CS4530 FINAL PROJECT: PARTY ROOM

A Host Led Party Game Platform with Mini Games

Authors

Corina Torres, Quinn Louie, Nick Connors, and Daniel Karma

Emails

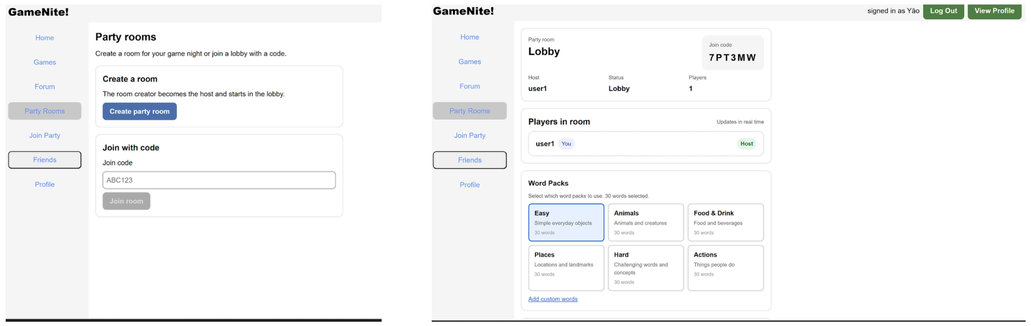
torres.c@northeastern.edu | louie.q@northeastern.edu | connors.ni@northeastern.edu | karma.d@northeastern.edu

Project Overview

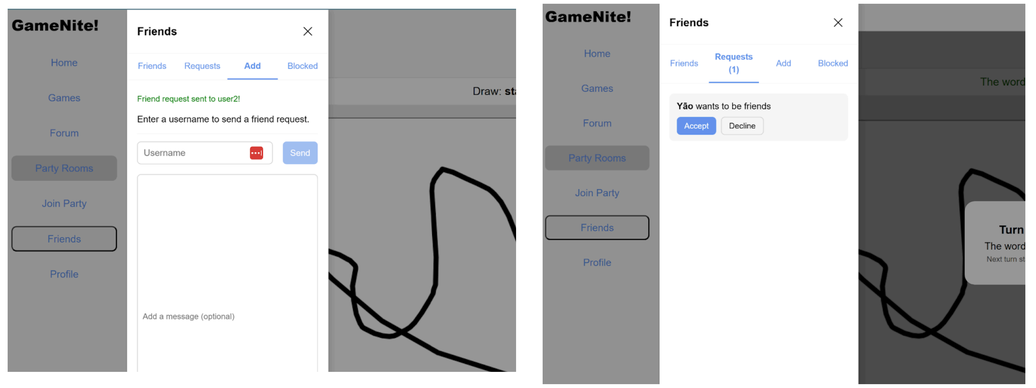
GameNite currently supports games, but it does not support a “party mode” - a room where multiple players can play multiple mini games together with a winner at the end. Without this dedicated feature, players must coordinate outside of GameNite, queue up into multiple separate game rooms, and then tally on their own who won their “party” session. It’s a much more isolated experience, with little social interaction between players. GameNite also does not have a friend system in place for players to add and play with each other.

Our project introduces a Party Room that adds a new kind of game experience: host-led, multi-round, real-time party games. A host can create a room, invite players through code or directly through GameNight as a friend, and start a session. Players join and participate in various mini games with the winner of each receiving a score, and a final winner decided at the end. Our project also introduces a small friend system, where players can add one another so that they can play games together, and invite each other directly through GameNite. We also implemented 2 new mini games.

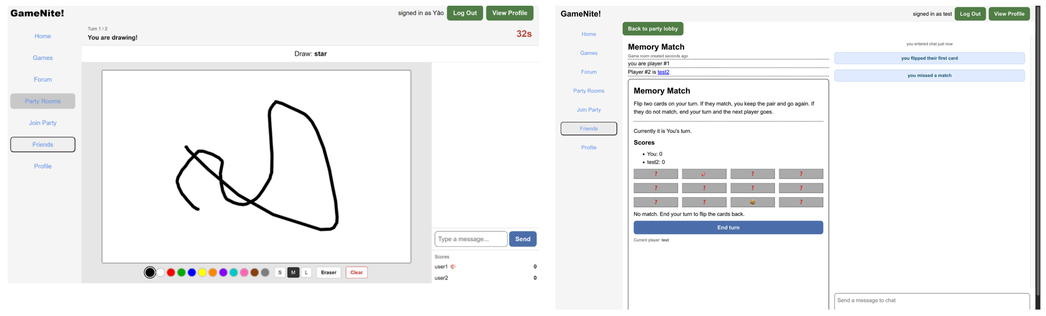
GameNite is a web application for asynchronous games and community features. Our team extended it with a party room experience: players gather in a shared lobby, vote on the next mini-game, and play in real time over WebSockets. Party rooms support multiple game types (Nim, Number guesser, Memory Match, and Skribbl), a scoreboard across rounds, and integration with the friends system (sending requests from the lobby, inviting friends into a room, and managing friends through the sidebar).



Creating a Party room, and the lobby where the host can choose word packs for Skribbl



Friend system UI - set up as a side bar over the main page. You can request friends with a message, accept, decline, and block



Skribbl and Memory Match games implemented as shown above

Technology Stack & Design

Our project extends the existing GameNite web application by introducing a real-time party room system built on a full-stack TypeScript architecture with a clear separation between client, server, and shared types. The frontend uses React with component-based UI and custom hooks for managing state and socket.IO communication, while the backend leverages Node.js with a service and repository layer to handle party room logic such as room creation, voting, and score tracking. Real-time synchronization is achieved through socket.IO events, enabling live updates for player lists, votes, and game state across all clients. A key design decision was to reuse and extend the existing game framework and shared type system to maintain consistency, while adopting a centralized server-authoritative state model to ensure reliable synchronization and simplify multi-user interactions.

Future Work

Future work could focus on expanding both functionality and system design. One direction is adding new features such as spectator mode, host transfer capabilities, or additional mini-games to further enhance the party experience. These were all features we had described in our conditions of satisfaction as extensions, so these would be “next up” to implement. Improvements could also be made to the architecture, such as refining abstractions between components and creating more modular interfaces for game integration, making it easier to add new game types without modifying core logic.

GitHub Repo and Demo Site

https://github.com/neu-cs4530/spring26-project-group-612

https://spring26-project-group-612-46ls.onrender.com/