VR Racing Game

Sophia Chen
Stanford University
schen10@stanford.edu

Suzannah Osekowsky
Stanford University
sosekows@stanford.edu

Abstract

This report describes a Unity virtual reality driving game with two cars. One is controlled by a first-person user and the other is controlled by an automated program. The user can look around the scene and drive using either the arrow keys or a VRduino steering wheel.

1. Introduction

The purpose of this paper is to describe a virtual reality car racing game designed in Unity for use on a head-mounted display (HMD), with optional VRduino inputs. The goal of the game is to complete a lap and set the best time. The game was designed with help from Jimmy Vegas's Unity tutorial [1]. We read inputs from the VRduino by modifying scripts from the sample project [2], and used them to control the head orientation and driving controls. To make things harder, an autonomous car is also racing.

2. Using the Game

The game begins with a countdown at the starting line. If head orientation tracking is enabled, the car may not be directly in the field of view depending on the position the head orientation VRduino is initialized and potential drift from bias over time. You may need to turn around to find it. For an example of what the scene should look like, see Figure 1. The object of the game is to drive a full loop of the track in the car. The faster the better!

2.1. View rotation

The program simulates a driving experience by matching camera movement to car and head movement in a natural way. The "forward" direction tracks the car's forward direction, but with view tracking enabled, the user can look around the scene as though they are sitting on the car. This allows the player to remain comfortably oriented while the car drives on uneven terrain.

2.2. Driving

Action	VRduino	Arrow Keys
Turn Left	Turn CCW	←
Turn Right	Turn CW	→
Accelerate Forward	Tip Forward	↑
Accelerate Backward	Tip Backward	↓

Table 1: Driving controls in each mode

The car can be controlled in two ways according to player preference or computational availability. The commands are summarized in Table 1. In VRduino driving mode, the VRduino controller behaves as a steering wheel. In computer driving mode, the arrow keys behave as one would expect.

2.3. Restarting the Game

Once you reach the end of the lap, the game will end and there will be a celebratory cut scene! If you beat the best time, the best time will be updated with your time. To restart the game, press the 'R' key.

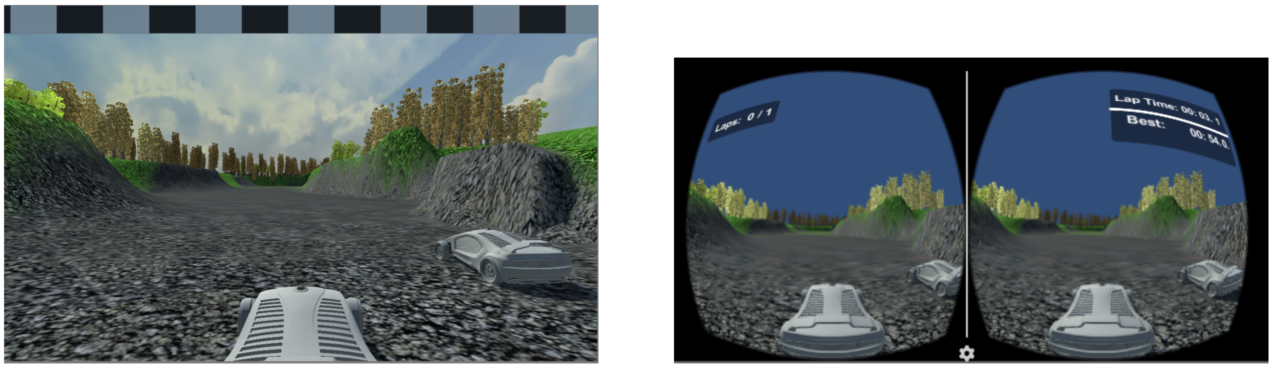
3. How it Works

3.1. Camera Movement

There are two important contributors to natural camera movement: translation with the car and rotation with the player's head.

