

논타겟팅 AOS 온라인 게임 설계 및 구현

이형구, 전익재
한국산업기술대학교 게임공학과
hgl@kpu.ac.kr, jejuuu@naver.com

Design and Implementation of the NonTargeting AOS Online Game

HyoungGu LEE, Ik Jae JEON
Dept. of Game & Multimedia, Korea Polytechnic University

요 약

본 연구에서는 2014년 게임시장에서 최고의 점유율을 보이고 있는 AOS 장르 온라인 게임의 한계점과 이를 해결하는 개선된 설계 사항과 구현 내용을 소개한다. 제안된 게임은 기존 AOS 장르의 형식을 바탕으로, TPS 시점의 논타겟팅 형식을 추가로 접목시켜 액션성을 강조하였고 그래픽, 애니메이션, 이펙트, 사운드 등을 적절히 조화하여 게임성을 향상 시켰다. 게임 서버는 IOCP 모델의 멀티 스레드로 제작되어 많은 클라이언트들을 수용할 수 있도록 하였다. 이에 관한 클라이언트와 서버의 기술적 설계 내용을 서술한다.

ABSTRACT

In this paper, the weak point of AOS genre online game which occupies the majority of Korea game market of 2014 year is analyzed and to solve the problem, the enhanced design and implementation content is introduced. The proposed game is based on the custom AOS genre format but has the TPS view and nontargeting format to enhance the action feel. To enhance the game quality, graphics, animation, effect and sound resources are organized. Game server is implemented by multithread of IOCP model to support many clients. The technical issue and design method of client and server are described.

Keywords : AOS - Aeon of Strife(대전액션과 공성전이 결합된 게임 장르), TPS-Third Person Shooter(3인칭 슈팅 게임), multithread (멀티스레드), synchronization (동기화)

Received: Sept. 15, 2014 Accepted: Oct. 15, 2014
Corresponding Author: HyoungGu LEE(Korea Polytechnic Univ.)
E-mail: hgl@kpu.ac.kr

ISSN: 1598-4540 / eISSN: 2287-8211

© The Korea Game Society. All rights reserved. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. 서 론

전 세계적으로 인기를 끌며, 2012년 국내에 서비스를 시작하여 국내 온라인 게임 순위 1위를 유지하는 LOL (League of Legend) 게임[1]과 국내에서 제작된 사이퍼즈 온라인 게임[2]과 같이 AOS(Aeon of Strife: 대전액션과 공성전이 결합된 게임)장르는 MMORPG가 주를 이루었던 국내 온라인 게임 시장에서 성공 가능성을 보여 주며, 스타크래프트 이후 주춤하던 E-Sports 문화를 새로이 개척해 나가고 있다.

AOS 장르는 쉬운 접근성과 빠른 진행으로 한 게임에 30분 이내의 시간을 소요 하여, 틈틈이 여유 시간에 집중해서 즐길 수 있다는 장점과 더불어 기존 MMORPG와 유사한 전투 방식에 팀원간의 협력을 통한 전략적 플레이로 게이머들에게 흥미를 유발할 충분한 요소를 가지고 있다고 볼 수 있다.

본 게임은 윈도우 PC 플랫폼의 서버-클라이언트 구조를 가진 롬 기반의 게임으로서, 기존 AOS 장르에서 볼 수 없었던 보물 상자, 용병, 대출과 같은 새로운 콘텐츠와 TPS(Third Person Shooter)시점의 논타겟팅 액션을 통해서 게임성을 향상 시켰다.

2. 게임 구성

본 게임인 러시 앤 캐슬(RnC)의 게임 기획과 게임의 진행, 클라이언트와 서버의 기술적 구현 방법을 서술한다.

2.1 게임 기획

제안되는 게임은 AOS장르 형식을 기본으로 한다. 맵은 하나의 월드 형식이 아닌 각각의 롬 기반으로 이루어져 플레이어가 선택한 롬에 접속을 하여 최대 6명의 플레이어가 매 게임마다 전사, 궁수, 마법사 중 하나를 선택하여 최대 3:3으로 팀을

구성하여 게임을 시작할 수 있다. 2팀으로 나누어진 플레이어들은 자신의 진영을 지키며, 상대방 진영을 먼저 점령 하는 것이 게임의 승리 조건이다. 게임이 시작되면 플레이어는 자신의 진영에서 시작하여, 상대팀 진영으로 가는 길목에서 만나는 AI로 움직이는 중립 몬스터, 방어 타워, 상대팀 몬스터와 상대팀 플레이어들과 전투를 하여 경험치와 돈을 얻어 더 강해지고, 상대팀 진영에 위치한 보스 몬스터를 먼저 쓰러트리면 게임에서 승리한다.

