A Cross-Platform Virtual Reality Experience

by Itamar David Belson
S.B., Electrical Engineering and Computer Science, M.I.T., 2016
S.B., Comparative Media Studies, M.I.T., 2016

Submitted to the
Department of Electrical Engineering and Computer Science
In Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTUTE OF TECHNOLOGY

December 2016 [February 2017]

Signature redacted

Author: ________________________________
Department of Electrical Engineering and Computer Science
December 16, 2016

Certified by: ____ Signature redacted ________
Chris Schmandt, Director, Living Mobile Group, M.I.T. Media Lab
Thesis Supervisor
December 16, 2016

Accepted by: ____ Signature redacted ________
Christopher Terman
Chairman, Masters of Engineering Thesis Committee

A Cross-Platform Virtual Reality Experience

by Itamar David Belson
S.B., Electrical Engineering and Computer Science, M.I.T., 2016
S.B., Comparative Media Studies, M.I.T., 2016

Submitted to the Department of Electrical Engineering and Computer Science
December 16, 2016
In Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Virtual reality refers to a realistic and interactive experience whereby a user is able to observe and interact with a simulated three-dimensional environment. However, although immersive by nature, most modern virtual reality systems require the use of specialized head-mounted displays that result in experiences that are wholly detached and isolated from other potential users as well as from the primary user's immediate surrounding. This thesis describes a system that aims to solve the isolative nature of virtual reality through incorporating a new form of multi-person interaction within virtual reality worlds that enables cross-platform observation and agency within the digital domain. Namely, this new form of human-to-human virtual interaction method allows two or more individuals to simultaneously observe and interact with the same virtual world through inherently different perspectives. More specifically, this thesis details an example application of such a system in which two users, one connected through a virtual reality interface and the other connected via a standard monitor interface, concurrently play a virtual air hockey game similar to the physical equivalent. Although a single example, the principles of the described system may be extended and applied to a variety of applications within the virtual reality domain.

Thesis Supervisor: Chris Schmandt
Title: Director, Living Mobile Group, M.I.T. Media Lab

Acknowledgments

I would like to express gratitude to my advisor, Chris Schmandt, for giving me the opportunity to join the group and for giving me the chance to explore my interests through this exciting project. I would also like to thank the Living Mobile Group within the Media Lab for providing me with a great working environment full of diverse interests and curiosities. Finally, I would like to thank my parents, my family, and my friends for their guidance and encouragement, not only in this project, but in all of my pursuits.

Contents

List of Figures

1 Introduction

1.1 Background

Although the concept of virtual reality has been around since the early 1900s, only recently have technological advances seen this once imagined figment of science fiction into a real-world actuality with an emerging hotbed of potential functions. From games to business, military to medical applications, virtual reality has begun to infiltrate a multitude of fields and markets, providing an immersive and versatile platform that is both adaptable and applicable to various applications [1].

With the recent emergence of hardware products such as the Oculus Rift, the Samsung Gear VR, and the HTC Vive, virtual reality has transformed from a previously experimental proof-of-concept into a democratized consumer product. In addition, due to relatively recent advances in computation and graphics alongside the introduction of platforms that enable the development of experiences in virtual reality, such as Unity and Unreal Engine, virtual reality has become a democratized medium, allowing everyday researchers and developers to seek previously unexplored use cases.

1.2 Previous Work

The effect of translating human interactions into the digital domain has been an ongoing development. Yet, up until recently, single-person interactive experiences remained the primary focus of research and development within virtual reality. However, as single-person interactive experiences have become a proven application, there has begun to emerge preliminary ideation and work to explore the potential applications of the virtual reality medium to multi-person experiences, a natural extension to existing experiences that has thus far proven successful.