3.1.1 Translation: Cube Tracking

The main camera rotates with the car's yaw rotation. However, the main camera does not rotate in the pitch or roll direction with the car. This is in order to maintain steadiness in the case where the user hits a wall and the car flips over. In that case, the camera should remain steady, and the user will only see the car flip over instead of flipping with the car. This is done by tracking a cube that sits on top of the car, and only rotates around the yaw axis.



(a) Normal view of the game (b) Stereo rendering for VR headset.

Figure 1: The starting line. Your opponent is usually faster than you.

3.1.2 Rotation: VRduino Tracking

The camera uses the rotation quaternion from the VRduino's serial port as defined in ReadUSB2.cs. However, if we used this rotation quaternion alone, it would track the car's location but its default "forward" direction would not rotate with the car. We therefore combined the Euler angles from the car quaternion and the VRduino quaternion to produce the camera quaternion. This causes the view to rotate when the car turns, following the driving naturally.

3.2. Driving

In VRduino driving mode, we collect the steering VRduino quaternion from the serial port. We then back-calculate the Euler pitch and roll and use these angles as inputs to the native Unity driving behavior. As the Euler angles using the Unity command quaternion.eulerAngles are given in the range $[0, 2\pi]$, we performed this calculation ourselves so we would have the desired range of $[-\pi, \pi]$.

4. Changing Modes

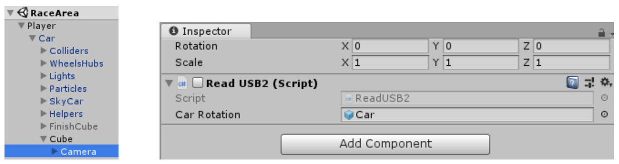
Playing the game with both visual VRduino control and VRduino steering can be laggy, so here are instructions for changing between options and experimenting with behaviors.

4.1. Head Tracking

To activate/deactivate orientation tracking, select the Camera object in the RaceArea scene. It has a few parents in its hierarchy, so use the image shown below to find it. Then, go to the Inspector window, scroll all the way down, and check/uncheck ReadUSB2 (as shown in Figure 2).

4.2. Car Movement

To switch between the two input methods (arrow keys and VRduino), select the "Car" child of the "Player" object of the RaceArea scene. Then, go to the Inspector window and scroll down until you see the attached scripts. The configuration shown here allows you to navigate using the arrow keys as input. If you would like to drive with the VRduino, uncheck "Car User Control" and check "Read USB" (Figure 3).

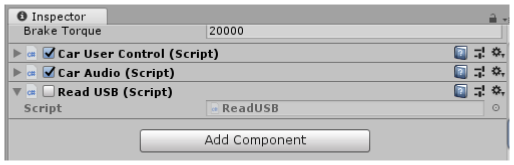


```
RaceArea
  Player
    Car
      Colliders
      WheelsHubs
      Lights
      Particles
      SkyCar
      Helpers
      FinishCube
      Cube
        Camera
```

```
Inspector
Rotation        X 0   Y 0   Z 0
Scale           X 1   Y 1   Z 1
[ ] Read USB2 (Script)
Script          ReadUSB2
Car Rotation    Car
```

(a) Camera parentage (b) Check "ReadUSB2" to activate head tracking.

Figure 2: Controlling head tracking



```
Inspector
Brake Torque    20000
[✓] Car User Control (Script)
[✓] Car Audio (Script)
[ ] Read USB (Script)
Script          ReadUSB
```

Figure 3: Check "ReadUSB" to activate VRduino steering. Check "Car User Control" to activate arrow key steering. For whichever one you select, you must unselect the other.

References

[1] J. Vegas, "How to make a driving racing game in unity." Youtube. `https://www.youtube.com/watch?v=MQ5GJPlAGS4`, March 2017.

[2] M. K. S. J. Varsha Sankar, Sagar Honnungar, "Vrduino fruit ninja," 2018.