

멀티 유저 인터넷 게임

김형민, 김규년
울산대학교 컴퓨터 정보통신공학과
humanfly@cic.ulsan.ac.kr knkim@uou.ulsan.ac.kr

A MultiUser Internet Game

Hyung-Min, Kim, Kyu-Nyun, Kim
School of Computer Engineering & Information Technology, Ulsan University

요 약

멀티유저 인터넷 게임이란 서로 다른 장소에 있는 사람들과 인터넷이라는 가상의 공간에 모여서 하나의 게임을 즐기는 것을 말한다. 멀티유저 인터넷 게임은 현재 가장 각광받고 있으며 발전 가능성이 무한하다. 하지만 이 멀티유저 인터넷 게임을 위한 서버와 클라이언트의 구축이 만만하지는 않다. 본 논문에서는 멀티유저 인터넷 카드게임을 구현한다. 현재 발표되어 있는 여러 가지 멀티유저 인터넷 게임 서버들의 구조와 문제점들을 분석한다. 그리고 여러가지 멀티유저 인터넷 카드게임 서버 설계의 표준이 될 수 있는 메시지 전달 방식, 메시지의 구성 방법 등을 제시한다. 최종적으로 본 논문에서 제안한 방식으로 멀티유저 인터넷 카드게임을 구현할 것이다.

1. 서론

오늘날 컴퓨터의 급속한 발전과 더불어서 그에 따른 소프트웨어들도 쏟아져 나오고 있다. 그 소프트웨어들 사이에는 컴퓨터 게임이 적지 않은 양을 차지하고 있다. 컴퓨터 관련 시장에서 컴퓨터 게임 시장은 아주 큰 비중을 차지하고 있다. 그 컴퓨터 게임들 중에서도 인터넷을 사용한 멀티유저 인터넷 게임 시장이 급속도로 커진 점은 Blizzard 사의 Starcraft라는 게임의 판매 양에서 확인할 수 있다.

컴퓨터 게임의 처음 시작은 사용자가 컴퓨터에 앉아서 컴퓨터가 주는 문제들을 받아 풀어나가는 형식의 게임이었다. 그러다가 인공지능을 요구하는 전략적 요소를 가지고 있는 게임들이 등장하였다. 하지만 컴퓨터의 인공지능이라는 것에는 한계가 있다. 컴퓨터는 언제나 같은 패턴, 혹은 그 패턴에서 약간 변형된 패턴들을 제공하기 때문에, 사용자들은 금방 식상함에 빠지게 된다.

그러다가 네트워크를 통한 사람들 간의 게임이라는 것의 급속도로 발전을 시작한 현재의 인터넷이 만나면서 게임은 인터넷을 타고 전세계 사람들이 만나서

즐기는 도구가 되었다.

하지만 멀티유저 인터넷 게임에는 다음과 같은 문제점이 있다.

첫 번째 문제인 물리적인 라인 속도의 제한은 현재 기술의 발전으로 많이 개선이 되었다.

두 번째는 서버의 부하가 크다는 것이다. 게임 서버에 많은 사용자가 접속함으로써 서버의 성능이 떨어지는 것은 당연한 일이다.

세 번째는 사용자들에게 게임 클라이언트 설치를 위한 저장 공간을 필요로 한다는 것이다. 멀티미디어 인터넷 게임의 경우는 더욱더 그러하다. 어떻게 생각해보면 큰 문제가 되지 않을 수도 있다. 하지만 다음 네 번째 문자와 함께 살펴보기 바란다.

네 번째는 서로 다른 버전의 클라이언트들 간에는 게임이 시작되지 않는다. 그래서 버전업이 있으면, 항상 업데이트를 받아야만 한다.

위와 같은 네 가지의 문제점이 현재 멀티유저 인터넷 게임이 가지는 문제점이라 할 수 있다. 본 논문에서는 멀티유저 인터넷 카드게임을 구현한다. 위와 같은 문제점을 완벽히 해결하지는 못하겠지만, 어느 정도 보완된

서버를 구현한다.

2 절에서는 멀티유저 인터넷 게임의 개요와 클라이언트/서버의 연결 방식을 기술한다. 3 절에서는 멀티유저 인터넷 카드게임만의 특성과 카드게임으로서만 가질 수 있는 클라이언트/서버 구조에 대하여 기술한다. 4 절에서는 클라이언트/서버간에 서로 주고 받는 메시지에 대하여 기술하고, 5 절에서 결론과 향후 방향을 제시한다.

2. 멀티유저 인터넷 게임의 개요

멀티유저 인터넷 게임이란 인터넷을 통해서 여러 사람이 같이 즐길 수 있는 게임을 말한다. 그 대표적인 예로 머드(Multi User Dungeon)를 들 수 있다. 머드는 채팅이 발전한 형태로, 채팅의 멀티유저 개념이 그대로 들어가 있다. 예전에는 텍스트 기반의 머드 게임이 전부였지만, 현재는 멀티미디어적인 요소가 포함된 그래픽 머드가 유행을 하고 있다. 대표적으로 Origin사의 Ultima Online을 들 수 있다.

