



A Design and Implementation of Network Virtual Keyboard Using Smartphones

Byoung-Chan Jeon¹, Sang Yeop Cho², Sang-Jeong Lee^{*3}

¹*Division of Liberal Arts, Chungwoon University*

²*Department of Internet, Chungwoon University*

³*Department of Computer Science Engineering Soonchunhyang University*

ABSTRACT

Recently light weight notebook that has good performance produced commonly. In order to reduce the size of a laptop keyboard, the keyboard arrangement is important but 108 key used in the most computer are difficult to reduce the size. As a result, almost all lightweight notebooks and net books in the case of produce light weight, drastically the existing 108-key keyboard with numeric keypad keys removed to reduce the size and weight. In this paper, users who were having discomfort of using the keyboards of size reduced in lightweight notebooks and netbooks, to use easy as before solved by Android smartphones. Proposed a virtual numeric keypad system, which is implement on Android smartphone and a laptop connected with a wireless network.

© 2014 KKITS All rights reserved

KEYWORDS : 108-key keyboards, Android Smartphones, Virtual Numeric keypads, Wireless Network, laptop keyboards

ARTICLE INFO : Received 14 February 2014, Accepted 11 April 2014.

*Corresponding author is with the Department of Computer science & Engineering, Soonchunhyang University, 22 Soonchunhyang-ro Sinchang-myeon Asan-city Chungnam, 336-745, KOREA.
E-mail addresses: sjlee@sch.ac.kr

1. 서론

최근의 반도체 가격 하락으로 인해 성능이 좋은 컴퓨터와 과거 고가였던 노트북의 가격도 하락 하였고, 무거운 노트북의 사양을 낮추는 대신 휴대성을 높이 살린 넷북이라는 제품들도 등장하기 시작 하였다. 넷

북이 보급되기 시작하면서 성능이 아닌 휴대성을 강조하는 디자인의 제품들이 나오기 시작 하였다. 휴대성을 높이기 위해서는 화면의 크기를 줄이는 것보다 키보드의 디자인을 바꾸는 것이 효율적인데 기존의 108키 키보드의 오른쪽에 위치한 17개의 숫자 키패드를 제거해 제품의 크기를 줄이게 되었다[1,2]. 노트북과 넷북의 오른쪽 숫자패드가 제거 되면서 기존의 노트북으로 업무를 보던 사용자들에게는 업무를 보는데 큰 불편함이 생겼다. 따라서 노트북과 넷북에서 숫자패드를 이용하기 위해서는 별도로 USB형 숫자패드를 구매해야 할 것이다[3,4]. 2007년 APPLE사의 아이폰 3GS를 시작으로 풀 터치스크린 기반의 스마트폰이 보급되기 시작하였다. 스마트폰의 보급으로 기존의 제한적이었던 무선 인터넷 시장도 활성화 되었는데 수도권 중심이었던 Wi-Fi존이 전국으로 확대되어 언제 어디서든 Wi-Fi를 통해 무선 인터넷을 사용할 수 있는 시대가 되었다. 본 논문에서는 숫자 키패드가 제거된 노트북 사용자들의 불편함을 해결해 주기위해 무선 인터넷을 이용할 수 있는 스마트폰을 이용하여 가상의 숫자패드를 구현하는 것을 제안한다. 본 논문의 구성은 다음과 같다. 제 2장에서는 이론적 배경과 사례에 대해서 살펴본다. 제 3장에서는 가상숫자패드 시스템 설계과정을 설명하며, 제 4장에서는 제안한 시스템을 구현한 후 수행결과를 보인다. 제 5장에서는 결론을 기술한다.

2. 이론적 배경 및 사례

2007년 iPhone 3GS를 시작으로 스마트폰이 빠르게 보급되어 지고 있다. 스마트폰의 운영체제로는 심비안, iOS, Android OS가 대표적인 스마트폰의 운영체제이다. 최근에 보급되기 시작한 Android OS는 심비안과 iOS의 뒤를 이어 빠르게 스마트폰 시장에 자리를 잡아가고 있다[5,6]. 스마트폰에는 기본적으로 무선 네트워크인 Wi-Fi를 지원하고 있다. 스마트

폰의 보급이 늘어나면서 무선 네트워크를 사용할 수 있는 Wi-Fi Zone이라 불리는 장소도 생겨나게 되었다. 따라서, 안드로이드 모바일 플랫폼과 안드로이드 유사한 어플리케이션 및 Robot Class 대해서 살펴 보고자 한다.

