3D Modeling and Navigation using Leap Motion

Mayank Anchlia[#1], Amit Rege[#2]

[#]*CS&IS Department, Birla Institute of Technology & Science, Pilani - Goa*
NH 17B, Bypass Road, Zuarinagar, Sancoale, Goa 403726, India

[1]f2014053@goa.bits-pilani.ac.in

[2]f2014081@goa.bits-pilani.ac.in

***Abstract*— This document gives an overview and the implementation details of the 3d modeling and navigation using leap motion developed as part of the course 'Software Development of Portable Devices'. The game is made on the Unity Game Engine. The game uses VR to view the three-dimensional environment created, and to track the orientation of the player. It detects gestures using a Leap Motion sensor to track other inputs from the user, for instance and tracks hands for recognizing gestures.**

***Keywords*— Virtual Reality, Leap Motion, Google Cardboard, Gesture Detection,3D modeling, Navigation, Unity Game Engine.**

I. Introduction

Virtual reality (VR) has become a very popular technology in recent years, which used in the field of multimedia for various purposes. One of the applications that widely used for simulating physical presence in the real world is walk-through VR. In the walk-through VR system will generate simulation

character or avatar for user which able to control movement, especially first person movement walk-through in VR with many input devices. In this paper, we present a human - computer interaction with the connection of Google Cardboard and Leap Motion, a new technological device for VR. Google Cardboard is VR headset or head - mounted display devices that have a small display optic in front of each eye. Google Cardboard can track head movement and change view point follow it. Leap Motion is in - air controller that can track hand gesture of the user. The combination of them will make users feel like immerse to VR. Users can move avatar any way in VR by their hand interact through the system via these devices. We introduce a new interactive hand gesture system with palm normal for control steering develop by the game engine Unity3D applies synchronization of google cardboard and Leap Motion. Our design and development method will allow users to adjust moving speed follows the hand gesture and pick objects and model them according of the user's hand that make a smoothly moving with acceleration

Interaction in VR is very challenging for developers to create new method or applications with respond to the needs of users. Especially, continuous movement control by using hand gesture that able to track hand ranges which combination with various devices. Interaction devices in VR for movement control base on hand gesture in this paper categorized into 2 types. The first one is HMD devices these are tracking and signaling actions are the primary means of input into virtual environments. The HMD device that we use for this project is Google cardboard. The other one is hand tracking devices or in - controller for track user's hand. We use Leap Motion for tracking and determination of a hand's position and orientation. Then this device track and report the position and orientation of the game engine. The game engine is provided for connecting between users and all devices. By the properties of the game engine Unity3D appropriate for using with already supported with Google Cardboard and Leap Motion

II. Modules Used

We used Unity3D software version 5.5.1f1 [1] to create the game. Unity provides a GUI which simplifies game development by providing various assets - graphics, scripts etc. - through the asset store [2] and through various other third party websites. It also allows the creation of one's own assets. Although we tried out a large number of assets, we are mentioning only the important ones for brevity:

- Google VR SDK for Unity [3]
- Leap Motion sdk [4] Leap Motion Core Assets
- Leap Motion Interaction module, Leap Motion hand module

The hardware modules used for the project were Leap Motion Orion modules, one Google Cardboards and one VR compatible mobile phones. We also used one laptops with a minimum of 8GB RAM.

III. Architecture

The game consists of three modes, each mode has some functionality with them, in all three modes, navigation is available, i.e. the player is allowed to walk in the scene in all three modes.

The first mode is basically being the paint mode. This mode gives user the choice to paint any object available in the game scene. The second mode is the build mode where the user can build the complex objects using simple objects like cube, cuboid, cylinder, sphere etc. The third mode is the pinch mode where user can create 3d strands of clay using pinch gesture of hand.

IV. Implementation

The implementation of the game is in three parts. First, the virtual environment and the rules of the game are created. Then, implementation of the gesture detection module and the virtual reality module. Later synchronization of the gyroscope Sensor with the leap sensor so as to give a perfect Virtual reality experience

A. *Environment Creation*

Using assets from the Unity Asset Store [2], and a few 3D graphical objects and patterns from photoshop, the mockup of the game was created. The backdrop of the game consists of a city with a build box in it, In the build box the user can design and build various complex objects using simple object.

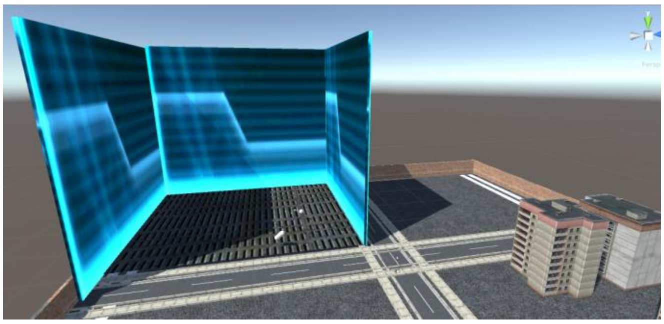


Fig. 1 The backdrop of the city in a screenshot from Unity

The user can walk freely in the game with hand gestures, hand gestures are also used to display of the menu as well in the game mode for various operations

B. *Extension to Virtual Reality*

The Google VR SDK for Unity [3] was used to create the game so that it would be compatible with Google Cardboard and the Android platform. We used the various scripts provided in this package, instead of many default Unity scripts.