위와 같은 머드가 사회형 게임이라면, 실시간 전략 시뮬레이션이나 스포츠 게임 같은 대전형 게임도 멀티유저 게임으로 개발되어 많은 인기를 차지하고 있다. 대표적으로 Blizzard 사의 StarCraft, EA 사의 FIFA 시리즈를 들 수 있다.

위의 사회형 게임과 대전형 게임에서의 서버와 클라이언트간의 메시지 전달경로는 상당히 달라진다. 우선 사회형 게임을 살펴보자.

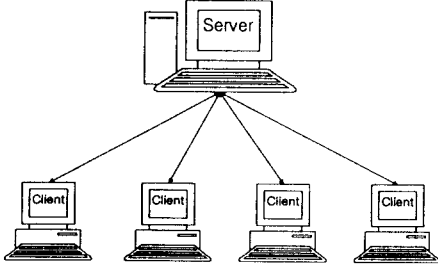


그림 1: MUD 게임과 같은 Client/Server 의 구조

그림 1에서 보듯이 모든 메시지들은 서버에 집중되고, 다시 서버에서 클라이언트에게로 분산되는 형태이다. 물론 서버에 큰 부하가 발생하게 된다. 대전형 게임을 살펴보자.

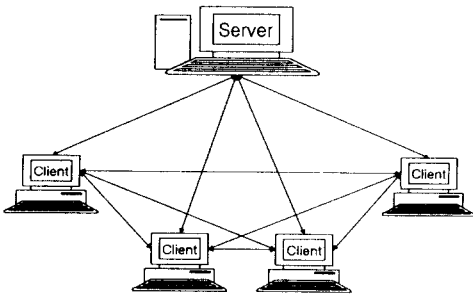


그림 2: 전략, 스포츠 게임의 Client/Server 구조

그림 2는 필요할 때만 서버와 메시지를 주고 받게 되고, 그 외의 게임 데이터는 클라이언트끼리만 주고 받게 된다. 다시 말하면 게임서버에 로그인 할 때는 클라이언트와 서버가 메시지를 주고 받고, 실질적으로 게임 시작되고 나면 서버와 클라이언트간의 메시지들은 필요가 없다. 서버는 게임이 끝이 났을 때, 누가 몇점으로 이겼는지만 서버에 기록하면 된다. 유동적인 메시지 연결이라고 할 수 있다. 모든 메시지를 서버가 총괄하지 않으므로 서버의 부하는 그만큼 줄어든다.

3. 멀티유저 인터넷 카드게임의 구조

본 논문에서는 멀티유저 인터넷 카드게임을 다루게 된다. 카드 게임은 게임의 특성상 위의 그림 2와 같은 대전형 게임의 구조를 따를 수 있다. 하지만 카드게임 가질 수 있는 특성인 제한된 인원, 작은 크기의 메시지 등을 생각해 볼 때 다른 방법을 모색할 수 있다.

카드게임만의 특성을 살펴보자. 우선 한번에 게임을 시작할 수 있는 사람의 수가 제한되어 있다. 인원의 제한을 생기는 장점은 뒤에 새롭게 제한하는 모델에서 언급한다. 그리고 메시지의 크기가 작다는 것이다. '누가 어떤 카드를 제출하였는가?', '누가 어떤 카드를 가져갔는가?' 하는 메시지가 주요 메시지라고 생각하면 된다. 마지막으로 카드의 제출이 항상 순차적이라는 것이다. 앞의 사람이 먼저 카드를 제출하지 않았는데 내가 카드를 먼저 낼 수 없다는 것을 말한다.

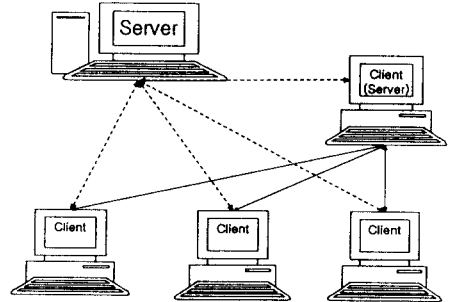


그림 3: Client(Client/Server)/Server 구조

위 그림 3을 살펴보면 처음에는 모든 클라이언트들이 서버와 메시지를 주고 받다가, 게임이 시작되면 게임을 생성한 클라이언트가 서버역할을 하게 된다. 그 게임에 새로 접속한 클라이언트들은 게임을 생성한 클라이언트를 서버로 여기고 그림 1과 같은 형태의 메시지 통신을 한다. 새로 접속하는 클라이언트의 수가 많아지면 게임을 생성한 클라이언트에 부하가 많이 걸리겠지만 카드게임은 제한된 인원의 사용자들만 게임을 할 수 있으므로 큰 부하는 걸리지 않는다. 단점은 클라이언트 프로그램이 서버 역할까지 맡아서 할 수 있어야 하므로 프로그램의 크기가 커질 수 있다. 이것은 게임 생성용 클라이언트와 게임 접속용 클라이언트를 다르게 구현함으로써 약간의 해결을 볼 수 있다.