AOS 장르는 상대팀 플레이어들과의 적극적인 전투가 이루어져야 동적인 게임 진행을 해 나갈 수 있다. 하지만 게임이 진행될수록 상대 플레이어와의 격차가 점점 벌어지게 되어 일방적으로 당하는 입장이 된다면, 그 게임은 흥미를 잃고 지루해지기 마련이다[3].

이를 개선하기 위해 본 게임은 TPS시점의 논타겟팅 전투 형식과 대출 콘텐츠, 보물 상자 등을 넣어서 좀 더 동적인 게임의 진행을 할 수 있도록 하였다. 플레이어의 숙련된 컨트롤을 통해 수치에 불과한 능력치를 뛰어 넘는, 결과를 알 수 없는 전투를 할 수 있으며, 전략적인 대출을 통해 아이템을 구입하여 상대팀과의 능력치 격차를 줄여 나갈 수도 있으며, 맵 중간 중간에 생성되는 보물 상자를 먼저 선취하여 보상 효과를 가질 수도 있도록 하였다.

타겟팅 방식은 공격 하고자하는 대상을 선택 후 공격이나 스킬을 사용하면 그 대상에게 공격이 가해지는 시스템이다. 공격대상만 선택해놓으면 확실한 타격이 가능하다는 점과 자동적으로 공격할 수 있어 비교적 쉬운 조작이 가능하다. 하지만 조작 패턴이 너무 단조롭고 쉽기 때문에 금세 지루해지는 단점이 있다.

논타겟팅 방식은 공격 대상을 선택하는 것이 아닌 캐릭터가 바라보고 있는 방향으로 공격을 하여 그 범위 내에 있는 몬스터들을 공격하는 방식이다. 적의 상태를 면밀히 파악해야 하고 캐릭터의 움직임과 위치가 중요하기 때문에 현실적이고 다이내믹한 전투가 가능하다. 빠른 판단력과 행동이 중

요하기 때문에 컨트롤이 중요시된다.

논타겟팅 게임을 설계함에 있어서 가장 고려했던 사항은 공격의 타격 판정여부를 어떻게 결정할지를 정하는 것이었다. 단순히 공격을 가했을 때 해당 범위 안에 있으면 바로 데미지를 입히는 경우, 공격 방향으로 나아가는 공격일 경우에는 먼저 맞는 개체와 나중에 맞는 개체 간의 시간차 판정 등을 고려해야되는 경우, 유지되는 시간이 긴 공격의 경우 사용 시점에는 범위 안에 없더라도 나중에 이동을 해서 들어온 경우 등의 복잡한 상황이 있기 때문에 고려해야 할 사항이 많았다.

다른 고려사항으로는 타겟팅게임의 경우 타겟이 되는 대상에 한해서만 타격처리를 하면 되지만, 논타겟팅의 게임은 어떤 대상을 공격했는지를 판단하기위해 맵 전체에 있는 유닛들에 대해 타격 판정을 거치다보면 처리 속도가 늦어질 수도 있는 문제점이 생긴다. 그래서 캐릭터의 시야 안에 있는 유닛들만 처리 할 수 있도록 서버에서 시야 처리를 담당하도록 하였다.

