Advanced Science and Technology Letters
Vol.113 (Art, Culture, Game, Graphics, Broadcasting and Digital Contents 2015), pp.134-139
http://dx.doi.org/10.14257/astl.2015.113.28

Game Development utilizing Several Input Devices

DongIk Lee[1], KiYeol Baek[1], JiHyun Lee[1], Hankyu Lim[*1]

[1]Department of Multimedia Engineering, Andong National University,
388 Songcheon-Dong, Andong-City, Gyeongsangbuk-Do, Republic of Korea
yoha3567@gmail.com, Qkkndq@naver.com, wlgus5403@naver.com,
[*]Corresponding author: hklim@andong.ac.kr

Abstract. With the rapid growth of virtual reality (VR) content in the gaming industry, VR devices, most notably head-mounted displays (HMD), have gained a lot of attention lately. One of the most promising VR devices is the Oculus Rift, a VR HMD that the company Oculus VR is developing for VR simulations and video games. Another popular VR HMD is Sony's Project Morpheus that is compatible with the games on the PlayStation 4. This paper develops a VR game that is designed to be played using an Oculus Rift, a Kinect, and a smartphone. The developed VR game is a first-person shooter (FPS), titled "Kinect OculusRift Smartphone (KOS)." The KOS shows that the smartphone can become a convenient and cost-effective input device for VR games.

Keywords: Oculus Rift, Kinect, Smartphone, Virtual Reality Game

1 Introduction

The growing popularity of VR has brought about a steady increase in VR content along with continuous innovations in VR devices. In particular, HMD technology has attracted significant interest from industry. One of the most widely known HMDs is the Oculus Rift, a relatively cheap and easy-to-use VR HMD developed by Oculus VR. It is expected that VR content combined with an HMD like the Oculus Rift will be used in a wide variety of industrial sectors, such as video gaming, film and media, education, sightseeing, the healthcare and medical field, sports, and advertising. However, the majority of VR games that have been in development over the past few years rely on traditional controllers like a keyboard, a mouse, and a gamepad. The expectations of fully immersive VR cannot be met by such devices, and a natural user interface using gestures and spoken commands is required to enable users to control and interact with objects in the virtual world. Nevertheless, there are still some types of inputs that need to be explicitly given via an input device, which makes the use of wired and wireless input devices inevitable.

The KOS, a VR game developed in this work, is controlled with a combination of a Kinect (a motion tracking input device) and an Oculus Rift (a representative HMD

[1] This work was supported by a grant from 2015 Seoul Accord project(R0613-15-1148) of MISP(Ministry of Science, ICT and Future Planning) and IITP(Institute for Information and Communication Technology Promotion).

Copyright © 2015 SERSC

product). It also uses a smartphone to offer more diverse input methods. The KOS was implemented using C++ and DirectX 11. The KOS manipulates the gun aiming based on the player’s movements tracked by the Kinect and utilizes smartphone's touch, swipe and drag features to play the game. The smartphone was chosen as a supplementary input device because of its high penetration rates. For dedicated input devices, additional costs are necessary to purchase third party hardware, but the majority of people nowadays already own a smartphone. Today's smartphone incorporates various sensors and advanced features that can be exploited to feed useful inputs to the VR content including video games [1-4].

The KOS is an FPS game that is centered on gun and projectile weapon-based combat through a first-person perspective. The KOS continuously generates enemies (monsters) that start to move toward the player, and the player can shoot, destroying them and harming nearby enemies. For aiming, the KOS considers the player's hand location sent by the Kinect that keeps track of the player's location and movements. In the smartphone, the touch gesture is used to shoot, the swipe is to choose the weapon that the player wants, and the drag is to let the player maneuver his avatar.

2 Related Research

2.1 Kinect

The KOS game was implemented using Kinect SDK 1.0. The Kinect SDK provides skeletal tracking, the capability to extract and track the skeleton image of one or two people moving within Kinect's field of view. A Kinect skeleton is made up of 20 body joints (head, hands, feet, hip center, etc.). Among the Kinect skeleton data (20 joints coordinates in 3D space), the KOS game uses those corresponding to the player's right hand in order to manipulate the gun aiming. Microsoft DirectX 11, a collection of application programming interfaces (APIs) used to implement the KOS, is based on a left-handed coordinate system, whereas the Kinect uses a right-handed coordinate system. To obtain the correct joint coordinates for DirectX 11, the joint location information from the Kinect is modified by multiplying -1 to the z-coordinate.

