Polybius

Team 9 - Product Backlog

Lucas Hahn - Leo Liu - Chris Sculley - Thomas Simons

Problem Statement

The modern video game community has had a tendency to not always interact with others face-to-face, due to the solitary nature of modern video games. Even though many multiplayer games exist, the vast majority of these games lack any personal communication and connection beyond text or audio. Our project enables a unique gaming experience, blending real-world interaction with imaginary gameplay in way seldom seen before. By not just creating static virtual objects, but interactable and multiplayer-focused scenes tied to real world locations, we can inject the power of augmented reality and 3D gaming into real world interactions to facilitate genuine connection between individuals.

Background Information

Audience

We are living in a society where networking and socialization are necessary skills. Most games allow players to socialize with each other in a virtual world, behind a digital screen. However, the lack of physical, face-to-face interaction hinders the ability to create meaningful relationships. Our target audience is gamers who struggle to have these “real” interactions. By creating high-quality, captivating games, we can utilize gamers’ interests to help target and nurture their real-life, social ability by creating virtual games with face-to-face interaction as a central component.

Similar Platforms

One significant game that aims to bring players together physically is Pokémon Go. Pokémon Go allows players to walk around and catch Pokémon, and in the process they may meet other players. Another game that also tries to achieve this is Ingress. Ingress uses GPS nodes to bring players to a specific location where they must complete a certain objective. Pokémon Go and Ingress both implement augmented reality in their gaming experience.

Limitations

Other solutions require the user to interact either in the real world and disregard their devices, or interact with their devices and disregard interaction in the real world. In Pokemon Go, users will commonly stand around and all focus on their devices during gameplay, often without interacting with each other until the current game is finished. While some platforms, such as Ingress, try to quickly flip between these modes of interaction in order to create compelling social gameplay, our unique augmented reality gameplay will achieve a smooth blend of real world interaction and virtual gameplay, creating a social experience like no other.

Requirements (Backlog)

Functional Requirements

1. As a user, I want to be able to create an account so that I can have a personalized experience
2. As a user, I want to add other players as a friends so that I can build established connections
3. As a user, I want to have a profile that others can see so that others can connect with me
4. As a user, I want to see many statistics about the games I've played so that I can track my performance
5. As a user, I want to be able to search for other users so that I can connect with others
6. As a user, I want a messaging system to message my friends so that we can continue to communicate outside of matches
7. As a user, I want to be able to find games around me so that I can easily join new games using a map
8. As a user, I want to host games to play with nearby players so that I can meet new people
9. As a user, I want bonuses for meeting up with new people so that I will be encouraged to meet others
10. As a user, I want to send challenges to players to beat my high score so that I can challenge others
11. As a user, I want a easy-to-use interface that feels satisfying to use so that I will enjoy using the application
12. As a gamer, I want to keep track of my highscores so that I can see my progress and share my achievements
13. As a gamer, I want to see the top players of any minigame
14. As a gamer, I want to play this game multiple times so that I can meet new people
15. As a gamer, I want to play Connect 4 using augmented reality against other players
16. As a gamer, I want to play Pong using augmented reality against other players
17. As a gamer, I want to play paper toss using augmented reality against other players
18. As a gamer, I want to play tic tac toe using augmented reality against other players
19. As a gamer, I want to spectate a game that is already in progress so that I can participate in the experience
20. As a gamer, I want to be able to create a tournament so that I can encourage more collaboration than normal
21. As a gamer, I want to be able to play in a tournament so that I can compete in larger and rarer events than normal
22. As a user, I want to be able to join a community for the game so that I can feel included
23. As a user, I want to be able to give feedback on the game so that it will be heard and considered
24. As a user, I want to be able to report other players if I feel they exhibit inappropriate behavior so that I can avoid interacting with them

25. As a user, I want to be able to find specific minigames so that I can play the games I like the most

Non-Functional Requirements

Architecture and Performance
We will use Unity3D as the engine to build and render our minigames in real-time. We will model our 3D assets in Blender, or source them from the Unity Asset Store. In addition, we will use the Vuforia API to track custom-generated QR codes and render our game onto it. The QR codes will be generated with a specific ID tied to each player, so that each game instance can only be played when the mobile camera is tracking a unique QR code.

Our web server will be built with Python and Flask. The web server will handle the game locations, gameplay networking, and score information, and implement simple HTTP authentication. Using SQLite, we will host a database to store user game statistics, messages, account information, and current game locations. The database can then store and send data when requested by the users. We will also use SmartFox as our networking platform for the minigames. SmartFox will handle Unity-specific networking, while the python server will handle any generic information.

Usability and Interface
The app itself will feature a user interface that allows players to check their statistics, highscores, message friends, and create or join games and tournaments. When joining or creating a game, a specific Unity scene for that game will open up. The messaging service will allow players to chat with each other to coordinate meetups. The interface of the minigames will be user-friendly and simple to use. We will opt to create a user interface for each game instead of using motion controls.

Networking
We will connect our app to a webserver, and the connection will persist while the app is open. Users will have to authenticate with an account in order to use the application. High scores and in-game statistics will be stored and managed in the database on our web server. Users can chat with each other via the server as well. In order to play a game with someone else, players will join a lobby for the game that they want to play, and then they will join a specific room to play against the player of their choice. The players must be close enough to each other for the game to be played, and we will ensure this by making both players have to scan the same QR code.

Hosting/Development

The python server will be hosted on Dreamhost under a team member’s domain name. The SmartFox server is self-hosted, and development will occur through Github’s “Git for Unity” plugin, which utilizes file-locking and git-lfs, among other benefits.