논 문
CODE 80202

****FPS레벨에디터를 이용한 1인칭 공간시뮬레이션에 관한 연구

A Study on the Tree-dimensional Space Simukation using First Person Shooter leverl Editor

김종현* / Kim, Jong-Hyun
전한종** / Jun, Han-Jong
김석태*** / Kim, Suk-Tae

Abstract

Digital simulation which was introduced to the architectural field due to the rapid growth of computer graphics, gave birth to a new type of contents called "virtual reality", led by the interaction with the users and real time processing. The public attention is drawn to the virtual reality's potential as a next generational space simulation it, having the unique characteristics of "simulation", "interactivity", "tele-presence", and "immersion", is capable of taking a virtual tour of a space with a size equivalent to that of a real space, as well as proceeding with the design progress.

Nonetheless, many problems impeding CPU's real time processing of an excessively loaded architectural model data have been pinpointed over the time. Yet such GPU based game engines as "DirectX" and "OpenGL", developed to deal with these impediments, have not been easily applied to the architectural simulation in the design process, due to the high license cost and the specific technical requirements for the system.

The virtual reality has been developed and distributed centering around the gaming field, and game developers recently show a greater tendency to include level editors in the package for the expandability purpose. Thus, we plan to propose architecture simulation which utilizes level editors in this study. In addition, the compatibility of the game engine based level editors of Quake and Unreal which form the standards for the open source FPS games, based on VRML, the standard format for the virtual reality, was compared and analyzed. Taking the example of Villa Savoye of Le Corbusier, its application possibility as an architecture simulation was assessed, by measuring the extent to which the performance of such characteristic features of the virtual reality as interactivity, immersion, and tele-presence, was improved.

키워드 : 가상현실, 게임엔진, 시뮬레이션, 디자인도구
Keywords : Virtual Reality, Game Engine, Simulation, Design Tool

1. 서론

1.1. 연구의 배경

1960년대 후반 이반 서덜랜드가 개발한 스케치패드가 디자인도구로서의 컴퓨터 그래픽의 가능성을 제시한 이후, 디지털 미디어는 짧은 역사에도 불구하고 디자인 작업에 지대한 영향을 미치는 도구로 성장하였다. 특히 건축, 실내, 환경디자인 등 공간디자인 분야에서는 완성될 디자인에 대한 프로젝트 관계자 간의 정보 공유 및 사전 검증 수단으로 매우 유용하게 활용되고 있다. 작업과정에서의 수작업 못지 않게 작업에 대한 시뮬레이션의 구체화는 건축될 공간에 대한 리스크를 사전에 방지하고, 더 나은 디자인의 발상을 유도하는 매체로도 그 역할을 담당하고 있다. 과거 스케치 형식의 2차원 이미지는 3차원 모델링 데이터를 2차원 이미지로 렌더링 하는 방식으로 발전하였고, 이제는 3차원 데이터를 실시간으로 처리하여 디자이너나 클라이언트에게 가상의 공간을 체험시킬 수 있는 가상현실 기술로 발전하고 있다. 현재 가상현실 분야는 이를 기술적으로 구현하는 그래픽, 네트워크, 상호작용 등 각각의 요소 기술에 대하여 다각도의 기초 연구가 수행되고 있다. 국내에서는 산업의 실제적인 적용은 미미한 수준이나, 군사용 및 엔터테인먼트

* 정회원, 버추얼빌더스 과장급 연구원
** 정회원, 한양대학교 건축학부 조교수, 공학박사
*** 정회원, 인제대학교 디자인학부/유니버설디자인연구소 조교수
**** 이 논문은 2005년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구의 일부임(KRF-2005-003-G00023)

산업의 적용을 중심으로 그 범위가 확대되고 있는 추세이다.[1)]

그러나 가상현실은 실시간 모델데이터 처리를 전제로 하기 때문에 이를 구동시키는 시스템이 매우 고가이며, 감가상각이 크기 때문에 공간디자인 분야에 도입되기 어려워 보편적으로 실용화되지 못하고 있는 문제가 있다. 또 하나의 큰 문제는 가상현실기술이 일반적인 컴퓨터 그래픽 툴의 운용에 비해 고도의 공학적 전문성을 요구한다는 점이다. 그러므로 디자이너가 자신의 디자인을 직접적으로 시뮬레이션 해보고 개선된 아이디어를 디자인프로세스에 반영하는 작업들은 불가능하게 되는 것이다. 이에 대한 대안으로 웹기반 데스크탑 가상현실 기술이 부분적으로 응용되고 있으나, 시스템 부하가 큰 공간모델을 PC에서 운용하기에는 부족한 점이 많다.

3차원기반의 게임과 가상현실 기술은 참여자에게 몰입감을 제공하는 디지털 매체라는 점에서 유사한 특성을 지닌다. 게임의 발달과 더불어 모델링 데이터의 실시간 처리를 위해 등장한 그래픽프로세싱유닛(GPU[2)])는 개인용 컴퓨터에서 효과적인 가상현실 구현의 가능성을 열어주고 있다.

최신의 GPU들은 1초에 수천만 개의 버텍스 좌표를 변환하고 수억개 이상의 폴리곤을 실시간으로 연산한다. 최근 발표되고 있는 NVIDIA와 ATI사 등의 그래픽 프로세서들은 버텍스[3)] 쉐이더[4)](Vertex Shader) 또는 픽셀 쉐이더(Pixel Shader)와 같은 프로그래밍이 가능한(Programmable) 쉐이더 기능을 지원하고 있다. 이렇듯 최근 영상 및 그래픽 처리 분야에서 CPU의 성능 발달 못지않게, 급속도로 발전되고 있는 GPU의 활용 방안 모색은 가상현실을 이용한 공간시뮬레이션 기법에 혁신적 방향전환을 제시해 줄 것이다.

1.2. 연구의 내용

1990년대 후반에 접어들면서 급속도로 게임시장을 장악한 3차원 모델기반 게임은 GPU의 성능개선에 크게 기여했으며, VRML[5)]과 같은 표준 포맷을 구동시키는 웹 기반 클라이언트나 개발사 별로 독자적으로 개발된 비표준 Web3D[6)]구현 기술보다 성능적으로 월등한 콘텐츠를 선보이고 있다. 이러한 게임은 DirectX나 OpenGL과 같은 그래픽전문API를 기반으로 하는 게임엔진에 의해 개발되고 있다. API는 실시간 모델데이터 및 텍스처의 처리와 저수준의 이펙트들을 제공하고 있으므로 개임개발에 필요한 기능들을 개발하여 고수준의 게임엔진개발환경을 제시하는 것이다. 그러나 게임엔진 또한 스크립팅환경으로서 디자이너가 직접적으로 제어하기에는 많은 문제가 있다. 게임엔진은 엔진별로 게임을 제작하는 게임레벨에디터를 제공한다. 일부 게임의 경우 레벨에디터를 게임패키지에 포함시켜 공개하기도 하는데, 이는 고가의 가상현실엔진에 비해 초저가로 구매가 가능하다. 게임엔진은 게임의 장르에 따라 FPS, RTS, MMOG등으로 분류되며 본 연구에서는 공간시뮬레이션으로 의미가 있는 1인칭 시점의 가상체험을 위해 FPS게임엔진을 대상으로 기존의 가상현실 표준포맷과의 호환성검토, 가상공간에 대한 정성적 평가, 화면갱신율의 측정을 통한 플랫폼 적용가능성에 대해 연구하였다.

