Final Project: Trivia trouble

application proposal

# overview

## description

Trivia Trouble will be a single page app web site that uses an API to pull form a selection of multiple choice and true/false trivia questions. The game will have two screens, the player join/start screen and the gameplay screen. 2-6 players can play. On the join/start screen, the players choose a target number of points to play to and add players to the game. On a player’s turn, they select a difficulty and topic. Once selected, a question appears with possible answers. Points will be awarded based on difficulty and question type. **The problem** I am trying to solve is people having fun. It’s a trivia game. **I’ve chosen this project because** I like games, have programmed some simple ones in the past and it sounds fun. The **Audience** is people who want to play games online.

[Trello Invite: https://trello.com/invite/b/oLHaXQTa/ATTId6f7e1fc40ae04800b2d039c37bcc562AB27C2D4/trivia-trouble-10](Trello%20Invite:%20https://trello.com/invite/b/oLHaXQTa/ATTId6f7e1fc40ae04800b2d039c37bcc562AB27C2D4/trivia-trouble-10)

[Trello URL: https://trello.com/b/oLHaXQTa/trivia-trouble-10](https://trello.com/b/oLHaXQTa/trivia-trouble-10)

## stretch

* Hints: I would like to have a mechanic where you can get hints. This would work on multiple choice questions by removing one wrong answer. Doing so would cost some accumulated points. I don’t know how to implement a hint system on true/false questions. I might just skip those entirely.
* Images: I want to incorporate dynamic images somehow. I would like to base them on the questions and pull them from an image API.

# schedule

|  |  |  |  |
| --- | --- | --- | --- |
| **TASK** | **DUE** | | **DESCRIPTION** |
| Finalize wireframe | 2/7 | Get the wireframes as close to their final form as possible | |
| Player join page | 2/9 | Create an HTML/CSS player join page | |
| Report on Tasks | 2/10 | Complete the canvas assignment | |
| 60-Second update | 2/10 | Complete the canvas assignment | |
| Player.mjs | 2/12 | Create a module to contain player information such as name, color, image and score. | |
| Gameplay page | 2/14 | Create an HTML/CSS page that has the game board on it | |
| Game.mjs | 2/16 | Create the module to manage the gameplay flow. | |
| 60-Second update | 2/17 | Complete the canvas assignment | |
| Finish portfolio document | 2/20 | Complete the canvas assignment | |

# details

## major functions/Methods

Game.nextPlayer() { Get the next player and run corresponding animations. }

Game.renderQuestion() { Render the question onto the page }

Game.end() { Display the winner in a modal and navigate to the player select page or modal. Reset states }

Data.loadQuestion(type, category) { Query the API and render the question. }

Game.giveHint() { Randomly eliminate a wrong answer and subtract points from the player )

Player.updateScores(int) { Update the players score data in local storage }

Player.delete() { Delete the players data }

Player.add(name, color, iconIndex ) { Add a player to local storage }

## module list

* Game: Keeps track of game state. Game and board might not need to be separate.
* Data: Interface with the API
* Player: Hold the name and the current score. Can hook to local storage for scoring history and high score.

## data sources/api

### OPEN TRIVA DATABASE

Web URL: <https://opentdb.com/>

Web API help: <https://opentdb.com/api_config.php>

API URL: <https://opentdb.com/api.php>

Useful End Points:

* **/api.php**: The primary endpoint to get questions
  + { response\_code, results [{ type, difficulty, category, question, correct\_answer, incorrect\_answers[], } ]
  + amount: The number of questions to return
  + category: There are many categories
  + difficulty: easy, medium, hard
  + type: multiple, Boolean
  + token: ensures repeat questions aren’t asked
* /**api\_token.php**: Manages tokens
  + command=request: get a token
  + command=reset&token=TOKEN: Reset the token
* /**api\_category.php**: Returns all the categories
  + { id, name}

## styling

The page will have mostly dark backgrounds with brighter foreground colors and generous use of transparency. Players can choose a bright color or image as their avatar. The primary and secondary color will be incorporated, hopeful with gradients.