3.1 서버의 내부

하나의 서버는 여러 개의 채널로 나눌 수 있다. 채널은 특정한 주제별로 사용자들이 모이는 집합을 말한다.

채널로 나눔으로써 각각의 채널을 각각 다른 서버로 분산시킬 수 있다. 아래 그림 4를 보면 DB 접속부, Message 처리부, 방 관리부, Chatting 처리부를 합쳐서 하나의 채널이라고 생각할 수 있다.

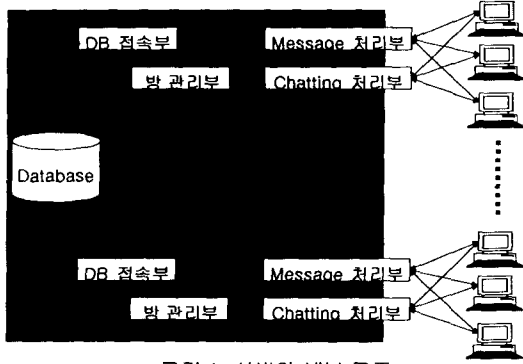


그림 4: 서버의 내부 구조

3.2 클라이언트의 내부

클라이언트는 점수 계산부가 따로 있어서 다른 사용자들의 카드들 배치만 가지고도 혼자서 사용자들의 점수를 계산할 수 있다. 그림 3에서 제안한 서버역할의 클라이언트를 만들기 위해서는 Client MSG 처리부를 조금 변형하면 가능할 것이라 생각된다.

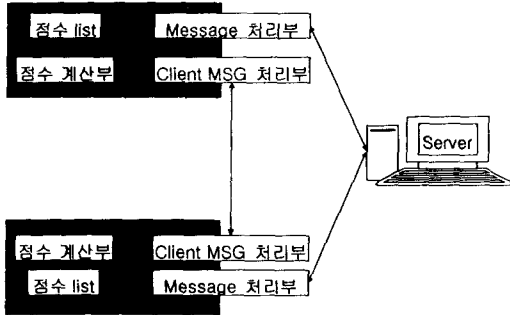


그림 5: Client의 내부 구조

4. 메시지

게임 도중에 클라이언트간에 흐르는 메시지는 '누가 어떤 카드를 내었고 어떤 카드를 가져갔는가?'하는 것이다. 이것을 메시지 리스트로 구성을 해보면 다음 표 1과 같다.

순번	User	낸 패	...
1	A	3	...
2	B	5	...
3	C	11	...
4	A	:	:

표 1: 메시지 리스트

메시지에는 차례로 몇 번째 메시지인지 순번이 매겨진다. 그리고 하나의 메시지에는 한명의 사용자가 행동한

모든 것이 포함되어 있다. 그리고 이 메시지는 하나씩 다니지 않고 몇 개의 메시지가 같이 묶어서 다닌다. 묶이는 개수는 현재 사용자 인원이 i 라면 묶여 다니는 메시지의 개수는 $i-1$ 개이다. 예를 들어 순번 n 인 사용자 C가 받는 메시지는 순번 $n-2$ 에서 $n-1$ 까지 이다. 이것은 A가 카드를 내고 나서, B가 낸 카드를 A가 받지 못했을 때, C가 카드를 내면서 A가 낸 카드 메시지까지 같이 보내주므로 모든 메시지를 좀 늦게라도 빠짐없이 처리할 수 있다.

5. 결론 및 향후 방향

본 논문에서는 멀티유저 인터넷 카드게임 구성을 위한 서버/클라이언트의 구성을 살펴보았다. 그리고 카드게임만이 가지고 있는 약간의 비동기성 메시지의 인정을 이용한 메시지들의 전달 방법도 제안하였다. 우선 테스트를 통해서 그림2의 클라이언트/서버 방식이 성능이 우수한지 그림 3의 클라이언트(클라이언트/서버)/서버 방식이 우수한지를 밝혀야 할 것이다.

그리고 게임 룰을 클라이언트에 포함시키지 않고 DB에 포함시켜서 클라이언트가 Server로부터 여러 가지 새로운 룰을 로드 할 수 있는 방법을 찾아야 할 것이다.

6. 참고문헌

- [1] 김기남 외 5명, "분산 네트워크 게임 서버의 설계 및 구현" pp57-59, 정보과학회, 1999,10
- [2] Jaeyong Park and Changhyeon Park, "Development of a Multiuser & Multimedia Game Engine Based on TCP/IP", IEEE, 1997
- [3] Ricky A. Bangun and H. W. p. Beadle, "A Network Architecture for Multiuser Network Games on Demand", ICICS, 1997
- [4] Yu-Chang Chen, Sing-Liang Chen, Ann-Yo Jeng and Wen-Shyong Hsieh, "Progressive Multimedia Mud System Designed AndPerformance Simulation ", IEEE, 1997