2. 가상현실 기술적용의 문제점

2.1. VR엔진과 게임엔진의 차이

VR엔진과 게임엔진의 가장 큰 차이점이라고 하면 그것은 VR엔진이 하이엔드급의 하드웨어와 저수준(Low Level)의 환경개발이 가능한 고가의 장비로 구축되어 있다는 점이다. 고가의 전용 실시간 렌더머신과 6D마우스, 사이버글로브, 모션캡쳐와 같은 전용 인풋 디바이스, CAVE나 HMD와 같은 전용 입체 디스플레이 장비와 같은 것들이 고가의 전문소프트웨어에 의해 운용되며, 저수준 언어를 지원하기 때문에 전문적 지식을 갖는 시스템엔지니어에 의해서만이 개발이 가능하다.

그러므로 VR엔진은 의학, 군사, 방송, 산업, 기계, 우주, 예술 등 전 학문분야에 대한 폭넓은 응용이 가능하나, 일반인들이 가상현실을 직접적으로 쉽게 다루기 어렵다는 문제를 동반하게 된다. 이러한 문제를 보완한 것이 2차원 디스플레이를 중심으로 하는 웹 기반 데스크탑 가상현실(WDVR; Web based Desktop Virtual Reality)이다.

2.2. 웹 기반 데스크탑 가상현실

WDVR은 포맷에 따라 표준과 비표준으로 구분된다. 표준은 1994년 W3C컨퍼런스에서 Open Inventor에 기초하여 최초로 언급된 VRML1.0에서 비롯되며, 2년 후 SGI의 그래픽스 무빙월드에 기초한 VRML2.0이 제안되면서 본격화 되었다.

---

1)김형석 · 원광연, 가상현실 기술과 게임_현황과 미래, 게임산업개발원 연구 보고서, 한국과학기술원 전산학과, 2002. 2, p.16

2)GPU(Graphics Processing Unit) : 컴퓨터의 영상정보 처리, 가속화, 신호전환, 화면출력을 담당하는 그래픽카드는 비디오 램과 그래픽 칩에 따라 성능이 달라진다. 중앙처리장치(CPU)의 그래픽 작업으로 인해 생기는 병목현상을 해결하기 위해 만든 것이 그래픽 가속기능으로, 과거에는 그래픽 가속기(Graphics Accelerator)라고도 하였다.

3)버텍스(Vertex) : 가상현실이나 3차원 게임의 모든 물체를 구성하고 있는 정점을 의미하며, 3개의 버텍스로 이루어진 삼각형의 면은 폴리곤이다.

4)쉐이더(Shader) : 3차원 그래픽을 구현함에 있어서 CPU가 담당했던 브분을 GPU가 처리하게 해주는 것

5)VRML(Virtual Reality Modeling Language) : VRML은 인터넷상에서 상호작용하는 3차원 멀티미디어를 기술하기 위한 언어로 ISO/IEC 14772-1의 국제 표준 파일 형식이다.

6)Web3D는 인터넷이라는 가상공간 속에서 구현되는 3D의 전반적인 개념이며, 인터넷상에서 구현되는 가상현실, 인터넷 웹 브라우저 안에서 실시간으로 사용자의 행동에 반응하면서 보이는 3차원적 그래픽 기술을 말한다. 박민수, Web3D 그래픽기술을 활용한 제품 표현기법에 관한 연구, 한성대학교대학원, 2002, p.9

VRML1.0이 인터넷에서 3차원 영상의 기술적 가능성을 보여주는 것에 그친데 반해 VRML2.0은 사용자와의 상호작용과 키프레임 애니메이션을 가능하게 하여 실제적인 표준 포맷으로 자리잡게 되었다. 그러므로 VRML1.0과 VRML2.0은 구조가 완전히 다르며, 문법적으로도 상호간에 거의 호환이 되지 않는다. 이후 VRML2.0을 기술적으로 명문화하고 기능을 일부 수정한 VRML97이 발표되었으며 이는 ISO에서 정식으로 승인된 최초의 국제적 공인 규약이 되었다. 이후 표준 포맷의 발전은 거의 답보상태를 이루다가 2000년대에 이르러 최초로 VRML을 개발한 W3C에 의해 XML과 통합 가능한 X3D포맷으로 논의되고 있다. 표준형 VRML은 노드와 필드의 조합에 의해 가상공간에 존재하는 오브젝트 모델과 환경설정 뿐만이 아닌 센서와 보간자(Interpolator)를 포함하며, 노드간의 데이터 전송 및 이벤트 루팅(Event Routing)을 이용한 노드간 인터랙션, 사용자와 공간간의 인터랙션을 가능하게 하는 명문화된 가상현실 컨텐츠의 기준이라 볼 수 있다. 그러므로 VRML은 다양한 모델링 툴과 가상현실 저작도구간의 데이터 호환용으로도 활용되고 있어 대부분의 모델링 툴과 가상현실 저작도구들이 VRML 입출력기능을 갖추고 있다. VRML은 파일포맷으로 이를 구동하는 클라이언트는 다수의 개발사에서 대부분 무료로 배포하고 있다. VRML이 SGI사에 의해 크게 영향받아진바와 같이 SGI의 Cosmo Player™[7]는 표준형 VRML클라이언트의 고전으로 통하며, 이외에도 일반적으로 많이 알려져 있는 웹클라이언트로는 마이크로소프트사의 World View™, 페러럴 그래픽스사의 Cortona™, 블랙선사의 Blaxxun Contact™ 등이 있다. 이들은 VRML저작도구나 3D커뮤니티를 개발하는 회사들로서 자신들의 생산품에 맞는 전용 뷰어를 개발하면서 표준노드 외에 클라이언트 전용 노드를 제시하고 있다. 이는 전체구조는 표준을 따르고 표준노드도 호환하지만 표준에 등록되어 있지 않은 노드를 추가적으로 사용할 수 있도록 하는 것으로 VRML이 제공하고 있는 확장노드(Proto Node)의 정의와는 성격이 다르다. 그러므로 이러한 전용노드가 사용되는 경우 클라이언트 간의 데이터는 호환되지 않는다. 확장성이 취약한 VRML클라이언트는 자바 라이브러리인 EAI를 이용하여 외부 어플리케이션과의 연동을 가능하게 하지만, 클라이언트에 따라서는 별도로 ActiveX컨트롤과 연동가능하게 하거나 전용 SDK를 별매하기도 한다.

