

# 3D 게임 제작을 지원하는 저작도구의 개발

이헌주<sup>†</sup>, 김현빈<sup>\*\*</sup>

## 요 약

최근 3D 게임의 개발이 활발해지면서 이를 지원하기 위한 저작도구에 대한 관심이 높아지고 있다. 본 논문에서는 개발된 3D 게임 제작용 저작도구를 소개한다. 개발된 저작도구는 맵 에디터, 사운드 에디터, 데이터 추출기 및 프리뷰어, 특수효과 에디터로 구성되어진다. 게임 디자이너는 본 저작도구를 이용하여 게임 맵, 사운드 및 특수효과 등과 같은 게임 데이터를 효율적으로 제작할 수 있다. 본 논문에서는 개발된 저작도구를 활용하여 실험용 콘텐츠를 만들어 저작도구의 유용성을 검증하였다.

## Development of an Authoring Tool for Producing 3D Games

Lee Hunjoo<sup>†</sup>, Kim Hyunbin<sup>\*\*</sup>

### ABSTRACT

Recently, there is a growing consideration that game authoring tools can play an important role in producing 3D games. In this paper, we introduce our 3D game authoring tool. The tool consists of a map editor, a sound editor, a game data exporter/previewer, and a special effect editor. This tool can help a game designer produce game data such as game maps, sound effects, and special effects, and so on. In this paper, we implemented a prototype game content to verify the effectiveness of the developed authoring tool.

**Key words:** Game Authoring Tool(게임 저작도구), Map Editor(맵 에디터), 3D Game(3D 게임), Game Engine(게임엔진)

## 1. 서 론

컴퓨터 게임산업은 지난 수십 년 동안 국내외적으로 많은 발전을 이루었고 주요 문화산업으로 자리잡아 가고 있다. 앞으로도 게임 등의 문화 콘텐츠 산업은 고부가가치 산업으로 국가 경쟁력의 척도가 될 전망이다. 게임 기술은 멀티미디어 콘텐츠 산업의 꽃으로 중요성과 발전가능성에 대해서 더욱 강조되고 있는 것이 현실이다. 특히 국내의 온라인게임 분야는

초고속 정보통신망 인프라의 발전, PC 보급의 확산, 인터넷 사용의 저변 확대 등으로 인하여 급속히 발전하여 세계적으로 인정을 받고 있다. 세계 게임산업은 PC 게임뿐만 아니라 아케이드 게임과 가정용 콘솔게임도 네트워크로 연결되어 온라인게임으로 전환되고 있는 추세로서 온라인게임 분야의 지속적인 성장이 예상된다.

한편, 개인 시스템의 처리 능력이 좋아지고 빠른 속도의 3D 그래픽 카드가 보급됨에 따라 3D 환경은 더 이상 고가의 그래픽 장비를 가지고 있는 특수 사용자만의 전유물이 아니다. 일반 사용자들은 자신의 데스크탑 PC를 이용해 언제든지 3D 환경을 접할 수 있게 되었으며, 3D 그래픽 기술의 발전으로 고품격의 실감나는 렌더링 화면을 실시간으로 생성해낼 수 있는 환경을 가지게 되었다. 이러한 환경에 맞추어, 게임 사용자들의 요구는 2D를 기반으로 하는 게임에

\* 교신저자(Corresponding Author): 이헌주, 주소: 대전시 유성구 가정동 161(305-350), 전화: 042)860-1054, FAX: 042)860-1051, E-mail: hjoo@etri.re.kr

접수일: 2003년 12월 8일, 완료일: 2004년 5월 4일

<sup>†</sup> 정회원, 한국전자통신연구원(선임연구원)

<sup>\*\*</sup> 정회원, 한국전자통신연구원(책임연구원/디지털콘텐츠연구단장)

(E-mail: hbkim@etri.re.kr)

서 3D를 기반으로 하는 게임으로 옮겨가고 있으며 현재 개발되고 있는 게임 가운데 3D를 기반으로 하는 게임이 상당한 비중을 차지하고 있다. 따라서 향후 게임산업은 3D 온라인게임이 서비스의 주류가 될 것으로 예측되며, 국내의 온라인게임 산업도 3D 온라인게임으로 빠르게 변화되고 있는 실정이다[2].

특히, 최근의 게임들은 대규모 게임 맵(map)의 환경에서 인터넷을 통하여 여러 사용자가 동시에 접속해서 게임을 즐기는 롤플레이팅(role playing) 형태의 게임이 발전하면서 게임의 크기가 대형화되어 가고 있다. 다수의 사용자들을 한꺼번에 수용하기 위해 게임 지형은 더욱더 넓어졌으며, 새로운 게임 환경을 구축하고 업데이트하는 작업은 온라인게임에서 가장 많은 시간을 필요로 하는 부분이 되었다. 이러한 작업은 새로운 맵의 구축, 아이템의 추가, NPC의 배치, 특수효과와 추가 등을 포함하게 된다. 따라서 대규모의 게임 환경을 제작하고 관리하기 위한 별도의 저작도구가 필요하게 되었으며, 게임을 개발함에 있어 게임엔진과 유기적으로 연결되어 사용할 수 있는 게임 저작도구의 활용은 더욱 중요시되고 있다[1].

따라서 본 논문에서는 게임엔진과 더불어 게임을 개발하는데 유용하게 사용될 수 있는 게임용 저작도구를 개발하고, 이를 적용하는 것을 목적으로 하였다. 2장에서는 기존의 게임용 저작도구에 대하여 알아보고, 3장에서는 개발된 저작도구에 대하여 구체적으로 기술한다. 4장에서는 본 저작도구의 역할 및 활용 사례를 보이고, 끝으로 5장에서 결론을 맺기로 한다.

## 2. 기존의 게임용 저작도구

게임 제작을 위한 저작도구는 크게 두 가지로 분류할 수 있다. 우선 협의의 게임 저작도구는 사용자들이 만든 도구로, 단순히 게임에서의 캐릭터들의 체력, 재산 등을 바꾸거나 난이도를 조정하는 등 게임 플레이어들이 텍스트 편집기로 바꿀 수 있는 단순한 기능을 지닌다. 이러한 협의의 저작도구를 통하여 플레이어는 지금까지 갈 수 없었던 곳 또는 가보지 못했던 곳을 쉽게 접할 수 있는 기능을 제공받기도 한다. 한편 광의의 저작도구는 전반적인 게임 환경을 구축할 수 있게 하는 것을 의미한다. 여기에는 배경, 아이템의 배치, NPC의 배치, 특수효과 등의 다양한 기능들이 포함되어야 한다. 이러한 작업은 게임을 만

드는 레벨 디자이너 또는 자기만의 게임 환경을 구축하려고 하는 게임 플레이어들에 의하여 이루어질 수 있다. 레벨 디자인은 게임 맵의 구성, 객체의 배치, 임무, 목표 등을 설정하여 게임의 난이도 및 균형을 적절하게 유지함으로써 게임의 흥미를 유발시킬 수 있도록 하는 중요한 작업이다. 이때, 저작도구를 이용하여 각 레벨 또는 스테이지를 직접 디자인할 수 있으며, 게이머가 원하는 많은 부분을 창조할 수 있는 기능을 제공하여야 한다[3,4].

