A cinematic screenshot from the video game The Last of Us. A female character, Ellie, is seen from behind, standing on a rocky, desolate landscape. She is looking towards a distant, ruined city with tall, jagged spires under a dramatic, sunset-colored sky. The scene is atmospheric and serves as a background for the title text.

게임 엔진

LEC 16 머티리얼 (1)



한국공학대학교
TECH UNIVERSITY OF KOREA

이대현 교수

학습 내용

- 물리 기반 렌더링(PBR: Physically Based Rendering)
- 머티리얼(Material)

Static Lighting OFF

✕ allow static lighting

▼ Engine - Cooker

Various cooker settings.

▼ Cooker

▼ Advanced

▼ r values that need to be versioned

15 Array elements + 🗑

Index [0]

r.AllowStaticLighting ▼

▼ Engine - Rendering

Rendering settings.

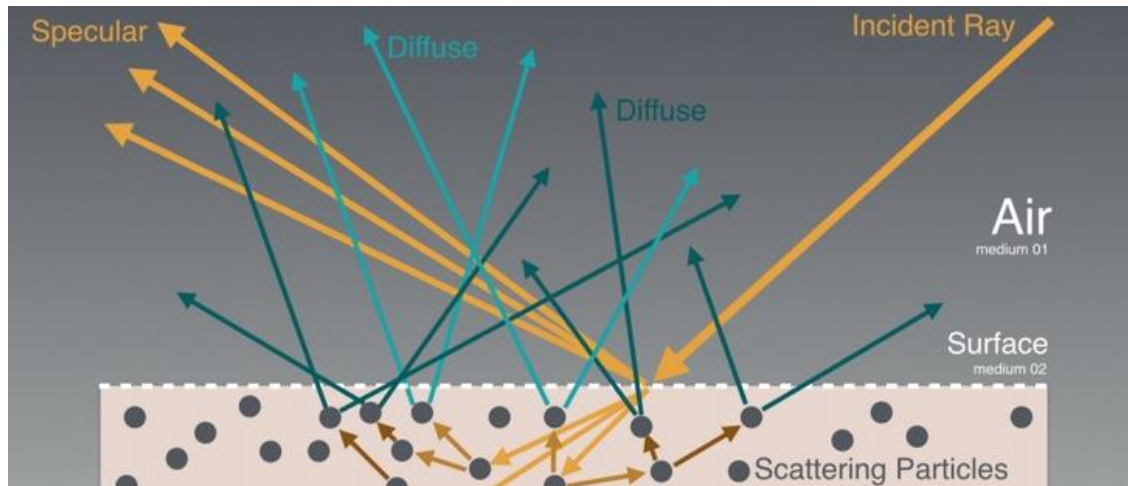
▼ Misc Lighting

Allow Static Lighting ☐

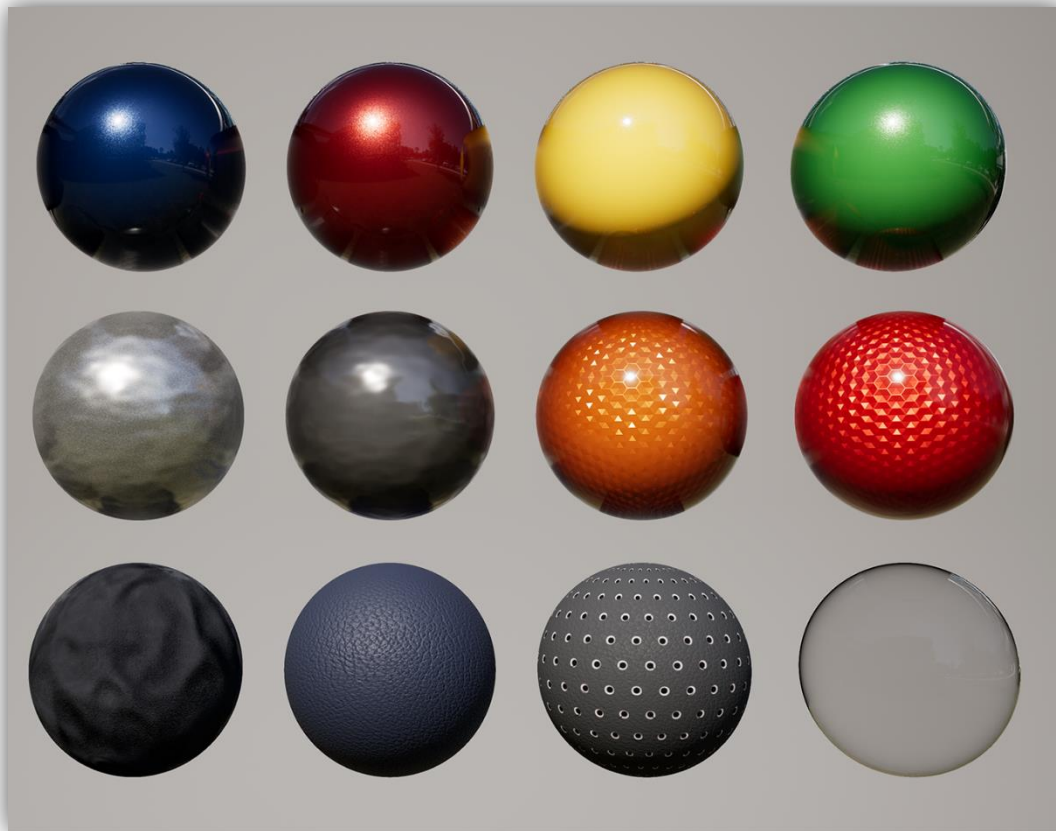
Whether to allow any static lighting to be generated and used, like lightmaps and shadowmaps. Games that only use dynamic lighting should set this to 0 to save some static lighting overhead. Disabling allows Material AO and Material BentNormal to work with Lumen Global Illumination. Changing this setting requires restarting the editor.

머티리얼(Material: 재질)

- 물체의 재질을 정의하는 애셋 - 재질은 표면의 시각적 모양을 결정.
- 물체를 색칠할 수 있는 " 페인트 " 라 보면 됨.
- 색상, 광택, 투과성 등을 정의할 수 있음.
- 수학적 관점
 - 광원들과 물체 표면과의 상호 작용을 모델링한 계산식 == 머티리얼
 - 계산 결과는 물체 표면 위의 점들의 색상



다양한 머티리얼




퐁 모델 - 전통적인 조명 모델

빛의 난반사에 의한 성분.

자체 발광 성분.

주변광원에 따른 성분.

빛의 전반사에 의한 성분.

$$I = I_e + I_a + I_d + I_s$$


풍 모델 계산식

$$I = K_a \otimes L_a + (K_d \otimes L_d) \cos \theta + (K_s \otimes L_s) (\cos \alpha)^n$$

입사각 → 법선에 따라 바뀜. 시야각과 반사각의 차이

PBR(Physically-Based Rendering)

- 빛의 작용을 경험적으로 추정하지 않고, 실제로 어떠한 작용을 하는지 추정함.
- 좀 더 정확하고 자연스러운 결과를 얻을 수 있음.
- 다양한 라이팅 환경에서 똑같이 작동.
- 사용되는 값들이 덜 복잡함.
- 직관적인 인터페이스 가능.
- 비실사 렌더링에도 적용 가능.



PBR(Physically-Based Rendering)

■ 기존 Phong 조명 모델의 한계

- 디퓨즈, 스펙큘러 파워 등의 속성들이 직관적으로 이해하기 어려움.
- 사실적인 표현을 위해 복잡한 추가적인 모델링 수식을 만들어 주어야 함.
- 결과적으로, 원하는 색을 얻기 위한 시간과 비용이 과다함.

■ 물리 기반 렌더링의 장점

- 쉽게 파악할 수 있는 속성을 사용.
- 물리 상황을 그대로 직접적으로 모델링하여 사용.
- 빠르게 원하는 색상과 질감을 얻을 수 있음.

<https://docs.unrealengine.com/latest/KOR/Engine/Rendering/Materials/PhysicallyBased/index.html>

PBR(Physically-Based Rendering)

- **Material의 핵심 속성은 딱 네가지!!!**

- Base Color - 바탕색
- Roughness - 거칠기
- Metallic - 금속성
- Specular - 반사성

Base Color

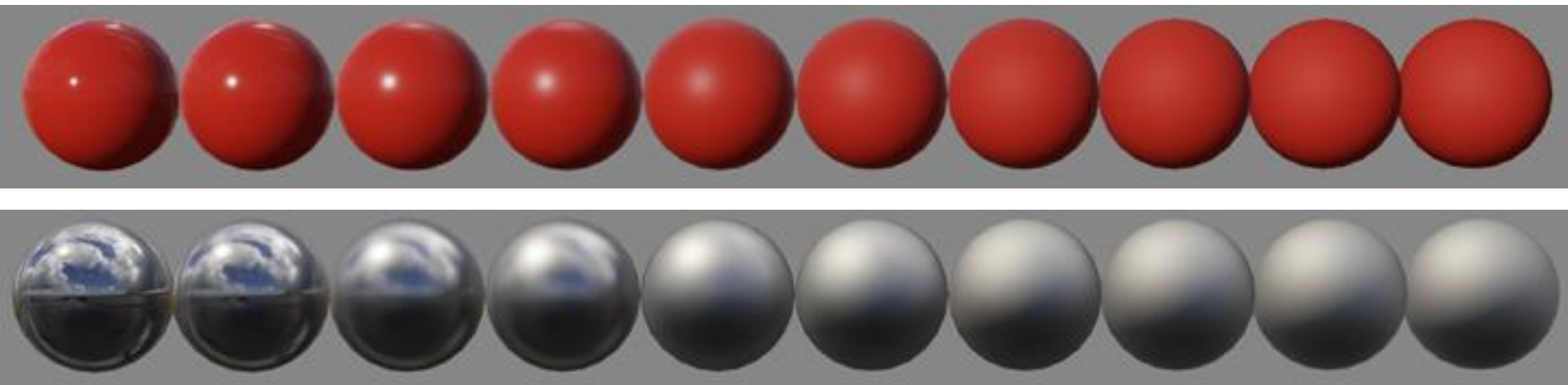
- 물체의 기본 색상, 전체적인 색상을 정의
- Vector3 – RGB 세개의 채널
- 각 채널은 0과 1사이의 실수값
- 편광 필터로 사진을 찍었을 때의 색(편광필터는 경반사 성분을 제거)

