**Computer Graphics PA4**

name : Lee Jaehee

student id : 20215160

github id : jaehee831

* **Indirect lighting**

**달이(가) 표시된 사진

자동 생성된 설명공, 구체이(가) 표시된 사진

자동 생성된 설명**

**원, 다채로움, 그래픽이(가) 표시된 사진

자동 생성된 설명공, 구체이(가) 표시된 사진

자동 생성된 설명**

scene 1,2에서 indirect lighting 전/후 모습

* + theorem

indirect lighting은 광선이 표면과 여러 번 반사되면서 간접적으로 장면의 다른 부분에서 조명을 수집하는 경로를 나타낸다. 이 과정은 ray tracing 알고리즘에서 중요한 역할을 하며, 현실적인 조명 효과를 재현하는 데 기여한다. 위의 사진과 같이 indirect lighting을 적용한 후, 부드럽게 퍼지는 그림자와 근처 객체들로부터 색상이 번지는 효과를 관찰할 수 있다.

* + how to implement

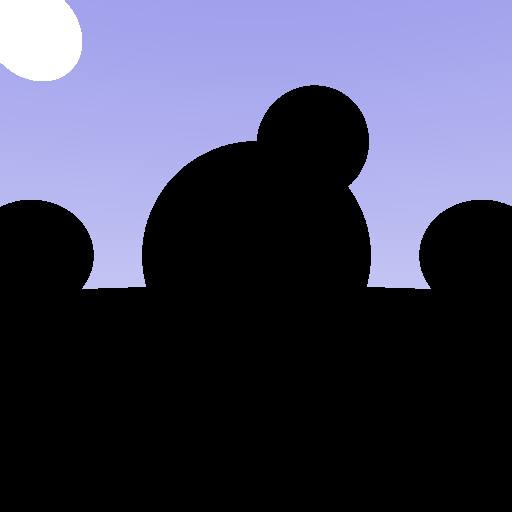
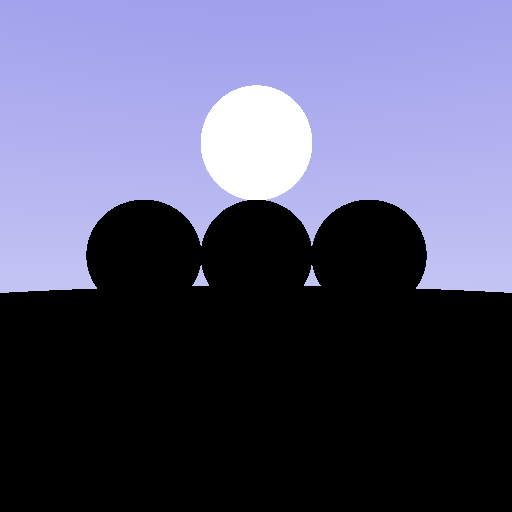
텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

위 코드에서는 광선이 표면에 닿을 때마다 반사(ray scattering)와 색 감쇠(color attenuation)를 계산하여 누적 조명을 갱신한다. 최대 10번의 반사를 통해 무한 반사를 방지하며, **scene->rayIntersect**와 **surfaceMaterial->scatter**함수로 각 반사 및 광선-표면 상호작용을 처리한다. 광선이 더 이상 반사되지 않거나 최대 깊이에 도달하면 누적된 조명을 반환한다.

* **Material**

Area light



* + theorem

Area light은 점광원(Point light)과 다르게 특정 면적에서 빛을 방출하는 광원이다. 실제 세상에서는 대부분의 빛이 면적을 가지고 방출되므로, 현실감 있는 조명을 구현하기 위해 자주 사용된다.

