Project #1: The Game

Overview

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

Everyone will get a chance to **be creative**, and work through some really **tough programming challenges** – since you've already gotten your feet wet with Tic Tac Toe, it's up to you to come up with a fun and interesting game to build.

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!

Technical Requirements

Your app must:

- **Render a game in the browser**
- **Switch turns** between two players
- **Design logic for winning** & **visually display which player won**
- **Include separate HTML / CSS / JavaScript files**
- Stick with **KISS (Keep It Simple Stupid)** and **DRY (Don't Repeat Yourself)** principles
- Use **Javascript or jQuery** 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)

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 when you need to! Think about adding relevant code to your 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.
- **Consult documentation resources** (MDN, jQuery, etc.) at home to better understand what you’ll be getting into.
- **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.

Potential Project Ideas

Blackjack

Make a one player game where people down on their luck can lose all their money by guessing which card the computer will deal next!

Concentration

Sometimes just called "Memory", it's a card game in which all of the cards are laid face down on a surface and two cards are flipped face up over each turn. If you get all the matching cards, you've won!

Self-scoring Trivia

Test your wits & knowledge with whatever-the-heck you know about (so you can actually win). Guess answers, have the computer tell you how right you are!

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)*

Project Feedback + Evaluation

- **Project Workflow**: Did you complete the user stories, wireframes, task tracking, and/or ERDs, as specified above? Did you use source control as expected for the phase of the program you're in (detailed above)?
- **Technical Requirements**: Did you deliver a project that met all the technical requirements? Given what the class has covered so far, did you build something that was reasonably complex?
- **Creativity**: Did you added a personal spin or creative element into your project submission? Did you deliver something of value to the end user (not just a login button and an index page)?
- **Code Quality**: Did you follow code style guidance and best practices covered in class, such as spacing, modularity, and semantic naming? Did you comment your code as your instructors have in class?
- **Deployment**: Did you deploy your application to a public url using GitHub Pages?