

3D 공개 게임 엔진 비교분석

허원^o 황요한 김동균 신동규 신동일
세종대학교 대학원 컴퓨터공학과 눈산멀티미디어 연구실
{heowon, xfilen, kd999, shindk, ds'in}@gce.sejong.ac.kr

Comparative Analysis of 3D Open Source Game Engine

Won Heo^o Yo-han Hwang Dong-kyun Kim Dong-kyoo Shin Dong-il Shin
Dept. of Computer Engineering, Sejong University

요 약

본 논문에서는 3D 공개 게임 엔진에 대해서 간단한 소개와 함께 게임 엔진으로서 중요한 요소를 기준으로 비교분석 하였다. 개인의 개발연구용과 학교 및 연구소에서 비영리적으로 이용이 가능한 3D 게임 엔진에 대해서 간단하게 소개하며 각각의 특성을 비교 분석함으로써 더 많은 개발자들의 게임 엔진 연구에 효과적인 역할을 하였으면 한다.

1. 서 론

현재는 2D에서 3D로의 게임 콘텐츠가 전환되는 시점에 놓여 있다. 이미 패키지 게임 산업은 화면의 구성 및 게임 콘텐츠의 구성을 3D로 가져간지 오래되었으며 온라인 게임과 네트워크 게임들도 차차 그 입지를 3D로 옮겨가고 있는 실정이다. 지금의 3D에서 게임 콘텐츠의 표현방식은 2D의 방식과는 달리 좀 더 복잡하고 정교하며 현실에 가까운 그래픽과 사운드를 접목하여 인터랙티브한 게임 환경을 제공하고 있다. 이러한 게임 환경을 제공하는 기술적인 중심어는 3D 게임 엔진이라는 핵심 기술이 존재하며 수많은 게임 개발 회사와 개발자 그룹에서는 좀 더 향상된 3D 게임 엔진의 연구와 개발에 대한 투자를 아끼지 않고 있다. 그 성과물로 다양한 종류의 3D 게임 엔진이 등장하였으나 엔진을 이용하여 3D 게임 콘텐츠를 개발하기 위해서는 높은 비용을 지불해야만 한다.

이에 본 논문에서는 개인의 개발연구용과 학교 및 연구소에서 비영리적으로 이용이 가능한 3D 게임 엔진에 대해서 간단하게 소개하며 각각의 특성을 비교 분석하였다. 2장에서는 3D 게임 렌더링 엔진에 대한 간단한 내용을 기술하고 3장,4장에서는 공개 엔진들의 비교분석을 논하고 5장에서는 결론 및 향후 전망에 대해 기술한다.

2. 관련연구

3D 게임 렌더링 엔진의 요소에는 다음과 같다.

2.1 렌더링 요소

2.1.1 광원효과(Lighting & Shading)

Lighting은 3D 렌더러에서 빛의 효과를 반영하는 과정이다. 실시간 3D에서 자연과 같은 빛의 처리는 무리이기 때문에 단순화된 모의 광원의 효과를 가정해 계산한다. Shading은 빛으로 인한 명암을 폴리곤에 반영하는 과정이다. 실시간 3D에서는 모든 점에 대해 lighting을 적용하는 것은 어렵기 때문에 면단위로 적용한 shading 테크닉이 주요하게 사용된다[1][2][3].

2.1.2 텍스처링(Textureing)

텍스처는 폴리곤 위에 입혀지는 2D 이미지라고 말한다. 텍스처 자체의 이미지는 단순한 2D 평면이지만, 이것이 3D Mesh에 알맞게 포장되면 빛의 효과로는 표현하기 어려운 세부적인 사실감이 표현 가능하다. 멀티 텍스처링은 동일한 Mesh에 대해 여러 개의 텍스처를 동시에 적용할 수 있는 기능이다. 각각의 텍스처는 스테이지별로 분리되어 순서대로 사용되며, 이때 스테이지간의 다양한 blending 효과를 지원함으로써 라이트 매핑, 환경매핑 그리고 그림자 같은 상위 레벨의 효과를 빠르게 처리하는 것이 가능하다[1][2][3].

2.2 Culling System

2.2.1 지형처리 방법

3D게임의 특징은 가상공간으로의 시점 및 카메라의 이동 제약이 적다는 것이다. 커다란 가상공간에 대한 효

울적인 공간처리 방법이 필요하다. BSP(Binary Space Partitioning)는 이분적인 공간분할 방법으로써 실시간으로 동시에 처리하기 불가능한 지형 데이터를 이진트리 형태로 분석해 노드 정보를 구성하고 이를 참조해 대상 물체를 탐색해간다. PVS(Potentially Visible Set)는 보여질 가능성이 있는 리스트를 기억해 두는 방식으로 BSP 같은 공간분할방식과 포털(Portal)이라는 테크닉을 같이 사용하여 구현한다. 포털은 실내 환경에서 주로 적용되는 것으로, 벽으로 나뉜 방과 같은 구조에서 연결되는 문이나 창과 같은 부분을 말한다[4][5].

