

C++ 환경에서의 온라인 보드게임 설계 및 개발

박용남*, 김민규*, 김성철*, 최재진*, 강형우*, 고석주*

*경북대학교 IT대학 컴퓨터학부

e-mail: pyn795@naver.com

Design and Development of Online Board Game in C++ Environment

Yong-Nam Park*, Min-Gyu Kim*, Sung-Chul Kim*, Jae-Jin Choi*,
Hyung-Woo Kang*, Seok-Joo Koh*

*School of Computer Science and Engineering, Kyungpook National University

요 약

최근 어렵고 복잡한 게임보다는 쉽고 친근한 게임들이 많은 주목을 받고 있다. 기존의 복잡하고 화려한 게임보다는 출퇴근 시간에 여유 시간을 활용할 수 있도록 쉽고, 간편한 게임을 사용자들이 원하기 때문이다. 본 논문에서는 기존의 오프라인 보드게임을 C++ 환경에서 온라인 보드게임으로 개발하여 일반 사용자들이 쉽고 친근하게 즐길 수 있도록 하였다. 이러한 구현 개발을 통하여 오프라인 보드게임에서 경험할 수 없는 요소를 추가한 온라인 보드게임의 장점 및 활용성을 보여줄 수 있다.

1. 서론

최근 스마트폰의 보급에 따라 게임 산업도 크게 변화하고 있다. 기존에는 고정된 환경 (PC)에서 화려한 그래픽을 이용한 복잡하고 어려운 게임들이 유행하였지만, 최근에는 출퇴근 등 이동 중에 간단하게 할 수 있는 모바일 게임이 유행하고 있다. 또한 과거에는 게임을 이용하는 연령이 10대 위주였다면, 현재는 10대부터 중장년층까지 다양한 연령대에서 게임을 즐기고 있다. 이러한 변화들로 최근에 출시되는 게임 역시 어렵고 복잡한 게임보다는 쉽고 친근한 게임에 초점을 맞추고 있고, 온라인 보드게임에 대한 관심도 높아지고 있다.

온라인 보드게임 (Online Board Game)이란, 보드게임과 온라인 게임을 합친 말로 일정한 규칙에 따라 네트워크 상으로 진행되는 게임이다. 보드게임이란 놀이판 및 간단한 물리적인 도구를 이용하여 진행되는 놀이를 말한다. 과거의 보드게임은 오프라인 상에서만 국한되어 있었다. 따라서 공간, 시간 제약으로 인하여 게임을 즐길 수 있는 한계가 있었다. 그러나 최근 인터넷 및 스마트폰의 보급에 따라 오프라인 보드게임의 제약을 보완하는 온라인 보드게임이 출시되고 있다.

따라서 본 논문에서는 오프라인에서의 제약을 보완하고 온라인에서만 느낄 수 있는 요소를 추가한 온라인 보드게임을 설계하고 내용을 소개할 것이다.

본 논문의 구성은 다음과 같다. 먼저 2절에서는 최근 온라인 보드게임들에 대해서 알아보고, 3절에서는 C++을 활용하여 온라인 보드게임을 설계 및 구현하여 내용을 소개할 것이다. 4절에서는 구현된 온라인 보드게임의 결과를

보여주고 5절에서 결론을 맺는다.

2. 관련연구

2.1 모두의 마블

1982년 씨앗사에서 출시한 보드게임인 블루마블을 모바일 게임으로 개발한 것이 모두의 마블이다. 2 ~ 4인이 2개의 주사위를 굴러 도착한 곳에 주권국의 땅을 사고 건물을 짓는 재산 증식형 게임으로, 게임회사인 넷마블에서 온라인으로 제작하였다. 기본 규칙은 블루마블과 같으며, 랜드마크, 독점, 제한 턴 등의 콘텐츠가 추가되어 기존의 블루마블과의 차이점을 갖는다[1].



