Capstone Project – Sprint 1

4x4 Tic-Tac-Toe

Administrator

2019

# Table of Contents

3. Application Description

4. User Stories

5. Entity Diagrams

6. Wireframe

7. Sprint 1 Reflection

8. User Documentation

9. Technical Documentation

10. Sprint 4 Reflection

Application Description

The function of this program will be to set up a game of 4x4 Tic-Tac-Toe. To begin I just want to set-up a single player game that will allow you to choose where you would like to place X’s and O’s in a turn based format, with a winner being declared after 4 X’s or O’s are placed in a row. I want to persist the game board as well as a running match total for both sides.

A WPF board will be created for the user to click where they want to place their letter, while a SQL database is used as the database. WVVM will likely be the design pattern.

I am planning on having a small table, via SQL to persist which buttons have been pressed and by whom. A box model with accompanying blank box, X box and O box will be implemented, as well as game logic.

User Stories

As a player, I want to press an empty box to insert either an X or an O so that the game has functionality.

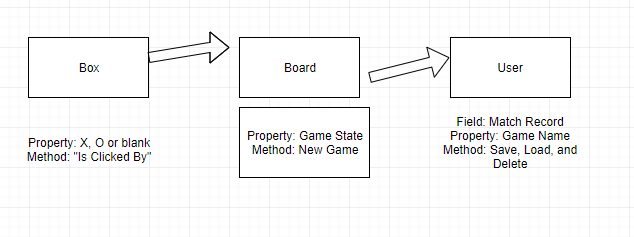
As a player, I want to see whose turn it is so that the game will be clearer, especially on loaded games.

As a player, I want to be able to save the game so that the game can be exited without losing my progress.

As a player, I want to see who won the game or if it was a tie so that the game feels playable.

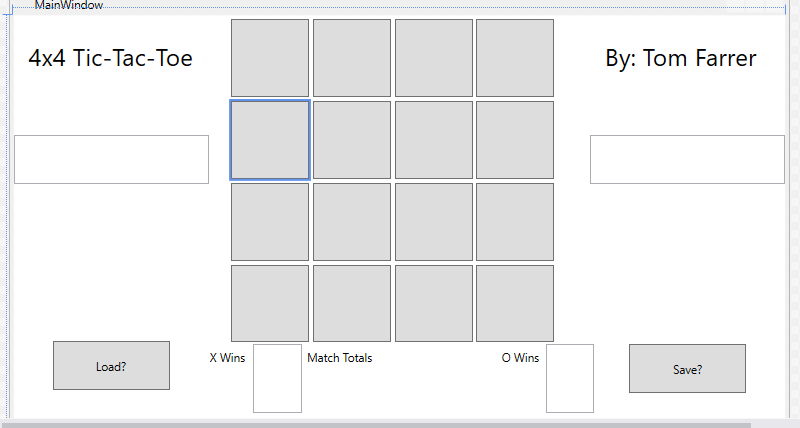
As a player, I want have a running total of match records so that a winner can be declared.

Entity Diagrams



Wireframe

The textboxes on the left and right will indicate whose turn it is or indicate when the end of the game has occurred, while a running tally of match records will be below the game board. Load will likely be where you start a new game or enter a previous game. Save should allow you to record your progress and the buttons on the board will determine where an X or O will be placed (depending on whose Turn it is).



Sprint 1 Reflections

The purpose of this sprint was largely to give myself a firm idea of what I wanted to accomplish. Setting up the game board as well as understanding what still needs to be done were the large accomplishments of this sprint.

Playing with stack panels can still be difficult for me to work with, but thanks to the calculator, I had a good template of how to resolve any issues.

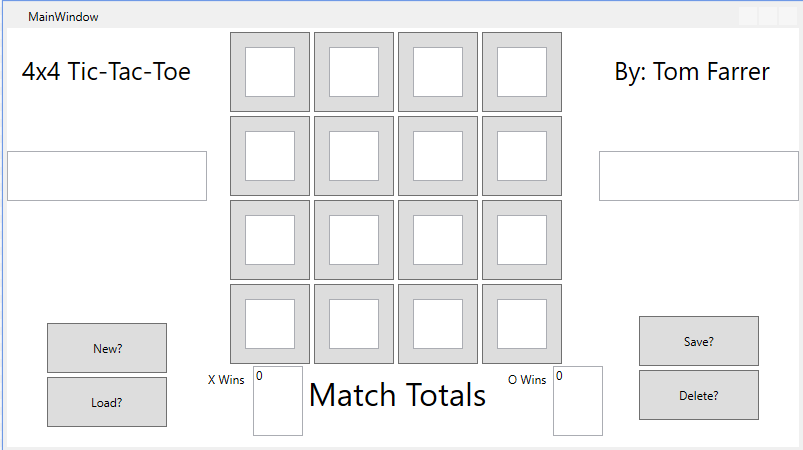
I am still not clear on how to get SQL to store my game progress, but I will leave myself open to trying out MongoDB, since we have a link on how to accomplish this.

I still struggle with conceptualizing what I want to build or how I want to create it, but I can take it one step at a time and not be afraid to do something now and change it later.

User Instructions

This is a classic game of Tic-Tac-Toe, click on a box in the main board (not the text boxes) and turn the box into an “X”. After you do this your computer opponent will respond with placing his own mark, an “O”. In order to win you must get four X’s in a row, column, or diagonally. After victory is achieved the victor is awarded a match win and another game is started, whoever lost gets to go first.

On the sides of the board you can choose any of the buttons at any time. The “New?” button will allow you to clear the board and records and start a new match. The “Save?” button will allow you to record the game and match details to be recalled with the “Load?” button. The “delete?” button clears your saved data. If you attempt to load data before you save anything a blank game board will replace the current game.



Technical Documentation

The application is a simple game of 4x4 Tic-Tac-Toe. Your opponent is the computer and the game will keep track of wins and losses. Thomas Farrer was the developer and the project was released on December 8, 2019. MVVM is the design pattern being used with WPF for the GUI. External persistence was not included in the application.

Sprint 4 Reflection

I fell way behind in the creation of this game, it was not until the end of the final sprint that I created a functional game.

I tried for a long time to get the contents of the buttons to be changed, instead I ended up with textboxes within the button to hold a mark. Initially the textboxes contained a string called “GetContent”, unfortunately since the textboxes all had the same name, when a button was clicked all of the textboxes changed to the same mark. To avoid this problem I gave each of the textboxes there own field and property.

I may have been to prideful to ask for help, but I have learned my lesson. The problem / solution seemed so elementary that I continued to struggle instead of gathering insight from my professor. I took so much time making the game functional that I missed out on the original purpose of the project, i.e. persistence.

Knowing to ask for help in a timely manner is the most significant takeaway from this assignment. My issues could have been solved a month ago if I would have faced the “Duh” moment (making individual fields for each box) with guidance. Getting the project to become functional was rather discouraging, but I always thought that I would learn better if I not only struggled, but came to the correct answer by myself.

Unfortunately my trying to use new technology took me away from fundamentals and I just kept trying to design the game how I think that it should be designed. I did a lot of research on how to code this game, but the most influential ideas came from YouTube videos. Examples of code seem to mean a lot more to me at this point, reading about the topic would work if I had a firmer grasp on what is being discussed. Watching an hour long video of a program being created from scratch helped me whereas snippets of code is much more difficult for me to get my head around, I like to code along to create a structure that I understand and can play with.

I am glad that I got so into making WPF work, even if it was not in the way that I wanted. I do want to create more WPF games, especially with external persistence, and with the greater idea of how to do this I am closer to being able to code my visions of what I want to create.