재질	베이스 컬러 (R, G, B)
철	(0.560, 0.570, 0.580)
은	(0.972, 0.960, 0.915)
알루미늄	(0.913, 0.921, 0.925)
금	(1.000, 0.766, 0.336)
구리	(0.955, 0.637, 0.538)
크롬	(0.550, 0.556, 0.554)
니켈	(0.660, 0.609, 0.526)
티타늄	(0.542, 0.497, 0.449)
코발트	(0.662, 0.655, 0.634)
백금	(0.672, 0.637, 0.585)

재질	베이스 컬러 강도
숫	0.02
새 아스팔트	0.02
현 아스팔트	0.08
맨땅	0.13
풀밭	0.21
모래사막	0.36
새 콘크리트	0.51
빙하	0.56
갓 내린 눈	0.81

Roughness - 거칠기

- PBR에서 가장 중요한 속성
- 거친 재질은 빛을 좀 더 여러 방향으로 반사 산란시킴.
- 반사가 희미한지 선명한지? 하이라이트가 얼마나 퍼져있는지?
- 0 (완전부드러움 , 거울 반사), 1(완전거칠. 무광 또는 확산)



Metallic

- 표면의 "금속성" 을 제어
- 비금속은 메탈릭 값이 0 이며, 금속은 1
- 부식되었거나 먼지 또는 녹이 낀 메탈같은 혼합 표면을 만들 때는, 0 과 1 사이 값이 필요할 수도 있음.. → 하지만, 거의 대부분의 경우 0 또는 1임. 중간은 일단 없다고 생각하는 것이 좋음.



Specular

- 비금속의 반사량을 세부 조정하는 데 사용됨. 금속에는 효과가 없음.
- 대부분의 경우 기본값 0.5, 연결하지 않아도 OK
- 고급 사용의 경우 굴절률 제어에 사용됨.

스페큘러 측정값:

재질	스페큘러
유리	0.5
플라스틱	0.5
석영	0.570
얼음	0.224
물	0.255
우유	0.277
피부	0.35

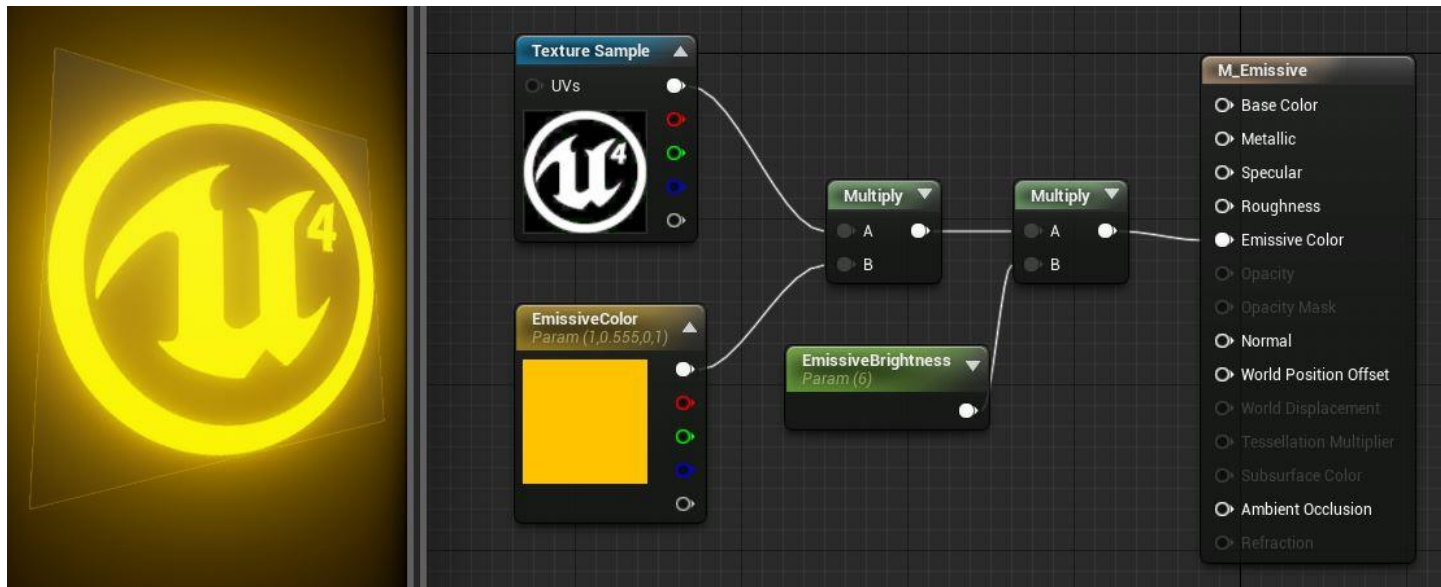
노멀 - 법선

- 개별 픽셀의 법선 정보.
- 표면의 미세한 물리적 디테일을 더함.