VRML의 결점 중 하나는 용량의 문제이다. VRML은 ASCII 형식으로 누구나 소스에 접근이 가능하다. 반면에 불필요하게 용량이 비대해지고 구조상 보안이 거의 불가능한 문제가 발생한다. 또한 현실감을 올리기 위한 맵 소스 이미지와 사운드는 바이너리이므로 텍스트 기반인 VRML이 이러한 소스들을 포맷 내에 수용하지 못하고 별도로 로딩을 해야 하므로 전체적인 용량은 더욱더 커지게 된다. 이러한 용량과 보안문제는 웹을 기반으로 하는 클라이언트에게 있어서는 치명적인 문제점으로 지적되지 않을 수 없다.

비표준 포맷은 가상현실 표준포맷을 배제하고 개발사별로 포맷을 만들어 전용 저작도구와 뷰어를 통합적으로 제공하는 것을 의미한다. 표준포맷의 한계를 극복하고 일반인들이 손쉽게 콘텐츠를 제작할 수 있도록 하며, 빠른 전송과 로딩을 위해 보통은 압축 바이너리 형식으로 구성되어 있다. 또한 이미지나 사운드를 포맷 내에 수용함으로서 복잡한 데이터 구조를 단일화 시키고 보안성을 강화한 것이 특징이다. 비표준은 개발사별로 포맷을 자체 설계했기 때문에 상호간에 호환이 되지 않으며, 범용적이지 않은 것이 단점이다. 예를 들어 싸이코어사의 Culd3D™나 뷰포인트사의 VET™의 경우 사이버 제품 카타로그를 위한 오브젝트 중심의 가상현실로서 공간을 구현하기 위한 네비게이션이나 충돌감지 등의 방법을 거의 제공하지 않는다.

가상현실 기술을 공간시뮬레이션에 적용함에 있어서 표준형과 비표준형의 공통적인 또 하나의 큰 문제점은 웹 기반이므로 불특정한 클라이언트에 대비하여 로우엔드로 설계되어 있는 것이다. 그러므로 하드웨어적 가속보다는 소프트웨어적 가속에 중점을 맞추고 있으며, 모든 콘텐츠가 웹브라우저나 윈도우즈 그래픽스 환경에서 구동되므로 2차원 기반의 CPU에 의해 제어되는 부분이 많다. 그러므로 가상공간을 상세하게 묘사하기 위한 목적으로 데이터의 용량을 함부로 증가시킬 수 없다. 또한 공간시뮬레이션을 주목적으로 제작된 것이 아니므로 공간환경을 고려한 고수준의 기능들이 거의 전무하며, 렌더링 방식도 대부분 한가지로 획일화되어 있다.[8]

이렇듯 전용 VR엔진은 국내 실내디자인분야 산업체나 관련 연구소 등에서 활용되기에는 높은 비용과 전문성의 요구에 비해 베네핏이 적고, 웹기반 데스크탑 가상현실은 공간시뮬레이션에 대한 한계적인 성능과 지원으로 기존의 시뮬레이션을 완전히 대체하기 어렵다는 문제점이 있다.

3. 게임엔진

3.1. GPU의 발달과 게임엔진의 출현

7)렌더웨어 엔진을 사용하며, 버전 2.1.5을 마지막으로 추가개발이 중지되었다.

8)대표적인 렌더링 알고리즘으로는 다음과 같은 것들이 있다.
- Flat shading : 3D상에 존재하는 오브젝트의 음영표현에 있어서 페이스별로 동일한 밝기(Intensity)로 표현하는 가장 간단한 렌더링 방식
- Gourand shading : 페이스가 아닌 페이스의 정점(Vertex)를 기점으로 밝기를 산정하고 각 정점사이의 면의 밝기를 보간하여 효과를 올린 렌더링 방식
- Phong shading : 페이스 내부의 점에 대한 Normal Vector를 얻어 렌더링하는 방식

과거 게임은 특정성별이나 연령층에서만 즐기던 일종의 기호적인 취미였다. 그러나 인터넷이 보급되면서 가정용 PC가 보편화되고 고급화되면서 최근 들어 게임수요자는 연령과 계층을 초월하고 있다.

1990년대 후반부터 게임분야에서 나타난 가장 두드러진 특징은 온라인화와 게임공간을 구성하는 요소들이 2차원에서 3차원으로 전환되기 시작하였다는 것이다. 이는 첨단기술에서만 실험적으로 응용되어오던 가상현실 기술을 일반인들이 친숙하게 접할 수 있는 계기를 제공하였으며, 게임시장의 폭발적인 확대로 이어졌다. 이에 발맞추어 과거의 게임개발의 주관점이 룰이나 점수 등에 초점이 맞추어져 있었지만 최근 들어서는 타격감이나 현실감 등이 게임의 성공요소로서 더욱 크게 작용하게 되었다. 이러한 게임산업의 성장은 그래픽프로세스유닛(GPU)의 발전으로 이어졌다. 컴퓨터의 운용체제가 GUI화 되고 이어 멀티미디어로 진화함에 따라 CPU는 MMX기술을 탑재하였으며, 그래픽 환경이 3차원 중심으로 전환되면서 그래픽 인터페이스 제작자들은 3차원 그래픽을 전문적으로 가속시키는 보조프로세서를 요구하였기 때문이다. 표1과 같이 프로세서들이 Direct3D나 OpenGL과 같은 그래픽 API 명령어 처리의 상당부분을 담당하게 되면서 고화질의 빠른 화면갱신률을 갖는 3차원 그래픽환경을 구축할 수 있게 되었다. 그러므로 최근 3차원 게임들의 그래픽 구현능력과 질적수준의 보장은 CPU보다 GPU의 성능에 의해 구동능력이 좌우된다 해도 과언이 아니다.

<표 1> PC용 그래픽 프로세서의 변화[NVIDIA 1999]

	1996	1997	1998	1999
Application tasks (move objects according to application, move/amin camera)	CPU	CPU	CPU	CPU
Scene level caculations (object level culling, select defail level, create object mesh)	CPU	CPU	CPU	CPU
Transform	CPU	CPU	CPU	CPU
Lighting	CPU	CPU	CPU	GPU
Triangle Setup and Clipping	CPU	Graphic Processor	Graphic Processor	GPU
Renderling	Graphic Processor	Graphic Processor	Graphic Processor	GPU

GPU는 과거 SGI의 GLZ보드와 같은 가상현실 전문 그래픽 장치가 아닌 일반 개인용 PC를 겨냥한 것이기 때문에 전문보드에 비하여 초저가이며, 소규모 PC에서 작동하는 3DSMax, Maya, Lightwave와 같은 모델러나 캐드시스템을 지원하므로 활용방안과 적용방법만 연구된다면 실내디자인이나 건축설계분야에서 상당히 유용하게 사용될 수 있을 것이다.