2.1 안드로이드 모바일 플랫폼

구글에서 2007년 11월에 발표한 안드로이드 플랫폼은 모바일 운영체제로 운영체제, 미들웨어 및 주요 응용프로그램을 포함하는 모바일 기기용 소프트웨어 모음을 의미한다. 안드로이드는 컴포넌트의 재사용과 재배치가 가능한 응용 프레임워크와 모바일 디바이스에 최적화 된 Dalvik 가상머신, Webkit 엔진 기반의 통합 브라우저, 구조화된 데이터 스토리지를 위한 SQLite를 가지고 있다. 디바이스 에뮬레이터, 디버깅 툴, 메모리 및 퍼포먼스 프로파일링 그리고 이클립스 IDE를 위한 플러그인을 포함하는 풍부한 개발환경을 지원한다. 모바일 플랫폼으로서의 안드로이드 플랫폼의 장점은 자기가 개발한 응용을 안드로이드 마켓을 이용해 자유롭게 올리고 다른 사람이 개발한 응용을 받을 수 있는 자체 모바일 마켓 플레이스를 운영하고 있다는 점이다. 안드로이드 마켓에는 개발자가 안드로이드 응용을 개발해 자유롭게 올릴 수 있으며 안드로이드 마켓을 통해 사람들에게 무료로 유료로 응용을 다운로드 할 수가 있다. 안드로이드 마켓에는 약 5만 여개의 유료, 무료 응용이 업로드 되고 있으며 응용 수는 빠르게 증가하고 있다 [7,8].

<그림 1>은 안드로이드 시스템의 구조를 나타내고 있다. 안드로이드의 시스템 구조를 보면 상위 계층으로 응용 프레임 워크가 존재하며 하위 계층으로는 라이브러리, 안드로이드 런타임, 리눅스 커널로 이루어져 있다. 2.6 기반의 리눅스 커널은 카메라, 키보드와 같은 각 장치를 제어하고 전원

관리와 IPC Binder 드라이버를 담당한다. 안드로이드 런타임은 Dalvik을 포함하는 가상 머신과 Core 라이브러리를 포함하고 있다. 안드로이드 라이브러리는 안드로이드를 구성하고 있는 C와 C++ 라이브러리로 이루어져 있으며 아래와 같은 라이브러리로 구성되어 있다. System C Library는 BSD 기반의 Standard C 시스템으로 이루어져 있고 Media Library는 MPEG4, H264를 지원하는 OpenCore 라이브러리를 포함한다. Surface Manager는 2D와 3D의 여러 개의 화면 레이어를 화면에 표시하기 위한 매니저 역할을 담당하고 LibWebCore는 웹 브라우저 라이브러리를 포함하며,

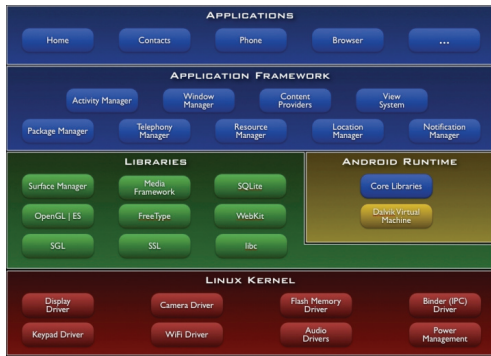


그림 1. 안드로이드 시스템 구조
Figure 1. Android System Architecture

안드로이드 자체 데이터베이스인 SQLite와 OpenGL 라이브러리를 가지고 있다. 응용 프레임 워크 층은 응용을 지원하는 각 매니저로 구성 되어 있다. 안드로이드 화면을 구성하는 View를 관리하는 View Manager, 응용과 응용 사이의 데이터 공유를 관리하는 Content Provider, 그래픽과 레이아웃 파일을 관리하는 Resource Manager, 안드로이드 상태를 알리는 Notification Manager, 안드로이드 응용을 구성하는 Activity를 관리하는 Activity Manager 등으로 구성 되어 있다.