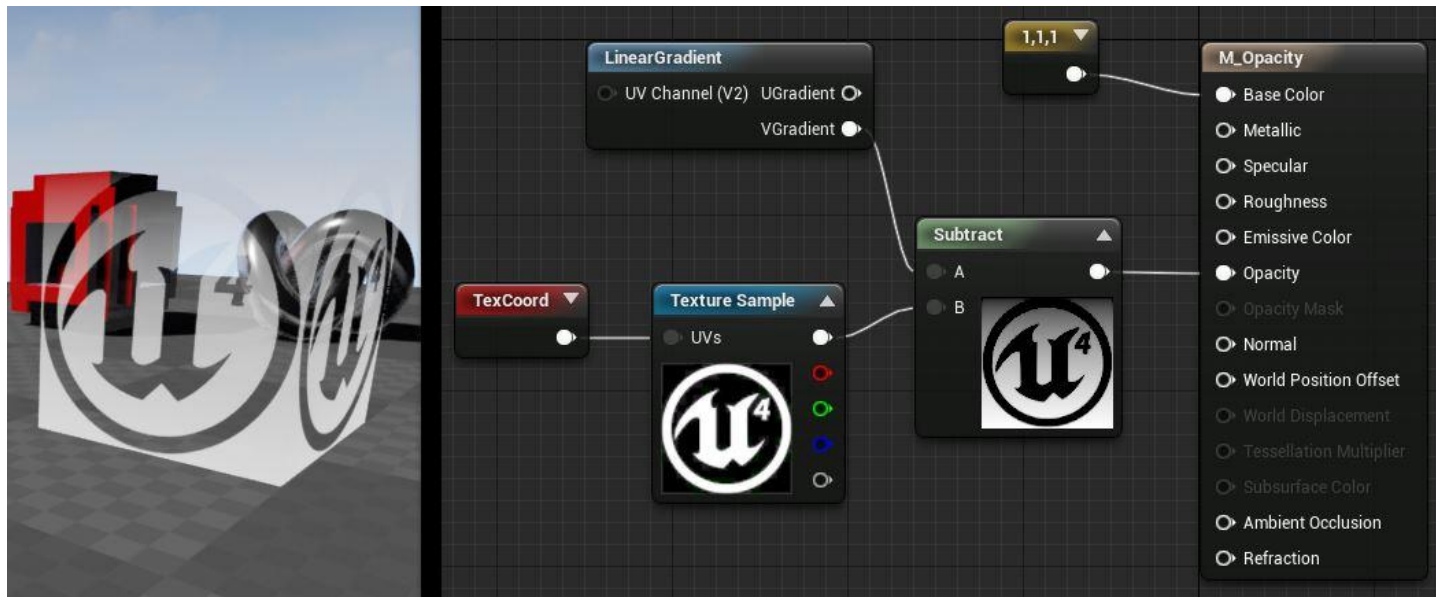
Emissive Color - 발광색

- 글로우 효과
- 1보다 큰 값이 허용



Opacity - 불투명도

- 투명 0 \leftrightarrow 1 불투명



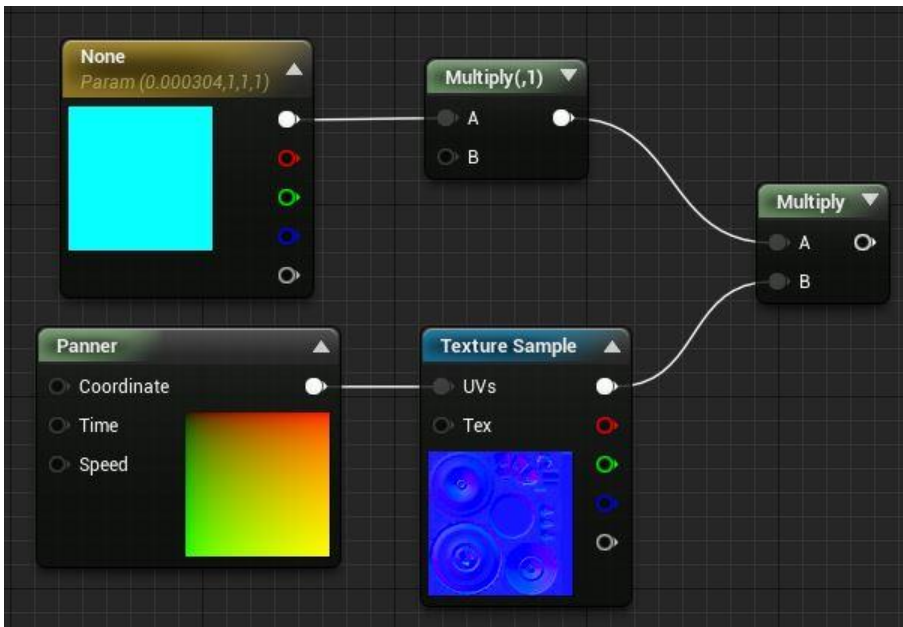


실습

간단한 머티리얼 제작

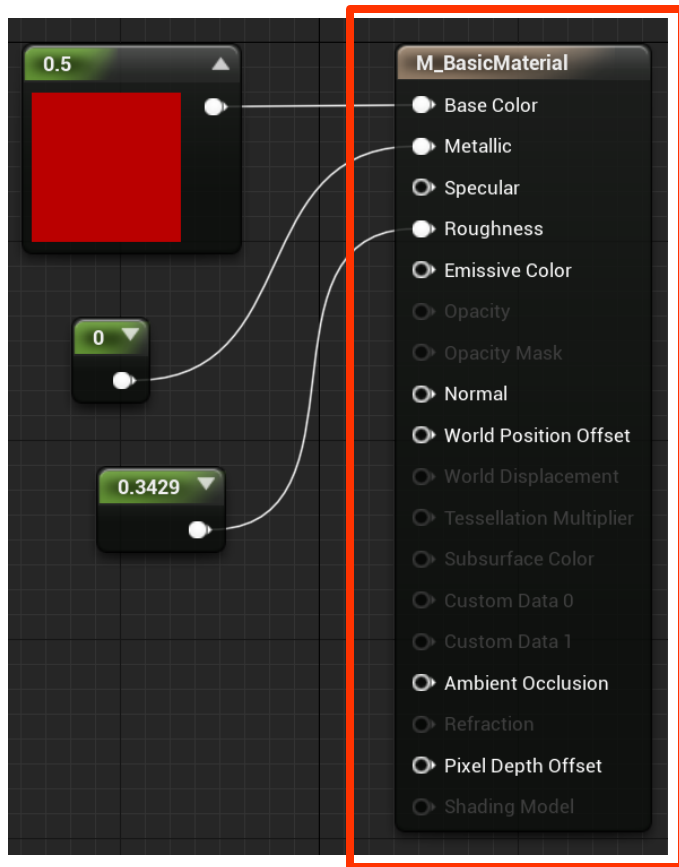
머티리얼 편집

- 블루프린트와 유사한 스타일로 노드와 와이어를 이용하여 표현식 망을 제작함.



메인 머티리얼 노드

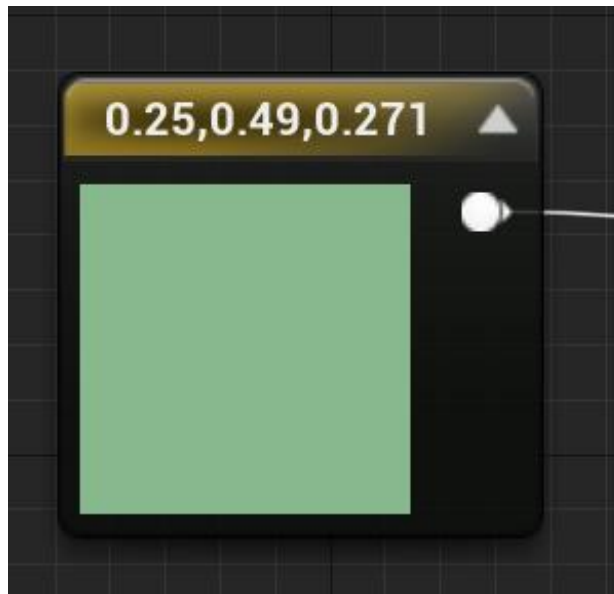
■ 머티리얼 속성들을 최종적으로 연결하는 결과 노드



Constant 노드

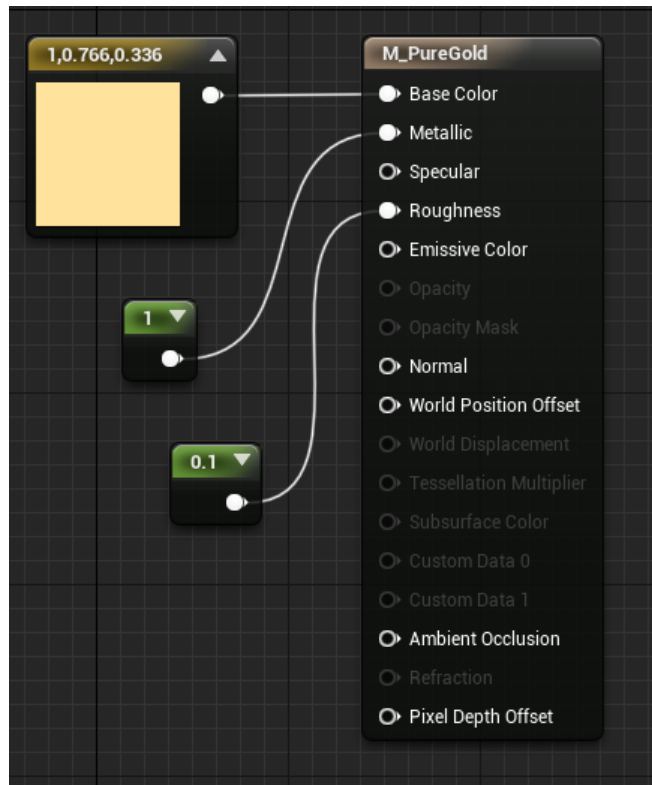


0-1 사이의 값을 정의



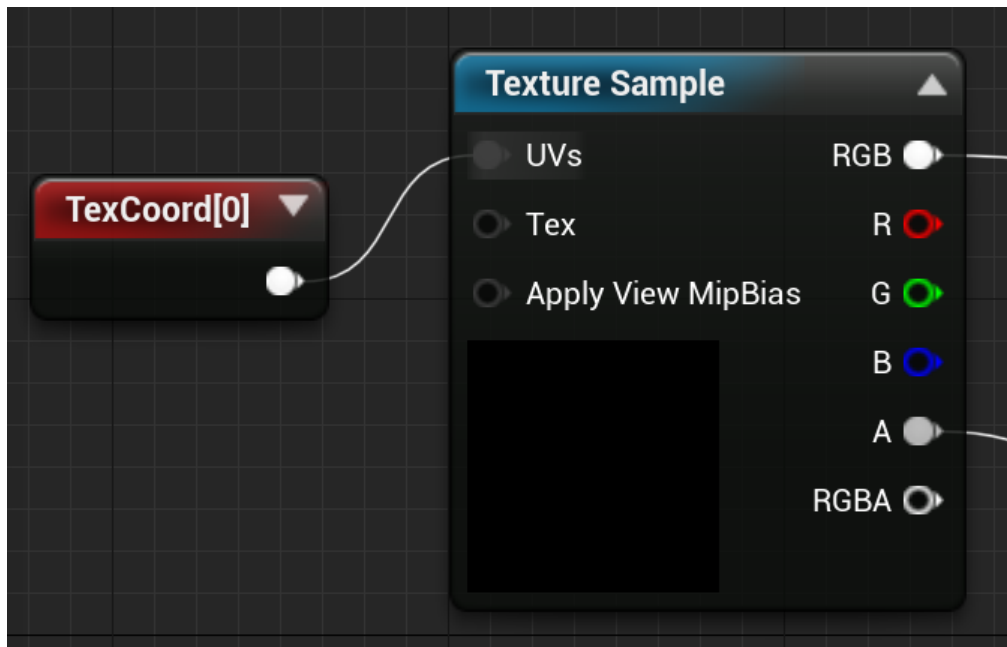
3차원 벡터값을 정의
주로 (R,G,B)의 색상값을 정의

금 - M_PureGold



Texture Sample 노드

- 이미지로부터 (R,G,B) 값을 가져옴.
- TexCoord 로 텍스처 UV 좌표 지정 및 변경.

