

게임 엔진 프로그래밍

LAB8 – Material



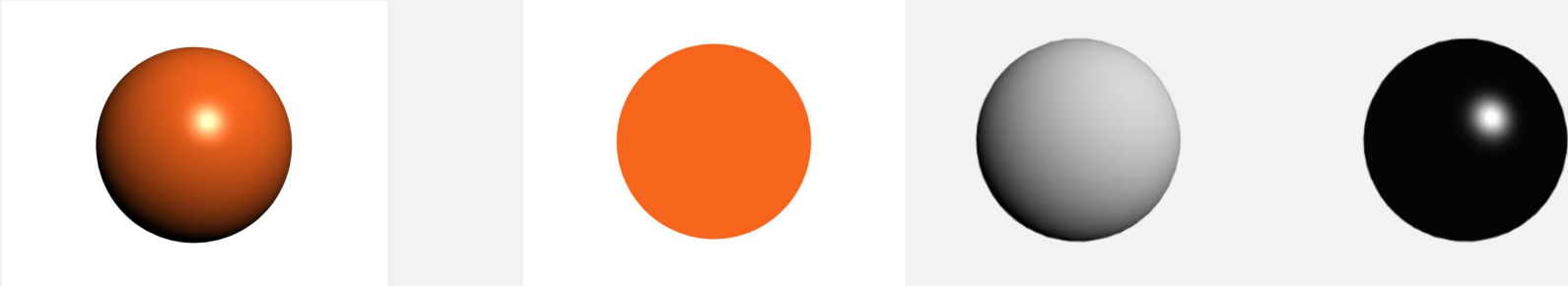
전통적인 조명 계산

• 자체 발광 성분.

• 주변광원에 따른 성분.

• 빛의 난반사에 의한 성분.

• 빛의 전반사에 의한 성분.

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


주변 반사(Ambient Reflection) 성분

■ 주변 반사 성분

- 주변광원에 대한 재질의 반응을 표현.
- 직접조명모델에서는 물체가 광원으로부터 직접 받는 빛만을 고려하기 때문에, 다른 물체에 반사된 빛을 처리하지 못함.
- 따라서 다른 물체를 통해서 반사된 빛을 근사적으로 간략하게 처리하기 위해 도입한 성분임.

