Tic Tac Toe

Extra: \*\*multi-player, and multi-device\*\*

---

What You've Learned

Remember: - objects, constructors, events.

??Browser Applications: Introduce AJAX, used by the Firebase library.

??Server Applications\*\*: To come

??Deployment\*\*: Host a static web site in a managed hosting environment **(Firebase).**

**Big Goals**

1. Build a web application from scratch
2. Separate HTML, CSS, and JavaScript files in your application
3. Build a dynamic game that allows two players to compete\*\* (bonus: compete from separate devices)
4. Craft a ``readme.md`` file that explains your app\*\* to the world

**Technical Requirements**

Your app must:

1. Render a game board in the browser
2. Switch turns between X and O (or whichever markers you select)
3. Visually display which side won. if a player gets three in a row or show a draw/"cat’s game" if neither wins
4. Include separate HTML / CSS / JavaScript files
5. Use Javascript for DOM manipulation
6. Deploy your game online, where the rest of the world can access it
7. Use semantic markup for HTML and CSS (adhere to best practices)
8. Use a prebuilt data storage system (simpleStorage, FireBase) to simplify data storage and retrieval

---

### 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

\* Add \*\*tableside chat\*\* to your game

\* \*\*Use simpleStorage or Firebase\*\* 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 on Firebase

\* 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, unsolved problems, etc.

\* Wireframes and user stories, with links present in the readme.

---

### Suggested Timeline

Week | Day | Deliverable

------ | -----------------| ------------

Week 2 | Monday Morning | Requirements Explanation / Project Setup

Week 3 | Monday Morning | Basic view set up with game board interactions

Week 3 | Thursday Morning | Storage system integrated

Week 3 | Friday | Project complete! Demo Day!

---

### 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\*\* (Firebase tutorial, jQuery tutorial) 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.

---

### Useful Resources

\* \*\*[MDN Javascript Docs](https://developer.mozilla.org/en-US/docs/Web/JavaScript)\*\* \_(a great reference for all things Vanilla Javascript)\_

\* \*\*[jQuery Docs](http://api.jquery.com)\*\* \_(jQuery API reference)\_

\* \*\*[Firebase Web Platform](https://www.firebase.com/docs/docs/web/)\*\* \_(for realtime data syncing)\_

\* \*\*[Firebase hosting](https://www.firebase.com/docs/hosting/)\*\* \_(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 stretch yourself and 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?

\* \_\_Problem Solving\_\_: Are you able to defend why you implemented your solution in a certain way? Can you demonstrate that you thought through alternative implementations? \_(Note that this part of your feedback evaluation will take place during your one-on-one code review with your instructors, after you've completed the project.)\_

\* \_\_Total\_\_: Your instructors will give you a total score on your project between:

Score | Expectations

----- | ------------

\*\*0\*\* | \_Does not meet expectations.\_

\*\*1\*\* | \_Meets expectactions; good job!\_

\*\*2\*\* | \_Exceeds expectations, you wonderful creature, you!!\_

This 'top-level' score will serve as a helpful overall gauge of whether you met the project goals, but at the end of the day, \_\_the more important scores are the category scores\_\_, since they can help you identify where to focus your efforts for the next project!