

Research Article

Android based Remote Game Controller

Akshay Kokate^{Å*}, Anay Karkhanis^Å, Anushka Bhat^Å, Chandrakant Khalse^Å and Shalini.V.Wankhade^Å

Department of Computer Engineering (Savitribai Phule Pune University, Maharashtra, India)

Accepted 15 Nov 2014, Available online 01 Dec 2014, Vol.4, No.6 (Dec 2014)

Abstract

Expansion of games industry came in modern life, accompanied with innovative features in a game controller. Nowadays mobile industry is booming with the expansion of the variety of novel technologies, such as the use of 3D accelerometer sensors, high resolution cameras, and the capability of connecting to wireless networks and so on. Therefore, a smartphone can be added with the game control functions to become a mobile game controller. In this paper, we propose the design and implementation of making a mobile phone as a game controller using User Datagram Protocol socket programming. This system is developed by Java 2 Standard Edition and Google Android Software Development Kits which has already provided sensor and multi-touch APIs.

Keywords: Game controller, Smartphone, Android, Java 2 Standard Edition, Sensor API

1. Introduction

Nowadays, mobile industry is booming greatly with the expansion of the variety of technologies accompanied such as the use of 3D accelerometer sensors, high resolution cameras, capability of connecting to wireless networks etc. Smartphone based applications, thus, has also been broadened, with scope for more innovative and novel applications. The development of mobile smartphone hardware and applications shows that it is interesting to add the game control functions on smartphone to become an All-in-One controller. The integration of mobile computing and face-to-face cooperation by combining multi-sensors, Bluetooth, 3G and Wi-Fi communications can make the mobile phone work as a multi-functional controller. As new technologies are being discovered day by day, tough competition is going between mobile companies to produce better phones. These applications have been among the important reasons for the growth of the mobile market. So making maximum use of smartphone's hardware for controlling computer.

In the near future, most of everything may be controlled by smartphone. People could totally control Personal Computer using mobile phones. In this paper, we explore how to design and implement a mobile phone as a multi-purpose controller for games running on Personal Computer by using the UDP/IP protocol. Our work is developed with game control mapping configurations on the server by the Robot Class provided by Java 2 Standard Edition, and with game control functions on the Android-based phone to control a game running on a PC. Configurations can be modified appropriately to fit different games.

Besides, players also can use 3D sensor accelerators which can sense both rotational orientation and translational acceleration. Also, by generating native system input events, we can emulate the control functions, such as inputting by keyboard or mouse, accessing folder and simply controlling your computer console.

2. Related Works

The modern world is contracting with the blooming of mobile phone technology which changes day by day and have become an essential part in our daily lives. Starting with merely providing a phone call function, mobile phones feature innumerable capabilities, such as picture taking, music listening, web page browsing, and even music searching (Leu, J.-S., Changfan, C., Su, K.-W., & Chen, C.-F., 2013). Meanwhile, wireless controlling mechanism has been widely applied to control remote devices from the traditional ones like a TV set, a music player to the modern ones like a computer game and a wireless robot (Arumugam, S., Kalle, R. K., & Rrasad, A. R., 2013). Assisted with the up-to-date technologies upon software and operating systems, smartphone becomes a smart remote controller which can master almost every device around us.

Remote control using a smartphone has been an interesting research area with many ongoing developments. With the idea of a mobile phone controller, we developed a phone based application to control game running on PC with Sensor and Bluetooth APIs based on Bluetooth communication (Fernandes, C., Ng, K. Y., & Khoo, B. H., 2011). They used 3D accelerometer sensors to measure the movement of the mobile phone which can be used to control menu handling, player input, scrolling, and a variety of other functions. In our work, we use Wi-Fi wireless communication which is much more robust and

*Corresponding author: Akshay Kokate

capable of reaching a higher data transmission and a wider connection range. Controlling PC can be executed by many methods such as web services via a Bluetooth connection (Kiran, B. R., & Sankita, P., 2010). In our work, communication between PC and the mobile phone controller is built with a client-server architecture using the UDP protocol (User Datagram Protocol, 2013). The UDP protocol is a simple connectionless protocol that can be used to transfer datagram packets between the sender and receiver in both directions. That means that the client and server can send and receive datagram packets to each other simultaneously. UDP transports the data between the two ends in the form of data packets, which have a basic and simplified header with only 8 bytes of overhead. User Datagram Protocol (UDP) is used in some typical network applications, such as Domain Name System (DNS), Simple Network Management Protocol (SNMP), Dynamic Host Configuration Protocol (DHCP), and the Routing Information Protocol (RIP). UDP is considered to be very useful in some circumstances, especially in speed and network management. UDP may lack a built-in reliability mechanism but have an advantage of a faster transmission rate compared to Transmission Control Protocol (TCP) from the application designer's point of view.