3.2. 게임엔진의 종류

게임엔진의 종류는 참여자 시점의 인칭에 따라 일반적으로 3가지로 분류한다. 첫 번째 Isometric Engine은 3인칭 관점으로 광범위한 데이터로 초기에는 2D그래픽을 기법을 이용하여 시작하였으나, 공간 내의 오브젝트가 점차적으로 3D모델로 전환되어 최근 출시되는 엔진은 Full-3D환경을 지원한다. Isometric 엔진을 이용한 게임의 장르는 대부분 RTS(Realtime Strategy)이므로 RTS엔진이라고도 한다.

두 번째, 3D FPS 엔진으로 과거 울프스테인3D를 효시로 하는 1인칭 FPS(First Person Shooter)류의 게임을 제작하는데 사용하는 엔진이다. 이 엔진은 실제 체험자의 시각으로 공간을 체험할 수 있기 때문에 실내공간 및 환경시뮬레이션에 매우 적합하다. 그러나 아이소메트릭 엔진에 비하여 렌더링 범위에 들어오는 데이터의 양이 불규칙하고, 경우에 따라 실제모델의 절반이 넘는 방대한 양의 데이터를 실시간으로 처리해야 하는 상황이 발생할 수 있다. 그러므로 엔진에 따라 LOD[9]나 Culling system[10]을 도입하는 경우가 있는데 이는 1인칭 공간가상시뮬레이션을 위한 엔진의 선택에 있어서 고려해야할 중요한 요소이다.

세 번째, MMOG(Massively Multi-player Online Role Playing Game)엔진은 동시에 많은 게이머가 동시에 플레이 할 수 있는 온라인 게임을 의미하며, 최근 인기를 끌고 있는 리니지, 바람의 나라 등이 여기에 속한다. MMOG게임은 대부분 1인칭과 3인칭의 중간형식(체험자의 시점이 자신의 아바타 포함하여 3인칭적으로 보여지는)으로 진행되고 네트워크 모듈의 성능에 치우쳐진 캐릭터 중심으로 동작하기 때문에 실내디자인분야에서 응용하기엔 적합하지 않다

3.3. 게임엔진의 역할과 장점

GPU를 직접적으로 제어할 수 있는 Direct3D나 OpenGL은 개발자를 위한 저수준 API이므로 LOD를 비롯한 폴리곤, 텍스처의 처리나 버텍스 셰이딩과 같은 기초적인 함수만을 제공한다. 그러므로 게임개발을 위해 필요한 것이 이러한 API를 이용하여 제작된 게임엔진이다. 게임엔진은 전장에서 분류한 게임의 장르별(RTS, FPS, MMOG)로 각각의 특성에 맞도록 서로 다르게 설계되어 있으며 대부분 <표 2>와 같은 모듈로 구성되어 있다.

<표 2> 게임엔진 구성모듈

모듈	개념
Graphic module for 2D/3D	폴리곤과 텍스쳐의 실시간 처리, 라이팅 연산 등 렌더링에 관련한 연산을 담당하는 모듈
Physic module	공간의 중력, 공기저항, 마찰력, 경도등 물성을 묘사하는 모듈
Collision detection module	오브젝트간의 충돌, 아바타와 공간과의 충돌을 감지하는 모듈
Input/output module	키보드, 마우스, 조이스틱등 입출력관련부분 제어 모듈
Sound module	입체음향, 샘플링 등을 제어하는 모듈
Artificial intelligence module	컴퓨터에 의해 동작하는 인물(Bot)을 제어하는 모듈
Network module	멀티유저게임을 위해 네트워크에 의해 공간을 공유하게 하는 모듈
Database module	환경설정 또는 게임에 관련한 데이터를 저장 및 추출하는 모듈
GUI module	사용자 메뉴 및 3차원상에서 2차원 HUD를 제어하는 모듈

9)Level of Depth : 장면의 거리별로 모델을 따로 설정하여 시점에서 가까울수록 디테일한 모델로 먼 것은 단순한 모델로 대체시켜 화면갱신율을 올리는 기법

10)시야각에서 제외되는 부분들을 연산에서 포함시키지 않는 기법으로 충돌감지나 라이팅 효과 등 CPU측의 실시간 연산을 요구하는 부분이 많은 콘텐츠 일수록 효과가 크다.

공간의 인지는 시각적인 부분이 압도적인 비중을 차지하기 때문에 공간시뮬레이션에서 가장 중요한 부분은 그래픽 모듈이다. 시각적 효과면에서 기존의 가상현실저작도구들은 공간적인 요소들에 대한 특별한 지원이 거의 없었다. 그러나 게임엔진에서는 종류별로 약간의 차이는 있지만 표 3과 같이 게임 배경디자인에 필요한 다양한 환경효과들을 제공하고 있으며, 대부분 디자이너들이 레벨에디터 상에서 제어가 가능한 수준이다.

<표 3> 게임엔진에서 제공하는 환경표현 기술

Culling system	시야에 들어오지 않는 오브젝트를 실시간 연산에서 제외시키는 기능
Mip map	자세한 텍스처를 미리 준비했다가 근거리에 접근했을때 묘사하는 기능
LOD	거리에 따라 모델의 디테일을 변화시켜 대규모공간을 효율적으로 관리
Environment Map	일몰, 일출, 기후 등을 공간에 통합적으로 매핑
Light Maps	간접광원에 의한 음영과 그림자를 모델에 통합적으로 매핑
Dynamic Shadow	실시간으로 움직이는 오브젝트의 그림자를 연산
Volume Light	구름사이로 비치는 태양광의 퍼짐이나 틈새로 들어오는 빛줄기 연출
volume Fog	일정거리 이상 떨어진 오브젝트에 안개효과를 가하여 원근감 증대
Terrain	지형생성 및 지표면의 텍스쳐 브러싱 기능
Particle system	연기나 불꽃과 같은 입자애니메이션
Mirror	대규모 공간을 구성하기 위한 BSP, PVS등의 공간포탈
Physics Engine	중력, 바닥의 마찰면, 물의 부력과 같은 물리적 효과
Advanced Shader	실제와 같은 음영을 갖는 텍스쳐 효과

현실감이 훼손되지 않으려면 최소한 초당 30회 이상 3차원 공간 데이터를 2차원 이미지로 재구성해야하기 때문에 GPU의 활용이 절대적이지만 공간이 대규모화 되거나 상세하게 묘사되는 경우 GPU의 성능만으로 해결되지 않는다. 특히 1인칭 시점의 시뮬레이션의 경우 체험자의 시야 내에 순간적으로 방대한 폴리곤 데이터가 들어오기 때문에 이를 PC수준에서 실시간으로 처리하는 것은 쉽지 않다. 고가의 VR엔진을 탑재한 전문장비의 경우는 하드웨어의 성능이 뛰어나기 때문에 이러한 문제를 어느 정도 해결할 수 있으나, PC환경에서 구동되는 웹기반 데스크탑 클라이언트는 원칙적으로 용량의 제한을 받기 때문에 실제감이 떨어질 수밖에 없다. 그러나 게임엔진의 경우 부하저감을 위해 Culling system, LOD, 1p mesh z-buffer와 같은 기술들을 제공하고 있으므로 이러한 문제를 상당부분 해소할 수 있다.