기존의 저작도구 가운데 퀘이크(Quake)라는 엔진과 더불어 사용된 Q3radiant라는 도구는 게이머가 추가로 게임 레벨을 디자인할 수 있도록 하였으며, 하프라이프(Half-Life) 게임에서 사용된 월드크래프트(Worldcraft)는 편리한 사용자 인터페이스를 제공하여 좋은 평을 받고 있다. 언리얼(Unreal) 엔진을 위하여 사용된 UnrealED라는 저작도구는 다양한 기능을 포함하고 있다는 장점을 갖고 있다. 단순한 맵의 구성뿐만 아니라 UnrealScript를 이용하여 스크립트를 편집할 수 있는 기능을 함께 제공함으로써 하나의 환경에서 게이밍과 관련된 모든 편집 작업이 가능하다. 터빈(Turbine)이라는 엔진에서 사용된 저작도구는 디자이너가 원하는 대로 기능을 변경할 수 있는 구조를 가지고 있는 특징이 있다. 따라서 새로운 게임 규칙이 필요할 경우 상대적으로 적은 시간에 구현이 가능하다. WAR3의 월드에디터(World Editor)라는 도구는 게임 플레이어들이 맵의 배치, 아이템의 배치, NPC의 배치, 특수효과 등을 자유롭게 구성할 수 있게 하여 자기 개성을 마음껏 표출할 수 있도록 만든 대표적인 게임 저작도구이다. Fly3D라는 저작도구는 트리-뷰 방식으로 게임상의 모든 엔티티들을 분류하여 나타낼 수 있고, 리스트-뷰의 파라미터 값의 설정에 따라 렌더링에 변화를 주고 그 결과를 확인할 수 있도록 하였다[5,6].

## 3. 게임용 저작 도구의 개발

### 3.1 저작도구의 구성

게임용 저작도구는 게임에서 사용자에게 보여질 그래픽 데이터, 객체, 게임환경 등을 제작하고 구성하기 위한 기능을 제공하여야 한다. 이를 위하여 본 저작도구는 맵(map) 에디터, 사운드 에디터, 데이터 추출기 및 프리뷰어(previewer), 특수효과 에디터를

포함한다. 이들은 각각 독립된 구조로 되어 있으나 게임 제작 시에 게임엔진과 연동되어 사용될 수 있도록 하여야 한다. 맵 에디터는 게임의 지형 및 환경의 제작을 지원하기 위한 것이며, 사운드 에디터는 게임에서 사용되는 음향효과와 제작을 지원하기 위한 것이다. 데이터 추출기는 그래픽 디자이너가 상용 그래픽 제작도구를 이용하여 제작한 그래픽 데이터 및 애니메이션 데이터를 추출하여 게임엔진에서 사용할 수 있는 화일 포맷으로 변환해주기 위한 도구이다. 프리뷰어는 데이터 추출기를 통하여 저장된 3차원 메쉬(mesh)와 애니메이션 화일을 실제 게임과 똑같은 화면으로 보여주고, 상용 3D 도구에서 조절하지 못하는 게임 속성들을 편집할 수 있도록 해주기 위한 것이다. 특수효과 에디터는 게임 상에서 보여지는 각종 특수효과와 제작을 지원하기 위한 도구이다. 그림 1은 게임용 저작도구의 구성을 나타낸다.

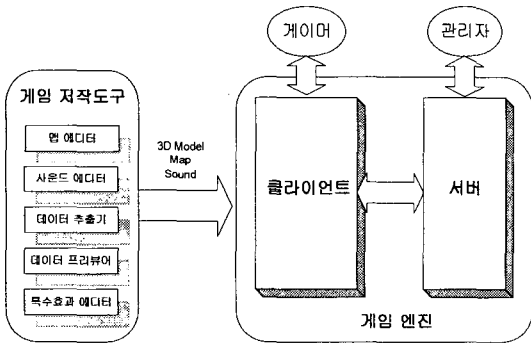


그림 1. 게임용 저작도구의 구성

### 3.2 맵 에디터

맵 에디터는 게임에서 사용될 게임환경을 제작하는 도구로, 게임 제작자의 의도에 따라 지형을 생성하고 객체를 배치하는 등 게임환경의 변화를 눈으로 보면서 손쉽게 제작할 수 있도록 한다. 뿐만 아니라 게임의 객체 혹은 지형에 여러 가지 속성을 설정하여 게임에 반영되도록 하는 기능을 제공하기 위한 게임 저작도구이다.

#### 3.2.1 실외용 맵 에디터

##### (1) 개요

실외용 맵 에디터는 대규모의 지형 및 객체들을 렌더링할 수 있고, 지형 편집, 버텍스(vertex) 색깔 편집, 지형 속성 설정 등의 기능을 통하여 편리하고

효율적으로 지형을 생성할 수 있도록 한다. 대규모의 실외 장면은 렌더링의 양이 많아지므로 이를 빠르게 처리하는 것이 중요하다. 이를 위하여 본 에디터에서는 CLOD(Continuous Level Of Detail) 알고리즘[13]을 이용하였다. CLOD 알고리즘은 넓은 실외 지형을 효율적으로 렌더링하기 위해 가까이 있는 지형 또는 복잡한 지형은 자세히 그리고, 멀리 있는 지형은 한꺼번에 뭉쳐서 간략하게 그리는 방법이다. 또한 게임 지형을 하이트맵(height map) 기반으로 구성하도록 하였다. 하이트맵이란 지형을 일정한 격자 간격으로 나누고, 그 높이 정보만을 기억하도록 하는 방법을 말한다. 하이트맵 기반 지형은 일반적인 지형 모델링 방법에 비하여 자유로운 표현에는 한계가 있지만, 게임 진행 중에도 빠른 시간 내에 지형의 높이 정보를 찾을 수 있으며 지형 관련 데이터 처리가 손쉽다는 장점이 있다. 그림 2는 실외용 맵 에디터의 화면을 보여주고 있다.

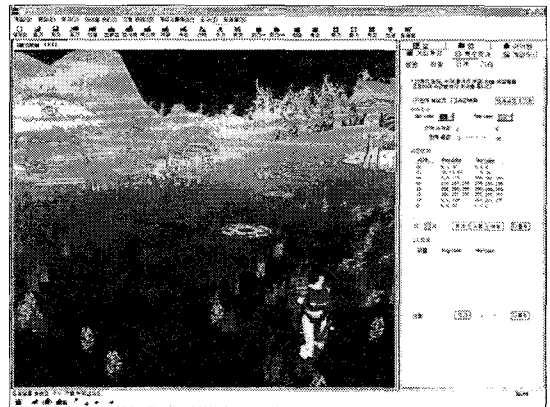


그림 2. 실외용 맵 에디터

##### (2) 지형 생성

###### (가) 지형 구조

대규모의 지형은 한꺼번에 렌더링하는 것보다 적절한 크기로 나누어서 처리하는 것이 효율적이다. 이를 위하여 전체 게임을 구성하는 지형인 리전(region)을 여러 개의 셀(cell)로 나누도록 하였다. 셀은 게임 진행상 게이머의 행위에 반응하고 이벤트를 처리해주는 지형의 단위가 된다. 또한 임의의 게이머가 속해 있는 셀과 주위 8개의 셀을 묶어서 AOI(Area Of Interest)라고 정의하고, 게임 프로세서는 게이머의 게임 진행을 AOI 내에서만 처리하도록 하였다.

하나의 셀은 일정한 간격의 그리드(grid)로 나뉘어진다. 그리드는 지형을 표현하는 구조 중 가장 작은 단위로써, 그리드 한 개가 하이트맵의 높이정보 한 개를 의미한다. 또한 지형에 텍스처를 입히는 단위를 타일(tile)이라고 한다. 즉, 타일은 한 개 이상의 그리드를 묶어서 하나의 텍스처를 동시에 입히기 위한 자료구조가 된다.

