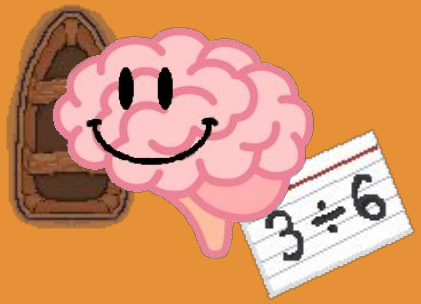
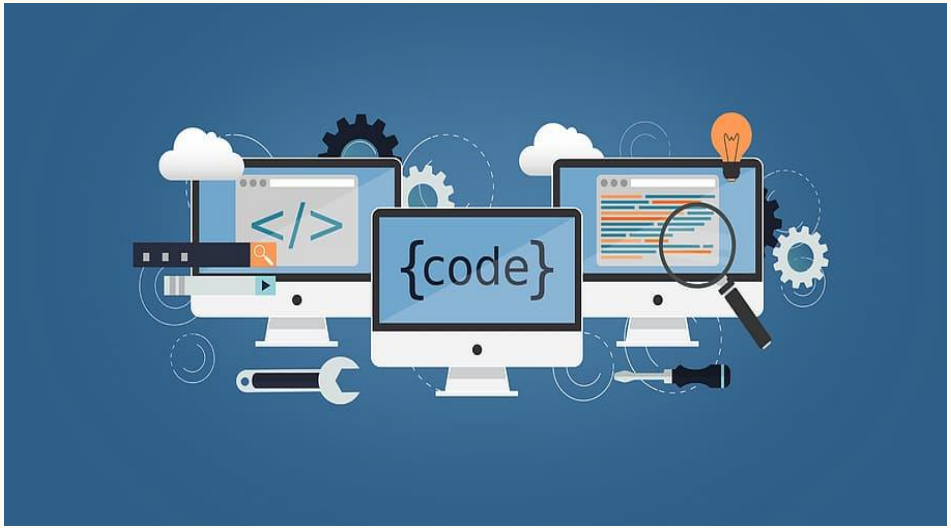


Epic Math Games

Riley O'Byrne, Chelsea Stockberger, Duke Manchester, Kylie Elbert, Davis Cohen & Jules Geneser

Tools Used

- Docker
- Node.js
- EJS (HTML)
- CSS
- JavaScript
- Github
- Jira
- PostgreSQL
- Heroku



WELCOME

Log In

Username Riley

Password ....

Login

New? Click to Sign Up

Games

Boat Game

Avoid obstacles and answer math problems at a fast pace!

Flashcards

Beat your high score in fun intense math flashcards!

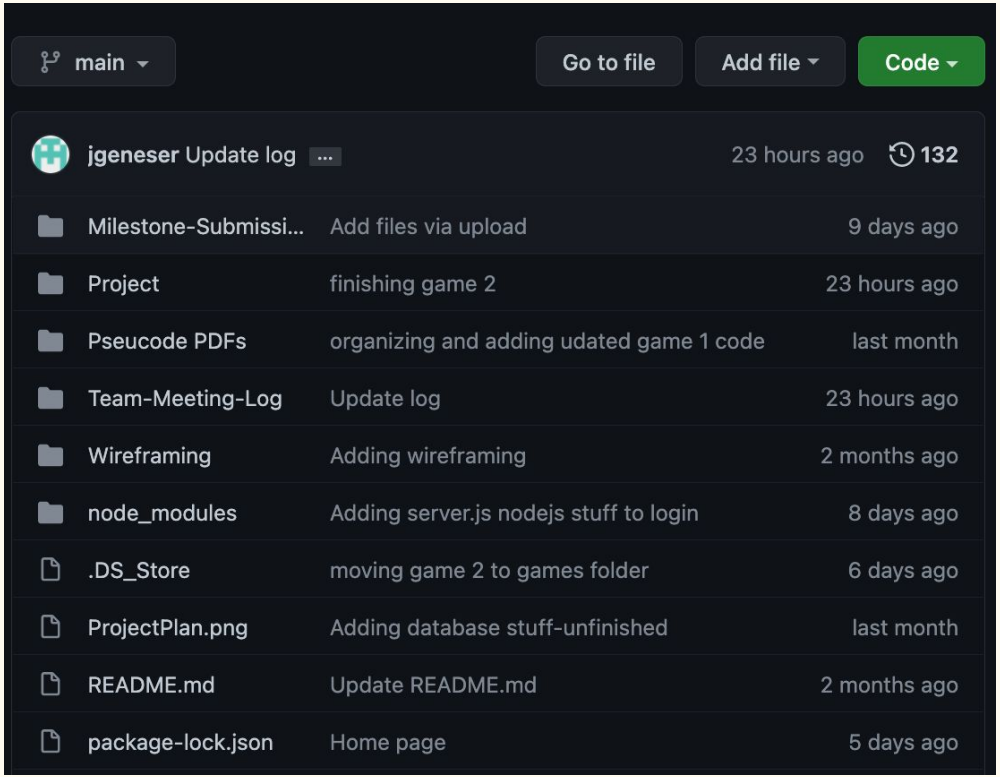
Riddles

Test your critical thinking skills with challenging riddles!