From online games to virtual conference rooms, researchers and developers alike have begun to actively investigate the manners by which virtual reality could incorporate multi-person digital experiences through online human interaction. Such technological advances would enable multiple individuals to co-exist in a common virtual space whereby barriers to human-to-human interaction, such as physical distance, may have otherwise prevented it. Papers such as *MetaSpace II: Object and Full-Body Tracking for Interaction and Navigation in Social VR* detail early prototypes of such multi-person virtual reality systems while providing a wealth of information about the process, successes, and failures of rendering such interactions within the virtual reality domain [2]. In addition, developers have begun to explore the potential of multiplayer virtual reality games and experiences in such early titles as *Hover Junkers* and *Pool Nation VR*. These games allow multiple users to connect and play collaboratively or competitively much like on any other platform.

1.3 Motivation

While there has been ongoing work in the field of multi-person virtual reality experiences, the majority of work has been focused on developing experiences that require all users to be equally immersed in the virtual world; most existing work in the field assumes the virtual reality medium to be the singular interface used by all of the system's users, much like prior video game consoles or modern chat rooms. For example, the multi-person virtual reality experience in existing virtual reality games assumes that every user is connected to the game via fully immersive virtual reality head-mounted display. Similarly, for full function, multi-person virtual reality conferencing systems require every user to be connected to and observing the scene through a comprehensive virtual reality interface.

However, what if a particular user does not have access to a complete virtual reality setup or is unable to tolerate the common side effects of virtual reality use, such as nausea or dizziness [3]? Existing multi-person virtual reality applications overlook such issues and assume any user not connected via virtual reality to be outside the digital experience. This drawback is similar to the issue that can stem from video game developers often requiring users of a multi-player game to be connected through the same console, which effectively prevents those users who wish to connect via a different platform from partaking in the experience.

This thesis describes a system that aims to solve this existing console issue of virtual reality by introducing a new form of multi-person interaction. This new form of interaction enables users to connect to a virtual reality experience through different platforms to allow them to partake in the experience without loss of agency within the digital domain. Along with enabling users to connect to the experience through a preferred interface, namely a virtual reality interface or a standard monitor interface, the system also takes each platform into account by adjusting the controls and perspective methods to the mode of observation to ensure that no user loses agency within the experience. As such, this system enables a segment of the user population to remain fully immersed in the virtual reality world while allowing other users to fully access the same experience through alternate, exterior modes of perspective.

2 System Architecture

2.1 General Overview

To demonstrate an application of a cross-platform virtual reality experience, this section details the implementation of such a system in which two users, one connected through a virtual reality interface and the other connected via a standard monitor interface, concurrently play a digital air hockey game similar to the physical equivalent. Through enabling each of the two users to select a preferred interface through which to connect to the game, the system effectively provides the user with the opportunity to elect a perspective method, whether dependent on preference or available equipment, while aiming to minimize loss of agency with the overall experience.

2.1.1 Gameplay

The system follows the general structure and gameplay of popular digital air hockey games, which are modeled after the physical equivalent. The game involves two opposing players constrained to opposite sides of an elongated table. Each player is provided a handheld pad that is used to guide and strike a thin hovering puck across the table toward the opposing player's goal. Upon scoring a goal, the puck is placed on the opponent's side and the gameplay resumes. The game continues until either player scores a predetermined number of goals.

Although the motions of the pad are free ranging, each player's pad must remain on the table at all times and must stay on the player's respective side without crossing the center divide. In addition, the player must physically remain on the side of the table pertaining to the goal that he or she is actively defending.

While common implementations of air hockey, digital and physical alike, require both users to access the game via the same interface, the implementation detailed in this section enables each user to indicate a preferred interface. Namely, upon connecting to the game, each user is able to select between a virtual reality interface and a standard monitor interface. After both users have selected a preference and are connected to the network, the described gameplay commences.

2.1.2 The Virtual Reality Interface

The first perspective method consists of a virtual reality interface. This interface method is designed to be completely immersive and interactive, similar to other virtual reality experiences. Through a head-mounted display and a handheld controller, the user is placed inside the virtual world and is able to interact with it directly. In order to connect to the game implemented in this thesis using the virtual reality interface, the user must have access to a HTC Vive and to a machine capable of supporting the rendering requirements of the HTC Vive [4].

