

온라인 게임 개발 사례 : 헬브레스

(주)시멘텍 고광현

1. 서론

국내 게임 시장은 온라인 게임의 전성시대를 맞이하였다. 그 동안 게임이란 엔터테인먼트 산업 중에서도 제품의 수명이 가장 짧은 분야 중 하나였다. 아무리 잘 만든 게임이라고 해도 보통 제품 출시 후 3~4개월의 수명을 갖는 것이 보통이다. 물론 바람의 나라, 어둠의 전설, 스타크래프트, 디아블로 2 같은 예외도 있다[1,2,3]. 하드웨어와 소프트웨어의 엄청난 발전 속도를 눈에 띄게 알 수 있는 분야가 바로 게임이다. 그런데 온라인 게임은 이런 게임은 짧은 생명주기를 갖는다는 기존 관념을 많이 뒤바꿔 놓았다. 한 게임을 몇 년간에 걸쳐서 할 수 있다는 것은 오직 온라인 게임에서만 있을 수 있는 일이다. 게이머(gamer)에게나 개발자에게나 무척 매력 있는 분야이다.

본 고에서는 온라인 게임 기획, 개발과 운영, 업데이트에 있어서 중요하게 고려할 사항을 개발 사례를 통해 논의하고자 한다.

2. 온라인 게임 기획 과정

게임의 기획단계에서 가장 고려해야 했던 것은 게임의 형태에 따른 서버-클라이언트 구조였다[4,6]. 네트워크 게임의 서버 구조는 게임의 형식에 따라 많이 바뀌는데, 스타크래프트와 같은 종류의 게임들은 게임을 할 사용자들간의 만남의 공간으로 배틀넷을 운영하며, 일단 게임이 시작되면 서버를 통하지 않고 각 클라이언트끼리 통신 메시지를 주고 받으며 동기화를 이루는 구조이다. 반면 헬브레스와 같은 MMORPG 형태의 게임은 모든 사용자의 클라이언트가 서버에 접속되어 게임이 진행되는 서버 구조를 띄고 있다[4,6].

헬브레스는 MMORPG 중에서 빠른 액션감을 구

현할 수 있도록 서버를 맵(map) 분할 방식으로 구성하기로 하였다. 서버의 맵 단위 분할 방식은 한 개의 서버 프로그램이 자신에게 할당된 맵 상의 모든 오브젝트의 처리를 맡아서 하는 방식을 말한다. 즉 한 서버 프로그램은 해당 맵에서 플레이 하는 모든 사용자 캐릭터와 NPC, Monster, 기타 오브젝트들의 처리를 담당하나 다른 맵에서 일어나는 이벤트는 전혀 처리하지 않는다.

베틀넷 형태의 온라인 게임들은 실제 게임에 들어가면 그림 2와 같이 최대한 서버와의 통신을 중단하고 클라이언트끼리 통신하면서 게임이 진행되지만 MMORPG 류의 온라인 게임은 그림 1과 같이 게임 중에도 서버와 통신한다.

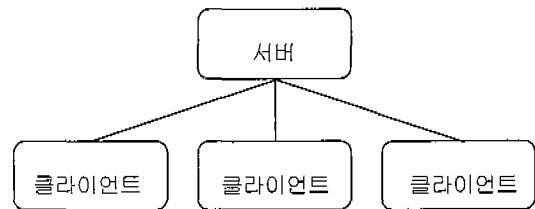


그림 1

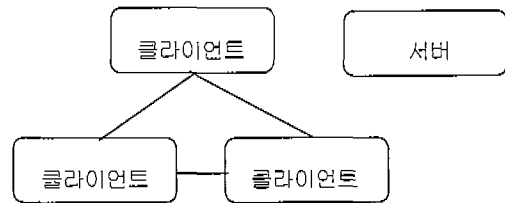
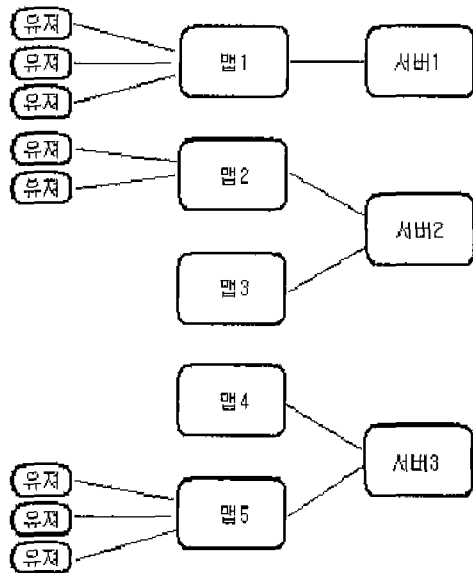


그림 2

이러한 서버들의 맵 단위로 할당되어 하나의 월드(world)를 구성하게 된다. 이런 맵 분할 방식의 서버

구조는 구현하기가 쉽고 에러 발생시 디버깅이 쉽다는 점, 그리고 해당 맵에 있는 오브젝트별로 처리하므로 속도가 빠를 것으로 예상되었기에 헬브레스의 서버 구조로 선택되었다.