#### (나) 지형 편집

지형 편집은 하이트맵의 높이 정보에 따라 지형 메쉬(mesh)를 구성하는 과정을 거치게 된다. 게임 지형을 3DS MAX나 MAYA 같은 전문 모델링 도구를 통해 만들 수도 있지만, 이 경우 게임을 진행할 때 필요로 하는 게임에 특성화된 정보들을 지형에 표현하기가 어렵다. 그러나 맵 에디터를 이용할 경우에 지형의 속성을 포함한 관련 정보들을 함께 처리할 수가 있다. 맵 에디터를 이용하여 지형을 직접 편집할 수도 있지만 이미 만들어놓은 3D 모델 파일의 높이 정보만을 읽어오거나, 높이 정보를 흑백 그레이(grey) 값으로 저장해놓은 로우(raw) 파일 등을 읽어 들여 게임 지형을 구성할 수도 있도록 하였다.

대규모 게임 지형의 경우, 하나의 게임 리전을 여러 개의 셀들로 나누어서 한번에 한 셀씩 생성하고 편집할 수 있도록 하였다. 또한 게임 리전 내의 수많은 셀들을 관리할 수도 있다. 예를 들어, 게임 리전의 크기를 조정하기 위해 셀을 붙이거나 없앨 수 있으며, 임의의 셀을 다른 셀로 이동하거나 복사할 수도 있다.

셀 단위로 게임 지형을 편집할 경우에 이웃한 셀들과의 연속성이 깨어지거나, 지형 텍스처, 버텍스 색깔, 속성들이 이웃한 셀들과 연속되지 않아 부자연스러운 장면이 연출될 수도 있다. 이러한 문제가 발생하지 않도록 하기 위하여 셀 단위로 편집할 때, 이웃한 셀들이 편집 화면에 같이 렌더링되어 참조하며 편집할 수 있도록 하였다.

#### (다) 텍스처 편집

게임 지형은 게이머가 게임을 즐기는 가상공간이므로 현실감을 느낄 수 있도록 만드는 것이 중요하다. 현실감있는 게임 지형을 만들기 위하여 텍스처를 입히는 작업이 필요하며, 지형 편집으로 전체적인 모양을 갖춘 지형에 실사 또는 만들어낸 이미지들을 적절히 매핑(mapping)하여 줌으로써 실제와 가까운 게임 환경을 생성할 수 있다. 본 에디터에서는 다음

과 같은 텍스처 편집이 가능하다.

- 타일 단위의 텍스처 편집 - 텍스처는 타일 단위로 입혀지도록 하였다. 한 타일당 하나의 텍스처가 사용되므로 디자이너들은 임의의 텍스처를 원하는 타일에 매핑시켜 주기만 하면 된다.
- 텍스처 코디네이트 설정 - 타일의 텍스처 코디네이트(coordinate) 편집이 가능하다. 텍스처 코디네이트란 텍스처가 면에 입혀지는 패턴을 지정하는 방법이며, 지형 메쉬의 버텍스마다 가지고 있는 값이다. 텍스처를 90도씩 회전하거나 상, 하, 좌, 우 플립(flip) 기능을 이용하거나, 직접 텍스처 코디네이트 값을 입력할 수 있도록 하였다. 따라서 하나의 텍스처 이미지를 가지고도 다양한 지형 효과를 표현할 수 있다.
- 멀티 텍스처 편집 - 하나의 면에 여러 개의 텍스처를 설정하고 이를 다양한 연산 방법으로 합성할 수 있도록 하여 텍스처 이미지 수를 줄이면서도 다양한 효과를 낼 수 있도록 하였다.

#### (라) 버텍스 색깔 편집

지형의 사실감을 살리기 위한 방법으로 지형 텍스처 매핑 외에 지형 버텍스 색깔의 편집이 가능하다. 버텍스 색깔은 지형 메쉬에 RGB 값으로 지정하며 주로 셰이딩(shading)이나 지형의 얼룩을 표현하기 위해 사용될 수 있다. 지정된 버텍스의 색깔은 그 위에 입혀질 텍스처의 픽셀 색깔과 광원(lightning)의 셰이딩 효과 등과 합성되어 표현된다.

#### (마) 속성 설정

게임 지형은 게이머에게 가시적인 가상공간을 제공해주는 일 외에도 게임을 진행하기 위해 필요한 여러 가지 정보들을 가지고 있다. 게임 지형의 높이 정보를 알아야 게이머의 캐릭터가 게임 중에 이동이 가능하며, 지형의 접근 가능/불가능, 전투 가능/불가능 등의 정보를 알아야 게이머의 키입력에 반응하고 게임을 진행할 수 있다. 이와 같이 게임 콘텐츠에 의존적인 정보에 대하여 디자이너가 에디터를 이용하여 지형을 편집하며 직접 설정할 수 있도록 하였다. 또한 속성을 지정할 경우에 속성맵을 이용하여 가시 화해줌으로서 디자이너가 손쉽게 작업할 수 있도록 하였다. 속성맵은 하이트맵과 마찬가지로 지형의 속성값을 2차원 배열로 기억하도록 하고 색깔로 표현되도록 하여, 디자이너가 눈으로 확인해가며 속성을 설정을 할 수 있도록 하였다.

### 3.2.2 실내용 맵 에디터

#### (1) 개요

실내용 맵 에디터에서는 실내 환경을 렌더링하기 위하여 BSP(Binary Space Partitioning) 트리를 이용해 실내 공간을 구분하고, 각각의 공간에서 보일 수 있는 다른 공간들을 기억하도록 하는 PVS (Potentially Visible Set) 데이터를 미리 만들어 놓도록 하였다. 따라서 게임 진행 중에는 게이머에게 보이는 지형만을 그려주면 되기 때문에 빠른 속도의 게임 진행이 가능하다. BSP 트리를 이용할 경우, 게임 지형은 솔리드 모델(solid model)이어야 한다. 본 에디터에서는 솔리드 모델을 생성하기 위한 방법으로 CSG(Constructive Solid Geometry)를 이용하여 게임 지형 객체들을 만들어내도록 하였다. 즉, 에디터에서 제공하는 기본 CSG 모델 데이터를 가지고 CSG 연산인 union, intersection, difference 등을 적절히 이용하여 복잡한 게임 환경을 구성할 수가 있다. 그림 3은 실내용 맵 에디터의 화면이다.

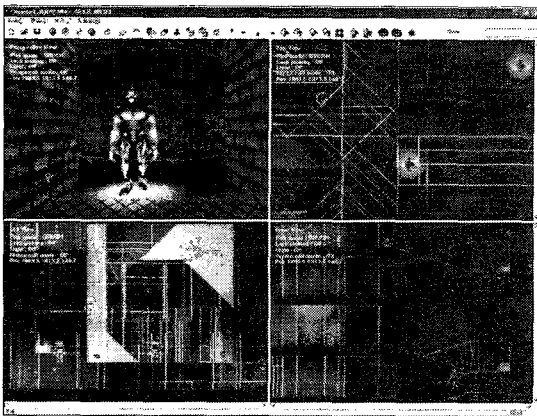


그림 3. 실내용 맵 에디터

#### (2) 지형 생성

실의 지형의 경우 하이트맵 기반으로 편집이 이루어지기 때문에 2D에서 작업하는 인터페이스와 유사한 반면, 실내 맵 에디터의 경우는 게임 환경을 구성하는 CSG 모델 데이터 및 아이템을 3차원 공간상의 적절한 위치에 배치해야 하기 때문에 Quarter View (Perspective, Top, Left, Front View)형식의 인터페이스를 제공하도록 하였다.

