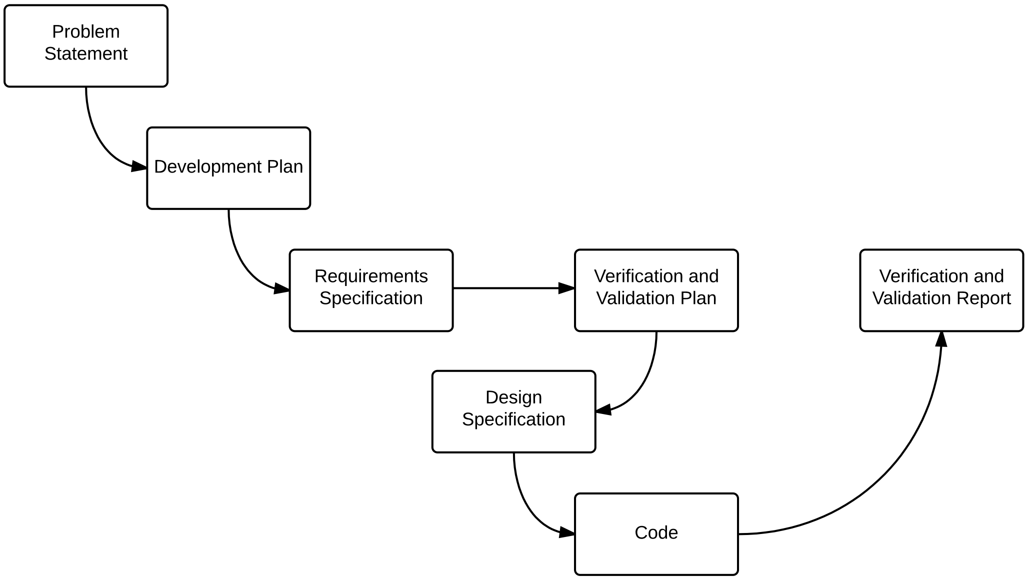
Problem Statement
Development Plan
Requirements Specification
Verification and Validation Plan
Verification and Validation Report
Design Specification
Code

Figure 1: Development Cycle

4.3 New Problems

Some new problems that have arisen during the development process are user problems such as miss click and poor internet connection. Additionally the planned server is not powerful enough to cope with our projected growth pattern.

4.4 Tasks

The development cycle will follow the modified waterfall life cycle as detailed by Dr. Smith. Refer to Figure 1 (Smith, 2016). In verification and validation, a set of test cases to ensure that all the game features are working. Developing a thorough set of use cases as well as user testing, the game will be tested. The design of this game will follow the standard MVC game structure. The model will be located on the server, the view and controller will be on each individual device. The code has been broken down into various milestones. Please refer to the Gantt Chart for further details. The final report will contain a detailed analysis of the result of the test cases and the reviews from the test users.

4.5 Migration to the New Product

The migration to this project will involve transferring the game logic to the server to allow for online game play if possible. The new product also will also change the graphics and the user interface. From a MVC standpoint, the view is the main change with the model shifting to the server. The controller will remain the same except making it friendly for mobile devices and all touch enabled devices.

4.6 Risks

There is a risk that the server that the game is hosted on could unexpectedly crash. There is also a security risk with a device being connected insecurely to the internet. We will ensure that only essential information is received and there are minimal permissions

4.7 Costs

We plan to complete this project with a zero dollar budget. All resources used to complete this project can be used free of charge. Additionally we plan to host this game on the McMaster Moore server which is also available to us for free.

4.8 User Documentation and Training

User documentation will be created in the form of game instructions. Comprehensive game rules have already already been created by mathwithbaddrawings.com (mathwithbaddrawings, 2016) . Our game instructions will be a version of theses rules with our own diagrams and explanations. There is no need for user training.

4.9 Waiting Room

After redeveloping the Ultimate Tic Tac Toe game, we plan to add Player vs Computer game mode. This AI should be able to intelligently perform moves based on the human player's previous move. Secondly User accounts which will be able to track player statistics such win ratio. Additionally we plan to add an online lobby to face strangers. Players will be matched based on their player statistics.

4.10 Ideas for Solutions

As discussed before we will be utilizing many off-the shelf solutions to help in the development process. We plan to Use CSS to style the game board. To make the game language ambiguous we will be including many images into the game instructions. To make this game easy to maintain we shall be using Model-view-controller (MVC) software architectural pattern to make it modular.

References

kennycason. Ultimate tic tac toe, July 2010. URL https://github.com/kennycason/ultimate_tictactoe/.

mathwithbaddrawings. Ultimate tic tac toe rules, October 2016. URL https://mathwithbaddrawings.com/2013/06/16/ultimate-tic-tac-toe/.

NielsLeenher. Html5 test, 2010. URL http://html5test.com/.

James Robertson and Suzanne Robertson. *Volere Requirements Specification Template*. Atlantic Systems Guild Limited, 16 edition, 2012.

Dr. Spencer Smith. Lecture slides, October 2016. URL https://gitlab.cas.mcmaster.ca/smiths/se3xa3/.

5 Appendix

5.1 Symbolic Parameters

MAX_PLAYERS: Maximum number of concurrent players.

Table 1: **Revision History**

Date	**Version**	**Notes**
October 3	1.0	initial setup
October 5	1.1	updated Project Drivers content
October 6	1.2	added Functional Requirements
October 6	1.3	added Non-functional Requirements
October 7	1.4	updated Functional & Non-functional Requirements
October 7	1.5	Made notes for Project Issues
October 11	1.5	finished for Project Issues
October 11	2.0	Format document