

완전한 3차원을 지원하는 온라인 RPG를 위한 맵 관리 방법

이 남 재[†] · 곽 훈 성^{††}

요 약

본 논문에서는 3차원(Full 3D)을 지원하는 온라인 RPG를 위한 맵 관리 방법을 제안하였다. 일반적으로 3차원 그래픽을 지원하는 게임은 클라이언트에서는 지원이 가능하나 온라인 게임 서버에서는 3차원 처리를 위한 데이터 처리량이 게임 월드의 크기에 따라 기하급수로 늘어나기 때문에 현재의 하드웨어 기술 하에서 가격대비 성능을 만족시킬 수 없다. 이를 해결하기 위하여 본 논문에서는 스페셜 오브젝트라는 개념을 도입하여 온라인 게임 서버에서의 3차원처리가 적은 하드웨어 사양을 가지고 동작될 수 있게 하였다. 제안된 방식은 클라이언트에서만 아니라 서버에서도 3차원 맵 관리를 수행할 수 있는 구체적인 형태를 제공하였다.

A Method to Manage the Map for On-Line RPG Supporting Full 3D

Nam Jae Lee[†] · Hoon Sung Kwak^{††}

ABSTRACT

In this paper, we suggested a map management method for on-line RPG supporting full 3D. In general, it is possible to support full 3D with client engine. But the online game server cannot show the expected performance under normal price constrains with current hardware technology, since the amount of data for management of 3D on-line game server increase exponentially with respect to the size of game world. To solve this problem, we introduced the "special object" which makes it possible for on-line game server with low performance hardware. This suggested method gave concrete form to manage full 3D for server as well as clients.

키워드 : 온라인 3D 게임(Online 3D Game), 맵(Map)

1. 서 론

게임산업은 21세기 문화 산업시대를 주도해 갈 중추적인 산업이다. 이중 네트워크를 이용한 온라인 게임 분야는 매년 50%이상의 성장이 지속될 것으로 예견하고 있다[1]. 온라인 게임은 고도리, 포커, 바둑 등 일반적인 사람들이 쉽게 접근할 수 있는 보드게임과 게임에 심취하여 게임의 배경을 현실과 같이 여기는 하드코어 매니아 층이 주류를 이루는 온라인 그래픽 RPG가 있다[1].

하드코어 매니아 층은 기꺼이 게임을 위해서 돈을 지불할 수 있는 사용자를 뜻하며 실제로 리니지, 바람에 나라 등 단기간에 고 수익을 올리는 게임 분야는 대부분 가상세계에서 자신의 분신인 게임 캐릭터를 성장시켜나가는 온라인 그래픽 RPG 형태이다[1].

게임 그래픽은 초기 사용자에게 얼마나 게임을 어필할 수

있는가를 결정짓는 매우 중요한 요소이다. 다시 말하면 어떠한 상품에 대한 첫 번째 호감은 상품의 디자인이나 포장에 따라 달라지듯 그래픽 머드게임에서의 그래픽 구현은 게임의 성패를 좌우할 매우 중요한 요소임에 틀림없다.

현재 게임 화면에 대한 그래픽은 2차원에서 3차원 단계로 진행되는 단계로 되어 있으며 기술적으로 보면 온라인 그래픽 RPG에서의 현재 배경 기술은 (주)엔씨소프트의 리니지와 같이 쿼터뷰를 이용한 2차원, (주)웹젠의 뮤와 같이 3차원 오브젝트를 구현하였으나 카메라 워크 등이 제한 받는 2.5차원 형태 및 (주)나코 인터랙티브에서 제공하는 라그하임과 같은 완전한 3차원 게임 등이 있다.

그러나 라그하임과 같은 3차원 게임은 오브젝트나 캐릭터의 구현이 3차원으로 되어 있는 것만은 확실하나 (그림 2)에 나타난 바와 같이 게임 캐릭터가 성벽에 사다리를 놓고 올라가거나 다리 위와 아래에 동시에 캐릭터가 존재하는 것과 같이 실제 생활에서와 같은 완벽한 3차원 구조는 아직까지 구현하지 못하고 있다.