#### (가) BSP 트리

게임 상에서의 객체 가운데 지면을 따라 움직이는

물체의 경우는 하이트맵 기반으로 충돌처리를 할 수 있지만, 지면에 상관없이 움직이는 물체의 경우는 하이트맵 기반으로 충돌처리를 하는 것은 불가능하다. 실내용 맵 에디터에서는 CSG 모델로 게임환경을 구성하고 게임환경에 대한 BSP 트리를 구성하여 움직이는 물체와 정적인 환경과의 충돌처리를 하도록 하였다.

BSP 트리는 CSG모델을 구성하는 모든 면들을 이진 트리 구조형식으로 표현한 것으로, 이진 트리의 각 노드는 면의 정보와 그 면의 앞쪽에 있는 면들에 대한 자식 포인터(child pointer)와 그 면의 뒤쪽에 있는 면들에 대한 자식 포인터로 이루어진다. 실제 이러한 BSP 트리를 이용한 충돌검사는 둌(Doom), 퀘이크(Quake) 등의 실내 게임에서 많이 사용하는 방식으로, 충돌검사의 시간을 줄이기 위해서 BSP 트리의 균형을 유지시켜 트리의 깊이를 줄이는 것이 중요하다.

#### (나) CSG 모델

CSG란 물체의 안과 밖이 구분되는 체적을 가진 물체를 말한다. 일반적으로 폴리곤(polygon)으로 구성된 물체들은 면을 가지고 물체를 표현하는 반면, CSG 모델은 폴리곤 정보와 면의 노말(normal) 정보를 이용한 체적의 개념을 사용해서 표현한다. 실내용 맵 에디터에서는 기본적인 CSG 모델 데이터(box, cylinder, cone, 계단)를 바탕으로 CSG 연산(union, difference, intersection)을 수행함으로써 복잡한 실내 게임 환경을 만들 수 있다.

#### (다) 라이트 맵

광원 효과를 위하여 라이트 맵을 사용할 경우, 라이트 반경내의 모든 면들에 대해서 라이트 맵(light map)을 계산해야 한다. 이 경우에 라이트 반경 내에 아주 세밀하게 표현된 물체가 있다면 라이트 맵을 계산하는데 많은 오버헤드(overhead)를 초래하게 된다. 따라서 실내용 맵 에디터에서는 DirectX에서 제공하는 기본 라이트와 라이트 맵을 이용한 광원 효과를 적절히 혼합하여 사용함으로써 게임의 속도와 그래픽 질을 향상시킬 수 있도록 하였다. 캐릭터 등 다소 정밀한 물체에 대해서 DirectX에서 제공하는 라이트를 사용하지만, 상대적으로 폴리곤수가 적은 정적인 게임환경에 대해서는 라이트 맵을 사용해서 광원 효과를 표현할 수 있다.

#### (라) 모델 데이터베이스

실내용 맵 에디터에서는 원하는 게임환경을 제작

하는데 필요한 시간과 노력을 줄일 수 있도록 하기 위하여 모델 데이터베이스를 제공하고 있다. 모델 데이터베이스는 계단, 기둥 등 실내 게임에서 자주 사용되는 CSG 모델들을 데이터베이스화해서 손쉽게 사용할 수 있도록 한 것이다. 데이터베이스에 등록될 CSG 모델들은 에디터를 사용해서 제작할 수도 있지만, MAYA나 3DS MAX와 같은 모델링 도구를 사용해서 만들 수도 있다.

### 3.3 사운드 에디터

#### 3.3.1 개요

사운드 에디터는 게임에서 사용될 오디오 데이터를 손쉽게 편집할 수 있도록 해주는 도구이다. 본 사운드 에디터에서는 비파괴 방식의 편집을 지원하며, 최대 16트랙의 멀티 트랙 편집환경을 지원한다. 또한 wav 포맷의 파일을 모노 및 스테레오 채널로 편집할 수 있도록 하였다. 하드웨어의 지원이 없이도 순수하게 소프트웨어 기능만으로 오디오 데이터를 처리하여 입체음향 효과를 생성할 수 있는 장점을 가지고 있다.

#### 3.3.2 사운드 에디터의 기능

##### (1) 화일 입/출력

오디오 데이터를 불러들일 수 있으며 새로운 프로젝트를 열 수 있다. 또한 프로젝트의 생성, 열기, 닫기, 화일의 입력, 저장 등 화일과 관련된 기능을 제공한다.

##### (2) 편집 기능

오디오 데이터를 편집하는데 필요한 다음과 같은 기능들을 제공한다.

- 작업 복구 및 재실행
- 잘라내기, 복사, 붙여넣기, 삭제
- 슬라이스 나누기: 필요한 슬라이스(slice)를 나누기 위하여 사용된다.
- 마커(Marker): 재생선이 위치한 곳에 마커를 삽입하는 Put Marker 기능이 있다.
- 모두 선택(Select All)

##### (3) 음상 정위 효과

- 포인트 기능: 그림 4와 같이 사용자가 임의의 위치를 선택하여 음원의 위치를 결정하면 해당 위치에 가상적으로 입체음상을 만들어 준다.

- 라인 기능: 사용자가 직선의 시작점과 끝점을 그려주면 음원이 그려진 직선상을 이동하는 듯한 입체음으로 만들어준다.
- 커브 기능: 사용자가 임의의 위치를 선택하여 음원이 움직이는 궤적을 만들면 음원이 지정한 커브에 따라 이동하는 입체음으로 만들어 준다.

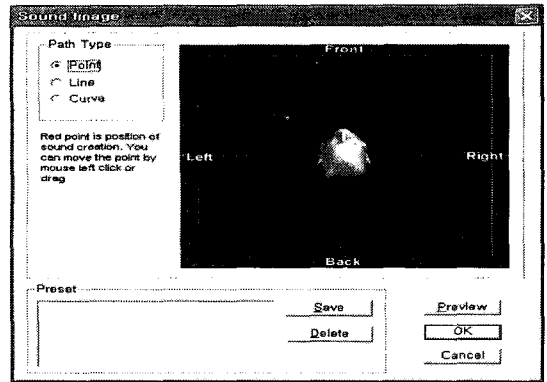


그림 4. 포인트를 이용한 가상적 음원 설정

##### (4) 음장 제어 효과

다양한 공간의 음향 특성을 효과적으로 반영하여 현실감을 극대화하고자 할 때 쓰이는 효과로서 음장 파라미터를 이용하여 콘서트홀, 동굴, 운동장, 집안 등과 같은 환경을 자유롭게 설정할 수 있도록 하였다. 음장 파라미터에는 공간의 크기, 청취자와 스피커의 위치, 벽의 흡음률, 고주파 흡음률 등을 설정함으로써 고유한 음장 효과를 생성할 수 있도록 하였다.

##### (5) 코러스

하나의 보컬 트랙을 둘 이상의 사람이 합창하듯 풍부하게 들리도록 하는 효과로서 입력 신호에 단일 에코를 더함으로써 코러스가 생성되도록 하였다.

##### (6) 플랜지(flange)

페이싱(Phasing)이라고도 하며, 현악기의 장력이 떨리는 것과 같은 효과를 생성할 수 있는 기능이다.

##### (7) 그래픽컬 이퀄라이저 graphical equalizer)

주파수 대역별로 각 대역에서 원하는 주파수 응답 크기를 설정하여 특정 대역을 키우거나 감쇄시킬 수 있는 기능이다. 출력신호의 주파수 응답을 사용자가 원하는 대로 조절할 수 있게 하는 효과로, 기호에 따라 어떤 주파수 대역은 증폭하고 어떤 대역은 줄일 수 있다.