### colors

|  |  |
| --- | --- |
| **PRIMARY** | **SECONDARY** |
| **POP** | **WHITE SPACE** |

### typography

**Primary Font**: 

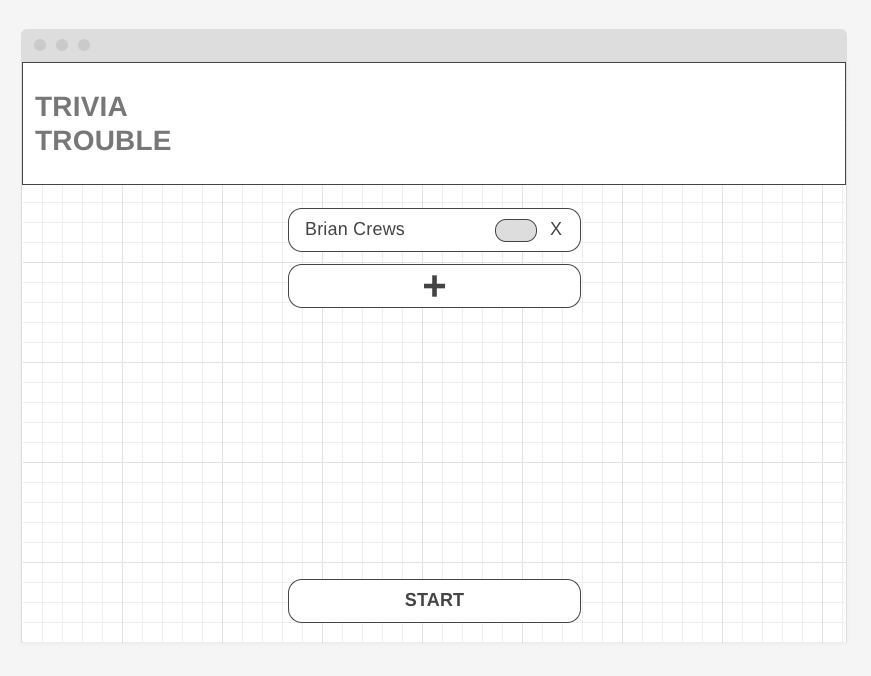
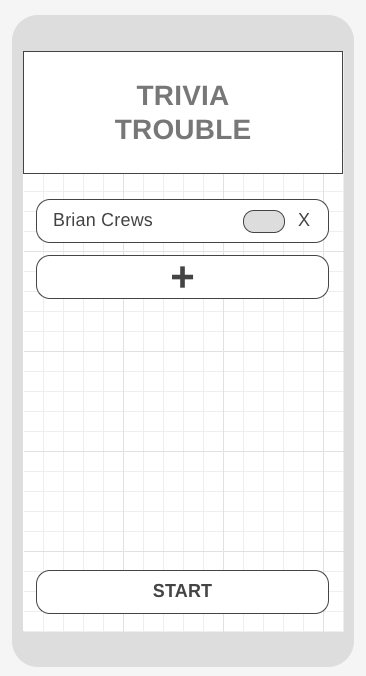
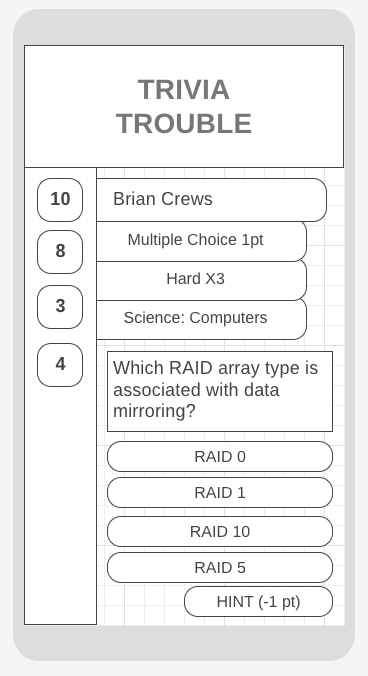
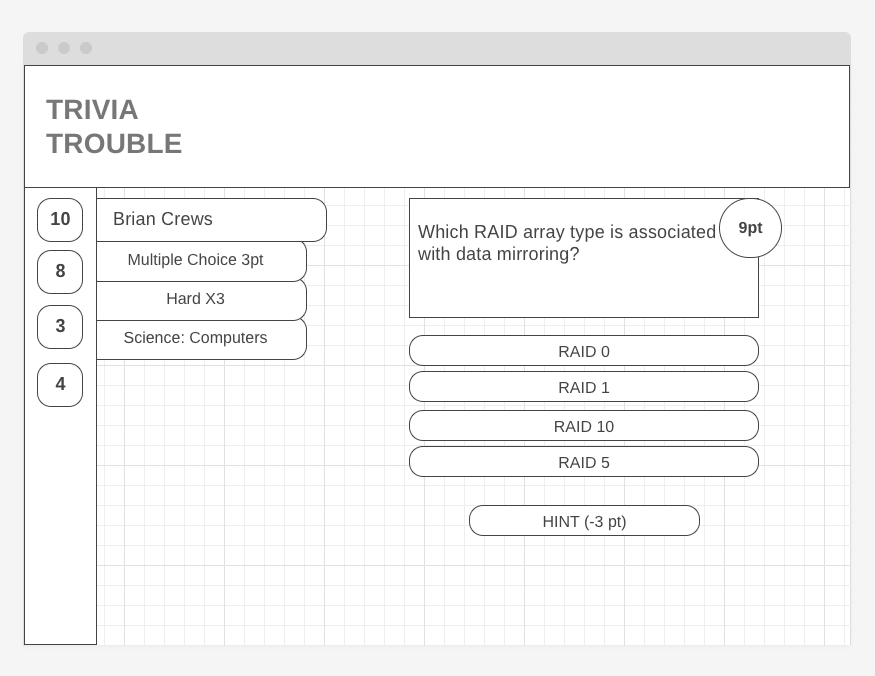
**Secondary Font:** 

### other

Other player icons will be taken from svg repo. They will be simple mono colored shapes.

### Wireframes

## mockup

