Team 21 - The Canaries Team Document - June 6, 2023

Suggested plan for project completion:

Hard Deadline: Sunday, 11:59 PM.

Artificial Deadline: Friday(?) 11:59 PM.

Why? Since we want to have a buffer - if we underestimated some of the work, we'll have two extra days to take care of it.

Tasks to do:

- **Finish the 4 mini-apps**
    - Cartomancy
    - Fortune Stick
    - Molybdomancy
    - Yin Yang Coin
    - We have 27 open issues - it'd be best if everyone would self-assign what they're currently working on and mark it also using the Project (i.e. GitHub's kanban board)
    - **Definition of finished:**
        - It works, and…
            - Finished design elements (you can change a bit here and there afterwards like colors / fonts / backgrounds / sound effects / music, but no large UI changes unless absolutely necessary)
            - Code is documented
                - JavaScript
                    - Use JSDocs where appropriate (variables, functions, event handlers, classes)
                    - Use normal comments to explain processes in the code (e.g. function body)
                - HTML
                    - Use HTML comments to explain what each component is (explain the DOM structure / purpose of components)
                - CSS
                    - Use CSS comments both to define sections (have all the general CSS modifiers at the top and mark them as "General", then have sections for "General UI Buttons" (the home / audio / info buttons), and other subdivisions as applicable) - make it as easy to read as possible
                - JSON?

          - You can’t put comments in JSON files, so you’re safe here :)
      - Code passes linting
        - JavaScript
          - **Suggestion:** Download the StandardJS - JavaScript Standard Style extension if you’re using VSCode. However, it is finicky - I had to mess about with the settings for a while in order to get it running. If you want help with this, I’ll do my best to get you set-up.
          - When you push a commit in your branch, you can go to the repository and see the workflow that was run (CI/CD Pull Request / General Push) and check the linter jobs. If you didn’t pass, click on them to see the linter’s outputs which tell you exactly what lines you’re failing.
        - HTML & CSS
          - After pushing to your branch, check the corresponding GitHub Actions workflow and see the HTML & CSS’ linter’s output. At present, that’s the only way to pass the HTML & CSS linter.
      - Code passes unit tests
        - We have Jest unit testing as a part of all the CI/CD workflows. Make a folder in the tests/ directory for your mini-app and put all your tests there.
        - Check the pull request with the FortuneEngine tests for reference
        - **Note:** You cannot test fetch() in Jest, so you can’t write a test that checks whether a JSON file is read correctly, for instance. That’s end-to-end test territory, but unfortunately we don’t have a framework for that setup.
      - Your mini-app directory is clean
        - If you’re using assets, either put them in their own folder or put them in the shared assets folder
          - **Note:** the assets folder will have to be moved inside of src/ once all the mini-apps are merged into the dev branch, so make sure you leave comments where you’re using an asset from the assets folder!
- **Once a mini-app is done:**
  - Make a pull request from the branch that contains your finished mini-app to dev (NOT main)
  - Verify that you pass the CI/CD for the pull request
  - Resolve any merge conflicts before merging the pull request into dev
- **Once all mini-apps have been merged into dev:**

  - Make a new branch (e.g., move-assets-folder) that’s an off-shoot of dev to move the assets folder into the src folder, and fix all broken paths (which you should be able to easily find with CTRL + F due to all the comments you put)
  - Make a pull request to merge this new branch into dev
- **Once the assets folder has been moved into the src folder:**
  - Make a pull request to merge dev with main
  - Resolve any conflicts / make sure everything works in the CI / CD
- **Once dev has been merged into main:**
  - Make a new release! (version 1.0.0) with a change log / release notes
  - Update dev from main
  - Close all old offshoot branches from dev (e.g. all the mini-app branches)
- **Once v1.0.0 has been released:**
  - Make a branch for any of the following tasks from dev:
    - Some little things for the main page:
      - Add our logo / product name
      - Fill info text for the main page with info about our app / credits / info about our team
      - **Potentially:** Images over each continent in the map / animation when you hover over the map? (If not too hard)
    - Fix bugs
      - Have left-over CSS bugs? Try to fix it now as much as possible
    - Code clean-up
      - Code is hard to read? Are there any touch-ups you can apply?
    - Documentation
      - We want to have all documentation necessary for someone else (e.g., the TAs) to continue working on our project after the quarter ends
      - Project Repository
        - Code documentation
        - CI / CD documentation
      - Team Repository
        - CI / CD documentation
        - ADRs
          - For the main page
          - For each of the mini-apps
        - Any and all design documents you’ve used / made during the project → put these in the documentation folder in the team repository (under a subdirectory if appropriate - e.g., have a folder for your mini-app)