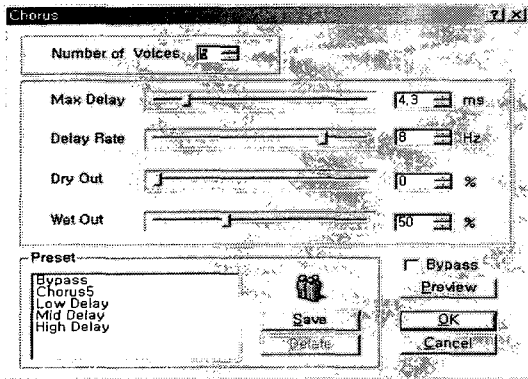


그림 5. 코러스 설정 화면

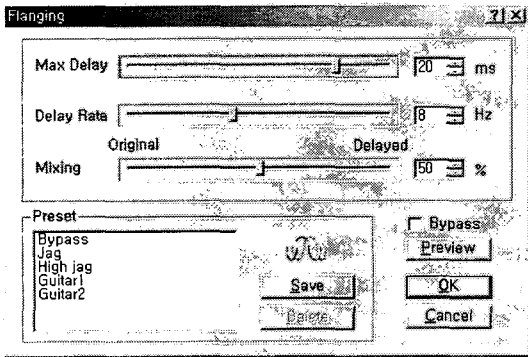


그림 6. 플랜지 설정 화면

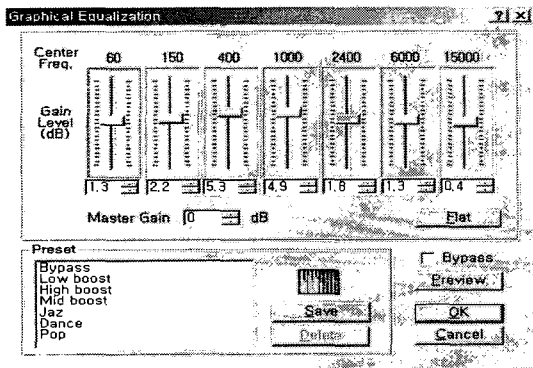


그림 7. 그래픽피컬 이퀄라이저 화면

### 3.4 데이터 추출기 및 프리뷰어

#### 3.4.1 데이터 추출기

##### (1) 개요

데이터 추출기는 그래픽 디자이너가 3DS MAX 또는 MAYA 등을 이용하여 제작한 3D 그래픽 데이터 및 애니메이션 데이터를 추출하여 게임엔진에서

사용할 수 있는 화일 포맷으로 변환해주는 도구이다.

그림 8의 화면은 3DS MAX에 데이터 추출기 플러그인(plug-in)을 추가하여 구동한 화면이다. 왼쪽의 그림은 그래픽 디자이너가 제작한 캐릭터 모델 데이터이다.

그래픽 디자이너들은 자신들에게 익숙한 상용 3D 도구를 이용하여 게임 데이터를 제작하기를 선호한다. 하지만, 이들 도구가 지원하는 화일 형식은 공개되지 않을 뿐만 아니라, 공개되더라도 개발하고자 하는 게임에 필요한 특성이나 속성, 애니메이션 등에 맞지 않는 형식일 수 있기 때문에 게임 개발자들은 자체적인 포맷으로 저장할 도구가 필요하다. 추출기는 이들 상용 3D 도구에 플러그인 형식으로 추가되어 동작하며 그래픽 디자이너는 작업 도중 실제 게임에서의 결과 화면을 보고 싶을 때 추출기로 저장한 후 프리뷰어로 확인할 수 있다.

데이터 추출기는 플러그인 방식으로, 상용 도구 내부의 데이터를 추출하는 부분과 이를 포맷에 맞게 화일로 저장하는 부분으로 나뉜다. 내부 데이터를 추출하기 위하여 각각의 자료구조에 맞는 클래스를 생성한 후 이를 분석하여 게임에 맞도록 변형한다. 여기에는 좌표 체계 변환, 텍스처 좌표 변환, 계층 구조 분석, 프레임별 애니메이션 데이터 추출 등이 포함된다. 이렇게 추출된 데이터를 화일로 만들기 위해서 사용자의 선택사항을 다이얼로그 박스를 통해 얻은 다음, 저장에 필요한 부분을 선택하여 화일 관리자를 통해서 텍스트 또는 바이너리 화일로 디스크에 저장하게 된다.

##### (2) 데이터 추출기의 기능

###### (가) 3D 모델 데이터 추출 기능

버텍스, 폴리곤, 텍스처, 텍스처 좌표, 노말(normal), 머터리얼(material), 변환(transform) 데이터 등 게임에 필요한 데이터를 추출하여 저장할 수 있다. 이 밖에 양면 머터리얼, 계층적 멀티 오브젝트 등 게임에 필요한 데이터 추출 기능을 지원한다.

###### (나) 뼈대 구조 및 애니메이션 추출 기능

캐릭터 애니메이션에 자주 사용되는 방식으로 3D 모델에 뼈대(bone)를 심어 그 뼈대의 움직임을 통해서 모델이 움직이게 하는 스킨(skin) 애니메이션을 지원하기 위한 포맷이다. 뼈대의 구조를 사용자가 편집할 수 있고, 버텍스에 적용되는 가중치(weight)도 1개 이상 적용이 가능하다.

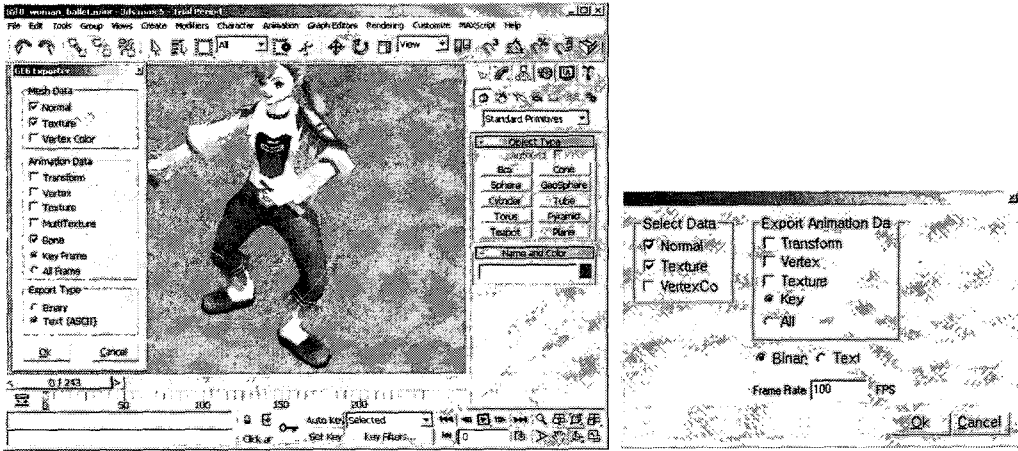


그림 8. 데이터 추출기

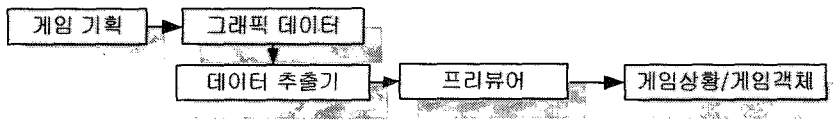


그림 9. 데이터 추출기와 프리뷰어의 역할

(다) 계층적 변환 애니메이션 추출 기능

각각의 오브젝트를 계층적 구조로 연결한 후, 그들간의 변환 데이터를 키 프레임 애니메이션하기 위한 포맷이다. 계층적 애니메이션을 사용할 경우 상위 개체의 애니메이션이 하위 개체에 영향을 끼치게 되고, 하위 개체는 추가적으로 자신의 애니메이션을 사용할 수 있다. 이를 이용하여 기계나 건물 등에 애니메이션 기능을 부여할 수 있다.