4. 공간시뮬레이션 도구로서의 평가

4.1. 평가대상 및 방법

시각적 측면에 대한 가상현실의 성능평가는 정량적으로 수행하기 어려운 난점들이 매우 많이 존재한다. 하드웨어적 특성과 탑재되는 모델의 형태에 따라 화면갱신율의 차이가 크며, 이를 구동하는 하드웨어 인터페이스의 종류에 따라서도 주행감이 크게 달라진다. 또한 콘텐츠를 구성함에 있어서도 서로 다른 수많은 매개변수가 개입됨으로서 비교하고자 하는 두 개의 가상공간을 동일한 수준으로 맞추는 것은 불가능하다. 그러므로 본 평가는 다음과 같은 3가지 요소에 초점을 맞추어 진행하였다.

첫 번째 게임엔진 기반의 가상현실콘텐츠가 기존에 실내디자인분야에서 사용되던 WDVR을 충분히 대체할 수 있는가에 대한 호환성 평가이며, 두 번째 CPU에 주로 의존하고 있는 WDRV와 게임엔진의 화면갱신률 향상에 대한 정량화된 상대평가, 세 번째 게임엔진에 의해 구축된 가상공간의 체험감이 WBVR에 비해 얼마나 성능이 향상될 수 있는가를 실시간상호작용, 몰입, 실존감, 3가지 척도에 의해 정성적으로 평가하였다.

평가대상으로는 기술 난이도와 3D 그래픽 기술이 우수한 FPS엔진 2종을 선택하였다. 대상게임엔진의 선정은 국내 게임평가 포털 사이트의 평가 자료[11]를 토대로 FPS게임 엔진의 양대 주축을 이루고 있는 id Soft사의 퀘이크 3엔진과 Epic game사의 언리얼 2004엔진을 대상으로 하였다. 이 어플리케이션들은 개인의 연구 이용과 학교 및 연구소에서 비영리적으로 이용이 가능한 실시간 3차원 게임엔진이며, 두 게임 모두 게임 환경 제작에 대해 실시간으로 확인이 가능한 레벨에디터를 제공하고 있어 디자이너에게 별도의 프로그래밍 능력을 요구하지 않고 즉시 활용이 가능하여 디자인 과정에 적용시키기 유리한 이점이 있다. 표준VRML 클라이언트는 패러럴그래픽스의 Cortona v3.0을 사용하였으며, CPU기반의 환경에서의 측정을 위해 렌더링 환경을 R98 Render로 설정하여 비교평가를 수행하였다. 평가단은 공간디자인분야 대학원이상의 학력을 가진 전문가그룹 5인으로 선정하였으며 시뮬레이션 평가의 결과가 폭넓은 플랫폼에 적용될 수 있도록 하기 위해 실험에 사용된 PC는 게임의 최소 PC 사양을 수준으로 설정하였다.

4.2. 게임엔진의 호환성 평가

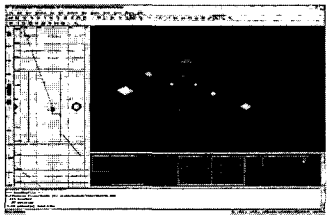
게임엔진의 표현에 대한 호환성 평가 요소는 공간 시뮬레이션 측면에서 정의된 가상현실 세 가지 특성을 바탕으로 하여 표준 포맷인 VRML의 노드에서 추출하였다. VRML에서 노드는 장면을 구성하는 기술 표현 요소로서 이를 바탕으로 게임엔진기반의 레벨 에디터가 이를 얼마나 지원할 수 있는 가로 그 가능성을 평가하였다.

디자인 분야에서의 적용성 평가요소는 가상현실 컨텐츠의 제작성, 호환성, 전문성 세 부분으로 분류하여 평가하였으며, 세부적으로 제작성 부분은 객체 생성에 대한 과정, 재질, 광원의 설정에 따른 작업 프로세스 전반에 대하여, 호환성 부분은 생성 파일의 범용성과 확장성에 대하여, 전문성 부분은 툴 프로그래머 중심의 작업을 반드시 거쳐야 하는가의 문제와 디자

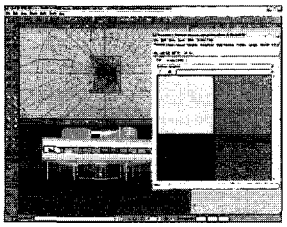
11)국내 게임평가 포털 사이트 9곳에서 1인칭 슈팅 게임에 대한 평가를 중심으로 선정하였다. 9곳의 포털 사이트는 게임조선(http://game.chsun.com), 게임세상(http://www.gamess.co.kr),게임터보(http://www.gameturbo.com), 게임스팟(http://www.gamespot.co.kr), 게임메카(http://www.gamemeca.co.kr), 게임그루(http://www.grui.co.kr), 게임신문(http://www.thegamenews.com), 경향 게임스(http://www.khgames.com), PC Player(http://www.pcplayer.co.kr)이다.

이너 중심의 전문성에 대하여 각각 평가하였다.[12]

분석은 공간표현을 위한 VRML노드를 지표로 하여 퀘이크와 언리얼의 레벨에디터인 GTK Radiant, UnrealED의 기능을 직접적으로 분석하였다. <표 4>는 애니메이션 기능을 제외한 VRML의 공간 표현 기술 노드와 선정된 2개의 게임레벨에디터의 비교평가 결과로서 VRML 공간 표현 기술이 퀘이크, 언리얼 레벨에디터로도 대부분 구현이 가능함을 알 수 있다. 다만 퀘이크의 경우는 LOD노드와 같은 기능은 제공하지 않고 있었다. <표 5>는 퀘이크 엔진과 언리얼 엔진의 기능을 좀더 세부적으로 비교한 것으로, 기능 비교의 기준은 3D 게임엔진의 품질 평가 기준의 내용에서 공간 시뮬레이션에 관한 기본 사항만을 선정하여 평가하였다.[14] <표 6>은 체험성과 화면갱신률 측정을 위한 실험모델을 제작하면서 퀘이크 엔진의 GTK-Radiant와 언리얼 엔진의 UnrealED의 제작성, 호환성, 전문성 영역의 비교로서 언리얼 엔진의 효율이 좋은 것으로 판단되어 이후 체험성과 화면갱신률 측정실험은 언리얼 엔진으로 수행하였다. <그림 1>과 <그림 2>는 실험모델 제작에 사용된 퀘이크3 엔진의 레벨에디터인 GTK-Radiant와 언리얼 엔진의 레벨에디터인 UnrealED의 사용자 인터페이스이다.