The gameplay controls associated with the virtual reality player are modeled to feel natural, following closely to the controls of physical air hockey. The player's virtual pad is moved across the digital air hockey table through moving the handheld controller, whose position is actively tracked in real-time through a pair of signal modems that are placed throughout the room. To make the user feel even more immersed in the virtual reality experience, a physical air hockey pad is attached to the head of the handheld tracking device, giving a tactile feeling similar to the real world experience as can be seen in Figure 1. In addition, the user is able to look around the virtual environment through the HTC Vive's head-mounted display, with the user's rotational and translational motion projected into the display in real-time to give the a sense of physicality within the virtual space. In a sense, a player connecting to the game through this perspective method should feel as if playing against another user connected through the same virtual reality interface.

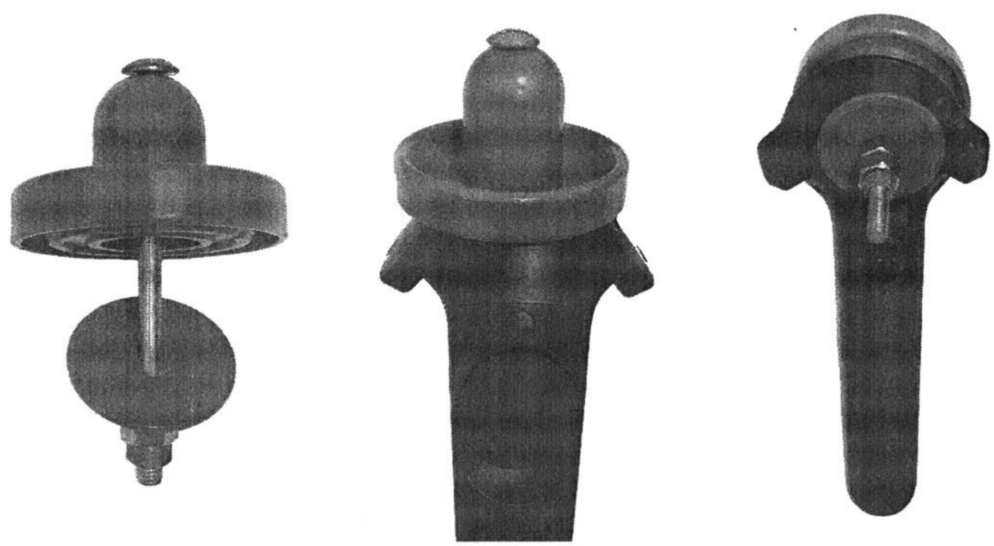

Figure 1: The handheld tracking device with tactile air hockey pad attachment.

2.1.3 The Standard Monitor Interface

The second perspective method consists of a standard monitor interface. This interface method is designed to be intuitive and reactive while still constraining the user to the limitations of a two-dimensional screen. In this case, the user is able to perceive the virtual world from a predetermined perspective, interacting with it directly through controls that are conducive to the platform. Through this interface, any user with access to a network-connected machine is able to connect to and partake in the experience without the need for virtual reality equipment.

The gameplay controls associated with the standard monitor interface are modeled to feel natural to the platform, modeled closely after similar digital air hockey games. The player's digital pad is moved across the virtual air hockey table through simple mouse movements. The digital pad follows closely to the mouse, tracking in real-time. In a sense, a player connecting to the game through this perspective method should feel as if playing against another user connected through the same standard monitor interface.

2.2 Implementation Details

This section provides implementation details of the system, highlighting particular approaches that may prove relevant when developing similar multi-person, cross-platform experiences.

2.2.1 Implementation Overview

While several development platforms exist, the application outlined in this section is implemented in Unity 3D. Unity 3D is a development platform primarily used for rapid game design and development [5]. With an easy to use interface, an interactive physics solver, C# scripting capabilities, and script-extendable prefabs, the platform is a common choice among those looking for solutions to develop fully functional games and applications. In addition to Unity 3D, this implementation utilizes the

SteamVR plugin, which provides connection protocols to the HTC Vive system [6]. The system utilizes the prefab architecture of Unity 3D, which closely resembles the object oriented programming structure common to other programming languages. There are three different implemented prefabs in the system.

