Title: Tic-Tac-Toe Max

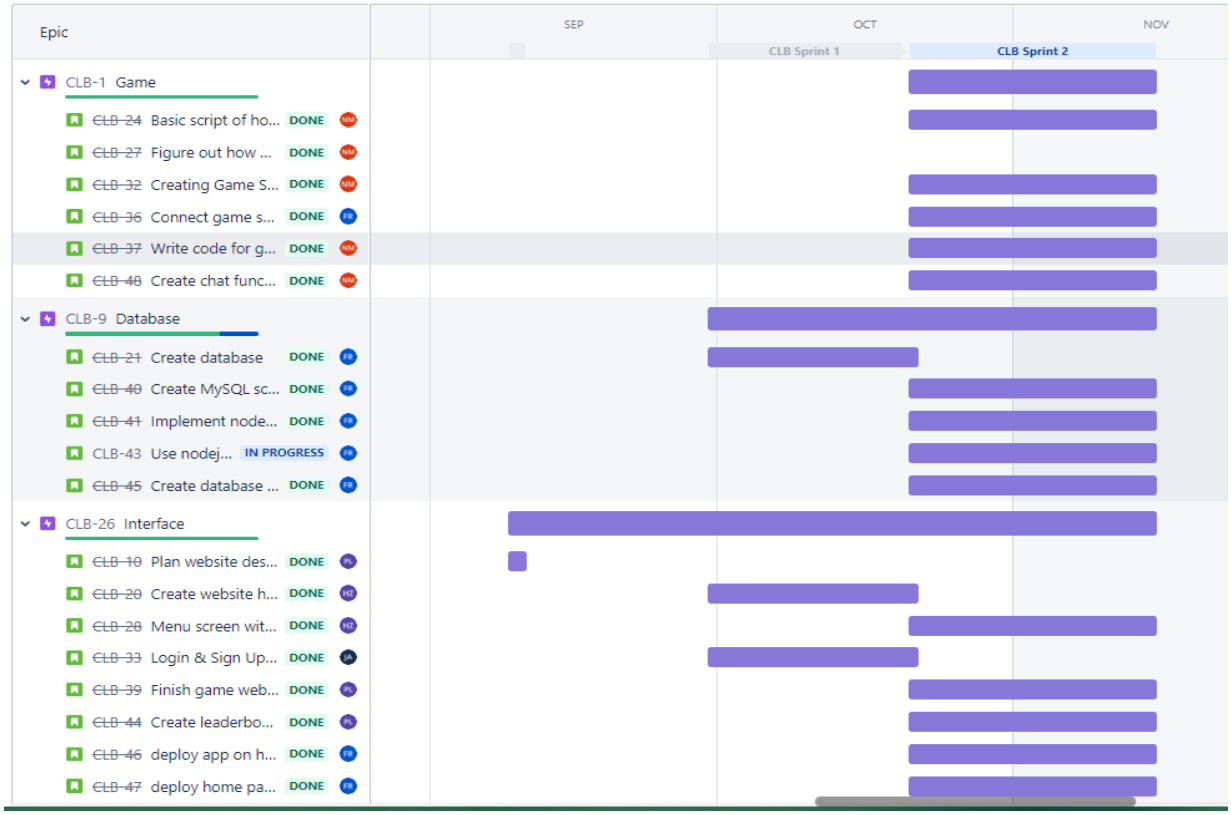
Team: Henock Zemenfes, Freddy Rodriguez, Peter Lee, Nicholas Mason

Project Description:

- For our project, we decided to create a TicTacToe game called, TicTacToe Max, a grid game with a bit of added features to make the original slightly more interesting. Our project contained a mix of HTML (website) and JavaScript files (server). Differences in our game compared to the original was the rule of 3 in a row which we changed to 5, in order to add a bit of difficulty and challenge to the game. Another added feature to our game was the ability for players to take spots on the grid which resulted in much longer and competitive games. We started the project by designing a website that gave a chance for users to learn the new rules of our game as well as hyperlinks to other functions of our project (leaderboard, rules, etc). We wanted users to interact with one another while playing and decided to create a chatbox embedded in the game which gave users a multiplayer experience. On top of that, we included a leaderboard to give again, a competitive platform amongst players. Ultimately, our project provides a different approach to a classic game but we felt that our added features gave users a unique experience in playing our project; TicTacToe Max.