(라) 버텍스 애니메이션 추출 기능

오브젝트 내의 버텍스 좌표값들을 키 프레임 애니메이션시켜 모프(morph) 애니메이션할 수 있다. 젤리 또는 얼굴 애니메이션 등과 같이 모델의 형태가 변화되는 곳에 사용된다.

(마) 텍스처 애니메이션 추출 기능

텍스처를 움직여 애니메이션할 수 있다. 텍스처 애니메이션은 멀티 텍스처를 이용한 애니메이션과 텍스처 좌표를 이용한 애니메이션으로 나눌 수 있다. 멀티 텍스처를 이용한 애니메이션은 애니메이션 영화와 같이 여러 장의 그림을 폴리곤에 시간의 순서대로 차례로 보여주어 애니메이션하는 기능이다. 텍스처 좌표를 이용한 애니메이션은 텍스처의 좌표를 바꿔주어 텍스처가 폴리곤 상에서 움직이는 것처럼 보이게 하는 기능이다.

3.4.2 프리뷰어

(1) 개요

프리뷰어는 데이터 추출기를 통하여 저장된 3차원 메쉬와 애니메이션 화일을 실제 게임과 똑같은 화면으로 보여주고, 상용 3D 도구에서 조절하지 못하는 게임 속성들을 편집할 수 있도록 해준다. 프리뷰어는 연동되는 게임엔진의 렌더링 모듈과 애니메이션 모듈, 사운드 모듈을 이용하여 동작하기 때문에 게임화면과 동일한 품질로 보여줄 수 있다. 상용 3D 도구에서 작성한 3D 데이터가 실제 게임에서 어떻게 보일 것인지 미리 볼 수 있으므로 그래픽 디자이너의 의도대로 작성된 결과를 예측하고 편집할 수 있다.

(2) 프리뷰어의 기능

- 애니메이션 도중에 원하는 프레임에서 정지하거나 플레이의 속도 조절이 가능하여, 그래픽 디자이너가 상용 3D 도구에서의 애니메이션과 게임화면과의 차이점을 쉽게 찾아내고 게임화면에 맞도록 고칠 수 있다.

- 다양한 카메라 뷰(front, top, bottom)를 지원하며 마우스를 이용한 이동, 회전, 확대 등을 통해 데이터의 전반적인 모양을 쉽게 파악하고 문제점을 해결할 수 있다.



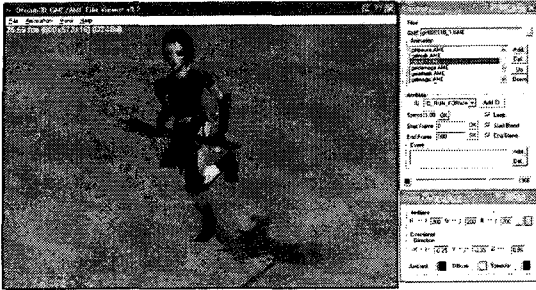


그림 10. 프리뷰어

- 3가지 렌더링 모드(point, wire, solid)를 지원하고, 뼈대 구조를 가지는 데이터의 경우 뼈대 구조를 볼 수 있다.
- 렌더링 엔진의 기능인 페이크 섀도우(fake shadow)를 이용하여 캐릭터의 그림자를 그려줌으로써 사실감있는 형태를 볼 수 있다.
- 창모드를 바로 실행할 수도 있지만, 명령행 인자(command line parameter)를 이용하여 어플리케이션의 시작 시 메시와 애니메이션을 지정할 수 있다.
- 데이터의 일반적인 정보와 성능 프로파일링(profiling) 기능을 제공한다.

### 3.5 특수효과 에디터

#### 3.5.1 개요

특수효과 에디터는 게임에서 사용되는 특수효과를 만들고, 이것을 게임상에서 표현되는 물체들에 유기적으로 결합할 수 있도록 하기 위한 도구이다. 작은 규모의 게임일 경우 프로그래밍으로 특수효과들을 직접 구현하여 사용할 수 있겠지만, 규모가 큰 경우 여러 물체와 특수효과들을 효율적으로 구현하고 관리하려면 에디터의 도움은 필수적이라 할 수 있다.

게임상에서 사용되는 특수효과는 여러 가지가 있으며, 이를 위하여 사용되는 기법으로는 빌보드(billboard), 파티클(particle) 시스템, 메쉬 이펙트(mesh effect) 등을 들 수 있다. 특수효과 에디터는 이러한 기법들로 구현되는 다양한 특수효과들을 게임상의 객체, 애니메이션 등과 함께 결합하여 편집하고 보여줄 수 있는 기능을 제공한다.

다른 에디터와 마찬가지로 특수효과 에디터도 이미 구현된 기술을 조합하여 새로운 것을 편리하게 만들기 위한 저작도구이므로 편리한 인터페이스가 가장 중요한 구성요소가 된다고 할 수 있다. 또한 실

제 게임환경과 비슷한 작업환경을 구성하기 위해서는 게임에서 사용되는 배경화면, 바닥, 객체, 애니메이션을 각각 불러오고 화면에서 보여 줄 수 있어야 한다. 그림 11은 특수효과 에디터의 화면이다.

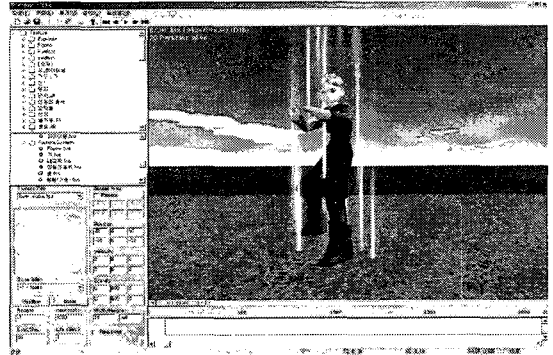


그림 11. 파티클 시스템을 이용한 특수효과

#### 3.5.2 특수효과 에디터의 기능

##### (1) 속성 설정

특수효과는 고유한 속성값을 설정하여 줌으로서 여러 가지 효과를 만들어 낼 수 있다. 예를 들어 파티클 시스템의 경우 초기 파티클들의 속도, 중력 위치 등을 이용하여 파티클들의 운동 방식을 설정할 수 있다. 이러한 속성값들은 게임에서 구현된 특수효과의 속성에 종속적일 수밖에 없으므로 특수효과가 지원할 속성의 범위를 미리 구성한 후에 에디터로 작업을 해야 한다.

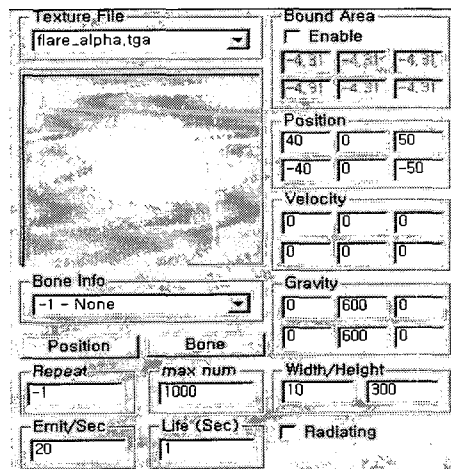


그림 12. 특수효과 속성 설정

(2) 타임라인 에디터

속성값을 애니메이션과 연동하여 세밀하게 편집하고 속성값의 변화를 자유롭게 구현하기 위해서는 타임라인 에디터를 사용할 수 있다. 그림 13의 타임라인 에디터에서 가로방향 축은 애니메이션의 키프레임을 나타내고, 세로방향 축은 속성값을 나타낸다.