게임 디자인 측면에서는 머그게임 사용자들간의 집단적인 전투를 고려하였다. 아마도 머그게임으로서는 최초가 아니었나 싶다. 기존의 머그 게임들이 사용자 개개인의 전투와 길드단위의 소규모 집단 전투를 구현했던 것에 비해 헬브레스는 디자인 단계에서부터 국가와 국가간의 대립구조로 모든 것이 설계되었다. 그래서 다른 국가 플레이어들끼리는 서로를 적으로 인식하여 자유로운 전투가 가능하도록 게임을 디자인했다.



맵 분할 서버 구조: 게임 서버는 자신에게 할당된 맵상에서 움직이는 오브젝트와 유저의 요청에 대한 처리를 전담한다. 다른 서버상의 일은 간섭하지 않는다.

그림 3 게임 디자인 과정

3. 온라인 게임 개발 과정

3.1 로그인 프로세스

게임의 기본적인 틀과 시나리오가 나온 후 실제 제작에 착수하였다. 처음 완성된 부분은 서버-클라이언트 접속 과정의 구현이다. 클라이언트가 로그인 과정을 거쳐 실제 게임에 접속할 때까지의 과정은 생각보다 여러 단계를 거치게 된다. 계정과 패스워드를

입력하여 서버로 전송하면 서버는 그 사용자를 인증 처리한 다음 해당 서버에 존재하는 캐릭터의 정보를 읽어서 재전송해 준다. 그리고 사용자가 접속할 캐릭터를 선택하여 요청을 하면 또 다시 인증 과정을 거친 다음 실제 게임서버로 연결시켜 준다. 기본적인 과정이지만 그만큼 더욱 더 중요한 부분이다. 헬브레스 클라이언트-서버 연결도 게임 클라이언트도 최초로 메인 로그인 서버로 접속하여 인증을 마친 후, 월드 로그인 서버로 접속한다.

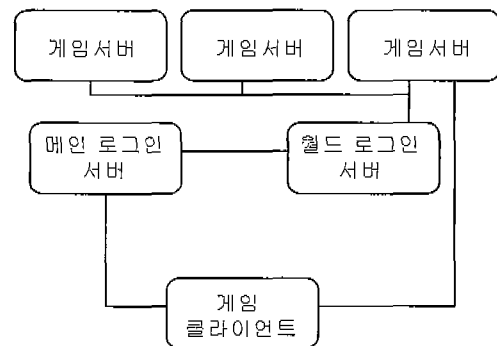


그림 4 온라인 게임 로그인 과정

3.2 온라인 게임의 기본 형태

다음으로 제작된 부분은 게임 속 캐릭터가 서버와의 통신 메시지를 주고 받으며 이동하는 단계였다. 맵이나 NPC, Monster 등의 오브젝트는 전혀 구현되지 않았지만, 일단 메인 캐릭터의 서버-클라이언트 구조 아래에서 이동부분이 구현되면 그 다음부터 다른 오브젝트들을 추가하는 것은 쉬운 일이었다. 여기서 한가지 기술적인 문제에 부딪혔는데, 그것은 캐릭터들의 역동적인 움직임을 어떻게 구현할 것인가에 대한 문제였다. MMORPG에서는 게임을 동시에 하는 플레이어가 수천명에 이르고 이들의 움직임들에 대한 제어할 모두 서버에서 처리해야 한다. 예를 들어 만약 두 캐릭터가 동시에 같은 위치로 움직였다고 하고, 그리고 한 위치에는 한 캐릭터만이 서 있을 수 있다고 한다면 둘 중 먼저 움직인 캐릭터의 행동을 인정하고 다른 캐릭터의 이동은 거부되어야 하며 이런 결정은 서버에서 이루어진다. 따라서 서버에서 내 캐릭터의 행동에 대한 승인/거부 메시지를 받기 전까지는 다른 행동을 해서는 안된다는 말이 된다. 1세대 온라인 게임들이 이러한 방식을 쓰고 있어 매우 느린 캐릭터의 움직임을 보여준다. 반면 헬브레스는 액션