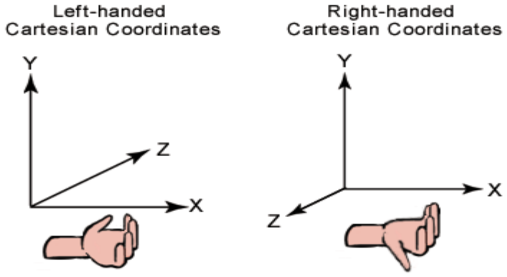

Fig. 1. Coordinates System [2]

2.2 Oculus Rift

The KOS was developed using the Oculus Rift DK2, the latest development kit for the Oculus Rift. The Oculus Rift requires graphic rendering for two different viewpoints of the left and right eyes. In addition, fish eye rendering is made for the left and right screens at a significant computational cost. The use of Oculus Rift thus increases the time of graphic rendering, which is a big obstacle in improving graphic quality and gaming performance. This is why the implemented KOS performs only lighting calculations with regard to the diffuse texture and bump mapping, without introducing shadow and blur effects. Like the Kinect, the Oculus Rift is based on a right-handed coordinate system. Hence, Oculus Rift's head tracking data passed to DirectX 11 that uses a left-handed coordinate system need to be corrected to compensate the differences [5].

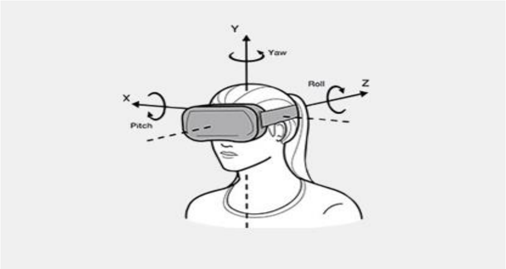

Fig. 2. Oculus Rift Coordinate System [3]

3 Functions and Implementation

3.1 Function Analysis

The developed KOS game can be divided into three parts: the KOS client implementing the VR game, the smartphone as an input device, and the server responsible for the communication between the KOS client and the smartphone. The communication between the server and the client is performed using Input/Output Completion Ports (IOCP). In the smartphone, the Channel Selector is used to process asynchronous sockets. It is possible that the KOS is implemented using the Transmission Control Protocol/Internet Protocol (TCP/IP) instead of IOCP. The KOS program includes features such as head tracking and display capabilities relying on the Oculus Rift, player motion sensing capabilities relying on the Kinect, gun aiming manipulations, and conversion of the smartphone packets sent by the server to an input data format. Figure 3 shows the data flow diagram of the KOS game.

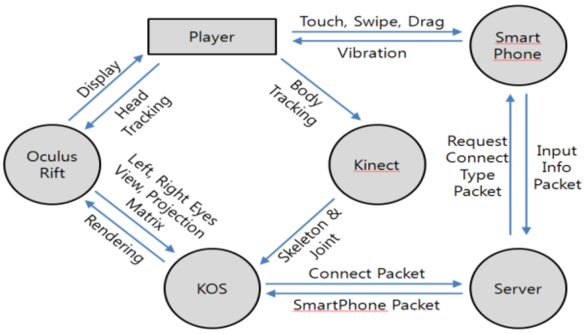

Fig. 3. DFD Structure Chart of KOS

3.2 Aiming Function utilizing Kinect

The KOS moves the gun aiming point along the x and y axes of the screen based on the x and y coordinates of the player's hand detected by the Kinect. To shoot, the KOS computes the 3D ray direction of the current x and y coordinates of the hand on the screen using the camera's view and projection matrix. The computed ray direction becomes the ballistic trajectory, and the effects of its encounters with virtual objects (monsters) are checked to harm the monsters [6].

3.3 View of Left and Right eyes of Oculus Rift, Projection Matrix Calculation

The KOS uses Oculus Rift's head tracking information and the camera's view and projection matrix in order to compute the view and projection matrix for the left and right eyes [7].