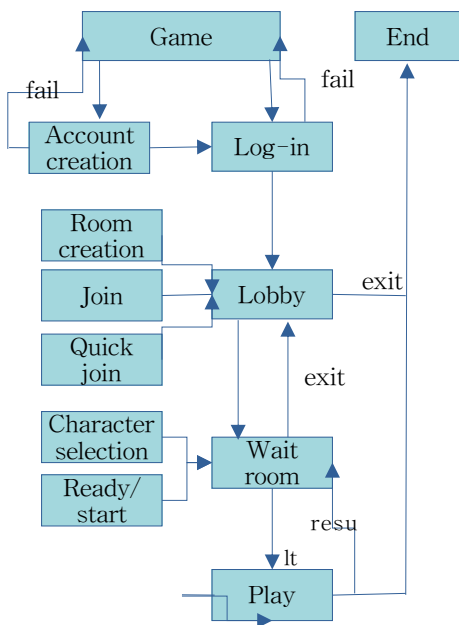
2.2 게임의 전체 흐름

[Fig. 1]은 게임의 전체 흐름을 보여준다. 로그인 성공시 로비에서 방을 선택하여 플레이를 시작할 수 있으며 로비, 대기방, 플레이 진행 형태에 대한 설명은 다음절에서 서술한다.

2.3 게임 진행



[Fig. 2] Log-in window



[Fig. 1] The flow of game

게임을 시작한 후 생성해 두었던 계정이 존재한다면 [Fig. 2]에서와 같이 해당 계정 정보를 입력하여 서버에 게임 접속을 요청한다.

생성한 계정이 없다면 아이디와 패스워드를 입력하여 서버에 계정 생성을 요구 한다. 이메일은 추후에 패스워드 찾기용으로 입력 받는다.

[Fig. 3]은 로비 화면을 보여주는데, 좌측 상단에는 생성된 방 정보가 나타나 있고, 우측 상단에는 로비에 존재하는 플레이어의 리스트가 나와 있다. 좌측 하단에는 채팅창이 존재하고, 우측 하단에는 자신의 정보가 보여진다. 방 생성을 통해 방을 생성하고, 생성된 방을 더블클릭 하거나, 바로 입장 버튼을 통해 대기방에 입장 할 수 있다.



[Fig. 3] Lobby window



[Fig. 4] Waiting room window

[Fig. 4]는 대기방 화면을 보여주며, 방을 생성한 유저는 방장이 되고, 팀원 간 짝이 맞고, 모든 유저가 준비 상태가 된다면 게임을 시작할 수 있다. 상대팀 슬롯이 비어있다면 팀 변경을 통해 상대팀으로 넘어 갈수 있고, 방장이 나갈 시에는 다음 슬롯에 존재하는 유저가 방장이 된다. 다른 플레이어의 슬롯을 클릭하면 해당 플레이어의 정보가 우측 하단에 표시된다.

[Fig. 5]는 사용자 인터페이스 화면을 보여주며, 좌측 하단에는 캐릭터 타입, 머니, TP(Team Point) 등의 정보를 보여주고 중앙 하단의 바는 자신 캐릭터의 체력, 마법력, 경험치를 나타내고, 그 밑의 슬롯은 물약 아이템과 스킬을 나타내는데 마

우스를 가져다 보면 해당 슬롯의 정보를 볼 수 있다. 우측 하단은 미니맵을 나타내어 자신 팀의 팀 몬스터와 팀원의 위치를 알 수 있다. 자신 팀의 진영 내에 위치한 물약 NPC, 버프 NPC, 대출 NPC를 통해서 머니나 TP를 효과적으로 사용할 수 있다.



[Fig. 5] User Interface (UI) window



[Fig. 6] Game Play Scene 1

[Fig. 6]은 게임 플레이 영상 화면을 보여주는 것으로 상대 팀원, 상대 팀몬스터, 상대 타워, 중립 몬스터를 사냥하여, 경험치와 돈을 얻어 레벨업을 하고 아이템을 구입하여 자신을 강화시켜 [Fig. 7]에서와 같이 상대팀 진영에 위치한 보스 몬스터를 쓰러트리면 게임에서 승리하게 된다.



[Fig. 7] Game Play Scene 2

2.4 클라이언트

2.4.1 클라이언트 구조

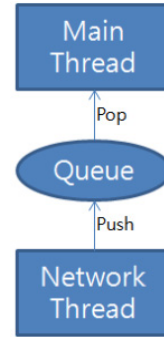
[Fig. 8]에 표현된 것처럼 클라이언트는 메인 스레드와 네트워크 스레드로 구성된다. 메인 스레드는 로그인 상태, 로비 상태, 방 상태, 게임 상태를 관리 할 수 있도록 유한 상태 기계(FSM) 구조로 이루어져있다. 네트워크 스레드는 IOCP(Input Output Completion Port) 모델을 사용했다. 서버로부터 받은 메시지를 네트워크 스레드에서 Queue에 Push하고 메인 스레드에서 매 루프마다 Queue에 메시지가 있는지 없는지 체크 하여 메시지가 있으면 처리하고 해당 메시지를 Queue에서 Pop 한다.

