

Method of Testing Mission Critical Data Transmission between an Android Device to a Windows PC via Wireless TCP Communication: PCMOTE

Syed Nasir Alsagoff¹

¹ Computer Science Department, National Defence University of Malaysia, Sungai Besi Camp, Kuala Lumpur 57000 Malaysia

Abstract

For a mission critical operation involving TCP communication, lag free and accurate data transmission is critical. Past research conducted by this author has solved the problem for wireless TCP communication between an Android device and a Windows PC. To expand upon the findings of that research, this paper will research a low cost implementation of motion controller arcade stick for the PC using an Android device. The final output will be called PCMOTE.

Keywords: android, games, accelerometer, motion controller, arcade stick.

1. Introduction

The war for the 7th generation gaming console success began when Microsoft released its Xbox 360 system on 22 November 2005 and Sony released the PS3, a successor to the PS2 which is the top selling console of all time on 11 November 2006. Nintendo was a bit late when it releases the Wii on 19 November 2006. At the time, Nintendo had the least successful gaming 6th generation gaming console in the Gamecube against offerings from Microsoft and Sony. According to [1], the Wii has very inferior hardware capability as compared to the PS3 and the

Xbox 360. Usually that would have been a death sentence for a new gaming console as proven by Nintendo Gamecube but it has a secret weapon. Nintendo Wii became the most successful 7th generation gaming console because it was the first gaming console to successfully implement a motion-sensing game controller [1]. The Wii is not the first console offering a motion-sensing game controller. Many had tried and failed. Such examples are the U-Force for the Nintendo Entertainment System (NES), the Power Glove also for the NES and the Sega Activator for the Sega Genesis, all ranked in top 10 worst game peripherals of all times. They have failed because of issues of responsiveness and accuracy when playing games.



Fig. 1: Several of the worst gaming controllers of all times.

1.1. Background of Study

The Nintendo Wii won the 7th Generation gaming console war with its innovative motion controller called the Wiimote [3]. Nintendo saw the gaming interactivi-

ty as the most important part of the console appeal. Akio Ideka, who designed the Wiimote stated on the Wii launch website that:

“Of course, when playing a game, the nearest thing to the player is the controller. The controller should therefore be regarded as an extension of the player rather than as part of the console. I always bear in mind the importance of the fact that the player will have far more contact with the controller and UI (user interface) than the console itself.”

The main hardware in the Wiimote is an accelerometer that translates the user’s movement in an XYZ axis.

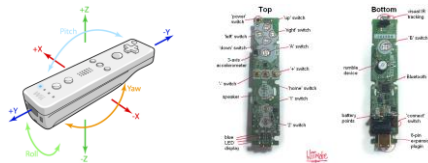


Fig. 2: The Wiimote

1.2. Objective of Research

This objective of this research is to develop a low cost motion-sensing gaming controller for the PC that is accurate and responsive. It will be based on a client/server architecture. In order to keep the cost low, a commercial of the shelf (COTS) hardware will be used. An Android phone will be used for this purpose. The motion-sensing gaming controller will be a replacement for the arcade stick. Once developed, a suitable game will be used to test the controller. Finally, the game must be fun for all to play. The developed controller itself will be motion-sensing gaming controller called PCMOTE.

2. Literature Review

2.1. Android OS

Android is a mobile operating system (OS) launched by Google. It is based on a 2.6 Linux Kernel and is made up with the OS itself, middleware, user interface and finally the applications.

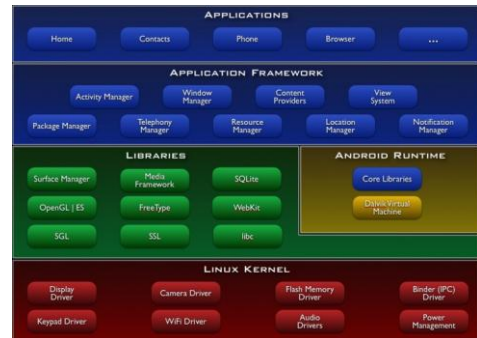


Fig. 3: Android Architecture

2.2. Android Hardware

Google has specified that all Android devices must comply with their Compatibility Definition Document (CDD) in order to use the Google Android trademark [6]. The document states that [6] “Device implementations SHOULD include a 3-axis accelerometer.” In addition it also stated that “Android 4.2 device implementations SHOULD include support for one or more forms of 802.11 (b/g/a/n, etc.)”

To keep the cost low, the Samsung Galaxy Y is chosen for the development process. It is one of the cheapest, branded and currently in production Samsung Android phone available here in Malaysia. It has WIFI and 3 axis accelerometer hardware which is the prerequisite for this motion-sensing gaming controller.

