How humans perceive multi-input controls in 3D scenes, a case comparison study

Ţocu Nicolae-Adrian[1], Govoreanu Valentin-Cătălin[1], Olescu Marco Leon[1], Niţescu Teodor Marian[2]

[1] Connected Intelligence Research Center
Lucian Blaga University of Sibiu
10 Victoriei Boulevard, Sibiu, Romania
{nicolae.tocu, valentin.govoreanu, marco.olescu}@ulbsibiu.ro

[2] Computer Generated Solutions
Romania Branch
319 Independentei Splai, Bucharest, Romania
teodor.nitescu@gmail.com

ABSTRACT

This paper presents an input device comparison for a race car simulator in Virtual Reality (VR). The study focuses on the usability of the following devices: Leap Motion , Manus VR, smartphone, keyboard, Xbox Controller and Vive Controllers. Each device is evaluated individually in order to see how the user interacts with them and to see if one of these is more prefered. The presented development details are followed by an experimental feedback session received from the testing group. The application uses Unity 3D to generate the Virtual Environment as it was the best suited choice based on its multiple functionalities.

Author Keywords

Virtual Reality; Manus VR; Leap Motion; HTC Vive; Xbox One Controller; Smartphone.

ACM Classification Keywords

H.5.1 Multimedia Information Systems: Artificial, augmented, and virtual realities.

H.5.2 User Interfaces: Input devices and strategies, Interaction styles, Prototyping.

INTRODUCTION

The test environment chosen for this research was Unity 3D, which gives good interaction with a series of devices. By this manner it is easy to make a comparison between all the controlling devices analyzing their differences and usabilities in certain scenarios.

The game that was chosen to be implemented was a car race, where the user attempts to reach the finish line as fast as possible. To get a fast clear time, the user must avoid car crashes and brakes. This type of application was chosen because car races are common game types and most of people have played it at least once.

The aim was to find out how the users perceive the task to finish a game while using different controllers with different interaction mechanics. Six devices were chosen, ranging from simple and common ones (smartphone, keyboard, console controllers), where the users already had previous experience, to more complicated and untested devices (HTC Vive, Leap Motion, Manus VR).

When talking about the interaction with VR environment and controllers [6], users tend to perform worse when the physical feedback received from a device does not exist. On the other side, previous experience plays an important role when it comes to performance. When bringing classical or known controls to a VR environment, the feedback is less important because users already have existing habits when it comes to the correlation between actions in the real world and how these transcribe in the digital world.

The application consists of six scenes. In the first scene, the user inputs his name and selects the controller. The remaining five have the same environment and the same rules, the only difference is the input through which the car is controlled. The car can be controlled using:

- Standard keyboard: controlling the virtual car can be made using the W, A, S, D keys or the arrow keys.
- Leap Motion: users can interact with virtual objects, like steering wheel, as it detects hand movements;
- Manus VR: physical gloves synchronized with virtual hands to interact with objects.
- A smartphone: a Google Pixel XL device to control the virtual car using the motion sensors.
- Xbox One Controller: a wireless controller for the Microsoft console Xbox One with analogic controls
- HTC Vive Controllers: special controllers for interaction with the virtual world. Both these controllers and the

Head Mounted Display (HMD) are tracked by two separate sensors called base stations.

The track has four interest points: Start, two Checkpoints and Finish. When the car passes the *start* line, the race begins and the timer starts. The car will pass through two *checkpoints* along the track, auto-saving your car's position. If you leave the track it is considered as a crash and you restart at the latest interest point. The timer doesn't reset and the mistake counter is incremented. The race ends when you reach the *finish* line. At this interest point, the timer stops and the results are saved in a file. These results consist of each user's name, time, number of mistakes and number of brakes.

For the track, the following metrics for each individual device will be analyzed: number of crashes, number of brakes used and the time it takes to finish the race.

