VR game with variable dynamics of movement for assessing the usability of a spherical walk simulator

Daniel Skrobot, s197267@student.pg.edu.pl

Abstract

This article describes the creation of virtual reality game for VR headset and VR CAVE environment. Designed to integrate with a spherical movement simulator. The primary objective was to develop a game that influences player movement in a controlled and user-friendly manner. This game is engineered to affect the player's current motion in hinting approach. By developing this application, we are able to collect valuable data from the VirtuSphere motion system. Collected data will be crucial in conducting future research focused on evaluating the usability of the spherical walking simulator as a method of navigating within the virtual world.

Keywords: Virtusphere, Predicting Player Movement VR, Walk Simulator VR Game

1. Introduction

With ongoing progress in technology, new techniques for creating immersive experiences such as virtual reality are emerging. Progress requires inventing new techniques of communication with virtual environments. One way to communicate with the virtual spaces is through controllers or haptic devices. These tools enable users to interact with the virtual world. However, when we take in consideration locomotion, then usage of such devices may limit user experience and immersion due to lack of actual movement that reflect real life.

Current technology doesn't limit itself to only controllers. Some investors created device that allow a user to move in similar manner to a human being and by utilizing HMD (Head Mounted Display) movement is reflected in the virtual world. These devices allow for omnidirectional movement in any direction. Notable examples include VirtuSpehre and Kat Walk.

A system known as VirtuSphere stands out for being the most user-friendly. That system allow for free movement in any direction. The primary challenge with this technology is ensuring user safety. As users are engaged in virtual reality, their heads are covered by an HMD (Head Mounted Display), limiting their real-world awareness and ability to react to their physical surroundings. Designed for immersive virtual environments, the VirtuSphere is an innovative platform enabling unrestricted, omnidirectional motion. Its design is inspired by a hamster wheel, granting VR users the ability to navigate smoothly in any direction they choose by stepping[1, 2].

Despite many advantages, it has disadvantages. The most important is platform lack of supporting player movement or working against it. So basically, user uses it as a platform that allows him to move. Although missing part such as simulating different terrain, weather condition or special effects that impact user motion. This lack of feedback decreases user's feeling of immersion[3, 4].

The motivation for developing a game to assess the usability and effectiveness of a spherical walk simulator originated from changes made to the sphere's motor system. The concept of the game was to guide user behavior at specific moments with feedback on actual behavior of user movement and new motor system. By gathering and analyzing data collected from gameplay, it allows determining the practicality and immersive experience of a new system.

The VirtuSphere, located in the Immersive 3-D Visualization Laboratory at the Gdańsk University of Technology, was used for the development and testing of the game that influences player movement[5].

2. State of knowledge

In today's technological landscape, there is a growing emphasis on developing systems that replicate movement within virtual reality. The overarching challenge in this domain, however, lies not in their development but more critically in their ability to deliver an authentically immersive experience. Such systems must not only be straightforward for users to engage with, but also need to feel natural and intuitive.

Previous research has primarily focused on developing a new sensory system for omnidirectional platforms Virtusphere, which allows users to move freely in a virtual environment. Current platforms of this type, however, do not provide any feedback to VR applications, and the user's orientation is

determined solely by the VR headset. To enhance this setup, an additional sensory system is needed that would collect data about the user's movement, not just their orientation[6].

Prior studies as well concentrated on integrating sensors with the VirtuSphere, utilizing these sensors and engines to enhance user movement feedback in virtual reality. An application was developed specifically to simulate ascending and descending hills. This application gathered data about the user's movements, which was then relayed to the engines. These engines, in turn, provided responsive feedback to the user, mimicking real-world physical movements in the virtual environment[7].

3. Solution proposal

To the best of the author knowledge, this work introduces virtual reality application that works with new motor system that has been developed in Immersive 3-D Visualization Laboratory at the Gdańsk University of Technology[5]. Basically, the idea of the game was to create an application which integrated with a spherical walking simulator. In this game, changes in the player's walking speed and direction would be deliberately enforced in a controlled manner.

As the execution environment for the project, Unreal Engine 5 was selected. The game was created in UE Blueprints and in C++ language.

