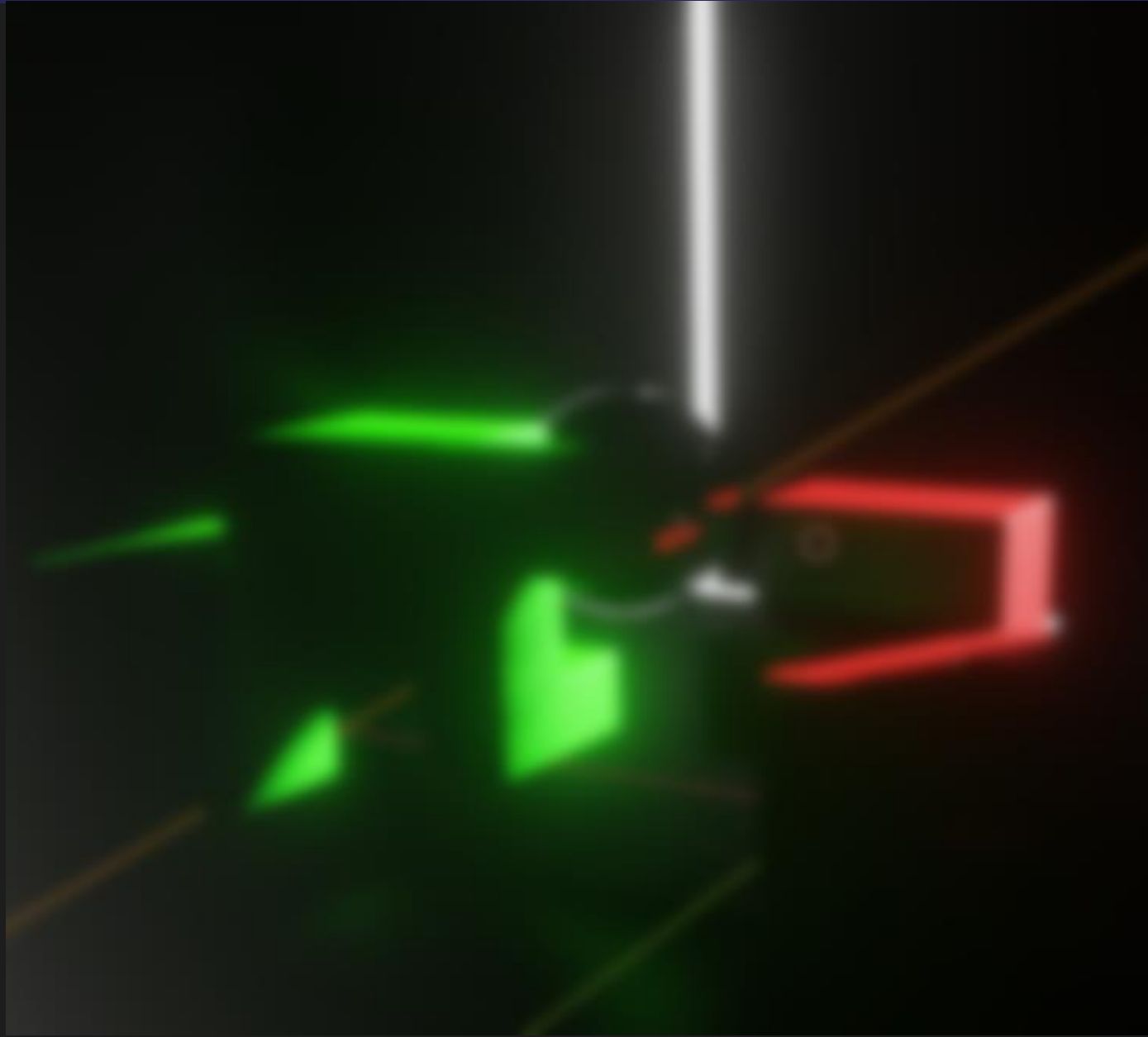


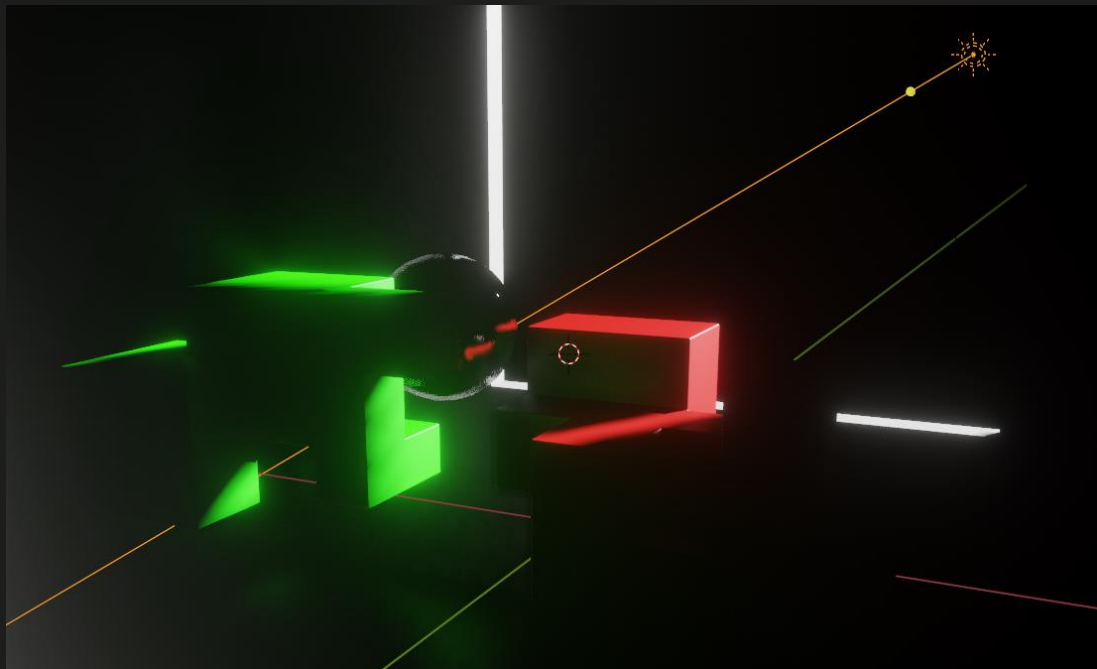
074강 Special Edition 2 : Eevee 렌더 엔진

Eevee 렌더 엔진 설정법



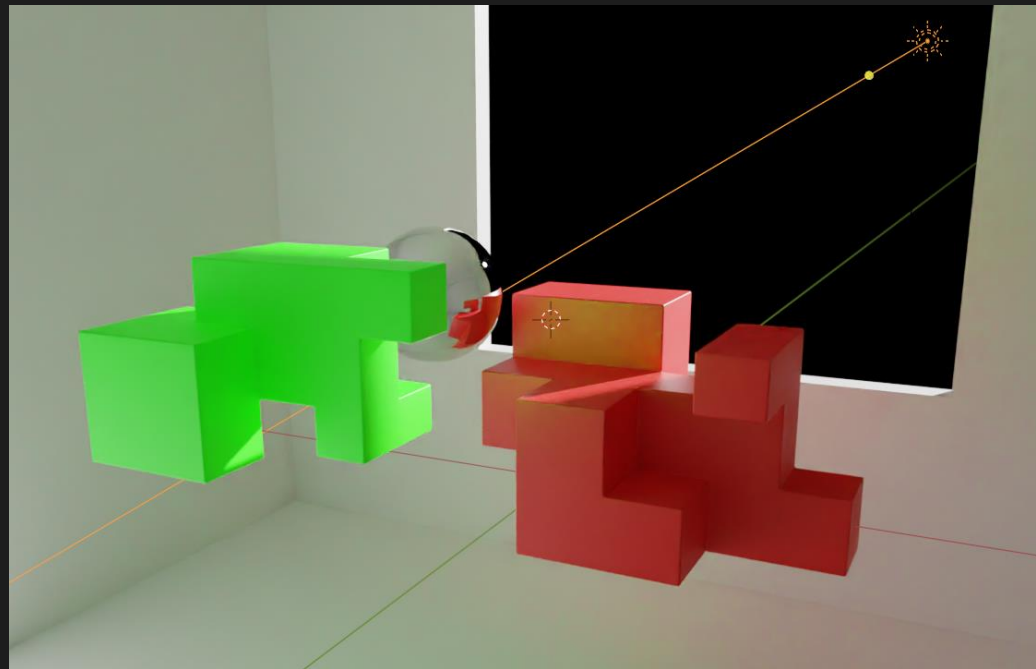
Eevee vs Cycles

빛의 반사를 실제로 구현하는 사이클과 다르게, 이브이는 각 물체끼리 상호작용을 하지 않습니다. 이브이가 상호작용 하는 것은 그림자와 배경이미지 뿐입니다.



