

# 무선 네트워크 기반의 온라인 실시간 게임 콘텐츠 설계 및 구현

## Design and Implementation of Wireless Network based On-line Realtime Game Contents

김유두\*, 문일영\*, 조성준\*\*

Yu-Doo Kim\*, Il-Young Moon\* and Sung-Joon Cho\*\*

### 요 약

현재의 PC 게임은 인터넷을 이용한 온라인 게임이 주를 이루고 있다. 하지만 모바일 분야에서는 네트워크를 활용한 게임이 전무한 실정이다. 이는 무선 인터넷 환경에서 모바일 네트워크의 낮은 속도 및 모바일 기기의 저사양이 가장 큰 문제로 분석되고 있다. 하지만 3.5세대의 HSDPA(High Speed Downlink Packet Access) 방식의 서비스가 상용화 되고 단말기의 성능이 향상되면서 이러한 문제는 해결되어가고 있다. 이에 본 논문에서는 무선 인터넷 환경에서 모바일 네트워크의 활용을 위한 모바일 온라인 실시간 게임을 구현하였다.

### Abstract

Recently PC games are trending toward online. But, mobile games are not using network actively. Because there are two problems that low speed of mobile network and low specification of mobile device in wireless internet environment. But, solving problems by these started HSDPA services and increased performance of devices. In this paper, we implement mobile on-line real time game for develop mobile network in wireless internet environment.

Key words : Wireless network, WIPI, Socket, JAVA, Online game,

### I. 서 론

국내 무선 인터넷은 유선 인터넷의 발달과 더불어 빠르게 성장하고 있으며, 이동통신사는 수익을 높이기 위하여 1세대 단순 통화에서 현재의 영상통화 서비스 까지 무선 인터넷의 빠른 발전을 위해 힘쓰고 있다[1],[2]. 이는 싱글플레이(네트워크를 활용하지 않고 혼자서 즐기는 방식) 위주의 게임이 주를 이루고 있는 현재의 상황이 조만간 무선 인터넷을 활용한

무선 네트워크 게임이 주를 이루는 형태로 변화 될 것으로 예측되고 있다. 이에 본 논문에서는 커플이 동시에 즐길 수 있는 모바일 게임을 구현함으로써 다가올 미래의 주력이 될 무선 네트워크 게임에 대한 기술을 보이고자 한다.

무선 네트워크 게임 구현을 위한 기술은 크게 서버와 클라이언트로 구분되며 서버는 ANSI C로 구현하였고, 클라이언트는 국내 무선 인터넷 표준 플랫폼인 WIPI(Wireless Internet Platform for Interoperability)

\* 한국기술교육대학교 인터넷미디어공학부(School of Internet Media Eng., Korea University of Technology and Education)

\*\* 한국항공대학교 항공전자 및 정보통신공학부(School of Avionics and Telecommunication Eng., Korea Aerospace University)

· 제1저자 (First Author) : 김유두

· 접수일자 : 2007년 3월 13일

로 구현하였다[3],[4].

논문의 구성은 다음과 같다. 2장에서는 구현을 위한 기반기술을 알아보고, 3장에서는 구현을 위한 시스템 설계에 대한 내용을 담고 있으며 4장에서는 시스템의 실제 구현된 모습을 보여주며 마지막으로 결론을 맺는다.

## II. 구현기술 소개

무선 네트워크 게임 개발을 위해서는 휴대폰에 동작하는 프로그램을 만들기 위한 WIPI 플랫폼 및 통신을 위한 TCP/IP 소켓 기술이 필요하다[3],[5].

### 2-1 WIPI

WIPI는 한국무선인터넷표준화 포럼의 무선인터넷 플랫폼 표준이며 한국정보통신기술협회(TTA)에 의해 TTA단체 표준 TTAS.KO-06.0036으로 채택된 이동통신 단말기용 응용프로그램의 실행 환경에 대한 표준 규격이다[5]. WIPI 규격은 플랫폼 이식성을 높이기 위한 표준화된 하드웨어 추상화 계층인 HAL (Handset Adaptation Layer)과 표준화된 플랫폼 호환성을 제공하여 다양한 응용 프로그램 개발을 촉진하기 위한 기본 응용 프로그래밍 인터페이스로 구성된다. 플랫폼 개발 언어로는 C 언어 및 자바 언어를 모두 지원하도록 규격화 하였으며 본 논문에서는 WIPI-JAVA를 이용하여 구현하였다.

