12-1-2021

TEXAS POKER VR

Allison Gardiner

TEXAS POKER VR

by

ALLISON GARDINER

Presented to the Faculty of the Honors College of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

HONORS BACHELOR OF SCIENCE IN COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2021

ACKNOWLEDGMENTS

I would like to acknowledge my team, Texas Game Developers, who made this project happen. The team consisted of Senior Design I students: Bailey Nguyen, Evan Park, Hadrien Nguyen, and JP Sims, and Senior Design II students: Madison Adams, Zinah Aljanabi, Hemanta Lewaju, and myself Allison Gardiner. This project would not have happened without them.

I would also like to thank my faculty mentor, Chris Conly. He was always willing to teach and willing to help, no matter the day or time. I have learned a lot from him, and appreciate the time he spent making this time.

Lastly, I would like to thank my family for always encouraging me to work hard even when life gets tough. My parents have encouraged me to set my expectations and my goals high. Their love and encouragement have allowed me to graduate from the Honors College, as a female Computer Science Engineer, and D1 athlete.

December 7, 2021

ABSTRACT

TEXAS POKER VR

Allison Gardiner, B.S. Computer Science

The University of Texas at Arlington, 2021

Faculty Mentor: Chris Conly

Texas Poker VR is a virtual reality (VR) poker game that provides players with the ability to play privately among friends. The Texas Game Developers team is working to develop a VR poker game that has a focus on the private room and socialization aspects of the game. The application is for the Oculus Quest 2, using the Unity game engine. We will be using a subscription server that will allow players to create a private room with other players they invite, visualize the poker screen, socialize with each other, and keep track of the fake money, all on a real-time basis. Because existing VR poker games have shortcomings in which players cannot create a private room to play poker with real friends, our primary focus as a team is to solve this problem and have more private socialization abilities, in addition to creating a VR poker game.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

PROJECT CHARTER

To understand the purpose of the honors aspect of the senior design project, it is important to explain the background and overview of the senior design project for the College of Engineering. The problem statement for this project was that the sponsors wanted a way to play Virtual Reality Poker with their friends, without being paired with random players online. There are multiple games that already exist that allow players to play VR Poker with random players, but there was none that exist to allow friends to play together, until now.

1.1 Problem Statement

Texas Poker VR is a virtual reality (VR) poker game that provides players with the ability to play privately among friends. This project came into existence because our sponsors, George Kramerov and Chris Conly, wanted to play poker with their friends while staying at home. Due to COVID-19, concerns such as social distancing, health, and safety have led to the growth in online and virtual gaming. Moreover, the idea of playing poker virtually with friends in a private room has not been a feature of existing VR poker games.

1.2 Methodology

We developed a VR-based poker game application for the Oculus Quest 2 VR system, using the Unity game engine and through Git collaboration (i.e. GitHub). We used a subscription server to host the game, since the poker game will require this in order to run the game. The subscription server will also allow players to create a private room with other players they invite, visualize the poker screen, socialize with each other in the private room, and keep track of the fake money, all on a real-time basis. Since existing VR poker games have shortcomings in which players cannot create a private room to play the poker with friends, our primary focus as a team was to solve this problem and have more private socialization abilities, in addition to creating a VR poker game.

1.3 Value Proposition

Existing VR poker games have a shortcoming in that they do not have the ability to rent or use a private room for players to play poker with friends. They put users in a room with random strangers where the game feels very rushed. This app allows players to be able to play with friends or invited players. This helps create a relaxing environment for the players to play the poker game without any rush in which they can take their time and friends can privately socialize in the ways they want.

1.4 Development Milestones

The following list is that of core project milestones and includes all major documents, demonstration of major project features, and associated deadlines:

- Project Charter First Draft - July 2021
- System Requirements Specification - July 2021
- Architectural Design Specification - August 2021
- Demonstration of Base VR Environment - September 2021
- Detailed Design Specification - September 2021
- Demonstration of Basic Poker Gameplay - October 2021
- Demonstration of Private Rooms - October 2021
- Demonstration of Basic Chat Functionality - November 2021

- Demonstration of Stats Monitoring & Player Ranking System - November 2021
- Final Project Demonstration - December 2021

1.5 Background

With virtual activities and hobbies getting more popular after the pandemic, virtual games, like VR Poker, are more in demand than ever. Available VR poker games currently offer features such as audio chat, avatar customization, private tables, and many more. However, the highly requested feature that these games lack is the concept of offering private rooms with the option of inviting only the player's friends, without the game adding a random person. Our customer desires to have this feature in the poker game, which will mimic a real-life poker experience, as if they went to a casino to play in person. The private room feature will attract plenty of players who want to only play with friends, have a choice to play with random players, and take their time playing without feeling rushed through the game.

