

Milestone 7

Software Dev

Group: 011-01

## **Epic Math Games**

Duke, Riley, Davis, Chelsea, Kylie, Jules

### **Description:**

Our project is a website for children (or anyone) to learn math in a fun way. It consists of three games: Riddles, Flashcards, and the Boat Game. It also includes a leaderboard page where top scores from each game may be viewed.

We have a functional login/registry system, where each users' scores are kept whenever they play a game. There is also a supervisor account in which can flag players for 'cheating', for example, if they see a user with a score of 99999 on the scoreboard.

Our project uses NodeJS and PostgreSQL in order to handle back-end tasks and information. The website is also available through Heroku, so anyone can access it anytime.

## Tracker:

<https://csci-3308-fall21-group1.atlassian.net/jira/software/projects/G00/boards/1/backlog>

Game 1: Boat game29 Sep – 4 Dec (6 issues)

This is the creation of the first game, which is the easiest level. Once the user passes the three levels within game 1, the next level will unlock!

5

6

4

Start sprint

...

G00-2

As a new user I want to create a new account, so I can access the application

REGISTRATION-PAGE

5

TO DO

...

G00-28

Research foundation on how to create a 2D game

1

DONE

...

G00-38

Research HTML and CSS that is essential to making a 2D game

3

DONE

...

G00-39

Start outlining the basic code

3

IN PROGRESS

...

G00-41

Add basic graphics

5

IN PROGRESS

...

G00-42

Add more detailed code

TO DO

...

+ Create issue

Game 2: Math Flash Cards29 Sep – 3 Dec (9 issues)

The second game, where binary or math problems are displayed on the screen. There is a set time limit to see how many conversions that an individual can do.

0

0

4

Start sprint

...

G00-36

Research coding and developing a quizlet styled game where you are asked a question and you insert your answer, and then you are prompted with a new question

1

DONE

...

G00-45

Research how to generate problems

1

DONE

...

G00-46

Research how to keep time

1

DONE

...

G00-49

Research how to code random generator in java

1

DONE

...

G00-43

Keep score

DONE

...

G00-44

Problem generator

DONE

...

G00-47

Finalize design and aesthetic

DONE

...

G00-48

Research HTML for webdesign

DONE

...

G00-66

Add game's HTML page to home's HTML page

IN PROGRESS

...

+ Create issue

Log in Page / data base4 Oct – 15 Oct (6 issues)

Create log in HTML log in page that logs in, creates new accounts, and access the database

0

0

0

Start sprint

...

G00-33

Research databases

DONE

...

G00-34

Research login pages

DONE

...

G00-56

Create html log in page, includes log in, create account

DONE

...

G00-57

Create create new account feature

IN PROGRESS

...

G00-58

create sign in feature

IN PROGRESS

...

G00-78

Create database

DONE

...

+ Create issue

Game 3: Word Problems / Riddle29 Sep – 3 Dec (18 issues)

0

0

0

Start sprint

...

G00-59

Write Pseudo Code

DONE

...

G00-37

Research user input, JavaScript, how to randomize questions

DONE

...

G00-60

Create html page

DONE

...

G00-64

Create Individual css page for game

DONE

...

G00-50

Write questions and riddles

DONE

...

G00-61

Create database for questions

DONE

...

G00-62

Function\_button\_return

DONE

...

G00-63

create\_get\_random\_riddle(int range)

DONE

...

G00-65

Create Timer and start button

DONE

...

G00-67

Create loadQuestion

DONE

...

G00-68

Create CheckRight

DONE

...

G00-69

Create Timer

DONE

...

G00-70

Update Score

DONE

...

G00-71

Create levels

DONE

...

G00-72

Sync timer to actually end the game

DONE

...

G00-73

Connect end score to database

DONE

...

G00-74

Make so start game only when load for first time

DONE

...

G00-76

Make database for questions

DONE

...

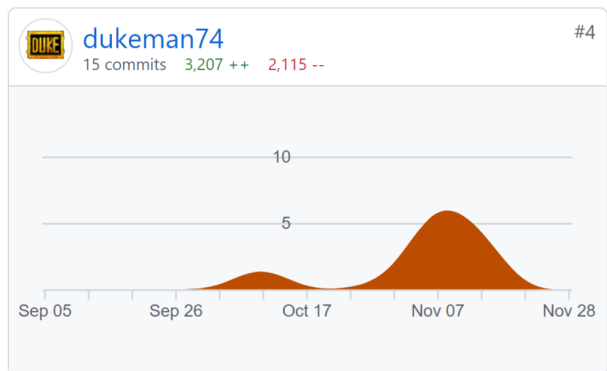
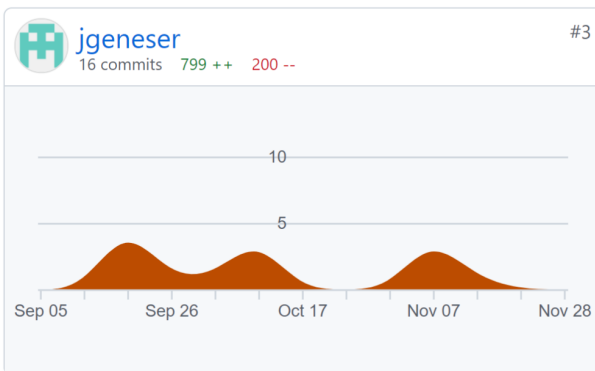
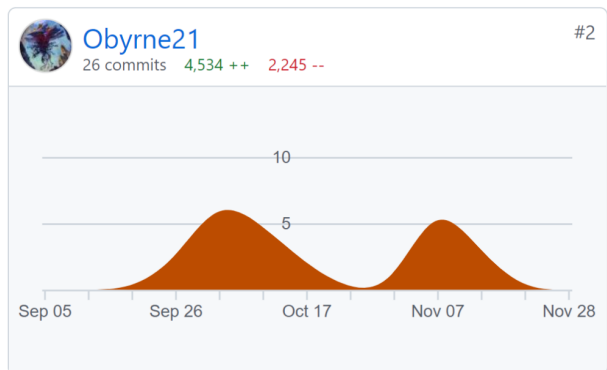
## Demo video:

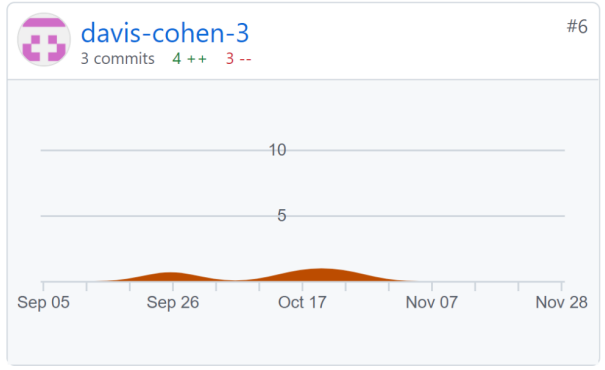
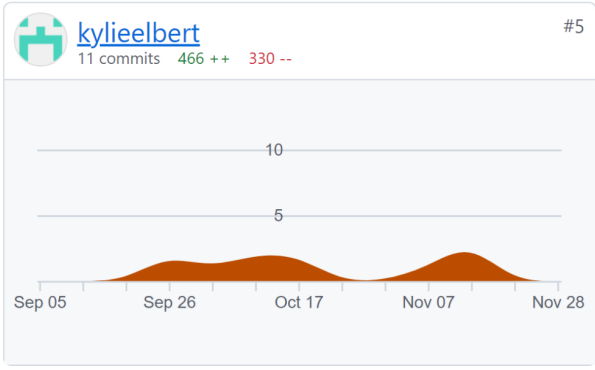
[video](#)

## VCS:

[github repo](#)

## Contributions:





### Duke Manchester:

Coded game 1, Implemented a backend using nodeJS, Implemented a leaderboard system using postgresSQL served by the backend, configured everything so that heroku would be happy with it and got the whole system up and running on heroku, and created demo video.

### Riley O'Byrne

I worked on the first draft of our html and style page for the website, though we would later use another draft. I made the majority of the javascript for game 3. For game 3, I created a function that will produce a random question within a certain level (the level of the user in game 3). I worked on the docker-compose file, contributed to creating the database, contributed in helping to plan/initialize the backend, and helped to connect the backend to game 3. I also tested the website and it's many features through user acceptance testing.

### Chelsea Stockberger:

Initialized back-end functionality with the docker-compose file and using NodeJS. Created/drew graphics for the boat game and general website icons/images. Designed

and created HTML pages for leaderboard, home, login. Created stylesheets/CSS for the html pages. Worked on initialization of heroku. Worked on general back-end stuffs

#### Davis Cohen:

I worked on the html and styling of game 3 and also worked on some of the javascript for game 3 to get the game running, however Riley ended up finishing up the javascript for the most part. I also helped in planning the way game 3 would work. I helped Duke figure out how to connect the back end to the front end through figuring out how to get the score of game 1 and figuring out how to submit that to the database. I also worked on user-acceptance testing with Riley to make sure everything was working properly for new and existing users. Once Duke got on track he and Chelsea really worked together to figure out how to get everything working functionally, however I was also recovering from knee surgery the final week, so it was tough for me to spend a lot of time each night that week.

#### Jules Geneser:

Throughout the project I work majorly on game 2 with Kylie. We not only worked on the planning of the game and how everything would work together but we also worked together on the css, html and javascript. I mainly did the programming for the score increments and problem generation as well as the programming for some of the buttons. Like I said before Kylie and I both worked together on every aspect of the

game! I also assisted in the making of the scoreboard but as we approached the end of the game, as a team we decided to go in a different direction.

#### Kylie Elbert:

Jules and I worked on game 2 together. I also initially started working on the backend of the scoreboard, but we didn't end up using it in the end. For game 2, Jules and I worked on all aspects of it together. I did the css that was included in the html file, specifically for the buttons and the "window" that we put the game in. For the javascript, I mainly worked on checking the users' answers and increasing the level like how to reset the level score and making sure that the game score didn't increment unless the user was actually done with the level and not going to restart it. Duke helped me get the scores to the backend for the scoreboard once the user pressed the finish button. Jules and I worked through the entire game together by sharing the files on VS Code and sitting with each other to talk it all through.

#### **Deployment:**

Link: [heroku app](#)