Multi-User Interactive Applications Using Augmented Reality on Mobile Devices

by

Matthew Seita

A Project Report Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science

Supervised by

Dr. Reynold Bailey

Department of Computer Science

B. Thomas Golisano College of Computing and Information Sciences Rochester Institute of Technology Rochester, New York

May 2017
The project “Multi-User Interactive Applications Using Augmented Reality on Mobile Devices” by Matthew Seita has been examined and approved by the following Examination Committee:

Dr. Reynold Bailey
Assistant Professor
Project Committee Chair
I am grateful for Dr. Reynold Bailey, my project advisor, for helping me along the way and making this project possible. I am also very thankful to Jeffrey Bauer, who has kindly allowed me to use his chess models and other materials from his team’s HoloLens chess project.
Abstract

Multi-User Interactive Applications Using Augmented Reality on Mobile Devices

Matthew Seita

Supervising Professor: Dr. Reynold Bailey

The main goal of this project is to design and build an augmented reality application on mobile devices that would allow multiple users to participate simultaneously. Ideally, one person could begin using a networked augmented reality application, e.g. a multiplayer game of chess, then anyone else with a mobile device would be able to tune in and see what the person is doing and potentially participate in the activity. A degree of interactivity is also involved. Each participant will be able see and respond to other participants’ actions in the application. The end result is an interactive experience that is enriched by augmented reality technology.
## Contents

Acknowledgments ................................................................. iii

Abstract .................................................................................... iv

1 Introduction ............................................................................. 1
   1.1 Introduction to Augmented Reality .................................. 1
   1.2 Problem Statement ......................................................... 2
   1.3 Motivation ......................................................................... 2
   1.4 Project Summary ............................................................ 3
   1.5 Agenda ............................................................................. 3

2 Related Works ........................................................................ 5
   2.1 Early Interactive Mobile Augmented Reality Applications .... 5
   2.2 Augmented Reality on Personal Computers ..................... 6
   2.3 Augmented Reality on Modern Mobile Devices ................. 6

3 Design ....................................................................................... 8
   3.1 System Design ............................................................... 8
   3.2 Software Used ............................................................. 11
   3.3 Gameplay Design .......................................................... 12
   3.3.1 Chess ..................................................................... 12
   3.3.2 Pong ...................................................................... 12

4 Implementation .......................................................................... 14
   4.1 Usage ............................................................................. 14
   4.2 Game Setup .................................................................... 16

5 Discussion .................................................................................. 19
   5.1 User Experience .......................................................... 19
   5.2 Challenges Faced ......................................................... 19
List of Figures

2.1 A virtual reality chess game played using the Microsoft HoloLens. . . . . . 7
3.1 System design flowchart. . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
3.2 The default Unity NetworkManagerHUD. . . . . . . . . . . . . . . . . . 9
3.3 The location of the IP address of a network on an iPhone 7 device. . . . . . 10
3.4 A chess model in Unity. . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
3.5 An early rendition of a pong model in Unity. . . . . . . . . . . . . . . . . . 12
3.6 The arc showing the range of directions that the ball could travel after being blocked by a paddle. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
4.1 The application Main Menu. . . . . . . . . . . . . . . . . . . . . . . . . . 15
4.2 The marker image used with the applications. . . . . . . . . . . . . . . . . 15
4.3 Showing the use of a digital marker mage. . . . . . . . . . . . . . . . . . 15
4.4 Screenshot of an AR Chess game on an iPhone 7 Plus. . . . . . . . . . . . 16
4.5 Screenshot of an AR Pong game on an iPhone 7 Plus. . . . . . . . . . . . 16
4.6 A typical scenario where two players play an AR Pong game. . . . . . . 17
4.7 An over-the-shoulder view of an user playing the AR Chess game. . . . . 18
4.8 An over-the-shoulder view of an user playing the AR Pong game. . . . . 18
4.9 A screenshot of a Spectator’ mobile device, watching an AR Chess game. . 18
4.10 An over-the-shoulder view of an user spectating an AR Chess game. . . . 18
Chapter 1