1.6 Related Work

PokerStars VR is a free-to-play VR poker game that launched in 2018 [1]. It offers various features, including audio chat interaction, ordering food & drinks, and private tables. It works with Oculus and Viveport VR platforms, as well as through Steam. Recently, the tournament feature was introduced among players. And it supports live streaming on Twitch and Oculus live [4]. Poker VR is a free-to-play VR poker game that launched in 2018 [7]. It offers various features such as personalizing avatars with life-like features [2], body and lip moments, and the exchange of gifts among players. It works with Oculus Rift, Oculus Go and Samsung Gear VR. There also is a tournament feature among players [5].

1.7 System Overview

For the VR Poker Game [8], we will have an Oculus device that will use Unity software. The VR environment will have two options - a private room and a public room to play. The data will be sent and retrieved through our database. A high-level diagram of this is shown below.

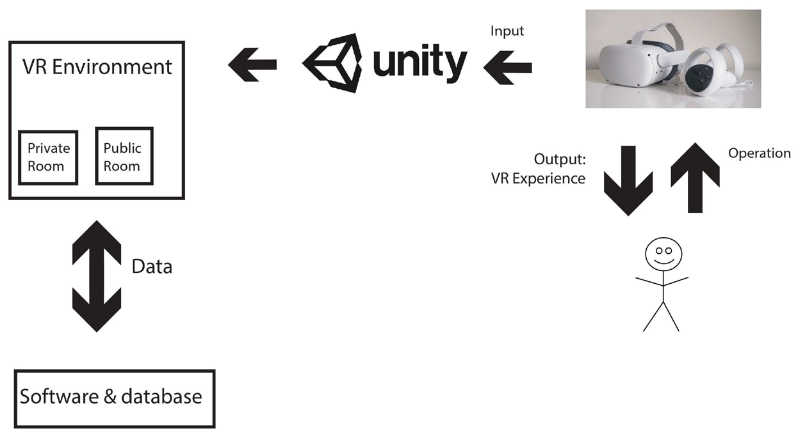

Figure 1.1: System Overview of Major Components

CHAPTER 2

ARCHITECTURAL DESIGN SPECIFICATION

2.1 Architectural System Overview

Players are be able to download and access the game through the Oculus store. Texas Poker VR has a system consisting of four layers: the User Layer, Hardware Layer, Cloud Layer, and the Database Layer. Each layer will be further discussed throughout this chapter. Below is an architectural layer diagram of the entire system.

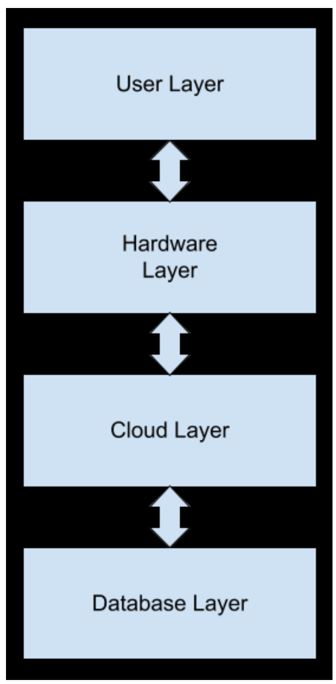

Figure 2.1: Architectural Layer Diagram

2.1.1 User Layer

The User Layer consists of User Input subsystem and User Interface subsystem. User input allows the player to communicate with the system and with other players. The user interface is what the player will see and hear, and it is how the system will communicate with each individual player.

2.1.2 Hardware Layer

The Hardware Layer consists of the Oculus 2 VR System/Kit subsystem and is related to the performance required for an optimal gaming experience for the player. It is the interface in which the player provides input through and receives output from. In order to use this VR system, the player needs two Oculus Touch controllers, the VR Headset, and the Oculus app (mobile or desktop). Each of these components should be fully connected to each other and working. The headset firmware should also be up-to-date.

