Milestone 7

Software Dev

Group: 011-01

Epic Math Games

Duke, Riley, Davis, Chelsea, Kylie, Jules

Description:

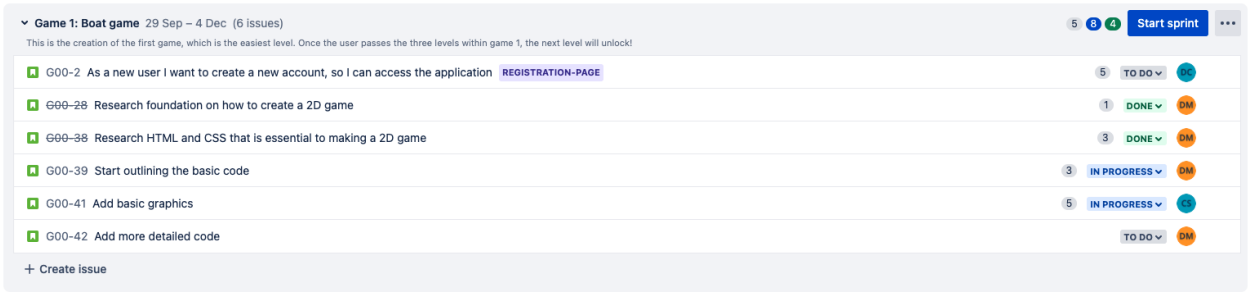
Our project is a website for children (or anyone) to learn math in a fun way. It consists of three games: Riddles, Flashcards, and the Boat Game. It also includes a leaderboard page where top scores from each game may be viewed.

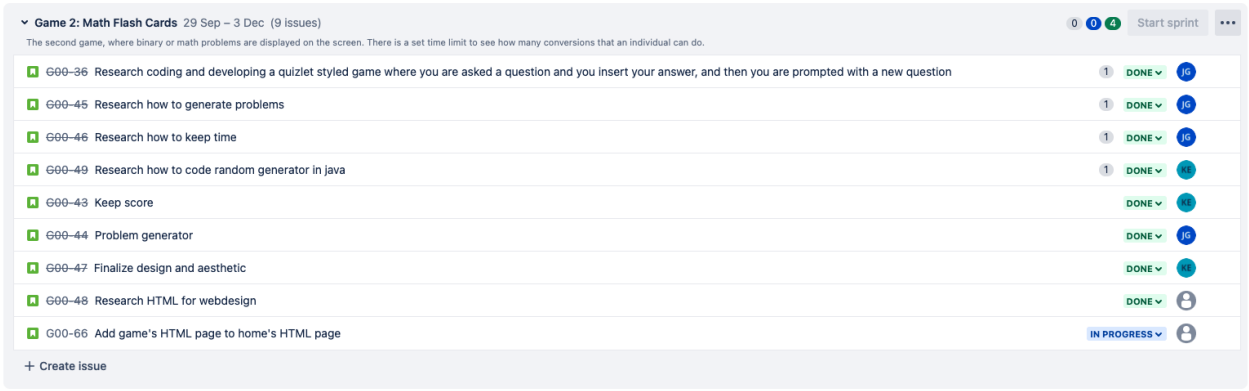
We have a functional login/registry system, where each users' scores are kept whenever they play a game. There is also a supervisor account in which can flag players for 'cheating', for example, if they see a user with a score of 99999 on the scoreboard.