Introduction

1.1 Introduction to Augmented Reality

Augmented reality technology has received much attention over the last few years, and is changing the way users view the world. As the technology begins to evolve, augmented reality is sure to become more integrated into our lives. Many people are starting to experience augmented reality firsthand through popular games like Ingress and Pokémon Go. Augmented reality takes a person’s natural view of the real world and the environment, and augments it by adding computer graphics, sensors, sound, and more. This is done by using the device’s camera to show the real-world scene, and overlapping a virtual scene so that users get a “mixed reality” experience with both real-world and virtual components. These virtual components can be beneficial since they are not necessarily bound by real-world limitations, such as natural laws, and are capable of displaying a myriad of information to the users.

Augmented reality applications typically are either marker-based or marker-less. A marker is an image that the device camera can detect, and once detected, a virtual scene is shown on top of it. Marker-less applications, instead of detecting a pre-defined image, detect real-world objects, such as trees. They typically require detection algorithms in order to be able to identify objects and show the virtual scene. A marker-less application has the advantage of being more universal and generic. The downside, however, is that, compared to a marker-based application, it is much more difficult to implement because the application does not know exactly what to expect when using the camera to analyze
the scene, and it requires more advanced algorithmic techniques and, potentially, increased processing power. A marker-based application is more suited for my project since the application knows exactly what to look for, since the marker image represents the game board, and there is less end-user processing power required which would improve performance especially for a networked application.

1.2 Problem Statement

Thus far, most of the augmented reality technologies created, whether it be on mobile phones, specially-made glasses, or other devices, have been designed for use by one person. To see what one person is doing, other people have to move from their location and crowd behind one screen, causing an inconvenience. Additionally, if multiple people want to work on the same project or task, it is currently difficult to find an augmented reality application that would support that idea well. Another problem is that some of the applications developed that use augmented or virtual reality require sophisticated equipment that not many people own or can afford to purchase. This project attempts to find ways to alleviate these issues, and pave the way for future interactive augmented reality applications on mobile devices.

1.3 Motivation

There have been several applications that use augmented reality or virtual reality to allow participants to interact with a three-dimensional virtual world. These devices include the Microsoft HoloLens and the Oculus Rift. These technologies, however, can be expensive and are relatively rare in that most people do not own one. Additionally, this application would allow users to be able to participate in activities without being bound by physical limitations. They will be able to play a realistic, three-dimensional game of chess anywhere they can bring their mobile devices and object markers, without having to bring a physical chess set. They will also be able to play a pong augmented reality game, which brings a
three-dimensional game experience that is not bound by rules of physics and natural laws. This project hopes to allow regular, everyday users to be enriched by augmented reality technology through the use of mobile devices that are already ubiquitous.

1.4 Project Summary

For this project, an augmented reality mobile application was developed for iOS devices, such as iPhones and iPads. Once opened, the application allows the user to select which mode they want to use, a multiplayer chess application or a multiplayer pong application. The chess application allows the user to play a game of chess using augmented reality technology. The chessboard and pieces are all three-dimensional, and users can move and capture pieces using their mobile devices. The pong application allows users to play a pong game. The pong stadium, paddles, and ball are all virtual and three-dimensional, and not bound by the laws of gravity. Users can play the game by moving their pong paddle using their mobile devices.

Users are able to host a server that allows other users to join and spectate what is happening or participate in the activity. For example, if a user opens the chess application, another user can join to play a game of chess. Other people who want to spectate will also be able to join the game, so they can spectate the game in augmented reality on their own mobile devices. This project hopes to offer an enjoyable, realistic, and interactive experience in the virtual world.

