Changes Made – Summary

Group P-2

Feedback Received:

In the Iteration 2, the following two feature suggestions were proposed by another team to enhance our game platform:

1. Customizable Board Sizes and Chess Rule Variations
2. In-game Chat Functionality
3. Blunder Count for Chess
4. Pro-League Tic Tac Toe AI

Our Implementation Decisions:

In-Game Chat - implemented

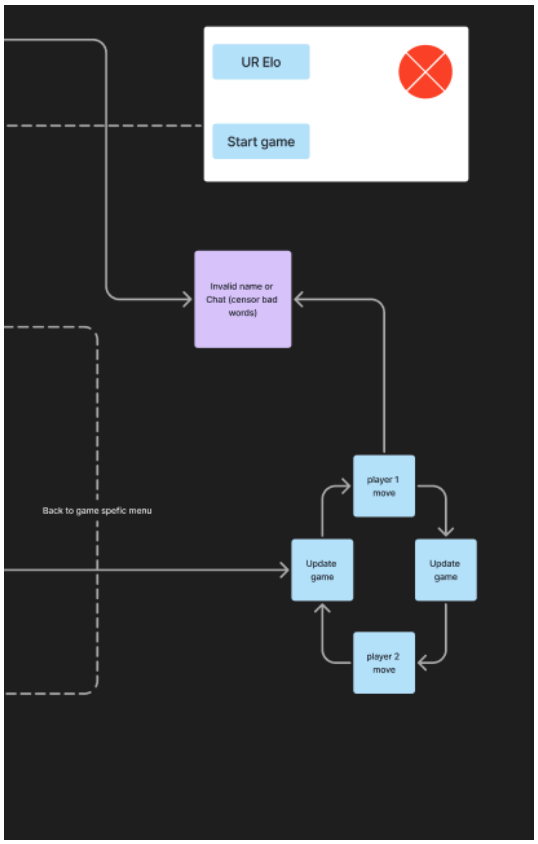
- We implemented an in-game chat system to enable real-time communication between players.

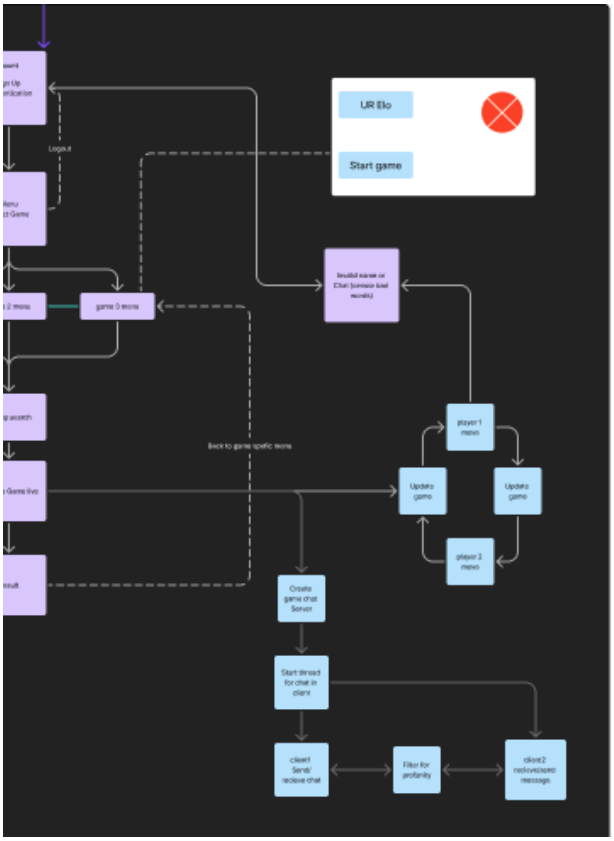
- The chat includes:

  - A collapsible chat window integrated into the game interface.

  - Real-time messaging with a text box and send button.

  - Blur any curse words





Changes made in the diagram:

1. Create game chat server

   - Initialize the server to manage chat communication between clients.

2. Start thread for chat in client

   - Each client starts a separate thread to handle chat messages.

3. client1 Send/receive chat

- Client 1 can send and receive chat messages.

4. Filter for profanity

- Messages are passed through a profanity filter before reaching the other client.

5. client2 receive/send message

- The filtered message is delivered to Client 2, who can also send messages back.
- The same filtering process applies to messages from Client 2 to Client 1.

Reasons to Implement the Feature:

1. Community building – encourages opponents to connect with one another, and friends to chat and engage in banter while they play.

2. Increased engagement - A chat helps to engage players and keeps them playing longer.

Custom Board Sizes and Rule Variations - not implemented

Suggested reasons to Implement the Feature:

1. Enhanced User Experience – Players could personalize games and increase replay value.

2. Competitive Differentiator – Few platforms offer customization, giving us a unique edge.

5. Scalability – Modular game logic now could help add new modes more easily later.

Reasons Not to Implement:

1. Increased Complexity – Custom rules and board sizes complicate game logic and win conditions.

2. Time and Scope Constraints – Not feasible to build and test thoroughly within the current project timeline.

3. Maintenance Burden – More variants mean more bugs, edge cases, and need for long-term support.

4. User Base Needs – Current users are mostly satisfied with standard rules.

Blunder Count for Chess - not implemented

Suggested reasons to Implement the Feature:

1. Low impact - is not majorly interdependent with any other part of the system, so it should be easy to implement

2. Enhanced user experience - Allows players to see their mistakes as they happen, helping them improve in the future

Reasons Not to Implement:

1. Evaluation Complexity:

   - Even simplified blunder detection requires complex logic to evaluate the board after every move.

   - Could introduce performance overhead or false positives.

2. Lack of Chess Engine:

   - Without a real chess engine, evaluating the actual "badness" of a move is unreliable.

3. Risk of Confusion:

   - Players might not agree with the system labeling a move as a blunder under simplified logic.

4. GUI Dependency:

   - The feature depends on GUI support after the match, which was out of scope for the current development phase.

5. Focus on Core Features:

   - We prioritized implementing and polishing core game mechanics over advanced analytics.

Pro-League Tic Tac Toe AI - not implemented

Suggested reasons to Implement the Feature:

1. Low impact - Is not majorly interdependent with any other part of the system, so it should be easy to implement

2. Single player - Allows players practice and play against a bot, helping them improve their play and have fun playing without the need for another human player.

Reasons Not to Implement:

1. Low Replay Value:

- Tic Tac Toe is a solved game; the AI would always force a draw or win, which quickly becomes repetitive.

2. Redundant for Simple Game:

- Tic Tac Toe already has low complexity, making a pro-level AI less essential for practice.

3. Time Constraints:

- Implementing Minimax and fully testing the AI behavior would require extra time we needed for other features.

4. Turn Management Complexity:

- Integrating AI into existing turn-based logic (e.g., fairness, timing, animations) would take additional effort.

Summary:

From the suggested features of: game chat, custom board sizes and rules, chess blunder count, and tic tac toe AI, we only decided to implement the in game chat. We felt that the in game chat would significantly enhance our user experience without much additional work, given that we were already planning to implement networking. We loved the idea of custom board sizes and game rules, as we felt it would add lots of replayability and personality to the games, but we decided it was not feasible while attempting to implement features like networking, a database, and an extra game. Similarly with tic tac toe AI, we were already stretched too thin to implement this. Finally, we decided not to implement a blunder counter for chess as we felt it wouldn't add much to the user experience, while adding unnecessary complexity to our chess systems.