WIPI 규격에서 정의하는 모바일 표준 플랫폼은 그림 1과 같은 개념적 구조를 갖는다. 그림 하단에 있는 단말기 기본 소프트웨어란 간단한 단말기 운영체제와 통신 기본 기능 및 각종 디바이스 드라이버가 포함된다. 단말기 기본 소프트웨어는 제조사에 따라 기능이나 규모가 다양할 수 있지만, 여기에 HAL 계층을 두어 플랫폼이 바라보는 단말기 소프트웨어를 추상화 할 수 있도록 하였다. 또한 Basic API 계층은 응용프로그램 개발자 입장에서 바라본 플랫폼이다.



그림 1. WIPI 플랫폼의 구조도  
Fig. 1. Structure of WIPI platform

### 2-2 TCP/IP 소켓

소켓(Socket)이란 TCP/IP 프로토콜을 이용하여 프로그래밍 언어에 대한 제한 없이 데이터를 주고받기 위해 만들어진 공통 API(Application Programming Interface)이다. 이를 활용하면 C, JAVA 등 언어에 상관없이 각 언어에 맞게 제공되는 API를 활용하여 TCP/IP 프로토콜의 네트워크를 통해 데이터를 주고 받을 수 있다.

그림 2에서는 JAVA에서의 소켓의 이용 예를 보여 주고 있다. 소켓은 서버와 클라이언트의 두 부분으로 나뉘지는데 서버에서는 소켓생성, 대기, 클라이언트와의 연결, 데이터 전송의 단계로 이루어지며 클라이언트에서는 소켓생성, 서버와의 연결, 데이터 전송의 단계로 이루어진다.

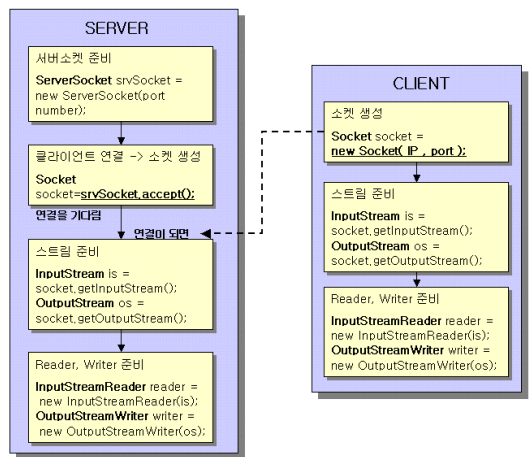


그림 2. JAVA에서의 소켓 프로그램  
Fig. 2. Socket programs in JAVA

### Ⅲ. 시스템 설계

무선 네트워크는 기존 유선 네트워크 게임과 같은 서버와 클라이언트의 구조로 이루어져 있다. 유선 네트워크와의 큰 차이점은 클라이언트와 서버로의 접속이 이동통신사의 게이트웨이를 거친다는 것이다. 모바일 커플 게임은 게임 구동을 위한 서버와 클라이언트 외에 데이터 저장을 위한 데이터베이스 및 데이터 교환을 위한 통신규약을 지정하여야 한다.

#### 3-1 시스템 구조

그림 3은 시스템의 전체적인 구조를 보여주고 있다. 국내 이동통신망은 휴대폰과 게임서버의 데이터 교환을 위한 중계의 역할을 하고 있으며 연결된 서버와 클라이언트는 다음과 같은 기능을 한다.

- 메시지 처리 서버 : 클라이언트, 게임 엔진으로부터 받은 메시지를 처리한 후 다시 클라이언트, 게임 엔진에 전송하고 데이터베이스를 수정하는 기능을 한다.
- 데이터베이스 서버 : 게임 사용자/아기의 상태 및 아이템 정보 등이 저장되어 있다.
- 게임엔진 서버 : 일정 시간에 따른 캐릭터의 상태를 변화시킨다. 컨트롤러를 제공하여 컴퓨터에 의한 자동 외에 관리자가 수동으로 값을 조정한다.
- 2개의 모바일 클라이언트 : 게임이 이루어지는 부분으로 핸드폰에서 실행될 부분이다.



그림 3. 모바일 커플게임의 시스템 구조  
Fig. 3. System structure of mobile couple game

#### 3-2 데이터베이스

표 1은 게임에 필요한 데이터를 저장하는 데이터베이스의 설계를 나타낸다.