Project Tracker:

- Jira
- https://csci-3308-fall21-013-02.atlassian.net/jira/software/projects/CLB/boards/1/roadmap?selectedIssue=CLB-1



Video

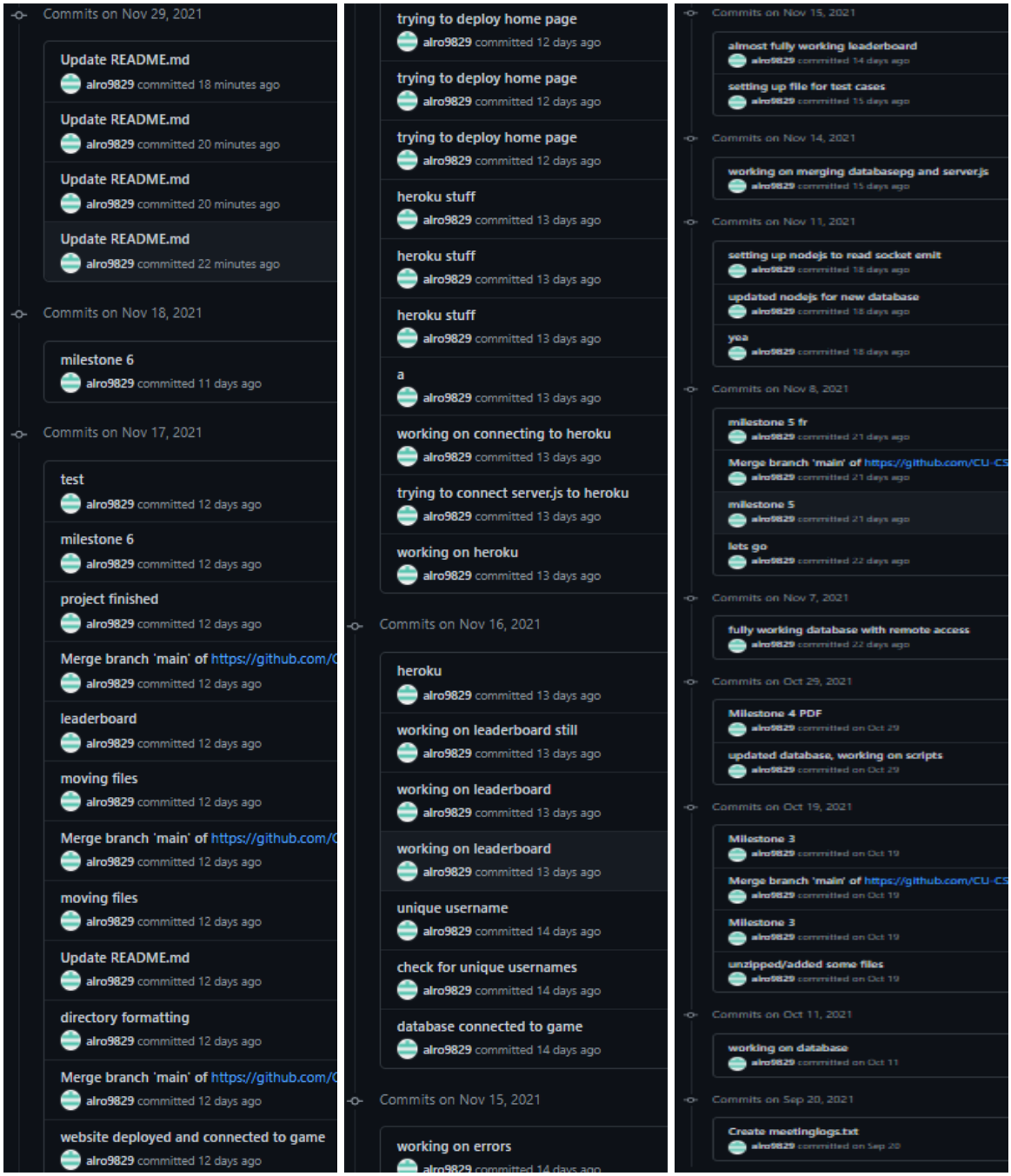
- https://www.youtube.com/watch?v=07ucXsubx14&ab_channel=Apten