2.1.3 Cloud Layer

The Cloud Layer consists of an AWS cloud server where the game service will be hosted and will be running 24/7. The build file running in this server interacts with the Hardware Layer when the Oculus user sends a request to connect to the running game service while playing the game, and to create the necessary environment for the user. Also, the game service will connect to the SQL server and database as needed. The AWS cloud server is where the SQL database server will be hosted and made available anytime. Real-time updates of the game data and database data will be performed and provided through the game service running on AWS.

2.1.4 Database Layer

Data from Texas Poker VR is stored and managed in a database within the server. This layer interacts with the Cloud layer to store or retrieve data as needed. Based on the scope of the game, we need to store player information and the private room attributes. The database layer stores information of the player once the account is created. These credentials will need to be retrieved and checked while performing the authentication and login process. In addition, the private room ID, player information, and the player's private room customization will be stored as well.

2.2 Subsystem Definitions & Data Flow

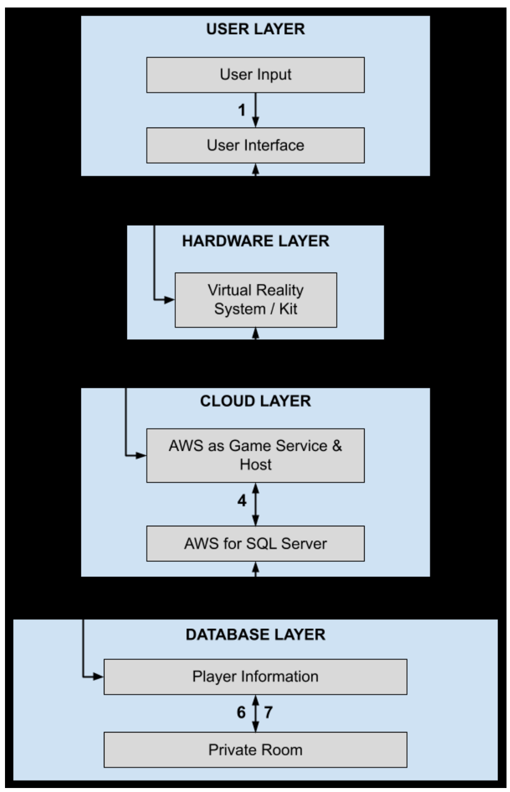

Figure 2.2: Data Flow Diagram

Below is a data flow diagram of the Texas Poker VR system that illustrates the subsystems of each layer and how they're connected.

2.3 User Layer Subsystems

The user layer subsystem is a main function of the system and will be the most obvious to the user and customer. The user layer has two main subsystems within itself: User Input and User Interface.

2.3.1 User Input

User input is defined as any decision the user makes within the application. This could be typing their name, entering or creating a public or private room, or making a move during gameplay. The system will respond to every user input and the response is shown with the user interface.

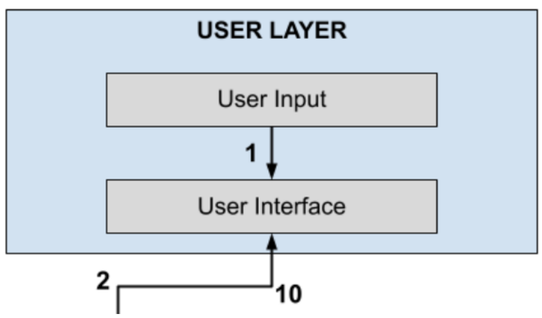

Figure 2.3: User Input Subsystem Diagram

2.3.1.1 Assumptions

All input data is valid input and is recorded in the database.

2.3.1.2 Responsibilities

The responsibility of the User Input subsystem is to allow the user to communicate with the system, and in turn, other users. The User Input subsystem and User Interface subsystem go hand in hand and interact very closely.

2.3.1.3 User Input Subsystem Interfaces

ID	Description	Inputs	Outputs
#1	Account	Name Email	Account Created
#2	Avatar	User will choose among list of avatars	Avatar added to account

Table 2.1: User Input Subsystem Interfaces

2.3.2 User Interface

The user interface is what the user sees and interacts with. It is an important aspect to the game and will be used to communicate with the player and between players. The interface will display options to the user and will react based on the particular choice of the user. The interface will display each room, public or private. It will also display the names of each user in a particular game.