2.2 Android의 유사한 어플리케이션

Google의 안드로이드는 애플의 아이튠즈에서 서비스하는 앱스토어와 유사한 기능을 하는 안드로이드 마켓을 운영하고 있다[10,11]. 기능은 아이튠즈의 앱스토어와 유사하다. 어플리케이션의 개수는 아이폰의 앱스토어에 비해 적은 수준이지만 그래도 상당량의 어플리케이션을 보유하고 있으며 앞으로 더 늘어날 것이다. <그림 2>는 안드로이드 마켓에서 인기있는 어플리케이션인 Gmote를 보여주고 있다.



그림 2. 안드로이드 Gmote 어플리케이션
Figure 2. Android Gmote applications

2.3 Robot Class

윈도우즈 운영체제 환경에서 실제 키보드와 마우스를 제어하기 위해서는 특별한 클래스가 필요하다. Java의 AWT 컴포넌트의 robot를 import를 하면 실제 윈도우즈 운영체제상의 마우스와 키보드를 제어할 수 있다. <그림 3>은 robot 클래스를 이용하여 마우스를 움직이게 하는 프로그램 예제이다.

```

public class MouseMoveEx{
public MouseMoveex()
{
try
{
Robot robot=new Robot();
robot.mouseMove(200,200);
}
catch (AWTException ae)
{
ae.printStackTrace();
}
}
}
public static void main( String() args)
{
new=MouseMoveEx();
System.exit(0);
}
}
    
```

그림 3. robot 클래스를 이용한 마우스를 움직이는 예
Figure 3. example using a mouse moving robot class

3. 가상 숫자패드 시스템 설계

3.1 시스템 설계

기존의 USB형 키패드의 단점은 별도의 비용을 들이 구입을 해야 한다는 것이다. 가상 숫자패드는 안드로이드 스마트폰을 이용하여 무선 네트워크에 접속을 하여 노트북에 연결한다. 무선 네트워크로 연결이 된 후 안드로이드 스마트폰은 노트북의 숫자키패드 역할을 수행하게 된다. <그림 4>는 가상 숫자패드 시스템 구성도를 나타내고 있다.



그림 4. 시스템 구성도
Figure 4, System Configuration

안드로이드 스마트폰은 노트북과 무선 인터넷으로 접속을 하게 된다. 안드로이드 스마트폰은 가상 숫자키패드, 가상 키보드, 가상 마우스, 가상 리모컨을 제어하는 기능을 가지고 있다. 어플리케이션은 4가지 기능을 이용하여 노트북을 제어하게 되는데 가상 숫자키패드의 경우 실제 숫자 키패드의 기능을 노트북에서 사용할 수가 있다. 노트북은 일종의 서버 역할을 수행하게 된다. 무선네트워크로 연결이 된 노트북은 어플리케이션으로부터 제어신호를 받아 해당되는 기능을 수행하여 실제로 노트북의 키보드와 마우스 값을 입력 시키게 된다. 노트북은 Java의 AWT 컴포넌트의 Robot 클래스를 통하여 스마트폰에서 온 제어신호를 가지고 키보드와 마우스를 구분하여 실제 윈도우즈 운영체제에서 동작을 하게 된다. 차후에 숫자 패드와 키보드뿐만 아니라 음악플레이어 리모컨과 같은 편의 기능도 추가할 수가 있다.

3.2 흐름도

가상 숫자패드의 흐름도는 <그림 5>에서 보여주고 있다. 스마트폰은 노트북에 접속을 하게 된다. 접속이 완료되면 스마트폰은 각 마우스, 숫자패드, PPT 리모컨과 같은 기능을 가진다. 스마트폰에서 마우스를 선택하면 마우스의 제어신호를 보내고 노트북으로 보내게 된다. 노트북은 Java의 AWT 컴포넌트의 Robot클래스를 이용하여 마우스를 움직인다. 스마트폰에서 숫자패드를 선택하여 숫자패드의 제어신호를 보내면 노트북의 Java의 AWT 컴포넌트의 Robot 클래스에서 키보드 제어 신호를 보내게 된다. 한 예로 동작을 살펴보면 스마트폰에서 PPT 리모컨을 선택하여 제어신호를 보내면 노트북 Java의 AWT컴포넌트의 Robot 클래스를 통해 PPT를 제어하게 된다. 이런 흐름도를 바탕으로 순차적으로 실행하게 된다.

