AnArU, a Virtual Reality Framework for Physical Human Interactions

Matias N. Selzer, Martín L. Larrea
Laboratorio de Investigación y Desarrollo en Visualización y Computación Gráfica VyGLab
Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur
Av. Alem 1253, B8000CPB Bahía Blanca, ARGENTINA
{matias.selzer, mll}@cs.uns.edu.ar

ABSTRACT

Virtual Reality has become, once again, a popular and interesting topic, both as a research and commercial field. This trend has its origin in the use of mobile devices as computational core and displays for Virtual Reality. Android is one of the most used platform in this context and Unity3d is a suitable game engine for such platform. In order to improve the immersive experience, some electronic devices, Arduino especially, are used to gather information, such as the movement of the user's arms or legs. Although Android, Arduino and Unity3d are often used independently in Virtual Reality investigations, few studies use all of them together. Furthermore, each time these technologies are used in a new project, the developers have to think about a new way of communication between them. In this work we present AnArU, a framework for physical human interaction in Virtual Reality. The goal of AnArU is to allow an easy, efficient and extensible communication between the hardware and software involved in the Virtual Reality System.

Keywords: Virtual Reality, Arduino, Android, Unity3d, Human Computer Interaction.

1 INTRODUCTION

In recent years, there has been a spate of interest in Virtual Reality and its interactions, especially regarding the creation of Head Mounted Displays (HMD) using mobile devices as the computational core ([1, 16, 17, 9, 21]). We have seen mobile phones and small tablets become an ideal platform for Virtual Reality (VR). The current generation of these devices has full color displays, integrated cameras, fast processors and even dedicated 3D graphics chips.

From a Human Computer Interaction (HCI) perspective, the majority of studies have focused on the visual aspects of a VR experience ([9, 21, 18]), even the interactions with the VR world are solved through visual elements. For instance, in [21] when users want to touch a virtual button, they must first look at it inside the HMD and then click a physical fix positioned one. Thus, no matter where users are looking, they always use the same physical button. This way of interaction decreases the immersive experience, leading to unpleasant results.

The use of Game Engines for the creation of VR content has been extensively studied in the past years ([22, 20, 10]). They provide developers a quick way of building virtual content without the necessity of develop directly in low-level languages. They also allow the application to process input from many different sources, including keyboards, cameras and microphones. For this project we decided to use Unity3d, not only for the creation of virtual content, but also because it has a plugin architecture that allows developers to extend the core functionality. Furthermore, Unity3d has multiplatform support, including Android.

There have been many publications addressing the problem of creating a more natural interaction between users and virtual environments, most of them include a particular electronic device ([19, 4]). However, each of those investigations has solved one particular problem without considering the creation of a practical and extensible way of communication between these electronic devices and the VR system.

The aim of this paper is to present a framework called AnArU which allows a transparent, efficient and extensible communication between electronic devices and the VR system by combining Android, Arduino and Unity3d.

In Sections 2 and 3 we provide a brief introduction to the background concepts related to AnArU framework and some relevant previous work done about this topic. Next, in Section 4, we describe the AnArU framework and its architecture. In Sections 5 and 6 we present an experiment to test the framework usability and the case study. Lastly, in Section 7, we conclude with some remarks on the framework and directions for future research.

2 BACKGROUND

In this section, we provide a brief introduction to the three main component of AnArU Framework: Android, Arduino and Unity3d.

Android

Android is a mobile operating system (OS) based on the Linux kernel and currently developed by Google.

With a user interface based on direct manipulation, Android is designed primarily for touchscreen mobile devices such as smartphones and tablet computers.

Its source code is released by Google under open source licenses. This has encouraged a large community of developers and enthusiasts to use the open-source code as a foundation for community-driven projects, adding new features for advanced users.

Taking advantage of these benefits, in a similar approach as [16, 17, 21], we developed a Head Mounted Display by using the power of a smartphone running Android.

Arduino

Arduino is an open-source computer hardware and software company that designs and manufactures microcontroller-based kits for building digital devices and interactive objects that can sense and control the physical world. Their products are commonly known as Arduino and they are available commercially in preassembled form. The hardware design specifications are openly available, allowing the Arduino boards to be manufactured by anyone. It is estimated that in mid-2011 more than 300,000 official Arduinos were commercially produced and in 2013 more than 700,000 official boards were in users' hands.

We decided to use Arduino Uno for this project because it is one of the most popular, powerful and cheap microcontrollers nowadays, and also there have been many investigations which use it to create intuitive Human Computer Interaction devices ([6, 19]).