성이 강조된 게임이므로 이런 방식으로는 게임을 제작할 수 없었다. 때문에 응답 시간의 지연을 최대한 보완할 수 있는 클라이언트-서버 네트워크 구조를 구현해야 했다. 헬브레스 클라이언트는 서버로부터 행동에 대한 승인/거부 응답이 없어도 일정한 수준에서 연속적인 동작이 가능하게끔 설계되었다. 그래서 캐릭터가 빠른 속도로 움직여도 허용 범위 내에서는 끊김 없는 매우 자연스러운 움직임을 보여준다. 다만 서버로부터의 응답이 한꺼번에 몰려서 수신되었는데 그전 행동들에 대한 거부 응답이 있다면 캐릭터는 그 전 위치로 되돌려지게 된다.

3.3 메모리 관리 기법

클라이언트의 많은 그래픽, 오디오 데이터를 관리하기 위한 메모리 관리 기법은 실시간 데이터 로딩 방식을 택했다. 헬브레스의 경우 다른 머그 게임들에 비해서 많은 그래픽 데이터를 필요로 하는데, 그 많은 데이터들을 모두 메모리에 로딩하기는 불가능했다. 그래서 화면상에 오브젝트가 등장한 경우 그 오브젝트를 표현하기 위해 지정된 스포라이트 데이터를 그때 그때 로딩하고, 또 이미 로딩되었지만 오랫동안 사용되지 않은 데이터는 메모리상에서 해제시키는 방법을 사용했다. 이런 방법은 많은 용량의 데이터를 효과적으로 관리할 수 있었지만, 메모리 단편화 현상을 가져와 게임이 진행될수록 속도가 느려지는 현상을 발생시키는 부작용이 있었고, 한 화면에 많은 오브젝트가 동시에 등장하는 경우 많은 데이터 로딩을 위해 게임의 진행이 끊기는 등의 부작용이 있었다. 이런 끊김 현상을 막기 위해서 헬브레스는 데이터 로딩을 위한 멀티 쓰레드를 구현하였다. 디스

크 액세스를 독립된 쓰레드에서 하므로 로딩을 위해 게임 진행이 멈추지는 않지만 대신에 보여야 할 오브젝트가 로딩이 끝날 때까지 보이지 않는 현상이 발생하였다. 이런 현상은 앞으로 좀 더 보완되어야 할 부분이다.

4. 온라인 게임 서비스 단계

4.1 서버 시스템 관리

게임 자체 시스템의 개발과 병행되어야 할 것은 서버 시스템 관리 시스템의 개발이다. 온라인 게임 서비스가 커지면 커질수록 많아지는 서버 장비를 관리하기 위한 자체 시스템의 필요성이 커진다. 서버의 재부팅이나 사용자 정보의 관리, 해킹 문제 처리를 위한 아이템 이동 등 서비스 중 발생하는 여러 가지 요구를 신속히 해결하기 위함이다. 서버 재부팅 시 마다 일일이 시스템 하나 하나를 수동으로 리부팅시키고 서버 프로그램을 다시 띄워야 한다면 불편한 일일 것이다.

4.2 게임 콘텐츠 추가시의 일관성 유지

국내의 온라인 게임은 국내 게임 개발 여건상 완전히 완성되지 않은 상태에서 유저들에게 선보인다. 사실 온라인 게임에 있어서 완성된 상태란 없다. 온라인 게임은 서비스 시작된 이후에 계속적으로 업데이트 되면서 진화해 가기 때문이다. 온라인 게임이 다른 패키지 게임들에 비해 엄청나게 긴 수명을 갖고 있는 것도 다 그런 이유 때문이 아닌가 싶다. 반면 이런 요소가 단점으로 작용하기도 하는데 그것은 게임의 일관성 문제이다. 일단 온라인 게임이 서비스에 들어가면 플레이어들은 보다 새로운 게임의 요소를 요구하게 되는데, 이것은 게임 개발자와 게이머들 간의 갈등을 유발한다. 개발자의 입장에서는 애초에 기획되었던 시스템의 구현을 하나하나 해 나가는 쪽으로 방향을 잡지만, 실제 게임을 하는 게이머들의 요구와는 틀려질 때가 많이 있다. 이것은 마치 마지막 회를 찍지 않은 채로 방영이 시작된 TV 드라마와 같은 경우이다. 드라마가 진행되면서 시청자들은 자기가 원하는 대로 주인공들의 삶이 결정되기를 원하고 결국에는 원래 시나리오상 죽었어야 할 주인공들이 버젓이 살아나 해피 엔딩으로 끝나버리기도 하는데, 과연 원작자의 심정은 어떠했을지 궁금할 때가 있다. 온라인 게임도 이와 마찬가지로, 게이머들의



헬브레스의 스크린 샷. 많은 그래픽, 오디오 데이터가 필요한 것만 선별, 실시간으로 로딩하여 출력한다.