The first two prefabs model the behavior of the pad of each of the two different perspective methods. Each of the two pads requires a separate prefab to allow for implementation of the different control mechanisms and functions that are inherent to the particular perspective method. Among these controls are the methods by which the pad can be moved and the particular limitations of the said movements. The virtual reality pad prefab continually tracks the user's handheld controller, projecting its relative position onto the virtual air hockey table to determine the current position of its pad within the virtual world. The standard monitor pad prefab continually tracks the user's mouse position, projecting its relative position onto the digital air hockey table to determine the current position of its pad within the digital world. Such controls are designed in close consideration of the particular perspective method to provide each user with complete agency over the experience regardless of perspective.

The third prefab models the behavior of the air hockey puck. In addition to the particular actions of the puck, this prefab also describes central gameplay logic. The puck prefab determines when a goal is scored and updates the score while resetting the game's state. In addition, the puck prefab handles key networking solutions for game state synchronization between the two users, described in greater detail in the next section.

Aside from the player pads and the puck logic, the remainder of the game is implemented using standard Unity 3D functions. For example, while most of the gameplay logic is handled by the puck prefab, the physical behavior of the puck, such as its realistic motion and its collisions with the pads and table, is conveniently handled by Unity 3D's built-in physics solver. In addition, the gameplay environment, consisting of the room and game models, are implemented through assets from the Unity 3D Asset Store.

2.2.2 Networking Considerations

After the development of the prefabs and the basic gameplay logic, a single user can play the game through a preferred perspective method. In order to allow for multiple people to concurrently play the game, as intended, the underlying game state must be continually synchronized between the two clients through the network. This section covers the networking approaches, considerations, and implementation details specific to this application.

The application utilizes a host-client architecture in which the first player hosts a game server to which the second player connects. Upon launching the game, each player must select a perspective method as well as whether to host or join an existing game as a client as can be seen in the GUI in Figure 2. Clients are connected to the host through a specified IP Address associated with the host's machine. In actuality, upon creating a game, the host also spawns a client session that directly connects to its own local IP address.

While the host-client architecture is common in many networking applications, a result of this architecture and most similar networking architectures is that data transport times can vary across clients. More specifically, the host user will have direct access to the updated game state data without having to deal with the network delays experienced by the client. What results are game state delays that are non-uniform across the clients, which can greatly affect the overall gameplay of real-time applications. Aside from the effects on user experience, network delays may result in more serious complications when dealing with game state synchronization of physics-simulated objects such as the puck in this application. As early implementation attempts revealed, unfitting networking implementations can cause minute state inconsistencies that result in wildly different observable game states between users due to the unsynchronized physics-affected components.