이러한 이유는 클라이언트에서의 3차원 구현은 현재 하드

† 정 회 원 : (주)다이스넷엔터테인먼트 대표이사

†† 정 회 원 : 전북대학교 컴퓨터공학과 교수

논문접수 : 2002년 9월 12일, 심사완료 : 2002년 11월 2일

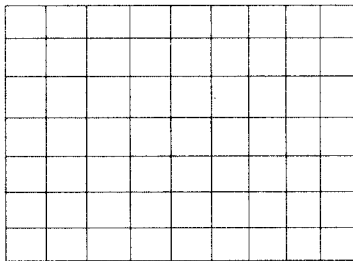
웨어 기술로 어느 정도 가능하나 게임 서버의 경우에는 일반적인 방법을 사용할 경우 현재 하드웨어 기술의 환경 하에서는 구현이 어려우며 구현이 가능하다 할지라도 구현한 하드웨어의 사양에 대한 비용이 천문학적으로 비싸기 때문에 가격대 성능 비에서 이익을 도출하기 어렵기 때문이다.

본 논문에서는 이를 해결하기 위하여 스페셜 오브젝트라는 기법을 도입하여 기존 게임이 이루지 못한 실제 세계와 비슷한 구조를 지원하는 3차원 구조의 게임 배경을 실현하기 위한 맵 관리 방식을 소개하였다.

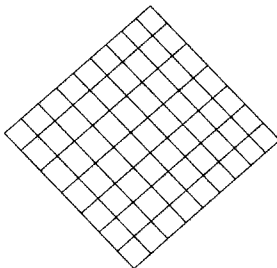
2. 기존 맵 관리 기술

2.1. 2차원 맵 관리 방법

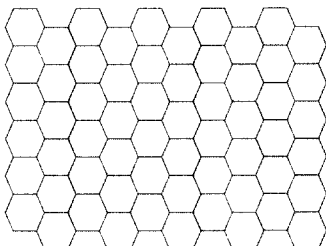
2차원 온라인 게임의 경우 플레이어의 위치 및 오브젝트의 위치를 결정하기 위하여 지형 맵 또는 오브젝트 맵(이하 맵)의 개념을 사용한다. 맵의 형태는 (그림 1)(a), (그림 1)(b)에 나타난 바와 같이 정방형 또는 마름모형의 맵을 사용하며, 경우에 따라서는 (그림 1)(c)와 같이 6각형의 맵을 사용하기도 한다[8].



(a) 정방형 오브젝트 맵



(b) 마름모형 오브젝트 맵



(c) 육각형 오브젝트 맵

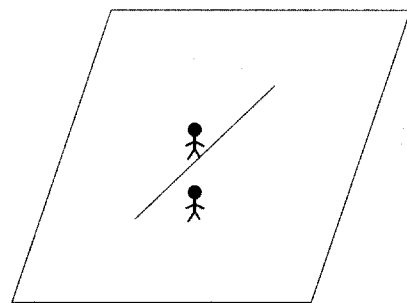
(그림 1) 오브젝트 맵 구조

맵에서 사각, 마름모, 혹은 육각의 형태로 나타나는 각각의 셀(Cell)들은 오브젝트 또는 플레이어가 위치할 수 있는 좌표이며 위와 같은 경우는 2차원의 좌표시스템을 사용한 경우다. (그림 1)(a)의 경우 좌표체계가 간단하여 적용이 쉬우나 쿼터뷰와 같이 입체감을 표현하기 어렵다. (그림 1)(b)와 같은 경우 쿼터뷰와 같은 입체감을 표현하기는 쉬우나 좌표값을 계산하는데 다소 복잡한 면이 있다. (그림 1)(c)의 경우는 특수한 경우로 (그림 1)(a), (그림 1)(b)와 달리 주변에 대한 셀이 8방향이 아닌 6방향으로 되어 있다. 따라서 좌표체계에는 복잡성이 있으나 공격 범위 등 일정한 지역에 대한 공정한 공격력 및 방어력을 적용하기가 수월하다[8].