Eevee

라이트 하나만 놓았을 때, 빛이 닿는 부분만 빛납니다.



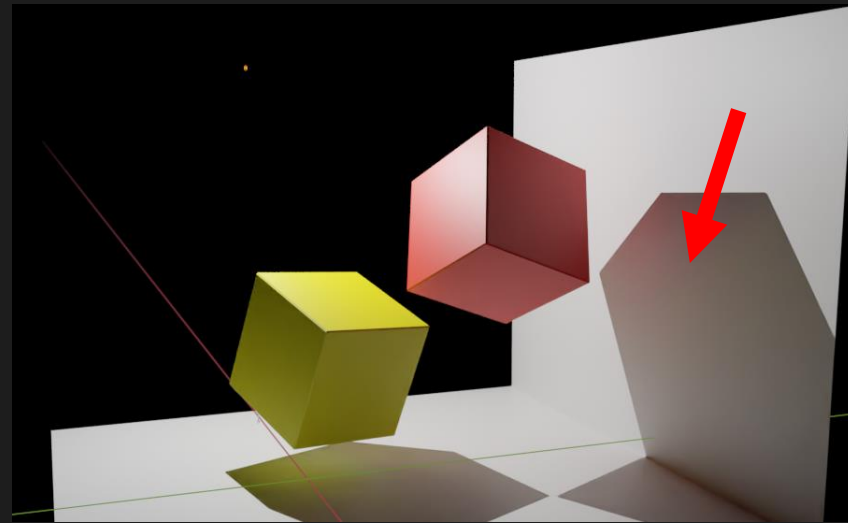
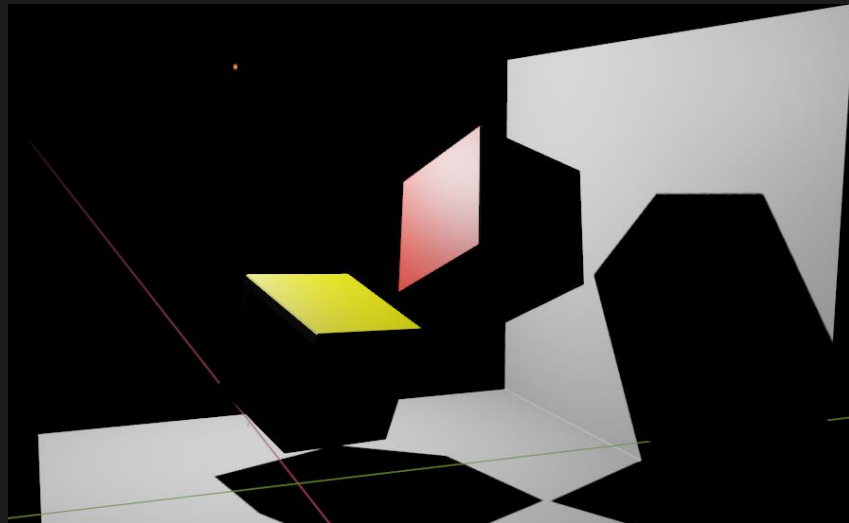
Cycles

들어온 빛이 사방으로 퍼져나가기 때문에, 조명 하나로도 실내를 밝힐 수 있습니다.

상호 작용의 두가지 측면

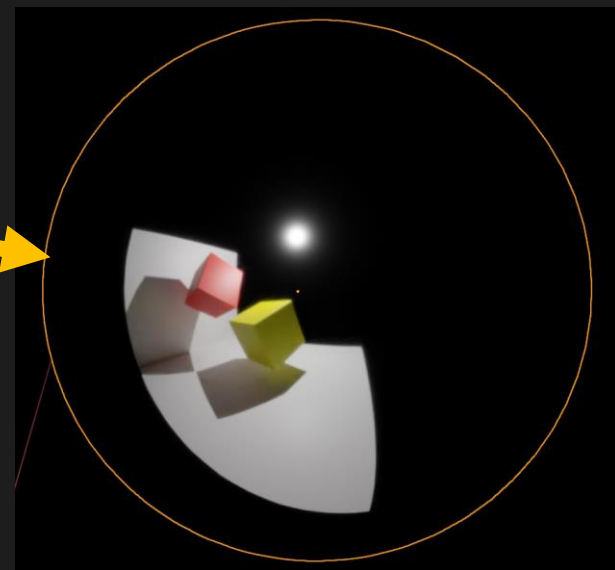
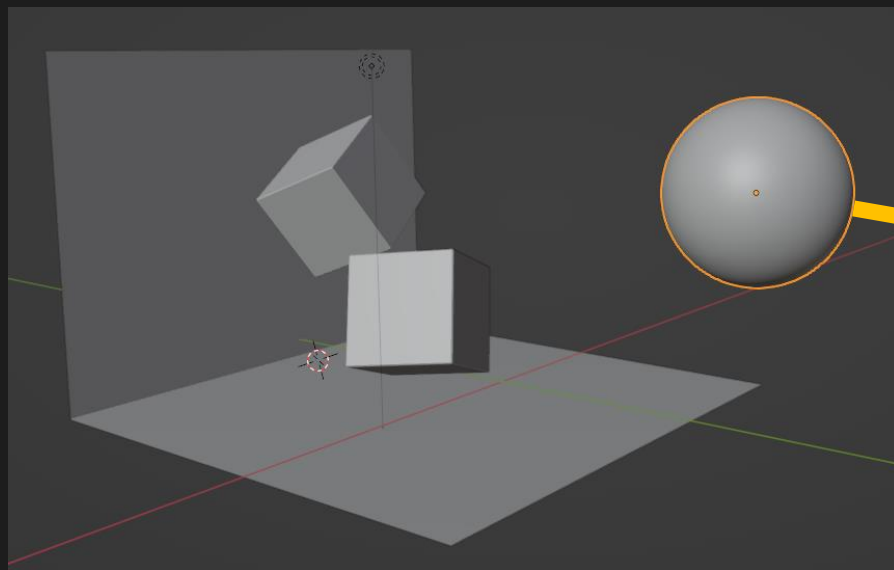
간접광 :

빛이 닿지 않는 부분도,
주변 물체의 빛을 받아 밝아집니다.
이때, 밝기 변화뿐만 아니라 주변 물체의
색깔도 따라옵니다.



반사 :

금속같이 매끈한 표면의 경우,
화면에 보이지 않는 물체도 반사되어
보일 수 있습니다.



Eevee가 가진 대안

HDRI

polyhaven.com 등에서, 현재 장면과 흡사한 HDR 이미지를 다운로드 받아 배경으로 사용합니다.