VCS

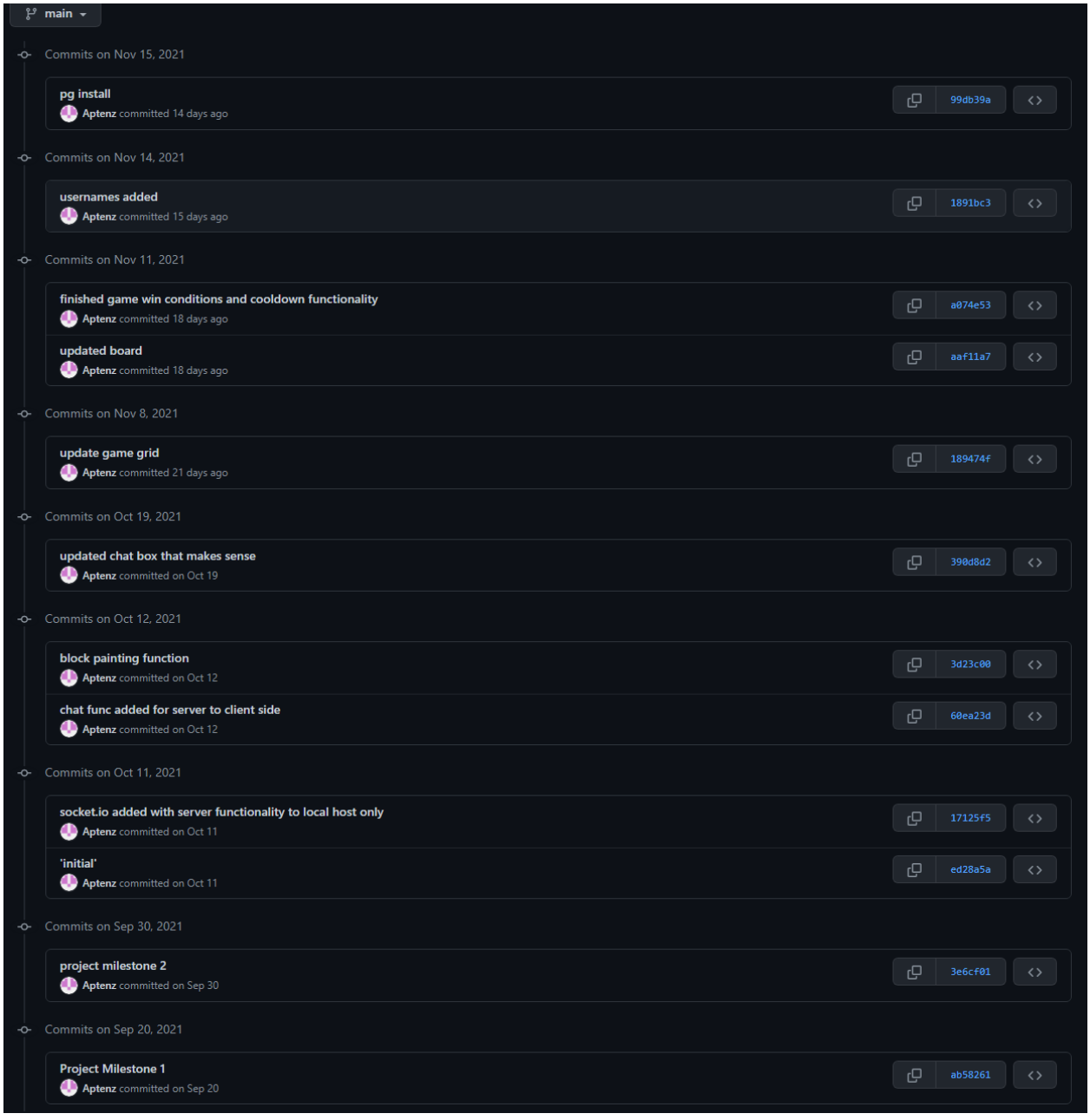
- https://github.com/CU-CSCI-3308-Fall-2021/CSCI-3308-Fall21-013-02