Unity3d

Unity3d is a cross-platform game engine developed by Unity Technologies and it is widely used to develop video games for PC, consoles, mobile devices and websites. Unity3d was developed with emphasis on portability, being capable of porting to Windows, Linux, Mac, Android and iOS, among others.

We chose Unity3d for this project because it is available for free, it is easy to use, and gives us the possibility to deploy Android applications with high quality 3D graphics.

3 PREVIOUS WORK

The combination of Android, Arduino and Unity3d as a platform for VR is not very common. There are few and recent works published about these technologies working together. In many investigations, virtual worlds are created in the context of cultural heritage; Davies et al. [6] worked with archaeological sites, allowing a simultaneous exploration of virtual and real environments. They used an Android device as the visualization platform and Unity3d to create the virtual buildings. Arduino is used to gather information such as orientation of the viewer, physical location, tilt, pan and other movements of the tablet. Lyons et al. ([14]) developed Loupe, a handheld near-eye display. Although it is not a HMD, it is very similar. In this case Unity3d is used for the GUI, Android is the computational core and Arduino provided sensor information.

There are several investigations about Augmented Reality (AR) by using Android, Arduino and Unity3d. AR is defined as the view of a physical real-world environment whose elements are augmented (or enhanced) by computer-generated sensory input, such as sound, video or graphics ([2]). Many research papers have used these technologies in AR ([15, 12, 13]).

Previous works have focused only on developing a solution for just a particularly problem. However, a reusable solution has never been considered. Therefore, each time they want to use these technologies in a new project, they need to develop a new way of communication. AnArU framework deals with this problem by integrating Android, Arduino and Unity3d technologies in a reusable and transparent framework.

4 ANARU FRAMEWORK

AnArU framework consists of three main modules:

- A Unity3d Module running on an Android device, responsible of connecting the virtual application to the rest of the system;
- An Arduino Uno Module running on an Arduino Uno device, responsible for controlling any electronic device attached to it;
- A Java Module, which is a Plugin running on an Android Device, responsible of the communication between the other two.

An overview of the framework is shown in Figure 1. A brief description of each module is presented in the following subsections.

Android Java Plugin

The VR application running on Unity3d has to be able to communicate to the Arduino application in order to transmit the necessary information. As Unity3d provides only high level programming languages, low level Android libraries, such as sensor information or Bluetooth information, can not be reached. Therefore, a Java Plugin running on the Android device was created to establish a bridge between Arduino Module and Unity3d Module.

The Plugin first task is the connection to the Arduino Module by using Bluetooth libraries. To accomplish this, the Arduino Bluetooth shield MAC is required. Thus, once the connection is established, a thread is executed in order to wait for any received

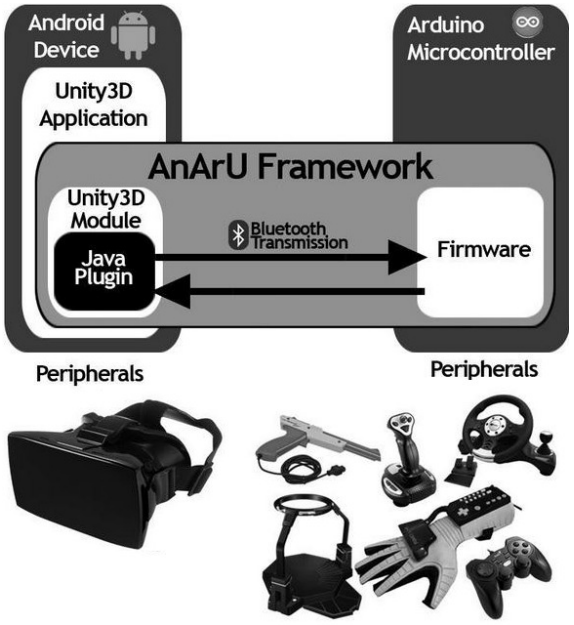

Figure 1: AnArU Framework Overview

message. Hence, special functions are provided to transmit information between Unity3d and Arduino Modules.

As we need a way to send and receive information from any electronic device, a protocol with a generic message format was created. The message format is a single string of text, surrounded by special characters. The Java Plugin automatically receives any new message from the Arduino Module and keeps a backup of the last message received. Then, that message is sent to the Unity3d Module. In the same way, when the Unity3d Module wants to send a message to the Arduino Module, an analog communication takes place. In order to save time, the protocol is Asynchronous. Furthermore, a retransmission system was implemented to detect and recover broken or lost messages.

Arduino Module

The main purpose of the Arduino Module is to obtain information from any electronic device or sensor connected to the Arduino Uno and send that information to the rest of the system via a Bluetooth communication. Thus, those electronic devices can be used as interactive devices in the VR system. The essential components of this module are an Arduino Uno board and a Bluetooth shield.

