CS4530 Final Project: UNO
Group 209

Ayush Pissurlenkar, Francesco Duca, Nolan Mungovan, Lukas Savarese

Our Feature: UNO

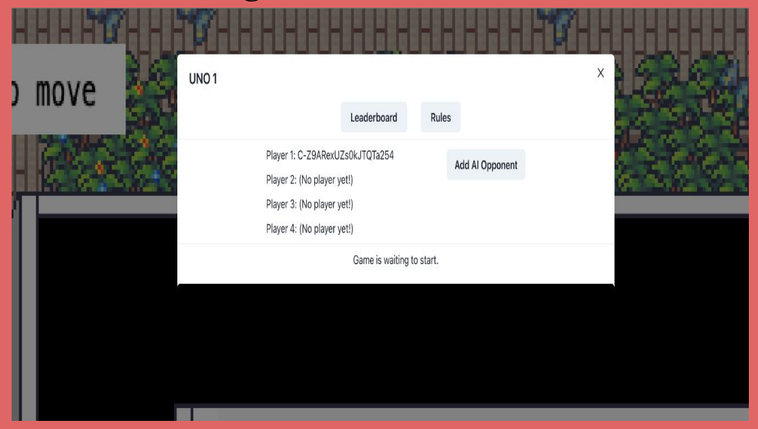
When determining what feature we wanted to add to Covey.Town, we started right away thinking about games. We thought that just TicTacToe wasnt enough for the town, and so decided to make a new game. We picked UNO for two reasons: First, we know that card games are very popular for coding, and have experienced coding them before. Second, we recognized that many people know the rules of UNO already, and therefore we would be offering a game that many people already know and love.

And so, we developed UNO for Covey.Town. Players can now enter an UNOGameArea, where they are able to spectate or join a game of UNO with up to 3 friends or random townsfolk. The rules of our UNO Game are very similar to UNO in real life, but we have a button to open the rules if anyone needs a refresher.

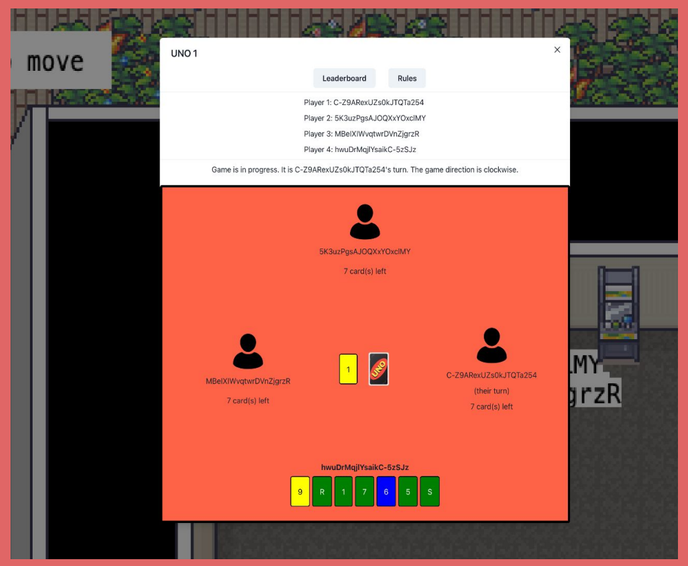
Past just playing with your friends, we offer the ability to press a button and add up to 3 AI players to the UNO game to play against. You get to pick the difficulty of the AI opponent, and they will play against you as though they are a real player. For this version of the game, we only have easy and medium AI implemented.

Additionally, we have connected our UNO Game to a database. Winners and losers of each game have their total wins and losses updated on the database. When in a game, the user may press the leaderboard button to see the top 10 UNO players across all towns.

Demo and Source:
Our demo is available at https://uno-frontend-9c12.onrender.com/ and our code at https://github.com/neu-cs4530/fall23-team-project-group-209.git



Waiting lobby for UNO



In progress UNO Game

When in a game, the player may click the leaderboard button to get a display of the top UNO players across all towns

Our Technology Stack & Design:

Our design for our game of UNO is handled very similarly to the existing game of Tic Tac Toe. We followed the Model, View, Controller strategy in order to implement a very common object-oriented design pattern.

The model in the backend holds necessary values and logic that handles creation and distribution of cards to players in the game along with handling moves from players and AI opponents. The controller handles calls from the view and sends respective calls to the model to progress the game. Emitting changes to listeners also happens in the controller. The view takes these values from the controller and displays it to the user in a way they can understand how the game is being played. It attempts to mimic the way a real UNO game would be laid out on a table.

The model with AI is held within the townService backend, while the controller and view are held within the frontend of covey.town. The leaderboard store is a Google Firebase database, and the deployments and hosting of our services is handled on Render.

Future Work:

In the future, there are a few things that we would like to see added to our UNO Game. Most importantly would be a hard level AI opponent. This hard opponent could be trained to play the game and would be a lot smarter than the current implementations. If we had more time we would have had this in our current game.

Additionally, we would definitely enhance our leaderboard feature. Some enhancements could be to allow the user to search for themselves in the leaderboard, allow them to see their win/loss ratio, or sort by wins and losses.

Finally, we would add some graphical enhancements to the view. We could bring in actual uno card images to use for the players deck, we could add an arrow to depict the flow of turns, and could add different Icons for AI players.