

렌더링 엔진, 각종 게임 그래픽 데이터를 다양한 환경에서 출력

실시간 3차원 그래픽의 수준으로 발달...저장된 영상물에 필적할 정도 높은 수준

디지털콘텐츠 여러 분야 가운데 '게임 분야'는 이미 상당한 수준에 이르렀다는 평가를 받고 있다. 하지만 현재에 만족하지 않고 발전하기 위해서는 업계의 정보공유와 보다 발전된 테크놀로지가 뒷받침돼야 할 것이다.

2003년 4월호부터 8회 동안 한국전자통신연구원(ETRI) 가상현실연구부의 연구원들이 3D 게임 엔진에 대해 상세하게 소개하는 지면을 기획했다. 국내 온라인게임 개발자들에게 도움이 되었으면 한다.

「편집자 주」

연 재 순 서

1. 3D 게임엔진에 대하여
2. 렌더링 엔진 (이번호)
3. 애니메이션 엔진
4. 사운드 엔진
5. 서버 엔진
6. 게임 인공지능
7. 맵 에디터
8. 게임진행 모듈

한국전자통신연구원(ETRI)
김현빈 외 가상현실연구부

초창기의 3차원 게임들은 선으로 표시되던 것이 대부분이었지만, 최근 몇 년 사이에 비약적인 발전이 이뤄져 실시간 3차원 기술로 이제 게임은 실사와 같은 실재감을 구현할 수 있을 정도로 발전하고 있다. 또한 요즘은 거의 대부분의 컴퓨터에 예전 같으면 상상하기 힘들 정도의 다양한 하드웨어 차원의 3차원 기능을 지원하는 카드들이 장착돼 판매되고 있다. 여기에 부응해 3차원 게임엔진은 점점 화려한 기술을 연출할 수 있는 방향으로 만들어지고 있다. 게임엔진에서 중요한 역할을 하는 부분들이 많이 있지만, 역시 화면에 그림을 출력해 기능을 제공하는 '렌더링 부분'은 게임엔진의 기본이 되는 중요한 부분이라 할 수 있다. 이에 본 고에서는 렌더링 엔진이 무엇이며, 어떤 기능들을 제공해야 하는지에 대해 알아보기로 한다.

렌더링 엔진이란?

렌더링 엔진에 대한 설명에 앞서 렌더링 엔진이란 무엇인가에 대해 설명하고자 한다.

PC 기반의 3차원 게임을 제작하기 위해서는 보통 DirectX나 OpenGL과 같은 3D 그래픽 라이브러리(Graphic Library)를 이용해 그래픽 가속 하드웨어의 기능을 활용하는 것이 일반적이다. 이 때, 이런 그래픽 라이브러리들은 입력된 3차원 다각형(polygon)을 일괄적으로 처리해 주어진 카메라 위치나 방향에 따라 2차원 화면에서 이들이 나타나는 위치와 서로 가리고 가려지는 관계와 광원과의 위치 관계 등을 계산해 화면을 구성하고, 이들 다각형들의 표면 속성과 이들에 입혀진 텍스처(texture)에 따라 다각형 내부의 색상을 채워주게 된다. 다시 말해 범용의 그래픽 라이브러리만을 사용하는 경우에도 3차원 게임 제작에 요구되는 기본적인 3차원 객체 출력 기능을 수행할 수 있으며, 이를 통한 게임 구현이 가능하다.

하지만 이런 그래픽 라이브러리 그 자체를 렌더링 엔진이라고 하지는 않는다. 일반적으로 게임 제작 과정에서 요구되는 각종 '렌더링 기능'은 각종 그래픽 라이브러리가 제공하는 단순한 기능 이외에 보다 많은 다양한 기능을 요구하며 경우에 따라서는 화면 출력과 관련된 기능 이외

에 게임 진행과 관련한 게임 환경의 변형이나 게임 로직, 게임 객체의 환경 내에서의 움직임 등에 대한 제어를 담당하는 경우도 있다. 여기서는 렌더링 엔진을 다음과 같이 정의하고 이에 대한 설명을 진행하고자 한다.

‘렌더링 엔진’이란 그래픽 디자이너에 의해 작성된 각종 게임 그래픽 데이터를 게임 진행 상황에 따라 게임 환경 내에서 적절히 변형, 이동시킨 후 빠르게 화면에 출력해주는 엔진을 말한다.

