

Game Design of Online Sutda using 3-Tier Structure

Syungog An*, Sung-Ho An**, Jinyoung Jung***, Soo-Kyun Kim****

Abstract

This paper introduces a program structure that minimizes the client's business logic by reducing the client's specifications as a way of minimizing the limitations of online game platforms. The objective of this thesis is to design an online game that is based on a 3-tier client/server structure, and thus using this 3-tier structure to ultimately create an online Sutda game that anyone can enjoy. The progression of newly released games is highly based on the mobile market and the number of online games being launched is decreasing. As the mobile game market grows, more and more multi-platform games are being developed and target platforms within online games are greatly diversifying.

Through these situations, this thesis minimizes the client's business logic through a 3-tier structure of application-middleware-database, and also proposes an online game client/server that allows three or more users to participate simultaneously through the server.

▶ Keyword : Sutda game, Database, Client, Online Game, 3-Tier

1. Introduction

다양한 게임들의 출시를 통해 성장해온 게임 시장은 최근 통신기술이 발달함에 따라 모바일 게임이 그 주를 이루게 되었고, 실제 컴퓨터뿐만 아니라 다양한 플랫폼에서도 모바일 게임을 즐기는 사용자가 늘어가는 추세다.

인터넷을 통해 사용자는 멀리 떨어져있는 서버를 통하여 다른 이용자와 함께 즐기는 게임 방식을 가진 온라인 게임 환경을 클라이언트/서버 시스템이라 한다. 온라인 게임에서는 많은 수의 사용자를 감당할 수 있는 클라이언트기술과 서버기술이 진화해 가고 있다.

본 논문은 3-tier 구조[1,2,3,4,5,6,7]를 이용하여 어플리케이션-미들웨어-데이터베이스의 3-tier 구조를 통해 클라이언트의 비즈니스 로직을 최소화하고, 실제 셋 이상의 이용자가 서버를 통해 이용이 가능한 온라인 게임 클라이언트/서버에 대해 제안한다.

본 논문의 구성은 다음과 같이 되어있다. 2장에서는 관련 연구로써 2-Tier 아키텍처와 3-Tier 아키텍처에 대해 설명한다. 3장에서는 클라이언트 및 프로토타입 구성에 대해 소개한다. 4장에서는 미들웨어 설계에 대해 설명하고, 5장에서는 실행결과에 대해 소개하고 결론을 내린다.

• First Author: Syungog An, Corresponding Author: Soo-Kyun Kim
*Sung-Ohk An (sungohk@pcu.ac.kr), Dept. of Game Engineering, Paichai University
**Sung-Ho An (minlog@empal.com), Dept. of Game Engineering, Paichai University
***Jinyoung Jung (jyung@hit.ac.kr), Dept. of Bio-Information, Daejeon Health Science College
****Soo-Kyun kim (kimsk@pch.ac.kr), Dept. of Game Engineering, Paichai University
• Received: 2015. 04. 23, Revised: 2015. 05. 27, Accepted: 2015. 07. 04.

II. Related Works

1. Related Works

1.1 2-Tier 아키텍처

그림1에서와 같이 2계층 클라이언트/서버 구조는 네트워크의 발전과 PC와 같은 지능형 터미널이 도입되면서 기존 더미 터미널이 할 수 없었던 기능을 클라이언트로 전이한 방식을 말한다. 이 방식은 클라이언트와 서버가 물리적으로 독립된 시스템에 존재하는 형태로 구성되었다. 시스템의 구성이 2계층 클라이언트/서버 방식[8]으로 발전함으로써, 기존에 서버의 부하가 상당부분 줄어들게 되었으며, 클라이언트가 대량의 자료를 접근하여 가공할 수 있게 됨으로서 더욱 정교한 자료 분석과 사용자의 정보 제공이 가능해 졌다.

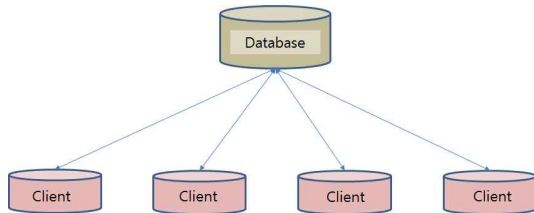


Fig. 1. 2-Tier Architecture

먼저, 발전방안의 기술을 살펴보면 이해 그룹의 장점과 단점을 살펴보면,

1.2 3-Tier 아키텍처

그림2에서와 같이 3계층 구조란 2계층 구조의 문제점을 해결하기 위해 물리적으로 프레젠테이션 로직과 데이터베이스를 미들웨어에 비즈니스 로직을 구현하여 연결하는 형태의 구조이다. 3계층 구조의 장점은 어플리케이션에서는 프레젠테이션 로직만 구현함으로써 어플리케이션의 성능이 향상되고, 확장성과 재사용이 용이하여 유지보수가 쉽다는 점이 있다. 본 논문에서는 3계층 구조를 이용하여 게임을 적용한 온라인 게임의 설계와 프로토타입을 구현하는데 그 목적이 있다 [2].