Contributions:

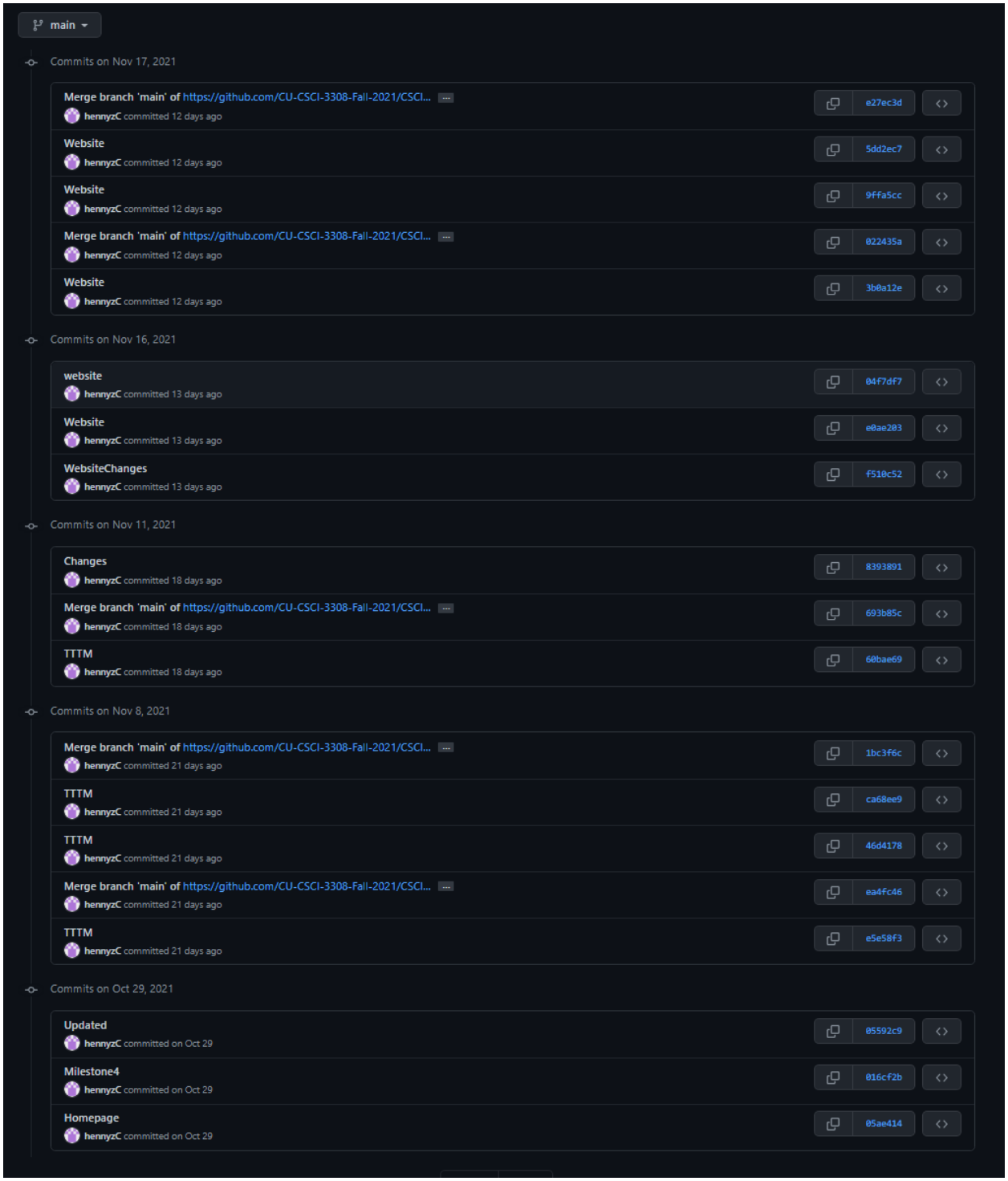
- Freddy: Created postgres database server on Heroku, created leaderboard server (js, html, php) and deployed it on Heroku, deployed homepage and game on Heroku, connected leaderboard.js and server.js to database server (js, nodejs: pg, express), hyperlinked leaderboard and game to homepage (html), wrote function to check for unique usernames in server.js (js)



