VR Dueling (multiplayer)
Robin Schmit (schmirob)

1 ELEVATOR PITCH

I'm interested in the challenge of syncing real-time information from two or more headsets in order to play games with multiple human participants. As a specific test case, I'm thinking of implementing a laser tag/dodgeball sort of thing, which satisfies the criterion of being extremely simple to implement and not requiring real-world collisions in the way that something like sword-fighting would. Players would be in [probably] separate areas, but virtualized into relative proximity.

2 EXTENDED OVERVIEW

Rec Room, a VR hangout that includes minigames such as dodgeball and paintball, is one of the most popular social VR apps on the market today. This makes sense to me, since one of the most satisfying ways to play games is with other humans. I want to build a multiplayer experience that allows gameplay not possible in the real world.

My original thought was to implement a fencing simulator in VR. Unfortunately, it is very difficult (possibly impossible) to allow a VR player to handle an object that would want collision physics, such as a sword. This would require the controller to be able to exert a force on the hand moving it. This is why most people, including me, choose to implement VR games in which player-held objects either slice through things or fire projectiles.

2.1 Technical Challenges

- The primary challenge is going to be the network stack aspect of the project. In order to have satisfying gameplay, the display for each player needs to be more or less synchronized, at least assuming similar network speeds for both players. This may not be a significant challenge, if I run the game on a single machine and project information to both headsets.
- Constructing the game itself (in Unity) will be a challenge in some sense, since I haven't worked much with Unity before. However, functionality will be able to be checked even if graphics are poor.

2.2 Key Risks and Mitigations

- The risk of inexperience in Unity can be mitigated by not attempting to do complicated things in it. Shooting a projectile in a straight line out of a controller should not be terribly complex.
- The most worrisome part of the project is running two different VR displays simultaneously, probably from the same device. This will be both computationally difficult and require a layer of networking code, which I assume I can find in a library but also might be able to write myself. If this doesn't pan out for some reason, I can always just add hostile artificial enemies and build a mediocre wave shooter/bullet hell.

3 HARDWARE AND SOFTWARE

Hardware

[Required] Two VR headsets. Wireless would probably be a large plus, if such headsets don't rely on a display extension abstraction the way our homemade devices do.

[Personal] Laptop with graphics card. My laptop can handle one VR experience relatively well. I believe it will be able to handle two.

Robin Schmit, schmirob@cs.washington.edu

Software

[Required] I don’t believe I’ll need money to get a network library, but I might.
[Personal] Unity is free software.

4 TEAM RESPONSIBILITIES

This is a solo project

5 DEVELOPMENT PLAN

- 3/2/19: Get the one-player game totally functional, talking to a separated server.
- 3/10/19: Add a second player and handle the complications therein (delay, everything else).
- 3/19/19: Final report and demo ready.