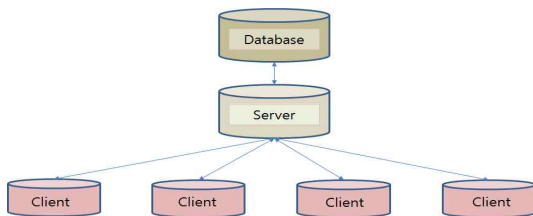


Fig. 2. 3-Tier Architecture

1.3 프로토콜 설계

프로토콜의 개념을 이용한 프로토콜 클래스를 사용하여 아래의 ProtocolMessage 클래스와 같은 형태로 전송될 데이터

의 앞부분에 프로토콜을 추가하여 어플리케이션과 미들웨어간의 정보 교환을 하도록 설계한다.

```
class Protocol{
    static final int PROTOCOL_EX_1 = 100;
    static final int PROTOCOL_EX_2 = 200;
    static final int PROTOCOL_EX_3 = 300;
    static final int PROTOCOL_EX_4 = 400;
}

new ProtocolMessage(프로토콜 , ... , ... , ...);
```

III. Game Design

1. 클라이언트 설계

본 절에서는 프로그램에 사용된 구조와 주요 프로그램의 슈도코드에 대한 내용으로 구성되며, 3계층 구조의 어플리케이션으로서 '온라인 섯다'라는 게임을 적용한 시스템 설계와 그 프로토타입의 구성에 대해 설명한다.

게임을 적용하기 위해 사용자의 입장에서 어플리케이션을 설계하였고, 소켓 프로그래밍을 이용하여 데이터 송수신 처리를 하였고 원활한 데이터 네트워크가 이루어지도록 하기 위하여 프로토콜의 개념을 이용하여 설계하였으며, 프로그램의 진행 프로세스는 그림3과 같이 구성한다.

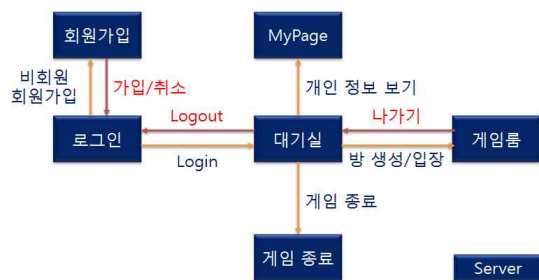


Fig. 3. Program Architecture

3계층 구조에서 어플리케이션 로직 부분으로서 실제 프로그램 사용자가 사용하게 되는 인터페이스 부분으로서 사용자의 입력내용과 서버로부터 받은 데이터의 출력기능을 담당하고 그 이외의 비즈니스 로직은 미들웨어에서 수행하도록 설계하여 어플리케이션의 성능을 향상시키고 수정과 재사용이 용이하도록 하였다. 네트워크는 메인 스레드에서 입력과 송신을 하며 미들웨어로부터 데이터 수신 받기 위해 하나의 스레드를 생성하여 수신 받은 데이터를 처리하도록 설계 하였다. 데이터를 수신하는 스레드의 형태는 다음과 같다.

```
public void run() {
    boolean flag = true;
    m_ois = new ObjectInputStream
        (m_socket.getInputStream());
    ProtocolMessage pm = null;
    while (flag) {
        pm = (ProtocolMessage) m_ois.readObject();
        switch (pm.getProtocol()) {
            case Protocol.IDCHECK : break;
            case Protocol.SIGNUP : break;
            case Protocol.LOGIN : break; }}}
```

2. 회원 가입 설계

네트워크상에서 상대가 사용자임을 인식하고 대화, 게임, 현재 상태 등을 파악하기 위해 회원가입기능을 추가하였고, 이를 통해 사용자에게 ID를 부여하고, 그림 4와 같이 로그인 할 수 있도록 설계한다.