3. System Architecture

The system is separated into two main parts: the game configuration at the application server site and the mobile controller at the user client site. The application server which configures buttons for each game and generates native events is developed in Java. Developing the server part by Java is convenient for achieving the network capability, and due to the cross platform characteristic of Java, the server can be executable in many different platforms, such as Windows or Linux. We develop the server programs by using the Net Beans IDE that supports the Java SDK. Figure 1 shows the diagram of the system overview. We also have developed the client application on the Android based smartphone platform, on which developers can develop the programs by employing the native Android APIs and the support of sensor APIs provided by Google Android.

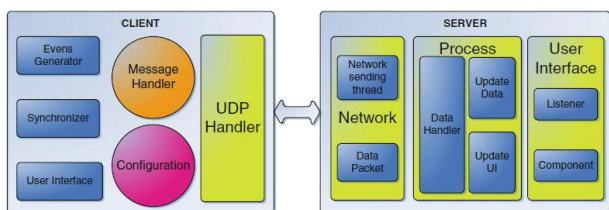


Fig. 1 System component

3.1 Mobile Client Design

3.1.1 Multi-touch and Sensor API

Google Android SDK 2.1 has already supported functional multi-touch and sensor APIs for developers. Sensors are accessible through the android hardware package. Android

uses the Sensor Manager, Sensor Event classes that allow users to access and control these sensors. To develop a smartphone application as a game controller which can connect fast and correctly control games running on PC, our application needs a good multi-touch mechanism. The Google Android SDK version 2.1 has already provided multi-touch functions. However, the multi-object function still has not been supported yet. That means users can totally sense the appearance of multi-finger but the system always returns the first object which the first user finger touches. Thus, in our work, we develop a well-worked multi-touch algorithm. Figure 2 shows the algorithm which can work out such finger events as ACTION_MOVE, ACTION_DOWN and ACTION_UP after tracking. Sometimes, some events may disappear on the multi-touch system since the fingers may move too fast, which would make the MOV event not run smoothly enough. Based on the multi-touch and sensor APIs, we develop the client component as Fig. 3 shows.

3.1.2 Format Message

User Datagram Protocol (UDP) has an optional checksum covering the payload and addressing information from the UDP and IP headers. Packets with incorrect checksums are discarded by the network stack in the operating system. The checksum is optional under IPv4 since the IP layer checksum may already provide the desired level of error protection. Thus, the message does not need the checksum or conduct the error detection, which has already provided by the IP layer. The message communication format can become simpler.

3.2 Game Server Design

The application server receives messages from the mobile client with the format shown in Fig. 4 and then generates native events such as mouse moving or keyboard stroking by using