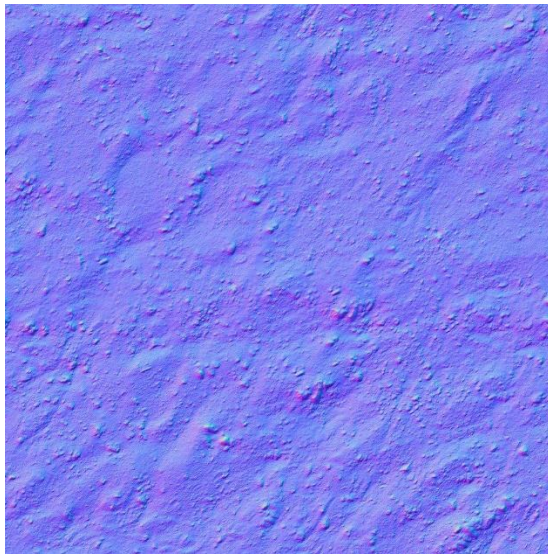
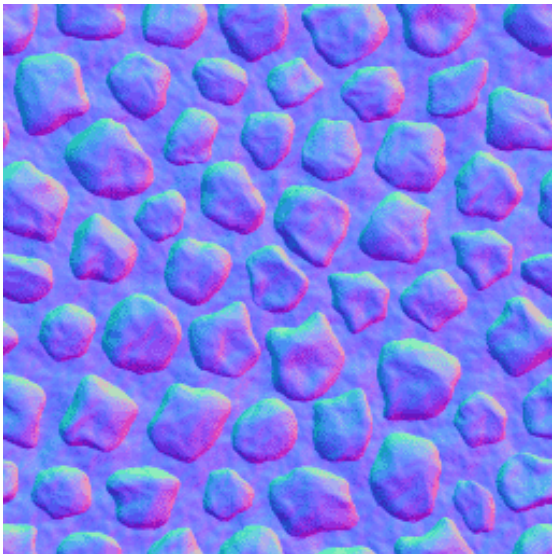


Base Color Texture

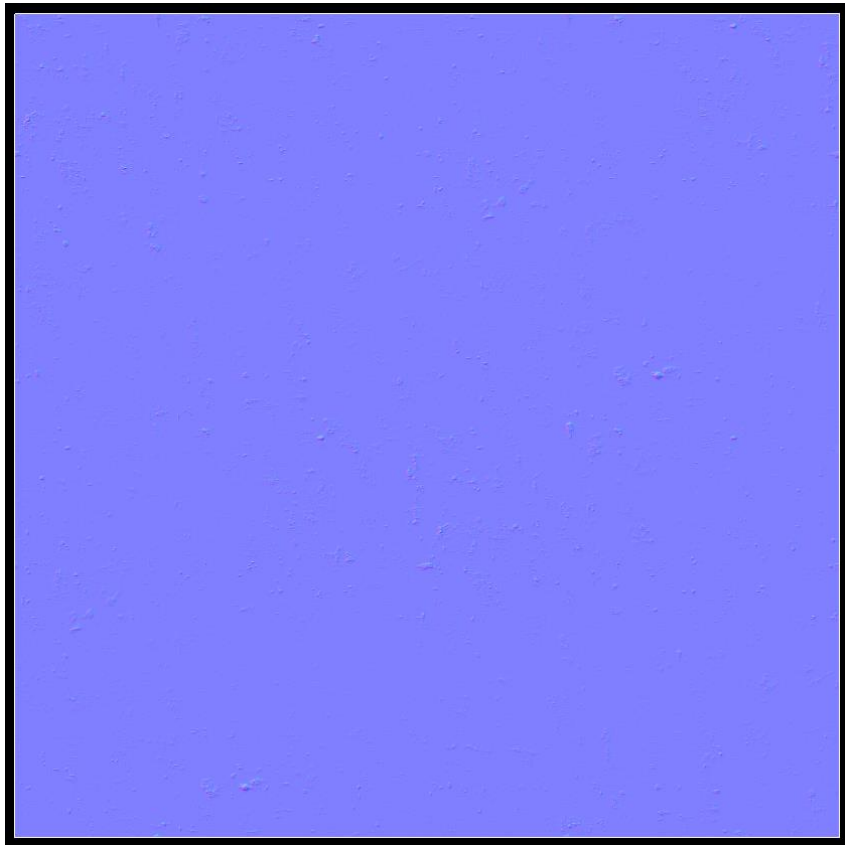


Normal Texture

- 개별 픽셀의 법선 정보를 나타내기 위해 Texture를 이용함.
- RGB 를 xyz 성분으로 해석 - 평면은 모두 Blue 임.

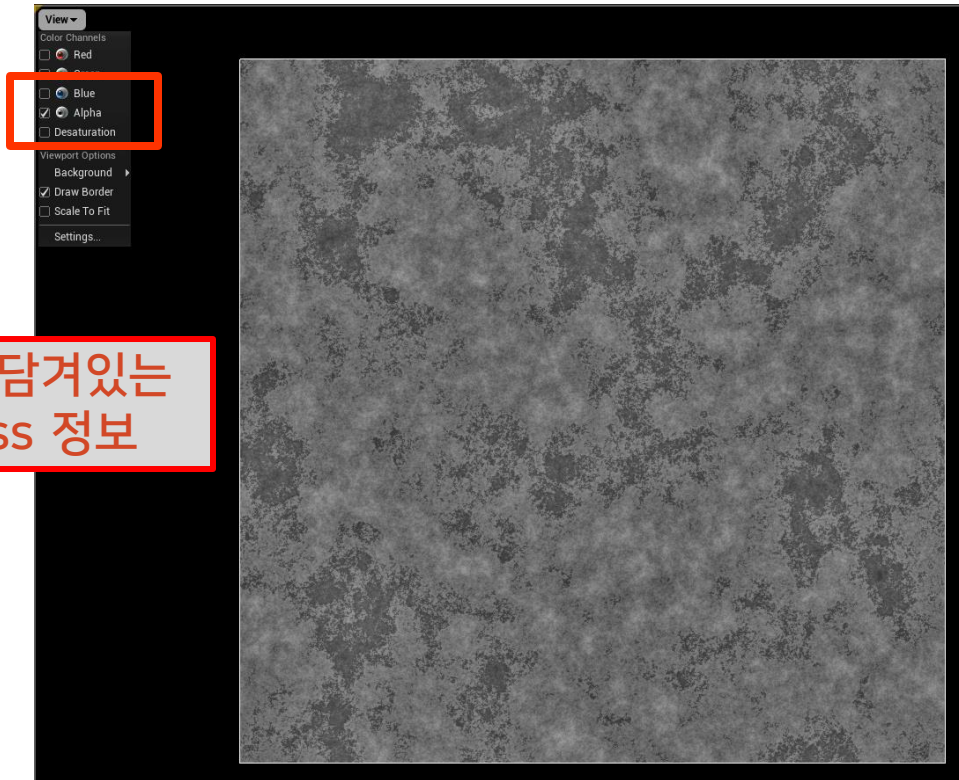


Normal Texture



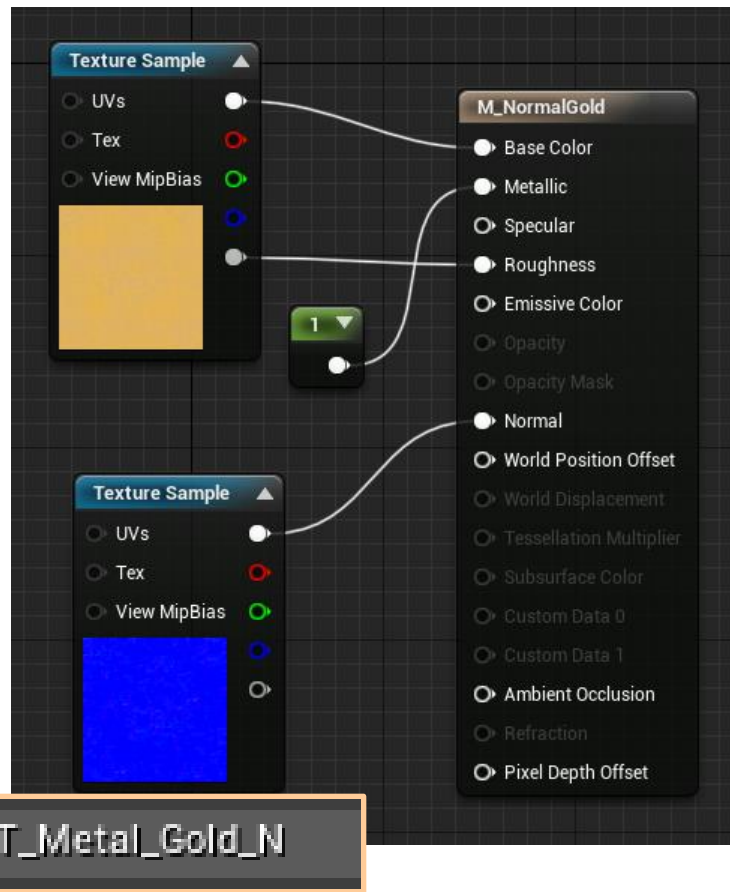
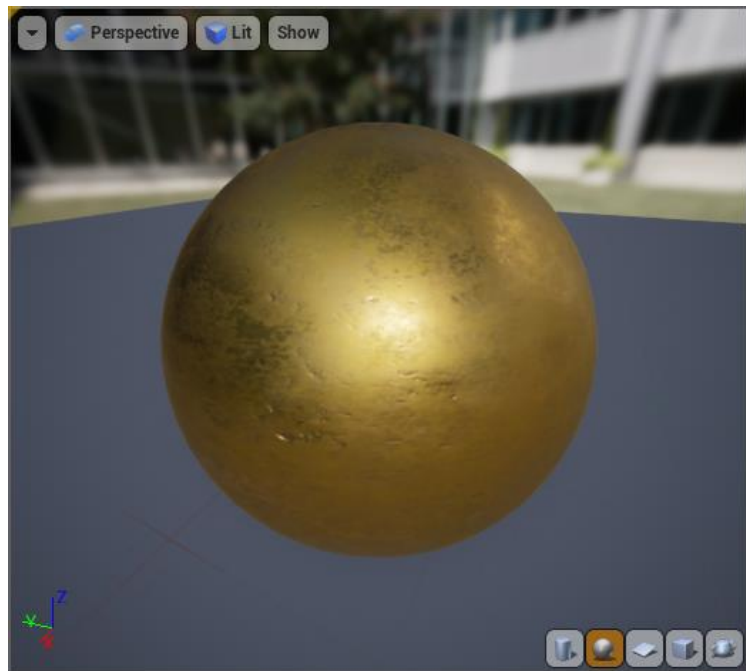
텍스처의 알파 채널 활용

- 알파 채널은 일반적으로 투명도를 나타내기 위해서 사용됨.
- Roughness 정보 등을 담기 위한 채널로도 빈번히 사용됨.