<그림 1> Gtk Radiant



<그림 2> UnrealEd

4.3. 화면갱신률 측정 실험

가상현실 컨텐츠에서 화면갱신률 측정은 장면 연출에 있어 중요한 평가 요소이다. 화면 갱신율의 수치가 높다는 것은 그만큼 자연스러운 장면 전환이나 순간 데이터 처리 능력이 높음을 의미하며, 이것은 컨텐츠에 대한 몰입이나 실시간 상호작용과 연결되는 중요한 요소가 된다. 역으로 해석한다면 화면갱신율이 높은 뷰어일수록 좀 더 넓고 정밀한 모델을 수용할 수 있는 것이다. 이에 기존의 방식인 VRML과 언리얼 엔진을 이용해 실험을 위해 제작된 가상공간의 FPS 측정을 실시하였다. 측정에 사용된 프로그램은 GRAPS로서 OpenGL이나 DirectX

12)프로세스 분석에 있어 세가지 분류는 한국게임산업개발원의 게임엔진 품질평가를 기준에서 그 해당요소를 추출하여 분석하였다. KGDI 연구보고서, 게임엔진품질평가 기술, 한국게임산업개발원, 2002, pp.45~57

13)PVS(Potentially Visible Set) : 보여질 가능성이 있는 리스트를 기억해 두는 방식이다. PVS는 그 특성상 BSP 같은 공간 분할과 포털이라는 테크닉을 같이 사용해 구현한다. 3D 엔진기술 해부, 월간 마이크로소프트웨어, *2001. 9*, p.209

14)KGDI 연구보고서, 게임엔진 품질평가 기술, 한국게임산업개발원, 2002, pp.45~57

<표 4> VRML 공간 표현 기술 노드의 게임 표현 (퀘이크 III 아레나)

Group	VRML Node	Quake3	Unreal 2004
그룹노드	Billboard	기능지원	기능지원
	Collision	기능지원*	StaticMeshActor Collision
특별노드	LOD	지원안함	기능지원
일반노드	AudioClip	기능지원	기능지원
	DirectionalLight	기능지원	기능지원
	PointLight	Lightjunior	Light
	Script	기능지원	기능지원
	Shape	기능지원	기능지원
	Sound	기능지원	기능지원
	SpotLight	Light Intensity	Light_LT_Steady
	WorldInfo	HUD	HUD
기하노드	Box	Brush Box	StaticMesh
	Cone	Brush Cone	StaticMesh
	Cylinder	Brush Cylinder	StaticMesh
	Sphere	Brush Sphere	StaticMesh
속성노드	Color	기능지원	StaticMesh
	Coordinate	기능지원	Surface
	Normal	기능지원	기능지원
	TextureCoordinate	기능지원	Surface
센서노드	TouchSensor	기능지원*	기능지원*
	TimeSensor	기능지원*	기능지원*
외형노드	ImageTexture	기능지원*	기능지원*
	Material	Texture	기능지원*
	PixelTexture	기능지원*	기능지원*
부가노드	Background	기능지원*	Skyzone
	Fog	기능지원*	ZoneInfo_FogColor
	NavigationInfo	HUD	HUD

<표 5> 게임엔진 기능 비교

기 능	Quake Engine	Unreal Engine
Light & Shading	O	O
Dynamic Lighting	O	O
Mip-Mapping	×	O
Culling System	BSP	BSP, PVS[13]
L O D	×	O
Character Animation	O	O
Script	Quake-C	Unreal Script, C++
API	OpenGL	OpenGL, DirectX
Level Editor	Gtk Radiant	UnrealEd

를 모두 지원하는 그래픽 카드에 대해 화면 출력 프레임의 초단위 측정이 가능하다. FPS 측정 방법은 건축가 르꼬르뷔제의 빌라사보아를 가상공간 내에 구축하고 이를 가상 체험하는 과정에서 발생하는 직선주행, 좌측주행, 우측주행, 상하 시점전환 네 가지 유형의 네비게이션의 시작지점의 FPS와 종착지점의 FPS를 측정하였으며, 측정결과는 <표 7>과 같다.

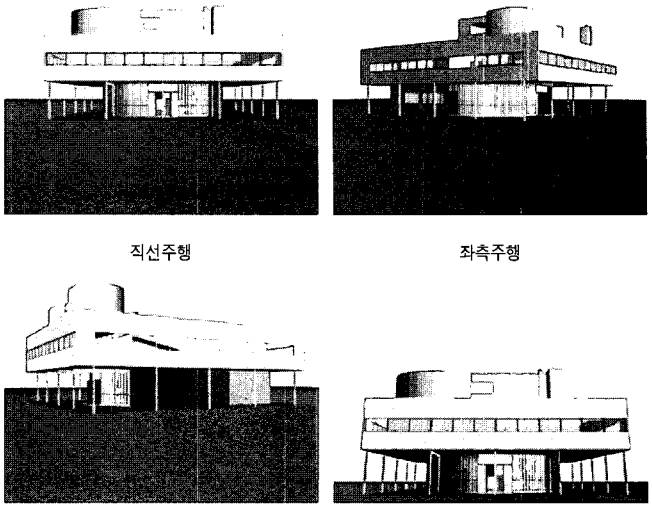
FPS 측정결과 직선주행의 경우 게임엔진의 측정값이 VRML에 비해 시작지점에서는 평균 5배 이상, 종착지점에서는 평균 2~3배이상 높게 나타났다. 좌측주행은 시작지점의 VRML 측정값이 직선주행에 비해 낮게 나타났으며 게임엔진의 측정값과의 차이는 6배 이상 차이를 보였다[15]. 또한 종착지점의 평균값 역시 게임엔진의 측정값이 VRML의 수치에 비해 5배 이상 높게 나타났다. 우측주행은 VRML의 측정값이 좌측주행의 측

15)VRML클라이언트에서는 초당 30프레임 이하로 나타나는 경우(음영표시부분)가 자주 나타났다.

정값보다 평균적으로 높은 수치를 나타냈으며 게임엔진의 측정값은 우측주행이나 좌측주행의 측정값이 비슷한 수치를 보였다. 상하시점전환의 경우 VRML의 경우 상하시점전환에 따른 측정값의 변화가 비슷하였으며 게임엔진의 경우 VRML과는 달리 시점전환에 따른 수치도 변화하여 장면전환에 따른 화면 출력을 빠르게 연산하는 것을 보여주었다. 이상의 네 가지 네비게이션 유형에 따른 결과를 정리하면 게임엔진기반 가상현실이 큰 용량의 데이터 처리나 화면에 나타나는 장면의 폴리곤 처리에 있어서 VRML에 비해 월등하게 우수함을 알 수 있다.