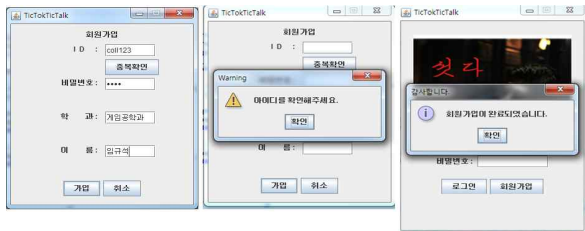


Fig. 4. Member registration design

3. 로그인 및 대기실 설계

회원 가입 시 입력 내용은 모두 미들웨어로 보내지며, 이후 미들웨어에서 보내오는 가입 가능 여부를 통해 회원 가입이 처리되었는지 알 수 있다. 마찬가지로 로그인시 로그인 화면에서 입력한 데이터들이 미들웨어로 보내지며, 미들웨어에서 보내오는 ID의 가입 유무와 비밀번호 확인 절차를 통한 결과값을 수신하여 대기실의 입장 여부를 결정할 수 있도록 설계한다.

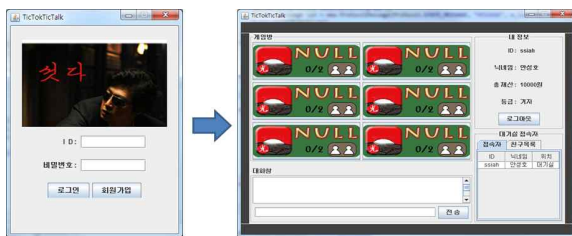


Fig. 5. Waiting room design

그림 5와 같이 대기실에 입장 시, 미들웨어로 신호를 보내 미들웨어가 로그인 사용자의 정보, 게임 룰 정보, 접속자 정보 등을 접속 사용자 모두에게 송신, 클라이언트는 수신 받은 데이터를 쓰레드에서 처리하여 사용자에게 보이도록 설계한다. 대기실에서의 대화 기능 또한 미들웨어로 대화 내용을 송신 접속 중인 모든 사용자에게 대화 내용을 전송하여 대화창에 내용이 출력되도록 설계한다(그림6참고).



Fig. 6. Chat window design

IV. Middleware Design

1. 미들웨어 구조

3계층 구조의 비즈니스 로직 부분으로서 클라이언트로부터 수신된 입력 데이터를 처리하여 데이터베이스와 클라이언트로 송신하는 데이터 연산기능을 담당하며 연결된 클라이언트마다 하나씩 쓰레드를 생성하여 1:1 대응을 할 수 있도록 멀티 쓰레드로 구현을 하였고, 아래의 HashMap 구조를 이용 접속 사용자의 ID와 그에 대응하는 미들웨어의 쓰레드를 key값과 value로 가짐으로서 접속 중인 특정 사용자, 또는 모든 사용자에게 데이터를 송신할 수 있도록 그림7과 같은 형태로 설계하였다.

```
HashMap<Key(UserID), Thread> clientList
    = new HashMap<Key(UserID), Thread>();
```

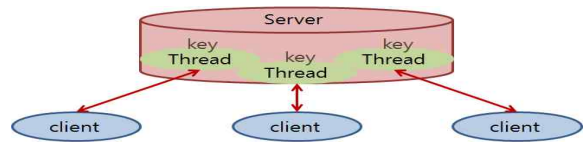


Fig. 7. Relationship between thread of middleware and client

2. 데이터 송신/수신 메서드

미들웨어의 구현에 필요한 메서드의 경우 크게 수신 메서드와 송신 메서드 2가지로 나누어 관리하였다. 수신 메서드의

경우 데이터의 연산에 필요한 비즈니스 로직과 데이터베이스의 데이터 수신과 그 연산처리를 담당하였고, 송신 메시드의 경우 위 내용의 clientList로 저장된 HashMap을 이용하여 송신 메시드와 지정 송신 메시드로 구분하였으며 그 내용은 아래와 같다.

```

roar(msg) {
if(생성된 쓰레드가 존재할 경우){
for(int i=0;i<생성된 쓰레드의 총 개수;i++){
메서드를 실행시킬 쓰레드 =
    생성된 쓰레드의 맵(i번째 key값);
t.allMessage(전송할 데이터, 데이터를 보낸 사용자의 ID);
}}}

select(key, msg){
메서드를 실행시킬 쓰레드 =
    생성된 쓰레드의 맵(보낸 사용자의 key값);
t.selectMsg(전송할 데이터, 데이터를 보낸 사용자의 ID);
}
    
```

본 연구에서 사용한 3-Tier 구조에서의 데이터베이스는 JDBC를 사용하여 접근하였으며 데이터베이스의 위치는 테스트 편의상 물리적으로 미들웨어와 같이 위치하도록 설계 하였다.

