Tic-Tac-Toe

Kasper Jacobsen

Project Duration (28-05-2020) – ()

Table of Contents

[Project Requirements & specifications 1](#_Toc41590539)

[Must have(s) 1](#_Toc41590540)

[Nice to have(s) 2](#_Toc41590541)

[Rules of the game 2](#_Toc41590542)

[DataLayer 2](#_Toc41590543)

[Tic-Tac-Toe 2](#_Toc41590544)

[Time spent 2](#_Toc41590545)

[References and tools used 3](#_Toc41590546)

[Tools 3](#_Toc41590547)

[References 3](#_Toc41590548)

# Project Requirements & specifications

## Must have(s)

* Single and multiplayer option
  + Single player should be controlled from one pc.
    - Play with yourself.
  + Multiplayer should be played from two different pc’s
    - Players playing in multiplayer mode will have to wait for another player to join.
* Gameplay over the internet (possibly in a browser)
* **A player loses the game if** the following happens:
  + If a player leaves the website (reload or closes the browser) he/she will lose the game.
  + The player did not get three in a row.
  + The game ended in a tie.
* Point system:
  + **If the game ends in a tie,** +o pointswill be given to both players.
  + **If a player wins**. +1 pointwill be given.
  + **If a player Loses**. +o pointswill be given.
* You cannot do the following.
  + Change display name while in a game.
  + Change game mode while playing a game.

## Nice to have(s)

* Login feature to track your game score across games.

## Rules of the game

This game is a turn-based game. This mean that after ‘player 1’ have made a move, the turn will go to ‘Player 2’.

The first player that gets three in a row (of their own symbol) wins the game/round.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# ServerAPI

The ServerAPI is made to connect two players in a game of Tic-Tac-Toe

**Languages:** C#

**Framework:** .Net Core 3.1

**ORM** *(object-relational mapper)***:**  ASP .NET Core Web Application

**NuGet packages:**

|  |  |  |
| --- | --- | --- |
|  |  | Version |
|  |  | Version |

# Tic-Tac-Toe

‘Tic-Tac-Toe’ is a webpage build to run the game Tic-Tac-Toe

**Languages:** C#

**Framework:** .Net Core 3.1

**ORM** *(object-relational mapper)***:**  WPF

**NuGet packages:**

|  |  |  |
| --- | --- | --- |
|  |  | Version |
|  |  | Version |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Time spent

* **Planning the project** – approximately (9 hours and 30 minutes)
  + Planning,
    - Setting up my documentation structure
    - Planning how I will build the program and what to build it with.
  + Building diagrams
    - Mock-up image
    - Flowchart
    - Class Diagram

All diagrams were made using draw.io: <https://www.draw.io/>

* Setting up the Project environments - approximately (33 minutes)
  + Creating the project types that was planed in the previous step.
  + Setting up a GitHub Repository so the project uses version control.
* Setting up a basic idea for the UI layout– approximately (1 hour)
* Working on the ServerAPI and connectivity – approximately ()

# References and tools used

## Tools

* Draw.io – Used for making all my diagrams.   
  <https://www.draw.io/>
* Visual Studio 2019 – Used for making the whole project   
  <https://visualstudio.microsoft.com/vs/>
* Google Translate – Used for correcting most of my spelling errors  
  <https://translate.google.dk/?hl=da&tab=TT>

## References

* Inspirational YouTube Videos:
  + <https://www.youtube.com/watch?v=_OUs2kuI_Yo>
  + <https://www.youtube.com/watch?v=yq0dSkA1vpM>
* Inspirational Websites:
  + <https://developer.okta.com/blog/2019/11/21/csharp-websockets-tutorial>
* Inspirational Code:

# Installation Guide

## How to install