Google Cardboard works by placing the phone at the optimal distance away from the eyes. Then, by using a physical divider and also dividing the screen into two parts, the lenses create a 3D effect when held up to your eyes, due to the stereoscopic vision in our eyes. The movement of the head is tracked by using gyroscopes present in the phone, which gives an effect so that it looks like the player is actually in the game world.

Fig. 2 Head mounted leap with Google cardboard.

Currently, Unity doesn’t support Google Cardboard by default, but does support other VR modules such as Oculus Rift, or HTC Vive. But in the future versions of Unity (version 5.6 onwards), Google Cardboard would be supported by default, and it would not be necessary to use this plugin.

Unity Remote 5 [6] is used to mirror the display of the game play on an android phone connected via USB, with Developer Mode switched on in the phone. This was done in order to make the synchronization easy with leap motion sensor and the computer, since no mobile module is available till date for Leap motion sensor.

C. Game Modes

(I) Implementation of Paint Mode

When the user enters paint mode, he is allowed to paint objects in the world. The user can select the color of the paint via the menu.

The paint mode is implemented by changing the color of individual pixels on the texture attached to the object. This is done using the Unity API call SetPixels(). Since this operation is performed on the GPU, we have to make sure this operation is performed as few times as possible to avoid making the game very slow. Thus, whenever the user points his right palm towards an object, the point of intersection of the palm normal and the surface of the object is calculated. At this point, we draw a circle of specified radius.

(II) Implementation of Build Mode

On entering Build mode, the user can interact with 3D objects in the scene and can join them to create more complex objects. Objects are put in proper position, they can be joined by closing the right fist. Internally, when the user closes his fist, scripts attached to the objects check for a collision between the two objects. If there is no collision, objects cannot be joined. If there is a collision, a Fixed Joint is created between the two objects at their point of collision. Now, the two objects have a fixed relative position with respect to this joint and therefore, have a fixed relative position with respect to each other.

D. Gesture Detection

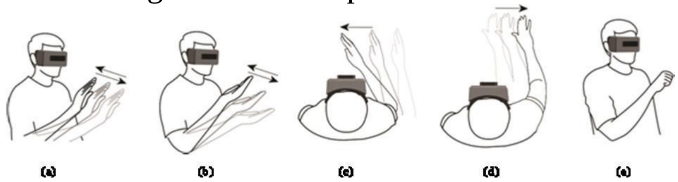
The main objective of this project is developing a new input method in VR. Therefore, implementation of gestures using google cardboard and Leap Motion are considered to apply for first person movement control in VR such as walk-through VR that allow user to immerse with the virtual environment by moving and observe them. In this section, we design single hand gestures by continuous motion of the hand during the walkthrough in VR separated as follows:

A.) Forward Movement

For hand gesture if user wants to move the avatar forward in our method just raise a user's left hand in front of their face with fist closed.

B.) Backward Movement

To move avatar backward (user still looking straight ahead not turn back) our method will use the left-hand gesture of fist open to move backward.



C.) Step Left and Right

Side step in this case does not turn left or turn right, but it means walk step to left side or right side while character's face continues looking forward. To move a character one has to use the head tracking of the VR and use the hand gestures of forward and backward movement. The reading from the gyroscope is used to calibrate so that one can feel the whole 360 virtual worlds.

D.) Stop (Hold Position)

To stop the movement the user has to make an intermediate position between fist open and close or can remove his left hand from the game screen.

E.) The pinch

To draw 3d strands of clay user has to make the pinch gesture and move freely in the environment.

V. Conclusions

Hand gesture interaction for first person movement in VR is useful for users to control the movement of their hand. In this project, we apply palm normal with hand gestures and implement for single hand control. Hand controllers with palm normal have more efficient and stable than hand gesture without palm normal. Another advantage of this method is controlling the speed of avatar movement. In particular, by the value of front and lateral are continuous values that able to control the speed like accelerator meter up to the user requirement. Improvement of HCI, hardware and software of interaction technologies will help VR system designers develop more natural and effective for participants to interact with the VR. The future work in this research, we would like to use both on hand for more control movement of a user's avatar including access of many functions or menu in VR systems. Furthermore, we are looking for add more devices such as Virtuix Omni, Sixense, Depth VR, etc. in our system and create new scripts contact with each device for work together.

Acknowledgment

We would like to thank Professor Sreejith V for his help throughout the semester, and for his comments and feedback as we completed the project.

We would also like to thank the CS & IS Department, BITS Goa, for providing us with the required hardware to create the game.

References

[1] Unity3D software. [Online] Available: https://unity3d.com/

[2] Unity Asset Store. [Online] Available: http://assetstore.unity3d.com/en

[3] Google. (2016) Google VR SDK for Unity. [Online] Available: https://github.com/googlevr/gvr-unity-sdk

[4] https://developer.leapmotion.com/

[5] Unity core assets for Leap motion sensor [online] Available: https://developer.leapmotion.com/documentation/unity/unity/Unity_Interaction_Engine.html

[6] Unity Remote 5 on Google Play Store. [Online] Available: https://play.google.com/store/apps/details