2.3. Client/Server Computing

Client/server comprised on two logical parts, a server (1st part) and a client (2nd

part) that receive services from the server [7]. Usually, client/server configurations consist of several clients connecting to the server as in Figure 4 below.

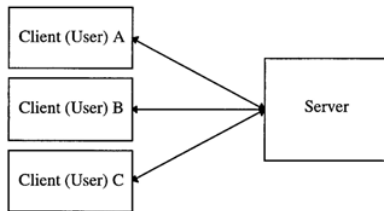


Fig. 4: Traditional Client/Server System [8]

For the purpose of this research, it will just be a single client connecting to the server. The communication will be two ways; the client will send the game commands to the server to be interpreted via the on screen game display. If the user failed or succeed in the game, the server will send the command back to the client as a feedback to the user.



Fig. 5: Proposed Client/Server Gaming System

3. Methodology

For this research, the game chosen is called Frogger. It is an arcade game introduced in 1981 by Konami and distributed by Sega, The aim of the game is to direct a frog to its home across the river. It will not be a full implementation of the game but a highly modified version suited for the PCMOTE. Since the aim of this reseach is not to develop the game itself

but the motion-sensing gaming controller, a Visual Basic 6 version of Frogger [9] was modified. The object of the game is to direct Frogger into all the 5 holes with 3 lives. The game will require 4 commands which are left, right straight and move. After modification, the game will be called Motion Frogger as in Figure 6.

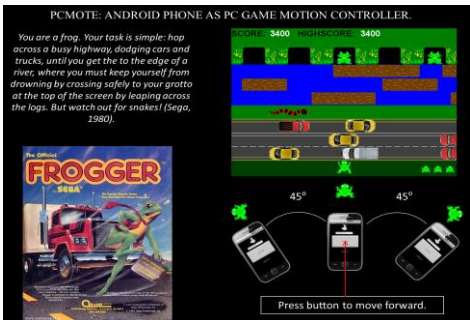


Fig. 6: Motion Frogger

3.1. PCMOTE

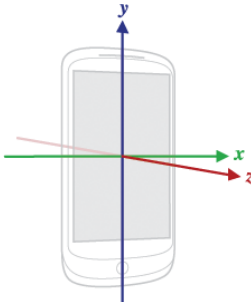


Fig. 7: Android XYZ axis

The PCMOTE will use the Galaxy Y's 3-axis accelerometer to translate the user's phone tilt into movement command for Motion Frogger. Here, only the accelerometer data from the Y axis is used

Table 1: PCMOTE Commands

45° right of the Y axis	Right command
45° left of the Y axis	Left command
Between 45° left and right of the Y axis	Sraight command
Button press	Move command

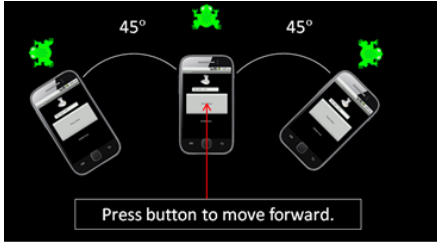


Fig. 8: Illustrated PCMOTE commands

The PCMOTE will be part of a client/server system. Both are developed concurrently, Motion Frogger will be the server while the PCMOTE is the client. It will be a single client connected to the server. All the command data are transmitted wirelessly via WIFI. The most important issue when developing a game controller is its responsiveness and accuracy. When using WIFI, this research has discovered there will be latency lag problems when transmitting data. That is the single most important issue that needs to be resolved before development of PCMOTE can proceed. For the purpose of this paper, this author cannot divulge the method as it is part of a commercial Android/PC client/server solution also by this author.

The Android application programming interface (API) allows for developer to directly access the raw data from the Android 3-axis accelerometer hardware.

Table 2: Android Accelerometer Hardware [10]

values[0]	Acceleration minus Gx on the x-axis
values[1]	Acceleration minus Gy on the y-axis
values[2]	Acceleration minus Gz on the z-axis
Note: All values are in (m/s ²) and G is gravity	

Table 3: Accelerometer Y axis data movement translation

values[1] > 4	Left command
values[1] < -3	Right command
values[1] < 4 and > -3	Straight command
A low pass filter is also used to smooth out the data and also to isolate the force of gravity on the Y axis (Gy)	

Finally, in order to send out the data to the server, a timer is used. A timer is used instead of the Android event so that the data is transmitted at an even and constant rate. The command data is sent twice a second.

Combining

- Proprietary Android/PC data transmission method
- Accelerometer data with low pass filter
- Fast, constant and even data transmission

equates to lag free, responsive and accurate gaming experience.