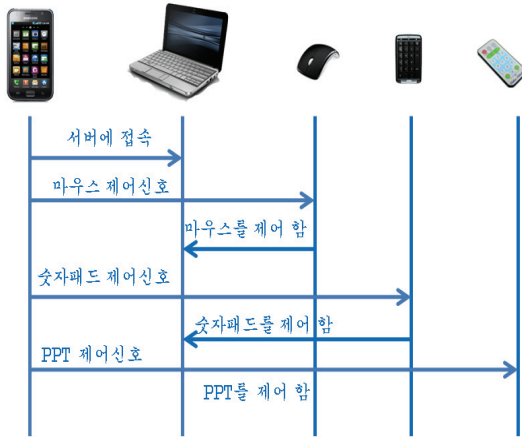


그림 5 시스템 흐름도
Figure 5. System flow chart

3.3 안드로이드 스마트폰 UI 설계

<그림 6>은 안드로이드 스마트폰의 기본 프로토타입 UI를 보여주고 있다.

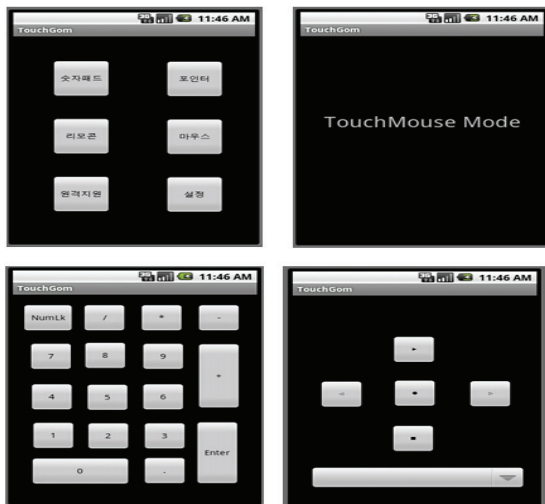


그림 6. 안드로이드 UI
Figure 6. Android UI

안드로이드 스마트폰 초기화면은 6개의 버튼으로 구성되어 있으며 각 버튼이 눌리면 자신이 해

당하는 액티비티를 호출하게 되고 호출된 액티비티는 무선 인터넷을 이용하여 노트북에 연결을 하게 된다. 가상 숫자패드는 모두 버튼으로 구성되어 있다. 구성된 버튼 하나 하나는 서로 다른 값을 네트워크 스트림을 이용하여 서버로 전송하게 된다. 서버인 노트북은 네트워크 스트림을 통해 안드로이드 스마트폰에서 보낸 값을 받게 된다. 서버는 이 값을 가지고 자신이 어떤 기능을 작동해야 하는지를 판별하게 된다. 예를 들면, 안드로이드 스마트폰에서 가상 숫자패드 액티비티가 '9' 라는 캐릭터를 네트워크 스트림을 통해 전송하게 되는데 Java 서버를 가진 노트북이 네트워크 스트림을 통해 '9' 라는 캐릭터를 받게 된다. 이 캐릭터는 서버 내부의 switch문 안으로 들어오면 캐릭터 '9' 에 대한 case에 걸리게 되어 이 case 문안의 명령문을 실행하게 된다. 명령문 안에는 Robot 클래스를 동작시켜 노트북의 실제 숫자 키패드를 누른 것과 같은 신호를 보내게 되어 노트북에 숫자 9가 찍히게 된다. <그림 7>은 구현중인 Java 서버와 안드로이드 액티비티간의 통신을 보여주는 프로그램이다. (a)프로그램은 onClick 이벤트가 발생하면 서버로 전송하는 일부분이고 (b)프로그램은 자바 서버인 안드로이드에서 보낸 값을 받아 키보드 이벤트를 발생시키는 과정이다.

```

Public void onClick(View v){
try {
        setSock(ip,port);
    }catch (IOException e){
        e.printStackTrace();
    }
    switch (v.getId()) {
        case R.id.nine:
            PrintWriterout9=new PrintWriter(netWorkWriter);
            char return_msg9='9';
            out9.println(return_msg9);
            break;
    }
}
    
```