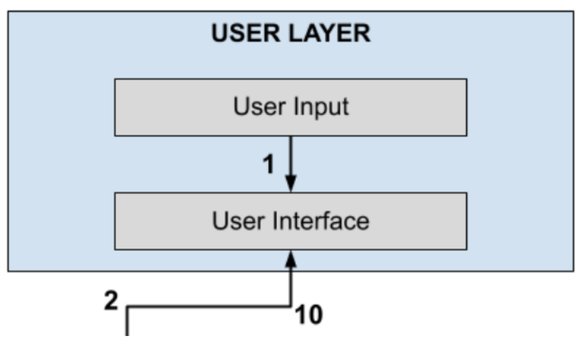

Figure 2.4: User Interface Subsystem Diagram

2.3.2.1 Assumptions

- User input only happens when there is a clickable overlay.
- The interface system will respond to every single input from the user.
- The user interface is connected to the database layer.

2.3.2.2 Responsibilities

This layer is responsible for communicating to the user from the database, and vice versa. The user interface will display all possible choices to make to the user and will respond correctly to the user based on the choice made by the user.

2.3.2.3 User Interface Subsystem Interfaces

ID	Description	Inputs	Outputs
#1	User interface processes inputs and sends to hardware layer	Valid input from user confirmed by clicking a button	Decision is displayed on interface screen
#2	Authentication	Email Password	Object to server
#3	Room Type	Select Private or Public	Public of Private room options will be shown

Table 2.2: User Interface Subsystem Interface

2.4 Hardware Layer Subsystems

The Virtual Reality System / Kit subsystem bridges the gap between the player and the VR poker game. It is used for the gaming and VR experience. This subsystem communicates with the User Interface subsystem within the User Layer and the AWS as Game Service & Host subsystem within the Cloud Layer. The player provides all input through the User Interface subsystem, which is sent to the Virtual Reality System / Kit subsystem. Then the Virtual Reality System / Kit subsystem sends this input to the AWS as Game Service & Host subsystem, which creates game updates and gameplay and sends it back. The Virtual Reality System / Kit subsystem receives this game output and sends this game output and physical output, which includes vibrations through controller, music and sounds from game through speakers, and graphics/visuals through headset, back to the player.

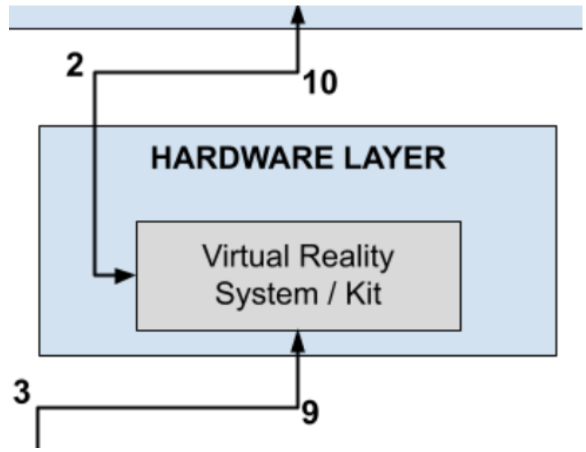

Figure 2.5: VR System/Kit Subsystem Diagram

2.4.1 Assumptions

Players have access to the internet and the Oculus Quest 2 VR system. Internet bandwidth and quality may affect loading times. The headset will be up-to-date and the player will be able to see the game and graphics. The buttons on the controllers will be working and provide input to the hardware. Any motion and movement will be sensed by the sensors and provided to the hardware as well. The system will be connected to WiFi and all of its components connected together. The server that the game is hosted in also must be up to prevent delays or lack of loading and ensure optimal gameplay and game updates.

2.4.2 Responsibilities

This subsystem is responsible for receiving and handling all the input that the hardware will receive from the player. This includes head tracking, hand tracking, button presses and movement from controllers, voice input through microphone, and from additional (but not required) sensors and buttons from the user [1]. The subsystem is also responsible for sending and receiving output and gameplay updates to the player. This includes all the output that the hardware sends to the player through the built-in speakers, lenses and headset include the VR environment, experience and performance, gameplay/graphics, and music & audio from the game [2].