DC

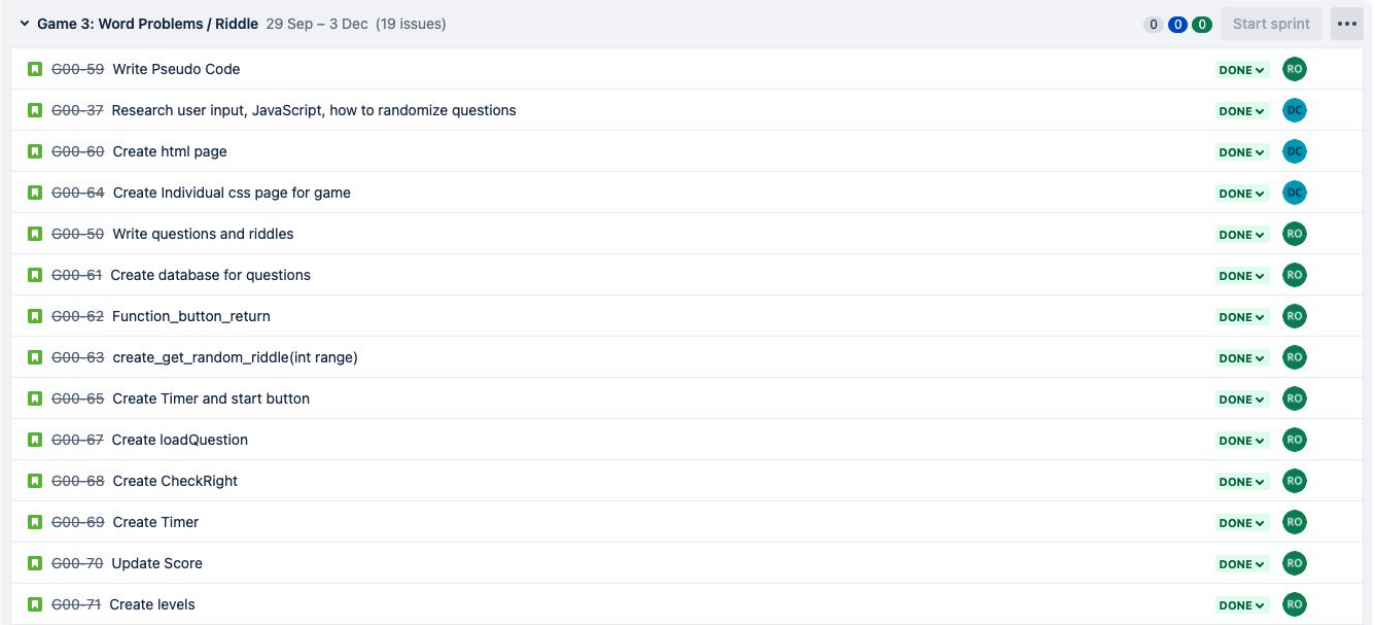
Github Use

- Our main recourse for centralizing our project
- Allowed us to work on the same project from our own devices



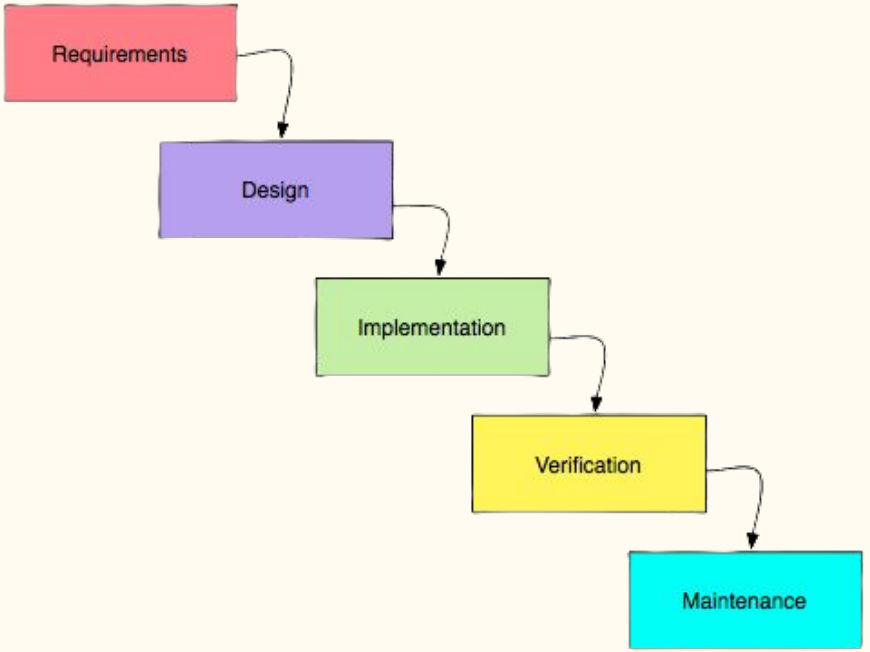
Jira Use

- Allowed us to organize a timeline and delegate tasks to specific individuals.
- Found texting in a group chat to be more efficient.