그림 5 헬브레스 스크린 샷

요구와 개발자가 지향하는 게임의 방향을 어떻게 적절히 조화 시키느냐가 운영 및 업데이트 단계에 있어서의 핵심이다.

4.3 테스트 서버의 필요성

온라인 게임의 운영 중 가장 큰 문제점 중 하나는 새로 추가한 기능이 오류로 인해 오작동 하는 경우이다. 패키지 게임의 경우 게임 중 어떤 치명적인 오류가 있어도 그리 치명적이라고는 말할 수 없다(아닌가?). 그러나 실시간으로 플레이어의 데이터가 계속 저장되고 있는 온라인 게임에서의 오류는 대단히 치명적이다. 오류는 그것을 경험한 게이머 뿐만 아니라 그와 게임상에서 교류하고 있는 모든 게이머에게 영향을 미친다. 또한 그 오류가 캐릭터의 데이터를 크게 손상시키는 오류였다면 문제는 더욱 커진다. 다행이도 웹브라우저의 경우 이러한 치명적 오류는 발생한 적이 없지만 이런 문제는 온라인 게임이라면 어떤 게임에나 잠재되어 있는 문제일 수 있다. 따라서 이런 현상을 막기 위해서는 내부 테스트 및 공개 테스트 서버의 작동을 통한 예러 검증작업이 반드시 필요하다.

4.4 데이터 변형을 통한 해킹 방지

클라이언트와 서버간의 통신 메시지를 통하여 온라인 게임은 작동된다. 이 메시지는 사용자에게 의해 읽혀질 수 있고, 또 조작될 수도 있다. 따라서 온라인 게임의 서버는 클라이언트로부터 수신된 메시지의 조작 여부를 빈틈없이 검사해야 한다. 또한 온라인 게임에서는 클라이언트에게 게임 내용에 대한 판단을 하게 해서는 안된다. 클라이언트 프로그램은 시중에 떠돌고 있는 각종 프로그램 해킹툴로 어떤 형태로든 변형될 수 있다. 만약 클라이언트에 캐릭터의 HP 수치가 저장되어 있고 그것이 그대로 게임에 적용된다면, 해당 변수의 메모리 위치를 찾아내어 조작, 무적의 캐릭터를 만들어 낼 수도 있을 것이다. 따라서 게임의 진행과 관련된 모든 데이터는 서버상에 있는 데이터만이 유효해야 한다. 클라이언트 측은 어떤 행동을 취하겠다는 '요청'만을 할 수 있으며, 그 요청에 대한 모든 판단과 결정은 서버에서 이루어져야만 한다. 이런 원칙을 지키지 않은 몇몇 온라인 게임은 지금도 각종 해킹 툴을 이용한 공격에 노출되어 있으며 구조가 그런 식으로 되어 있는 한 해킹을 방지하기란 불가능하다.

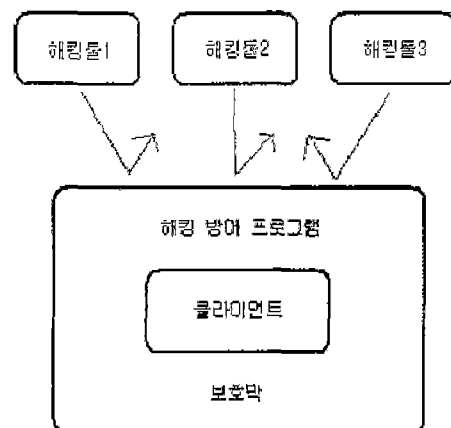
5. 보다 나은 온라인 게임 시스템 모델

5.1 서버-클라이언트 작업 분담

게임서버에서 처리해야 하는 부하가 많아짐에 따라 클라이언트를 분산처리 시스템과도 같이 서버에서 처리해야 하는 작업을 분담시켜 그 결과물만을 얻어와 처리하는 방식이 요즘 많이 등장하고 있다. 그런데 이런 작업 분담은 서버의 부하를 덜어줄 수는 있지만 앞에서 설명한 것과는 같이 해킹을 당할 위험성에 노출되어 있다. 그러나 메시지의 암호화와 더불어 그리 중요하지 않은 작업의 경우 클라이언트의 남은 CPU자원을 이용하여 처리한다면 보다 나은 게임의 구축이 가능할 것이다.