(a)

```

while (trace){
Socket client=serverSocket1.accept();
System.out.println("S:Receiving...");
try{
    rebot=new Robot();
    BufferedReader in= new BufferedReader(
    new InputStreamReader(client.getInputStream()));
    char str=(char) in.read();
switch(str) {
case 'g':
    robot.KeyPress(numKey9);
    rebot.delay(DeISec);
    rebot.keyRelease(numKey9);
    System.out.println("S: Received.'" + str +"'");
break;
}
}
}

```

(b)

그림 7. (a) 안드로이드 어플리케이션 (b) Java 서버
 Figure 7. (a) Android Application (b) Java Sever

4. 구현 및 테스트

본 논문의 주요 내용은 가상 숫자패드가 구현되어 있는 액티비티이다. 키보드의 키패드와 동일한 구성으로 이루어져 있으며 숫자패드 버튼을 누르면 네트워크 스트림을 통하여 서버로 캐릭터 값을 전송하게

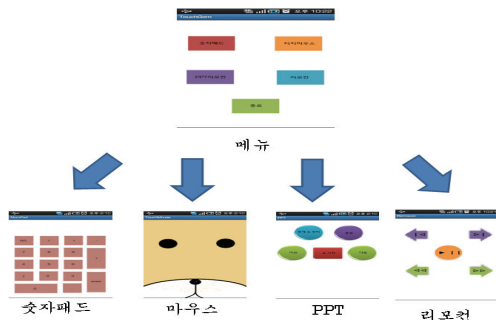


그림 8. 최종 어플리케이션 구성
 Figure 8. final application configuration

된다. <그림 8>은 안드로이드 스마트폰을 활용한 가상 키보드를 보여주고 있다. 그림에서 보여주는 것처럼 안드로이드 어플리케이션을 실행하게 되면

화면에 나타나는 메뉴, 숫자패드, 마우스, PPT, 리모컨의 액티비티들을 볼 수 가 있다. <그림 9>는 메인 액티비티에서 onClick 이벤트가 발생하면 해당 액티비티를 호출하는 일부의 프로그램이다.

```

// ...
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.numpad:
            startActivity(new Intent(this, TouchGomPad.class));
            break;
        case R.id.touchmouse:
            startActivity(new Intent(this, TouchGomMouse.class));
            break;
        case R.id.ppt:
            startActivity(new Intent(this, TouchGomPpt.class));
            break;
        case R.id.remocoon:
            startActivity(new Intent(this, TouchGomRemocoon.class));
            break;
        case R.id.exit:
            finish();
            break;
    }
}
}

```

그림 9. 메인 액티비티 onClick 이벤트
 Figure 9. Main Activity onClick event

onClick에서 이벤트가 발생하면 switch문이 동작하게 된다. switch문에서 4개의 이벤트인 숫자패드, 터치마우스, PPT, 리모컨, 종료 액티비티를 호출하는 이벤트가 발생하게 된다. 메인 액티비티에서 숫자패드 버튼을 누를 경우 onClick 이벤트에서 TouchGomPad 클래스를 호출하게 된다. <그림 10>은 onClick 이벤트가 발생시 호출되는 숫자패드 액티비티의 일부분이다. 그림에서 보는 것처럼 17개의 버튼으로 구성되어 있으며, 해당 액티비티의 동작을 살펴보면 우선 숫자패드 액티비티도 onClick 이벤트의 발생으로 액티비티가 동작하게 된다. 버튼을 누르면 onClick 이벤트가 발생하며 이벤트가 발생하면 서버에 접속을 시도하게 된다. 예외처리를 통해 서버가 Alive 상태면 이벤트가 발생된 값을 가지고 swith 문을 실행하게 된다. 이벤트 id 값을 이용하는데 버튼 9가 눌렀다고 가정하면 버튼 9는 XML 코드에서 'nine'이라고 주어지게 된다. switch문에 버튼 9에 해당하는 nine의 값이 넘어가게 된다.

```
public void onClick(View v) {
    try {
        setSocket(ip, port);
    } catch (IOException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
    switch (v.getId()) {
    case R.id.nine:
        PrintWriter out9 = new PrintWriter(networkWriter, true);
        char return_msg9 = '9';
        out9.println(return_msg9);
        break;
    }
}
```

