INTRODUCTION

Tron VR is a virtual reality adaptation of the classic Tron arcade game. The game features dynamically generated light trails behind players as they move, collisions and an AI opponent. There is also a multiplayer version in development that allows different users to play over a network on mobile phones or personal computers. With multiple speed settings, 3D outer space graphics and the ease of use from Google cardboard, this is a fun and exciting new way to play a competitive action game with friends.

OBJECTIVES

• Game Components:
  ✓ Create a player object with fast turning capabilities, controlled by gaze-and-click input
  ✓ Generate dynamically-built light trails behind players that has low-latency and triggers collisions upon contact
  ✓ Create clean singleplayer mode with a smart AI player, crisp graphics, sound effects that requires only the Google cardboard headset to play
  ✓ Build framework for multiplayer version that allows players to simultaneously compete in same playing space from different devices
• Create a compelling version of a classic game that has been remade many times
  ✓ Perfect earlier models of 3D/VR Tron games by implementing more robust multiplayer capabilities, better graphics (e.g. terrain, background, etc.), and larger/infinite field of play

CONCLUSIONS

Overall, creating this game was a highly enjoyable process of trial and error using Unity and the Google cardboard API. These systems provide good frameworks for incorporating Physics concepts into games, such as force and velocity, as well as a broad range of pre-existing game objects and assets that facilitate the creative process for first time game designers. However, some of the key components we hoped to include in our game have limited support in Unity’s virtual reality framework, including minimaps/custom on-screen messages, setting up a custom multiplayer network for mobile devices, and general bug support. We found that while often helpful for creating standard game features, the built-in objects in the Unity editor were helpful, but became clunky when they needed to be used in conjunction with a lot of custom scripting.

REFERENCES

• Proton Pulse: https://youtu.be/ZnL6Hyxw5yA?t=112
• Oculus Tron: Light Cycles: https://share.oculus.com/app/tron-light-cycles
• Strider Van tutorials: https://www.youtube.com/user/Stridervan
• 3dbuzz.com tutorials: https://www.3dbuzz.com
• Unity3d.com tutorials and API: http://unity3d.com

Acknowledgements

This project was completed as part of CS 234/334 Mobile Computing (Winter 2016), taught by Prof. Andrew A Chien, with TA support by Yun Li and Yan Liu.