- Member : 게임 이용자의 접근 ID와 포인트 등 게임 이용자의 개인정보를 저장한다.
- Character : 게임 이용자에 의하여 육성되는 캐릭터에 대한 정보를 저장한다.
- Item : 게임을 진행 하는 데에 필요한 모든 아이템을 저장한다.
- Hasitem : 게임 이용자 별 보유중인 아이템에 대한 관리를 한다.

표 1. 데이터베이스 설계

Table. 1. Database design

Table	용도
Member	회원의 정보를 관리한다.
Character	캐릭터 정보를 관리한다.
Item	게임 아이템을 관리한다.
Hasitem	보유중인 아이템을 관리한다.

#### 3-3 통신규약

표 2는 서버와 클라이언트간의 데이터의 구분을 위한 메시지 형태를 보여준다. 메시지는 크게 회원가입, 로그인 등의 게임 시작 전에서의 처리를 위한 100번대와 클라이언트에서 서버로 전달하는 메시지인 300번대, 그리고 서버에서 클라이언트로 전달하는 메시지인 700번대 메시지로 구분한다.

메시지를 주고받는 서버와 클라이언트는 이 형태에 따라 명확하게 데이터를 처리 한다.

표 2. 메시지 구분

Table. 2. Message section

메시지	용도
100번대	회원가입, 로그인
300번대	클라이언트 -> 서버
700번대	서버 -> 클라이언트
chr(2)	각 메시지의 구분기호

표 3은 실제로 주고받는 메시지의 예를 통하여 캐릭터의 상태 메시지를 보여주고 있다. 현 상태는 서버에서 클라이언트로 보내는 것이며 클라이언트의

요청에 따라 캐릭터의 상태를 변화하고 그 결과를 알려주는 역할을 하는 메시지이다.

표 3. 캐릭터 상태 메시지

Table. 3. Character state message

<p>701 : stat 변화(아기성별, 자기여부, 응가여부, 기분, 키, 몸무게, 지능, 걷기, 건강, 인성, 청결, 친밀1, 친밀2, 배고픔, 레벨)</p> <p>protocol chargender sleep dung feel height weight intelligence walking power humanism cleanness intimacy1 intimacy2 hungry level</p> <p>701 아기성별 자기여부 응가여부 기분 키 몸무게 지능 걷기 건강 인성 청결 친밀(아빠1 or 엄마2) 배고픔 레벨</p> <p>ex)701 1 0 1 1 50 3 10 10 10 10 10 10 10 1</p>
---

#### IV. 시스템 설계 및 구현

구현하는 무선 네트워크 기반의 커플 게임은 동시에 두 명의 사용자가 하나의 캐릭터를 육성하는 게임으로 이벤트가 발생 할 때마다 두 클라이언트의 정보의 동기화가 이루어져야 한다. 따라서 클라이언트의 이벤트가 발생 할 때마다 서버가 처리 후 두 개의 클라이언트에 신호를 전달하여 준다.

서버는 클라이언트에서 들어오는 메시지를 받아 적절한 처리를 해 주는 게임엔진의 역할을 함께 하고 있으며 이 처리를 관리자가 수동으로 할 수 있도록 게임 컨트롤러를 MFC(Microsoft Foundation Class)로 구현하였다[7].

##### 4.1 서버/엔진

게임서버는 ANSI C로 구현하였으며 여기에서는 클라이언트로부터 받아온 메시지를 분석하여 해당 작업을 처리하고 결과를 데이터베이스에 저장하는 등 게임의 중요한 처리가 모두 포함된 엔진 역할을 하고 있다.

네트워크 게임의 특성 상 게임의 동작에 필요한 모든 처리는 클라이언트가 아닌 서버에서 처리하여 각각의 클라이언트에 보냄으로써 클라이언트간의 동기화를 이루어 낼 수 있다.

표 4는 서버에서 메시지를 수신하기 위한 루틴을 보여주고 있다. 서버는 메시지를 수신하여 게임 컨트롤러에서 온 메시지, 새로운 사용자의 로그인요청, 로그인된 사용자의 이벤트 처리 요청으로 구분하여 처리한다.