(그림 1) 모두의 마블

2.2 하스스톤 온라인

하스스톤은 Blizzard에서 자체 룰을 제작한 카드를 이용한 보드 게임이다. 카드에는 유닛카드와 마법카드가 존재하며, 이 카드들을 사용하는 영웅이 존재하여 2명에서 1:1 대전을 하는 게임이다. 이 게임은 기존에 존재하는 워크래프트3 및 월드오브워크래프트에 등장하는 요소들을 카드로 만들어 서비스하고 오프라인 카드게임에서는 경험하기 힘든 호쾌한 액션과 빠른 진행으로 큰 인기를 끌고 있다 [2].



(그림 2) 하스스톤

3. 온라인 보드게임의 설계 및 구현

본 장에서는 기존에 존재하는 오프라인 보드게임인 마피아를 모티브로 하여 C++환경에서 DirectX를 이용하고 오프라인에서는 즐길 수 없는 온라인 게임의 특성을 추가한 새로운 온라인 보드게임을 설계 및 구현하였다[3].

본 게임에는 클라이언트와 서버, 2개의 프로그램으로 구성되었다. 우선 클라이언트 부분은 GameSystem, Scene, NetworkManager, GuiManager, 그리고 ResourceManager로 구성되었다. 서버는 Entry, ServerLoop, ProcProtocol, GameManager로 구성된다.

3.1 클라이언트 설계

3.1.1 GameSystem

GameSystem은 전체적으로 게임을 관리한다. 게임의 시작에 필요한 각종 준비과정을 담당하며, 게임 엔진의 전체적인 부분 (화면 렌더링, 통신, 업데이트, 유저 입력 등)을 최초로 수행하며, 해야할 역할들을 하위 컴포넌트들에게 적당히 할당해 주는 역할을 한다.

3.1.2 Scene

Scene은 현재 보여지는 부분을 담당하는 요소이다. Scene의 하위 계층으로는 처음 시작 시 IP와 닉네임을 입력하는 Scene1, 그 다음 게임 시작 전 대기실 역할을 하는 Scene2, 그리고 실제 게임이 진행되는 Scene3로 구성되어 있다. Scene3의 경우, 실제 게임플레이 화면, 채팅창 화면, 상태 표시 화면, 정보 및 로그를 보여주는 화면, 그리고

미니맵을 보여주는 화면, 이렇게 총 5개의 하위 View가 존재한다. 또한 게임의 각 요소가 되는 캐릭터, 맵, NPC, 아이템 등을 전체적으로 관리한다.

3.1.3 NetworkManager

GameSystem의 하위에 위치하며, 게임엔진 중 통신을 담당한다. 통신 프로토콜로는 TCP를 사용한다[4-6].

3.1.4 GuiManager

Scene3에서 사용되는 GUI를 관리한다. GUI로는 기본적으로 Direct X 기반의 DXUT에서 제공하는 GUI를 사용한다[7, 8].

3.1.5 클라이언트 프로그램 설계

그림은 클라이언트 프로그램의 간략한 수도코드를 나타낸다.

```

게임 구동;
게임 초기화;
while(게임이 종료되기 전까지)
{
    (1) 서버로부터 온 통신 내용을 읽음;
    (2) 현재 상태를 업데이트;
    (3) 렌더링;
    (4) 업데이트한 내용을 서버로 전송;
}
게임 종료;
    
```

(그림 3) 클라이언트 코드

클라이언트 프로그램은 다음과 같은 순서로 실행된다. 우선 서버로부터 통신이 오면 사전에 정해진 프로토콜 중 어떤 것에 해당하는지 NetWorkManager가 Scene별로 나누어 파악한다(1). 클라이언트의 실제 게임 진행 중 마우스, 키보드 입력, 각종 행동, 그리고 GUI 조작 등을 모두 포함하여 상태를 업데이트 한다(2). 그리고 전체 Scene을 렌더링하면서 해당 Scene에서 하위 컴포넌트 (캐릭터, 맵, GUI, 각 View)를 렌더링 한다. 마지막으로 (1)에서 받은 정보를 토대로 업데이트 된 자신의 정보를 다시 서버로 전송하게 된다(4).

3.2 서버 설계

3.2.1 Entry

서버의 시작 지점이며 서버의 구동 및 최소 설정을 담당한다. 프로토콜은 TCP 프로토콜을 사용하며, 여기서는 Microsoft의 IOCP 기법을 활용한다.