2.4.2 클라이언트 구현

캐릭터는 키보드와 마우스의 입력을 통해서 조작이 가능하다. 유저의 입력으로 이벤트가 발생하거나 캐릭터의 상태가 변경되면 메시지를 서버로 보낸다. 캐릭터의 위치와 방향은 수시로 서버에게 전송하여 서버와 클라이언트를 동기화시킨다.

카메라는 TPS (Third Person Shooting)시점으로 구현하여 캐릭터를 중심으로 회전 및 줌 인/아웃이 가능하도록 했다. 설정에 따라서 1인칭 시점, 쿼터뷰 시점(머리 위에서 일정 위치 앞을 바라보도록

함)으로 변경 할 수 있도록 개발 했다.



[Fig. 8] Client thread structure

카메라로부터 캐릭터 방향으로 Ray를 발사하여 다른 오브젝트가 있으면 캐릭터가 가려져 보이지 않으므로 해당 오브젝트의 알파 값을 50%로 수정해서 반투명하게 보이도록 했다. 반투명하게 된 오브젝트의 포인터를 가지고 있다가 Ray에 걸리지 않게 되면 알파 값을 복원해 반투명 효과를 해제하도록 했다.

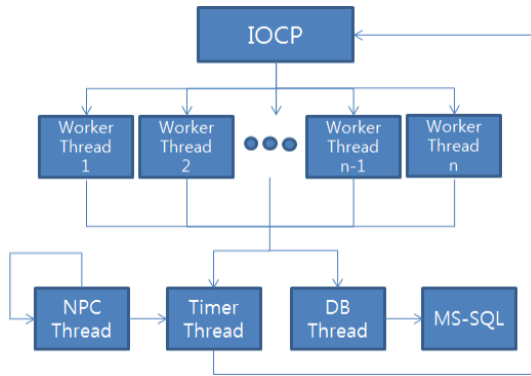
캐릭터와 오브젝트간의 충돌은 캐릭터의 이전위치에서 움직인 후의 위치로 Ray를 발사하여 직선상에 물체가 있으면 충돌이라 판정하여 움직일 수 없도록 했다. 지형과는 단순 높이만을 계산해서 갈 수 있는 곳인지 없는 곳인지 판단한다.

캐릭터가 스킬을 사용하면 스킬을 재사용하기까지 시간이 필요한데 UI (User Interface) 하단의 해당 스킬 이미지 위에 재사용까지 남은 시간을 숫자로 표시해주고, [Fig. 7]에서처럼 이미지의 가운데를 중심으로 스킬의 총 쿨 타임에서 스킬을 사용하고 지난시간을 계산하여 시계 방향으로 밝아지도록 하여 육안으로도 알아보기 쉽도록 표현하였다.

SkyX 라이브러리[4]를 이용해 하늘의 밤낮효과를 구현했고, 그에 따른 태양의 방향벡터를 계산하여 빛의 방향을 변경해주어 더욱 실감나는 날씨효과를 구현하였다.

UI는 CEGUI라이브러리[5]를 사용하였다. CEGUI라이브러리의 Editbox는 한글이 입력되지 않으므로 한/영 키를 누르면 Editbox에서 한글 입

력이 가능하도록 바꾸어 한글 채팅이 가능하도록 했다.



[Fig. 9] Server thread structure

최종 보스를 잡고 게임을 클리어하면 보스의 죽는 애니메이션부터 성 내부의 상자가 열리는 애니메이션까지 보여 줄 수 있도록 카메라 무빙 애니메이션을 추가하여 게임 종료 후 미묘함을 없앴다.