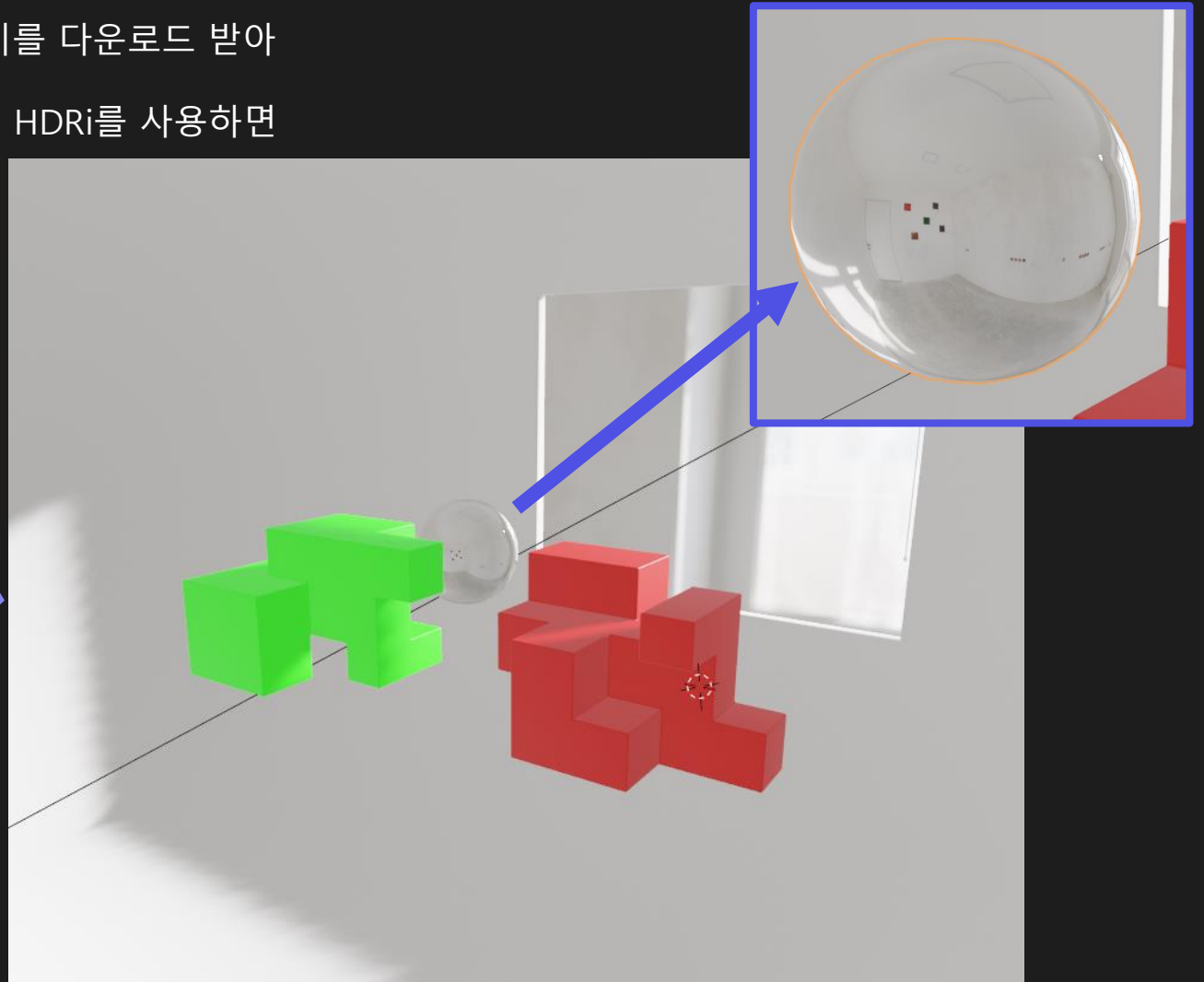
예컨대, 앞의 경우엔 하얀 벽의 실내이므로, 아래와 같은 HDRI를 사용하면 빛이 닿지 않는 부분에 이미지가 사용되어 밝아집니다.

창문 밖으로 보이는 배경은 따로 설정해 줍니다.

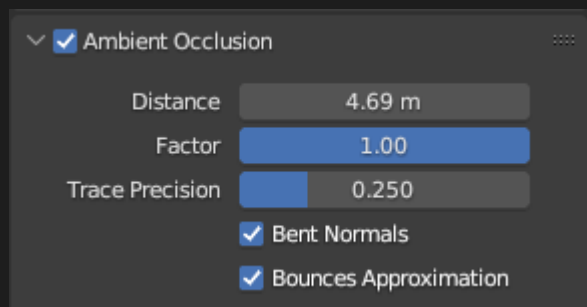


<https://polyhaven.com/hdris>

금속 구체에 반사되는건
실제 존재하는 흰색 벽이 아니라
HDRI의 흰색 벽입니다.



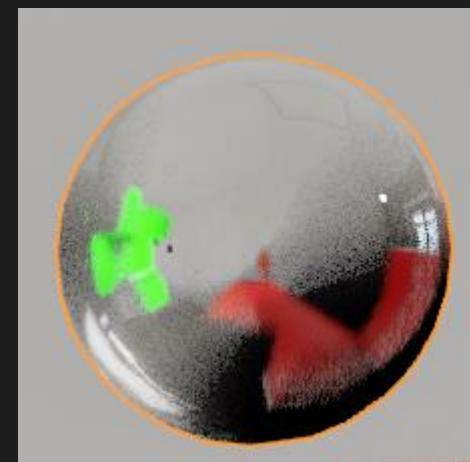
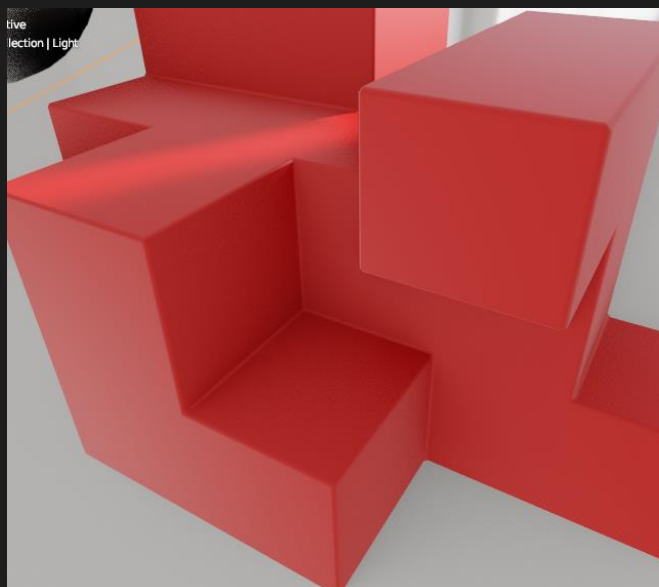
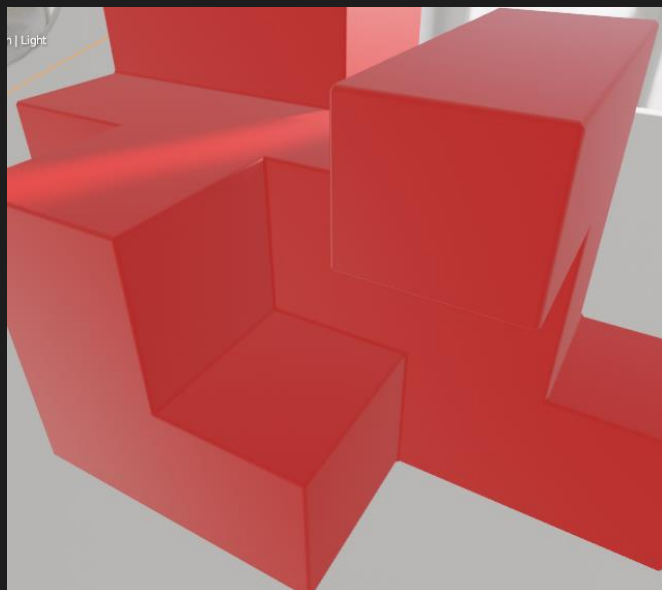
Eevee가 가진 대안



Ambient Occlusion :

구석진 곳 / 물체의 뒷부분 등 일반적으로 어두워지는 부분을 강조하여 입체감을 살려줍니다. 정확한 계산이 아니므로 사이클과 비교하면 상당한 차이가 있습니다.

많은 경우 Distance를 크게 올리는 쪽이 효과적입니다.



※ Roughness가 낮은 금속의 경우 Distance를 많이 올리면 이상하게 어두워질 수 있습니다. 불가피한 현상이므로, 적절한 거리를 맞추는 것이 중요합니다.