2.4.3 Virtual Reality System/Kit Subsystem Interfaces

Each of the inputs and outputs for the VR system / kit subsystem are defined in the table below.

ID	Description	Inputs	Outputs
#1	User interface sends user inputs to game through VR system	User interface input	Game data sent to AWS layer
#2	Processed game data and gameplay from game server is sent to user through VR system	Game data	VR system outputs sent to user

Table 2.3: Virtual Reality System/Kit Subsystem Interfaces

2.5 Cloud Layer Subsystems

AWS will be used as the game service host environment and as the server for database layer. All the caching, virtualization, and book-keeping behind the scenes will be done here.

2.5.1 AWS as Game Service & Host

AWS will be used to host the VR poker game service and will be running 24/7. This will ensure that any player is able to join the game environment any time they need and can play whenever they want. The player will be able to join/play the game from the server by using an Oculus Quest 2. The AWS cloud server will connect each user using their Oculus Quest 2 device providing the game environment. Currently, the Oculus Quest

2 is the only interface that the player/user can use to connect to this poker game service. The necessity of AWS is for real-time connection and online communication between players in different regions and to provide synchronization between them. Moreover, every player has the ability to create a separate game environment anytime through the use of that player's private room. This is only possible by the virtual environment and background book-keeping provided by the service running on AWS.

2.5.1.1 Assumptions

Users will be using the Oculus Quest 2 VR system to connect to the game.

2.5.1.2 Responsibilities

The game service hosted on AWS will be built by using the Unity game engine, the OpenXR API and programmed to connect the hardware using built in Oculus API.

2.5.1.3 AWS as Game Service & Host Subsystem Interfaces

Inputs are all of the user moves, visuals, voice/chats incoming from the player, all the calculations regarding player scores, and fake money. Outputs are all the updated information to the SQL database and the visual, other voices/responding chat, and updated information to the user.

ID	Description	Inputs	Outputs
#1	Oculus with its built in API is how it connects hardware to the build game service provided by AWS host	Users input Calculated information	Oculus Quest 2 SQL database

Table 2.4: AWS as Game Service & Host Subsystem Interfaces

2.5.2 AWS for SQL Server

The player's information such as the amount of fake money, points and game scores will be stored using SQL database server, which will also be hosted using the AWS cloud server. The real-time updated information will be stored in SQL as well. The Oculus Quest 2 user needs all this data to be persistent and available anytime they start, play, and/or end a game. For this functionality to be available, AWS as the server of the SQL database will be key.

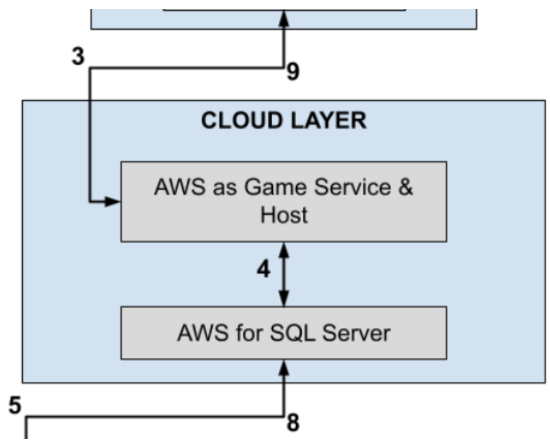

Figure 2.6: AWS for SQL Server Subsystem Diagram

2.5.2.1 Assumptions

Users will be using Oculus Quest to connect to the game. Also, the game will be correctly connected to the database in order to perform database management and store/retrieve data.

2.5.2.2 Responsibilities

The SQL server is how we will be storing the users' information and different components of game information. Real-time updates will also be done based on players status in the game. AWS will be hosting the SQL server keeping it online all the time and available to the game as needed.

2.5.2.3 AWS for SQL Server Subsystem Interfaces

Inputs are all the user scores incoming from the players calculation and fake money. Outputs are all the updates and updated information from the SQL database to the user.

ID	Description	Inputs	Outputs
#1	Oculus with its built in API is how it connects hardware to the build game service provided by AWS host and then to the SQL server hosted on AWS	Users' data Calculated information	Oculus Quest 2

Table 2.5: AWS for SQL Server Subsystem Interfaces

2.6 Database Layer Subsystems

The database layer is the layer where all the information and the data are stored. The database layer is divided into two different subsystems based on the type of the data stored.