(3) 미리보기 창

사용자는 현재 편집하고 있는 특수효과가 실제 게임에서 어떻게 보여질지를 미리보기 창을 통해 확인하고 수정할 수 있다. 따라서 미리보기 창은 실제 게임환경과 유사한 환경을 만들 수 있도록 하였다.

(4) 프레임 조정

특수효과와 실제 게임 애니메이션과의 유기적인 결합을 위해서 애니메이션의 각 프레임 별로 특수효과와의 변화를 간단하게 확인하고 조정할 수 있도록 하였다. 이를 위하여 그림 14와 같이 프레임간을 쉽게 내비게이션(navigation) 할 수 있는 인터페이스를 제공하고 있다.

4. 적용

4.1 게임엔진과 저작도구의 역할

개발된 저작도구와 함께 사용된 게임엔진은 온라인 3D 게임을 위한 엔진으로, 서버 기능을 담당하는 부분과 클라이언트 기능을 담당하는 부분으로 나뉘어진다.

게임 서버는 온라인 게임에서 가장 중요한 역할을

담당한다. 사용자의 관리 및 게임에 등장하는 여러 캐릭터들의 제어 등을 포함한 실제 게임의 진행과 관련된 부분을 서버가 담당하게 된다. 서버는 사용자의 요구를 네트워크를 통하여 받아들인다. 각 사용자의 요구는 이벤트라는 명령 형태로 IOCP(IO Completion Port)를 사용하여 스레드풀(thread pool)로 전달된다. 이렇게 전달된 스레드는 이벤트 프로세서(event processor)를 통하여 처리되며, 처리된 결과는 다시 네트워크로 전달된다. 게임 서버 엔진은 이러한 구조를 사용할 수 있는 기본적인 틀과 이벤트의 처리를 위한 구조를 제공하며, 게임 콘텐츠를 제작할 때 이 틀을 사용하여 게임의 내용을 처리하기 위한 이벤트와 처리 함수를 작성하게 된다.

게임 클라이언트는 온라인 게임에서 사용자에게 게임의 진행 및 상태를 표현하는 역할을 담당하는 부분이다. 실제 게임의 진행은 서버가 담당하며 처리된 결과는 이벤트의 형태로 클라이언트에게 전달된다. 전달된 이벤트는 클라이언트내의 각 세부엔진에 의하여 사용자가 보고 듣고 즐길 수 있는 화면과 음향으로 처리되어 전달된다. 게임의 상태를 사용자에게 표현하기 위하여 렌더링 엔진, 애니메이션 엔진 및 사운드 엔진을 이용하며, 게임의 진행 상태를 서버에 보고하고 서버에서 처리된 결과를 받아들이기 위하여 네트워크 엔진을 이용한다.

게임 저작도구는 사용자에게 보여질 게임환경을 구성하고 게임에 필요한 데이터들을 생성하는 기능을 한다. 게임 저작도구 및 엔진을 이용하면 개발자와 디자이너들은 게임 개발에 있어서 많은 시간과

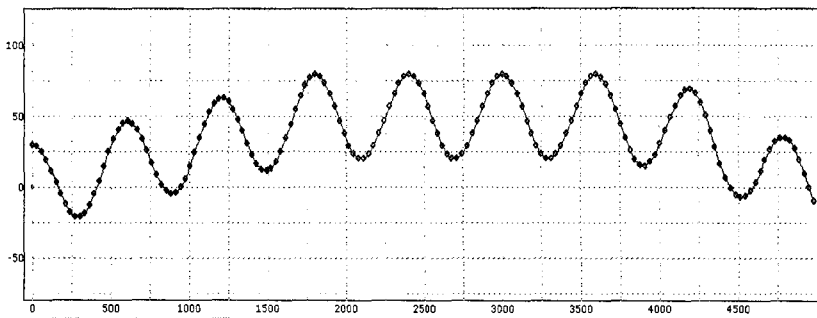


그림 13. 타임라인 에디터



그림 14. 프레임 조정 인터페이스

노력을 절약할 수 있다. 현재 잘 알려진 저작도구에는 케이크라는 엔진과 더불어 사용되는 Q3radiant, 언리얼 엔진에서 사용되는 UnrealED 등이 있다. 표 1은 본 논문에서 개발된 저작도구의 주요 기능과, 대표적인 게임엔진인 Unreal, LithTech, NetImmerse를 지원하는 저작도구들의 기능에 대하여 비교한 결과를 나타낸다.

4.2 시범 콘텐츠 제작

개발된 저작도구의 유용성을 검증하기 위하여 시

범 콘텐츠를 제작하였다. 서버는 4개의 제온(Xeon) 프로세서를 가진 윈도우즈 플랫폼 상에서 구현되었으며, 클라이언트는 엔비디아(NVIDIA)의 지포스(GeForce) 3 그래픽 가속기와 펜티엄 4 상에서 구현되었다.

그림 15는 게임 콘텐츠의 진행 흐름도를 나타낸다. 우선 메인 루틴에서 각 세부엔진들은 초기화되고 게임 데이터들이 파일로부터 로드된다. 다음으로 메인 루프에서는 사용자의 입력에 대한 이벤트나 게임 환경 등이 게임 규칙에 따라 처리되고, 필요에 따라

표 1. 게임 저작도구의 기능 분석(○: 지원, △: 미흡, ×: 지원안함)

| 구성             | 개발된 저작도구<br>주요 기능  | 대표적인 게임엔진별 저작도구 |          |            |
|----------------|--|-----------------|----------|------------|
|                |  | Unreal          | LithTech | NetImmerse |
| 맵 에디터          | ○실외용 에디터<br>- Multi Height map 기반의 지형 데이터 편집<br>- 대형지형 데이터 관리<br>- 게임객체 속성 편집<br>- 스크립트 및 트리거 기반의 FSM 편집<br>- 시간변화, Light 등의 환경 설정 | ○               | ○        | ○          |
|                | ○실내용 에디터<br>- BSP/CSG를 이용한 지형 편집<br>- CSG모델의 Vertex, Face 편집<br>- 텍스처 변경 및 좌표 편집   | ○               | ○        | ○          |
| 사운드 에디터        | ○2채널기반 3D 음상 정위  | △               | △        | △          |
|                | ○3D 음장 제어  | ×               | ×        | ×          |
|                | ○다이나믹 사운드 믹서   | ○               | ○        | ○          |
|                | ○음원모델/음장모델 처리  | △               | △        | △          |
|                | ○특수음향 효과 처리  | ○               | ○        | ○          |
| 데이터 추출기 및 프리뷰어 | ○게임 데이터 변환(MAYA, 3DS MAX)<br>○게임 콘텐츠 프리뷰어  | ○               | ○        | △          |
| 특수효과 에디터       | ○Mesh, Point sprite, Particle system 효과 편집   | △               | ○        | ×          |

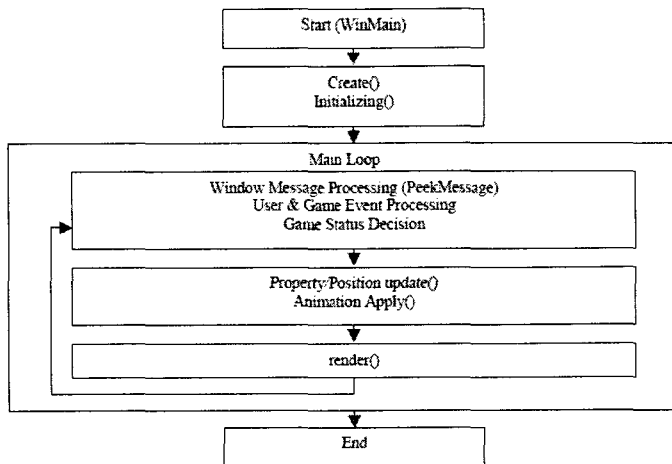


그림 15. 게임 진행의 흐름도

서 애니메이션의 제어, 카메라 및 광원의 변화, 객체의 변화 등을 처리하게 된다. 이후에 각 게임 객체 및 환경은 렌더링 과정을 거쳐 모니터에 보여지게 된다.