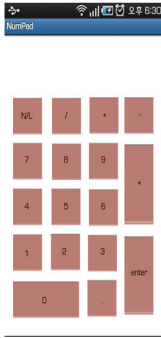


그림 10. TouchGomPad onClick 이벤트
Figure 10. TouchGomPad onClick event

<그림 11>은 메인 액티비티에서 발생한 이벤트가 TouchGomMouse의 경우 실행되는 액티비티이다. 앞절에서 잠깐 언급한 TouchDown, TouchMove, TouchUp 이벤트를 이용하여 마우스가 어떻게 이동할 것인가를 결정해주는 부분이다. 마우스는 모션 이벤트는 크게 down, move, up이 발생하게 되는 스마트폰의 화면을 누른 상태에서 손가락을 움직일 경우 move 이벤트가 발생하는데 이때 down 이벤트때 받았던 처음 좌표에서 움직인 만큼의 값을 빼 이동한 거리를 서버에 전송한다. 서버는 이 좌표 값을 받아 실제 윈도우즈 상의 마우스 좌표에 증감 값을 적용하여 마우스를 움직이게 된다.

```
public boolean onTouchEvent(MotionEvent event) {
    point = new Point();
    float x = event.getX();
    float y = event.getY();
    switch (event.getAction()) {
    case MotionEvent.ACTION_DOWN:
        touch_down();
        touch_startX(x);
        touch_startY(y);
        break;
    case MotionEvent.ACTION_MOVE:
        touch_move(x, y);
        // drawLine();
        break;
    case MotionEvent.ACTION_UP:
        touch_up();
        // drawLine();
        break;
    }
    return true;
}
```

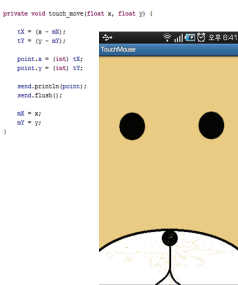


그림 11. TouchGomMouse의 onTouch이벤트
Figure 11. TouchGomMouse onTouch event

<그림 12>는 메인 액티비티에서 PPT 버튼을 누를 경우 onClick 이벤트에서 TouchGomPpt가 실행

되는 액티비티의 일부분이다. 총 5개의 버튼으로 구성되며 마이크로소프트 오피스의 파워포인트를 안드로이드 스마트폰으로 컨트롤하게 해주는 액티비티이다.

```
switch (v.getId()) {
    case R.id.nowshow:
        PrintWriter outnow = new PrintWriter(networkWriter, true);
        char return_msg9 = 'n';
        outnow.println(return_msg9);
        break;
    case R.id.exit:
        PrintWriter outexit = new PrintWriter(networkWriter, true);
        char return_msg8 = 'e';
        outexit.println(return_msg8);
        break;
}
```



그림 12. TouchGomPpt 클래스의 onClick 이벤트
Figure 12. TouchGomPpt Class onClick Event

파워포인트에서 가장 중요한 쇼시작, 이전, 다음, 현재 쇼부터, 종료를 구현한 것으로 역시 TouchGomPad와 같이 해당버튼이 눌리면 그 버튼의 값을 NetWorkWriter를 통해 서버에 전송하게 된다. <그림 13>은 메인 액티비티에서 리모컨 버튼을 누를 경우 실행되는 액티비티이다.

```
public void onClick(View v) {
    try {
        setSocket(ip, port);
    } catch (IOException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
    switch (v.getId()) {
    case R.id.ppt:
        PrintWriter outnow = new PrintWriter(networkWriter, true);
        char return_msg9 = 'p';
        outnow.println(return_msg9);
        break;
    case R.id.off:
        PrintWriter outexit = new PrintWriter(networkWriter, true);
        char return_msg8 = 'f';
        outexit.println(return_msg8);
        break;
    }
}
```

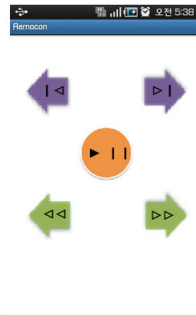


그림 13. TouchGomRemoon의 onClick 이벤트
Figure 13. onClick event of TouchGomRemoon

이 액티비티는 윈도우즈 상에서 실행되고 있는 다른 응용프로그램을 접근하여 제어하는 액티비티이다. TouchGom Ver1.2에서 추가된 기능으로 현재는 KMPlayer만 제어가 가능하다. 아래 화살표 부분은 5초 앞으로, 5초 뒤로 가게 해주는 버튼이고 위에 있는 화살표 버튼은 이전 파일과 다음 파일을 넘어가는 버튼이다. 가운데 버튼은 시작과 일시정지로 구성되어 있는 버튼이다.