3.2. Immersiveness

Motion Frogger itself comes with a set of sound effects.

Table 4: Motion Frogger sound effect

Moving Frogger	“Boing” sound effect.
Frogger into a hole	“Chime” sound effect.
Frogger into all the 5 holes	Firework show with the appropriate sound effect.

The Galaxy Y itself is capable to playing sound effect. All the sound effects are using WAV file.

Table 5: PCMOTE Sound effect

Dying Frogger	Pacman dying sound effect.
Frogger into a hole	Victory sound effect.
Game over	“Game over” vocal.

Since the Galaxy Y is a phone, it also has a vibration alert function. The Android API also allows for the developer to access the vibration hardware. The developer has to send the amount of time that the vibration hardware needs to vibrate. This allows for a haptic feedback along with the sound feedback from PCMOTE. To make PCMOTE even more immersive, once the user tilts the Galaxy Y beyond the 45° threshold on the left and right side of the Y axis, the phone will slightly vibrate and offers a slight resistance to the user. This along with the visual cue on Motion Frogger will indicate to the user that Frogger has turned. The same goes when Frogger goes straight.

Table 6: PCMOTE vibration function

values[1] > 4	Vibrate 50ms once
values[1] < -3	Vibrate 50ms once
values[1] <4 and > -3	Vibrate 50ms once
Game Over	Vibrate 1s once
Frogger in hole	Vibrate 1s once
Dying Frogger	Vibrate 1s once

4. Testing

The prototype of Motion Frogger and PCMOTE was first shown at the National Defence University of Malaysia (NDUM)’s booth during the Malaysian Technology Expo 2013 (MTE 2013) from 21-23 Feb 2013 at Putra World Trade Center, Kuala Lumpur, Malaysia. The general public was allowed to play Motion Frogger. Prizes are given to anyone

who can get a certain number of Frogger into the holes.

Two sets of Motion Frogger and PCMOTE was setup during MTE 2013. It was a smash hit. More than three hundred people tried the game from young to adult and male to female. The key to the game is timing and patience. The timing to press the button to move Frogger into the hole is very critical to success. If the low cost motion-sensing gaming controller is not accurate, not responsive and laggy, the testing would not have been a success.



Fig. 9: Testing Motion Frogger and PCMOTE during MTE 2013

Motion Frogger and PCMOTE got a silver medal during MTE 2013.

5. Conclusion

The success of Motion Frogger and PCMOTE during MTE 2013 has shown that it is possible to have a low cost motion-controller for the PC. It is hoped that this research product can be commercialized. This author is currently actively seeking potential partners to licence the research product. Unfortunately, it is not easy to get the suitable partners in Malaysia as there are not many PC gaming developers in Malaysia

6. Future Work

The next step in PCMOTE is to convert it from a single button arcade stick for the PC to a steering wheel controller for the PC. To make it more immersive, an Android tablet such as the Samsung Galaxy Tab 2 7.0 can be used to replace the Samsung Galaxy Y. The biggest device is more suitable as a steering controller.

7. References

- [1] K. Sung, "Recent Videogame Console Technologies", Entertainment Computing Column, IEEE Computer, PP. 91-93, February, 2011.
- [2] "Top 10 Tuesday: Worst Game Controllers". IGN. 2006-02-21. Retrieved 2009-08-07.
- [3] Vajk, T., Bamford, W., Coulton, P., and Edwards, R., 2008. Using a mobile phone as a "Wii-like" controller for playing games on a large public display. Int. J. Computer Games Technology, article ID 539078.
- [4] "Leap Motion Control Technology Shipping with New Asus PCs" <http://www.tomshardware.com/news/Leap-Motion-Control-Asus-Partnership,20175.html>, Retrieved 19 May 2013.
- [5] Butler, M., 2011. Android: Changing the mobile landscape. IEEE Pervasive Computing 10 (1), 4–7.
- [6] "Android Open Source Project Frequently Asked Questions: Compatibility". source.android.com. Retrieved 13 March 2011.
- [7] Lewandowsky, S. M., "Frameworks for Component-Based Client/Server Computing," ACM Computing Surveys, vol. 30, pp. 4-27, 1998.
- [8] Orfali, R., Harkey, D., And Edwards, J. 1996. The Essential Client/Server Survival Guide. John Wiley, New York.
- [9] "Frogger1", <http://www.planet-source-code.com/vb/scripts/ShowCode.asp?txtCodeId=37584&lngWId=1> Retrieved 12 Oct 2012
- [10] "SensorEvent" <http://developer.android.com/reference/android/hardware/SensorEvent.html#values> Retrieved 19 May 2013