The application running on Arduino is in charge of waiting for the arrival of any message, and sending new messages whenever necessary. Therefore, once users know how to communicate, they can connect any devices or peripherals to the Arduino Module and start transmitting information to the other modules. Then, a new level of abstraction is introduced because users can communicate by using such interface regardless of the connected devices.

Unity3d Module

The Unity3d Module consists on a collection of scripts that allow the initialization and communication of the Java Plugin. Hence, users can make any application in Unity3d, communicate to the Arduino Module by using the provided methods, and do anything they need with the received values. Note that the values come in form of a string of text. However, as the user knows the specific structure of that message, it can be parsed in an easily manner and the values can be retrieved.

5 EXPERIMENT

In an ideal VR system, the minimal delay between users' movements and the corresponding visualization should exist ([8]). In general, a maximum accepted delay is in the order of milliseconds. Hereby, a quantitative speed analysis to measure the communication time, based on the transmission time of a specific size message was applied. As [7] suggested in 2004, the average speed of a Bluetooth 2.0 communication was 3Mbit/sec. Despite of the improvements of Bluetooth transfer rate achieved during last years, we used this value as a lower bound in the current calculations.

For this experiment we tested the communication time of a specific message containing three float values corresponding to the values provided by the gyroscope. For this message, with a size of 9 Bytes, after a test session, we obtained an overall transmission time of 0.04ms, compared to an ideal transmission time of 0.024ms. In case longer messages are needed, we also tested the communication time of messages with a size of 23 Bytes. This experiment showed an overall transmission time of 0.07ms. We must consider that newer Bluetooth adapters have even a faster transmition rate. Hence, these results suggests that the communication time of AnArU framework is by far fast enough to fulfill its objectives.

6 CASE STUDY

In recent years, numerous investigations has focused on the ability of walking in VEs, leading to an increment of users' immersion ([5]). Our current investigation involved building an omnidirectional walking platform controlled by Arduino which communicates to a VR Unity3d application running on an Android mobile device, by using AnArU framework.

The platform consists of a circular wood base and iron pipes with a ring in the middle in order to hold the users inside of it. In addition, elastic ropes are used to hold users exactly in the middle of the platform. They are also equipped with a pair of rollers so they

can walk freely inside the ring while maintaining their position. In order to sense users' movements, an Arduino Uno Microcontroller equipped with a gyroscope is attached to one of their legs. Thus, by measuring the leg angle respecting to the vertical, the system knows whether users are walking or not, and also the corresponding direction. Users are also wearing a Head Mounted Display running a Unity3d application which recreates a complete model of them in a virtual environment. That is, a virtual representation of the user's body and environment in an appropriate scale. By using AnArU framework, a communication between the Arduino on the leg of the user and the HMD application is performed, giving users the sensation that they are really walking inside the virtual environment, increasing the immersion level. A user with all mentioned components is shown in Figure 2.

Figure 2: User on the omnidirectional platform. He is wearing a HMD running a Unity3d application on an Android tablet, and the Arduino Uno is attached to his leg.

Baraka et al. ([3]) showed that in most cases the developer has to be very engaged in the designing of the communication protocol between Android and Arduino modules. In our study, the communication between the different modules is transparent. Hence, there is no need to configure a new communication each time a new way of interaction between these technologies is done. Lai et al. ([11]) used a commercial Plugin to perform a similar connectivity. However, this Plugin is not open source and just a few aspects of the microcontroller are available.

During the first experiments, we observed some problems when the communication went from Arduino Module to Unity Module and the other way around, at the same time. A retransmission mechanism was implemented to solve this problem and we have not observed any more similar issues.

7 CONCLUSIONS

Prior work has documented some interactivity between Arduino, Android and Unity3d technologies. However, none of them defines a transparent and extensible method of communicating these technologies. In this study we created and tested a framework capable of interconnecting these technologies in a straightforward way. Furthermore, we found that the communication is fast enough to satisfy the necessities of VR interactions.

AnArU framework can be used for any VR interaction that uses these technologies, helping developers to save time since they would not need to design any new communication protocol. However, some limitations are worth noting. Although Arduino Uno is a powerful tool for prototyping, it has little memory and computational power, being not functional in cases where more power is needed. On the other hand, Unity3d could not always provide the necessary tools for an specific project. Also, using an Android device as a HMD and computational core for VR applications may not provide the best graphic quality compared to a desktop computer.