ENVIRONMENT

For this application, the following 3D models were imported for the environment:

- Lake Race Track [2]: is a field with several 3D objects that form the race track (the track, the lake, the bridge and other environment-related objects).
- Aston Martin DBS Volante 2010 [1]: is a 3D model of the car, which contains several sub-elements, like the wheels, the steering wheel, the body etc.

Main Menu

In this first scene, the user must type in his name and select the car controlling system from the available list of controllers. The username and the scene name are saved in two variables that would be later used in the final scene to record each player's progress.

Collision system

Figure 1 presents the collision system as highlighted which is composed of two colliders: a squared one for the car and a mesh one for the track.

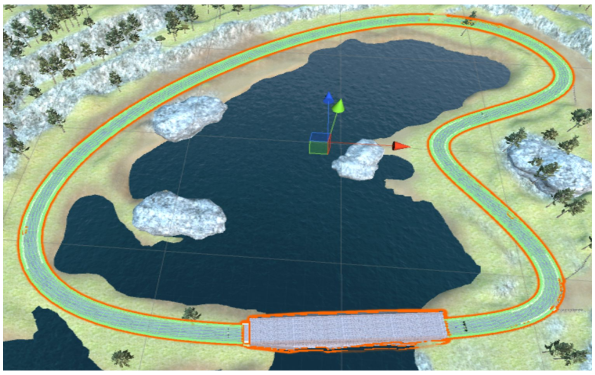

Figure 1. Track Mesh Collider

A car crash is considered when the car oversteps the mesh collider of the track. When it occurs, the mistake counter is incremented and the scene will be restored to the last interest point crossed without resetting the timer.

Interest Point: Start

When the car passes through this interest point (see Figure 2) for the first time, the timer starts and the initial position of the car is saved. If the user crashed the car until it got to the next checkpoint, the car's position resets and spawns to the start point, on the initial position. The timer does not reset.

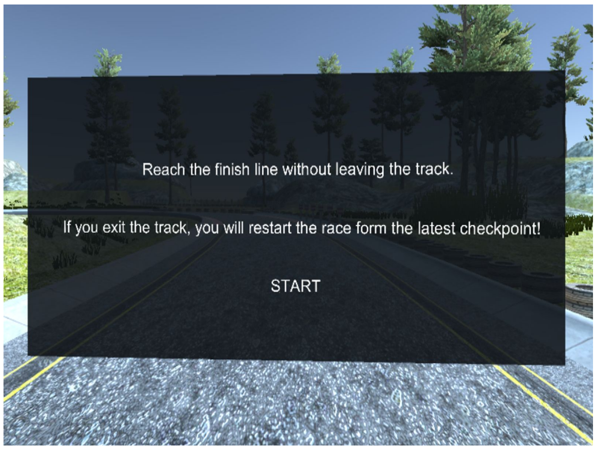

Figure 2. Interest Point - Start

Interest Point: Checkpoint

Once the car reaches a checkpoint, as shown in Figure 3, the position and rotation of the car are temporarily saved. If a crash occurs, the car restarts at the checkpoint using the new data saved. Along the track there are two checkpoints set at even distances with no differences in their functions.

Figure 3. Interest Point - Checkpoint

Interest Point: Finish

When the car arrives at the finish line (see Figure 4), the timer stops and the progress is saved in a file. The saved data is: the player's username, the scene's name, the time spent on the track, the number of crashes occurred and the number of brakes used.

Figure 4. Interest Point - Finish

DEVICES

This interactive game was developed to study the differences between several devices that can be used to interact with VR objects and how humans perceive different multi-input controls. Multiple controllers were chosen, since every single device interacts differently with the environment. Because the aim was to analyze each user's behaviour and how it changes based on the way they control the car, simple and common devices were implemented (such as a simple keyboard, a smartphone or console controller), but also devices and technologies that are emerging recently (Manus VR, Leap Motion etc.). By giving users these new devices, where they have no prior experience with them, it became possible to determine whether or not they can adapt fast enough and have similar performance with the common ways of controlling a car. Skalski et al. [5] showed that different types of interactions can create different levels of comfort and therefore giving the users a certain level of enjoyment. Uncommon mapping styles can make an environment difficult and can hinder the performance level and the desire to acquire distinct knowledge (especially for the controllers that are harder to be manipulated).