2.5 서버

2.5.1 서버 구조

[Fig. 9]는 본 게임의 서버 스레드 구조를 보여주는 것으로, RnC서버는 마이크로소프트 Windows 환경에서 동작하는 IOCP 모델의 멀티스레드 서버이다[6]. IOCP는 비동기적으로 발생한 IO를 WorkerThread들이 감지하고, 감지한 스레드가 작업을 처리하는 동안 다른 작업도 처리하기 위해서 ThreadPool 기법을 통해서 컨텍스트 스위칭을 최소화하고 쉬고 있는 WorkerThread를 깨워내는 방식으로 윈도우즈 환경에서 서버가 최상의 성능을 낼 수 있는 방법으로, AcceptEx와 함께 사용할 때 1 프로세서에서 65,536개의 연결이 가능하다[7].

현재 구현된 RnC서버는 하나의 프로세서 내에서 실행되는 형태이다. 서버 가동 시 생성 되는 스레드로는 다수의 WorkerThread와 DBThread, TimerThread, NPCThread가 있다.

WorkerThread는 IOCP를 효율적으로 사용하기 위해서 코어 개수의 2배수로 생성된다.

비동기적으로 발생한 I/O에서 수신 받은 데이터를 조립한 패킷과 PostQueueCompletionStatus 함수를 통해 넘겨받은 데이터는 여러개의 WorkerThread 마다 병렬적으로 처리를 하기 때문에 동기화를 통해 공유 메모리에 안전하게 접근하도록 하였다.

DB는 MS-SQL을 사용하며, Windows내에 ODBC 관리자를 통해 SQL-Server에 액세스 하여 DB에 접근하도록 하였다. DB Query를 처리하기 위해서는 WorkerThread, TimerThread에서 발생한 Query 요구를 공유되는 메모리의 CircularQueue에 Push하고, DBThread에서는 순차적으로 Pop하여 DB서버에 Query를 전송해 DB access로 인한 서버의 응답 시간 대기 지연을 없애도록 하였다.

2.5.2 서버 구현

서버가 실행되면 SQL접속과 IOCP 연결이 수행되고, 게임 초기 데이터가 로드 된다. AcceptEx 함수와 소켓 재사용을 통해서 비동기적으로 다수의 유저가 한꺼번에 빠른 접속이 가능하게 하였다.

게임 내의 상점에서 판매되는 아이템 정보와 각 캐릭터의 스킬 정보는 XML파일로 저장되어 싱글톤으로 이루어진 아이템 매니저와 스킬 매니저를 통해서 데이터가 메모리상에 로드 되어, 해당 데이터가 요구될 때마다, 해당 매니저의 유일한 인스턴스를 통해서 데이터를 읽어 들인다.

사용자가 로그인을 시도하면 아이디와 패스워드를 DB에 Query를 전송해 결과를 리턴 받아 아이디 유효성과 기존에 이미 접속 된 유저인지를 판단한다. 접속이 가능한 유저라면 유저의 기본 정보를 리턴 받아 메모리상에 로드하고, 유저에게 전송한다.

로비에 접속한 유저는 현재 로비에 보여 지는 룸 정보 리스트와 접속한 유저 리스트를 직렬화

하여 패킷으로 전송한다. 일반적인 고정된 구조체 형식의 패킷으로는 가변적인 크기의 데이터를 담아서 전송하기에는 부적합 하므로, 객체/데이터를 직렬화 하여 패킷을 구성하고 전송해서 클라이언트 단에서 역직렬화를 통해 패킷 처리를 하도록 하였다. 직렬화는 Boost 라이브러리와 STL을 활용하여 데이터에 포함된 여러 자료형의 각각의 크기를 체크해서 스트림으로 만들어 낸다.

게임이 시작되면 플레이어는 시야를 관리하여, 자신의 시야 내에 들어 온 유저에 대해서만 각종 공격 연산과 로직 처리, 패킷 전송을 하기 때문에 시야 동기화는 자주 실행되는 만큼 그에 따른 부하가 크기에 RWLock[8]을 사용하였다.

RWLock은 ReadWriteLock의 줄임말로 쓰기용과 읽기용으로 구분 되어 스레드 동기화시, 읽기만 하는 스레드 간에는 공유 메모리에 동시에 접근을 하여도 문제가 되지 않기에 읽기 스레드는 공유적 접근을 하고, 쓰기 스레드는 배타적인 접근 권한을 가진다. 기존의 CriticalSection과 같은 유저모드스레드 동기화에 비해 성능이 더 향상된 방법이다[8].