2.2.2 LOD

LOD(Level Of Detail)란 게임 사용자의 시점에서 먼 거리에 있는 물체는 적은 수의 폴리곤으로 표현하고 가까이 있는 물체는 폴리곤의 수를 증가시켜 표현하는 기술이다. 이전에는 몇 단계의 모델을 거리별로 직접 제작해 사용했다. 하지만 요즘 새로 개발되는 게임은 실시간으로 폴리곤의 개수를 조절할 수 있는 기능을 가지고 있다[4][5].

2.3 Dynamic Effect

2.3.1 그림자

그림자는 최근 들어 활발하게 적용되기 시작했는데, 이는 자연스러운 그림자 처리를 위한 수학 연산과 이에 대한 CPU 사이클링에 대한 비용이 많이 걸리기 때문이다. 적절한 그림자 효과는 장면의 볼륨감 및 원근감의 표현을 보다 사실적으로 나타내어주어 게임 환경을 보다 현실감있게 구성할 수 있는 장점이 있다[1][2][3].

2.3.2 라이트 매핑

기존의 라이트 방식은 작은 정점으로 이루어진 Mesh에 적용할 때 정확성이 떨어지고 만족스러운 결과물을 얻을 수 없는 단점이 있다. 라이트 매핑의 기본 원리는 기본 텍스처 위에 밝기를 나타내는 또 하나의 텍스처(라이트 맵)를 준비해 이를 멀티 텍스처링하는 것과 같다. 이때 사용되는 라이트 맵은 대개 기본 텍스처보다는 아주 작지만 필터링이 이루어져 부드럽게 보인다[1][2][3].

2.3.3 환경매핑

환경매핑은 주위의 환경을 텍스처로 투여해 준비하고, 이것을 실시간으로 시점과 반사되는 각도로 보이도록 텍스처 좌표를 설정해 주위의 물체가 실제 표면에서 반사되는 것처럼 보이는 효과를 내는 테크닉이다[6].

3. 3D 공개 게임 엔진 비교분석

3.1 공개 게임 엔진에 대한 소개

본 논문에서는 3D 공개 게임 엔진으로 퀘이크2, Genesis 3D, Crystal Space, Fly 3D를 소개 및 비교분석을 하겠다.

3.1.1 퀘이크2

ID 소프트웨어에서 만든 퀘이크2[7] 엔진의 특징은 높은 자원의 rednerer에 있다. 공간분할은 BSP를 사용하였고, 애니메이션에 있어서는 vertex 애니메이션 기법을 사용하였다. 최초의 라디오시티 라이트 맵을 사용한 엔진답게 라이트 맵의 활용이 매우 뛰어나다. 실시간 빛 효과를 내기 위해 컬러 라이트 맵을 사용해 빛의 움직임을 표현한것과, 환경 매핑을 실제 게임에 사용하는 등 뛰어난 화면을 위한 많은 기술적 시도가 있었다. 놀라운 사실은 Geforce 2[8]부터 지원하기 시작했던 vertex shader가 퀘이크 엔진에서는 이미 지원되고 있다는 것이다. 그러나 다른 상용엔진에 비해 사용하기가 어렵다는 단점이 있다. 이를 증명하듯 거의 비슷한 평가를 받는 언리얼 엔진에 비해 퀘이크 엔진을 사용한 게임은 소수에 불과하다.

3.1.2 Genesis 3D

Eclipse Entertainment에서 개발된 Genesis3D[9]는 공개엔진이라 하기 무색할 정도로 일반 상용 게임 엔진과 같은 비슷한 기능들을 제공한다. 라이트맵을 이용한 원근 텍스처 매핑, 움직이는 물체에 대한 텍스처 처리, 반투명한 텍스처 등 사실적인 표현을 위한 텍스처 기법들이 사용되었다. 동적인 그림자 효과와 빛의 회절, 다양한 색의 빛을 표현할 수 있다. 실시간으로 물에 대한 표면의 뒤투림을 표현해 줄 수 있다. 그리고 월드 객체에 대한 강체(rigid body)물리학 시뮬레이션을 제공한다. LOD기법이 사용되고, 지형처리방법으로는 포털이 사용된다.