렌더링 엔진이 제공해야 하는 기능

3차원 게임의 제작을 위해서는 렌더링 엔진은 다음과 같은 기능을 제공해야 한다.

〉 3차원 객체화면 출력 기능

렌더링 엔진은 기본적으로 그래픽 디자이너에 의해 작성된 각종 게임 객체를 화면에 출력할 수 있어야 한다. 객체의 형상, 표면 속성, 그리고 객체에 입혀진 각종 텍스처가 렌더링 엔진에서 올바르게 처리돼 최종 화면에 적절하게 출력되어야 하며, DirectX나 OpenGL, 최신 그래픽 하드웨어 가속장치의 기능을 최대한 살려낼 수 있도록 구성돼야 한다. <그림 1>은 렌더링



엔진에 의해 3차원 객체가 출력되는 모습을 보여주고 있다.

〉 실시간 객체 출력을 위한 각종 게임 객체 제어 기능

DirectX나 OpenGL 등의 그래픽 라이브러리를 활용할 경우 다각형 메쉬(mesh)의 형태로 구성된 각종 게임 객체들은 그래픽 하드웨어 가속장치 내에서 2차원 영상의 형태로 렌더링돼 화면에 출력되게 된다. 이 때 아무리 최신의 그래픽 하드웨어 장치를 활용하는 경우에도 처리해야 하는 다각형의 수가 많아지면 영상 생성에 소요되는 시간이 많아지게 돼 렌더링 속도가 줄어들게 된다.

이런 현상을 방지하기 위해 렌더링 엔진은 전체 게임 환경 중 카메라 시야에 나타나지 않는 게임 객체나 화면에 나타나더라도 아주 적은 수의 픽셀(pixel)만 차지해 그 화면상 중요도가 낮은 게임 객체의 경우 미리 렌더링 과정에서 제거하거나 보다 적은 수의 다각형으로 표현될 수 있도록 해 전체적으로 렌더링 속도를 향상시키는 기능을 가지고 있어야 한다.

〉 객체 애니메이션 지원 기능

3차원 게임의 제작을 위해서는 주어진 3차원 객체의 화면 출력뿐만 아니라 각 객체의 움직임을 올바르게 표현하고 출력하는 기능이 중요하다. 이를 위해 렌더링 엔진은 애니메이션 엔진과 함께 통합돼 게임 콘텐츠로 구성되며, 경우에 따라서는 렌더링 엔진 내부에 객체 동작 제어 기능이 포함되기도 한다. 렌더링 엔진은 메쉬 스킨링(mesh skinning) 애니메이션, 계층구조(hierarchical) 애니메이션, 정점(vertex) 애니메이션, 텍스처 애니메이션 등 다양한 애니메이션 결과에 따라 내부의 게임 객체의 형상이나 위치를 변형시킬 수 있어야 하며, 경우에 따라서는 기본적인 애니메이션 제어 기능을 자체적으로 제공해야 한다.

〉 게임 환경 제어 및 처리 기능

3차원 게임을 진행하다보면 각종 3차원 객체의 전체 지형 내에서의 위치, 지형과 객체 간 충돌 여부 등을 빠르게 파악할 수 있어야 한다. 또한 그런 과정에도 주어진 카메라 위치에 따라 지형을 빠르게 화면에 출력할 수 있어야 한다. 이런 지형에 대한 연산의 종류,

실시간 출력에 대한 요구 때문에 지형의 표현을 위한 많은 자료구조와 알고리즘이 제안돼 있다.

이런 지형 표현 방식과 알고리즘들은 크게 실내 지형을 위한 방식과 실외 지형을 위한 방식으로 구분할 수 있다. 3차원 렌더링 엔진은 게임 환경 제어 및 처리 부분에 있어 실내 지형과 실외 지형에 대한 제어 및 처리 기능을 제공해야 한다. <그림 2>는 실내 지형과 실외 지형의 처리 모습을 보여주고 있다.