* + how to implement

텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

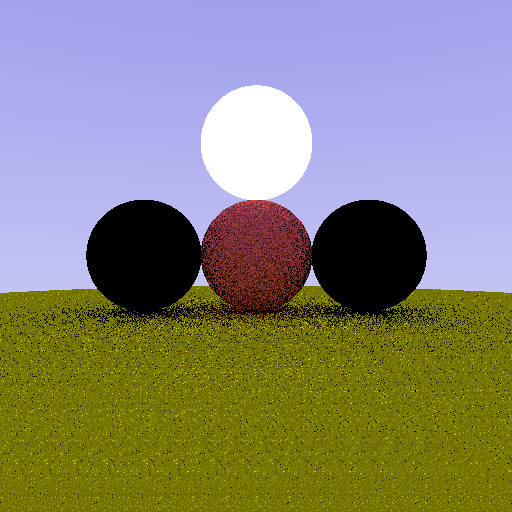
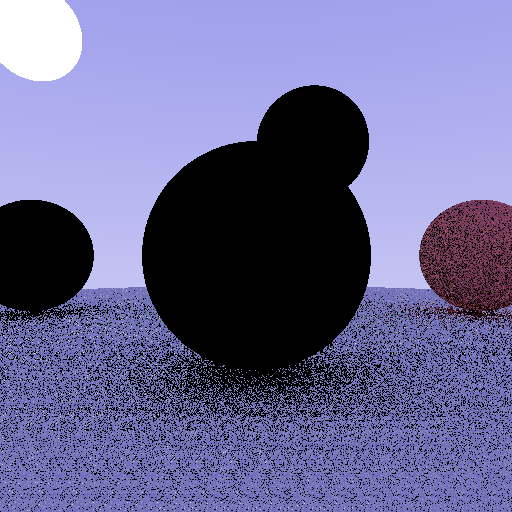
자동 생성된 설명

**DiffuseLight::scatter** -> 광선이 물체에 맞았을 때 어떻게 반사(혹은 산란)되는지를 정의한다. Area light의 경우 빛을 반사하지 않고 단순히 빛을 방출하기 때문에 false를 반환한다.

**DiffuseLight::emitted** -> 광원이 방출하는 빛의 색상과 강도를 정의한다. 이 예제에서는 흰색 빛을 방출하도록 설정되어 있으며, 실제로는 이 값을 다른 색상으로 변경하여 다양한 색상의 빛을 구현할 수 있다.

**DiffuseLight::attenuation** -> 빛의 감쇠를 정의한다. 감쇠는 빛이 거리에 따라 약해지는 정도를 나타내며, Area light에서는 특별한 감쇠를 정의하지 않으므로 0.0f를 반환한다.

Lambertian



* + theorem

Lambertian 반사는 물체 표면의 모든 방향으로 균일하게 빛을 반사하는 모델이다. 반사된 빛의 강도는 광선이 표면에 닿는 각도에 따라 달라지며, 이는 Lambert's Cosine Law에 의해 설명된다.

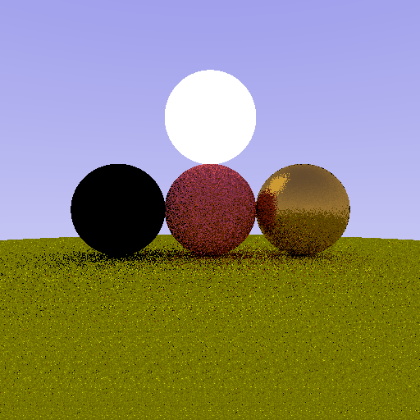
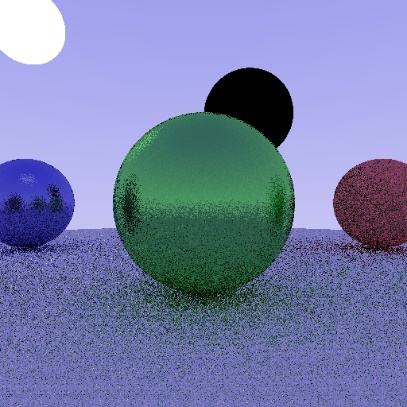
* + how to implement

텍스트, 스크린샷, 소프트웨어, 운영 체제이(가) 표시된 사진

자동 생성된 설명

**DiffuseLight::scatter** -> 광선(**r\_in**)이 물체에 충돌했을 때 새로운 반사 광선(**r\_out**)을 생성한다. **hit.n**은 충돌 지점의 법선 벡터이다. **glm::sphericalRand(1.0f)**는 단위 구의 임의의 점을 생성하여 법선 벡터에 더한다. 이를 통해 랜덤한 방향으로 광선을 산란시킨다. 또한 산란 방향의 길이가 매우 작은 경우, 법선 벡터를 사용하여 산란 방향을 설정한다. 이는 산란 방향이 유효하지 않을 때(예: 매우 작은 벡터) 발생하는 문제를 방지하기 위함이다. 이후 충돌 지점 hit.p에서 새로운 산란 방향 **scatter\_direction**으로 광선을 생성한다.