$$I_a = K_a \otimes L_a$$



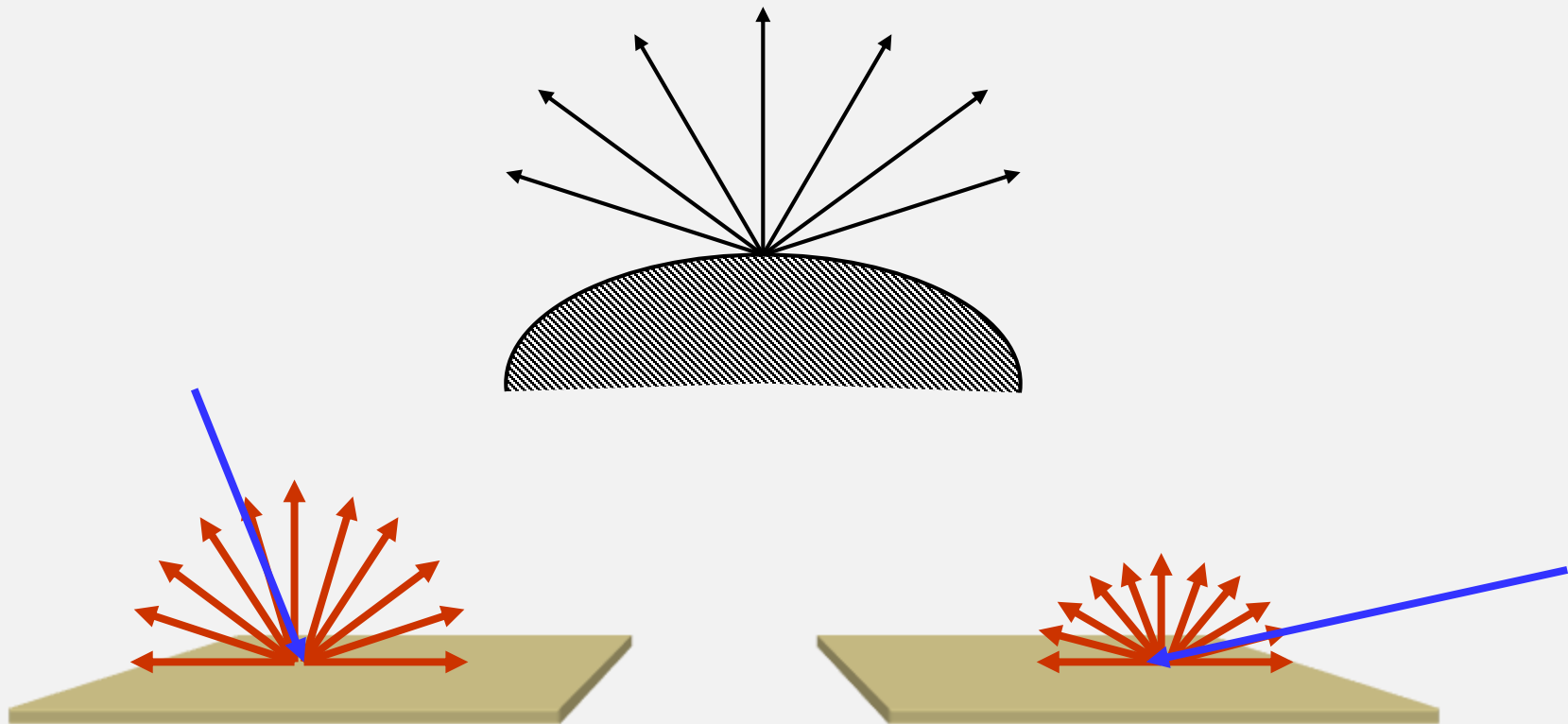
• 주변재질색상

- 광원 색상.
- `setAmbientLight()` 로 설정.

분산(Diffuse) 성분

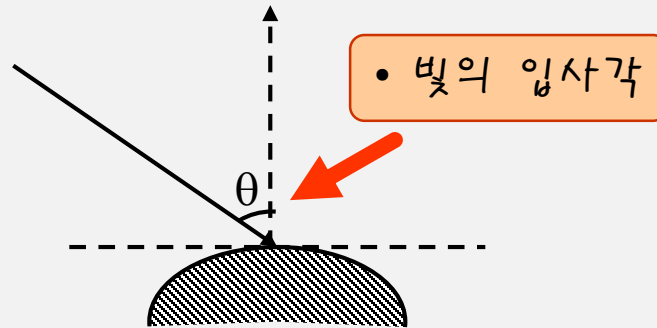
■ 분산 성분

- 광원으로부터 빛이 들어올 때, 모든 방향으로 똑같은 양을 반사시키는 완전한 난반사체를 모델링. → 재질의 거친 표면을 모델링.
- 관찰자의 위치에 상관없이 동일한 양의 빛이 반사됨.



■ 분산 성분의 계산

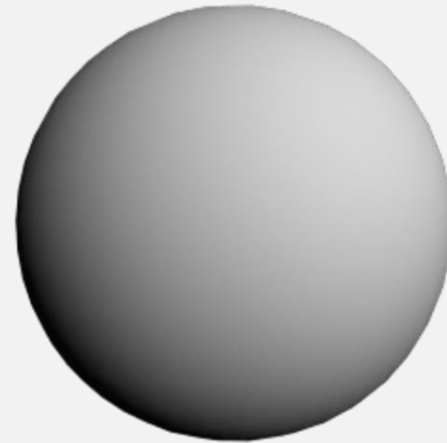
- 반사되는 빛의 세기는 빛의 입사각에 따라 달라짐.



$$I_d = (K_d \otimes L_d) \cos \theta$$

• 재질 색상

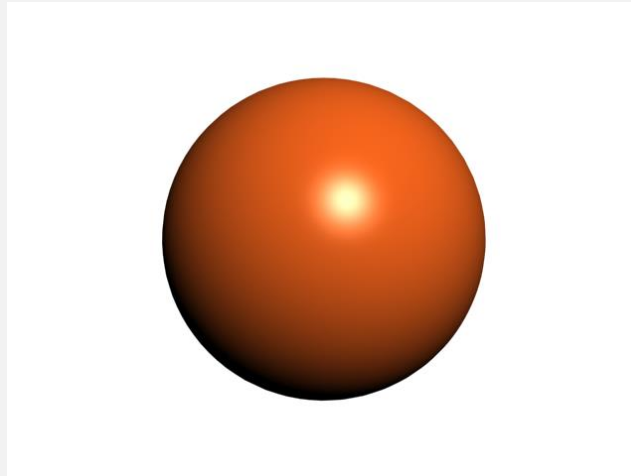
• 광원 색상.
• `setDiffuseColor()` 로 설정.



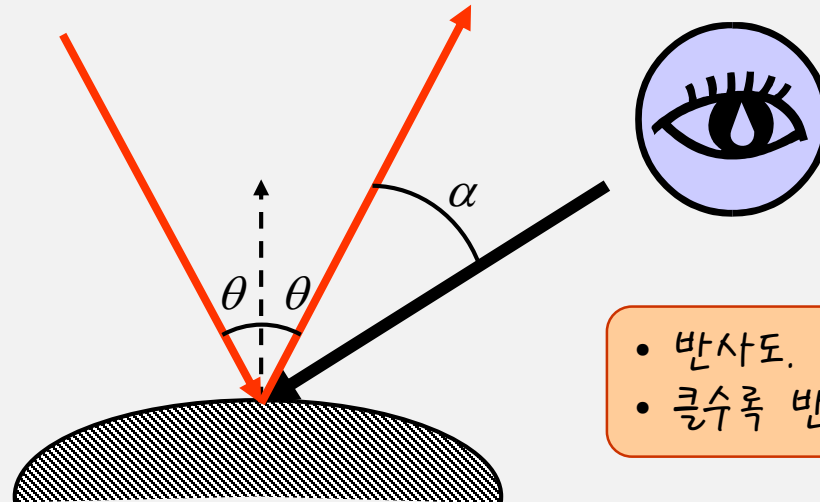
전반사(Specular) 성분

■ 전반사 성분

- 매끄러운 재질의 표면을 모델링.
- 관찰자의 시점의 따라서 반짝거리는 정도가 달라짐.
- 전반사 재질 색상과 반사도에 따라 영향을 받음.