Leap Motion

This peripheral USB type device can detect hand movement using two cameras and three infrared LEDs. It can be positioned on a desk or can be mounted on a VR headset.

This device correlates hand movements with the key interactive part of the car, the steering wheel, and also accomplishes the start, accelerate and break functions. To accelerate and stop the car two specific hand gestures were used (see Figure 5 and Figure 6).

To recognise these gestures a functionality from Leap Motion called **detectors** was leveraged. A detector is an object which observes an aspect of a scene and returns "true" when the specified conditions are met. These conditions can be related to palm and finger direction, or finger extension.

Leap Motion provides several types of detectors, each having the task to recognize a basic gesture:

- Extended Finger Detector - detects if the specified fingers are extended or not. It activates when each finger from the detected hand meets certain criteria.
- Finger Direction Detector – detects when the fingers are pointed towards the desired direction.
- Palm Direction Detector – detects if the palm indicates the specified direction.
- Detector Logic Gate – observes the other detectors and triggers when all of them meet the conditions.

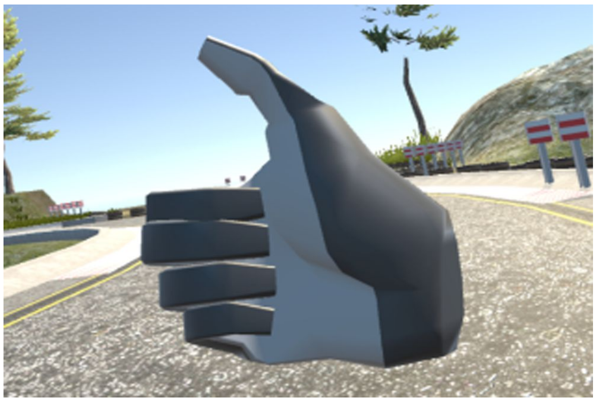

Figure 5. Start/Accelerate Gesture

Figure 6. Decelerate/ Stop Gesture

A flaw was discovered while using Leap Motion as a controller: your hands must remain at all times in your visual field, otherwise you cannot control the car. Therefore, additional gesture recognition was implemented to balance this.

Manus VR

These are haptic gloves which contain several sensors like: magnetometer, accelerometer and gyroscope. These sensors have the capability of measuring the glove's orientation. The motion of the fingers is tracked using two sensors and the entire data is sent through a wireless transmitter. These gloves can give haptic feedback with the help of a vibration motor when the user is interacting with virtual objects. Manus VR doesn't have sensors for positional tracking and for this the HTC Vive Trackers were considered to be used.

Manus VR allows the user to grab the virtual steering wheel and control it like a real one. To accelerate and break there is a lever which has values from 0 to 1, where 0 means that the car stops and 1 means the car is moving with 100% of the maximum speed. If the user grabs the lever and sets the speed to maximum, the speed will increase gradually. The same rule goes from 100% to 0%, the speed is decreased gradually.

The lever function cannot be used until the user starts the engine. For this, he must press a virtual button as you can see in Figure 7.

Figure 7. Car Steering Wheel, Lever and Start Engine Button

Smartphone

For our application the Google Pixel XL was used, along with a different application made in Unity for smartphone. This application connects with our main application with the help of a service from Unity called Multiplayer. This service sends data via network to the car scripts. This data represents phone orientation, because with the help of the accelerometer one can detect the movement of the phone. The guiding system of the car uses the following four orientations: forward to accelerate, backwards to break, left or right to turn the car (see Figure 8) . To start the engine, you must tap on the phone screen. Without starting the engine, the car will not accelerate.

