CS4530 Final Project: "JukeBox Area"

Group 706: Adeyinka Adedewe, Joshua Lin, Michael Guo, Ashwin Rupakula

Our Feature: JukeBox Area

Covey.Town is a place where users are able to connect to a Town, which has a old arcade-style map with various areas that users can walk around. One problem that we saw was that while there is a viewing area present, there is not much of a place for users to potentially listen to music. What if users what to listen to music and share songs with each other and queue up songs for people to listen to?

We decided to then develop the new feature of now listening to music for our project: JukeBoax Area. A JukeBox Area is a part of the map thaat allows users to enter and start a music session with themselves and potentially with others in the area. This gives even more interactive elements for users to experience within Covey.Town.

Our Technology Stack & Design

We created this new JukeBox Area in the existing covey.town codebase. The Jukebox area is represented as an object in the tilemap and can potentially be changed up. The actual objects are created with the use of Phaser and these obejcts are contructed when the map is loaded up. When a user presses the space bar, there would be a white box to first tell them to sign in, afterwards the area would then show muliple components for users. There would be a search box for users to search musics to listen to or to add to a queue. The queue itself, would be be a list that would be updating when users are adding songs to the queue.
The voting portion would be on the left side and will allow people to vote on songs that they like, this relies of React hooks as it gets updated frequently similar to that of the queue. Our continuous integration pipeline runs an automated test suite primarly on the communication of the backend server to the Spotify API and on the frontend communication to the backend, all test were based of mocks as testing API was the most importing thing.

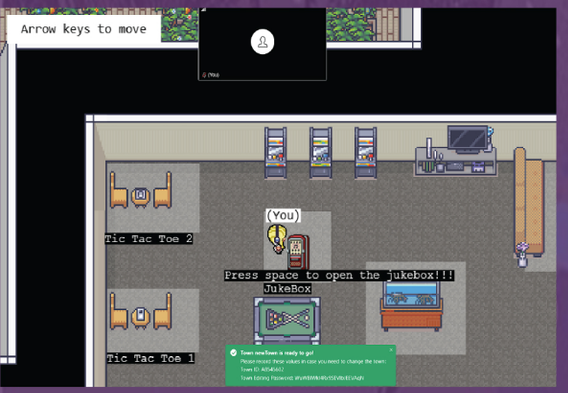
Searching for songs to play pause and add to queue

Demo and Source Code

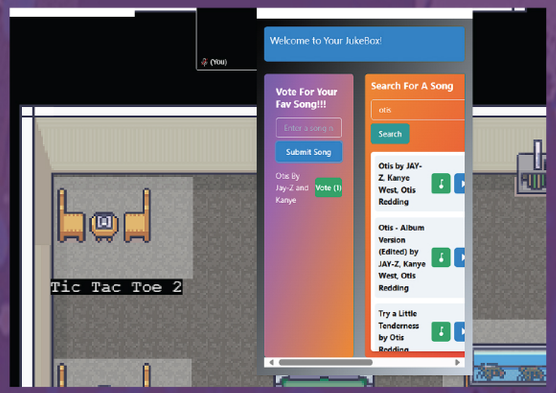
Our demo site is available at https://jukebox-gt0r.onrender.com/ and our code is at https://github.com/neu-cs4530/fall23-team-project-group-706/tree/final-submission

Future Work

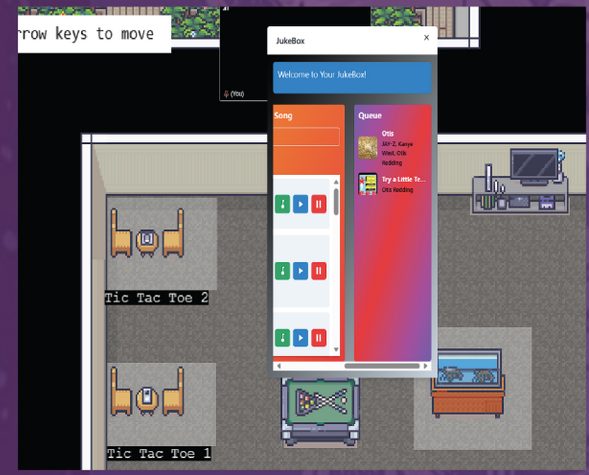
There was a lot of struggling going into this project, one thing was having good enough encapsulation with the express routes, at first we tried to link them to methods in our main class but soon found out communication to the front end was not working. So most of the work was done one the server.ts file. Additionally, while we were able to make songs be played, there was a heavy reliance on the Spotify API and it's restrictions, while it was not too hard to grasp, the API made it very hard to implement some of the original ideas we had. Future work might be to find loopholes around this, more research would be needed and having to find a way to have a centrilized music player where not everyone needs to have a spotify to use the jukebox. There is a work around but it invloves trust between users. Additionally, there was not visual way of showing a song is being played besides acutally hearing it, so finding a way to visualize songs being played. inally, having private rooms would be ideal with people being able to listen to songsat the same time but that went againt another of our user stories.



JukeBox area shown as a Phaser Object that the user can interact with



One of the features is being able to vote for songs, shows user input for song and vote for that song as many



This component is for the Queue allows the users to queue songs but is mainly for other users to see what people are