안드로이드 스마트폰에서 가상 키보드인 4가지 액티비티별로 살펴보면 <그림 14>는 가상 숫자패드 실행 과정을 보여주고 있다.

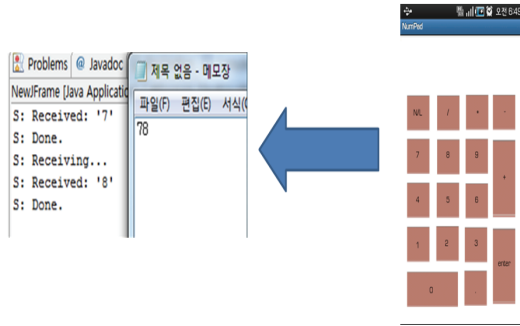


그림 14. 가상 숫자패드 실행
Figure 14. virtual number pad Run

실제로 구동된 어플리케이션에서 버튼이 눌린 경우 윈도우즈 메모장에 숫자가 표시되는 것을 보여주는 예이다.

메모장뿐만 아니라 워드, 한글 등의 문서편집이 가능한 곳이라면 버튼에 해당되는 값들이 표시가 된다. <그림 15>는 어플리케이션을 구동시켜 마우스를 움직이고 마우스의 버튼을 누르는 과정을 보여주고 있다. 마우스의 왼쪽버튼과 오른쪽버튼, 그리고 더블클릭의 기능을 모두 실행할 수가 있다.

<그림 16>는 PPT 리모컨을 실행한 동작을 나타내고 있다. 실행 후 마이크로소프트 오피스의 파워포인트를 실행하고 PPT 리모컨을 누르면 파워포

인트를 컨트롤 할 수가 있다.



그림 15. 터치 마우스의 실행
Figure 15. Touch Mouse execution

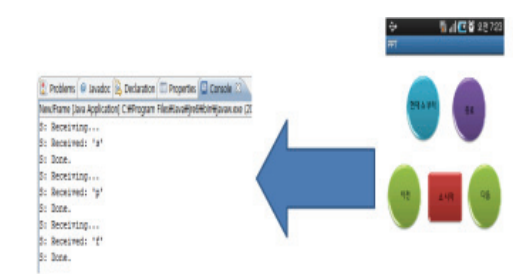


그림 16. PPT 리모컨의 실행
Figure 16. PPT Remote Execution

5. 결 론

본 논문에서는 무선 인터넷을 이용하여 스마트폰과 노트북 사이의 가상 숫자 키패드를 구현하여 기존의 경량화된 노트북에서 사라진 숫자키패드를 대신할 수 있음을 보였다. 현재 우리나라는 스마트폰 보급률이 2000만대 이상이 보급되면서 무선 인터넷을 이용할 수 있는 공간이 확대 되었고, 노트북을 이용하는 사람 중에 스마트폰을 이용하는 사람들도 많이 늘어나게 되었다. 논문에서 제시한 무선 인터넷 가상 키패드를 구현하여 어느 정도 실현 가능성을 보여 줬다. 또한 어플리케이션의 장점은 하나의 기능이 아닌 통합적인 기능으로 사용자의

편의도 크게 도모할 수 있었다. 향후 연구로는 추가적으로 멀리서도 자신의 컴퓨터와 관리할 수 있는 원격지원 기능을 구현하여 간단한 숫자 키패드를 제어로 시작하여 컴퓨터 전체를 제어하는 통합적인 프로그램을 개발하고자 한다.