1.5 Agenda

The rest of the paper is organized as follows: first, in Chapter 2, some more background and related works on this subject are discussed. Next, Chapter 3 discusses the design of the project. Then, the implementation of the project will be described in Chapter 4. A discussion of the final application and the project as a whole will be presented in Chapter 5. Finally, Chapter 6 concludes with the current status of the project, possible directions
for future work, and the lessons learned throughout this journey.
Chapter 2

Related Works

2.1 Early Interactive Mobile Augmented Reality Applications

Mobile interactive augmented reality applications are not a very recent phenomenon. Augmented reality has been a topic of interest for several decades. For example, in 2001, Reitmayr and Schmalstieg developed a mobile augmented reality application that had multiplayer functionality and was capable of playing a chess game [4]. Their work uses a marker image to display the virtual world and allows several users to collaborate in real time. However, their mobile setup is very cumbersome and requires specialized software and equipment, such as headgear and a pen-and-pad interface. We hope to provide a simpler setup that would allow everyday users to be able to play through mobile devices they already own.

Wagner et al. [5] published a paper that described their approach towards massively multiplayer augmented reality applications. They described how previous approaches to multiplayer augmented reality, which included headgear and backpack mounted devices, were hindered by a lack of robustness and their relatively expensive and cumbersome nature. They developed a system architecture for moving towards handheld devices for augmented reality. They successfully created a simple multi-user augmented reality game for PDAs and took this as a sign of encouragement for similar projects in the future. One of my goals was to draw from this line of thinking and apply it to interactive augmented reality applications on mobile phones.
2.2 Augmented Reality on Personal Computers

Fahn et al. [2] developed a more modern interactive augmented reality game for personal computers. They were able to use the webcam on the PC and marker-based augmented reality to create two games: an interactive monopoly game and an interactive fighting game. Their findings were promising as they were able to keep a degree of interactivity in their games, similar to the degree of interactivity people feel when playing a tabletop board game with friends, while relegating the boring computational parts of the game, such as money management, to the computer itself. This is a major reason why many people are interested in playing augmented reality games because of the degree of interactivity maintained and the ability to interact with a three-dimensional virtual environment.

2.3 Augmented Reality on Modern Mobile Devices

Collaborative augmented reality applications on more modern mobile devices have been developed previously as well. For example, Second Surface, developed by Kasahara et al. is a spatial collaborative system that uses augmented reality [3]. Second Surface uses what is known as a shared canvas where users can add their own creative designs to the canvas. This addition is automatically reflected on other users screens, so changes that one users makes are visible to all other users. Participants can use their own mobile devices, such as an iPad, within a network to show their drawings and three-dimensional objects, placed in the shared canvas, in real-time. My project here was inspired in part by the work presented in [3], but instead of creating a marker-less shared canvas application, I set out to create a marker-based interactive application that would allow users to play multiplayer games over a network.

Additionally, students at the Rochester Institute of Technology developed a virtual reality chess game in 2016 using the Microsoft HoloLens [1]. This project allowed users to play a multiplayer game of chess using their hands to move and capture pieces. A screenshot of this virtual reality chess game is shown in Figure 2.1. This was another inspiration for this
project and I wanted to implement something similar using augmented reality on mobile phones, because the Microsoft HoloLens are relatively expensive and not ubiquitous in the general population. Jeffrey Bauer, one of the students who worked on this project, has graciously allowed me to use some of the materials from his team’s project, including the chess models and more.
Chapter 3

Design

3.1 System Design

The overall system diagram flowchart is shown in Figure 3.1. The following paragraphs are explanations of components of the system diagram flowchart.

To play the games, all the users have to have their own mobile device and be connected to the same network. A simple way to do this is to have all the devices be connected to the same wireless network (Such as the RIT Wi-Fi). If this is not done, then the mobile devices will not be able to see each other on the network and no connection can be made since joining a game room is done by connecting via IP address on the network.

The first player that creates a game, which is done through the NetworkManagerHUD, becomes the host of the game. The NetworkManagerHUD is the default Unity networking HUD and is shown in Figure 3.2. A player can become the host by clicking the topmost button on this HUD, labeled "LAN Host (H)". The game is then hosted on the network that the first player is on. This player is simultaneously a local client and a server, and is the designated "Player1". Even though the client and the server are on the same device, the general networking design principles still apply. The local client provides input, such as clicking the screen to move a chess piece to a certain tile, to the server which updates the view on the client.