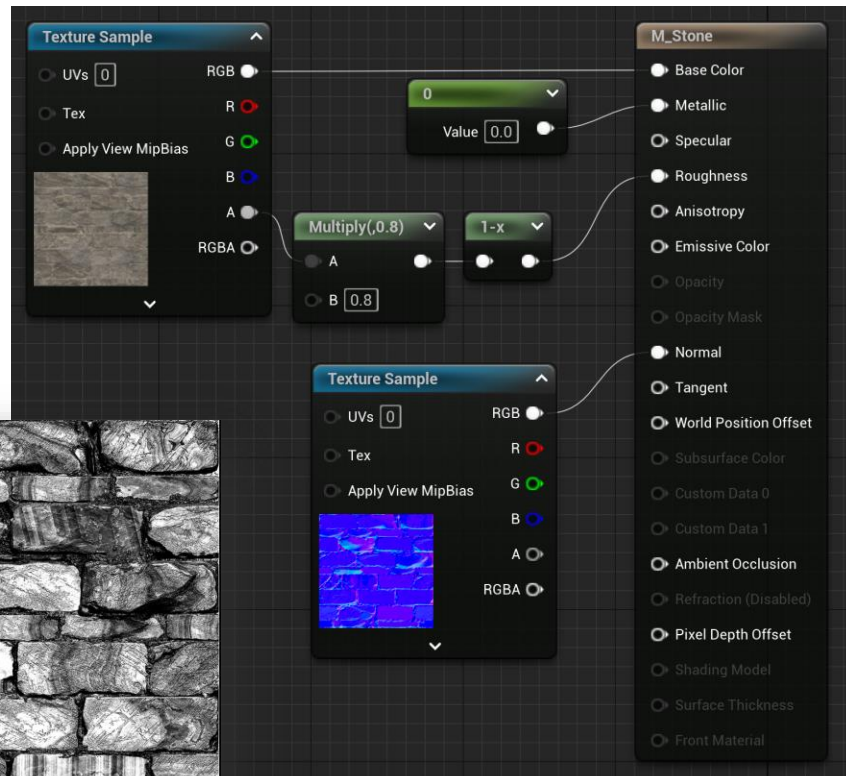
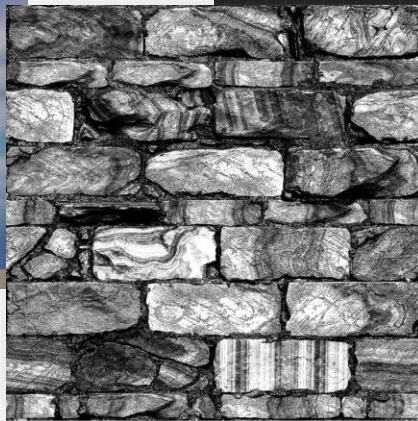


알파 채널에 담겨있는
Roughness 정보

금 - M_NormalGold

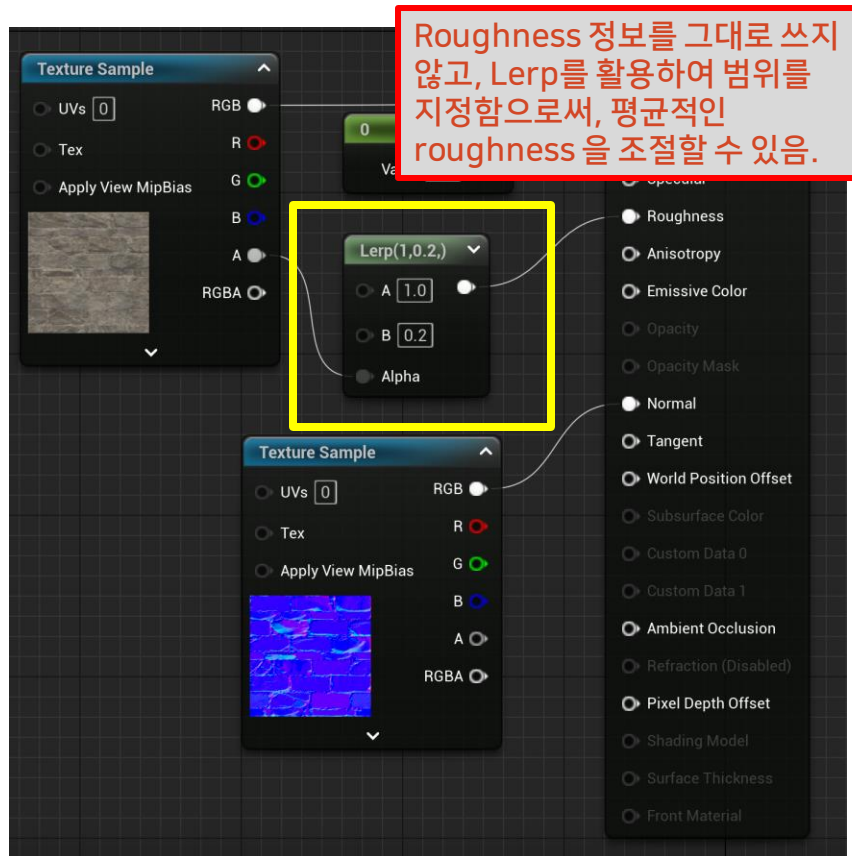
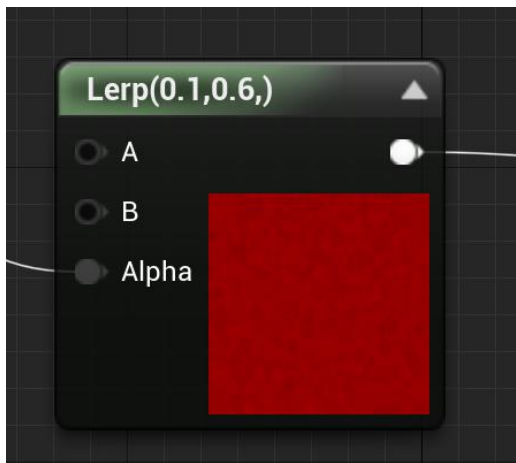


Roughness 튜닝

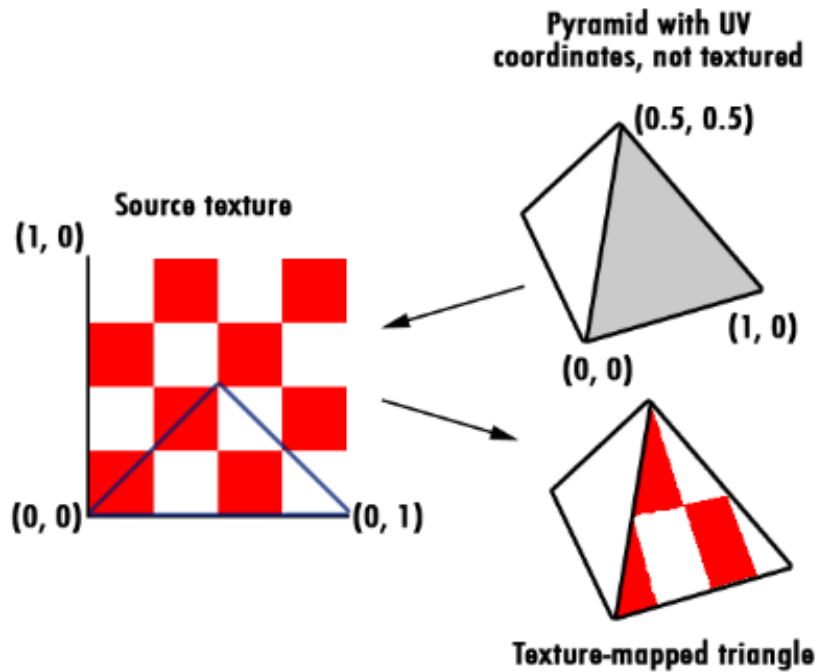


Lerp 노드의 활용

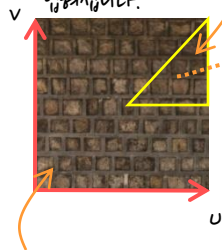
- A와 B사이를 Alpha(섞는 비율) 만큼 선형 보간.
- A와 B 두개의 정보를 섞어서, 두 개의 내용을 비율 대로 반영하는 의미.
- A와 B를 단순히 더하면, 결과값이 1을 넘어갈 수 있기 때문임.



UV Mapping



삼각형 안의 점들의 색상이 그대로
UV 맵을 통해 스테틱 메시의 표면에
입혀집니다.

