**Requirements**

* **working browser game**
* hosted on GitHub Pages.
* using version control:
  + Sharing your work through a **git repository hosted on Github**.
  + **frequent, cohesive commit** with **good commit messages**.
* Produce **documentation** in the form of a **README**, which must:
  + **Link to your hosted game** in the **URL section** of your Github repo.
  + **List technologies used**.
  + **Document your planning** and tell a story about your **development process** and problem-solving strategy.
  + **List unsolved problems** which would be fixed in future iterations.
  + Link to **wireframes and user stories**.

Your app **must**:

* Be a **single-page** application.
* **Render a game board in the browser**.
* **Switch turns** between X and O (or whichever markers you select).
* **Visually display which side won** or show a draw/"cat’s game" if neither wins.
* Support playing **multiple games**, one at a time.
* Use **jQuery** for **DOM manipulation** and **event handling**.
* Use **AJAX** for interacting with a provided API. Specifically, your app must:
  + **Visually display** the results of retrieving game statistics, such as total games won by a user. (READ)
  + **Create** new games on the server. (CREATE)
  + **Update** a game by storing new moves. (UPDATE)
* Have **login**, **logout**, and **change password** functionality.

Your app **must not**:

* Rely on **refreshing the page** for **any** functionality.
* Display **non-functional** buttons, nor buttons that **do not successfully complete** a task.
* Have **any** user-facing **bugs**.
* Be playable **after finishing a game**.
* Allow players to move in the same square **more than once**.
* Change players when an **invalid move** is made.

**Nice-to-Haves**

* Practice separation of concerns by:
  + Using the [js-template](https://github.com/ga-wdi-boston/js-template) to store HTML, CSS, and JavaScript in the appropriate places.
  + Storing DOM manipulation code and network code in separate files.
* **KISS (Keep It Stupidly Simple)**.
* **DRY (Don't Repeat Yourself)**.

Finally, you **should not**:

* **Use alerts** for anything.
* **Display errors** or warnings in the console.
* **Display debugging** messages in the console.

**Bonus**

Once (and only once) you've satisfied the core requirements, here are some additional goals that you can shoot for:

* 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 localStorage** to persist data locally, allowing games to continue after page refresh or loss of internet connectivity.

**Getting Started**

Here's a rough sketch of what you should do and in what order:

1. Sketch some rough wireframes for how the front end will look and act.
2. Write out some user stories for the app. User stories are short of how a user interacts with your app, and follows the format "As a <role>, I want to <do something>, so that <some goal>.
3. Model the entities in your app. Draw a diagram. Use your wireframes and user stories to drive you modeling process by asking "What 'things' are a user interacting with?"
4. Create a repo that your project will use, and add a README to it.
5. Create a simple front-end with HTML and CSS, and host the front end on GitHub Pages. Use your wireframes to guide your layout.
6. Create the code to manage your game logic.
7. Write jQuery code to handle browser interaction.
8. Start communicating with the back-end using curl. Use this to begin writing your AJAX code.
9. Add any additional features to your app.
10. Finish your documentation. Make it high-quality.