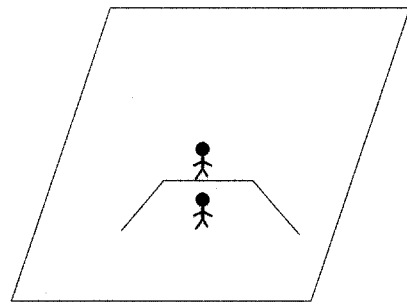
현재의 2차원 맵 기술은 스타크래프트와 같이 마름모 모양의 쿼터뷰 타일을 기본으로 만든 후 이를 집단으로 붙여 직사각형을 만든 집단 타일 형태로 구성되는 형태가 대부분이다. 그러나 앞으로는 모든 게임에서 3차원을 지원하게 될 것이다.

2.2 3차원 맵 관리 방법

3차원 좌표를 사용하는 게임은 일반적으로 2차원 맵을 확장시켜 사용한다. 가장 단순한 형태의 3차원 게임은 모든 개념을 2차원 게임과 동일시하면서 지형 또는 오브젝트만을 3차원으로 처리하는 것이다. 이와 같은 경우, 지형 또는 오브젝트에 대한 3차원 렌더링만 클라이언트에서 처리하면 되므로 간단하게 구현할 수 있으며 이때 서버에서 관리하는 오브젝트 맵의 구조는 2차원 게임과 동일하다. 현재 인기 있게 사용되는 "뮤"와 같은 게임 역시 이런 방식을 사용한다.



(a) 사다리의 표현



(b) 다리의 표현

(그림 2) 기존 방식의 맵 구조에서 3차원 오브젝트의 표현 불가능에

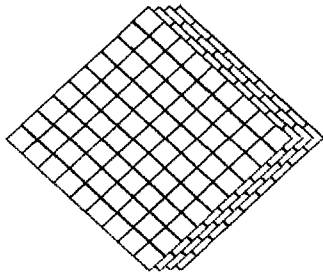
그러나 이 같은 방식을 취한 게임들은 카메라 워크 등을 할 수 없기 때문에 3차원 게임이라기보다는 2.5차원의 게임으로 분류한다. 또한 이 경우 (그림 2)와 같이 사다리나, 다리와 같이 평면좌표상의 동일한 위치에 높이가 다른 2개 이상의 오브젝트가 위치하는 것을 표현할 수 없다. 따라서 완전한 3차원을 지원하기 위해서는 대개 다음과 같이 3가지 방식이 주로 사용된다.

2.2.1 큐브 구조

3차원을 구현하기 위한 일반적인 방법으로는 맵을 큐브와 같이 완전한 3차원 좌표체계를 사용하는 경우이다. 그러나 이런 방식의 경우, 각 오브젝트의 위치 처리를 위해 3차원 좌표 계산을 위한 실수 연산이 반복적으로 사용되게 되며, 오브젝트를 검색하기 위해 많은 시간이 소요되기 때문에 오브젝트 위치 선정을 위한 또 다른 처리가 요구된다. 또한 게임 배경의 크기가 커지면 이에 따른 계산량도 기하급수로 증가하기 때문에 고성능의 게임 서버가 필요하다.

2.2.2 다층 구조

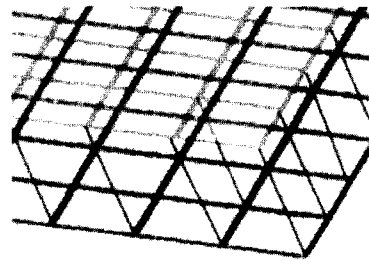
다층 구조 방식은 (그림 3)과 같이 2차원 맵 구조를 여러 층으로 쌓아 3차원 구조를 만들어 주는 것이다. 이 경우 기존의 맵 구조를 가장 손쉽게 확장하여 3차원 게임을 만들 수 있다는 장점이 있다. 그러나 이 경우에도 맵을 표현하기 위해 큐브 방식보다는 적지만 상대적으로 많은 양의 맵 저장 공간을 필요로 한다. 예를 들어 10240×10240의 2차원 맵을 100층으로 쌓아서 3차원을 표현하려 할 때 각 셀의 저장 공간이 4바이트 일 경우 400MB×100 = 40GB의 메모리를 필요로 한다. 이런 경우 층의 개수를 줄이면 어느 정도의 메모리 사용을 줄일 수 있지만 표현할 수 있는 경우의 수가 적어져 게임이 단조로워 지는 문제가 발생한다.