표 4. 메시지 수신 대기

Table. 4. Waiting for message receive

<pre>while(1) // 메시지 대기 루프 { // 게임 컨트롤러에서 메시지가 오면 if (FD_ISSET(controller_sock, &amp;tr_fds)) // 메시지 처리 후 컨트롤러에 결과 전송 // 새로운 사용자의 로그인 요청 else if (FD_ISSET(game_listen_sock, &amp;tr_fds)) // 새로운 사용자 로그인 처리 // 로그인된 사용자의 이벤트 처리 요청 else // 사용자에 따른 게임 이벤트 처리 }</pre>
---

게임 컨트롤러를 통한 메시지는 관리자의 임의 지정 값이므로 특별한 처리 없이 바로 적용한 후 컨트롤러에 결과를 전송해 준다. 일반 사용자의 메시지 처리는 새로운 사용자의 로그인 요청인지와 로그인이 되어있는 사용자의 게임 이벤트 처리 메시지인지를 구분하여 처리한다.

표 5에서는 새로운 사용자가 접속하여 로그인을 요청할 경우의 처리를 보여준다.

두 명의 사용자가 함께 로그인인 된 상태에서만 진행이 되는 게임의 특성으로 인하여 A사용자와 B사용자가 모두 로그인 될 때까지 대기하며 처리하고 세 번 째 로 요청을 하면 잘못된 요청을 알린다. 표 6은 로그인이 성공되어 현재 실행상태에 있는 사용자들의 메시지가 수신 되었을 경우 이벤트 메시지를 게임 엔진에 보내 주는 것이다.

##### 4.2 게임 컨트롤러

그림 4는 관리자가 임의로 게임 동작에 대한 조정을 할 수 있도록 하는 게임 컨트롤러로써 이를 통해 게임 동작에 대하여 관리자가 임의적으로 이벤트 발

생 및 캐릭터 상황을 변화 할 수 있도록 하였다. 이를 통해 게임 동작 테스트 및 난이도 조절 등이 가능하도록 하였다.

표 5. 로그인 요청

Table. 5. Login request

```
// 로그인 대기자 리스트에 등록
if (con_user_cnt == 0) // A 사용자
{
    printf("Hi User1Wn");
    con_table[0].bebe_link = bebe_list->next;
    con_table[0].gender = 1;
    con_table[0].mySock = accept_sock;
    con_table[0].toIndex = 1;
    strcpy(con_table[0].charid, userid);
    con_user_cnt = 1;
}
else if (con_user_cnt == 1) // B 사용자
{
    printf("Hi User2Wn");
    con_table[1].bebe_link = bebe_list->next;
    con_table[1].gender = 2;
    con_table[1].mySock = accept_sock;
    con_table[1].toIndex = 0;
    strcpy(con_table[1].charid, userid);
    con_user_cnt = 2;
}
else // A,B 모두 접속 후 다시 접속 요청 할 경우
{
    printf("max connection overflowWn");
    exit(1);
}
```

표 6. 이벤트 메시지 전송

Table. 6. Send event message

```
// 로그인 대기자 리스트에 등록
if (con_user_cnt == 0) // A 사용자
{
    printf("Hi User1Wn");
    con_table[0].bebe_link = bebe_list->next;
    con_table[0].gender = 1;
    con_table[0].mySock = accept_sock;
    con_table[0].toIndex = 1;
    strcpy(con_table[0].charid, userid);
    con_user_cnt = 1;
}
else if (con_user_cnt == 1) // B 사용자
{
    printf("Hi User2Wn");
    con_table[1].bebe_link = bebe_list->next;
    con_table[1].gender = 2;
}
```

```
con_table[1].mySock = accept_sock;
con_table[1].toIndex = 0;
strcpy(con_table[1].charid, userid);
con_user_cnt = 2;
}
else // A,B 모두 접속 후 다시 접속 요청 할 경우
{
    printf("max connection overflowWn");
    exit(1);
}
```

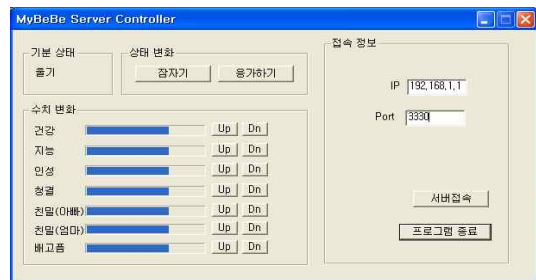


그림 4 관리자용 게임 컨트롤러  
Fig. 4. Game controller for administrator

### 4.3 모바일 클라이언트

그림 5는 모바일 클라이언트의 실행화면을 보여주는 것으로 게임 이용자가 휴대폰을 통하여 보게 되는 화면이다.

