Project Milestone 7

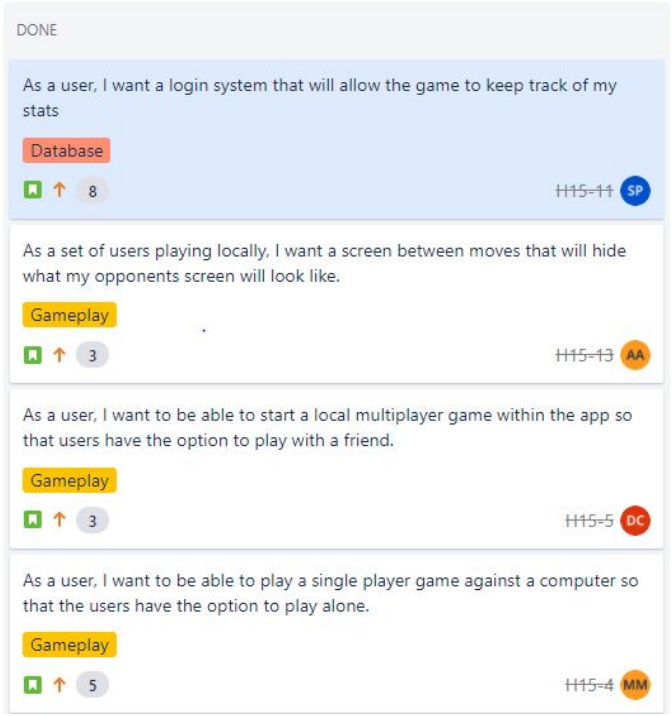
CSCI 3308: Software Development & Tools
Team: 113-5
Title: History Through Battleship
Who: Salvatore Pacifico, Kai Schuyler Gonzalez, Andres Montealegre, David Crockett, Matthew Ruiz Diaz

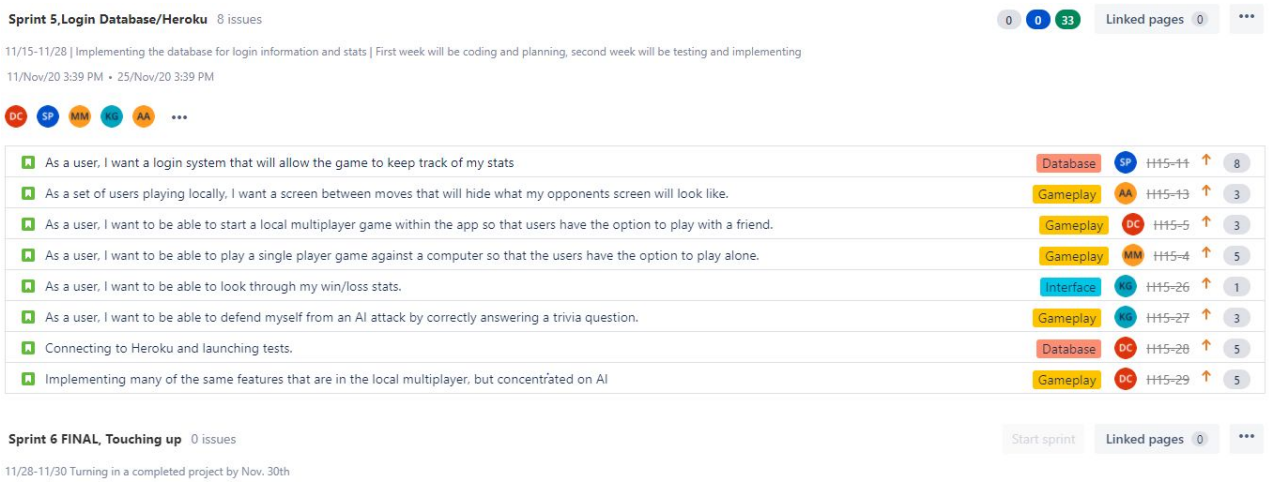
Project Description:
History Through Battleship is an educational web-game developed to test and refine students' history skills. The application, which features both single player and local multiplayer, runs similarly to your typical battleship game. However, the game has a fresh educational twist that is meant to offer practical use. When launching a missile, a user must successfully answer a trivia question. If the user incorrectly answers their question, they lose their turn.

Within the single player game mode, the user plays against an intelligent AI system that can launch missiles based on previous hits/misses. This competitive system can be a hard match for even the smartest history buff. The local multiplayer mode is played between two users on the same system. Once player 1 has placed his/her ships, they pass their turn to player 2 who will do the same. When both players' ships have been placed, they can start the game. The users will then alternate turns by clicking the "End Turn" button at the bottom of the screen.

On the backend, History through Battleship utilizes a database to keep track of user's usernames, passwords, and stats. This database is updated as new users register. Similarly, users stats are updated as trivia questions are answered correctly and incorrectly. Furthermore, History Through Battleship connects to the Trivia DB API and retrieves the trivia questions seen within the game. These questions could be changed to other subjects as well.

Project Tracker:





Screenshots of our final sprint being completed in our Jira board.
Link to Jira:
https://csci-3308-fa20-113-5.atlassian.net/secure/RapidBoard.jspa?projectKey=H15&rapidView=1&view=planning.nodetail&atlOrigin=eyJpIjoiNzk2N2I1NjBmYTVhNGMwYTg2ZGRkYTcxNWQxYTcxMTUiLCJwIjoiaiJ9
*When completing the final sprint on JIRA, we are not sure if it was an intended feature or a bug but the project was closed and everyone was taken off the shared list. I will reshare with our TA Carl Mueller so you can have access to the link, but every team member did work in the JIRA although it only shows one of us in there.