Metal



* + theorem

Metal Material은 광선이 표면에 닿았을 때 반사되는 성질을 가진다. 이 재질은 일반적으로 광선을 표면 법선에 대해 반사시키며, 표면의 거칠기에 따라 반사 광선의 분산 정도가 달라진다. 반사된 광선이 다시 표면과 상호작용하며, 이를 통해 간접 조명 경로를 형성한다.

* + how to implement

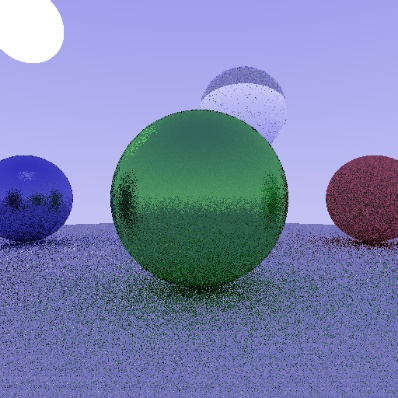
텍스트, 전자제품, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

scatter 함수에서, **reflected**는 입사 광선을 표면 법선에 대해 반사하여 계산된다. scatter\_direction은 반사된 방향에 재질의 거칠기(roughness\_)와 무작위 벡터를 더하여 계산된다.

attenuation 함수에서 금속 재질의 색(color\_)을 반환해 감쇠를 계산한다. 금속 재질이 광선을 반사하고, 거칠기에 따라 반사 방향이 변동되는 효과를 구현함으로써 금속 표면의 반사 특성을 재현했다.

Dielectric

 **공, 하늘, 잔디, 구체이(가) 표시된 사진

자동 생성된 설명**

* + theorem

Dielectric Material은 빛이 표면을 통과할 때 반사와 굴절을 모두 고려한다. 이 재질은 빛이 두 매질 사이의 경계에서 어떻게 행동하는지 설명하며, Snell 법칙과 Fresnel 방정식을 따른다. Snell의 법칙은 빛이 매질 경계를 통과할 때의 굴절각을 설명하고, Fresnel 방정식은 입사각에 따라 반사와 굴절되는 빛의 비율을 결정한다. 유전체 재질에서는 모든 반사 및 굴절이 에너지 보존 법칙을 따른다.

* + how to implement

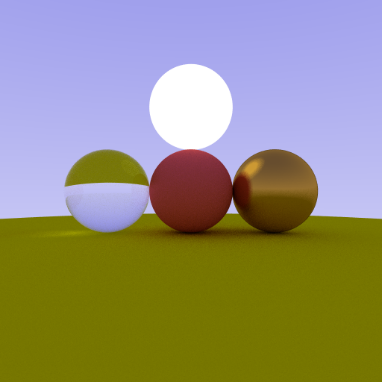
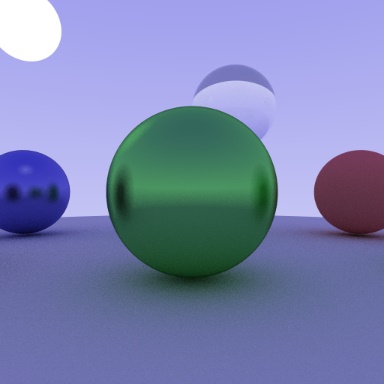
**텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명**

위 코드는 유전체 재질의 광선 산란(scattering)을 구현한 것이다. **outward\_normal**은 광선이 표면에서 반사되는 방향을 계산한다. **reflected**변수는 입사 광선을 표면 법선에 대해 반사하여 계산된다.

굴절 계산을 위한 변수를 초기화하고, 입사 광선이 표면을 통과하는 경우와 반사되는 경우에 대해 법선 방향과 굴절률 비율을 설정한다. 이후 random 값과 반사 확률을 비교하여 반사 광선 또는 굴절 광선을 결정한다.

* **Antialiasing**

****

* + theorem

렌더링 시 이미지의 계단 현상을 줄이기 위해 사용되는 기술이다. 이 현상은 해상도에 비해 샘플링 밀도가 낮을 때 발생하며, 경계선이 매끄럽지 않고 들쭉날쭉하게 보이는 현상을 말한다. antialiasing은 픽셀의 여러 샘플을 사용하여 평균을 내어 경계선을 부드럽게 만드는 방법으로 이를 해결한다.