■ 전반사 성분의 계산

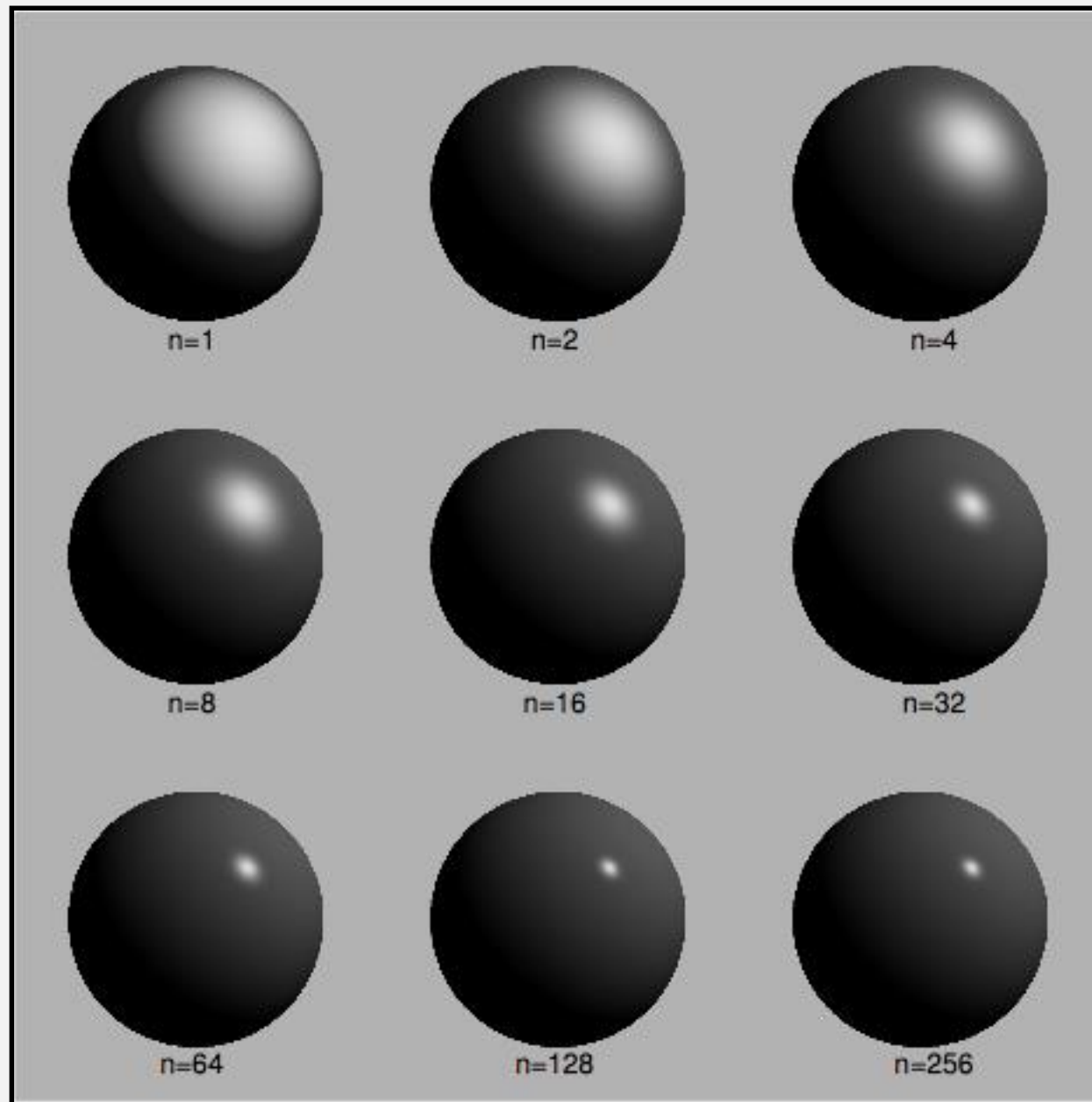


- 반사도.
- 클수록 반사 범위가 좁아짐.

$$I_s = (K_s \otimes L_s)(\cos \alpha)^n$$

- 재질 색상

- 광원 색상.
- `setSpecularColor()` 로 설정.



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

언리얼 엔진의 조명 계산

■ PBR(물리 기반 렌더링)

머티리얼 시스템의 "물리 기반" 측면에서 볼 때, 익혀둘 필요가 있는 속성은 딱 네 가지입니다:

- **Base Color** 베이스 컬러, 바탕색
- **Roughness** 러프니스, 거칠기
- **Metallic** 메탈릭, 금속성
- **Specular** 스펙큘러, 반사성

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