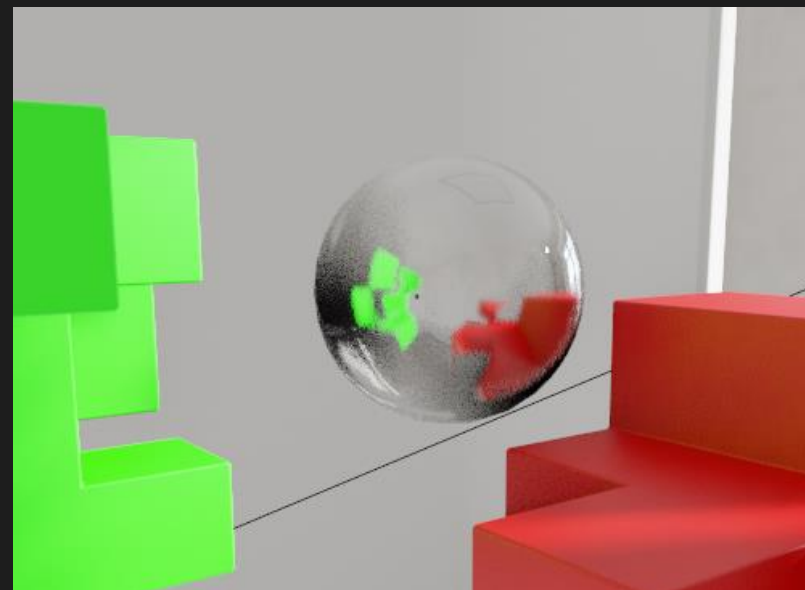
Eevee가 가진 대안



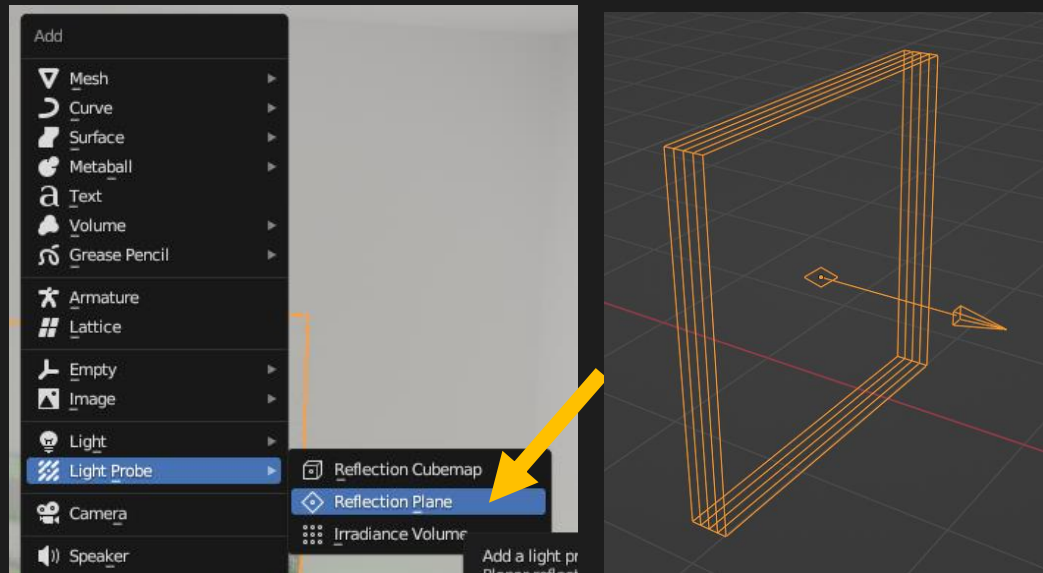
Screen Space Reflection :

화면상에 보이는 오브젝트를 바탕으로 정반사를 계산합니다.
화면에 보이지 않는 것은 반사되지 않지만, 어색하지 않을 만큼은 도움을 줍니다.

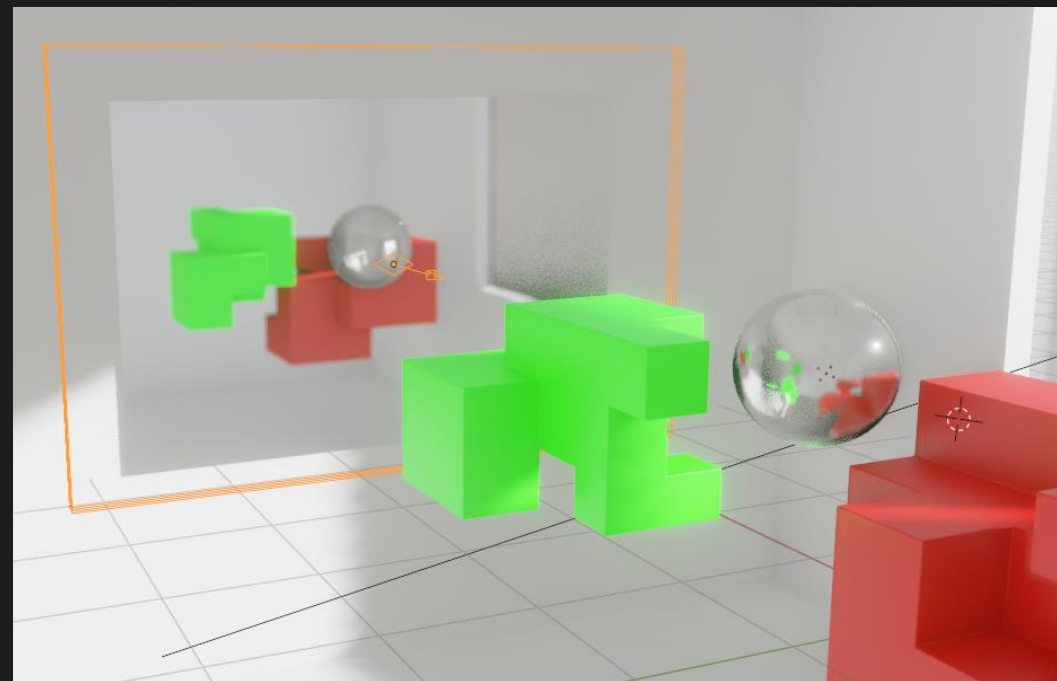
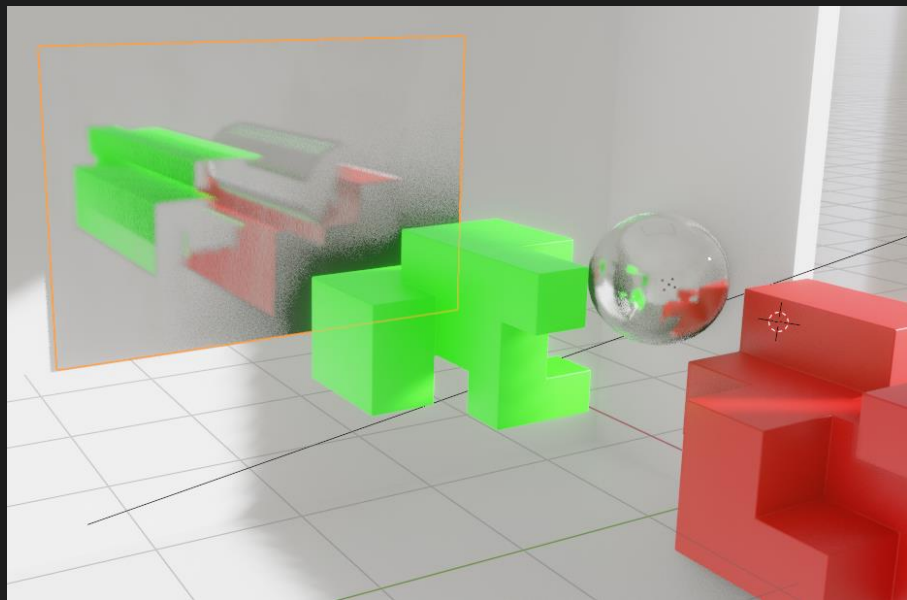
Half Res Trace : 계산을 빠르게 하기 위해 해상도를 절반으로 줄여 사용합니다.
따라서 **체크 해제**하는 편이 품질에 긍정적인 영향을 줍니다!