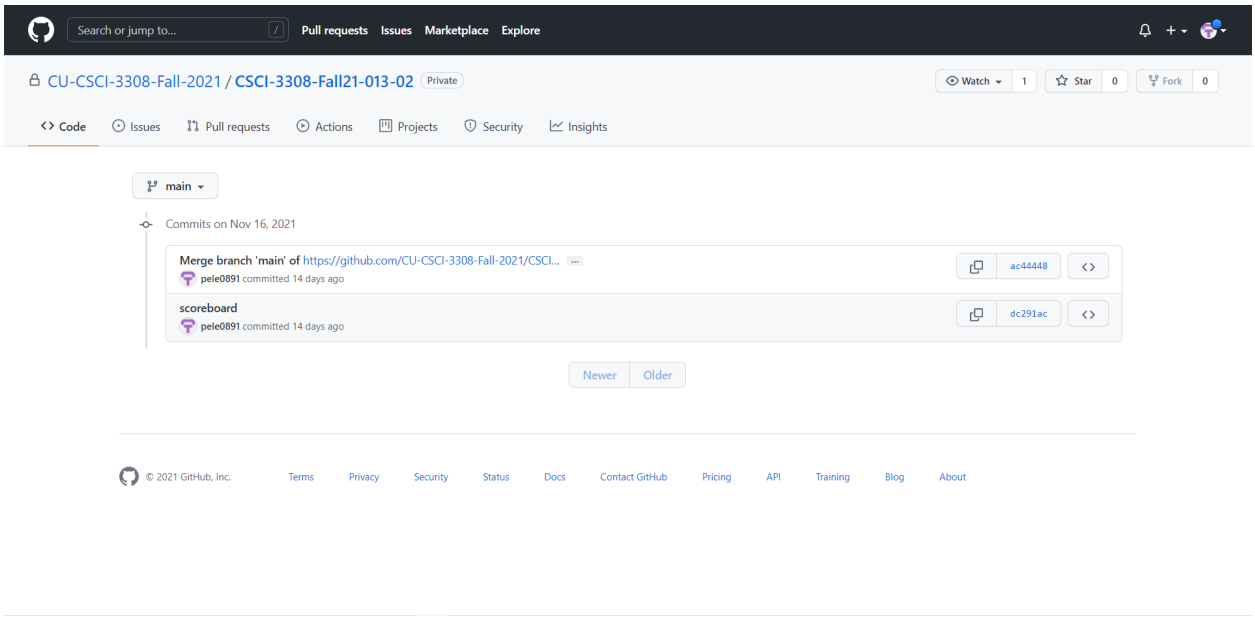
- Nicholas: Created game and socket.io for multiplayer functionality for both the frontend and backend. For the game, created the conditions that worked in tandem with socket.io in order to have live updates to clients to create a multiplayer experience. Also, made the chat function as well with socket.io for users to communicate with each other as well as showing information pertaining to the game for users to see in the chat box. Used html, socket.io, node.js, js to create the game for users to play online.



- Henock: I created the homepage website that connected to both the actual game and the leaderboard. The game was connected through the start button which was highlighted in the middle of the page. The leaderboard could be accessed from the navbar which was at the top right of the page. The navbar also contained hyperlinks to the rules of the game for the user. The HTML code would behave so the user could click on rules and would navigate to that part of the homepage.



- Peter: During the project my main task was to develop the homepage and the leaderboard for the application. For the homepage, I used bits of HTML to create the foundation for the homepage. Also using CSS to style the homepage and improve the UI. The main feature in the homepage was to incorporate a moving nav bar that would stay at the top of the screen while the client would scroll around. For the leaderboard I incorporated HTML to create the format and javascript to dynamically connect the wins from the database to the leaderboard.



Deployment

- Home page: https://clb-home.herokuapp.com/
- Other pages (accessible through home page):
    - Leaderboard: https://clb-leaderboard.herokuapp.com/
    - Game: https://fierce-basin-18108.herokuapp.com/