Figure 8. Visual feedback in the main application

Because the user doesn't have any visual feedback from the phone and he does not know at all times what is the direction of the phone, a visual system to help them better adapt to the environment was implemented (see Figure 9). The car's steering wheel has four arrows that turn green to indicate the current direction received from the device.

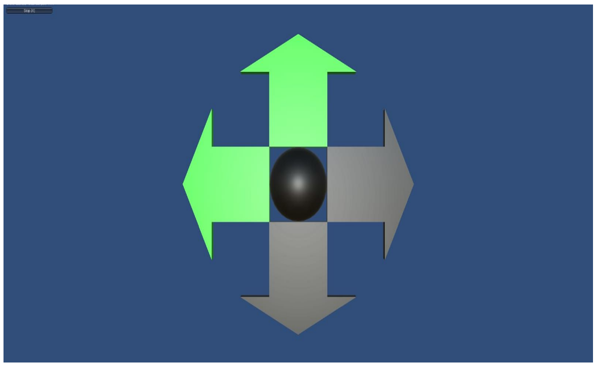

Figure 9. Smartphone application

Keyboard

A standard keyboard was used. As input buttons, the W, A, S, D keyboard keys were the best choice, as these were the most intuitive because they are used a lot in gaming scenes/environments. A secondary set of control keys was also implemented: the arrow keys. As expected, the W key (or the Up arrow) accelerates the car, the S key (or the Down arrow) initiates the break, the A key (or the Left arrow) turns the car to the left and the D key (or the Right arrow) turns the car to the right. The car's engine is started the first time the user hits the W key (or Up arrow).

The same lack of visual feedback was encountered that was encountered while using a smartphone as a controller. As a solution, a set of signals that are providing visual feedback to the user were implemented. These signals correlate with the car's movement. The signals are: accelerating turns a green circle with a forward arrow on the car's board; breaking turns a green circle with a backwards arrow on the car's board; turning left or right also turns the steering wheel left or right.

Xbox Controller

The wireless Xbox controller (see Figure 10) used has offset analog sticks which provide a better user experience when driving the car. The right trigger button starts and accelerates the car while the left trigger button is used to initiate a break. The remaining buttons not listed do not accomplish any function and do not interact with the game.

Due to it's convenient and easy-to-use design, no visual feedback aid was implemented to help the users. Unity Input Manager allowed us to sync the controller easily.

Figure 10. Xbox One

Vive Controllers

The movement of these controllers (see Figure 11) is tracked by two base stations. With these controllers the interaction with the virtual environment is a lot easier because they have several buttons that can be used to do different actions.

The Grip button was used to grab the virtual steering wheel for driving the car and the lever to accelerate or to stop. To start the engine the user must push the start button with the virtual controller. The guiding system was the same with the Manus VR.

Figure 11. HTC Vive Controllers

PROCESSING DATA

For understanding how humans can interact with different controllers used in a VR environment, a comparison was made between several devices taking into account these metrics: how fast the user can finish the race, how many times he uses brakes and how many crashes he produces along the race (see Table 1).

To gather this data, a group test consisting of 15 people with ages between 15 and 30 was selected. Some of them never experienced any virtual reality while others had previous experience with both the environment and the chosen controllers.

Devices	Average time	Total crashes	Total brakes
Keyboard	01:13	19	26
Smartphone	01:45	31	134
Leap Motion	02:30	60	22
Manus VR	02:34	40	38
Xbox Controller	01:35	24	42
Vive Controllers	02:08	35	27

Table 1. Data from all devices

Looking over the above data, we can conclude that most users have preferences, some obtain good results on certain devices, while others are having a hard time figuring out how to use them. The first result was that common devices (such as the keyboard, the smartphone and console controllers), with very good scores while being the most preferred by users because they interact with them frequently. When talking about handling a new device that a user may have never used (such as Leap Motion, Manus VR and HTC Vive), it was taken into account that during the first attempts he has to accommodate first and only after that they perform the task. Therefore, as expected, users will tend to prefer the trusted devices (see Figure 12).

