

3D 그래픽스 셰이더 공간 변환의 시각화

이상원^o

^o청강문화산업대학교 게임콘텐츠스쿨

e-mail: one@ck.ac.kr^o

Visualization of 3D Graphics Shader Space Transformation

SangWon Lee^o

^oSchool of Game, ChungKang College of Cultural Industries

● 요약 ●

3D 그래픽스 환경에서 입체적인 물체가 2D 모니터에 보여지는 과정에는 여러 단계의 공간 변환과 행렬 계산을 거치게 된다. 이러한 공간 변환은 각 단계가 어떤 의미인지 쉽게 이해하기에 어려운 면이 있다. 본 논문에서는 유니티 3D 엔진에서 작동하는 셰이더를 통해 각 단계의 공간을 시각화 하여 학생들이 보다 쉽게 공간 변환을 이해하는 시각화 방법을 제시한다.

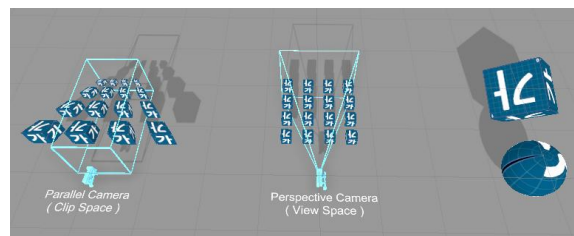
키워드: 셰이더(Shader), 시각화(Visualization), 유니티 3D (Unity 3D)

I. Introduction

교육 현장에서 그래픽 아티스트를 대상으로 셰이더를 강의하면서 느꼈던 여러 어려움들 중 하나가 공간 변환에 대한 이해였다. 특히 클립 공간(Clip Space)의 x, y, z, w 값이 어떤 의미인지 학생들에게 직관적으로 설명하기 쉽지 않았고 수학적 기반이 약한 그래픽 아티스트일수록 어려움이 컸다.

이런 경우 이론적인 설명보다 공간을 시각적으로 구성하여 직접 보여주는 방법이 좀더 효과적이라는 판단에 의해 공간 변환의 각 단계에 직접 개입할 수 있고 시각적으로 볼 수 있는 셰이더를 제작하였다.

이 셰이더는 유니티 3D 엔진에서 작동하며 그래픽 아티스트에게 익숙한 환경, 용어, 조작 방식으로 구성되어있다.



이 프로젝트는 Unity 3D 2021.2.15.fl, URP 이상 환경에서 정상 작동한다.

프로젝트의 VisualizeClipSpace 씬에 샘플이 들어있으며 Assets > VisualizeClipSpace 폴더에서 찾아볼 수 있다.

사용된 셰이더는 Assets > VisualizeClipSpace > Data 폴더의 LambertMVP 셰이더이다.

셰이더가 적용된 재질의 모습은 다음과 같다.

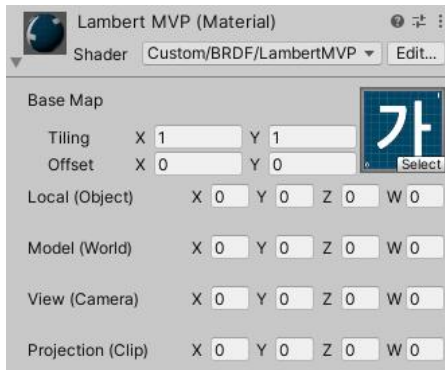
II. Materials & Method

1. 샘플 프로젝트와 셰이더 리소스

GitHub 샘플 프로젝트:

<https://github.com/lswsox/VisualizeClipSpace>

Unity 3D 엔진에서 완성된 모습을 살펴보면 다음과 같다.