3.2.2 Serverloop

실제 IOCP를 적용하여 활용하는 부분이다. 통신(send,

receive)를 실제로 수행하는 코드가 위치하며, 게임이 구동되고 난 후에 서버 역할을 하게 된다.

3.2.3 GameManager

서버의 안쪽에 위치하며, 전달받은 데이터를 기반으로 실제 게임 상의 로직을 처리하는 부분이다.

3.2.4 서버 프로그램 설계

그림은 서버 프로그램의 간략한 수도코드를 나타낸다.

- (1) 서버 구동 및 시작;
- (2) IOCP 설정;
- (3) 메인 서버는 대기상태에 빠짐;
- (4) ServerIOCP에서 해당 프로토콜 처리;
- (4-1) ProcProtocol에서 프로토콜 별로 나눔;
- (5) 결과를 GameManager에게 전달, 게임 로직 처리;
- (6)클라이언트에게 전송;

(그림 4) 서버 코드

우선 서버 구동 및 시작을 하고 (1), IOCP 설정을 하게 된다 (2). 메인 서버는 IOCP를 설정하고 대기상태에 들어가면, Worker Thread가 IOCP설정을 체크하게 된다 (3). 그 후 ServerIOCP에서는 클라이언트로부터 데이터가 온 경우 ProcProtocol이 프로토콜별로 나누어 해당 프로토콜 별로 처리를 하게 된다(4, 4-1). 위 결과를 GameManager에게 전달하면, GameManager는 그에 따라 게임 로직을 처리하고 (5), 다시 클라이언트에 전송하게 된다(6).

3.2 설계된 게임의 특징

3.2.1 시야

구현된 게임에서는 ‘시야’라는 개념을 활용하여 오프라인 게임과의 차별성을 두었다. 그림 5은 게임에서의 시야를 보여준다.



(그림 5) 초이점으로의 초대 - 시야

3.2.2 움직임

일정한 한 가지 방향으로 움직일 필요 없이 상하좌우 중 자신이 가고 싶은 방향으로 움직일 수 있도록 온라인 게임의 특징인 자유도를 주었다.

3.2.3 각 캐릭터별 특성

각 캐릭터는 다음과 같은 특성을 가지고 있다. 사기꾼-사기, 부자-부유, 여행자-매의눈, 경찰관-공무집행, 탐험가-탐험, 수해자-수해, 이단자-간파, 예언자-천리안, 런닝맨-질주 그리고 부동산업자-건물관리. 10가지의 캐릭터마다 고유한 스킬은 제공하고, 게임 시작 시 캐릭터를 랜덤으로 배정하여 게임의 흥미도를 높인다.

3.2.4 다양한 이벤트

매 게임마다 맵에 랜덤하게 배치되는 NPC를 두었다. 또한 중간 발표 및 모든 맵 탐색 등 기존 오프라인에서 즐길 수 없는 요소를 추가하였고, 매 게임마다 다른 게임을 즐길 수 있도록 하였다.

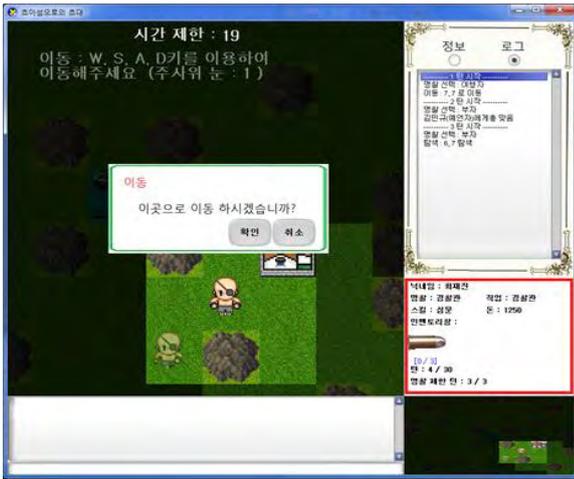
4. 구현 결과

구현된 온라인 보드게임은 4명의 플레이어가 각 직업을 랜덤으로 부여받아 서로의 직업을 맞추는 게임이다. 각 턴마다 행동을 수행하는 턴제 게임으로 구현되었다. 그림 6는 구현된 게임의 화면을 보여준다. 메뉴를 선택하여 이동, 탐색, 아이템 및 종료 기능을 수행할 수 있다.