5.2 해킹 공격에 대한 지능적 방어

얼마 전 온라인 게임 업계는 시스템 클럭을 조작하여 게임의 속도를 비정상적으로 빠르게 만드는 모 프로그램으로 인하여 일대 혼란을 겪은 적이 있다. 개인만의 게임플레이에서의 치트의 사용은 경우에 따라 게임의 흥미를 높여주는 요소가 될 수도 있지만 온라인 게임에서의 치트 사용은 게임의 존립여부를 흔들 수 있을 정도로 심각한 문제이다. 때문에 이런 여러 가지 종류의 해킹시도를 방어할 수 있는, 마치 요즘의 바이러스 백신과도 같은 체계적인 온라인 게임 해킹 방어 시스템이 필요하다.

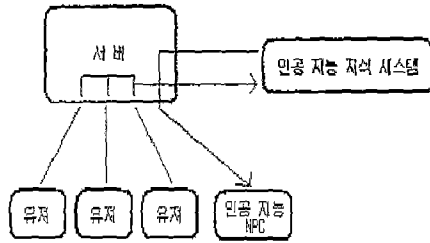


온라인 게임의 위상이 높아질수록 게임 클라이언트의 해킹 방지 기술은 바이러스 백신과도 같이 대단히 중요한 소프트웨어 장르가 될 수도 있다.

그림 6 외부 해킹 방어

5.3 게임상의 인공 생명

온라인 게임은 단순히 게임이라기 보다는 거대한 가상 커뮤니티이다. 게다가 서버-클라이언트 모델로 구현된 온라인 게임은 여러 가지 클라이언트 모듈을 결합시킴으로써 발전될 수 있는 가능성이 무한한데, 그 중 하나가 인공 생명과의 결합이다. 온라인 게임 속에서는 여러 가지 사용자들의 다양한 상호작용이 이루어진다. 이러한 상호작용을 받아들여 지식을 습득하며 성장해 가는 인공 생명을 등장시켜 그 온라인 게임과 함께 성장시켜 그 온라인 게임을 즐기는 사용자 층의 생각과 사고를 대변하는 가상 인격을 만들어 낼 수도 있지 않을까 예측해 본다.



사용자들의 대화, 행동물을 분석, 지식화한 인공지능을 통해 만들어 진 게임상의 가상인격 NPC의 등장. 게임을 오래 한다면 유저가 무슨 행동을 할 지 NPC가 먼저 알고 떠날지도 모른다.

그림 7 게임상에서 구현된 인공생명

6. 맺음말

최근 들어 게임산업 전반에 걸친 재평가가 이루어진 것은 매우 바람직한 일이다. 게임이라는 것이 단순히 어린 아이들의 놀이거리가 아니라 영화와 같은 고부가가치의 문화산업이라는 것을 알리게 한 장본

인은 바로 온라인 게임 인구의 폭발적인 증가였다.

현재의 온라인 게임들은 순수한 게임이라기 보다는 보다 거대한 가상 세계의 건설에 앞서 등장한 일종의 청사진이다. 앞으로 등장할 온라인 게임들은 게임 이상의 방대한 가상 세계를 구현하여 영화산업 이후의 새로운 엔터테인먼트 산업의 주역으로 자리잡을 것으로 기대한다.

참고문헌

- [1] 다크에이지오브카멜롯 홈페이지 : <http://darkageofcamelot.com/>
- [2] 어둠이 전설 홈페이지 : <http://lod.nexon.co.kr/>
- [3] 바람의 나라 홈페이지 : <http://baram.nexon.co.kr/>
- [4] 헬브레스 홈페이지 : <http://helbreath.com>
- [5] 김정주, 온라인 게임 바람의 나라, 정보과학회지, 제 15권 8호, pp.33-41, 1997.8.
- [6] Todd Barron and Andre Lamothe, Multiplayer Game Programming, Prima Tech., 2001.

고 광 현



1995년 단국대학교 전산통계학과(석사)
 1998년 단국대학교 전산통계학과(박사)
 1998.2~2001.12 현 (주)시엔텍 헬브레스 개발 팀장
 관심분야: 온라인 게임, 컴퓨터그래픽스
 E-mail: soph@netsgo.com