3.1.3 Crystal Space

Crystal Space[10]는 진정한 의미의 6DOF(degree's of freedom)[11]를 구현하여 사용자가 머리를 앞으로, 뒤로, 왼쪽으로, 오른쪽으로, 위로, 아래로 움직일 수 있다. 텍스처는 크기가 두배가 되어도 상관이 없고 꼭 사각형일 필요가 없다. 또한 여러 종류의 그림파일들을 텍스처에 적용시킬 수 있다. 회전시키거나 크기와 좌우를 바꿔서 폴리곤에 텍스처를 입힐 수 있으며 수면이나 유리창과 같이 투과되거나 반 투과되는 텍스처를 만들 수 있다. 물체 반사 효과를 지원하고 빛과 그림자가 미리

계산 된 후 표현된다. bezier와 같은 곡면을 만들 수 있다. 지형처리 방법으로는 포털과 Octree, BSP트리와 c-버퍼를 함께 사용한다.

3.1.4 fly3d

fly3d[12]는 한 유니트에 4개까지의 텍스처를 매핑 시킬 수 있고, 세부적인 텍스처를 표현할 수 있다. 동적인 빛과 라이트 맵을 이용한 그림자를 표현해 줄 수 있다. 파티클 시스템이 구현되어 있고, 기본적인 물리현상이 구현이 되 있어서 간단한 물리 시뮬레이션이 가능하다. 충돌검사기능이 구현되어있다. 메시들에게 움직임을 줄 수 있고, 귀엽고 따뜻한 느낌의 3D효과를 주는 카툰렌더링이 사용되었다. 텍스처로 사용될 수 있는 파일은 TGA와 JPG 두 개의 파일만 가능하다. 엔진이 완전하게 플러그인 형식으로 제어가 된다. 즉 게임을 만들거나 게임에서의 객체를 만들 때에는 플러그인을 만드는 형식으로 만들면 된다. 지형처리 방법으로 BSP와 PVS를 동시에 사용하였다.

3.2 3D 공개 게임 엔진 비교 분석

본 논문에서 소개한 3D 공개 게임 엔진들을 [표1]요소들을 토대로 비교분석 하였다.

	Quake2	Genesis 3D	Crystal Space	Fly3d
culling system	BSP	BSP	BSP, Octree, 포털 c-buffer	BSP,PVS
LOD system		yes	yes	
픽셀 렌더링	yes	yes	yes	
Lighting & Shading	yes	yes	yes	yes
Dynamic Lighting		yes	yes	yes
파티클 효과	yes	yes		yes
shader				
물리엔진				yes
스크립트	C++		C++	C++

[표1] 3D 공개 게임 엔진 비교분석표

4. 결론 및 향후 전망

본 논문에서는 3D 공개 게임 엔진에 대해서 간단한

소개와 함께 게임 엔진으로서 중요한 요소를 기준으로 비교분석 하였다. 게임 엔진을 제작하거나 연구하는 학교 및 기관 연구소 혹은 개인들에게 잘 알려지지 않은 공개 엔진에 대한 내용을 언급함으로써 좀 더 많은 개발자들의 게임 엔진 연구에 많은 도움이 되었으면 한다. 3D 게임 엔진을 이용한 다양한 방면으로의 접근이 시도 될 것이라고 예상이 된다. 특히 캐릭터에 대한 사실적인 표현에 바탕을 둔 애니메이션 엔진과 디지털 방송매체에 큰 영향을 줄 것이고 실시간 지형 렌더링을 통한 지형 좌표 시스템에도 영향을 줄 것이 예상된다.

이후에는 공개 게임 엔진과 상용화 엔진에 대한 비교분석대상을 좀더 세분화하여 자료를 정리할 예정이다.

5. 참고문헌

[1]Real-Time Rendering, Tomas Moller, Eric Haines, A K Peters.ltd, 1999
 [2]3D Computer Graphics, Alan watt, Addison wesley, 1999
 [3]Black Art of 3d Game Programming, Andre Lamothe, Waite Group, 1995
 [4]Game Programming Gems, Mark Deloura, Charles River Media
 [5]<http://spica.u-aizu.ac.jp/soe/magazine/soe9.html>
 [6]<http://home.san.rr.com/thereindels/Mapping/Mapping.html>
 [7]Quake2, 2001 id software, 2001, <http://www.quake.com>
 [8]Geforce 2, 2001 NVIDIA Corporation, 2001, <http://www.nvidia.com>
 [9]Genesis3D, Eclipse Entertainment, 1998, <http://www.genesis3d.com>
 [10]Crystal Space, <http://crystal.sourceforge.net>
 [11]Robot Dynamics And Control, Mark W.Spong, M. Vidyasagar, Wie Wiley, 1989
 [12]fly3d, <http://www.fly3d.com.br>