

CS4530 Final Project: "Covey.QuizShow"

Group 212: Bradley Karmin, Max Correia, Neo Isak, Nick Sokolović

Our Feature: Covey.QuizShow

In the version of Covey.Town we were given, there were interactables for games of Connect Four and Tic Tac Toe. They allowed players to interact with the space by competing against one another. However, our group came to the conclusion that there were no single-player game elements in Covey.Town for users to interact with while waiting for their peers. We therefore asked ourselves: what type of game would be best oriented to accommodate for both a single player and multiplayer experience?

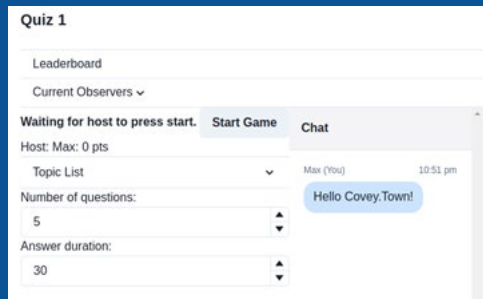
We came up with the idea of **Covey.QuizShow**. It is an interactable that allows for both single and multiplayer interactions. Players can set the amount of questions and the time between questions per game, and they can compete against one another to see who can score the highest in this race against the clock and their peers.

Demo and Source

Our demo site is available at <https://spring24-project-s24-group-212.onrender.com/> and our source code can be found at <https://github.com/neu-cs4530/spring24-project-s24-group-212>.



Players are displayed the correct and incorrect answers after all players answer.



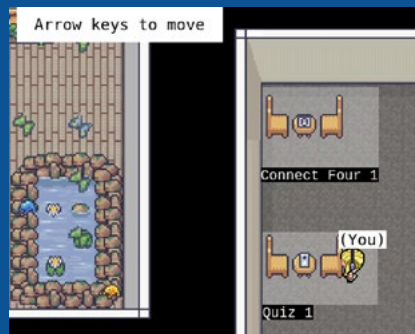
The host player has the ability to set the number of questions, answer duration, and the topics to select questions from each round.

Our Technology Stack & Design

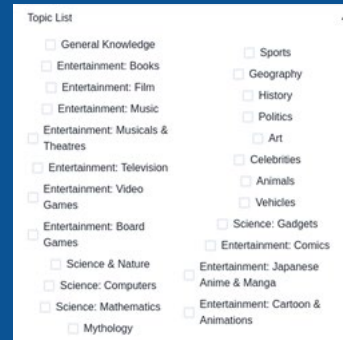
We implemented our interactable in the existing Covey.Town codebase. The area is represented on the map based on previously existing interactable areas. When a player joins this area, they are either designated the host or a regular player. The host has the ability to select topics, answer duration and the number of questions in a game. The area and board are rendered through Chakra UI elements. These rely on React Hooks to update the current question, player scores and the final scores of each of our players. Our continuous integration pipeline runs an automated test suite on the frontend and backend components through GitHub, and then deploys the site using Twilio and Render for the frontend and backend respectively.

Future Work

When compiling our project proposal, we also gave ourselves some optional implementation goals. Some of these that were not put in this implementation include giving players the ability to create their own questions and add them to our database, giving players the ability to filter questions by difficulty, preventing repeat questions from appearing in future rounds, and displaying the ratio of correct to incorrect answers in a round.



Users can join the quiz area in the Covey.Town overworld.



Users can select from a variety of topics. Questions are retrieved from a mix of a local database and a remote API.