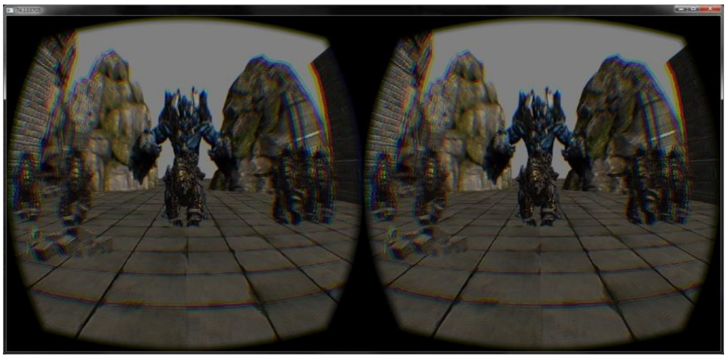

Fig. 4. Rendering of KOS with Oculus Rift

Copyright © 2015 SERSC

4 Conclusion

The VR game proposed in this paper allows the player to make a free movement while playing the game and enables input methods other than those offered by the Kinect and the Oculus Rift. The proposed game has some shortcomings as well. The Kinect requires a practical tracking area and its motion tracking accuracy is not perfect. With the Oculus Rift, the wearer's movement range can be further restricted, which makes the use of Kinect less effective.

Apart from the Kinect SDK 1.0 and the smartphone, other types of input devices that are suitable for use in VR content are currently under development and continue to grow in number and diversity. Oculus VR is also releasing motion controllers to be used with the Rift. It might be desirable to employ Oculus VR's own controller in place of the motion-sensing Kinect. It is important to note that various built-in sensors and features in a smartphone can be used effectively to provide the information required by VR content, thus making it an effective general-purpose input device for VR content.

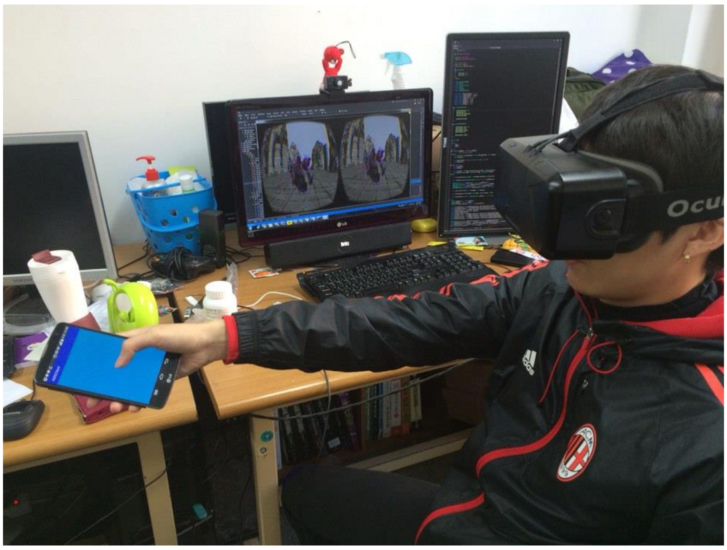

Fig. 5. Play KOS with Kinect, Oculus Rift, Smartphone

References

1. Junghyun Kim, Hankyu Lim, “Game Implementation based on Event utilizing gameUnity 3D”, 2015 Proceedings of KIPS Fall Conference, , 2015. 10, pp. 977-979,
2. Dongik Lee, Hankyu Lim, “Virtual Reality Contents using the OculusLift and Kinect”, Proceedings of the MCSI, 2015. 08, pp. 102-105,
3. Hangchan Jung, Hankyu Lim, “Kids Game Story-telling”, 2015 Proceeding of KMMS Fall Conference, 2015. 5, pp. 416 – 418,

Copyright © 2015 SERSC

4. Daegeun Kwon, Hankyu Lim, “Action Simulation Game Design using the Leap Motion Controller”, Proceeding of ICMEM 2014, 2014. 11, pp. 381-391,
5. Coordinate Systems (Direct3D 9), https://msdn.microsoft.com/ko-
6. kr/library/windows/desktop/bb204853(v=vs.85).aspx
7. Oculus Rift Coordinate System , https://developer.oculus.com/documentation/