Video:
https://github.com/CSCI-3308-CU-Boulder/113-5/tree/master/milestones/ProjectMilstone7_113-5/Video%20Demo

Git Repository:
https://github.com/CSCI-3308-CU-Boulder/113-5.git

Contributions:
All of us must write a little about our commits below!
Matthew:
Throughout this project I worked with HTML, CSS, JavaScript, Figma, and the Heroku CLI for my contributions. Firstly I created wireframes for the main pages of our web application using Figma. Kai and I both created a functional modal from the bootstrap implementation and combined them using my CSS and his HTML. This eventually evolved into the modals that pop up when a player clicks a grid square. When I was working with JavaScript I modified the javascript functions that Sal created for single player and removed the embedded AI functionality without breaking them. I then also added multiple changes to allow the JavaScript file to accommodate more than one player. These modifications also had to be made to the HTML multiplayer page that David had created. Other than that I had to add many features that are present in multiplayer like hiding the board, adjusting the rules, and making sure each player sees only what they should see during their turn. The features that required the most debugging were the end game feature where the game is aware when end game conditions are met, and being able to display hits on ships to both players without revealing the location of the rest of their ships. Sal and I worked on debugging these and were successful. Another feature I worked on was making sure a player can only do one question per turn, and given that multiple functions related to clicking on, answering, and validating a question, each function had to be made aware of whose turn it was, and whether they had already used it.
https://github.com/CSCI-3308-CU-Boulder/113-5/tree/master/milestones/ProjectMilstone7_113-5/MatthewCommits

Sal:
Throughout this project I worked with HTML, CSS, JavaScript, Heroku, jQuery, and Embedded JavaScript templates for my contributions. I worked on the beginning stages of the HTML/CSS for the Settings and Main Menu as well as the initial CSS for the Abouts Us, Login, and Register. Kai and Andres were able to clean up much of the styling and CSS aspects toward the end. I also connected the HTML pages before our conversion to ejs files, as well as implemented the side navigation bar for the game. For the singplayer's JS, I implemented all of the ship placement, start game and end game validation, AI gameplay, and checkers for all possible errors a user could encounter (placing a ship out of bounds, starting a game before ships are placed, etc.). I also worked on the ship and hit/miss HTML/CSS for the gameplay. I worked closely with Matthew to then implement the local multiplayer. Matthew took the lead and I was able to help clarify and fix any errors or necessary functions for the JS, as the functions for single player and multiplayer were closely related. I was also involved in the manual testing of both the single and local multiplayer games.
https://github.com/CSCI-3308-CU-Boulder/113-5/tree/master/milestones/ProjectMilstone7_113-5/SalCommits

Andres:

Throughout our project, History Through Battleship, I focused on the frontend design of the battleship gameplay/End Game Screen, users' statistics, and creating a responsive design. Between each of these tasks, I was able to practice my HTML, CSS, and JavaScript skills. Within the project, I was able to create several basic templates for what we wanted from gameplay/end game/statistics screens. I then worked closely with Sal to have the battleship front end working with his backend gameplay. After completing the battleship board, I then moved onto working with the end game screen and statistics page. These were both formatted to match the battleship aesthetics. The statistics page was then passed on to Kai, who connected the screen to the statistics database. After creating these individual screens, we noticed that differing screen sizes dramatically affected the game's visual aspects. To counter this, I went through our CSS Pages and created a responsive design by using the view width property rather than fixed sizes. This allowed us to create a design that would look similar, regardless of the user's screen size.
https://github.com/CSCI-3308-CU-Boulder/113-5/tree/master/milestones/ProjectMilstone7_113-5/Andres Commits

David:

During development for our project, History Through Battleship, I was involved in several sub-projects and tasks that aided in the completion of the application. At the start of the semester, I developed the HTML for the About Us page of the application, which was then modified by Sal, Kai, and Andres, using CSS to match the aesthetic of the entire application. I then proceeded to create diagrams for both the front end design and the architecture of the application using the diagram editor from Zygomatic. The front end design showcased how our users could navigate from page to page as well as the main functionality of each page. The architecture diagram provided a larger picture of how each piece of the whole application functioned with the others. I also came up with the test plan for debugging our main three functional pieces of the project, which were the user database, the multiplayer gameplay, and the single player gameplay. Lastly, I converted both the single player HTML and CSS to multiplayer versions. Javascript was then added by Matthew to complete the multiplayer version of the application.
https://github.com/CSCI-3308-CU-Boulder/113-5/tree/master/milestones/ProjectMilstone7_113-5/David Commits

Kai:

Throughout the project I focused mostly on the node.js application and integrating the PostgresQL database. At the start of the project I made the login/register page HTML/CSS using Bootstrap. I also added the basic interface for our trivia feature, which required making requests to an external API called *Open Trivia Database* to retrieve a JSON file that populates a multiple choice question that pops up in a modal. I then took the HTML/CSS everyone worked on and converted them into EJS template files and merged the css files into one main css file to create a basic Express app. After our Express app was working, I focused on integrating NPM packages Passport, Bcrypt, Node-Postgres and others to add login/register/authentication features and a stats page, which required the implementation of a postgres database to keep track of user stats. At the end of the project I built a simple shell script that is included in the git repository that makes the app easier to host locally.
https://github.com/CSCI-3308-CU-Boulder/113-5/tree/master/milestones/ProjectMilstone7_113-5/KaiCommits

Deployment:
We added Carl Mueller, our TA, as a contributor to the Heroku just in case any depth of the Heroku needs to be looked at. The link below brings you to our Heroku deployed app. Instructions for running locally can be found in the README in our Git repository.
https://history-through-battleship.herokuapp.com/login
https://github.com/CSCI-3308-CU-Boulder/113-5