(그림 3) 다층 구조의 맵

2.2.3 공간분할 구조

공간분할 구조 방법 (그림 4)와 같이 서버가 좌표처리를 위해 맵 전체를 한꺼번에 처리하는 대신에 전체 맵을 작은 지역으로 나누고 캐릭터는 자신이 속한 지역만을 검색하는 방식이다. 이 방법을 사용할 경우 다층 구조보다는 계산량이 줄어들지만 캐릭터가 배경 전체에 걸쳐 폭넓게 분산되어 있다면 결국은 모든 공간을 다 처리해야 한다는 결론이 나온다.



(그림 4) 공간 분할구조의 맵

그러나 기본적으로 온라인 RPG의 동시 접속자 수가 수천 명에서 수만 명 이상인 것을 감안하면 이 방법 또한 게임 진행을 원활하게 하기는 어렵다.

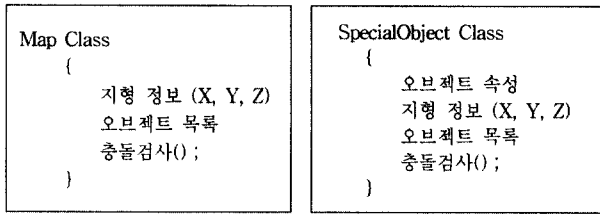
3. 스페셜 오브젝트(Special Object)를 이용한 3차원 맵 관리

본 논문에서 제안한 스페셜 오브젝트를 이용한 3차원 맵 관리 방법은 (그림 5)에 나타난 바와 같이 기본적으로 지형을 표현하는 맵은 2차원을 기본으로 하여 일정간격으로 높이만 주고, 건물이나 기타 오브젝트는 속성을 나눠서 캐릭터가 오브젝트 위로 올라갈 수 있는 경우만 스페셜 오브젝트로 따로 정의하여 실제 3차원 처리를 하는 방식이다. 즉 먼저 우리가 살고있는 실세계를 생각해보면, 우리는 2차원의 지표면 위에서만 움직일 뿐 어떠한 3차원 오브젝트가 존재하지 않는 한 3차원인 빈 공간에 떠있거나 그 공간 안을 자유로이 이동할 수 없다. 즉, 현재 우리는 3차원에서 생활하지만 실제로는 기본적으로 지표면 2차원의 공간에다가 다리나 건물 등 사람이 올라설 수 있는 특수한 오브젝트를 이용하여 3차원의 공간을 활용하고 있다.

(그림 5) 스페셜 오브젝트를 이용한 3차원 맵관리

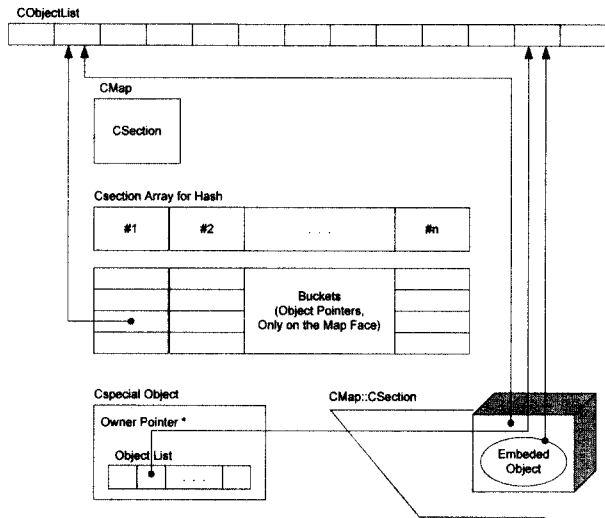
이러한 스페셜 오브젝트를 이용한 3차원 처리 방식은 대부분의 모든 캐릭터가 건물 안에 있는 것이 아니기 때문에 전체 맵에 비해 상대적으로 2차원 구성이 많기 때문에 오브젝트간 충돌처리가 매우 적어지며 이에 따라 맵 관리를 위한 데이터 처리량도 매우 작아져 실제적인 구현 및 적용이 가능하다. 이를 구현하기 위하여 기본적으로 다음과 같은 맵 클래스와 스페셜 오브젝트 클래스를 가진다.

전체 맵은 지형정보와 맵 안의 모든 오브젝트를 가지고 있으며 스페셜 오브젝트는 일반 오브젝트처럼 속성을 가지고 있으며, 맵처럼 지형정보와 오브젝트 목록을 가지고 있다.