2. 각 공간에 개입하는 셰이더 연산

다음은 셰이더에서 중요 프로퍼티가 선언된 모습이다.

```

_L("Local (Object)", Vector) = (0, 0, 0, 0)
_M("Model (World)", Vector) = (0, 0, 0, 0)
_V("View (Camera)", Vector) = (0, 0, 0, 0)
_P("Projection (Clip)", Vector) = (0, 0, 0, 0)
    
```

4 개의 `_L`, `_M`, `_V`, `_P` 프로퍼티가 어떤 의미인지 확인할 수 있고 HLSL 블록에서 4개 모두 float4 자료형으로 선언되었다.

각 공간에 셰이더가 어떻게 개입하는지 다음 버텍스 셰이더 코드로 확인할 수 있다.

```

IN.positionOS += _L; // 오브젝트 공간에 개입
float4 posWS = mul(UNITY_MATRIX_M, IN.positionOS); // M
posWS += _M; // 월드 공간에 개입
float4 posVS = mul(UNITY_MATRIX_V, posWS); // V
posVS += _V; // 뷰(카메라) 공간에 개입
OUT.positionHCS = mul(UNITY_MATRIX_P, posVS); // P
OUT.positionHCS += _P; // 클립 공간에 개입
    
```

참고로 `UNITY_MATRIX_M`, `UNITY_MATRIX_V`, `UNITY_MATRIX_P`는 유니티 3D 엔진의 렌더 파이프라인이 제공하는 내장 행렬 변수이며, 각각 Model, View, Projection 공간 변환의 내용을 담고 있다.

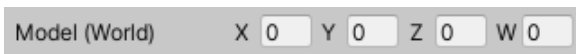
3. 각 프로퍼티 값의 역할과 시각화

3-1 Local (Object)



`VisualizeClipSpace` 씬의 MVP_Cube 오브젝트에 지정된 재질의 Local (Object) 값을 다양하게 변경해보면 오브젝트의 로컬 좌표계에 개입함을 시각적으로 확인할 수 있다.

3-2 Model (World)



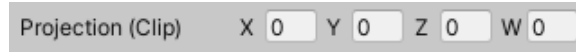
앞서와 같은 재질의 Local (Object) 값을 다양하게 변경해보면 오브젝트의 로컬 방향과 무관하게 월드 좌표계에 개입함을 시각적으로 확인할 수 있다.

3-3 View (Camera)



앞서와 같은 재질의 View (Camera) 값을 다양하게 변경해보면 Scene Camera 혹은 Game Camera의 좌표계에 개입함을 시각적으로 확인할 수 있다.

3-4 Projection (Clip)



앞서와 같은 재질의 Projection (Clip) 값을 다양하게 변경해도 View (Camera) 값과 다른 점을 확인하기 쉽지 않다. 특히 Z 값의 변경은 아무 변화 없는 것처럼 보인다.

여기서 Projection (Clip)의 Z 값을 설명하기 위해 다음의 순서대로 `VisualizeClipSpace` 씬을 설정한다.

먼저 Hierarchy에서 Perspective Camera 오브젝트를 비활성화한 다(Game View는 Parallel 시점이 됨). 그리고 Parallel Camera 오브젝트의 하위에 있는 CubeGroupPerspective 오브젝트의 로컬 Z 위치 값을 다양하게 변경해본다.

여기서 CubeGroupPerspective 오브젝트는 클립 공간에서 변형된 버텍스들을 시각화한 샘플이며, 로컬 Z 위치 값이 변할 때 Scene View와 Game View에서 보여지는 모습을 통해 클립 공간의 Z 값의 개념을 시각화할 수 있다.

III. Conclusions

본 논문에서 제시한 셰이더와 시각화는 셰이더의 공간 변환을 설명하는 한 방식이며 강의 학생들을 대상으로 긍정적인 피드백을 얻을 수 있었다. 그러나 매우 많은 학생들을 대상으로 조사한 것은 아니어서 객관적인 효과를 확인할 수 있는 단계가 아님은 아쉬운 부분이다.

REFERENCES

[1] Opengl-tutorial, Tutorial 3 : Matrices <http://www.opengl-tutorial.org/beginners-tutorials/tutorial-3-matrices/>