* + how to implement

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

안티앨리어싱을 위해 각 픽셀에 대해 여러 번의 샘플을 생성하여 색상을 계산한다. for (int s = 0; s < spp; ++s) 루프가 각 픽셀에 대해 spp번 반복되며, 무작위 위치에서 샘플을 생성한다.

dis(gen)을 사용하여 무작위 오프셋을 추가하여 샘플링 위치를 결정한다. 이를 통해 픽셀 내에서 다양한 위치를 샘플링하여 평균 색상을 계산할 수 있다.

**Direct light sampling**

**공, 구체이(가) 표시된 사진

자동 생성된 설명공, 하늘, 구체이(가) 표시된 사진

자동 생성된 설명**

* + theorem

직접 조명 샘플링(Direct Light Sampling)은 장면의 광원에서 오는 직접적인 빛의 기여를 계산하는 과정이다. 이는 광선이 광원과 교차하는 지점에서 샘플링하여 광원의 빛이 표면에 어떻게 영향을 미치는지 평가한다. 몬테카를로 적분(Monte Carlo Integration)을 사용하여 여러 샘플을 통해 빛의 기여를 평균화하여 더 정확한 조명 효과를 얻는다.

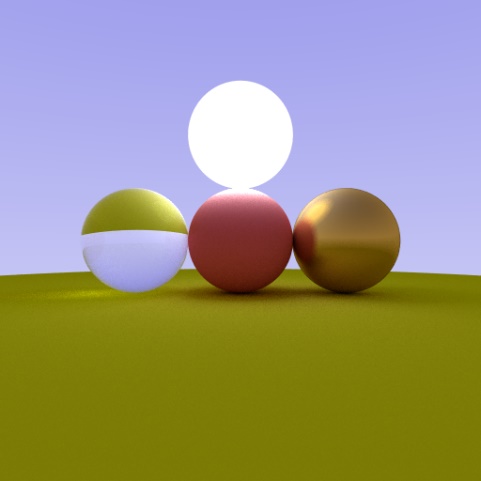
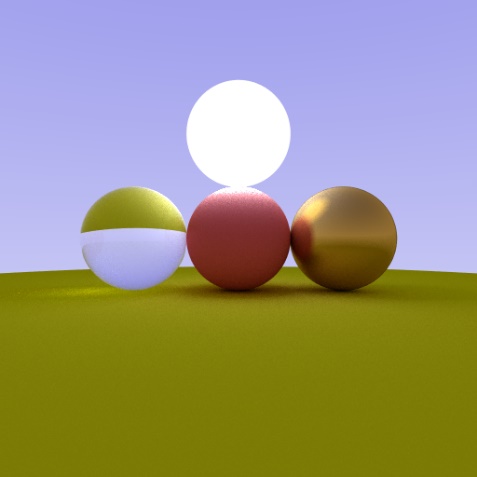
* + how to implement

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

**randomPointOnSphere** 함수로 랜덤하게 광원의 표면에서 샘플링을 수행한다. 선택한 점에서 광선을 hit 지점으로 향하게 만든다. 만들어진 광선으로 **shadowRay**를 생성하며, shadowRay가 다른 객체에 의해 가려지지 않고, **cosTheta**(표면 법선과 광원의 방향 벡터 사이의 각도 cos 값을 계산해 표면의 조명 기여도를 결정)와 **distSquared**(거리 제곱 값을 계산하여 거리 감쇠를 적용)가 양수이면 **lightResult**에 조명의 기여를 누적한다.

코드의 **sampleCount** 변수를 설정함으로써 샘플링 횟수를 정할 수 있다. 이를 증가시키면 이미지의 노이즈가 줄어드는데, 그만큼 계산 비용이 증가하는 문제가 있다. 또한 장면에 따라 최적의 샘플링 횟수가 다르므로 적절한 수를 찾아야 한다. 아래 사진은 sampleCount가 10일 때(좌측 이미지)와 50일 때(우측 이미지)를 비교한 모습이다. 빨간 공이 area light을 받는 부분에서 특히 우측 이미지의 노이즈가 감소된 것을 확인할 수 있다.



* **Conclusion & Future Work**

지금까지 다양한 물성의 구체를 렌더링하며 직접 조명과 간접 조명 샘플링의 효과, 안티앨리어싱 효과를 직접 적용해보았다. 렌더링 이미지의 퀄리티를 더욱 높이기 위해서는 적절한 유저 파라미터의 설정, 효율적인 샘플링 방식 등이 필요해 보인다. 기회가 된다면 다양한 방법으로 시도해보고자 한다.