VR squash practice

By Noofar Ophir and Dor Brekhman

Supervisors: Boaz Sterenfeld and Yaron Honen

2022
# Table Of Contents

Introduction ........................................................................................................................................... 3  
Game Overview ..................................................................................................................................... 3  
Software Development ......................................................................................................................... 5  
  Unity .................................................................................................................................................... 5  
  C# Scripts .......................................................................................................................................... 5  
  Unity Oculus SDK ............................................................................................................................... 6  
Hardware – Oculus Rift Set .................................................................................................................... 7  
Problems and solutions .......................................................................................................................... 7
Introduction

The goal of the project is to simulate a practice environment for squash players. The scene is a squash court in which the player can grab the ball and hit it with a racket. During the game, the player can practice his squash skills and his counter strikes to a random canon’s serve.

Game Overview

During the game, the player has a racket (which vibrates when hitting the ball) as a right hand, and can grab the ball with his left hand by pressing the L-touch. We also added an option for the player to summon the ball using the left index trigger, and an option for bringing the ball to a specific location, for an easier game experience. In addition, the player can move to different areas of the court using the left controller’s joystick. The game has two modes. The first one is practice mode, in which the player can practice his strikes by playing against himself. The second mode includes playing against a simple bot. In this mode the game flow is according to rules of a real squash game. When it is the player’s turn to start, he can practice his serve by throwing the ball and hit it with the racket in order to hit the walls. When it is the other player’s turn to start, the ball is thrown automatically from a fixed location in a random speed and a random direction from a collection of directions that are aiming for the front wall, like in a correct serve. The player can practice his counter strike to the automatic serve. During the entire game, a score board is presented on the front wall. The score board is updated according to game events and squash rules.
A state machine demonstrating squash rules and game flow:
Software development

Unity:

Unity is a cross-platform game engine developed by Unity Technologies, first announced and released in June 2005 at Apple Worldwide Developers Conference as a Mac OS X game engine. The engine has since been gradually extended to support a variety of desktop, mobile, console and virtual reality platforms.

We used Unity for building the scene of the game with different game objects and materials. All game objects have a Transform component that allowed us to adjust their location and rotation.

Unity Physics:

We used Unity’s physics features for making the game dynamic and for giving the user a real world experience.

For the ball – We made the ball bouncy using physics material, and gave it gravity using Rigidbody component. The Rigidbody also allowed us to give the ball different velocities in different directions for the automatic serve.

For the racket – We made the racket follow the player’s right hand by giving the racket velocity towards the position of the right controller.

Unity Colliders:

We used different collider components for the game objects to collide with each other. Collisions are necessary for the game flow and for real world experience in the game.

Main uses:

- Keep the ball inside the court’s walls. The ball has to hit the different walls and return from them.
- Hit the ball with the racket to move the ball.

C# scripts:

Unity supports the C# programming language natively. We used C# scripts with UnityEngine library, to control the scene dynamics and to manage the game logic.

Main scripts:

- CollisionManager – A script connected to the ball. When the ball collides with another object the script is triggered and outputs the collision kind. Distinguishes between collisions with the racket and different areas on the walls. We used transparent planes to recognize an area on a specific wall according to the side of the plain the ball is in during the collision.
• GameState – The state machine of the game, defining the game states and the transitions between them according to squash rules and game flow. Game state also contains the current turn, and transitions can switch between the players.
• GameManager – Holds the current game state and advances it according to collision events notified by CollisionManager. Also holds the current score.
• Bot – Waits until it is the bot’s turn to serve, then the ball is moved to a specific position and is thrown in a random velocity and a random direction from a collection of serve directions.
• Racket – Makes the racket follow the position of the right controller by changing its speed vector.

Unity Oculus SDK:

We used this SDK for connecting the oculus set to the game in unity and transform the game to a virtual reality game.

Main features we used:

• We used OVRPlayerController which belongs to this SDK, to control and manage the position and movement of the oculus set in the scene. This controller contains OVRCameraRig which we used instead of the regular unity’s camera so that the player can watch the scene while wearing the oculus headset and feel like they are inside it. The camera is connected to the headset and moves with it, therefore the part of the scene the player is watching is according to the position of the player in the scene and the direction he is looking at. The controller also allowed us to move the player (and the camera with it) around the scene using the controllers’ joystick. In addition, we tracked the positions of the controllers using LeftHandAnchor and RightHandAnchor objects (children of the camera).
• We used OVRInput to recognize presses on the different controllers’ buttons. For example, to summon the ball.
• We used OVRGrabber component for the ball and LeftHandAnchor to implement the grabbing of the ball.
• We used OVRHaptic to implement vibration of the right controller when the racket hits the ball.
• We used a left hand avatar and connected it to the left controller’s position in the scene.
**Hardware – Oculus Rift Set**

Oculus Rift is a discontinued line of virtual reality headsets developed and manufactured by Oculus VR, a division of Meta Platforms, released on March 28, 2016.

We used Oculus Rift for the VR experience of the game.

The components of the set we used:

- **Headset** – Allows the player to watch the scene and feel like they are inside it. Connects to the camera and moves it accordingly.
- **Sensors** – Track the position and movement of the other connected components in the real world space. Enable calibration of the field of view.
- **Left controller** – Appears as hand avatar in the scene. We used its grip trigger for grabbing the ball, its index trigger for summoning the ball and its joystick for moving around the scene.
- **Right controller** – We attached the racket to its position and moving it moves the racket accordingly.

**Problems and Solutions**

- At first, we worked on a project with capto gloves. During the research stage we found that there is almost no documentation for those gloves and the SDK we found for working with them was very poorly written. We also found that there is no vibration feature in the gloves. This feature was critical for the project, and so we decided to start from scratch and began working on a different project, a VR squash game with an Oculus set.
- We started working on the project with Oculus Quest 2 instead of Rift and had problems with connecting the Quest 2 to the computer. We tried different settings and different computers, but every time the computer did not notice the connection. In the end, we switched to Oculus Rift and then the connection worked perfectly using cables and the Oculus application on the computer.
- When we first connected the racket to the right controller, we just matched the racket’s position to the controller’s position. That made the racket’s movement discrete. We found that the discrete motion damaged the racket’s ability to collide with the ball because the racket could easily move past the ball without hitting it. In order to make the racket’s movement continuous, we gave it velocity in the direction of the controller so the racket still followed the controller, but moved continuously.