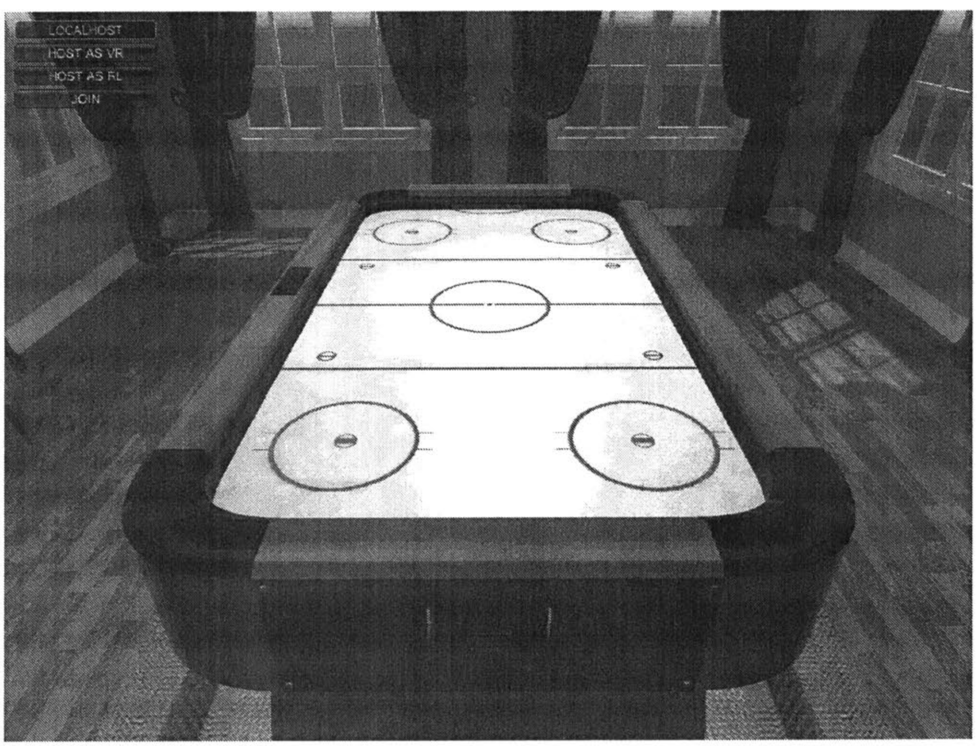

Figure 2: The GUI for selecting preferred connection and perspective methods.

2.2.3 Networking Approach

To combat the issues described in the previous section, this application utilizes two different networking protocol schemas to tackle the task of real-time game state synchronization. While both protocol schemas are based on simple UDP protocols, each one solves a different task with distinct considerations of the underlying game state synchronization process.

The first protocol schema uses Unity 3D's real-time transport layer to synchronize the position of the opposing user's pad. This protocol schema is only concerned with transporting two-dimensional translational data between users across the network; each client sends its pad's positional data to the server, which in turn sends the data to the opposing client. Through the transferred data, each client is able to position the opposing client's pad in near real-time. Although UDP may be unreliable in that it may drop packets across the network, the short time interval between positional updates makes it so that the dropped packages are virtually unobservable to the user.

The second protocol schema builds upon the first to solve the networking problems associated with synchronizing the physics-based position of the puck. While each client has independent instances of the two pads, which are continually updated according to data received from the server, the two clients must share a single instance of the puck to prevent minute positional differences stemming from network delays from resulting in positional inconsistencies calculated by the physics solver. As described in the previous section, if each client were to have an independent puck instance and the game state positions were slightly offset between the two clients due to the inherent delay of the network, the physics solver associated with each player may result in wildly different puck locations between the clients. What results is that each player witnesses the puck at a different position, resulting in a wholly unsynchronized game state.

To combat this issue, the second protocol assumes there to be a single puck instance and physics solver between the two clients. As opposed to the pads, whose positions are determined by their respective client, the ownership of the puck instance and its associated physics solver are passed between the two clients according to the puck's current position. For example, if the puck is currently on Player 1's half of the table, then Player 1's client owns the puck instance and its associated physics solver. However, once the puck passes entirely across the center divider, the ownership over the puck instance and its associated physics solver is passed to the opposing player's client through the server. Whichever client is the owner of the puck instance is responsible for continually updating the server and the opposing client of the puck's current position. This is done using a UDP protocol similar to the first one described in this section. Through this approach of ownership transfer, the physics solver is able to apply determined positions of all the objects that may interact with the puck at a particular moment in time without having to deal with positional discrepancies brought upon by network delay. In essence, this method assumes that physical possession over the puck implies complete possession over the puck instance and its associated physics solver to make for a successful user experience.