Future work should therefore consider using a more powerful microcontroller in case more memory or computational power is needed. Other ways of communication, such as USB or Wifi, should also be considered in case that Bluetooth transfer rate is not enough. Another Videogame Engines should also be considered in case Unity3d does not fulfill the user's necessities.

REFERENCES

[1] A. Amer and P. Peralez. Affordable altered perspectives: Making augmented and virtual reality technology accessible. In *Global Humanitarian Technology Conference (GHTC), 2014 IEEE*, pages 603–608. IEEE, 2014.

[2] R. T. Azuma et al. A survey of augmented reality. *Presence*, 6(4):355–385, 1997.

[3] K. Baraka, M. Ghobril, S. Malek, R. Kanj, and A. Kayssi. Low cost arduino/android-based energy-efficient home automation system with smart task scheduling. In *Computational Intelligence, Communication Systems and Networks (CICSyN), 2013 Fifth International Conference on*, pages 296–301. IEEE, 2013.

[4] J. Blake and H. B. Gurocak. Haptic glove with mr brakes for virtual reality. *Mechatronics,*

IEEE/ASME Transactions on, 14(5):606–615, 2009.

[5] T. Cakmak and H. Hager. Cyberith virtualizer: a locomotion device for virtual reality. In *ACM SIGGRAPH 2014 Emerging Technologies*, page 6. ACM, 2014.

[6] C. Davies, A. Miller, and C. Allison. Virtual time windows: Applying cross reality to cultural heritage. In *Proceedings of the Postgraduate Conference on the Convergence of Networking and Telecomunications*, 2012.

[7] A. Dursch, D. C. Yen, and D.-H. Shih. Bluetooth technology: an exploratory study of the analysis and implementation frameworks. *Computer Standards & Interfaces*, 26(4):263–277, 2004.

[8] R. A. Earnshaw. *Virtual reality systems*. Academic press, 2014.

[9] W. Hürst and M. Helder. Mobile 3d graphics and virtual reality interaction. In *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology*, page 28. ACM, 2011.

[10] J. Jacobson and M. Lewis. Game engine virtual reality with caveut. *Computer*, 38(4):79–82, 2005.

[11] A. S. Lai and S. Leung. Mobile bluetooth-based game development using arduino on android platform. In *Applied Mechanics and Materials*, volume 427, pages 2192–2196. Trans Tech Publ, 2013.

[12] C.-F. Lin, P.-S. Pa, and C.-S. Fuh. Mobile application of interactive remote toys with augmented reality. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacific*, pages 1–6. IEEE, 2013.

[13] C.-F. Lin, P.-S. Pa, and C.-S. Fuh. A mar game design via a remote control module. In *Augmented and Virtual Reality*, pages 3–18. Springer, 2014.

[14] K. Lyons, S. W. Kim, S. Seko, D. Nguyen, A. Desjardins, M. Vidal, D. Dobbelstein, and J. Rubin. Loupe: a handheld near-eye display. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 351–354. ACM, 2014.

[15] H. Olmedo and J. Augusto. Towards the commodification of augmented reality: Tools and platforms. In *New Trends in Interaction, Virtual Reality and Modeling*, pages 63–72. Springer, 2013.

[16] J. L. Olson, D. M. Krum, E. Suma, M. Bolas, et al. A design for a smartphone-based head mounted display. In *Virtual Reality Conference (VR), 2011 IEEE*, pages 233–234. IEEE, 2011.

[17] B. Petry and J. Huber. Towards effective interaction with omnidirectional videos using immersive virtual reality headsets. In *Proceedings of the 6th Augmented Human International Conference*, pages 217–218. ACM, 2015.

[18] L. Pujol-Tost. Realism in virtual reality applications for cultural heritage. *International Journal of Virtual Reality*, 10(3):41, 2011.

[19] D. Schmidt, R. Kovacs, V. Mehta, U. Umapathi, S. Köhler, L.-P. Cheng, and P. Baudisch. Level-ups: Motorized stilts that simulate stair steps in virtual reality. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 2157–2160. ACM, 2015.

[20] M. F. Shiratuddin and W. Thabet. Utilizing a 3d game engine to develop a virtual design review system. *Journal of Information Technology in Construction–ITcon*, 16:39–68, 2011.

[21] A. Steed and S. Julier. Design and implementation of an immersive virtual reality system based on a smartphone platform. In *3D User Interfaces (3DUI), 2013 IEEE Symposium on*, pages 43–46. IEEE, 2013.

[22] S. Wang, Z. Mao, C. Zeng, H. Gong, S. Li, and B. Chen. A new method of virtual reality based on unity3d. In *Geoinformatics, 2010 18th International Conference on*, pages 1–5. IEEE, 2010.