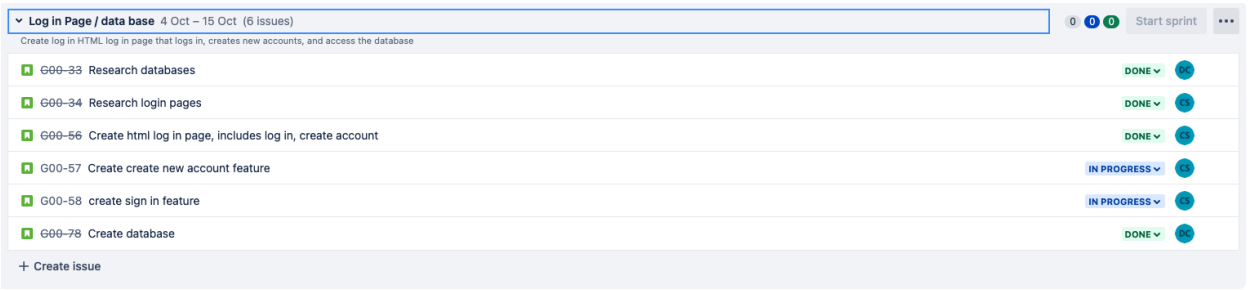
Our project uses NodeJS and PostgreSQL in order to handle back-end tasks and information. The website is also available through Heroku, so anyone can access it anytime.

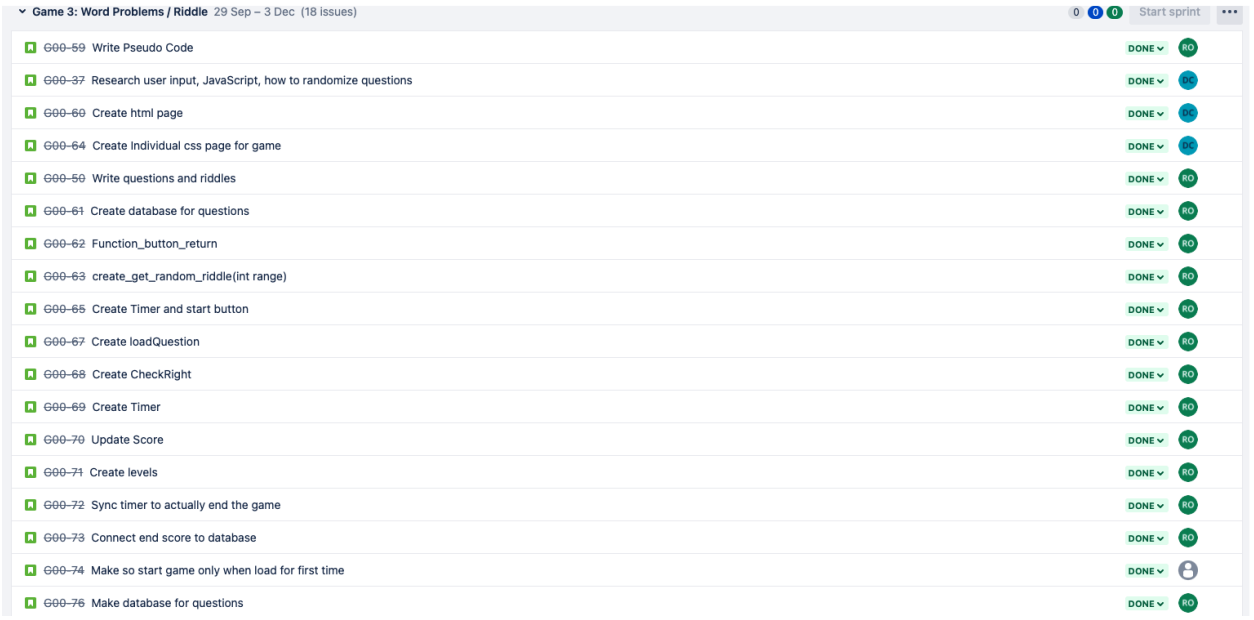
Tracker:

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









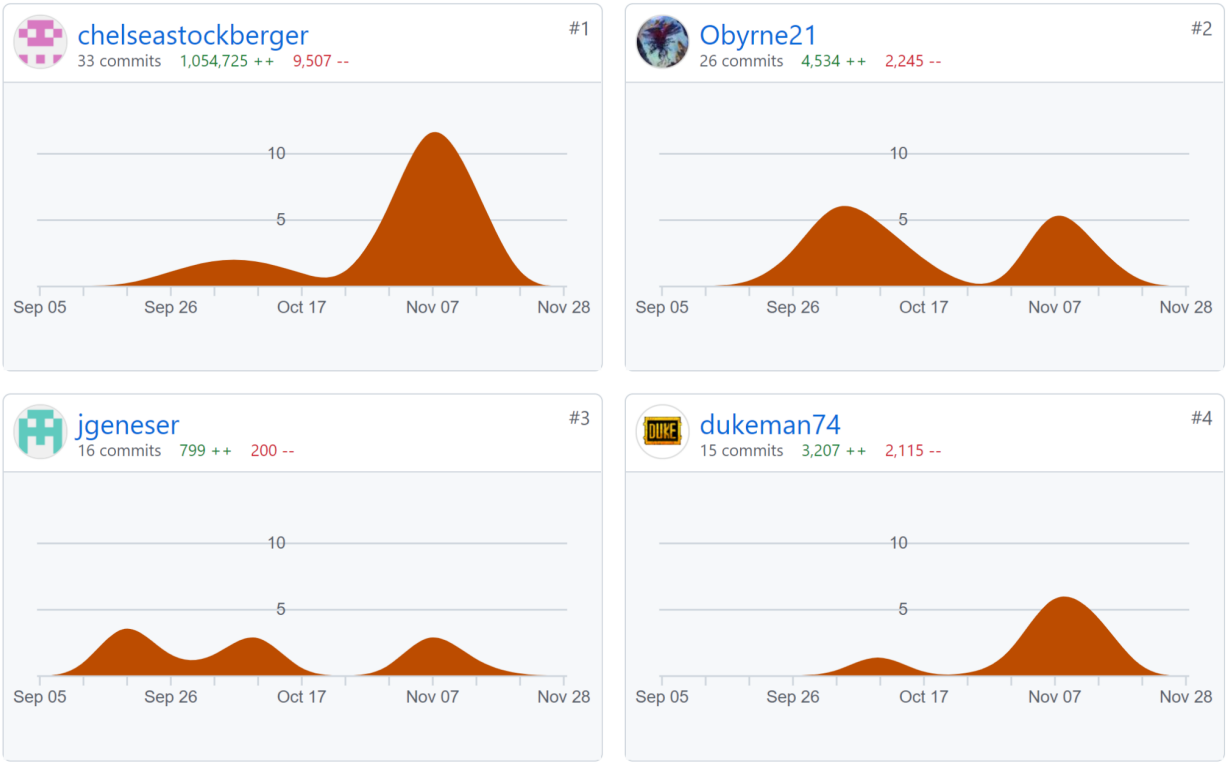
Demo video:

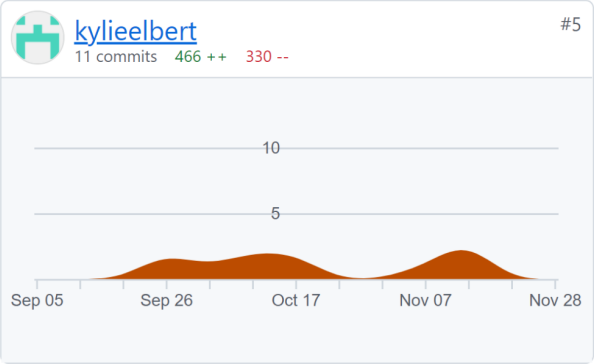
[video]

VCS:

[github repo]

Contributions:




kylieelbert
11 commits 466 ++ 330 --
#5
10
5
Sep 05
Sep 26
Oct 17
Nov 07
Nov 28


davis-cohen-3
3 commits 4 ++ 3 --
#6
10
5
Sep 05
Sep 26
Oct 17
Nov 07
Nov 28

Duke Manchester:

Coded game 1, Implemented a backend using nodeJS, Implemented a leaderboard system using postgreSQL 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