Reflection Plane

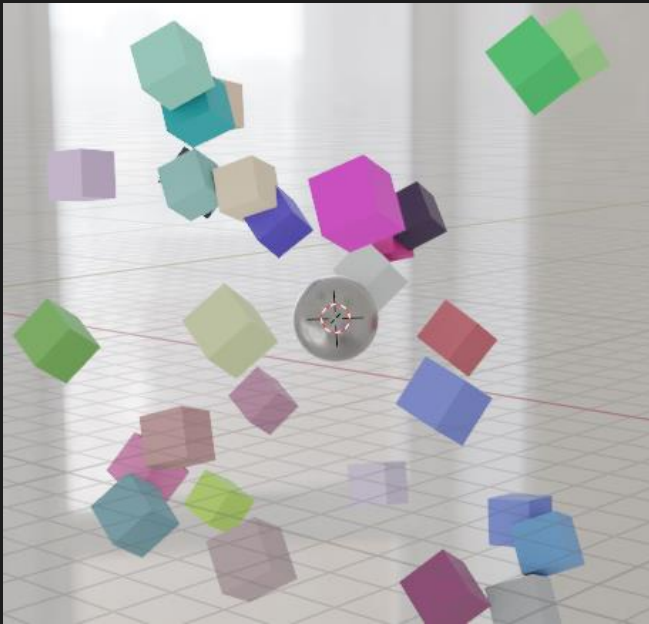


일반적인 경우 Screen Space Reflection으로도 충분하지만, 거울같이 깨끗이 반사되어야 하는 경우라면 곤란해집니다. 이 경우, Reflection Plane을 반사가 일어날 위치에 갖다놓으면 간단히 해결됩니다.

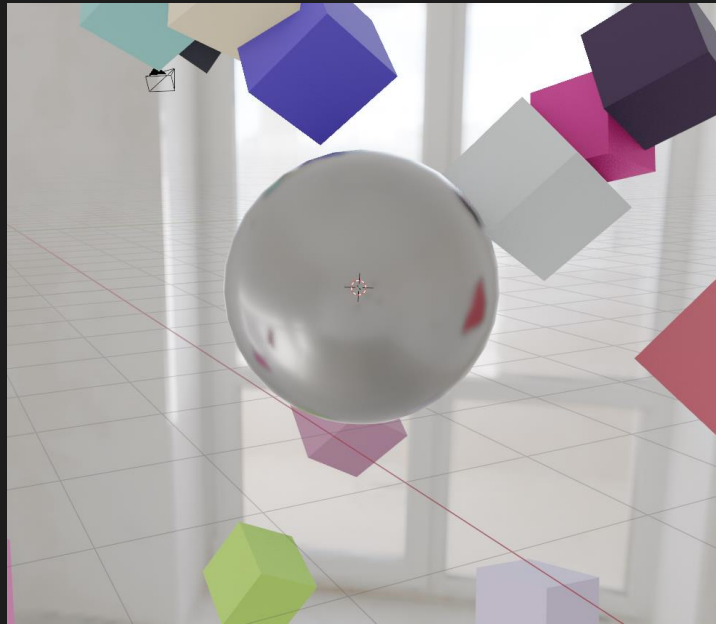


Indirect Lighting

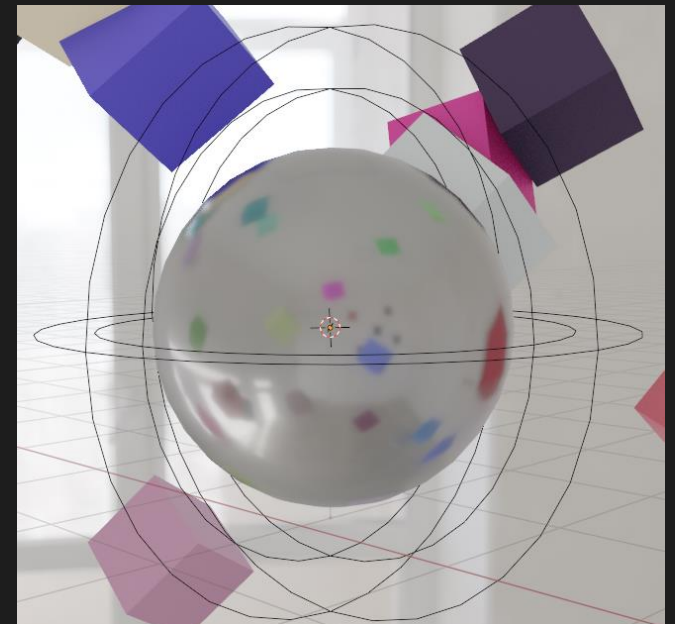
Reflection Cubemap : 오브젝트의 위치에서 일종의 'HDRi'를 미리 계산해서, 주변의 오브젝트에 적용합니다.



금속 주변에 큐브가 흩뿌려져 있습니다.

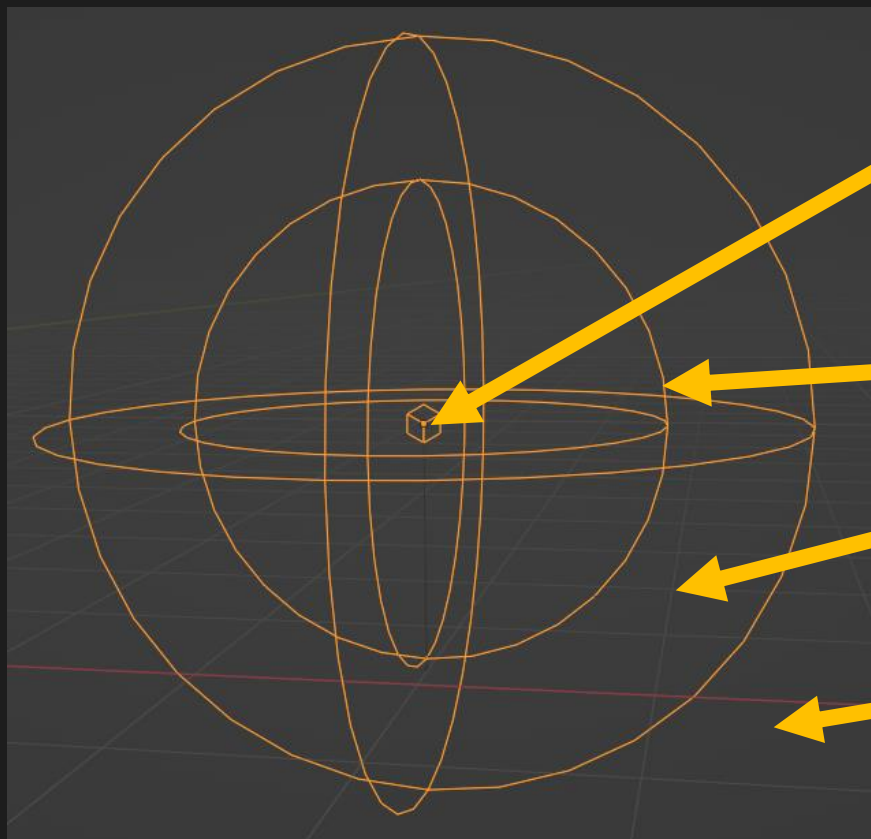


가까이 다가가면, 큐브가 화면에서 사라지므로 금속에도 비치지 않습니다.



Reflection Cubemap을 사용하면 화면 밖의 물체도 반사할 수 있습니다.

Indirect Lighting

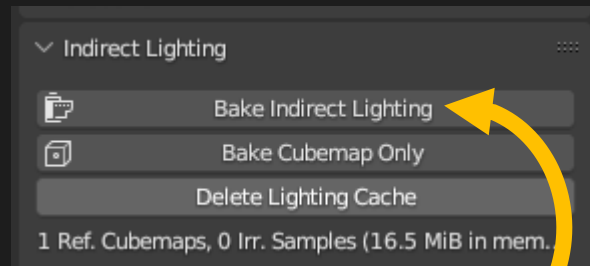


Reflection Cubemap은 중심 지점에서 사진을 찍습니다.
Reflection Plane과 다르게, 렌더 설정에서 미리 이미지를 저장해야 합니다.
※베이킹할 때, 금속 재질은 찍히지 않으니 숨겨두셔야 합니다.

첫번째 원 안쪽에 있는 오브젝트는,
Reflection Cubemap이 찍은 이미지를 배경이미지로 사용합니다.

첫번째와 두번째 원 사이에 있으면, Reflection Cubemap과 원래 배경이미지를
적당히 섞어 사용합니다.