Joining the game can be done via IP address. First, the host of the game must find their own IP address and then inform the other players what the IP address is. For example, on an iPhone 7, the procedure to do this is to go to ”Settings” → ”Wi-Fi” → Press the blue
Figure 3.1: System design flowchart.

Figure 3.2: The default Unity NetworkManagerHUD.
circle, with an "i" in the center, on the Wi-Fi network you are connected to → find the IP address in the "IP address" field, which is highlighted with a red box in Figure 3.3. The other players must then enter this IP address on their own devices, in the text box which has the text field "localhost" to the right of the second button on the HUD in Figure refnetworkHUD, labeled "LAN Client (C)". After entering the IP address in the text box, the other users then click the "LAN Client (C)" button to join the game. The first player that joins the game becomes the designated "Player 2" and will be the host’s opponent in the match. The second player is a remote client, because the server is not being hosted on their phone. Thus, their input commands will be sent over the network, and the server then updates the view on the remote client’s phone.

Other players beyond the second player can join the game by using the same procedure, but they will be relegated to "Spectator" positions. These spectators won’t play the game, but they can watch the game unfold in real time through their own mobile device. Since they are not players, their inputs and commands will not be sent to the server. However,
because their mobile devices still need to be notified when either Player 1 or Player 2 makes a move, the server will still update the spectators in real time.

### 3.2 Software Used

This project was created with the aid of several pieces of software. Augmented reality technology was enabled by the Vuforia Software Development Kit. Unity3D was used for the object modeling, iOS development, and implemented much of the app functionality. Some of the models done in Unity3D for the chess and pong games are shown in Figure 3.4 and Figure 3.5. Unity Networking (UNET) was used for multiplayer functionality and server/client connections, and scripting was done using C#. Finally, Xcode was used to port the application to iOS devices. The application will work on any iOS devices (iPhone 7, iPhone 7 plus, iPad, etc.). An android application is not available at this time.
3.3 Gameplay Design

3.3.1 Chess

The chessboard itself was modeled in Unity3D and divided into an 8 by 8 grid. Each tile in the grid is a separate object and can be selected by clicking on it on the device screen. Each piece is also an individual object and is selectable, again by clicking on it on the device screen. The models for the chess pieces are courtesy of [1]. A player can move a piece by first clicking on a piece then by clicking a tile where the piece is to be moved. If that tile has an opposing piece, it is captured and removed from the board. The game is won when checkmate is achieved and the opposing King cannot avoid being captured.

3.3.2 Pong

The pong stadium was also modeled in Unity3D. The server owns the stadium and the ball, while the players own the paddles. Only the host (Player 1) has the ability to start the game, which is done by pressing the "Start Game" button. Once pressed, the button disappears and the ball begins to move. Players must prevent the ball from crossing their side of the stadium. They do this by moving the paddles left or right by tapping their device’s screen. The paddle moves a fixed distance in a certain direction depending on which side of the screen the user taps. If a collision is detected between the ball and the paddle, the ball
Figure 3.6: The arc showing the range of directions that the ball could travel after being blocked by a paddle.

Ricochets in a random direction, within a 90-degree arc, centered about the normal to the surface of the paddle, towards the opponent. This concept is illustrated in Figure 3.6. If the ball hits the side of a stadium before reaching the opposing player, it bounces off the side in the perfectly reflected direction (the angle of reflectance is the same as the angle of incidence). If a player fails to block the ball from crossing their side, that player loses and the game is reset to its original state and can be restarted by clicking the "Start Game" button again. This time, however, the ball first moves towards the person who recently lost the game.
Chapter 4