V. Middleware Design

1. 게임의 방법 및 설계

셋다는 화투를 이용한 노름 중 하나이다. 화투에서 피와 오동(11월) 및 비(12월)를 제외한 20장을 가지고 두 명 이상이 경기를 한다. 잘 섞은 더미에서 각자 2장씩 패를 나눠 받아 겨루는데 2장의 패의 각 조합에 따라 높은 조합과 낮은 조합 및 특수 조합이 있고 이러한 조합을 '족보'라고 부른다. 굉장히 간단한 규칙을 가지고 있고 심리적인 싸움이 무척 중요하다.

프로토타입의 3계층 구조의 프로그램[8,9]에 게임을 적용하기 위해 카드게임의 일종인 '셋다 게임'을 설계하여 적용하였고, 게임의 진행순서는 그림8과 같다.

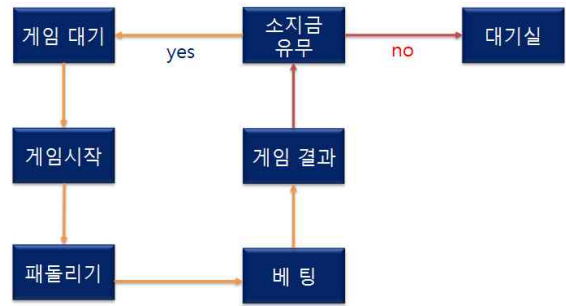


Fig. 8. Game Process

2. 게임 규칙 및 게임 룸 설계

우선 게임 룸에 입장한 사용자중 우선 입장한 사용자가 '선'을 가진다. 게임 시작 버튼을 통하여 게임을 시작하며 게임 시작 이후 자기가 소지한 카드를 확인한 뒤 베팅을 시작한다. 베팅은 비즈니스 로직의 단순화를 위하여 쿨, 하프, 다이, 올인 4가지로 구성하며 이후 족보와 같은 기타 규칙은 '셋다 게임'과 동일하게 구성하였다.

그림 9의 게임 룸 내에서는 모든 접속자가 아닌 게임 룸에 접속 중인 클라이언트에게만 데이터를 송/수신한다. 게임 룸의 방 번호별로 프로토콜개념[10]을 이용하여 게임 룸 간의 데이터 전송이 이루어지지 않도록 구분하여 처리한다. 게임 도중 베팅, 승리, 패배 등의 회원의 정보 변동사항이 있을 경우 미들웨어를 통해 데이터베이스의 갱신 데이터를 받아 클라이언트에 전송하여 접속자간의 정보를 지속적으로 동기화 시킨다.



Fig. 9. Game room

VI. Development Environment And Result

1. 개발 환경 및 개발 결과

온라인 셋다 프로그램 개발은 Java를 이용하였고, 게임을

서비스하기 위한 가입자의 정보, 접속자의 상태 및 게임 데이터를 저장하기 위하여 데이터베이스로 Oracle Database Express Edition 11g를 사용하였으며 os는 별도의 서버용 운영체제를 사용하지 않고 윈도우7을 사용하였다.

Table 1. Development environment

항목	내용
개발툴	Eclipse EE(luna)
데이터베이스	Oracle Database Express Edition 11g
개발 언어	Java JDK8.0, SQL

그림10과 같이 프로그램을 실행하면 가장먼저 로그인 화면을 볼 수 있다. 기존의 사용자일 경우 계정을 입력하여 접속할 수 있고 그렇지 않을 경우 회원가입 버튼으로 회원가입을 할 수 있는 화면으로 넘어간다.



Fig. 10. Scene Change

그림 11은 계정을 입력하여 로그인 하였을 경우 게임 룸으로 이동되며 접속 중인 사용자나 게임 룸의 상태를 확인 할 수 있으며 대기실의 사용자들과 대화가 가능하다.



Fig. 11. Change of condition in game room

게임 룸에 혼자 입장한 경우 상대사용자가 입장하기 전까지 대기 상태(그림12참조)가 되며 상대방이 게임에 입장하여 게임 룸 안에 2명의 사용자가 있을 경우 게임 시작 버튼이 활성화 되며 게임 규칙에 따라 게임을 이용할 수 있으며 사용자의 데이터는 데이터베이스에 저장되어 종료 후에도 유지된다.

