Virtual Reality Game Development with External Hand Motion Sensor

Erik Dobra
Alba Regia Technical Faculty
Óbuda University
Székesfehérvár, Hungary
erik.dobra@stud.uni-obuda.hu

Éva Hajnal
Alba Regia Technical Faculty
Óbuda University
Székesfehérvár, Hungary
hajnal.eva@amk.uni-obuda.hu

Abstract— **In this paper the addition of a hand recognition device to VR glasses was demonstrated. In the investigation a leap motion sensor was added to a pair of Windows Mixed Reality VR glasses, which allows hand tracking without a hand controller. The implemented solution was tested in a sample application. The theme of the game is horror, which encourages the player to react quickly. Three alternative implementation of moving in virtual space in the application was tested. The development of the control unit redesigned for virtual space was done in a Unity based project. The system allows players to perform specific gestures using the thumb and index finger, specifically to press buttons. The software using the built-in profiling system was evaluated, which shows what future optimizations are possible within the project. In summary, we have succeeded in creating a hand recognition system that tracks accurate hand movements in real time, which is a significant improvement in hand recognition implementation compared to the driver produced for the system.**

Keywords— VR, Virtual Reality, Hand-Tracking, Leap Motion, Windows Mixed Reality, Unity.

I. INTRODUCTION

Virtual reality is an environment simulated by a computer that you can interact with.

VR replaces reality with an imaginary world that can be seen through a special head-mounted device (HMD). In this environment, hand tracking is an important task [1-10]. Many devices do not have this functionality, or it is not sufficiently accurate for an industrial application. It is common that hand motion tracking is designed to follow a special hand controller, whereas many applications would require the use of someone's naked hand. Our goal was to create a VR game that is both fun to use and opened up a new way for users to use handheld interactions and in-game movement within the game. It is a very important question, because VR sector is currently at a turning point. Developers are now having to adapt to the hardware, and in the same way, the hardware is still evolving. Sometimes it is very straightforward to look at the problem from a developer's point of view: how can we create a software that meets the needs using specific hardware, how difficult is it, how can it be done. The basic idea with VR is to translate as many sensory experiences as possible into virtual space and to make them immersive for the user. Bringing our limbs into the virtual space is a milestone to this long way.

Our target is not only the entertainment industry, but can also benefit many other industries in the field of simulation [9]. We know of applications in the hazardous or heavy industry from our present studies. There is a wish that our system could be used for other industries daily practice later on.

Also, there is an increased tendency for people to stay more and more indoors (specifically because of the last years' health situation) and consequently people are looking for ways to break away from everyday reality, so there is an increased willingness to connect to virtual reality. The research questions are the ways of hand control implementation methods, their usefulness from the aspect of the user and from the aspect of system's performance. The structure of this paper is the next: section II. and III. contain applied hardware and software solutions, section IV. deals with the investigation of the usefulness of the introduced solutions and chapter V. contains the Conclusion.

II. HARDWARE IMPLEMENTATION AND TEST ENVIRONMENT

By our plan a Leap Motion control unit was required physically connecting to the HMD head unit. Based on the manufacturer's recommendation, the Leap Motion was mounted in the middle of the front of the HMD. For this purpose, a special adapter was designed. According to the manufacturer's website [1] it was modelled in Fusion 360 [2]. A frame around the model was drawn and then cut out the port location (Figure 1). Along the frame, two bumps on each side were created, and a support element to fit the curve of the VR Headset, which fits at the points of the bumps to ensure it is held in place.

Design was created in Ultimaker Cura [3], which implements the printing mesh for the model. The elements were then 3D printed (Fig. 2) [4-8].

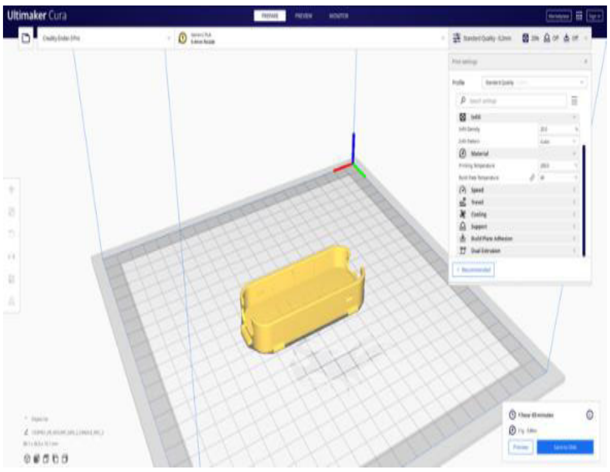


Figure 1: Modelling in Ultimaker Cura software.