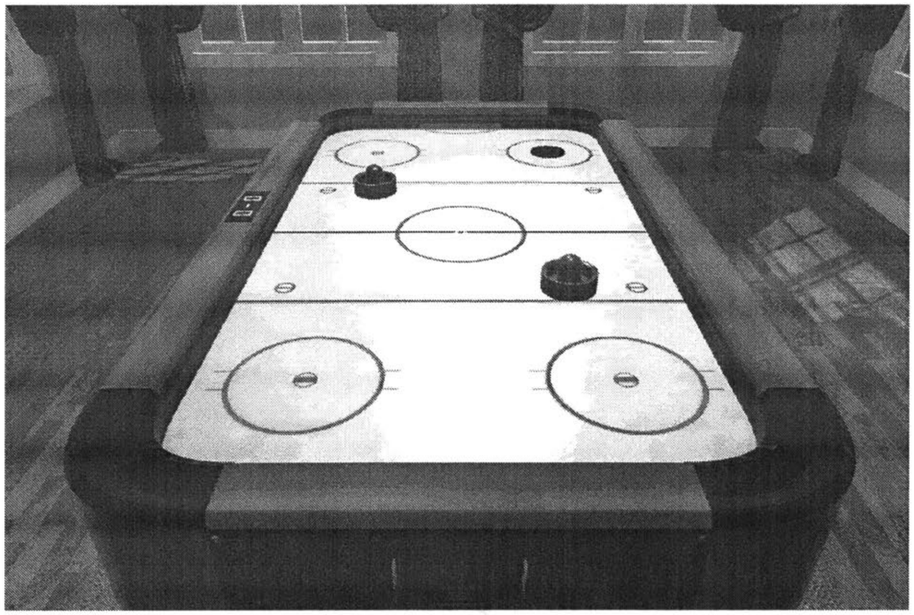

Figure 3: Gameplay perspective as seen from the standard monitor interface.

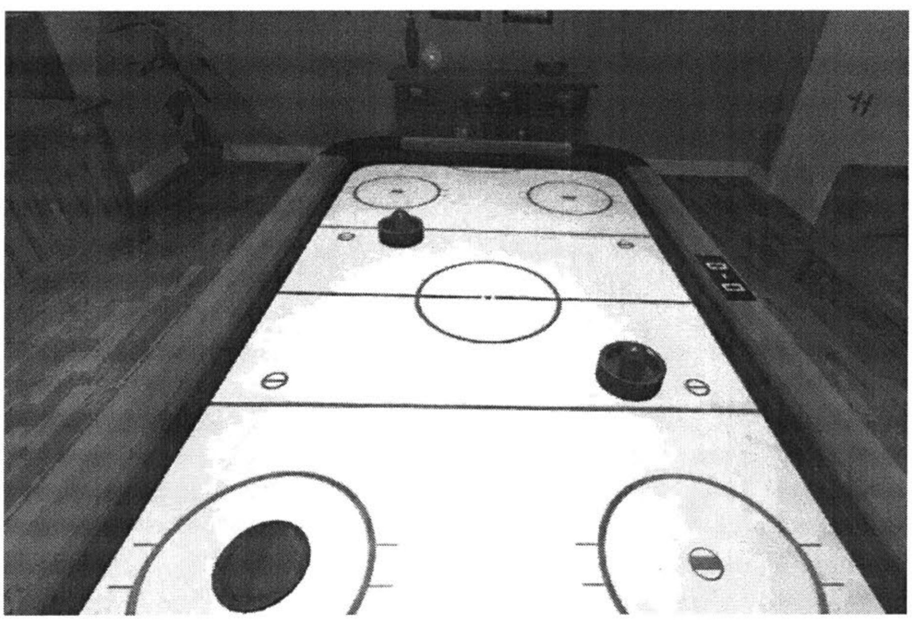

Figure 4: Gameplay perspective as seen from the virtual reality interface.

3 System Evaluation and Conclusions

3.1 System Evaluation

Following the implementation, the system was user tested to determine its success in achieving the underlying goals. During testing, the system proved successful in allowing each user to select a preferred perspective method. After selecting between a virtual reality interface or a standard monitor interface, the system connected both users to the same game, and gameplay successfully commenced. The controls of the pads seemed to be intuitive and responsive in both perspective modes, allowing users to switch between interfaces with ease. Furthermore, the motion tracking of the pad seemed to be effective in the case of the handheld controllers in the virtual reality perspective as well as in the case of the mouse tracking in the standard monitor perspective.