(그림 6) 구현된 게임의 화면

그림 7는 이동기능을 수행하였을 경우 게임 화면을 보여준다. 그림에서 볼 수 있듯, 자신이 이동하고 싶은 위치로 이동을 할 수 있다. 이동은 한 턴에 한 번만 수행할 수 있으며 주사위를 굴려서 나온 수만큼 이동할 수 있다.



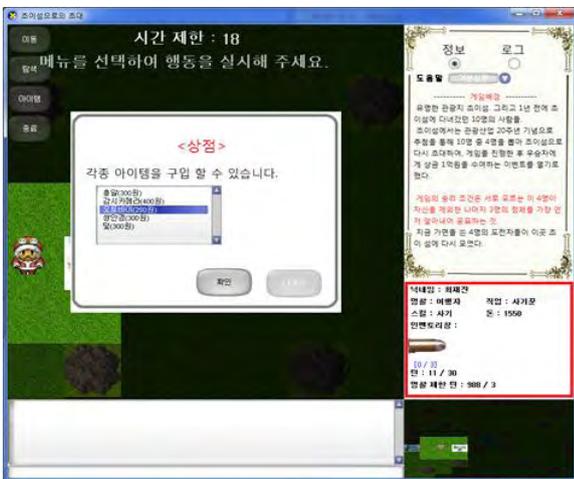
(그림 7) 구현된 게임에서의 이동 화면

그림 8은 아이템 기능을 수행하였을 때 화면을 보여준다. 자신이 소유한 아이템을 사용하면 다음 턴에 적용되고, 사용한 아이템은 그 자리에 버려지게 된다.



(그림 8) 구현된 게임에서의 아이템 사용

그림 9는 게임 상에 존재하는 NPC를 보여준다. 다양한 NPC를 이용하여 게임을 좀 더 즐겁게 이용할 수 있다.



(그림 9) 구현된 게임에서의 NPC 활용

5. 결론

본 논문에서 우리는 쉽고 친근한 오프라인 보드게임을 온라인 게임으로 개발 및 구현하였다. 기존 오프라인 보드게임에서 환경의 제약으로 인하여 경험할 수 없는 요소들을 온라인 보드게임에서 구현하여, 기존에 있던 보드게임의 친숙함과 새로 추가된 게임의 요소들로 인하여 새로운 즐거움까지 줄 수 있다.

기존에 존재하는 물을 단순히 이용한 오목 같은 게임들은 이미 많이 존재하고 있다. 앞으로는 기존 오프라인 보드게임을 단순히 복제하는 것이 아니라 차별화 된 흥미요소가 추가된 새로운 보드게임이 더욱 선호될 것이다. 따라서 기존에 존재하는 오프라인 보드게임에 온라인 게임의 특성을 추가한 게임이 많이 개발 될 것으로 예측되고, 이러한 게임이 게임 시장에 큰 영향을 줄 것이라 생각된다.

6. 감사의 글

본 연구는 경북대학교 컴퓨터학부의 서울어코드활성화사업의 지원으로 수행되었음.

참고문헌

- [1] Roger S. Pressman "Software Engineering A Practitiners' Approach" 3rd Ed. McGraw Hill
- [2] Blizzard Hearth Stone Home Page, <http://kr.battle.net/hearthstone/ko/> (August 2, 2014)
- [3] Scott Meyers, "Effective C++", Pearson Education Korea, 2006.5
- [4] 한동훈, "온라인 게임 서버 프로그래밍". 정보문화사, 2007.6
- [5] 신동훈, "클릭하세요 온라인 게임 네트워크 프로그래밍", 대림, 2003.5
- [6] 최연만, "온라인 네트워크 게임 프로그래밍", 가남사, 2003.11
- [7] Frank D. Luna, "DIRECTX 9를 이용한 3D Game 프로그래밍 입문", 정보문화사, 2004.2
- [8] 김용준, "3D 게임 프로그래밍", 한빛미디어, 2010.4