References

- [1] Y. S. Hangu, lkeyboard Project Retrieved from <http://code.google.com/p/hangulkeyboard/>, 2009.
- [2] H. W. Kim, and P. Park, *Usability evaluation of android-based smart phone soft keyboard*, Korean Institute of Industrial Engineers, Spring Joint Conference, pp. 1-6, 2010.
- [3] D. Lysytskyy, and A. Pavliv, *Virtual keyboard with visually enhanced keys*, Washington, DC: U.S. Patent and Trademark Office, U.S Patent Application No. 2, pp. 612-873, 2010.
- [4] B. Obermaier, and G. Pfurtscheller, *Virtual keyboard controlled by spontaneous EEG activity*, IEEE Trans. Neural Syst. Rehab. Eng, Vol. 11, pp. 422-426, 2003.
- [5] T. Buttons, *Thick buttons usability tests*, retrieved from <http://thickbuttons.com/>, 2009.
- [6] P. Meroni, E. Pagani, And G. P. Rossi, *An opportunistic platform for android-based mobile devices*, MobiOpp 10, ACM, 2010.
- [7] Android. <http://www.android.com> Retrieved March 1st, 2011.
- [8] M. Lee, And W. Woo, *ARKB: 3D vision-based augmented reality keyboard*, International Conference on Artificial Reality and Telexistence, pp. 54-57, 2003.
- [9] <http://www.realvnc.com/docs/rfbproto.pdf>, reviewed on June 20th, 2011.

- [10] T. Richardson, S. F. Quentin, K. R. Wood and A. Hopper, *Virtual network computing*, Reprint from IEEE Internet Computing Vol. 2, No. 1, January/February, 1998.
- [11] W. Szeto, Y. Iraqi, and R. Boutaba, *A multi-commodity flow based approach to virtual network resource allocation*, in Proceedings of the IEEE Global Telecommunications Conference, pp. 3004-3008. 2003.

스마트폰을 이용한 네트워크 가상키보드 설계 및 구현

전병찬¹, 조상엽², 이상정³

¹청운대학교 교양학부

²청운대학교 인터넷학과

³순천향대학교 컴퓨터공학과

요 약

최근의 성능이 좋고 무게도 가벼운 노트북들이 많이 나오고 있다. 노트북의 크기를 줄이기 위해서는 키보드의 배열이 중요한데 이러한 키보드는 일반 컴퓨터에서 사용하는 108키의 방식으로는 크기를 줄이는데 힘들다. 따라서 대부분 가벼운 경량 노트북과 넷북의 경우 과감하게 기존의 108키 키보드의 오른쪽에 배열되어 있는 숫자 키패드를 제거하여 키의 배열을 최소화하는 방법으로 크기를 줄이고 있는 추세이다. 본 논문에서는 경량화를 거쳐 크기가 줄어든 노트북과 넷북에서 기존의 숫자 키패드를 사용하던 사용자들의 불편함을 덜어주기 위해 안드로이드 스마트폰을 사용하였다. 노트북과 안드로이드 스마트폰을 무선 네트워크로 연결하여 스마트폰에 가상으로 숫자 키패드를 구현하는 시스템을 제안한다.

감사의 글

본 논문은 청운대학교의 2013년도 학술연구조성비 지원을 받았음.



Byoung-Chan Jeon received the bachelor's degree in the Department of Computer Science from the Hanbat National University in 1989. He received the M.S. degree in the Department of Computer Science from Suwon University in 1994. He received the Ph.D. degree in the Department of Computer Science Engineering from Soonchunhyang University in 2001. He is currently a professor in the Division of Liberal Arts at Chungwoon University, Korea. His current research interests include computer architecture, network Application, Embedded systems. He is a member of the KKITS.

E-mail address : jbc66@chungwoon.ac.kr



Sang-Jeong Lee received the B.S., M.S., Ph.D. in the Department of Electronic Engineering from the Hanyang University in 1983, 1985 and 1988, respectively.

He is currently a professor at the Department of Computer Science and Engineering in Soonchunhyang University, Korea. His current research area include computer architecture, network application and embedded systems.

E-mail address: sjlee@sch.ac.kr



Sang Yeop Cho received the bachelor's degree in the Department of Computer Engineering from the Hannam University in 1986. He received the M.S. degree and the Ph.D. degree in the Department of Computer Engineering from Chungang University in 1988 and 1993, respectively. He is currently a professor in the Department of Internet at Chungwoon University, Korea. His current research interests include artificial intelligence, intelligent systems, fuzzy sets. He is a member of the KKITS.

E-mail address : sycho@chungwoon.ac.kr