유저의 hp, exp, 머니 등과 같이 중요한 유저 정보는 여러 스레드에서 동시에 공유 자원에 접근할 수 있고 이들 중 하나가 write할 때 발생하는 data race 현상의 문제를 해결하기 위해서 atomic한 연산을 하도록 Interlocked API를 사용하였다.

상점 내의 아이템 구입은 아이템 고유 인덱스를 이용해 싱글톤인 아이템 매니저 인스턴스에서 STL map에 저장된 아이템 객체를 검색하여 해당 아이템의 구입 조건을 비교 검사하게 된다. 아이템의 사용은 아이템 객체를 ItemHandler 모듈에 넘겨서 실행되고, 장비 아이템이 아닌 일회성으로 사용되는 아이템의 경우는 스킬로 연결된다.

스킬의 사용은 스킬 고유 인덱스를 이용해 싱글톤인 스킬 매니저 인스턴스에서 STL map에 저장된 스킬 정보를 find하고, 해당 스킬객체를 SkillHandler에 넘겨 처리 하게 된다. 논타겟팅 처리를 위해 공격 스킬은 6가지 타입으로 나뉘어 스킬 객체에 저장된 공격 범위와 사거리, 마나 소비,

쿨타임, 데미지 배수, 속도 등의 정보를 이용하여 계산 된다.

(1) Shoot 타입은 원거리 공격으로 유일하게 hit 되거나 사거리를 넘어서게 되면 없어지는 스킬이다. (2) ShootPass 타입은 hit가 되더라도, 관통되어 나아가 여러 대상을 공격 할 수 있다. (3) Quadrangle 타입은 자신을 기준으로 하여 전방으로 사각 영역에 위치한 대상을 공격 한다. (4) HoldDegree 타입은 자신을 기준으로 하여 전방으로 부채꼴 영역에 위치한 대상을 공격 한다. (5) HoldAround 타입은 자신을 중심으로 원 영역 이내에 위치한 대상을 공격한다. (6) PosAround 타입은 플레이어가 픽킹한 위치에서 원 영역 이내에 위치한 대상을 공격한다.

서버 내의 타이머에 따른 이벤트 처리를 위해 생성되고, 비동기 전송 시 사용되는 기존 Overlapped 구조체를 확장한 OverlappedEx 객체는 다수의 WorkerThread내에서 메모리 할당 및 해제가 매우 빈번하게 수행하게 되는데, 그에 따른 오버 헤드와 내, 외부 메모리 단편화 문제, 멀티스레드 환경에서의 속도 및 효율성이 떨어지기에, 윈도우 멀티 스레드 환경에 최적화된 메모리 할당 라이브러리인 nedmalloc[9]를 사용하여 메모리를 할당 및 해제하였다.

2.6 리소스

2.6.1 모델링

게임 캐릭터는 3ds Max 2013을 이용하여 제작하였다. 사용자 캐릭터는 제작기간과 게임성능을 감안해 폴리곤 5천개 전후로 제작하였고 게임내에서 다수 생성되는 몬스터의 경우 3천개 전후의 폴리곤을 사용하였다.

모든 캐릭터는 장구류를 포함 캐릭터 당 1장의 텍스처에 포함시켜 로딩속도를 줄였다.

캐릭터 애니메이션의 경우 바이패드에 본을 붙여서 사용했고 모든 정점은 4개 이하의 가중치를 가지도록하여 정점 데이터의 크기와 연산을 줄였

다. 텍스처는 PhotoShop CS5 버전으로 제작하였으며 사용자 캐릭터는 1,024*1,024 의 크기를 가지며, 몬스터 캐릭터는 512*512 의 크기를 가진다.

2.6.2 이펙트

이펙트 파일은 Particle Universe[10] 툴을 사용하여 스크립트로 만들어졌고, Ogre엔진과 연동되는 라이브러리를 추가하여 적용 시켰다. 샘플 스크립트 파일과 그 외 파티클 리소스는 툴을 제작한 회사에서 판매하는 데이터팩을 라이선스 비용을 지불 하고 구입하여, 일부 변형 시켜서 게임에 적용하였다.

