CS4350 Final Project: “UNO”

“Group 410”: Kyle Seto, David Dixon, Jake Rawlings, Zane Walter

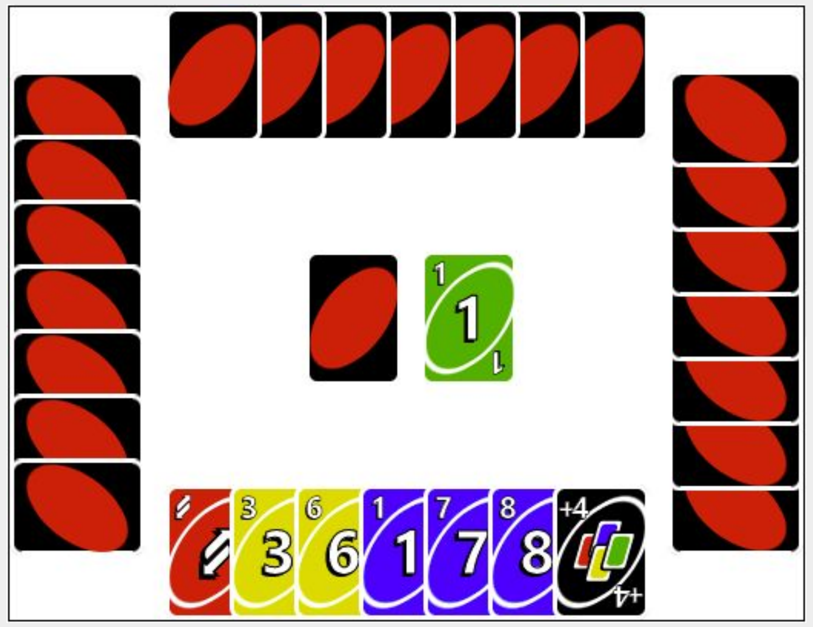
Our Feature: UNO

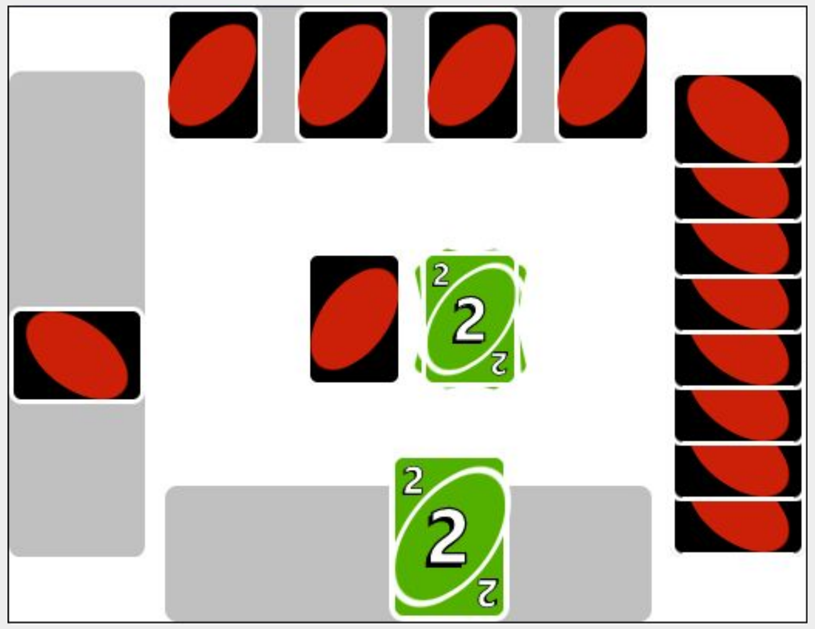
The current CoveyTown implementation has limited game areas that users can interact with and play games with each other. These games only support two players maximum at a time and lack randomization that can add to the fun of the games. It is easy to get bored of the current games after a couple of matches. We can solve this issue by implementing a new game area based on a party game that allows more than two players to participate and has an element of randomness: UNO.

Our team is interested in building an UNO game inside of CoveyTown because it would allow users to interact with more peers at the same time in a more engaging way. Historically, UNO is a game that appeals to friends and family who want to liven up a get-together or a party. A lot of emotions are usually felt during this game, whether it would be excitement from shouting “UNO!!!” as a player is about to get rid of their last card or anger that the player before them gave them a draw 4 card. In a standard game, no one knows what cards have been dealt to each player except themselves, which adds a sense of mystery to the game. All players have to pay attention to the discard pile and make decisions to get rid of their deck the quickest without unintentionally helping the other players out.

Demo and Source

Our demo site is available at https://covey-deployment-uno.onrender.com, and our code is at https://github.com/neu-cs4530/spring24-project-s24-group-410/


The UNO game when every player is ready to start.


The UNO game when a player is about to win.

Our Technology Stack and Design

We implemented our UNO game feature in the existing covey.town codebase. It was inspired by the other games that have already been implemented, including tic-tac-toe and connect four. We created an UNOGameState interface that extends GameState which is shared by the frontend and townService directories. In townService, the functionality of the UNO game and the handling of commands sent from the frontend are outlined. In frontend, the state of the UNOArea and the UNOBoard is maintained through the area controller and rendered with chakra-ui. UNOArea is represented by a small sitting area similar to other game areas in covey.town. When a player interacts with UNOArea, a GUI representing UNOBoard is displayed. UNOBoard displays the four players in the session, a graphic of the game rules that can be toggled, and the actual board that updates after a player makes a move. These components are reliant on React hooks to receive the updates from townService.

Our continuous integration pipeline runs an automated test suite on the frontend and backend services. The deployment is made possible by Twilio, Heroku, and Render.com.

Future Work

One aspect of UNO we couldn’t fully implement was the wild cards. It required a second handshake between the client and server within a player’s turn which can give the player the option to choose a color that the next person needs to match with their cards. Since the area controller was set to update after a single move, we opted to instead randomize the new color of the wild card in the backend. Future work may involve overhauling this component to better fit the rules of UNO. This implementation also does not include an UNO button for when a player places their second-to-last card. The framework is in the backend, but the frontend does not use this information. Future work may include adding this piece.

Currently, our feature is designed to mostly fit the regular rules of UNO. There can be custom rules that affect the gameplay. This may include introducing new types of cards or giving special properties to existing cards. Future work may involve adding these capabilities and toggling the rules that best fit the players’ preferences. It could also include adding implementation to increase the player size to between 5-8 players.

In the future, some implementation of giving a card description when highlighting the card could be included as well.