

Jeopardy - Assessment 2 - SpringBoard

by: Etienne Deneault

GitHub Repository: <https://github.com/edeneault/jeopardy-assessment-2>

Files

- index.html - jeopardy.js - markUp.js - jeopardy.css
- assets folder

Features

- Main game board with jservice API providing data for clues - answer/question.
- On Hover, the question boxes grow, allowing for a better reading experience.
- Materialize library used for css.
- Loading progress bar.
- LoDash for sampling function and higher order function map to retrieve category id's and build clue object.
- Custom SVG built in Adobe Illustrator CS6.

Code Details

- index.html file is essentially empty of any mark-up, it is all generated with javascript.
- jeopardy.js file handles initialization of global variables, event handling and game logic.
- markUp.js file handles markUp and other utility functions.
- API used is jservice (jservice.io).
- youtube iframe included at in the intro.
- Very little jQuery utilized, almost all vanilla javascript.

Todos and Improvements

- Implementation of progress bar is a bit "hacky". I used a progress bar from the materialize library, I was not sure how to implement and did it a bit poorly. For example, the dynamic html makes some divs move when the progress bar appears. To be rebuilt more elegantly.
- Potential add-ons would be question values and score. Two player with buzzer version. The thought of implementing these features did occur, but to manage the build-time, I decided to forgo them for this assessment.
- It is not fully responsive. It is fine in landscape mode on mobile devices but not in portrait mode. I explored a way to lock the view to landscape mode but did not successfully implement. For the sake

of time, I decided to forgo for this project.

- Add jasmine testing of functions.

Process && Learnings

- I set the following goals at the beginning of the project:
 - Game design reflection: functions needed and game logic.
 - Assignment step-by-step to build basic app functionality and mark-up.
 - Use Materialize CSS library (challenge quickly learn new library).
 - Generate all content dynamically.
 - Refactor the code into to components: Game Logic and markup/utils.
 - Documentation.
- I decided to use mostly vanilla js and not jQuery. In some instances, I did use jQuery when I struggled to preform some functions in vanilla js.
- I also avoided string literals for the mark-up, but got a bit tired of typing so much code and ended-up using a few string literals.

Technologies && Third Party Libraries && Fonts

- Materialize CSS Library.
- Google Font APi: <https://fonts.googleapis.com/css?family=Abel>
- jQuery Library and LoDash Library.
- Adobe Illustrator CS6
- HTML, CSS and JavaScript

Sources && References

- Code Base provided by SpringBoard - Jeopardy Assessment 2
- Documentation Used: MDN, StackOverflow, jQuery, LoDash, Materialize
- Media: pictures, videos, and sound files are expressively allowed to be used for a nonmonetary purpose.