같은 캐릭터를 동시에 육성하고 있는 두 사용자의 화면을 각각 보여주는 것으로 B 사용자가 아기의 방을 청소하고 있는 동안 A 사용자는 다른 작업 지시를 할 수 없도록 메시지를 표시하여 두 사용자가 동시에 작업을 지시하는 문제점이 생기지 않도록 하였다.



그림 5. 게임 실행 모드  
Fig. 5. Game playing mode

이와 같이 다른 이벤트에 대해서도 동기화 작업을 수행함으로써 두 사용자가 한 캐릭터를 동시에 육성할 수 있도록 구현하였다.

## V. 결 론

본 논문에서 모바일 클라이언트는 국내에서 개발된 통합 플랫폼인 WIPI-JAVA를 사용하고 서버는 ANSI C로 구현하여 이 종의 언어에서도 무리 없이 시스템이 운영되는 것을 보여주었다. 이는 모바일 네트워크 게임 개발의 다양한 환경에 따라 다른 언어를 선택해야 할 경우를 위한 대비가 될 수 있으며 서버와 클라이언트에서의 각자 최고의 성능을 위하여 다른 언어를 선택할 경우에 대한 부담감을 덜어주었다. 또한 실질적인 이동통신사의 게이트웨이를 이용하여 게임을 테스트함으로써 현 상황에서도 무리 없이 모바일 네트워크 게임이 동작할 수 있음을 보였다. 하지만 장시간 플레이를 할 경우 약간의 지연현상이 발생하는 경우가 있었다. 이는 3.5세대 이동통신이 본격적으로 상용화되면 해결 될 것으로 예상하고 있다. 추가적으로 휴대폰 대기화면 상태에서 항상 실행되어 있는 형태로 클라이언트를 구현한다면 육성 게임의 특징을 더욱 살릴 수 있을 것이다.

## 감사의 글

본 논문은 산업자원부 한국산업기술평가원 지정 한국항공대학교 부설 인터넷정보검색 연구센터의 지원에 의함.

## 참 고 문 헌

- [1] 유용덕, 박충범, 최훈, 김우식, “위피 응용프로그램 개발환경 설계 및 구현”, *정보처리학회 논문지 C 제12-C권 제 5호*, 2005.10.
- [2] 박현주, “모바일 인터넷 시장 동향”, *정보통신정책 제19권 7호 통권 414호*, 2007.04
- [3] 모바일 표준플랫폼 WIPI 2.0, WISFS.K-05-002.
- [4] 위피개발자포럼, [developer.wipi.or.kr](http://developer.wipi.or.kr).
- [5] 한국무선인터넷표준화포럼, [www.kwisforum.org](http://www.kwisforum.org).

- [6] 조경민, 백창우, 최영호, 윤경훈, 윤상배, “TCP/IP 소켓 프로그래밍”, *한빛미디어*, 2005.07.
- [7] MFC 프로그래밍, <http://yalli.new21.org>.

## 김 유 두 (金裕斗)



2007년 2월 : 한국기술교육대학교 인터넷공학(공학사)  
 2007년 3월~현재 : 한국기술교육대학교 대학원 정보미디어공학과 (석사과정)  
 관심분야 : 무선 네트워크, 모바일 콘텐츠, 스마트카드

## 문 일 영 (文日永)



2000년 2월 : 한국항공대학교 항공통신정보공학과 (공학사)  
 2002년 2월 : 한국항공대학교 대학원 항공통신정보공학과 졸업(공학석사)  
 2005년 2월 : 한국항공대학교 대학원 정보통신공학과 졸업(공학박사)  
 2004년 ~2005년 : 한국정보문화진흥원 선임연구원  
 2005년 3월~현재 : 한국기술교육대학교 인터넷미디어공학부 조교수  
 관심분야 : 무선 인터넷 응용, 무선 인터넷, 모바일 IP

## 조 성 준 (趙成俊)



1969년 2월 : 한국항공대학교 항공통신공학과 (공학사)  
 1975년 2월 : 한양대학교 대학원 (공학석사)  
 1981년 3월 : 오사카대학 대학원 (공학박사)  
 1972년 8월 ~ 현재 : 한국항공대학교 항공전자 및 정보통신공학부 교수  
 관심분야 : 이동통신, 무선통신, 환경전자공학, 이동무선 인터넷