WaterFall Approach

- Allowed us to focus on what we were learning in class with what we were working on with our project.
- As our games are targeted for a younger audience, good constructive feedback can be hard to come by.



R

Architectural Diagram

Writing To Database

- Registration
- Game scores/attempt storage
- Supervisor modifications in Scores

Pulling/Comparing From Database

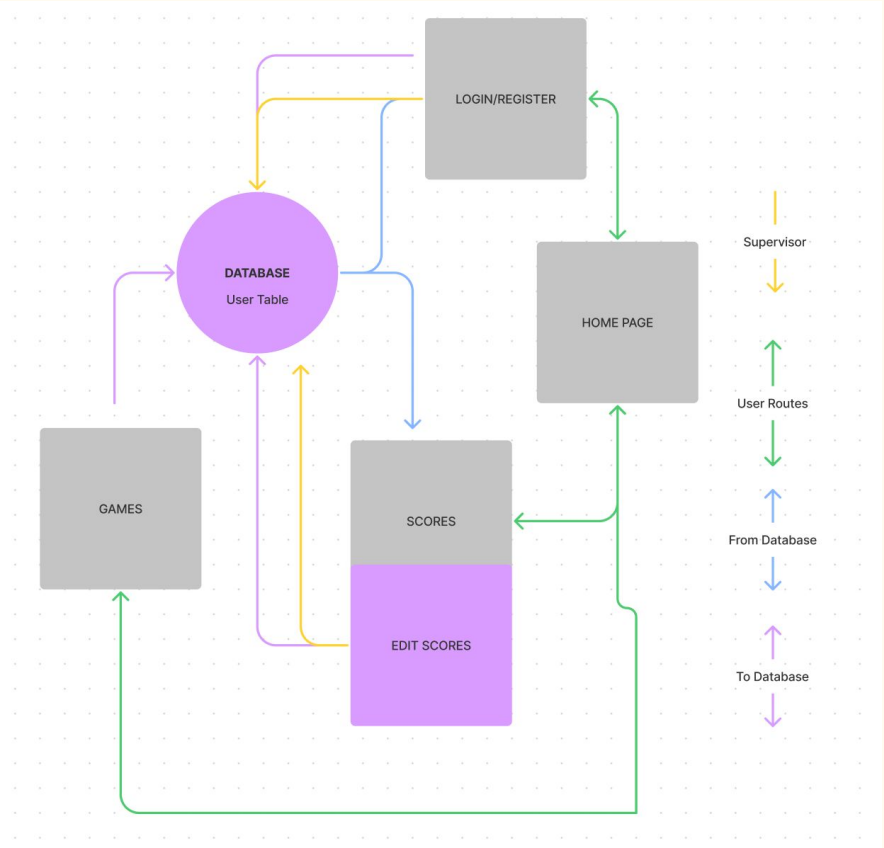
- User Login
- Pulling scores

User Paths

- Paths to/from each page

Supervisor

- Specific accounts made with supervisor variable
- Can report people in leaderboards and edit their score



C

Challenges

Tool Use: Mainly in terms of back-end work, the initialization of learning the best tools for our project and implementing them was a struggle. In relation to planning for our project, we had to switch from assigning initialization of the database on one member to having other members contribute and research. This ended up allowing us to get our database up and running.

Connecting Everything: In terms of styling, we ran into a few issues with getting everyones independent contributions to work together. Adjustments in styling had to be made and it created a challenge due to not knowing exactly how other members' styling and scripts worked in general.

PostgreSQL

```
<link rel="stylesheet" type="text/css" href="../Style/login.css" />
<link rel="stylesheet" type="text/css" href="../Style/styles.css" />
```

K

Testing

- User Acceptance Tests
  - Made sure website was easy to use
  - Double checked that the listed features worked as intended
    - Log in
    - Create new account
    - Completed game writes to database
    - Top scores can be viewed
    - Supervisors have the ability to flag accounts

DC

Group Members

- Riley O'Byrne - Worked on Game 3, wrote the majority of the javascript for game 3, worked a bit on the docker-compose, server.js. Worked on testing.
- Jules Geneser - Worked mainly on the development of Game 2, specifically the problem generation and score calculation. Also assisted in the html & css of Game 2.
- Davis Cohen - Worked on Game 3, mainly styling and presentation as well as plan for functionality. Assisted with getting started on the backend implementation as well as testing. Worked on figuring/solving back-end creation problems.
- Chelsea Stockberger - Initialized back-end functionality and created stylesheets, designed pages for login, leaderboards, and home, worked on server.js and docker-compose, created/drew graphics for icons, boat game, etc. Worked on pulling info for use from database and pushing into it. Worked on Heroku got initial project to be on heroku
- Duke - Created Game 1, finalized functionality of pulling/posting back-end information such as in server.js, leaderboard page, etc. Created working scripts for back-end and managed final steps of the project, including releasing the app on heroku.
- Kylie Elbert- worked on game 2: made the functions for the buttons, checking if the answer was correct, progressing to the next level, the design of the webpage, connected the game to the server