Implementation

4.1 Usage

Opening the application itself first directs the user to a Main Menu, which is shown in Figure 4.1. The Main Menu allows the user to either start an AR Chess game, start an AR Pong game, or Quit the application. After opening either AR Chess or AR Pong, the user must first allow the device to use the camera and then point the camera at a marker image. The marker image I chose for these two applications, shown in Figure 4.2 was arbitrary. The marker just needs to be asymmetrical and preferably of high quality to allow the camera to capture the position and orientation of the image and virtual scene. After pointing the camera at the marker image, the virtual scene of the game, either a chess board or a pong stadium, will show up. The marker image can either be printed out on paper or it can be a digital image, as shown in Figure 4.3. There is a network interface, the Unity NetworkManagerHUD, that allows users to host a game (server) and join an existing game by inputting the hosts IP address. Once two players join the game, they can begin playing. More than two users can join, but the rest will be spectators.

The AR Chess game is shown in Figure 4.4. Users have as much time as they want to perform their move, and they can move pieces by first clicking on a piece and then clicking on a space on the board. The game is played using conventional Chess rules and a player wins by capturing the opposing King.

The AR Pong game is shown in Figure 4.5. After two players join the game, The host of the game can press the Start Game button on their phone screen to start moving the ball.
Figure 4.1: The application Main Menu.

Figure 4.2: The marker image used with the applications.

Figure 4.3: Showing the use of a digital marker mage.
Users can block the ball from passing their side of the stadium by moving their paddle. A player loses when the ball passes their side, and if this happens the game resets and another game round can be played.

### 4.2 Game Setup

Two players play the game by placing the marker image on a flat surface then sitting or standing on opposite sides of the marker image. The positions of Player 1 and Player 2 are predetermined. The marker image used was the Hiro image, so Player 1 is the person who is placed on the side such that he can read the letters on the image. "Player 2" is the person who is on the opposite side, and the letters are upside down relative to Player 2. The host is always designated as Player 1, so the host should position themselves in the appropriate side of the marker image. In Figure 4.6, the man pictured is on the Player 1 side of the marker image, so he is the host of the game. The woman is then assigned Player 2. For reference, Figures 4.7 and 4.8 show an over-the-shoulder view of the players when
Figure 4.6: A typical scenario where two players play an AR Pong game.

playing a chess and pong game.

The positions of the spectators do not matter, since they have no impact on the game. The spectators, once they join the game, can position themselves wherever they want, depending on their preferences and where they think they would get the best view of the game. Figure 4.9 shows what a spectator’s screen would look like if they were positioned to the side of the marker image, and Figure 4.10 displays an over-the-shoulder view of the spectator. Once the players are in the appropriate positions, then they can proceed to play the games as described in the previous sections of this paper.
Figure 4.7: An over-the-shoulder view of an user playing the AR Chess game.

Figure 4.8: An over-the-shoulder view of an user playing the AR Pong game.

Figure 4.9: A screenshot of a Spectator’s mobile device, watching an AR Chess game.

Figure 4.10: An over-the-shoulder view of an user spectating an AR Chess game.
Chapter 5

Discussion

5.1 User Experience

The goal of this project was to create augmented reality applications that were interactive and enjoyable. Most of the users who tried the applications were very interested in it and found it enjoyable. It was a new and exciting experience for them, as many of them were not extensively familiar with augmented reality technology. This reveals a promising outlook for the future of interactive mobile augmented reality applications if they are created and are available, users will probably be interested in downloading and playing with them.

5.2 Challenges Faced

5.2.1 Development

Throughout development of the applications outlined in this project, there were some issues regarding creating applications with augmented reality. Development in Unity had to be done with Vuforia in mind, and this requires that several things be done differently compared to how things would be done in a regular, non-augmented reality application. For example, in augmented reality applications using Vuforia, development is centered around a GameObject called an ImageTarget. This is the target image on which the device ’s camera is pointed to display the virtual environment. Everything that needs to be displayed using augmented reality needs to be made a child of the ImageTarget object. This caused an issue when testing a networking implementation because the GameObjects in Unity were not
attaching appropriately to the ImageTarget and clients could not see changes done by other users, even though the implementation worked in a non-augmented reality environment.