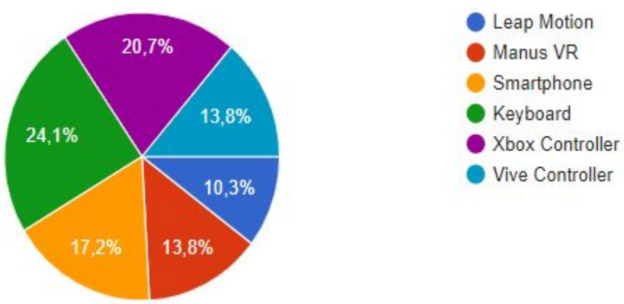

Figure 12. Favorite device

CONCLUSION

Having multiple devices with different ways of interacting with 3D environments is a good way to understand human perception diversity and to create different workflows for different people. While matters of habit and subjectivity can interfere with the tests, certain results can be depicted from having all these controls tested. According to data, the keyboard is better than all devices because it is used by everyone and it has simple intuitive moves. Also, users gave the best feedback to the smartphone, saying that the way the movement interacted with the environment was very natural, as if they were holding a real steering wheel. While considering the other technologies, most users complained that these were new devices for them and they had to get accustomed to them. Also, the way the devices gave feedback back to the user was counter-intuitive and there was no physical feedback to correlate with their visual feedback. Leap motion had the slowest finishing time. Due to a lack of haptic feedback, the users could not get accustomed to it. The high number of crashes was a direct result of the hands leaving the visual field when the user instinctively tried to avoid the crash by moving their hands too much. Users who tested Manus VR had difficulties interacting with the virtual objects themselves. The steering wheel was too attached to the virtual hand and it was difficult to release the object. Nonetheless, being able to interact with the environment with bare hands gave the users the feeling they were more in control and made for a better experience. Also, the haptic feedback through gloves vibration helped a lot with the control of the car, resulting in a smaller number of brakes and crashes. Xbox Controller had a good score because almost all users had already interacted with them and the control of the car was pretty simple and intuitive. Even though the Vive Controller obtained the same score as Manus VR, from the user's perspective, the Vive Controller was better because of its superior human computer interaction method. The users had a pleasant experience in the virtual environment with the help of these controllers. The steering wheel and the lever were easy to interact with, the tracking was very good and they didn't had unexpected problems.

The results of this case comparison study are considered relevant in the context of developing cyber-physical-social systems because it reveals user preferences and user adaptability to new devices. The current results will be used to conduct future research into understanding the psychological impact of immersive technologies. The end purpose is to build a gradual roadmap that goes from classic peripherals to special devices in order to accommodate the users with an agnostic way of interacting with the machine.

ACKNOWLEDGMENT

This work is supported through the DiFiCIL project (contract no. 69/08.09.2016, ID P_37_771, web: http://dificil.grants.ulbsibiu.ro), co-funded by ERDF through the Competitiveness Operational Programme 2014-2020.

REFERENCES

1. Aston Martin DBS Volante 2010. Available: https://www.cgtrader.com/free-3d-models/car/sport/aston-martin-dbs-volante-2010--3.
2. Lake Trace Track. Available: http://blog.niandrei.com/2016/02/19/lake-race-track.
3. Leap Motion. Available: https://www.leapmotion.com.
4. Manus VR. Available: https://manus-vr.com.
5. Skalski, P., Tamborini, R., Shelton, A., Buncher, M., and Lindmark, P. Mapping the road to fun: Natural video game controllers, presence, and game enjoyment, *New Media & Society*, 13(2), 224-242, 2011.
6. Lindeman, R., Sibert, J., and Hahn, J. Towards Usable VR: An Empirical Study of User Interfaces for Immersive Virtual Environments, *CHI'99 Proceedings of the SIGCHI conference on Human Factor in Computing Systems*, 64-71, 1999.