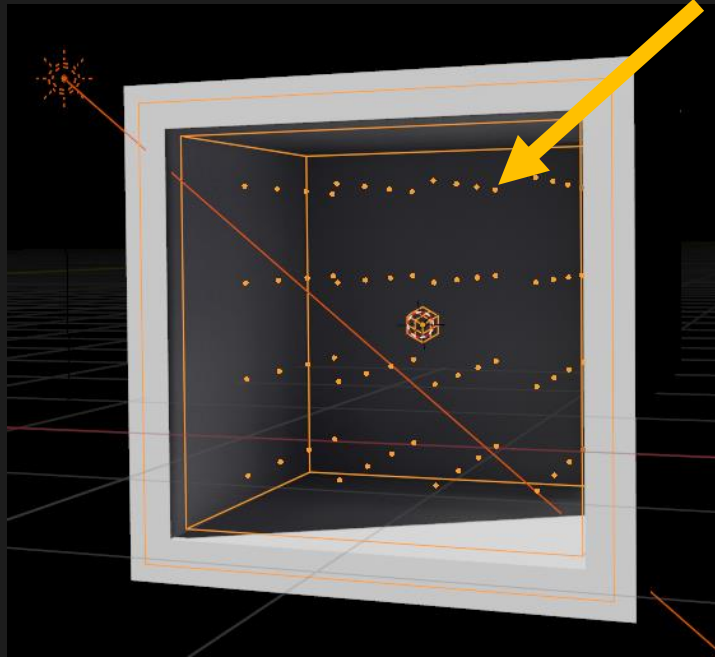
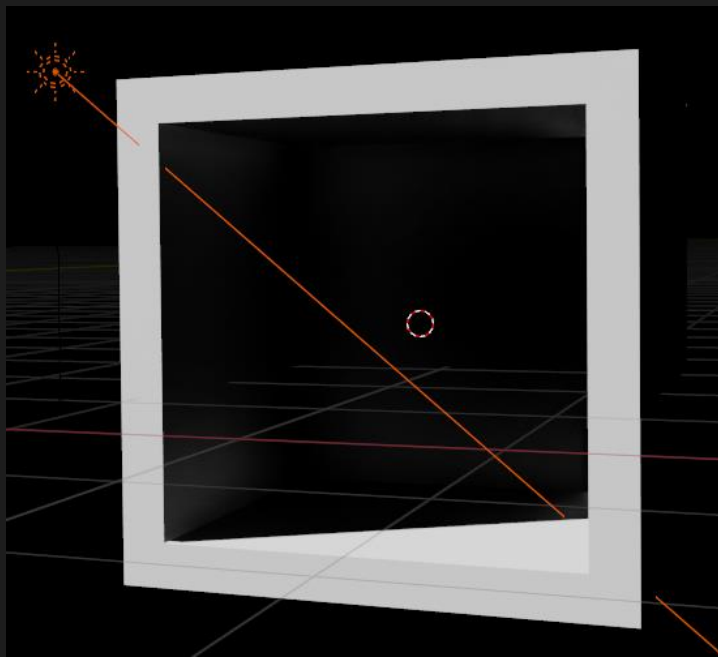
두번째 원 바깥에 있는 오브젝트는 Reflection Cubemap을 사용하지 않습니다.



Indirect Lighting

Irradiance Volume 을 사용하면 Reflection Cubemap과 비슷한 방식으로 간접광을 구현할 수 있습니다.

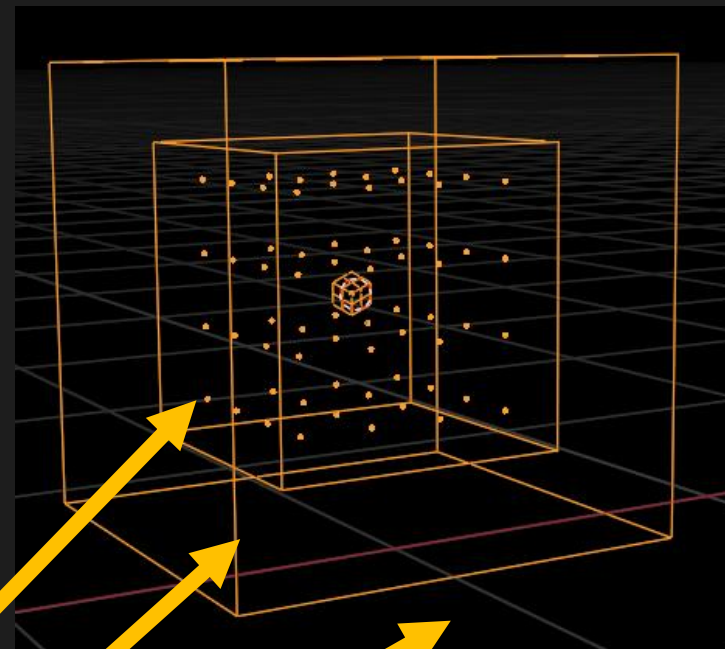
각 점의 위치에서 빛을 계산하여
그 주변에서 사용합니다.



Irradiance Volume의 점 주변에서는 그 점에서 계산한 밝기값을 사용합니다.

첫번째와 두번째 큐브 사이에서는 Irradiance Volume이 계산한 것과 원래 밝기를
적당히 섞어 사용합니다.

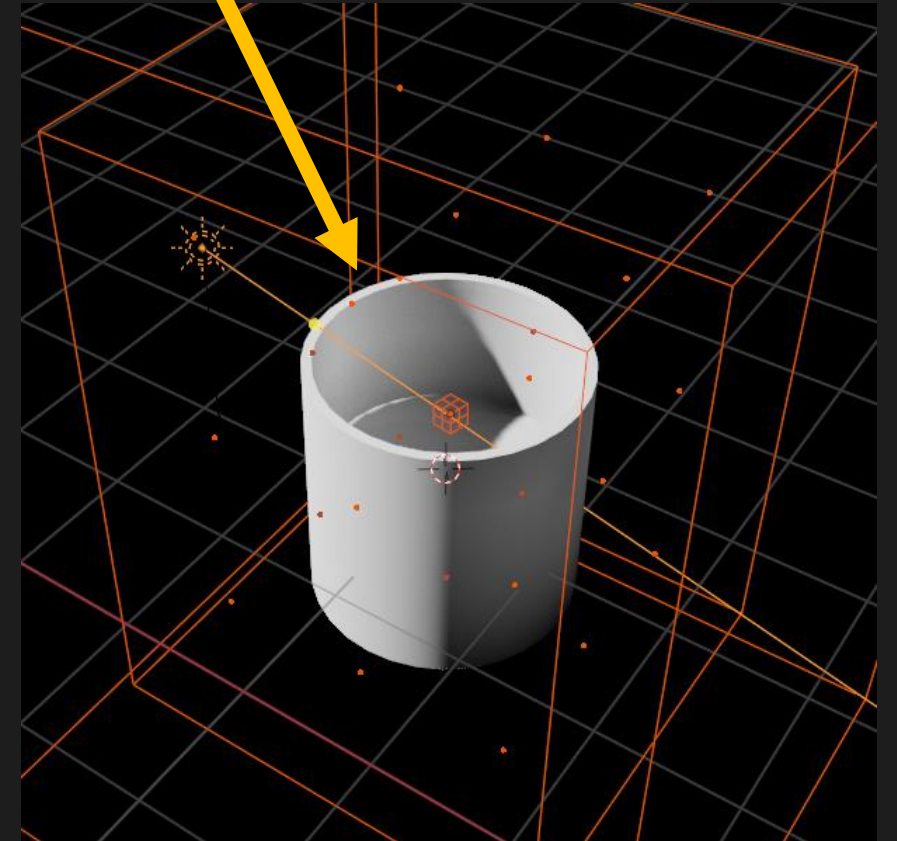
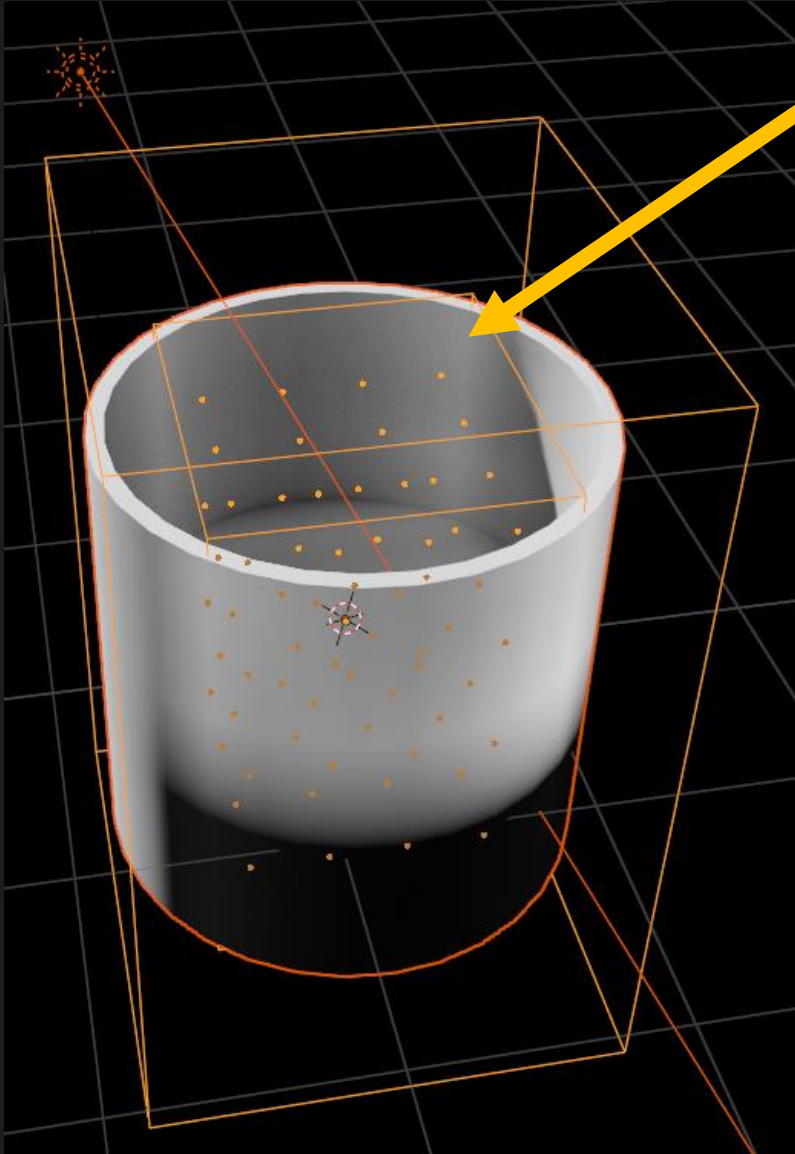
두번째 큐브 밖에서는 Irradiance Volume을 사용하지 않습니다.