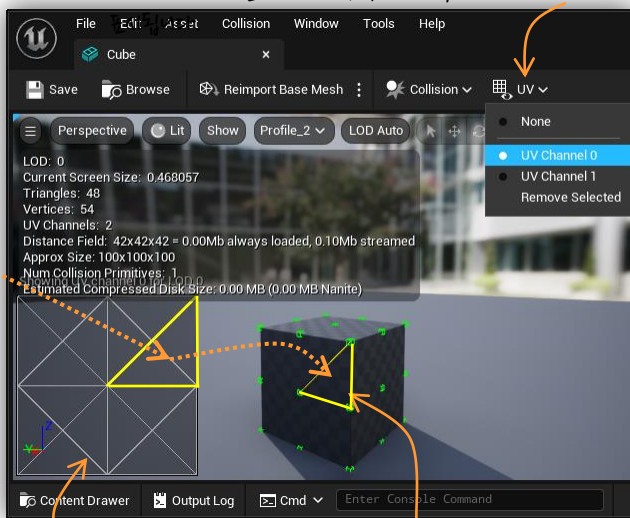


텍스처 상의 어떤 점의 위치를
나타내기 위해 U와 V축으로 구성되는
UV 좌표계를 사용합니다. U, V는 0과
1사이의 실수값을 갖습니다.

UV 맵입니다. 이 맵은 텍스처
이미지가 어떤 식으로 스테틱 메시의
표면에 할당되는지 알려줍니다.

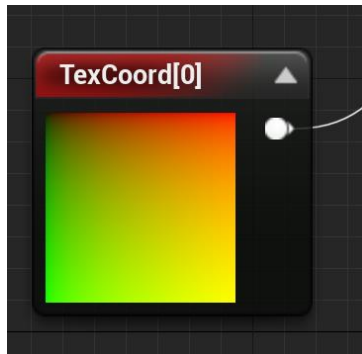
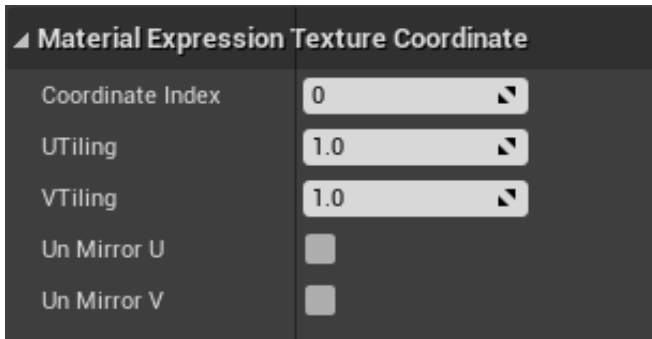
폴리곤을 구성하는 선분을 클릭하면
노랑색으로 바뀌고, UV 맵상에서
해당되는 선분도 함께
표시됩니다.

UV channel 0을 선택하면, 뷰포트 좌측 하단에 UV 맵이

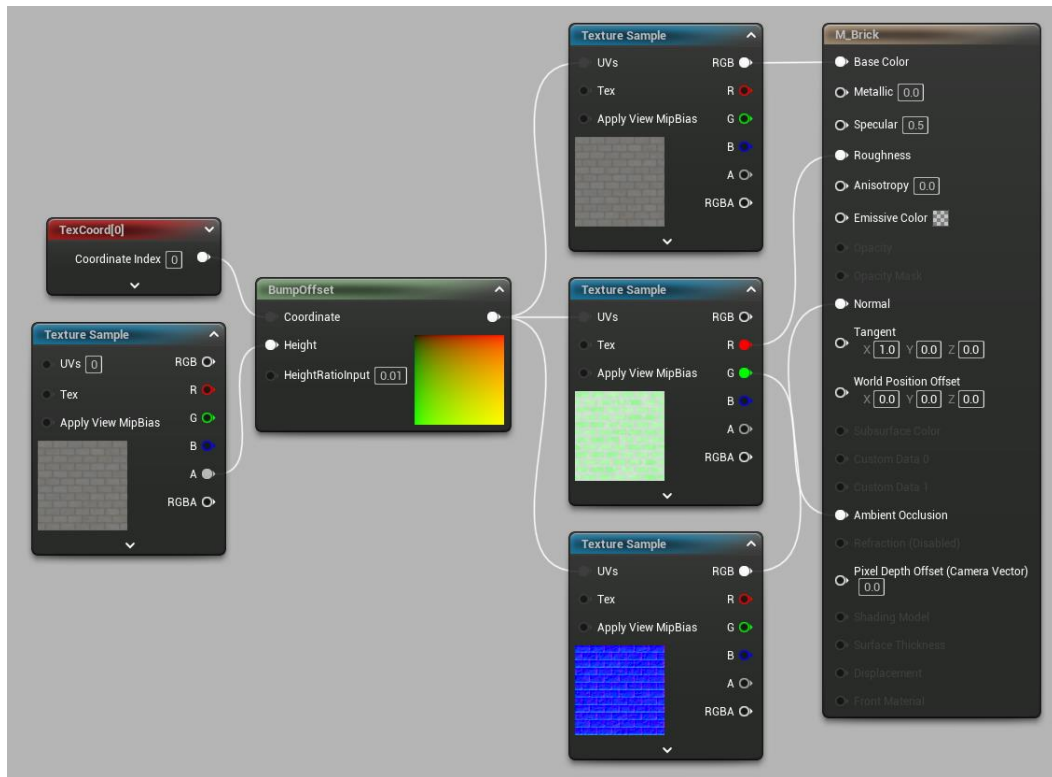
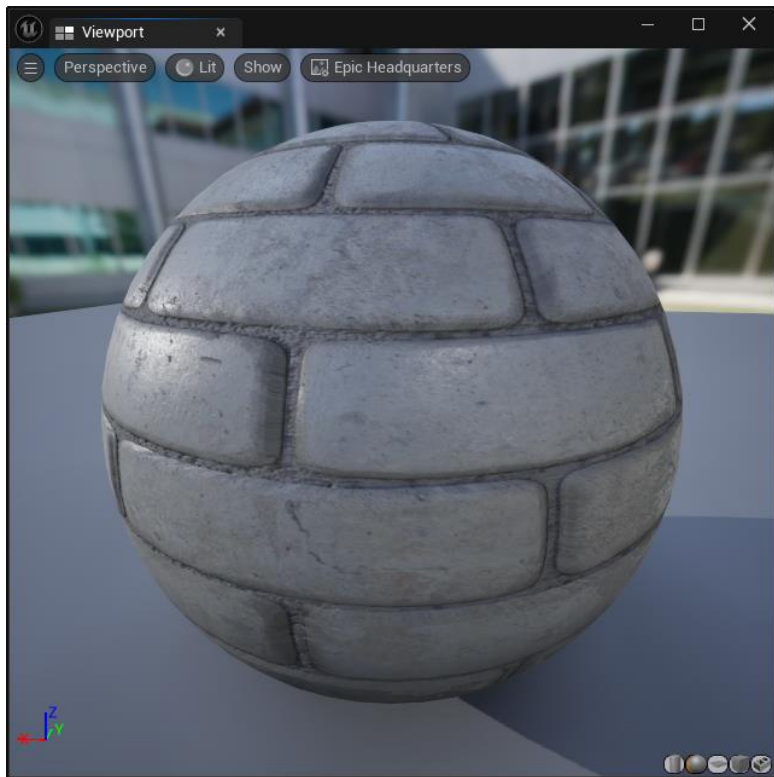


TexCoord 노드

- 텍스처에 제공되는 UV 좌표값. U,V 모두 0과 1사이의 범위.
- Tiling 값을 클수록 단위 면적당 텍스처 이미지의 반복이 많아지는 효과.

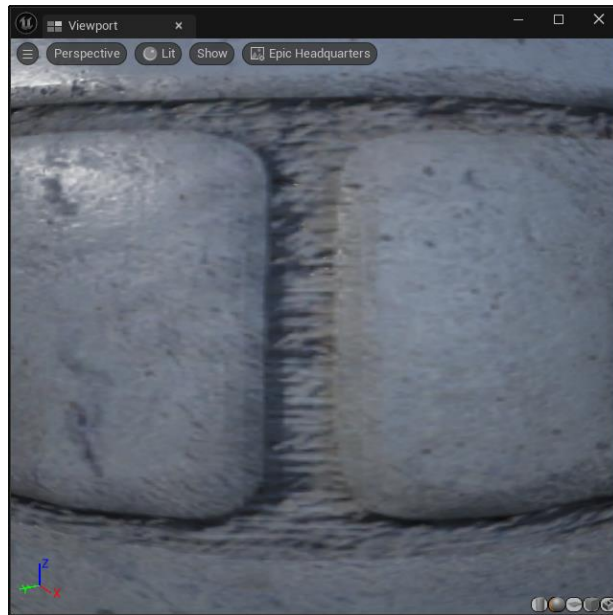
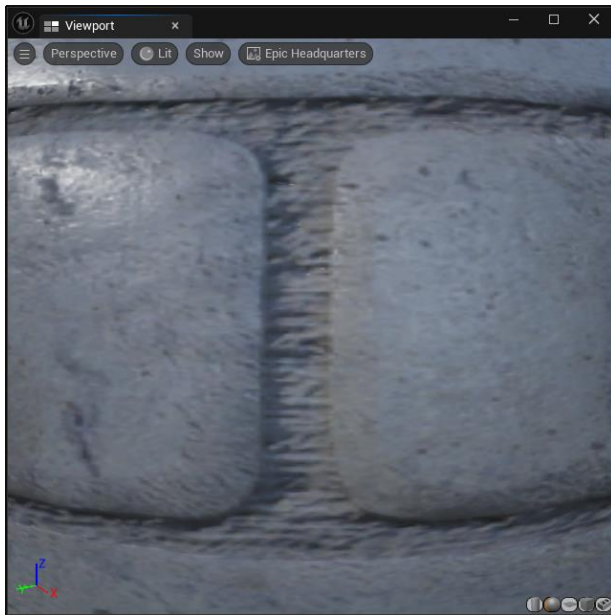