Although each perspective method has inherent differences and constraints, there did not seem to be any significant observable loss of agency with the overall experience between the two modes of interaction; both players were able to successfully compete on relatively level playing fields, completing several matches of the game without feeling hindered by either interface method. In addition to the particular user controls associated with the different interfaces, the overall gameplay seemed to function relatively smoothly. The physics of the gameplay, handled by Unity 3D's physics solver, provided a grounding in reality and offered users a sense of physicality within the digital space.

Although the game worked well in a single-user mode, a major challenge of the system was to enable multiple users to concurrently interact with the environment regardless of their preferred perspective method. After implementing the networking solution described in the previous section, users found the system to work reasonably well under multi-person circumstances. The system seemed to successfully synchronize the pad positions of the users, allowing each user to observe the position of the other in near real-time. Although the system did occasionally experience instances of lag and jitter in game state synchronization, inherent to any networked application, the overall gameplay did not seem to be significantly afflicted by the network connection. In addition to the success in synchronization of pad positions, the synchronization of puck position between the clients also proved effective. The networking schema described in the previous section proved effective due to the invariant that each user is constrained to a particular side of the table and, as such, cannot affect the puck's position while it is on the opposing side of the center divider.

3.2 Conclusions

This thesis describes a system intended to serve as an example of a new form of cross-platform interaction in the digital domain that allows two or more individuals to simultaneously observe and interact with the same virtual experience through inherently different perspectives. More specifically, the system is intended to allow two users, one connected through a virtual reality interface and the other connected via a standard monitor interface, to concurrently play a virtual air hockey game similar to the physical equivalent. In addition to providing users the opportunity to elect a preferred perspective method, a major goal of the system is to minimize loss of agency with the overall experience, regardless of the selected interface method. This is accomplished through providing modified controls that are altered for the inherent differences and constraints associated with each perspective method.

After implementing and testing its functionality with users, the system proved successful in achieving its intended goals. The system allowed users to select a preferred mode of perspective; to concurrently partake in a shared experience; and to do so without loss of agency over the experience, regardless of the differences and constraints associated with the selected perspective method. As such, the system not only demonstrates that multi-person experiences can successfully exist within virtual reality, but also that users connecting to the experience need not be restricted to the single perspective method.

4 Future Work

All work up to now concentrated on implementing the single proof-of-concept example consisting of a virtual air hockey experience. Although the system described in this thesis demonstrated the potential of the new form of multi-person digital interaction, there exist many opportunities to further explore such digital interaction methods and to develop extended cross-platform virtual reality experiences that take advantage of different perspective methods.

Several fields that have thus far benefited from the introduction of virtual reality may also benefit from similar cross-platform virtual reality experiences. Gaming applications may be the most obvious field of application as such systems may allow multiple players to engage not only from different platforms, but also from different points of hierarchical perspective. As an example, future virtual reality games may benefit from an external user who manipulates the gameplay environment through an overseeing perspective via a standard monitor interface. Other fields, such as medicine and military, may also see applications of such systems in the teaching and training of personnel. For example, medical staff may manipulate the conditions of a virtual patient from an external perspective while a trainee is performing a training surgery in a virtual reality environment. Similarly, military trainers may engage with and oversee a virtual reality environment from an external perspective in response to a trainee's action. As such, while this thesis describes a single example of such an experience, there exist several manners by which the system described in this thesis can be built upon and applied to any number of applications.

5 Bibliography

[1] "History of Virtual Reality." Virtual Reality Society. Virtual Reality, 2016. Web. 2 December 2016. <www.vrs.org.uk/virtual-reality/history.html>.

[2] M. Sra, "MetaSpace II: Object and Full-Body Tracking for Interaction and Navigation in Social VR." ARXIV, arXiv:1512.02922, 2015.

[3] S. Pappas, "Why Does Virtual Reality Make Some People Sick?" Live Science, 2016. Web. 2 December 2016. <http://www.livescience.com/54478-why-vr-makes-you-sick.html>.

[4] HTC Vive. http://vive.com.

[5] Unity 3D. http:// unity3d.com.

[6] SteamVR. http://store.steampowered.com/steamvr.