2.6.1 Player Information

In this database layer, information and credential of the player will be stored like username, password, player avatar customization. This information will be sent to the cloud server layer. The server layer will interact with this subsystem layer and store this information. Player credentials will also be checked by the server layer base on the information stored in this layer.

2.6.1.1 Assumptions

Player is already signed up and the information stored in this layer. The player login and authentication is done by authentication function of AWS on the server layer.

2.6.1.2 Responsibilities

The Responsibility of this subsystem is to store and manage the information of the player. Also, this subsystem will retrieve player data based on the query done by the server.

2.6.1.3 Player Information Subsystem Interfaces

ID	Description	Inputs	Outputs
#1	Saving player's data	Player data	Save data
#2	Retrieve player's data	Database query	Data objects

Table 2.6: Player Information Subsystem Interfaces

2.6.2 Private Room

2.6.2.1 Assumptions

Player already signed up and the data is already stored in this layer. Player login and authentication is done by the authentication function of Amazon Web Services (AWS) on the server layer.

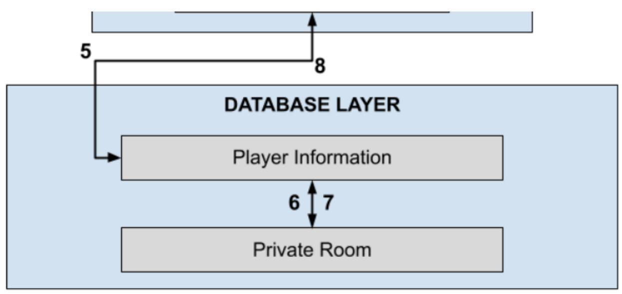

Figure 2.7: Private Room Subsystem Diagram

2.6.2.2 Responsibilities

The main responsibility of this layer is to store and manage the Room information. This subsystem also retrieves player information and sends to the server layer based on the query done by the server.

2.6.2.3 Private Room Subsystem Interfaces

ID	Description	Inputs	Outputs
#1	Saves Private Room information	Room ID, name and customization	Saves data
#2	Retrieve Room information and player's data	Database query	Data objects

Table 2.7: Private Room Subsystem Interfaces

CHAPTER 3

HONORS REQUIREMENTS

Although the game was completed as a whole by a group, my individual honors contribution consisted of creating and establishing a private room. Each private room can be created by a user, and the user can choose who he or she would like to invite to play with them in their private room. Then, each owner of each private room can customize the room to his or her liking based on color of the table, color and texture of the walls, design of cards, and more.

3.1 Visual Design

The initial environment base was purchased as a Unity Asset – "Poker 3D and 2D, Texas hold 'Em, Photon PUN, Multiplayer Template Unity" by Vagho. This consisted of the table, chips, cards, and minimal poker logic. From there, I added the ability to personalize any room by any room owner.

3.1.1 Public Room Visuals

This image is what a user would see in a public room. This is the case if they choose to be paired to play with random players via Internet, rather than close friends they have chosen.

Figure 3.1: Public Room Example

3.1.2 Private Room Visuals

This image is what a user might see in a private room. This is the case if they choose to be play with close friends they have chosen. This is an example UTA room, but it is not the extent of the customization properties.

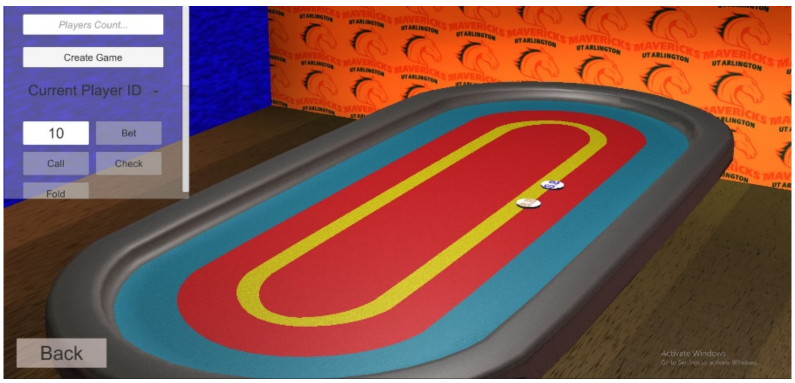

Figure 3.2 Private Room Example

<u>3.2 Server Connection</u>

3.2.1 Home Screen

This image is what a user would see when first logging in and opening the game. They would choose either *New Game* or *Server Game*. *New Game* would send the player into a public room with players randomly selected to play against. *Server Game* would send the player to a second screen, asking the user to create or join a room. If they choose to create a room, they will be given a code to pass to friends to start a game. That player would also be the owner of that room and be able to design it however they would like. If the player chooses to join a room, they will have to enter a code that they got from a friend. That player will be able to play with their friend who invited them, but not be able to customize anything in that room.