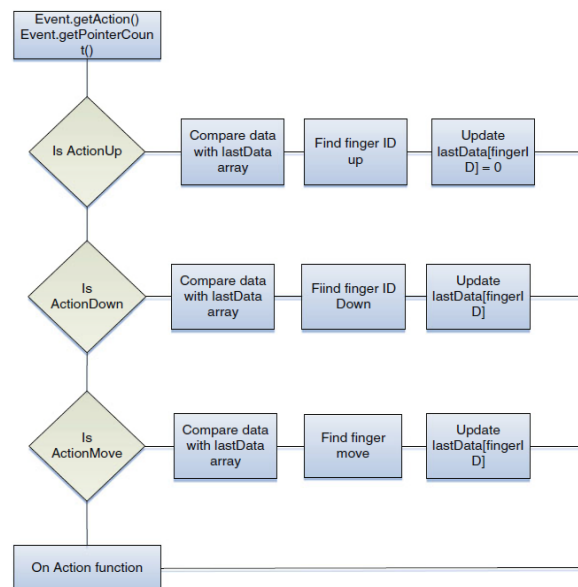


Fig. 2 Multi-touch algorithm

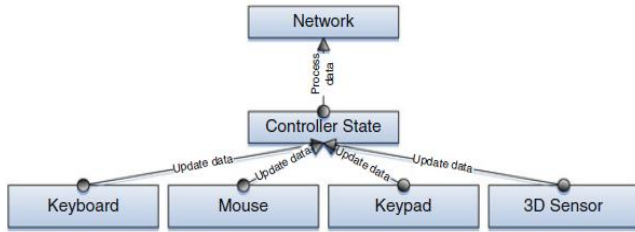


Fig. 3 Client component

ID Command (1byte)	Gamepad Button State (8bytes)	Mouse Info (8bytes)	Keyboard Info (2bytes)	Sensor State (6bytes)
--------------------	-------------------------------	---------------------	------------------------	-----------------------

Fig. 4 Message transmission format

4. Interface of Mobile Client Design

4.1 Keyboard and Mouse

By choosing the keyboard and mouse from the setting interface, the user can simply control the PC by simple tasks such as using the mouse to access the folder, start, and stop the application,

4.2 3D Sensor

Similar to a Wii-remote, we can control a game such as “Need for Speed”, which is a game allowing users to drive a car, by the 3D sensor.

4.3 Gamepad Interface

Our mobile gamepad application could control games running on PC smoothly and fast. With the multi-touch algorithm, users can use mouse and keypad simultaneously.

Conclusions

In this paper, we have presented our design for a game controller on a smartphone phone. The system supports a gamepad user-interface for PC games, with a capacity of button configuration. The application includes 3D accelerometers sensing and opens up the possibility for the same innovative capacity of a Wii games controller.

The multi-functional mobile controller, instead of mouse and keyboard, can work on a personal computer or laptop. Using the UDP protocol, the system is separated into two parts: the game server and the user client. The application server is developed by the Java technology with networking capability whereas the client is developed based on an Android mobile phone with Google Android SDK which fully supports the multi-touch and sensor APIs.

References

Wii_remote, http://en.wikipedia.org/wiki/Wii_Remote.
 Kinect, <http://en.wikipedia.org/wiki/Kinect>.
 Leu, J.-S., Changfan, C., Su, K.-W., & Chen, C.-F. (2013). Design and implementation of a fixed-mobile convergent music search engine (FMC-MUSE). *Wireless Personal Communications*, 70(4), 1911–1923.
 Arumugam, S., Kalle, R. K., & Rrasad, A. R. (2013). Wireless robotics: Opportunities and challenges. *Wireless Personal Communications*, 70(3), 1033–1058
 Fernandes, C., Ng, K. Y., & Khoo, B. H. (2011). Development of a convenient wireless control of an autonomous vehicle using apple iOS SDK. In *IEEE region 10 annual international conference, proceedings/TENCON* (pp. 1025–1028)
 Kiran, B. R., & Sankita, P. (2010). Implementing mobile phone as a multi-purpose controller using 3D sensor technology. In *Proceedings 3rd IEEE international conference on computer science and information Technology, ICCSIT*.
 Sakane, Y., Tsukamoto, M., & Nishio, S. (2002). Rajicon remote PC GUI operations via constricted mobile interfaces. In *Proceedings of the annual international conference on mobile computing and networking, MOBICOM* (pp. 251–262).
 User Datagram Protocol. (2013). http://en.wikipedia.org/wiki/User_Datagram_Protocol#Applications. available.
 Shang, H.-L., Xu, R.-M., & Yuan, J.-K. (2012). A smart home system based on ZigBee and iOS soft-ware. In *Proceedings of the international conference on parallel and distributed systems—ICPADS* (pp. 940–943).
 Android. (2013). <http://www.android.com> Retrieved March 1st, available.
 Java Robot Class Introduction. (2013). <http://www.developer.com/java/other/article.php/2212401/Introduction-to-the-Java-Robot-Class-in-Java.htm>, available.
 Horipad ex2 turbo, Game controller for Xbox 360. (2013). http://www.hori.jp/us/products/xbox360/us_xbox360_horipad_ex2_turbo/index.html, available.
 Android Sensor API. (2013). <http://developer.android.com/reference/android/hardware/SensorEvent.html>, available