Figure 3: 3D Printed Leap Motion holder from different angles.

After that we had to plan of use cases, which VR gestures could be implemented if You use your own hands in the virtual space. First is to direct the movements in the labyrinth. There were three implementations of player movement that were examined.

The second is the fine handling of objects in the VR space.

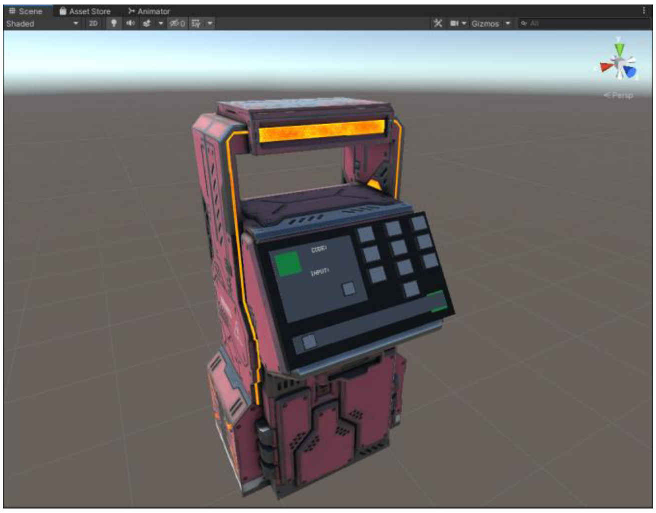


Figure 4: Interactive buttons within the game.

An ATM-like model was created with different operating controls on them (Fig. 3). The question is whether fingers can be recognized and interacted with. First implementations were buttons, next to the buttons, sliding cubes were also placed that need to be dragged to continue the game.

There is also a life level indicator system (Fig 4.) that monitors the rotation of the carpal bones on the left arm and appears when you turn your palm towards you. The player has three lives, and this is presented in VR.

Figure 2: Life indicator fitted to wrist movement.

III. Hand movement recognition

A. *Hand-Bone technique*

It is based on the recognition of the rotation of the hand.

The most important of these is the calculation of rotation vectors - Euler angles or quaternions.

The Euler angle theorem is useful and important because it describes the rotation of a body with help of three angles. Since three perpendicular coordinate axes can be distinguished in space (X, Y, Z), three separate rotations can be performed about each axis. An acceptable alternative is to use quaternions, which allow us to perform any rotation, regardless of orientation, with a given angle and rotation value given by the vector. The first solution was preferred, as its calculation is more complex, but Unity Engine has a service for this calculation.

This principle could be used to control our character, and knowing that the hand recognition implementation was successful, we dared to think boldly and implement the movement of the planes to control the movement manually.

As we know, it can rotate on three axes, which helps it to move in space. Yaw, Pitch, Roll, i.e. lateral, vertical and longitudinal rotation. If we put our hands out in front of us, we can rotate our wrists in these three planes, because wrist movement is a movement with 3 degrees of freedom. If we map this logic into the above-mentioned hand recognition sensor, we can not only in principle, but also in practice, make the movement work with "gestures" in the absolute space

B. *Magicsee R1 controller*

The first, in which we used a tiny Magicsee R1 joystick ring [11-13]. This ring has an analogue lever that can be hidden on the inside or outside of the hand, which doesn't interfere with hand recognition, so it retains the advantage of both worlds, which is its own specialty. Fine hand movement can still be used to solve in-game tasks, but movement is handled by the joystick and headset.

This is a safe approach to the problem that will ensure proper, guaranteed operation.

C. *Virtual VR keystrokes with hand recognition*

Another, more daring approach, which is not provided by the vendors, is to push the boundaries of the framework with our own implementation. As some elements are not implemented in the Unity development environment, creative solutions were introduced in one or two cases for direct the movements in the labyrinth.

Figure 5: Activating hand gesture illustration.

Here's how it works:

If the player extends the thumb and index finger (forming a pistol) with both hands, a script is activated, which results in a virtual keystroke (in my case "W") (Figure 5).

A keystroke will force the virtual keyboard to move forward, and then you can use the north point of the headset to move forward. If one of your hands is no longer showing the gesture, the forward progress will stop.