The game was based on the platform game genre. The key to succeeding is to overcome the entire map and collect all the necessary points. The map will consist of platforms where we will have limited movement abilities. The game suggests to the player how they should move at a given moment. The game suggest optimal moves, e.g., a turn of 45°, 90°. In the game, there are ramps that will force the player to approach or descend them. The game map contains a place where the player will be exposed to instability such as swaying, a rope bridge or affect of the wind. The game indicate what pace of player movement is optimal at a given moment, e.g., whether they should stop, walk slowly, or run.

The pivotal aspect was the development of a map editor, as we anticipated from our research and testing phase the need to effortlessly experiment with various movement scenarios. The concept involved utilizing a PNG image file, which could be easily edited using various image editing software. This allowed us to change the colors and RGB values that corresponded a specific type of platform and movement.

RGB values corresponding to specific information about platform:

- Red - defines platform type
- Green - defines platform direction and player movement direction
- Blue - specifies player movement type

The value of the **red channel** represents the type of platform in the following way:

- 255 – this value indicates that there is no platform at this location
- 0 – this value means that there is a default platform at this location
- 10 – inclined platform type (upward movement)
- 20 – inclined platform type (downhill movement)
- 30 – right turn platform type
- 40 – left turn platform type
- 50 – swinging platform type

Meanwhile, the value of the **green channel** represents the direction of a given platform:

- 255 – this value indicates that there is no direction at this location
- 0 – this value means that the direction of the platform at this location is forward
- 50 – platform direction to the right
- 250 – platform direction to the left

However, the value of the **blue channel** represents the type of movement on a given platform:

- 255 – no defined movement
- 0 – running

- 125 – marching
- 250 – a special type indicating the player's starting position

This kind of image then is processed by Unreal Engine 5 which reads the map content in PNG format with RGB encoding. The logic for reading the PNG image and interpreting its values is written in C++, enabling precise color mapping to platform properties. The map is generated based on Actor blueprint types, which override the basic version of the class created in C++. These blueprints allow for further customization and the addition of unique features to each platform, such as specific behaviors or interactions with the player.

Figure 1: View on scene from debug camera perspective

In the game, player movement parameters have been implemented, which include identifying the type and direction of their movement, as well as the way these movements are integrated with a specific gaming platform. Key aspects such as the player's movement direction vector, their speed, and rotation are precisely monitored. These parameters, in particular, play an important role as they affect how the player experiences and interprets the virtual world. All this data is collected and processed by the game application.

Additionally, a collision checking system has been implemented in the game, which allows for the precise determination of the player's current position and identification of the platform they are on. This system not only detects the player's contact with various elements of the game environment

but also analyzes the type and properties of the platform on which the player is currently standing.

Parameters originating from the sphere are read by a set of sensors which are embedded in four wheels located centrally under the simulator. These data are transmitted through a server, which is the controller of the entire system. Information transmitted by the server is made available through a specially designed plugin, which allows for the integration of my game with the VirtuSphere.

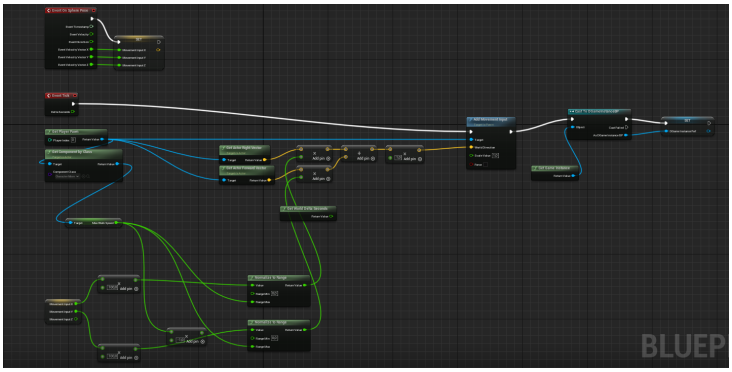

Figure 2: Presents a blueprint responsible for reading data and player movement based on information received from the system operating the sphere

Implemented a logging system in the application that enables the collection of data into a CSV file.

