

Feature Proposal

Features

Feature 1: [Game Difficulty Adjustment] **High-**

Level Description:

Cloud saving for user settings, game progress, and statistics, allowing seamless transitions between different devices.

Expected Impact:

Ensures that players can access their progress from any device without losing data, improving convenience and continuity in gameplay.

Suggestions for Implementation/Design:

- Utilize related subsystems from the class structure diagrams, specifically `updateProfile (playerID: string, stats: Stats): void`, to handle user progress syncing.
- Implement cloud storage solutions or a dedicated game server for storing and retrieving player data

Feature 2: [Actions To Player]

High-level Description: Provide a set of actions to perform onto a player for the user's convenience. For example, sending a friend request or blocking them from interacting with you.

Expected Impact: It will give the user more control over their game.

Controlling who they play with will certainly improve player experience, by giving a straightforward way to play with people you enjoyed playing with and prevent playing with people that disrupted your enjoyment.

Suggestions for Implementation/Design: Within the profile of a given player (based on the mockup, the top-right would be a suitable spot), offer a dropdown menu where the user can choose to send a friend request, or to block that specific player.

Feature 3: [Recover Account Credentials]

High-level Description: If a player forgets their username or password, there should be a mechanism by which they can recover their account, without having to make an entirely new one.

Expected Impact: It makes it easier to recover their account in case they forget their username/password. Users won't need to create a new account.

Suggestions for Implementation/Design: On the login page, they can include a small prompt or feature that allows users to recover the password or username in case they forget. This could be similar in size or font to the prompt/deture that allows the user to sign up for an account; just placed beneath the Login button.

Feature 4: [Viewing Options for Visually Impaired Users]

High-level Description: If users are colourblind, or otherwise visually impaired, there could be accommodations made in their interest that make the UI fully readable despite their impairments.

Expected Impact: The reputation of the game will be greatly enhanced by its inclusivity towards people with physical impairments, and it will also increase the client base for the game if implemented properly. Accessibility features will also imply a higher level of intelligence from the developers by showing that they considered very intimate aspects of the people playing their game.

Suggestions for Implementation/Design: This must not be implemented in a way that is blatantly pandering towards minorities. It should be in a separate tab in the 'Settings' menu, under 'Accessibility' or something similar. It should also be easy to navigate to if someone is seeking it out.

Improvements

Improvement 1: [Banning for Disruptive Persons]

Description: If someone repeatedly shows a pattern of behaviour wherein, they are bullying, harassing, or degrading other users on a regular basis, they ought to be reprimanded and held accountable for their actions via ban from online interactions.

Expected Impact: This will create a safer, friendlier atmosphere for all players. If people are being insulted and made to feel lesser than by those lobbying with them when they queue up for a game, then they will not want to play the game very much. Implementing this change will ensure that does not occur.

Suggestions for Implementation/Design: Adding a 'Report' button next to people's profiles would allow developers to scrutinize specific

individuals for problematic behaviour. There should ideally be a log kept in non-volatile format of the player's recent interactions with others, where bans can be initiated according to a standard protocol decided upon by the developer team as a whole.

Risks & Mitigations: What counts as 'problematic behaviour' can be incredibly subjective, so much research and debate must occur in order to implement this in a way that does not wrongfully cause people to be banned via technicality or such and such. As well, banning needs to be properly enforced, not just a slap on the wrists. There need to be consequences for behaving in ways that violate the online feature's TOS.

Improvement 2: [Cross-Platform Compatibility] Description:

There is no current cross-play functionality, meaning people playing on the web, mobile or desktop cannot interact. The support of cross-play would allow for more seamless integration of users with different devices, improving accessibility and expanding the player base.

Expected Impact:

- Allows cross-platform connections enabling a higher number of players to join which improves matchmaking efficiency.
- Solves the user experience problem by allowing friends to play together irrespective of the device they own.
- Supports multiple devices for longer platform life and increased competitiveness.

Suggestions for Implementation/Design:

- Use a common communication system like WebSockets or APIs so different platforms can connect smoothly.

- Make sure players can log in with the same account on any device and keep their progress.
- Adjust controls and screen layouts so the game works well on phones, tablets, and computers.

Risks & Mitigations:

- Risk: Performance variations across platforms could lead to an unfair advantage for certain players.
- Mitigation: Implement platform-specific balancing adjustments to ensure fair play.

Improvement 3: [Reevaluating Access Modifiers in Networking Class Diagram]

Description: There are certain methods contained in the 'Networking' class structure that should not be made public to the rest of the project, as they harm modularity and possibly overcomplicate the logic flow:

```
+ establishConnection() +  
  distributeRequests() +  
  monitorTraffic()  
+ startServer(port: int)  
+ logError(error: String)
```

Expected Impact: It is paramount to get the access modifiers correct for this class structure so that there are no circular dependencies that occur throughout the code. As well, implementing this change will eliminate an egregious security concern, in that laypersons will not be able to run false errors to the server, monitor network traffic, or track requests made throughout the server.

Suggestions for Implementation/Design: Should the aforementioned methods be made private instead, the risk of security breaches will go

down substantially due to limited access from malicious persons, the logic flow for the class structure will be easier to follow due to its more linear nature, and future developers will have a much easier time making changes to the code without completely shattering the integrity of the structure.

Risks & Mitigations: If people are in need of the information that each of the newly 'privated' methods contain, there should be getters and setters put in place so that the information can be accessed. However, this should not be done in a way that allows the private information to be transformed or otherwise modified – it should only be done according to what is necessary for developers to do their job, or users to send networking failure reports to the development team.

Improvement 4: [Limited and Vague Player Interactions]

Description: The options that are given for users to interact with other players seem vague/limited. For example, on the `player_lookup.png`, the button intended for user interactions does not clearly indicate its function. It lacks a clear label or purpose, which can make it unclear for how users are supposed to respond or interact. This lack of clarity can reduce the systems overall useability; making interactions feel unintuitive and adds unnecessary friction. Implementing a button that is much clearer can reduce confusion and create a system that is much more streamline and easier to use.

Expected Impact: Incorporating more intuitive and clearer buttons/ player interactions can make the system feel more streamlined and easier to use. Having more clarity with these user interactions and buttons gives the user a smoother experience, which will bring better engagement and a more intuitive ui. A clearer system reduces friction, improves usability and encourages player retention.

Suggestions for Implementation/Design: For example, on player_lookup.png, the button beside the player profile seems very vague on its purpose. Incorporating 3 rectangular buttons stacked on top of each other with explicit labels seems much clearer and more purposeful. Ex:

Feature 1
Feature 2
Feature 3

Risks & Mitigations: One possible problem that could arise is having the UI seem more cluttered. The risk that could come with making buttons and interaction more explicit is that it could compromise the simplicity of the UI. Incorporating more labels and explanations could pose the risk of making the UI more cluttered and overwhelming for users.