Indirect Lighting

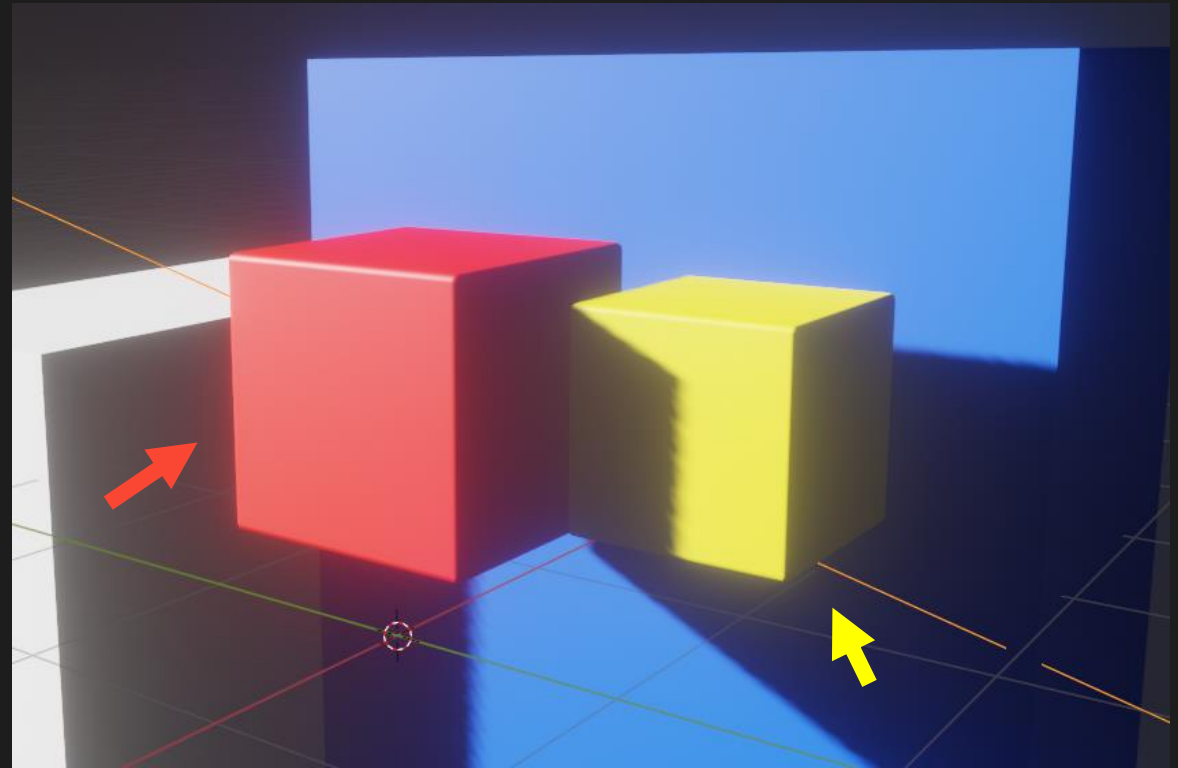
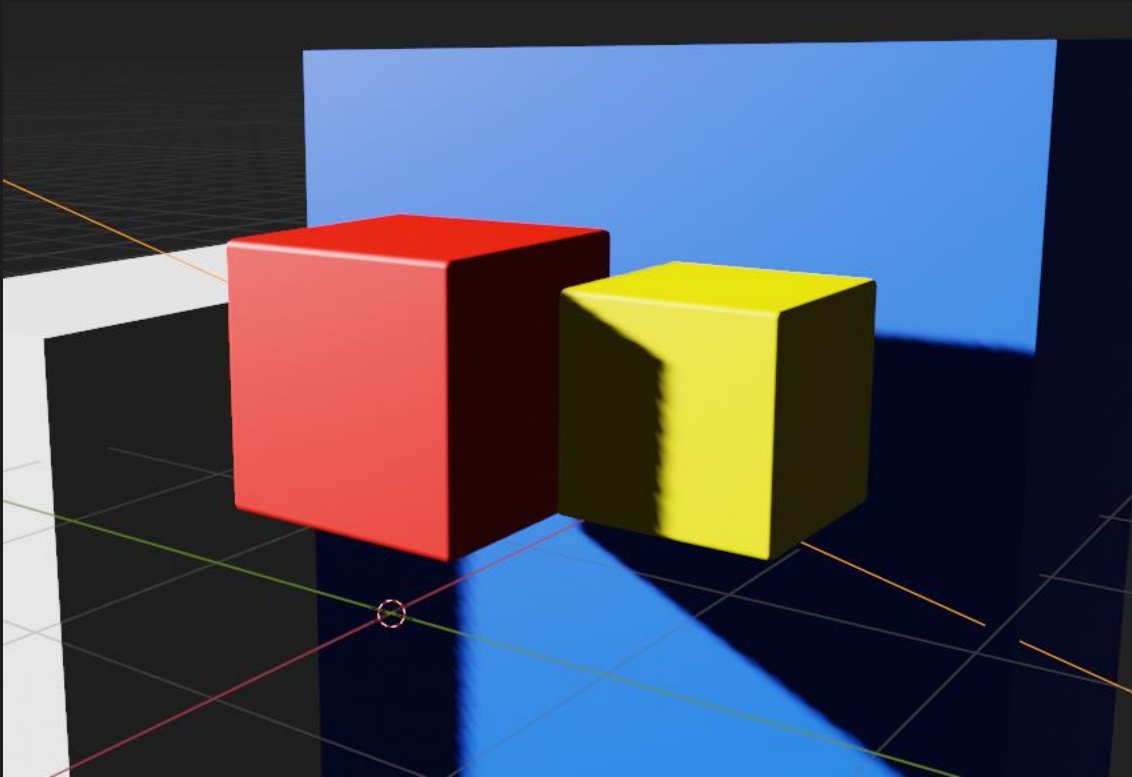
Irradiance Volume은 그 방식상, 상자 형태가 아니거나, 두께가 얇고 복잡한 형태의 경우 이상한 그림자를 만들게 됩니다. 생각보다 사용이 편리하지 않아서 저는 잘 쓰지 않는 편입니다.