2.6.3 사운드

사운드 파일은 이펙트 사운드 모음 CD와 토치라이트1 데이터 팩에서 wav 파일을 추출하고, fmod 라이브러리를 추가하여 게임에 적용하였다.

3. 결 론

소개된 RnC 게임은 기존 AOS 장르의 형식을 바탕으로 TPS 시점에 논타겟팅 전투 형식을 추가로 접목시켜 액션성을 강조했다. 기존 AOS장르 게임의 경우에는 카메라가 쿼터 뷰 시점으로 고정되어 있지만, 본 게임은 사용자가 TPS 시점과 쿼터 뷰 시점을 선택할 수 있어 사용자가 원하는 카메라 컨트롤이 가능하다. 이것은 기존 MMORPG의 TPS 시점 전투에 익숙했던 사용자들에게 AOS 장르로의 접근을 용이하게 해준다. 또한 보물 상자 콘텐츠로 복불복시스템을 도입하여 재미요소를 추가하였으며, 대출 기능을 이용하여 어려운 상황을 극복해가거나 다양한 전략으로 활용할 수 있도록 하여 한층 업그레이드된 재미를 느낄 수 있다. 한계점으로는 현재 서비스중인 AOS장르의 타 게임에서는 여러 가지의 캐릭터중 하나를 선택하여 게

임을 하는 것인데, 본 게임은 아직 캐릭터가 3종류로 선택의 폭이 좁아 보완해야 할 사항이다.

서버의 경우에 현재 하나의 프로세스 내에서 분할된 스레드로 운용되고 있어 기존 유사한 다른 게임들에 비해 접속자수의 증가로 인한 부하가 생길 수 있다. 향후 안정성과 신뢰성, 병목 현상, 부하 분산, 확장성, 보안, 일관성, 최고 지연, 관리비용 등을 고려한 분산 서버를 구성하여 접속자수가 증가할 때마다 필요한 서버가 추가로 연결될 수 있도록 구현하여 접속자수의 증가로 인한 부하를 분산시킴으로써 빠르고 안정적인 게임을 제공할 수 있도록 할 계획이다.

REFERENCES

- [1] John Gaudiosi, "Riot Games' League Of Legends Officially Becomes Most Played PC Game In The World", Forbes, July 11, 2012
- [2] Cyphers home, cyphers.nexon.com, neople
- [3] Dong-Seong Choi, "A Study on the Creative Game Design Framework", Journal of Korea Game Society, Vol.11, No.4, pp. 49~60, 2011
- [4] Xavier Verguin Gonzalez, "SkyX : Sky and weather simulation", Ver. 0.4.0, Hydrax & SkyX at Paradise Studios, 2014
- [5] Qingfeng Wu, Huailin Dong, Tianmao Xu, "Research on 3D Visualization Framework via CEGUI in OGRE Engine", Journal of Convergence Information Technology, Vol.7, No.19, pp.538~545, 2012
- [6] Zhong-Yong Che, Sang-Chul Kim, "A Game Placement Method Considering Load Balancing between Game Servers in a Game Server Cluster", Journal of Korea Game Society, Vol.12, No.3, pp. 49~58, 2012
- [7] Jong-Gwan Choi, Hye-Young Kim, Won-Sik Woo, "A Study of a Game User Oriented Load Balancing Scheme on MMORPG", Journal of Korea Game Society, Vol.12, No.3, pp. 69~76, 2012
- [8] Nicole Hamilton, "Single-writer Multi-Reader lock for Win98", comp.programming.threads, 19 October, 1999

- [9] Niall Douglas, “User Mode Memory Page Allocation”, Int. Symposium on Memory Management, June 4-5, 2011
- [10] Particle Universe, <http://www.fxpression.com/>



이 형 구 (LEE, Hyoung Gu)

1993 서울대학교 계산통계학과 이학사
1996 서울대학교 대학원 계산통계학과 이학석사
2001 서울대학교 대학원 컴퓨터공학부 공학박사
2001-2003 한국전자통신연구원 영상정보처리연구팀
 선임연구원
2003-현재 한국산업기술대학교 게임공학과

관심분야 : 인공지능, 영상처리



전 익 재 (Jeon, Ik Jae)

2008-2014 한국산업기술대학교 게임공학과

관심분야 : 기획, 게임프로그래밍
