게임 그래픽

응용 프로그래밍

35기 이찬규

목차

[1. 프로젝트 설계 및 분석 2](#_Toc535892803)

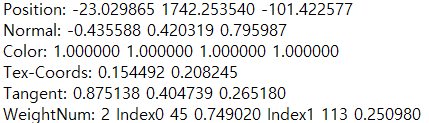
[1.1. 설계 2](#_Toc535892804)

[1.2. 분석 3](#_Toc535892805)

# 프로젝트 설계 및 분석

## 설계

캐릭터에 쓰이는 정점들은 기존에 쓰이던 정점과는 달리 현재 자신과 연결된 본들의 행렬과 블렌딩 해야 한다. 이를 위해 기존에 쓰이던 정점구조체에 자신이 속한 본의 인덱스를 저장하는 공간과 가중치를 저장하는 공간을 추가로 정의해야 한다. 또한 캐릭터의 애니메이션 정보를 가지는 클래스를 정의하고, 이 클래스에서 애니메이션과 관련된 처리를 하게한다. 기존에 쓰던 Mesh클래스를 상속하는 SkinnedMesh라는 클래스를 만들고, 해당 클래스는 추가되는 기능을 멤버로 갖는다.



[그림 1-1] 가중치가 들어간 익스포터

이제는 익스포터에선 정점의 정보를 읽어올 때 영향을 받는 본의 인덱스와 그에 해당하는 가중치를 받아와서 저장해야 한다. 가중치는 최대 4개까지 저장할 수 있는 구조로 되어있고, 이를 초과하는 캐릭터의 정점 같은 경우는 모든 가중치를 내림차순으로 정렬하고 위에서 4개를 끊어서 적용하는 방식을 택했다. 스킨파일을 불러올 때 추가적으로 요구되는 사항은 오프셋 행렬이다. 각 본들은 오프셋 행렬을 통해 자신의 본 공간으로 돌아가고 거기서 애니메이션 변환을 한 뒤 다시 부모의 공간으로 돌아간다. 오프셋 행렬은 각 노드들의 역행렬을 계산하여 저장하면 된다.

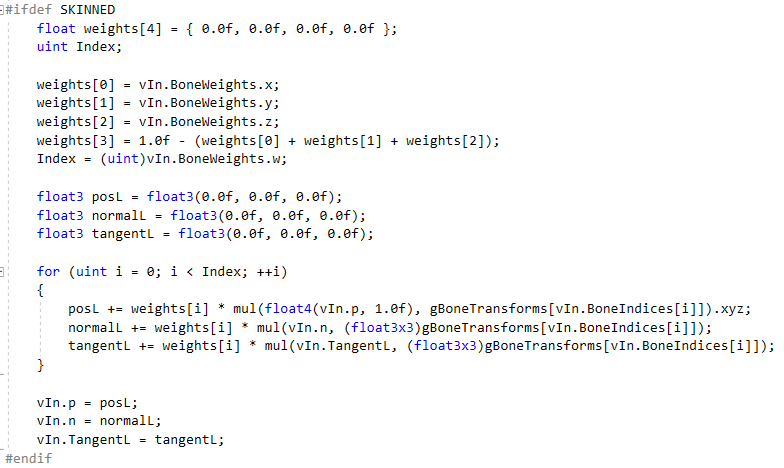
Post Effect작업은 소벨필터로 하며, ComputeShader로 처리한다. 이를 위해 장면을 백버퍼에 렌더링 하지 않고, 화면크기 만한 텍스쳐를 하나 만들고 그 텍스쳐로 만든 렌더타겟에 우선 렌더링을 한 뒤, ComputeShader 입력에 ShaderResourceView를 넘겨서 처리하는 방식으로 한다.

## 분석



[그림1–2] 바인드 포즈 상태의 캐릭터

스킨파일을 읽어오면 캐릭터는 기본 상태인 바인드 포즈 상태로 있게 된다. 애니메이션을 불러오면 그때부터 캐릭터는 불러온 애니메이션을 순차적으로 진행하게 된다. 애니메이션을 처리하는 클래스는 SkinnedData 라는 클래스로 이 클래스 안에는 각 본들의 부모 인덱스와 오프셋행렬, 그리고 여러 애니메이션을 map에 저장하는 구조로 되어있다. 스킨메쉬의 프레임이 돌때마다 해당하는 애니메이션의 현재 틱을 계산하고 이전 틱과 이후 틱을 보간해서 행렬을 계산하고 있다.

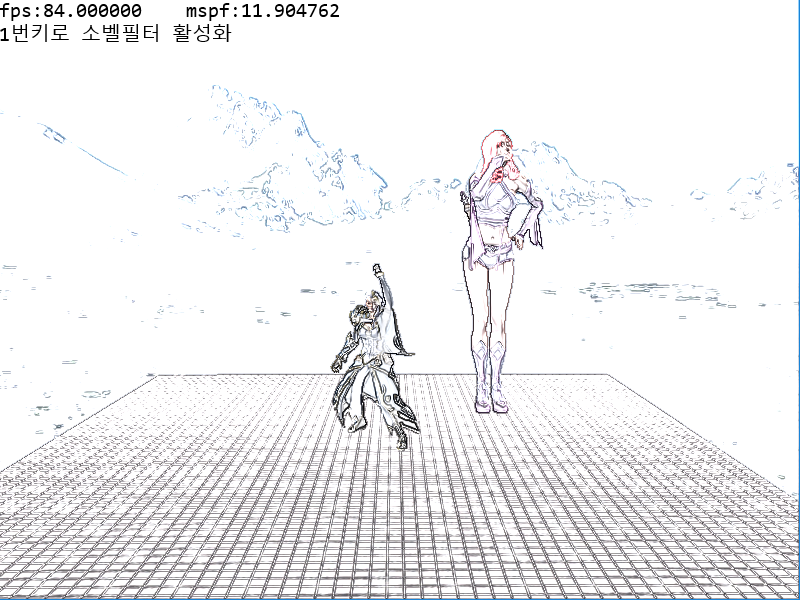


[그림 1-3] 스킨메쉬의 정점 셰이더

정점의 구조가 바뀌어서 정점 셰이더 또한 바뀌는데 굳이 새로운 hlsl을 작성하는 것보다 d3d\_shader\_macro 구조체에 디파인을 걸어서 처리하는 방식으로 진행하였다.



[그림 1-4] 애니메이션이 적용된 모습



[그림 1-5] 소벨필터가 적용된 모습

Post Effect는 소벨필터를 사용하는데, 이를 위해 DxRT클래스의 텍스쳐를 화면크기 만큼 만들고 장면을 DxRT 렌더타겟에 렌더링을 하였다. 그 후에 셰이더뷰를 SobelFilter 클래스에 넘기면 SobelFilter는 ComputeShader를 통하여 진행되는데, 입력으로 들어온 텍스쳐를 읽고 장면의 엣지를 검출한 뒤 출력을 한다. 그 후에 SobelFilter가 계산한 텍스쳐와 DxRT의 텍스쳐를 셰이더에 입력으로 넘긴뒤 두 픽셀의 색상을 더해서 출력을 하면 그림과 같이 윤곽이 아닌 곳은 거의 하얀색에 가깝게 출력이 된다.