벽돌벽

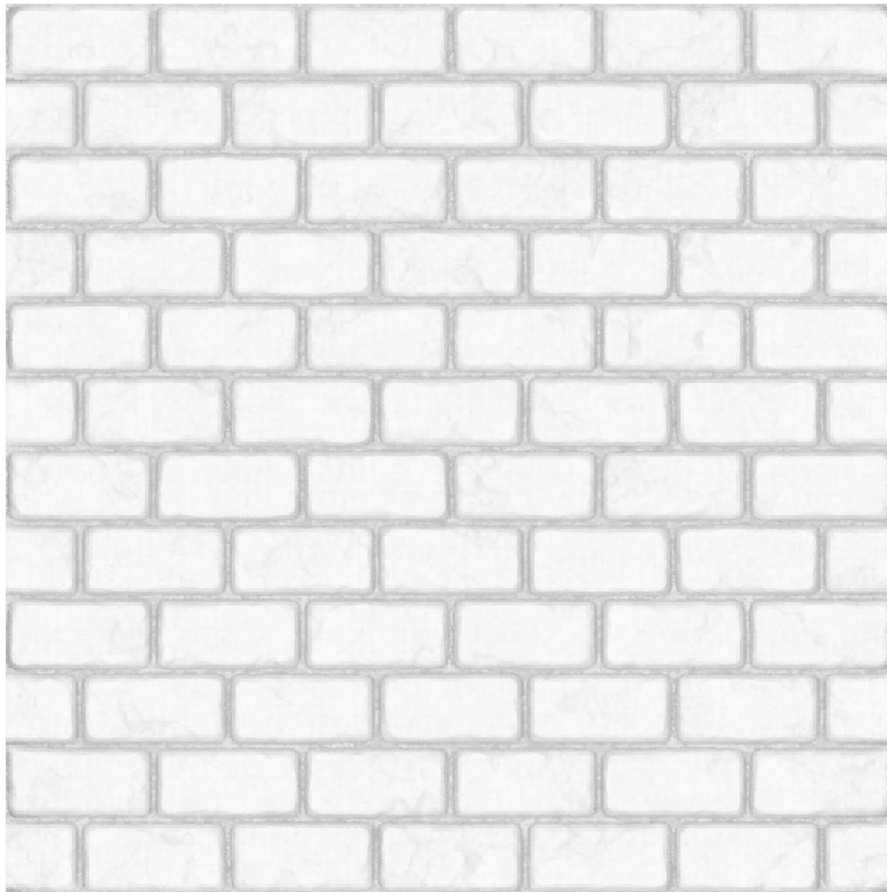


AO(Ambient Occlusion) 효과 - 구석 그림자의 표현

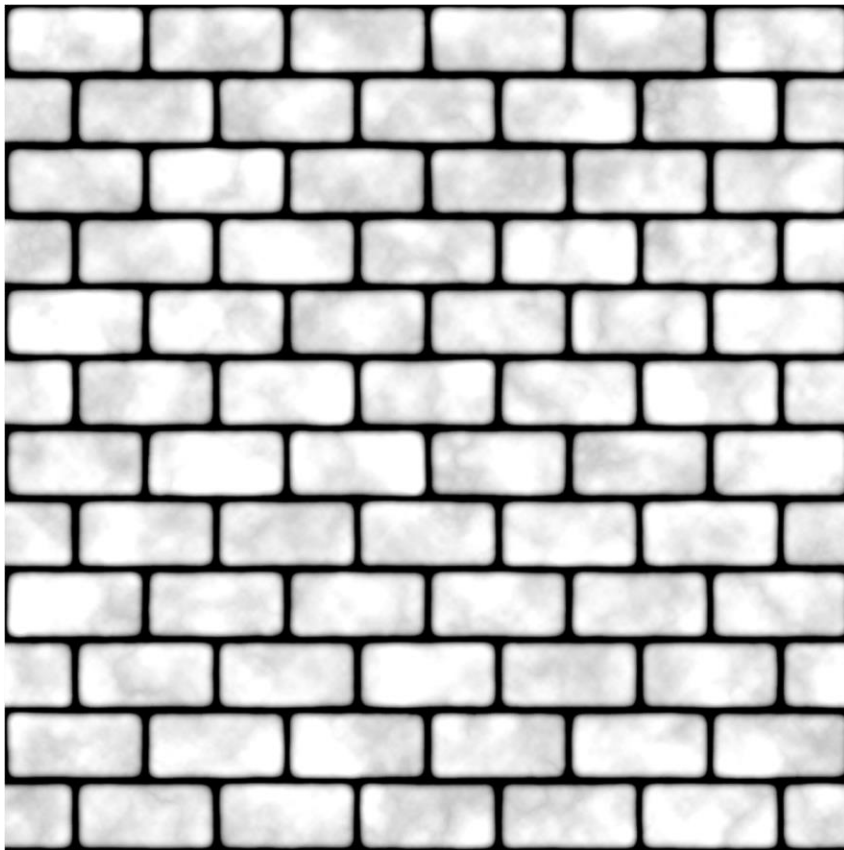
- 가짜 돌출 효과에 따른 그림자까지도 표현하기 위해 사용



AO (Ambient Occlusion) Texture

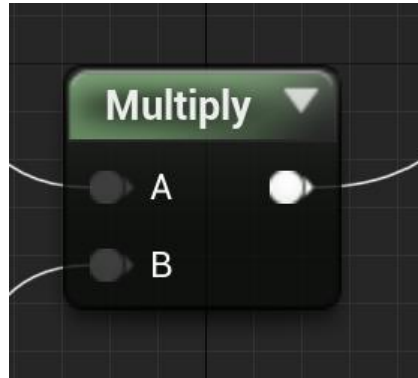
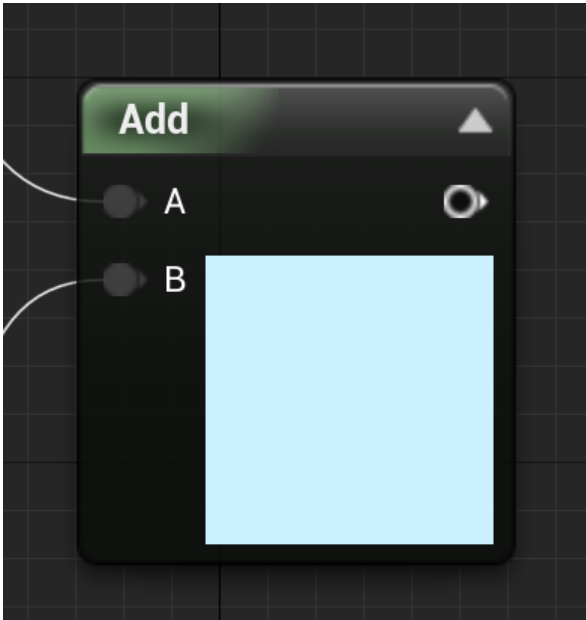