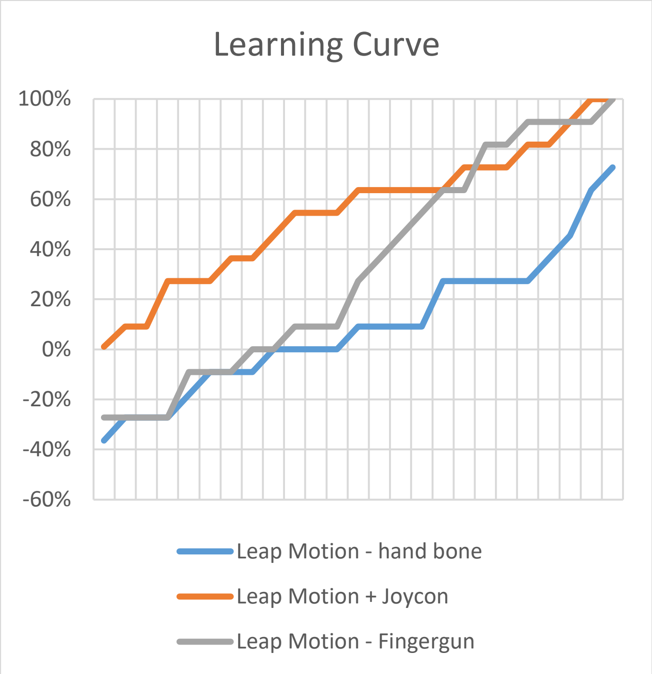
It is important to note, once again, that this tricky solution is not native, it is not feasible by default in the current development environment, but it is representative of what this build would be capable of. For me, this was important to point out because it was conceptually suitable, but a solution that could affect the stability of the computer would not be considered as final (in my case, virtual keystrokes meant that even if I minimized the program, it still monitored the hand movements and guaranteed keystrokes outside the program) [14-16].

IV. Comparative Analysis of Motion Control Methods

Learning curves of the above-mentioned three solutions were analyzed in a non-representative investigation. The implemented three different locomotion technologies were compared by handling speed using the timer implemented in the game. Each successful run was recorded, measured and saved in a table, rounded to minutes for the simplicity. The 25 runs of each type were normalized with eq. 1 to show the performance that can be achieved with these three types of movement methodology.

TABLE I. Results of the investigation of three implemented methodology

Leap Motion Fingergun (C method)		Leap Motion hand bone (A method)		Leap Motion Joycon (B method)	
z_i	F3 /min	z_i	F2 /min	z_i	F1 / min
-27%	19	-36%	20	1%	16
-27%	19	-27%	19	9%	15
-27%	19	-27%	19	9%	15
-27%	19	-27%	19	27%	13
-9%	17	-18%	18	27%	13
-9%	17	-9%	17	27%	13
-9%	17	-9%	17	36%	12
0%	16	-9%	17	36%	12
0%	16	0%	16	45%	11
9%	15	0%	16	55%	10
9%	15	0%	16	55%	10
9%	15	0%	16	55%	10
27%	13	9%	15	64%	9
36%	12	9%	15	64%	9
45%	11	9%	15	64%	9
55%	10	9%	15	64%	9
64%	9	27%	13	64%	9
64%	9	27%	13	73%	8
82%	7	27%	13	73%	8
82%	7	27%	13	73%	8
91%	6	27%	13	82%	7
91%	6	36%	12	82%	7
91%	6	45%	11	91%	6
91%	6	64%	9	100%	5
100%	5	73%	8	100%	5



6. Figure: Learning rates calculated by eq.1

The following is the formula for the calculation:

$$z_i = -\frac{x_i - x(max_n)}{x(max_n) - x(min_n)} \qquad (1)$$

where the xi is the actual measured value, n is the set with the smallest interval and the smallest variance (this is F1 in this case)

In each case, the test was executed 25 times to see the potential for improvement. The best results were 5 minute with fingergun and joycon solutions. The learning curve is better at joycon solution (Fig. 6.). As the figure above shows, the most difficult movement technique to learn was the wrist movement based one, which made it even more difficult to learn due to the absolut orientation (lack of dynamic north point selection mentioned above). The joycon solution is the most effective, but it still does not give up the idea of having empty hands, so the golden mean is the "fingergun" method, which is more difficult to learn than joycon, but guarantees similar results (Fig 6.).

V. Conclusion

Our goal was to solve the problem of using hands in the VR environment, replacing handheld controllers. My choice of topic was motivated by the fact that there are many applications where this might be necessary. This was supported by the literature research, but it is not solved in many HMD devices. Where it is solved, the accuracy of hand tracking and gesture recognition typically leaves something to be desired. As part of this exercise, several devices were tested and selected a hand recognition device, the Leap Motion sensor, to pair with my existing VR glasses. To fit it together, a 3D printed mounting element was made to secure the device. We reviewed the literature to see what options were available if we could incorporate the hand recognition device for in-game movement and interaction. Then a virtual console was implemented that could be controlled by buttons and a special display that worked based on gesture evaluation. We implemented 2+1 separate types of solutions for controlling spatial movement and compared their usability. I created a game that randomly generated a maze and included all the features. Based on my tests, I was able to implement a low-latency hand recognition algorithm. As an extension, I proposed additional new program features that could have specific behaviors. We hope that our investigations will be useful in VR practice, especially in the industry.