Personally, another challenger for me was that I am new to mobile development and game development in general. I had minimal experience with Unity and no experience with any mobile development at all. It was a learning process for me. I learned a lot as I progressed through the semester and I had to rely a lot on guides and sources that I could find on the internet. Overall, though, I am happy to have had the opportunity to embark on this project, as it taught me new skills related to application development and could be useful in the future. Additionally, despite the frustrations faced, it was fun and entertaining project to work on.

5.2.2 Testing

Testing the applications for this project was also time-consuming. This was especially true because this project combines both augmented reality and networking. Developers of regular, non-augmented reality applications who want to test networking can launch an instance of the game and test connections using Unity’s Play mode. Things become more difficult when testing games using augmented reality. While Unity certainly has the capability to test augmented reality applications using the computer’s webcam, the computer itself has only one camera. Thus, only one augmented reality application can be running on a computer at a time. Testing networking and multi-device connections requires two or more devices to be running. To thoroughly test the applications, I had to build the game directly on the phones and test it on these mobile devices. Building the game from Unity to Xcode to the iPhone required approximately 4 to 6 minutes per build. Testing any changes in the workspace environment necessitated another build onto the iPhones.

The length of time required to build onto the iPhones sometimes caused frustration. Take, for instance, a situation where I needed to test several methods through trial and error. For example, when testing which of three functions would be the best fit for a certain script in Unity, I had to try one function, compile and build it, and test it on the application,
and repeat this process two more times. In this situation it would take almost 20 minutes to test one line of code. Changing lines of code in various scripts are not the only thing that requires a re-building of the application on the phone. Testing any change in the Unity editor, such as adjusting object transformations, attaching scripts, or adding a component to a GameObject requires a build. This is another problematic issue for testing because to circumvent the long build time, several changes were made at once before compiling rather than testing each change one by one. If the application does not work as expected, it can be difficult to know which change in particular was causing the bug. Long build times and testing applications is definitely a big challenge for developing networked, interactive augmented reality applications.
Chapter 6

Conclusions

6.1 Current Status

The successful use of augmented reality with interactive games on mobile devices is a good sign for the future of interactive augmented reality applications. The most important finding out whether it was possible to integrate augmented reality with Unity for interactive networked applications, and this is indeed the case. The second step was determining its feasibility. This project shows that an augmented reality interactive application can be created in a timely manner without insurmountable challenges. It is definitely possible to expand on this idea to create more intricate and complex interactive augmented reality applications in the future.

Currently, the project consists of a single iOS mobile application that users can download onto their devices. Opening up the application allows the users to play either chess or pong, enhanced by augmented reality technology, although the two games have minimal features, are not necessarily well-polished, and are not highly sophisticated. This opens up several paths for future work on this project.

6.2 Future Work

There are several possible future directions for this project. The functionality for the applications in this project are relatively minimal, so one idea for future work could consist of polishing the application, improving aesthetics, and adding more features. Some ideas for additional features include adding an undo move button and a turn timer for chess, and
adding the option to increase the speed of the ball, which ups the difficulty, increasing the number of players to 4, and adding a scoreboard in the pong game.

The application can additionally be improved by finding ways to reduce latency, as sometimes the changes one users makes are not reflected on other clients devices in a timely manner. The networking component of this project can probably be optimized in order to make the gameplay more fluid.

Another approach for future work is working on a different set of interactive applications, not just games. An example could be an object modeling application where users can create, modify, and destroy different three-dimensional shapes to create interactive models on a shared canvas.

Finally, another idea could be to create these games and other apps for android phones, not just iOS devices.

### 6.3 Lessons Learned

This project was definitely a learning experience. Mobile application development is very different than working in a PC-based environment, and I didn’t realize how much the learning curve would impact me, time-wise. Additionally, latency and networking is also a concern for the project, as even simple games like Chess and Pong have noticeable delays when playing, although this is more of a concern for real-time games, such as pong, compared to turn-based games like chess.
Bibliography