3.1 스페셜 오브젝트 처리를 위한 자료구조

스페셜 오브젝트 처리를 위한 자료구조는 전체 오브젝트의 실제 데이터를 가지고 있는 CObjectList와 전체 월드의 지형형태를 가지고 있는 맵을 표현하기 위한 CMap, 그리고 맵을 일정크기로 나눈 것으로 해시형식을 취하고 있으며 맵 위에 올려진 오브젝트만 버킷에 가지고 있는 CSection 및 3차원처리 되는 스페셜 오브젝트로 바닥페이스와 자신이 속해있는 오브젝트(Owner Pointer)와 자신의 내부에 포함된 오브젝트를 리스트로 가지고 있는 CSpecialObject등이 있다. 전체 자료 구조는 (그림 6)과 같다.



(그림 6) 스페셜 오브젝트를 위한 기본 자료구조

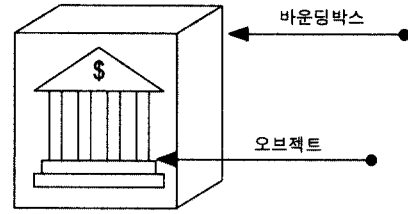
3.2 3차원 오브젝트 간의 충돌

3.2.1 3차원 오브젝트의 기본 구성

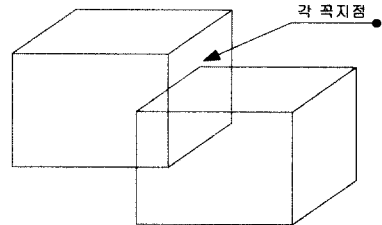
일반적으로 3차원 오브젝트는 (그림 7)(a)와 같이 오브젝트의 크기를 나타내는 육면체의 각 꼭지점 8개로 구성된 바운딩 박스와 오브젝트 고유의 아이디, 오브젝트 속성, 타입으로 정의된다.

(그림 7)(b)는 두 3차원 오브젝트가 충돌하였는지를 체크하는 방법을 보여준 것으로 검사하려는 두 오브젝트 중에 하나를 선택하여 8개의 점 중에 하나라도 다른 오브젝트의 내부에 있다면 충돌한 것으로 판명한다. 그러나 이때, (그림 7)(c)와 같은 경우 문제가 발생할 수 있다. 즉 (그림 7)(c)처럼 두 오브젝트가 내포관계에 있을 때, 작은 오브젝트 입장에서 큰 오브젝트의 꼭지점은 자신의 안에 없으므로 충돌

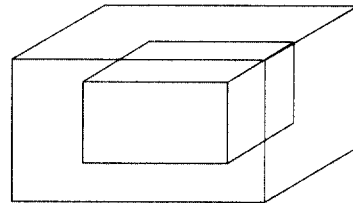
하지 않은 것이 된다. 따라서 충돌처리는 충돌하려는 양자간의 충돌처리를 모두 실시해야한다.



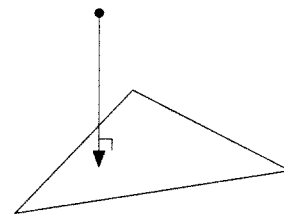
(a) 3차원 오브젝트의 기본구성



(b) 3차원 오브젝트의 충돌



(c) 3차원 오브젝트의 내포



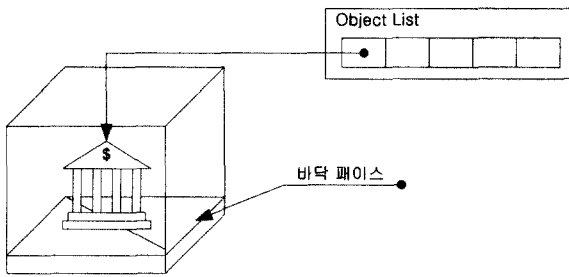
(d) 충돌 판정

(그림 7) 3차원 오브젝트의 충돌 검사

충돌판정을 위한 검사 방법은 (그림 7)(d)에 나타난 바와 같이 8개의 점에 순서를 일정한 순서를 매겨, 삼각형 페이스를 구성하고 이것을 평면의 방정식에 대입하여 원하는 점이 6개의 모든 평면(혹은 12개의 모든 페이스)에 대해 0이하의 값을 가지면 충돌로 판정하는 방식을 사용하면 된다.