Acknowledgment

The paper was supported by HHTDK-21 grant.

References

[1] "Tracking | Leap Motion Controller | Ultraleap." https://www.ultraleap.com/product/leap-motion-controller/ (accessed Nov. 10, 2021).

[2] "Fusion 360 | 3D CAD, CAM, CAE, & PCB Cloud-Based Software | Autodesk." https://www.autodesk.com/products/fusion-360/overview?term=1-YEAR&tab=subscription&plc=F360 (accessed Oct. 02, 2022).

[3] "Ultimaker Cura: Powerful, easy-to-use 3D printing software." https://ultimaker.com/software/ultimaker-cura (accessed Oct. 02, 2022).

[4] D. Holmes et al., Usability and performance of Leap Motion and Oculus Rift for upper arm virtual reality stroke rehabilitation Communication Style and Exercise Compliance in Physiotherapy (CONNECT) View project PhD Motion Tracking in Digital Images View project Usability and performance of Leap Motion and Oculus Rift for upper arm virtual reality stroke rehabilitation. 2016. [Online]. Available: https://www.researchgate.net/publication/312868469

[5] A. G. D. Corrêa, N. R. Kintschner, V. Z. Campos, and S. M. Blascovi-Assis, "Gear VR and leap motion sensor applied in virtual rehabilitation for manual function training∗ an opportunity for home rehabilitation," in PervasiveHealth: Pervasive Computing Technologies for Healthcare, Sep. 2019, pp. 148–151. doi: 10.1145/3364138.3364169.

[6] D. E. Holmes, D. K. Charles, P. J. Morrow, S. McClean, and S. M. McDonough, "Using fitt's law to model arm motion tracked in 3D by a leap motion controller for virtual reality upper arm stroke rehabilitation," in Proceedings - IEEE Symposium on Computer-Based Medical Systems, Aug. 2016, vol. 2016-August, pp. 335–336. doi: 10.1109/CBMS.2016.41.

[8] "Oculus Quest 2 Hand Tracking Has So Much Untapped Potential." https://www.thegamer.com/oculus-quest-2-hand-tracking-has-so-much-untapped-potential/ (accessed Nov. 10, 2021).

[9] F. Nainggolan, B. Siregar, and F. Fahmi, "User Experience in Excavator Simulator using Leap Motion Controller in Virtual Reality Environment," in Journal of Physics: Conference Series, Jul. 2020, vol. 1566, no. 1. doi: 10.1088/1742-6596/1566/1/012093.

[10] F. Fahmi, K. Tanjung, F. Nainggolan, B. Siregar, N. Mubarakah, and M. Zarlis, "Comparison study of user experience between virtual reality controllers, leap motion controllers, and senso glove for anatomy learning systems in a virtual reality environment," in IOP Conference Series: Materials Science and Engineering, May 2020, vol. 851, no. 1. doi: 10.1088/1757-899X/851/1/012024.

[11] Valve Corporation, "Valve Index Controllers: Your hands, in VR," https://www.valvesoftware.com/en/index/controllers, Jun. 28, 2019.

[12] G. 'Pierre-Loup A., "Valve Index controller Extender," https://github.com/ValveSoftware/IndexHardware/blob/master/Controller/index_controller_example.stl, Jun. 25, 2019.

[13] W. 'Mason, "Palmer Luckey Speaks Candidly About Xbox One Controllers Being Bundled with the Rift," https://uploadvr.com/palmer-luckey-speaks-candidly-about-xbox-one-controllers-being-bundled-with-the-rift/, Jun. 12, 2015.

[14] "Quest Gets Hand Interaction SDK & Tracked Keyboard SDK." https://uploadvr.com/quest-interaction-sdk-tracked-keyboard-sdk/ (accessed Apr. 17, 2022).

[15] C. Khundam, V. Vorachart, P. Preeyawongsakul, W. Hosap, and F. Noël, "A comparative study of interaction time and usability of using controllers and hand tracking in virtual reality training," Informatics, vol. 8, no. 3, Sep. 2021, doi: 10.3390/informatics8030060.

[16] "Leap Motion Hand Tracking vs Oculus Quest One vs Vive One | PaleBlue." https://pale.blue/2020/07/09/leap-motion-hand-tracking-vs-oculus-quest-one-vs-vive-one/ (accessed Nov. 10, 2021).