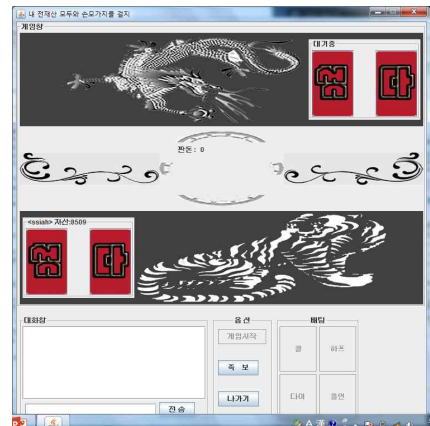


Fig. 12. Entrance to the game room

VII. Conclusions

본 논문은 온라인 게임 플랫폼의 제약을 최소화하기 위한 방안인 클라이언트의 사양을 줄이기 위한 방법으로 클라이언트의 비즈니스 로직을 최소화하는 프로그램 구조를 도입하였고, 이를 통해 3-tier 구조를 이용하여 최종적으로 누구나 즐길 수 있는 온라인 섯다 게임을 설계 하였다.

향후 연구에서는 3-tier 클라이언트/서버 시스템 구조를 기반으로 어플리케이션에서 이루어지는 연산 작업을 줄여 동일 플랫폼의 성능측면에서 플랫폼의 중속성을 최소화하여 어플리케이션의 원활한 작동이 이루어 질 수 있도록 3-tier 클라이언트/서버 시스템 구조를 최대한 활용할 수 있는 프로그램 구조에 대한 연구가 필요하다.

REFERENCES

- [1] Multitier architecture, http://en.wikipedia.org/wiki/Multitier_architecture
- [2] NAVER IT(data dictionary) <http://terms.naver.com/entry.nhn?docId=862171&cid=42346&categoryId=42346>
- [3] Jaeun Shin, "3-Tier Grid-based Cloaking Area Creation And Location Information Transmission Scheme for Privacy Protection in Location-Based Services", Kunkuk Graduate School of Information and Communications, Master Thesis, 2012
- [4] Jongwon Yang et al., "The Structure of Agent based 3-Tier Server/Client for Processing Context-Awareness on Ubiquitous Computing", Korean Society for Internet Information, 2004
- [5] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion

Stoica, and Matei Zaharia, "A View of Cloud Computing," Association for Computing Machinery, Communications of the ACM, Vol. 53, Iss.4, pp.50, 2010

[6] Donald Wolfe, "3-Tier Architecture in ASP.NET with C# tutorial", Amazon Digital Services, Inc., Jan. 2013

[7] NAVER IT (Data Dictionary) <http://terms.naver.com/entry.nhn?docId=865047&cid=42346&categoryId=42346>

[8] James Charles, Sylvester Kabajo and Kyewalabye, "Enhancing property rates management using N-TIER server architecture: 2-tier, 3-tier and/or N-tier client server architecture. The case of Kampala City Council", LAP LAMBERT Academic Publishing, March 29, 2012

[9] Dan Harkey, "Client/Server Survival Guide, 3rd Edition", Wiley 3 edition, January 25, 1999

[10] Doopedia, <http://terms.naver.com/entry.nhn?docId=1167759&cid=40942&categoryId=32851>

Authors



Syungog An received the Ph.D. degrees in Computer Science and Engineering from Korea University, Korea, in 1989. She is currently a Professor in the Department of Game Engineering, Paichai University.



Sung-Ho An received the B.S. degrees in Game Engineering from Paichai University, Korea,



Jinyoung Jung received the Ph.D. degrees in Computer Science and Engineering from Korea University, Korea, in 2002. He is currently a Professor in Dept. of Bio-Information, Daejeon Health Science College.



Soo-Kyun Kim received Ph.D. in Computer Science & Engineering Department of Korea University, Seoul, Korea, in 2006. He joined Telecommunication R&D center at Samsung Electronics Co., Ltd., from 2006 and 2008. He is now a professor at Department of Game Engineering at Paichai University, Korea. His research interests include multimedia, pattern recognition, image processing, mobile graphics, geometric modeling, and interactive computer graphics. He is a member of ACM, IEEE, IEEE CS, KACE, KMMS, KKITS and KIIT.