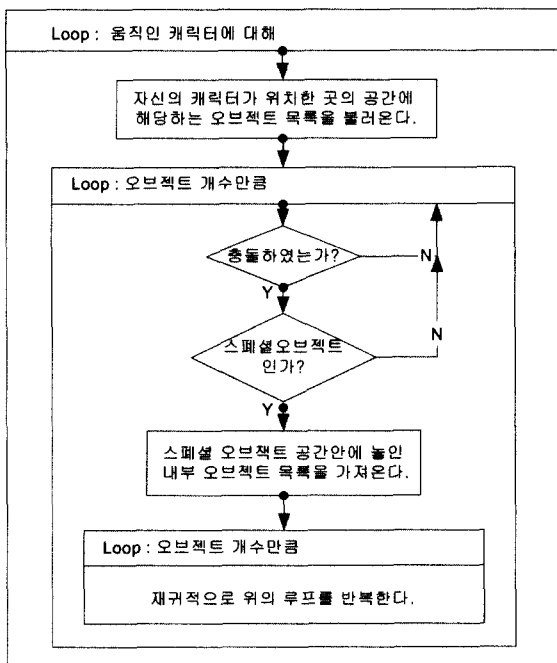
3.2.2 스페셜 오브젝트에서의 충돌과 이동

스페셜 오브젝트는 게임에서 게임 캐릭터가 오브젝트위에 올라 설 수 있도록 정한 오브젝트로써 한 스페셜 오브젝트 안에 부속되는 또 다른 스페셜 오브젝트를 가지고 있을 수 있다. 이를 위하여 (그림 8)에 나타난 바와 같이 바닥 페이스 정보와 거기에 부속되어 있는 오브젝트에 대한 리스트를 가지고 있다.



(그림 8) 스페셜 오브젝트의 구성

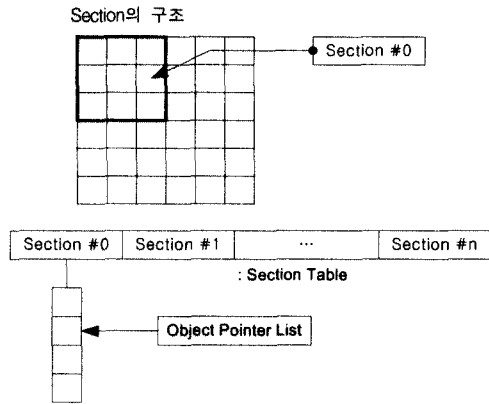
게임 캐릭터가 스페셜 오브젝트를 만나게 되면 먼저 일반적으로 오브젝트의 충돌을 미리 검사한다. 충돌한 오브젝트가 스페셜 오브젝트 일 경우 이동할 좌표가 해당 스페셜 오브젝트의 바운딩 박스 안에 있다면, 그 안의 오브젝트에 충돌을 물어본다. 충돌된 것이 더 이상 없다면 바닥 페이스를 이용하여 Y좌표를 수정해준다. 이런 식으로 재귀적인 방식을 취하고 있다. (그림 9)에 대략적인 순서도를 나타내었다.



(그림 9) 스페셜 오브젝트 처리 방법

3.2.3 스페셜 오브젝트를 적용한 맵 관리 방법

스페셜 오브젝트를 가진 맵을 관리하는데 있어서 하나의 클라이언트가 맵 안의 모든 오브젝트와 충돌 검사를 하는 것은 매우 큰 부하이다. 이러한 문제점을 해결하기 위해서 본 논문에서는 전체의 지형 맵공간을 섹션으로 분할하여 게임 캐릭터가 위치한 부분의 섹션내의 오브젝트만을 충돌 검사한다. (그림 10)에 맵공간을 각 섹션으로 분할하여 관리하기 위한 자료구조를 나타내었다. 여기서 섹션의 크기는 서버의 성능에 따라 조정될 수 있다.