<표 6> 게임엔진 호환성 비교 분석

분류		Quake 3 Engine	Unreal 2004 Engine
제작성	객체생성	• 3D 모델 데이터의 직접 활용 가능 • Gtk Radiant을 이용한 모델 데이터의 활용	• 3D 모델 데이터의 ASE 파일 포맷 변환을 통해 UnrealEd에서 데이터 활용
	텍스처링	• Diffuse, Specular, Self-Illumination, Bump, Reflection/Refraction Mapping표현 가능	• Diffuse, Specular, Self-Illumination, Bump, Reflection Mapping 표현 가능
	라이팅	• 3D 응용 어플리케이션의 라이팅 효과를 수용하지 못하여 라이트 Map을 추출한 Texture Mapping의 방법을 이용하여 광원 효과 표현	• 퀘이크와 동일
호환성	범용성	• 1996년 최초 개발이후 지속적 업그레이드 • C언어로 프로그래밍되어 있어 거의 모든 운영체제에 이식 가능 • 윈도우, 콘솔 게임의 PlayStation2나 X-Box와 같은 플랫폼 지원 가능 • 윈도우 모든 OS 버전과 Linux, Mac에서의 지원과 Playstation2, X-Box와 같은 콘솔 기기에서 운용가능	• 퀘이크 엔진을 벤치마크하여 개발하여 퀘이크의 기본 장점을 모두 갖춤 • 최신 버전의 윈도우나 콘솔 기기 대부분에서 작동하도록 설계 • 최근 출시로 인해 PC사양에 있어 퀘이크 엔진보다 고성능의 재원을 필요로 함 • 플랫폼의 범용성에 있어 퀘이크 엔진에 비해 떨어짐 • 윈도우 95를 제외한 모든 OS 버전과 Linux를 지원하며 Playstation2, X-Box와 같은 콘솔 기기에서 운용가능
	확장성	• 퀘이크 엔진의 소스 코드 공개 • 출시된지 5년이상 지난 게임엔진으로 3D 응용 프로그램의 데이터 호환이 쉽지 않음. • 버전별 다른 엔진을 사용하고 있어 버전별 Map에 대한 확장성 취약 • 게임엔진의 소스를 공개함으로서 다양한 게임엔진의 개발을 유도	• 엔진 공개 대신 패키지 내에 포함되어 있는 에디터에 대한 학습 정보가 넓고 방대한 자료를 제공 • 일반 3D 응용 어플리케이션의 파일 포맷을 수용하여 디자이너 측면에서의 확장성이 높음 • 버전에 해당하는 에디터를 통해 제작된 컨텐츠는 해당 버전에서만 작동 가능한 단점 • 엔진 소스는 공개하지 않지만 표준 파일 포맷을 지향함으로서 다양한 응용 어플리케이션을 이용한 객체 생성 및 표현 가능
전문성	스크립팅	• 윈도우 운영체제의 전환으로 퀘이크 C 프로그래밍 언어 역시 윈도우 플랫폼으로 변경 • 퀘이크 C와 Visual C++로 구성 • 디자이너가 프로그래밍 언어를 코딩하여 컨텐츠를 제작하는데 어려움 • 프로그래밍 언어와 관련된 에디터 프로그램을 이용하여 디자이너가 원하는 형태의 컨텐츠 제작에 접근 • 퀘이크 엔진의 경우 소스가 공개되어 전문 엔지니어에 의한 변형 프로그래밍 이용 가능	• 언리얼 스크립트 언어는 게임의 전반적인 분야를 컨트롤 할 수 있도록 게임엔진에 최적화 • 언리얼 스크립트는 C++를 개량하여 사용하므로 유지보수성이 높음. • 퀘이크 보다 코딩이 간편하고 사용이 편리한 스크립트 제공 • 프로그래밍 과정의 에러 발생 가능성이 낮음. • 에디터를 통해 스크립트의 편집이 가능하여 하나의 환경에서 Map 생성에 관련된 모든 편집 작업 가능 • 에디터를 이용한 스크립트의 편집 기능과 에디터 이용에 대한 정보나 온라인 학습 지원 우수
	조작성	• 객체 생성에 있어 Gmax Tempest 응용 어플리케이션과 게임 레벨 제작에 이용되는 Gtk Radiant 응용 어플리케이션 존재 • Gmax Tempest의 경우 3DS MAX와 같은 인터페이스 및 기능을 제공 • Gtk Radinat는 디자이너가 사용하기에 인터페이스나 조작성이 나쁨. • 편집 에디터의 활용은 컨텐츠나 디자인 공간에 대한 실시간 확인의 편리함과 프로그래밍 언어를 직접 조작하지 않아도 되는 장점	• 3D 모델 에디터와 데이터를 호환하는 Map 에디터를 제공하기 때문에 디자이너의 측면에서 사용 용이 • 에디터 내에서 제작된 컨텐츠에 대한 스크립트 편집이나 조작이 가능하여 컨텐츠 제작 유용성 • 에디터에 대한 버전별 정보를 온라인 서비스 • Map 에디터의 설명과 지원은 디자이너들이 컨텐츠를 제작함에 큰 역할 수행



<그림 3> FPS 측정 네비게이션 유형

<표 7> VRML과 언리얼엔진의 화면갱신률 실험결과 비교

분 류		Subject-A		Subject-B		Subject-C		Subject-D		Subject-E	
		VRML	Unreal	VRML	Unreal	VRML	Unreal	VRML	Unreal	VRML	Unreal
직선주행	Start	31~35	42	32	70	26	124	17~19	118	26~28	109
	End	30~46	143	34~60	166~191	30~51	156~184	30~57	182~187	40~48	161~187
좌측주행	Start	16	110	27	75	26	120	10~20	120	18	120
	End	30~60	161~166	46~61	165~187	38~58	173~211	30~52	175~196	49~57	118~176
우측주행	Start	32	172	19~23	95	10~19	115~120	20~29	132	44	122
	End	45~58	173~214	55~57	149~182	41~59	131~205	30~45	185~195	35~53	191~200
시점전환	Start	32	110	29	122	26	119	26	131	31	120
	End	15~36	129~175	37~43	148~254	42~51	159~187	40~41	185~186	37~42	157~189

4.4. 공간체험성에 대한 정성적 성능평가

3차 실험은 체험성 평가로 피실험자에 대한 인터뷰를 병행한 관찰조사를 실시하였다. 평가는 2차 실험과 동일한 가상공간을 VRML과 언리얼엔진 뷰어로 피실험자에게 체험하도록 하였으며 체험과정 관찰과 인터뷰를 통한 분석을 실시하였다. 평가 항목은 상호작용성, 몰입감, 실존감으로 3개의 카테고리를 9개의 항목으로 나누어 수행하였다.

응답자의 성향은 <표 8>과 같으며 두 개의 뷰어를 비교한 후 상호비교우위에 있어서 주관적으로 우수한 뷰어를 지명하도록 하여 결과는 <표 9>와 같게 나타났다.