> 카메라 이동, 회전, 줌인, 줌아웃 제어 기능

3차원 게임은 카메라의 시점이 2차원 게임에 비해 매우 자유롭다는 특징이 있다. 그렇기 때문에 렌더링 엔진은 사용자의 카메라 조작에 따라 카메라를 이동 및 회전시키고, 초점거리를 변화시켜 자유롭게 시야를 확대 및 축소하는 기능을 제공해야 한다. 또한 각 게임 캐릭터의 움직임과 카메라를 연동시켜서 캐릭터 1인칭 시점, 3인칭 시점, 쿼터 뷰(quarter view) 시점 제어 등의 기능을 제공해야 한다.

> 광원 및 그림자 제어 기능

DirectX나 OpenGL과 같은 그래픽 라이브러리의 경우 점광원(point light)이나 방향광원(directional light), 주변광원(ambient light) 및 스포트라이트(spot light)와 같은 기본적인 광원 제어 기능을 제공한다. 이들 광원을 사용하는 경우 그래픽 하드웨어 가속장치에 의해 처리돼 화면에 표현되지만 광원의 수가 늘어날수록 속도가 느려진다는 단점이 있다.

따라서 렌더링 엔진에서는 이런 그래픽 라이브러리에서 제공하는 광원 제어 기능 이외에도 메쉬의 정점 컬러(vertex color)를 이용한 광원, 그리고 텍스처를 이용한 광원 맵(light map) 기능을 제공해야 한다. <그림 3>의 왼쪽은 광원 맵 출력 기능을 보여주고 있다. 또한 자연스러운 객체의 출력을 위해서는 자연스러운 객체 그림자 처리 기능이 필수적이다. 렌더링

엔진은 이를 위해 그림자 맵(shadow map) 기능을 비롯한 다양한 그림자 생성 및 출력 기능을 제공해야 한다.

> 특수효과 출력 기능

게임을 진행하다보면 눈이나 비, 안개와 같은 각종 대기효과나 캐릭터의 마법, 순간 이동, 공격 및 이에 따른 각종 데미지(damage) 효과의 표현, 폭발, 구름이나 물, 연기와 같은 형체가 없는 물체의 표현을 위해 여러 가지 특수효과 출력이 필요하게 된다. 렌더링 엔진은 이와 같은 다양한 특수효과를 출력할 수 있는 기능을 제공해야 한다. [그림 3]의 오른쪽은 특수효과 기능의 예를 보여주고 있다.

> 텍스처 기반 특수 기능

최근 그래픽 하드웨어 가속장치의 발전에 따라 과거에는 활용이 불가능했던 텍스처 기반의 특수 기능을 게임 내에서 활용할 수 있게 되었다. 대표적인 것이 거울면 표현을 위해 사용되는 환경 맵 기법과 3차원 객체 표면의 울퉁불퉁한 정도를 텍스처로 제어하는 범프(bump) 맵 기법이다. 렌더링 엔진은 이런 텍스처에 기반한 여러 특수 기능을 제공해야 한다.

> 각종 셰이더(shader) 활용 기능

과거에는 주어진 그래픽 하드웨어를 이용할 경우 하드웨어에서 제공하는 렌더링 기능만을 이용해 3차원 객체 출력 루틴을 구현해야만 했다. 하지만 최근 하드웨어 기술의 발달과 GPU(Graphic Processing Unit)의 등장으로 그래픽 하드웨어 가속장치 내의 내부 렌더링 연산을 개발자가 직접 작성한 셰이더를 통해 수행할 수 있게 돼 과거에는 상상할 수 없었던 새로운 렌더링 알고리즘을 활용한 게임이 점차로 등장하고 있다. 따라서 렌더링 엔진은 게임 개발자가 새로운 셰이더를 작성한 경우 이를 렌더링 엔진에 포함시켜 렌더링을 수행하도록 해 주는 기능을 제공해야 한다.

렌더링 기술의 발전과 더불어 3차원 가속기 성능의 비약적인 발전으로 인해 실시간 3차원 그래픽의 수준은 사전에 렌더링 돼 저장된 영상물에 필적할 정도의 높은 수준을 낼 수 있게 되었다. 따라서 이제는 뛰어난 3차원 게임의 제작을 위해서는 단순히 프로그래밍에 관한 지식뿐만 아니라 그래픽적인 감각도 점차 중요하게 인식돼 가고 있다. 🇰🇷