(그림 10) 섹션으로 분리한 맵 공간 관리

4. 결 론

본 논문에서는 클라이언트에서뿐만 아니라 게임서버에서도 3차원 좌표를 처리 지원하는 완전한 3차원을 지원하는 온라인 RPG를 위한 맵 관리 방법을 제안하였다. 기존의 방법 등은 일반적으로 게임 캐릭터와 게임 내 존재하는 오브젝트의 단순한 충돌만을 계산하였으며 충돌시 해당 오브젝트 안, 혹은 위에 위치할 수 있는 처리는 수행하지 않는다. 그 이유 중 가장 큰 것은 서버에서 3차원 좌표처리를 수행하기에는 너무 많은 부하가 발생하기 때문이다.

이를 해결하기 위해서 본 논문에서는 “스페셜 오브젝트”라는 개념을 적용시켜 서버에서 완전한 3차원 좌표가 처리되는 맵관리 방식을 제안하였다. 제안된 방식은 지표면들은 기본적으로 2차원으로 처리하면서 게임 캐릭터의 이동에 의해 충돌할 수 있는 오브젝트들을 충돌시 게임 캐릭터가 오브젝트를 비켜가거나 멈추는 일반 오브젝트와 충돌한 오브젝트 안으로 좌표가 이동되거나 그 위로 올라갈 수 있는 스페셜 오브젝트로 분리하여 처리하였다.

또한 충돌처리를 위한 검사 횟수를 줄이기 위하여 지형맵을 일정한 범위의 섹션으로 나누어 캐릭터가 그 안에 있을 때만 충돌처리를 하게 하여 시스템의 부하를 줄였다. 제안된 방식은 본사에서 개발하고 있는 온라인 RPG인 “Project IF”에 개발에 적용중이며 충돌처리 검사 횟수를 줄이기 위한 적절한 섹션의 크기와 맵 타일(cell)의 크기를 정하는 것은 앞으로의 과제이다

참 고 문 헌

- [1] “게임백서 2002”, 한국게임 산업개발원, 2002.
- [2] <http://www.gamasutra.com>.
- [3] 이남재, 곽훈성, 인터넷온라인 게임을 위한 서버의 분산구성 기법, 한국게임학회논문지, 제1권 제1호, 2001.
- [4] 이남재, 곽훈성, “진화하는 온라인 롤플레이 게임을 위한 분산형 게임 서버 모델”, 한국게임학회논문지, 제2권 제1호, 2002.
- [5] 두길수, 게임 서버 프로그래밍 2002 한국게임학회 동계학술

대회 발표논문집, 2002.

- [6] 주정규, 이원규, 온라인 게임 서버기술연구 2001 한국게임학회 학술발표논문집, 2001.
- [7] 온라인 게임 기술 동향 2002 한국게임학회 동계학술대회학회지, 2002.
- [8] Mickey, Kawick, Real Time Strategy Game Programming Using MS DirectX 6.0, Wordware Publishing, Inc. 1999.

이 남 재

e-mail : njlee@dicenet.co.kr

1988년 전북대학교 전자계산기공학과 (공학사)

1991년 전북대학교 대학원 전산통계학과 (이학석사)

1995년 전북대학교 컴퓨터공학과 박사 수료

1999년~현재 (주)다이스넷엔터테인먼트 대표이사

2001년~현재 한국게임학회이사

2001년~현재 한국게임학회 온라인 게임 분과 부위원장

관심분야 : 인공지능, 컴퓨터 게임 등

곽 훈 성

e-mail : hskwak@moak.chonbuk.ac.kr

1970년 전북대학교 전기공학과(학사)

1979년 전북대학교 대학원 전기공학과 (공학박사)

1981년~1982년 미국 텍사스주립대학 연구교수

1988년~1990년 대한전자공학회평의원 및 전북지부장

1994년~1995년 국가교육연구전산망추진위원

1997년~1998년 전주영상축전조직위원장 및 전북대 영상산업 특성화사업단장

1998년 과학기술법령정비정책위원

1978년~현재 전북대학교 전자정보공학부 교수

1997년~현재 (사)영상산업연구센터대표

1999년~현재 조달청우수제품(정보통신)심사위원

현재 전북대학교 영상산업(학부전공)및 영상공학과(대학원) 주임교수, 한국게임학회 종신회원

관심분야 : 영상처리, 인공지능, 컴퓨터비전, 컴퓨터 게임 등