<표 8> 응답자 성향

분류	전공	빌라사보아에 대한 경험	가상현실에 대한 지식	FPS게임에 대한 경험
Subject-A	실내디자인	없음	기초적	없음
Subject-B	건축설계	있음	기초적	없음
Subject-C	건축설계	없음	없음	많음
Subject-D	건축설계	있음	없음	많음
Subject-E	건축설계	없음	기초적	많음

<표 9> 시뮬레이션 평가 실험 결과

요소	Subject	A		B		C		D		E	
		VRML	Unreal	VRML	Unreal	VRML	Unreal	VRML	Unreal	VRML	G
실시간 상호작용	네비게이션		O		O		O		O		O
	장면전환		O		O	O		O		O	
몰입감	조작성		O		O		O		O		O
	관찰시점		O		O	O			O		O
	주변환경차단		O		O		O	O			O
실존감	색상	O		O			O	O		O	
	밝기		O		O		O	O		O	
	사실감	O			O		O		O	O	
	물리적 속성		O	O			O		O	-	-

네비게이션과 조작성에 있어 게임엔진을 이용한 시뮬레이션이 우수하다고 평가할 수 있다. VRML의 네비게이션과 조작은 거의 마우스만으로 진행하여야 하지만 게임엔진은 시점의 전환이나 상하좌우의 조작을 통해 시점 전환이 자유롭기 때문인 것으로 나타났다.

장면전환의 경우 공간 시뮬레이션 목적 아래 제작된 콘텐츠는 게임에서와 같이 빠른 장면전환을 요구하는 것이 아니라 정확한 반응을 유도하는 것이 중요하다는 관점에서 VRML의 장면 전환을 선호하는 반응을 보였다. 그러나 대규모 공간이나 데이터 처리에 있어서는 응답자 대부분이 게임엔진이 우위에 있다고 평가하고 있었다. 색상은 VRML이 우수하다는 응답이 많았으며 밝기는 게임엔진이 많은 것으로 나타났다. 색상은 VRML이 면의 구분이 뚜렷하게 나타난 것에 반해 게임엔진은 전체적으로 면의 구분이 뚜렷하지 않아 나타난 현상으로 파악되며, 별도의 측정을 통해 모델러와 뷰어의 색상오차도 게임엔진이 큰 것으로 파악되었다. 사실감의 경우 실험자의 시점에 따라 상하좌우의 조작과 네비게이션 과정의 항목 등이 이를 뒷받침하며 물리적 속성 또한 이러한 항목과 연관되어 게임엔진이 상대적으로 우수하다는 것을 의미한다.

5. 결론

VR엔진에 기반한 전문 가상현실 장비는 고비용과 공학적 전문성의 요구 등의 이유로 디자인분야에 적용되기 힘들며 웹기반 데스크탑 클라이언트는 공간 환경구성에 대한 지원이 미약하고 대용량의 모델 데이터를 수용할 수 없기 때문에 공간시뮬레이션에 적용하기에 미흡한 점이 많았다. 이에 본 연구에서는 FPS게임엔진 기반의 레벨에디터를 응용한 공간가상현실 시뮬레이션을 대안으로 제시하였다. 이상으로 게임엔진에 대한 고찰과 공신력 있는 2개의 레벨에디터를 분석하고 기존의 웹기반클라이언트의 성능과 비교한 결과 다음과 같은 이유로 게임엔진기반의 가상현실이 향후 공간시뮬레이션분야에 적합하다는 결론을 얻게 되었다.

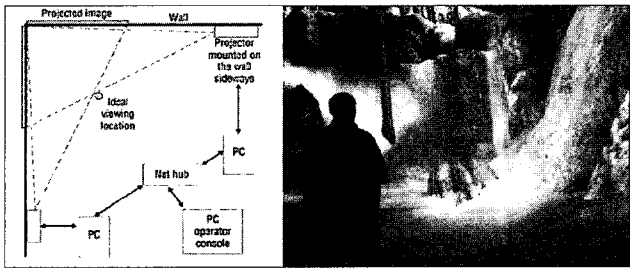
첫 번째, 저가, 저사양의 저작도구와 플랫폼에서 구현이 가능하다. 다만 게임엔진의 BPS, PVS가상공간구조는 모델링 툴에서 다루는 3차원 모델과 차이가 있으므로 이에 대한 고려가 필요하다.

두 번째, 공학적 지식이 상대적으로 적은 디자이너들이 직접 제어할 수 있는 그래픽기반의 저작환경을 제공한다.

세 번째, 범용가상현실 저작도구와 달리 FPS레벨에디터는 다양한 환경효과와 공간구축기능들을 제공한다.

네 번째, DirectX는 Input/Output모듈을 가지고 있으므로 향후 전문가상현실 입출력디바이스에 연동시킬 수 있는 가능성을 가지고 있다. <그림 4>는 언리얼 엔진의 디스플레이를 개조하여 CAVE를 구축한 사례이다.[16]

연구과정에서 게임엔진에서는 모델링 툴과의 데이터변환 과정에서 색상의 오차가 발생하는 문제와 저작도구로 모델데이터 변환절차의 복잡성 등 디자인분야에 충분히 실용화되기 까지는 해결해야할 몇 가지 문제가 제기되었으며, 지속적인 연구를 통해 머지않은 미래에 개선될 것이다.



<그림 4> CaveUT

참고문헌

1. Jijun Wang, Michael Lewis, Jeffry Gennari, A game engine based simmulation of the nist urban search and rescue Arenas, Proceedings of the 2003 winter simulation conference
2. Bendik Stang, Game engines, features and possibilities, IMM DTU, 2003
3. S.T.Kim, H.J.Jun, A study on correcting visual changes of models that occur in the process of designing backgrounds of FPS games, Proceedings of ASPD2006
4. S.Y.Yoon, Uddin, M.S.(2003). House X, Scheme G: From 3D game engine to virtual representation of architecture. Journal of the Design Communication Association Representation
5. 김종현, GPU기반 게임에디터를 활용한 공간 가상현실 시뮬레이션의 표현에 관한 연구, 인제대학교 대학원, 2004
6. 구 슬, FPS 게임의 변화와 발전, KDGC. 2002
7. 김정훈, 3D 게임엔진 기술 개방 동향, 한국과학기술정보연구원, 2003
8. 김형석 · 원광연, 가상현실 기술과 게임_현황과 미래, 게임산업개발원 연구 보고서, 한국과학기술원 전산학과, 2002
9. 박경환 · 황성진, 게임 개발 도구, 한국멀티미디어 학회지 제5권 제2호. 2001
10. 박은순 · 신유진, 가상현실에 구현된 공간의 현실감 향상을 위한 연구, 대한건축학회학술발표대회논문집 제22권 제2호, 2002
11. 서해림, 디지털 매체 몰입 경험에 관한 연구, 이화여자대학교 대학원, 2002
12. 이한민, 도시경관 실시간 시뮬레이터 개발, 서울산업대학교 산업대학원, 2001

<접수 : 2006. 8. 29>

16)Jeffery Jacobson, Michael Lewis, Game Engine Virtual Reality with CaveUT, University of Pittsburgh, IEEE Computer society, 2005.4