Information contained in each row of the log generated by the application:

- CurrentTime – current game / application time
- PlatformType – type of platform
- PlatformDirection – direction of the platform
- PlatformMovementType – type of movement on the given platform
- PlayerVelocity.x – player's velocity on the X-axis
- PlayerVelocity.y – player's velocity on the Y-axis
- PlayerVelocity.z – player's velocity on the Z-axis
- PlayerPosition.x – player's position on the X-axis in the scene

- PlayerPosition.y – player's position on the Y-axis in the scene
- PlayerPosition.z – player's position on the Z-axis in the scene
- PlayerRotation.Pitch – rotation around the Y-axis of the entire player character
- PlayerRotation.Roll – rotation around the X-axis of the entire player character
- PlayerRotation.Yaw – rotation around the Z-axis of the entire player character
- PlayerCurrentSpeed – scalar speed of the player
- UnixTimeStamp[us] – time from the sphere controller, expressed in microseconds
- ControllerId – controller identifier
- MotorFlags – motor flag
- HallPosition – reading from the Hall effect sensor
- EncoderPosition – encoder position
- MotorVelocity – rotational speed of the motor expressed in revolutions per minute
- MotorVoltage – voltage in the motor expressed in volts
- MotorCurrent – current in the motor expressed in amperes

4. Experiments and testing

Testing was carried out in the VirtuSphere equipped with an upgraded motor system. The initial tests focused on verifying the communication between the application and the server managing this new motor system, as well as ensuring the accurate transmission of data from the controllers and the motor's state to the application.

CurrentTime	PlatformType	PlatformDirectio	PlatformMovementTyp	PlayerPosition.x	PlayerPosition.y	PlayerPosition.z	PlayerRotation.Pitch	PlayerRotation.Roll	PlayerRotation.Yaw	PlayerCurrentSpeed	UnixTimeStamp[us	ControllerI	MotorFlags	HallPosition	EncoderPositio	MotorVelocity	MotorVoltage	MotorCurrent
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320840908000	2	0	-720	-196577	0.0	36.200001	0.000000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320840908000	3	0	560	152890	0.0	36.320000	-0.065000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320840908000	4	0	639	173769	0.0	36.279999	-0.065000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841088000	1	0	-643	-175927	0.0	36.360001	0.000000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320840908000	2	0	-720	-196577	0.0	36.200001	0.000000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320840908000	3	0	560	152890	0.0	36.320000	-0.065000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320840908000	4	0	639	173769	0.0	36.279999	-0.065000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841088000	1	0	-643	-175927	0.0	36.360001	0.000000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320840908000	2	0	-720	-196577	0.0	36.200001	0.000000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320840908000	3	0	560	152890	0.0	36.320000	-0.065000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320840908000	4	0	639	173769	0.0	36.279999	-0.065000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841088000	1	0	-643	-175927	0.0	36.360001	0.000000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320840908000	2	0	-720	-196577	0.0	36.200001	0.000000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320840908000	3	0	560	152890	0.0	36.320000	-0.065000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320840908000	4	0	639	173769	0.0	36.279999	-0.065000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841088000	1	0	-643	-175927	0.0	36.360001	0.000000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320840908000	2	0	-720	-196577	0.0	36.200001	0.000000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320840908000	3	0	560	152890	0.0	36.320000	-0.065000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320840908000	4	0	639	173769	0.0	36.279999	-0.065000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841088000	1	0	-643	-175927	0.0	36.360001	0.000000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841243000	4	0	639	173769	0.0	36.279999	-0.065000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841372000	1	0	-643	-175927	0.0	36.360001	0.000000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841372000	2	0	-720	-196577	0.0	36.200001	0.065000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841372000	3	0	560	152890	0.0	36.320000	-0.032000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841243000	4	0	639	173769	0.0	36.279999	-0.065000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841372000	1	0	-643	-175927	0.0	36.360001	0.000000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841372000	2	0	-720	-196577	0.0	36.200001	0.065000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841372000	3	0	560	152890	0.0	36.320000	-0.032000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841243000	4	0	639	173769	0.0	36.279999	-0.065000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841372000	1	0	-643	-175927	0.0	36.360001	0.000000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841372000	2	0	-720	-196577	0.0	36.200001	0.065000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841372000	3	0	560	152890	0.0	36.320000	-0.032000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841243000	4	0	639	173769	0.0	36.279999	-0.065000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841372000	1	0	-643	-175927	0.0	36.360001	0.000000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841372000	2	0	-720	-196577	0.0	36.200001	0.065000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841372000	3	0	560	152890	0.0	36.320000	-0.032000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841497000	1	0	-643	-175927	0.0	36.340000	-0.032000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841497000	2	0	-720	-196577	0.0	36.200001	-0.032000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841497000	3	0	560	152890	0.0	36.320000	0.032000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841497000	4	0	639	173769	0.0	36.299999	0.000000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841626000	2	0	-720	-196577	0.0	36.200001	-0.065000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841626000	3	0	560	152890	0.0	36.340000	0.000000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841626000	4	0	639	173769	0.0	36.299999	-0.065000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841759000	1	0	-643	-175927	0.0	36.360001	-0.097000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841759000	3	0	560	152890	0.0	36.340000	0.065000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841759000	4	0	639	173769	0.0	36.279999	0.032000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841892000	1	0	-643	-175927	0.0	36.360001	0.000000
2024.01.15-13.14.01	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841892000	2	0	-720	-196577	0.0	36.220001	0.065000
2024.01.15-13.14.02	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841759000	3	0	560	152890	0.0	36.340000	0.065000
2024.01.15-13.14.02	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841759000	4	0	639	173769	0.0	36.279999	0.032000
2024.01.15-13.14.02	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841892000	1	0	-643	-175927	0.0	36.360001	0.000000
2024.01.15-13.14.02	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841892000	2	0	-720	-196577	0.0	36.220001	0.065000
2024.01.15-13.14.02	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841759000	3	0	560	152890	0.0	36.340000	0.065000
2024.01.15-13.14.02	0	0	250	0.0	9900.0	138.9	0.0	0.0	0.0	500.0	1705320841759000	4	0	639	173769	0.0	36.279999	0.032000