Parallax Texture

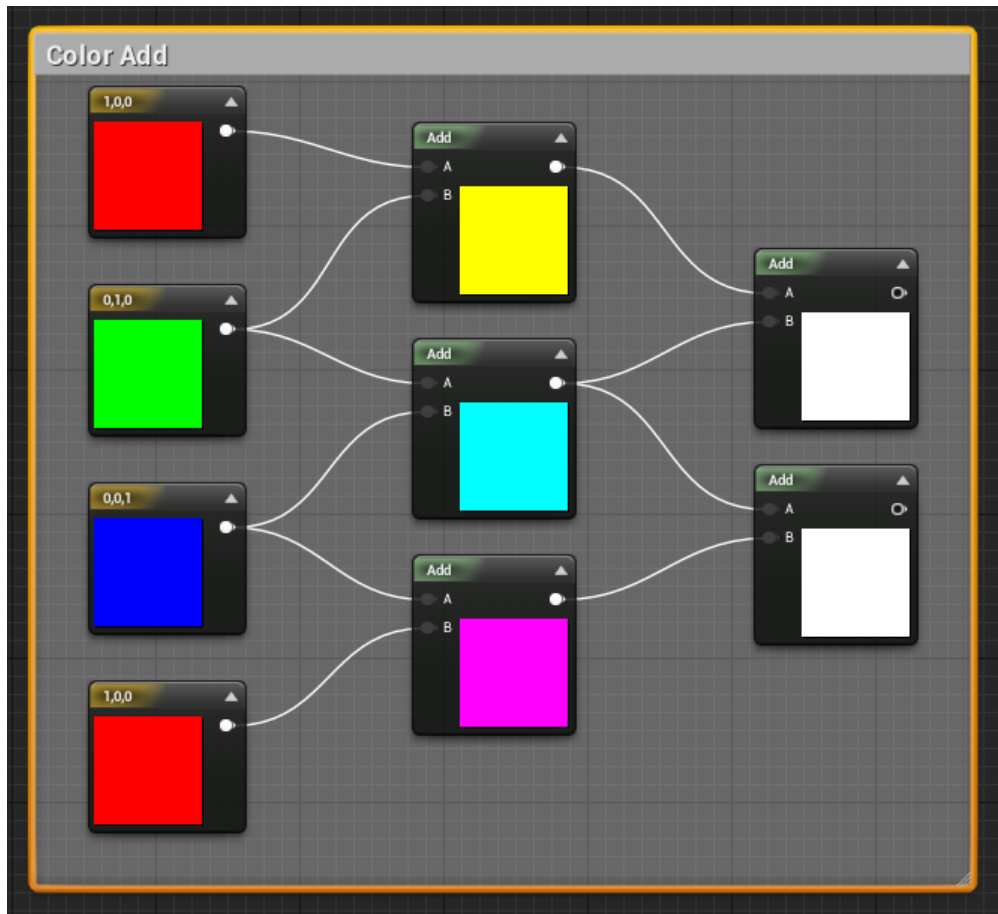


Add 와 Multiply 노드

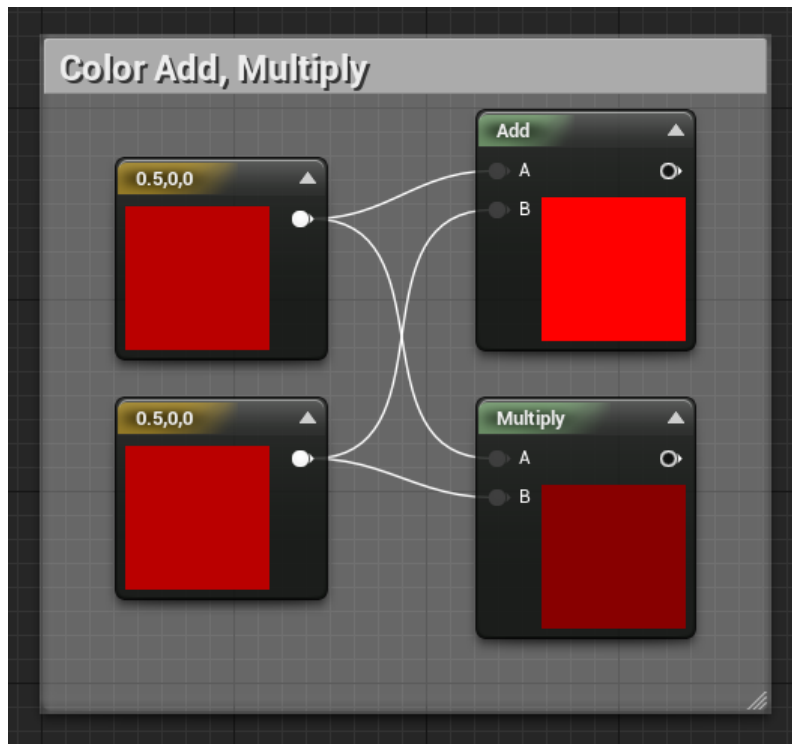
- 두개의 입력 벡터를 더함.
- 채널 단위로 연산이 이루어짐.



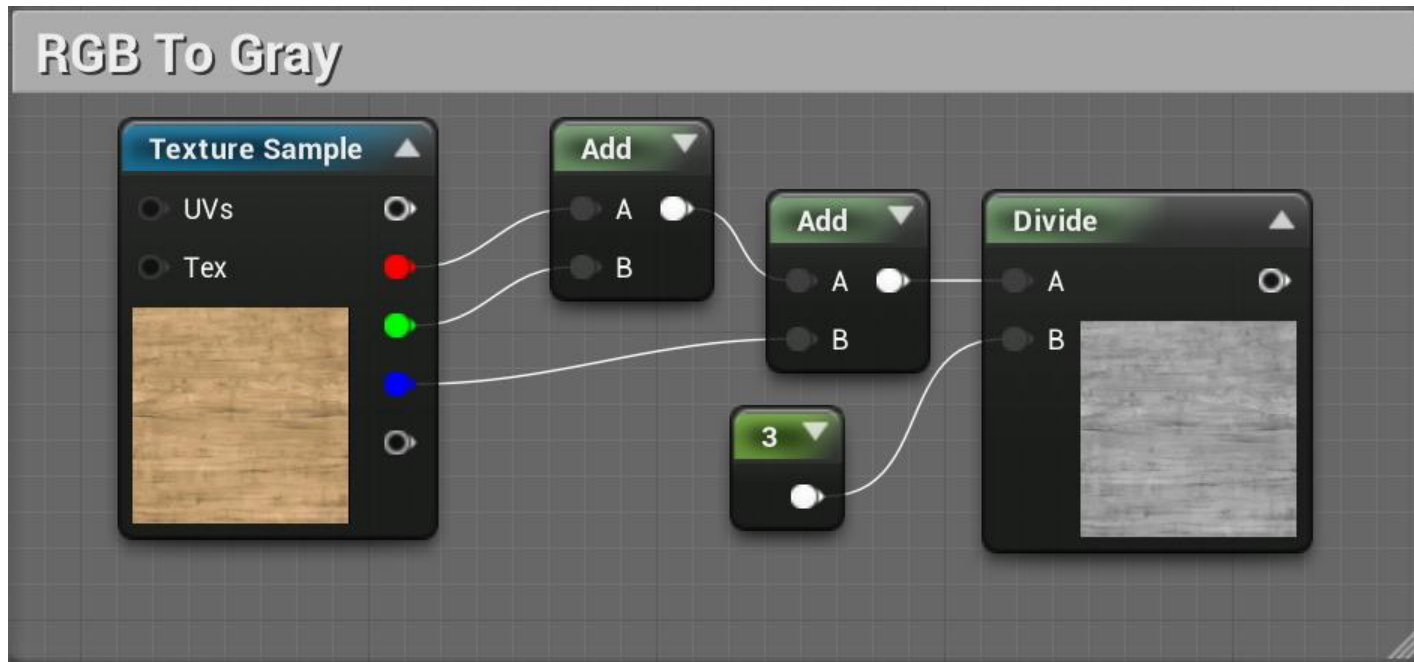
Color Add



Color Add와 Multiply의 효과



RGB를 Gray로 바꾸기

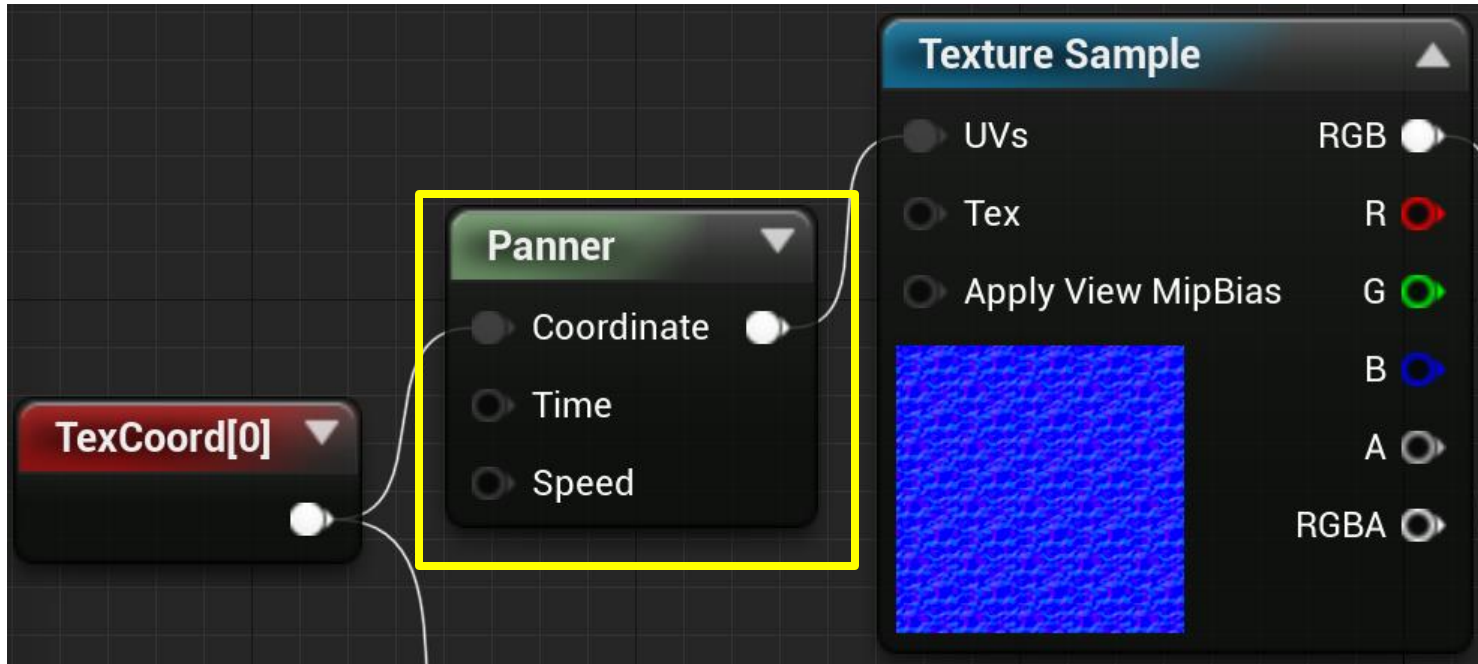


Material 창에서 'L'을 누른 채, 회전하면, Light를 회전시킬 수 있음



Panner 노드

- UV 좌표를 이동시켜서 텍스처의 애니메이션을 구현.



물 - M_Water