게임에서 사용되어질 게임 환경을 만들기 위해서 맵 에디터를 사용하였다. 게임의 지형 편집은 실외 지형의 기반이 되는 하이트맵의 높이 정보를 편집하고, 이에 따라 지형 메쉬를 새로 구성하는 과정을 거쳐게 된다. 그림 16은 에디터를 이용하여 게임 지형을 편집하는 모습을 보여주고 있다.

게임의 캐릭터도 에디터를 이용하여 디자인하고, 모션 데이터를 직접 키프레임(key frame) 형태로 편집하였다. 그림 17은 게임에서 사용된 캐릭터의 제작

화면을 보여주고 있다. 에디터를 이용하여 제작된 캐릭터 및 움직임 데이터는 애니메이션 엔진으로 로딩하여 애니메이션을 처리하게 된다. 게임에서 사용되는 객체 가운데 3DS MAX를 이용하여 제작된 데이터는 데이터 추출기를 통하여 게임엔진에 직접 적용될 수 있는 형식으로 변환되었다. 그림 18은 게임용 저작도구와 엔진을 이용하여 제작된 시범 게임의 실행화면을 나타낸다.

### 5. 결 론

본 논문에서는 3D 게임의 제작을 지원할 수 있는 저작도구를 개발한 결과에 대하여 기술하였다. 본 저



그림 16. 맵 에디터를 이용한 지형 편집

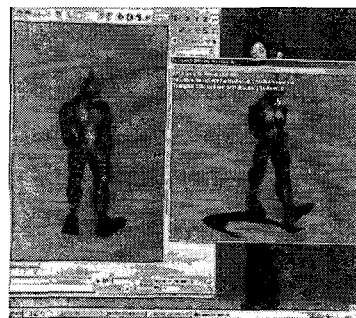
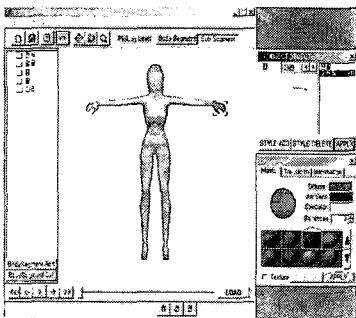


그림 17. 캐릭터 및 애니메이션 편집

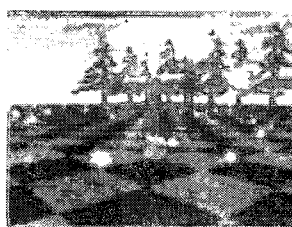


그림 18. 시범 게임의 실행화면

작도구는 게임의 지형 및 환경의 제작을 지원하기 위한 맵 에디터, 게임에서 사용되는 음향효과 제작을 지원하기 위한 사운드 에디터, 게임 디자이너의 그래픽 데이터 제작을 지원하기 위한 데이터 추출기와 프리뷰어, 게임 상에서 보여지는 각종 특수효과의 제작을 지원하기 위한 특수효과 에디터로 이루어져 있다. 이러한 게임 제작 지원도구들은 각각 독립된 구조로 되어 있지만, 게임의 제작 시에 게임엔진과 연동되어 유기적으로 작업이 이루어질 수 있도록 하였다. 또한 게임 제작을 지원하기 위하여 필요한 다양한 기능을 제공하고 있으므로 별도의 상용도구를 사용하지 않아도 편리하게 콘텐츠를 제작할 수 있다. 상용도구를 이용하여 만들어진 게임 객체의 경우에도 데이터 추출기에 의해 파일 포맷을 변형하여, 개발된 저작도구나 게임엔진 등에 손쉽게 적용할 수 있도록 하였다. 개발된 저작도구를 이용하면 다양한 게임 객체를 편리하게 만들 수 있을 뿐만 아니라, 개발 시간을 줄일 수 있는 효과를 얻을 수 있을 것이다.

개발된 저작도구의 유용성을 검증하기 위하여 실험적으로 3D 시범 게임을 제작하였다. 현재 시범 게임 제작을 통하여 나타난 몇 가지 문제점들을 보완하기 위한 연구가 진행 중이며, 향후 컴퓨터 이외의 다양한 플랫폼 상에서의 게임을 개발하는 데에도 적용할 수 있도록 하는 연구가 이루어질 예정이다.

### 참 고 문 헌

- [1] D. H. Eberly, "3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics", Morgan Kaufmann Publishers, 2001.
- [2] Tae-Joon Park, Soon Hyoung Pyo, Chang Woo Chu, and Byoung Tae Choi, "Design and Implementation of a Rendering Engine for Developing Computer Games", Japan Korea Computer Graphics Conference 2001 proceedings, 2001.
- [3] A. Watt, F. Policarpo, "3D Games: Real-time Rendering and Software Technology", Addison-Wesley, 2001.
- [4] A. Watt, "3D Computer Graphics, 3rd edition", Addison-Wesley, 2000.
- [5] 김현빈 외, 온라인 3D 게임엔진 개발, 한국전자통신연구원, 2003. 3.
- [6] 최진성 외, "온라인 3D 게임 기술", 한국정보과학회지, 2001. 3
- [7] 이현주, 박태준, 김현빈, "온라인 3D 게임엔진 개발에 관한 연구" 한국게임학회 논문지, 제3권 제2호, pp 42-55, 2003.
- [8] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Mesh Optimization", SIGGRAPH 1993, pp 19-26, 1993.
- [9] H. Hoppe, "Progressive Meshes", SIGGRAPH 1996, pp 99-108. 1996.
- [10] P. Lindstorm, D. Koller, W. Ribarsky, L. F. Hedges, N. Faust, and G. A. Turner, "Real-Time, Continuous Level of Detail Rendering of Height Fields", SIGGRAPH 1996, pp 109-118, 1996.
- [11] D. R. Begault, "3-D Sound for Virtual Reality and Multimedia", New York, Academic Press Inc., 1994.
- [12] Andrew Rollings, Dave Morris, "Game Architecture and Design," Coriolis, 2000.
- [13] P. Lindstorm, D. Koller, W. Ribarsky, L. F. Hedges, N. Faust, and G. A. Turner, "Real-Time, Continuous Level of Detail Rendering of Height Fields", SIGGRAPH 1996, pp 109-118, 1996.



이 헌 주

1991년 중앙대학교 컴퓨터공학과(공학사)  
1993년 중앙대학교 대학원 컴퓨터공학과(공학석사)  
1998년 중앙대학교 대학원 컴퓨터공학과(공학박사)  
2001년 ~ 2002년 Iowa State

University(Post-Doc.)

1998년~현재 한국전자통신연구원(선임연구원)

관심분야: 인공지능, 가상현실, 게임엔진



김 현 빈

1985년 중앙대학교 학사(응용통계학 전공)  
1988년 중앙대학교 대학원 석사(응용통계학 전공)  
1996년 Okayama Univ. 대학원 박사(전산통계학 전공)  
1991년 3월~1993년 3월 Nagoya

Univ.(연구원)

1984년 11월~현재 한국전자통신연구원(책임연구원/디지털콘텐츠연구단장)

관심분야: 가상현실, 게임엔진, HCI