해상도(점 개수)를 늘리는 것은 한계가 있으므로, 차라리 해상도를 많이 낮춰서 부드러운 변화를 노리는 것이 나을 수도 있습니다.



Bloom

Bloom의 원래 목적은 빛나는 물체의 색번짐을 구현하는 것이지만, 물체끼리 색이 섞이므로, 마치 빛이 반사되어 비친 것처럼 보이게 만들 수 있습니다. 우측 하단 이미지는 다른 기능을 모두 끄고 Bloom만 사용한 것입니다. 빨간색과 노란색 큐브의 색이 번지는 것을 확인할 수 있습니다. 물론 그 방식상 뿌옇게 보이는 단점이 있으므로, 효과적인 값을 찾는 것이 중요합니다.



유리

Eevee 에서 유리는 뒷면에서 굴절이 일어나지 않습니다.

결과는 정확하지 않으며, 특히 면이 이중으로 된 경우 (컵 같은 것) 설득력이 떨어집니다.

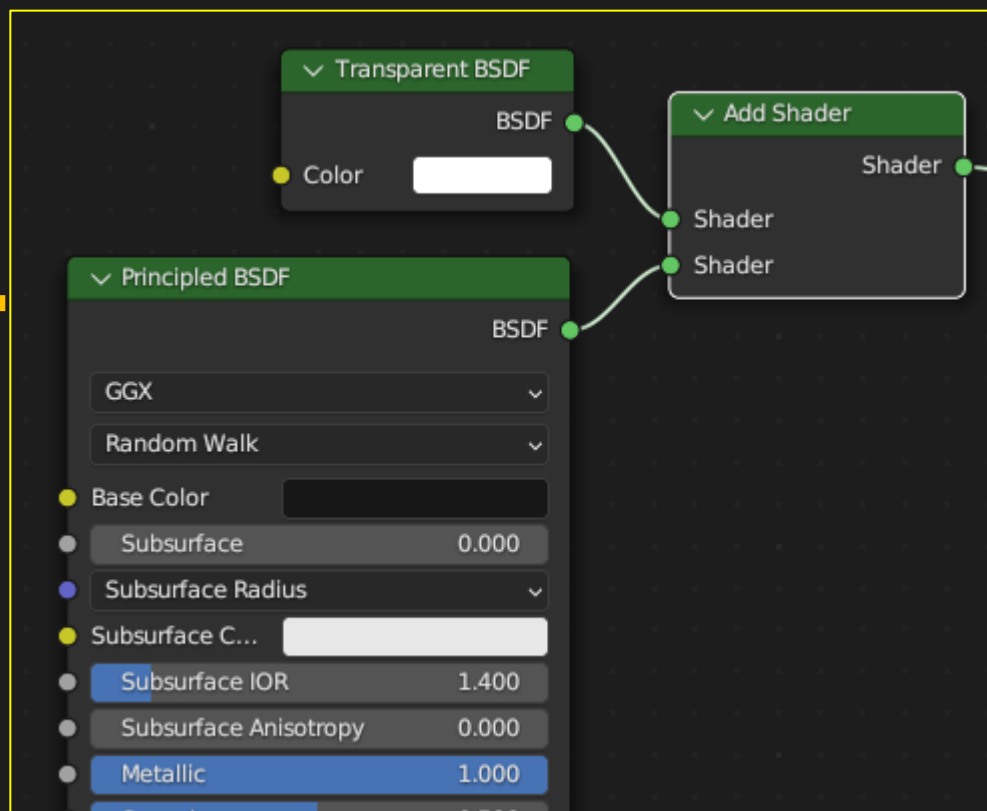
이 경우 굴절 효과를 포기하고 반사와 투과 효과만 사용하는 것이 효과적일 수 있습니다. (11강 참고)



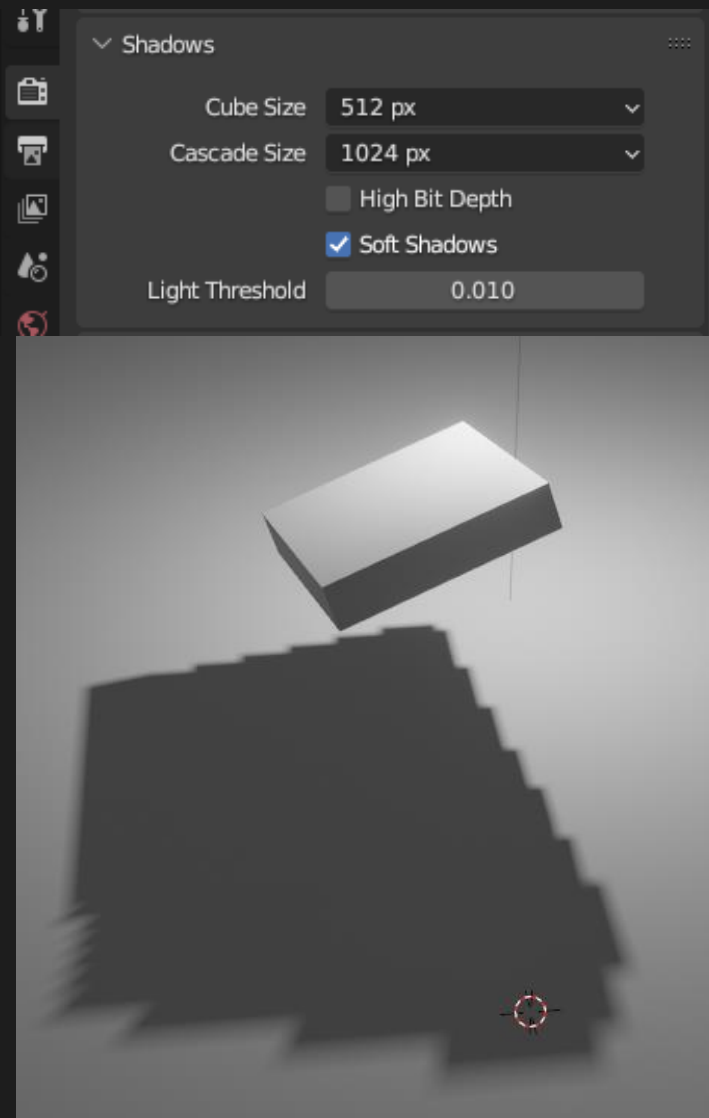
왼쪽 위 : 일반적인 Eevee유리.

왼쪽 아래 : 유리로 창문을 만들면 의도치 않게 왜곡됩니다.

오른쪽 위/아래 : 11강처럼 금속에 투명을 더한 것.



그림자



Eevee의 그림자는 오브젝트의 외곽선을 추적하지 않으며, 유한한 방향의 그림자를 계산합니다. 따라서 그림자의 형태는 정확하지 않으며, 깨끗한 그림자를 얻기 위해서는 연산량을 늘려야 합니다.

Cube Size : 일반적인 Point/Spot/Area light의 그림자 해상도를 조절합니다.

Cascade Size : Sun Light가 만드는 그림자 해상도를 조절합니다.

High Bit Depth : 그림자 계산에 더 높은 비트를 사용합니다. 정확도가 더 올라갑니다.

Soft Shadows : 연산을 반복해서 왼쪽과 같은 각진 그림자를 부드럽게 만듭니다.

Eevee의 그림자는 물체가 겹치는 경계에서 정확하지 않은 특성이 있습니다. 조명 설정에서 Contact Shadows를 켜 주시면 조금 나아집니다.