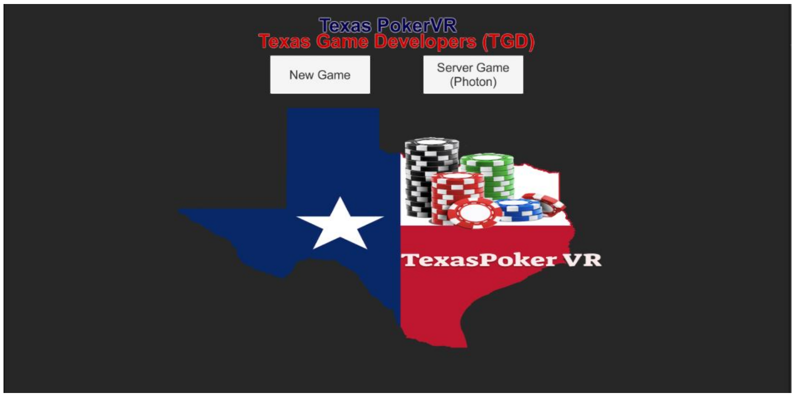

Figure 3.3: Home Screen

3.2.2 Server Connection

AWS Sumerian is used to host the VR poker game service and will be running 24/7. This ensures that any player is able to join the game environment any time they need and can play whenever they want. The players are able to join the game from the server by using an Oculus Quest 2. The AWS cloud server connects each user using their Oculus Quest 2 device providing the game environment. Currently, the Oculus Quest 2 is the only interface that the player can use to connect to this poker game service. The necessity of AWS is for real-time connection and online communication between players in different regions and to provide synchronization between them. Moreover, every player has the ability to create a separate game environment anytime through the use of that player's private room. This is only possible by the virtual environment and background book-keeping provided by the service running on AWS.

3.2.2.1 Amazon Elastic Block Store

The player's information such as the amount of fake money, points and game scores will be stored using SQL database server**,** which will also be hosted using the AWS cloud server. The real-time updated information is stored in SQL as well. The Oculus Quest 2 user needs all this data to be persistent and available anytime they start, play, or end a game. For this functionality to be available, AWS as the server of the SQL database is key.

REFERENCES

[1] Anuj Arora. Pokerstars vr expands beyond poker, now supports blackjack and slots, Oct 2020.

[2] Poker Dictionary. Virtual reality: The future of online poker?, Sep 2020.

[3] Daria Dubrova. Virtual reality apps development: 8 problems to overcome, May 2018.

[4] PokerStars. Pokerstars vr: Poker with no limits.

[5] Hamza Siddiqui. The mutant update and poker vr championship, Oct 2018.

[6] Atlassian Support. Burndown chart, Aug 2018.

[7] Poker VR. Poker vr - multiplayer poker in virtual reality.

[8] Hui Zhang. Head-mounted display-based intuitive virtual reality training system for the mining industry. *International Journal of Mining Science and Technology*, 27, 05 2017.

BIOGRAPHICAL INFORMATION

Allie Gardiner is graduating from the University of Texas at Arlington (UTA) in December 2021 with an Honors Bachelor of Science in Computer Science. She has focused on theoretical computer science and algorithms, software design and development, and robotics and unmanned systems.

One of the most interesting projects during her degree was creating an app for the use of a local Recreation Center for a church. Another interesting project was programming two robots to go from start to goal while avoiding obstacles – one path planning and one reactive with vision sensors.

Outside of class, she was a pitcher on UTA's softball team, and competed in NCAA D1 and Sun Belt. In 2019, her team won the National Invitational Softball Championship. After graduation, she plans move to Los Angeles, California to start her Master of Business Administration at California State University Northridge and continue her softball career.