Figure 3: Presents logged data from the test of the application and the sphere

However, during these tests, several challenges were encountered. A significant issue was the loss of signal between the Head Mounted Display (HMD) and the sensors tracking the headset. This problem was attributed to the sphere's composition, which is made of a composite material, and a shortage of sensors in the Immersive 3-D Visualization Laboratory. Consequently, this led to frequent connection losses. A potential solution to this problem would be the addition of more sensors to ensure a broader coverage for more reliable tracking.

Another concern identified during the testing phase was the presence of latency issues. These occurred between the application and the server, which operates the control system for the new motor

5. Summary

This article delves into the development of a virtual reality game, specifically designed for the VirtuSphere, which is enhanced by its new motor system. The primary goal of this game is to guide player movements in a controlled and intuitive manner. A significant aspect of this project is the collection of data that will allow understanding the correlation between the user's walking characteristics and the parameters of the spherical walking simulator.

References

[1] L. Bazavan, H. Roibu, I.-C. Resceanu, N. Bizdoaca, Study regarding improving of full immersion for virtusphere system (2020).

[2] W. S. Medina Eliana, Fruland Ruth, Virtusphere: Walking in a human size vr hamster ball, Proc. of the Human Factors and Ergonomics Society Annual Meeting 52 No. 27 (2008) 2102–2106.

[3] L. Bazavan, H. Roibu, N. Bizdoaca, Strategy control of drive mechatronic system for virtusphere with two actuators (2022).

[4] M. T. Zdzisław Kowalczuk, Sphere drive and control system for haptic interaction with physical, virtual, and augmented reality, IEEE Transactions on Control Systems Technology PP (2018) 1–15.

[5] J. Lebiedź, A. Mazikowski, Image projection in immersive 3d visualization laboratory, Proc. Comput. Sci. 35 (2014) 842–850.

[6] L. Bazavan, H. Roibu, S. Cismaru, D. Rosca, N. Bizdoaca, Design of a new sensor architecture for mechatronic systems interconnected with virtual environments (2022).

[7] L. Bazavan, H. Roibu, S. Cismaru, N. Bizdoaca, Vr application for virtual environment with response to users' motion (2023).