



# Project #1: The Game

## Overview

Let's start out with something fun - **a game!**

**We'll be making Tic Tac Toe**, a game that takes seconds to learn but minutes to master! Everyone will get a chance to **be creative**, and work through some really **tough programming challenges** to get your feet wet in the world of web development.

Those who dare will have the tools to make it **multi-player, and multi-device** – so you can say to someone, "hand me your phone," load up the game, and play a quick round!

**You will be working individually for this project**, but we'll be guiding you along the process and helping as you go. Show us what you've got!

---

## What You've Learned

By the time you submit this project, you will have covered new ground in many of the big themes of the course:

- **Command Line:** Practice interacting with the computer and navigating the filesystem from the command line.
  - **Source Control:** Manage and interact with a git repository to store changes to code.
  - **Programming Fundamentals:** Work with objects, constructors, events, while learning how to strategically solve problems and resolve errors.
  - **Web Fundamentals:** Learn how communication happens over the internet, and how to structure, style, and animate documents within a browser. Also learn how to respond to actions your users take and the data they input into the browser.
  - **Products and Teams:** Document your code and your code repository so others understand what you've built.
- 

## Big Goals

- **Build a web application from scratch**, without a starter codebase
- Use your programming skills to **map out the game logic for a simple game like Tic Tac Toe**
- **Separate HTML, CSS, and JavaScript files** in your application
- Build an application **to a spec that someone else gives you**

- **Build a dynamic game that allows two players to compete** (bonus: compete from separate devices)
  - **Craft a `readme.md` file that explains your app** to the world
- 

## Technical Requirements

Your app must:

- **Render a game board in the browser**
  - **Switch turns** between X and O (or whichever markers you select)
  - **Visually display which side won** if a player gets three in a row or show a draw if neither wins
  - **Include separate HTML / CSS / JavaScript files**
  - Stick with **KISS (Keep It Simple Stupid)** and **DRY (Don't Repeat Yourself)** principles
  - Use **Javascript** for **DOM manipulation**
  - **Deploy your game online**, where the rest of the world can access it
  - Use **semantic markup** for HTML and CSS (adhere to best practices)
- 

## Bonus

These are for extra credit! Don't focus on these until you've hit the core requirements.

- If allowing players to compete from separate devices, **display a "Waiting..." message** while users are waiting to be matched
  - Keep track of **multiple game rounds** with a win counter
  - Allow players to **customize their tokens** (X, O, name, picture, etc)
  - **Get inventive with your styling**, e.g. use hover effects or animations to spiff things up
  - **Use LocalStorage or SessionStorage** to persist data locally to allow games to continue after page refresh or loss of internet connectivity
- 

## Necessary Deliverables

- A **working game, built by you**, hosted somewhere on the internet
  - A **link to your hosted working game** in the URL section of your Github repo
  - A **git repository hosted on Github**, with a link to your hosted game, and frequent commits dating back to the very beginning of the project
  - A **`readme.md` file** with explanations of the technologies used, the approach taken, installation instructions, unsolved problems, etc.
- 

## Suggested Ways to Get Started

- **Break the project down into different components** (data, presentation, views, style, DOM manipulation) and brainstorm each component individually. Use whiteboards!
  - **Use your Development Tools** (console.log, inspector, alert statements, etc) to debug and solve problems
  - Work through the lessons in class, **ask questions and come to office hours** when you need to. Think about adding relevant code to your Tic Tac Toe game each night, instead of, you know... *procrastinating*.
  - **Commit early, commit often.** Don't be afraid to break something because you can always go back in time to a previous version.
  - **Check out Tutorial and Documentation resources**
  - **Don't be afraid to write code that you know you will have to remove later.** Create temporary elements (buttons, links, etc) that trigger events if real data is not available. For example, if you're trying to figure out how to change some text when the game is over but you haven't solved the win/lose game logic, you can create a button to simulate that until then.
- 

## Useful Resources

- [MDN Javascript Docs](#) (a great reference for all things Vanilla Javascript)
- [jQuery Docs](#) (if you're using jQuery)
- [Github Pages](#) (for hosting your game)