

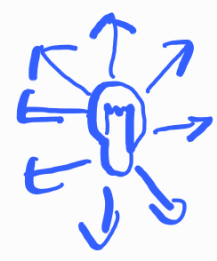
조명(Light)

Directional Light



direction (unit vector)

Point Light



position
fallOfStart, End

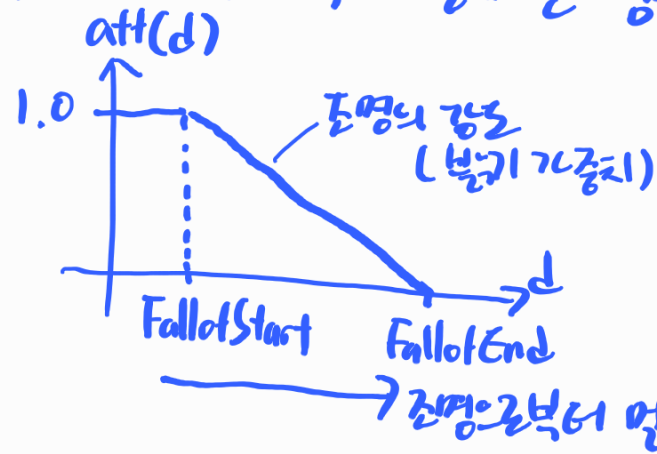
Spot Light



direction, position
fallOfStart, End, spotPower

FallOfStart & FallOfEnd

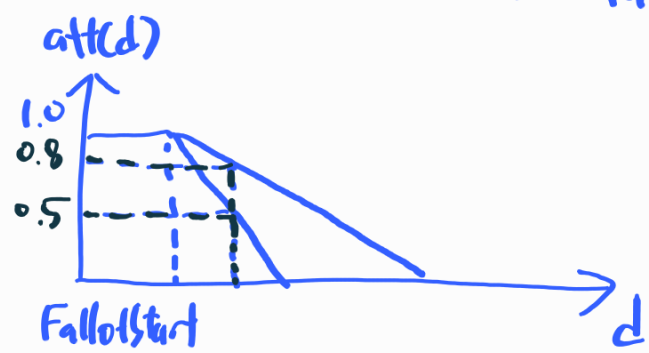
물체를 조명으로부터 멀어지면 점점 어두워진다. (천천히 밝은)



$d(\text{거리}) = \text{렌더링 할 Point와 조명의 거리}$

조명으로부터 멀어진다.

동일한 거리 d에 대해서 fallOfEnd가 더 큰 경우 att(d)가 더 크다



더 멀리 빛을 쏘아줄수록
빛의 강도가 감소하므로 같은 거리이면
더 밝게 빛날 것이다.

Point Light와 Spot Light은 너무 멀리 있으면 ($d > \text{fallOfEnd}$)
조명이 적용되지 않으므로 렌더링하지 않습니다. (그림자와 비슷함)

```
float Saturate(float x) {
```

```
    return max(0.0f, min(1.0f, x));
```

```
} // 셰이더에서 많이 사용되는 함수다. HLSL에 내장되어 있다.
```

x 를 $0 \leq x \leq 1$ 로 범위를 고정한다. (clamp)한다고

```
float CalcAttenuation(float d, float fallOffStart, float fallOffEnd) {
```

```
    // Linear Fall off
```

```
    return Saturate((fallOffEnd - d) / (fallOffEnd - fallOffStart));
```

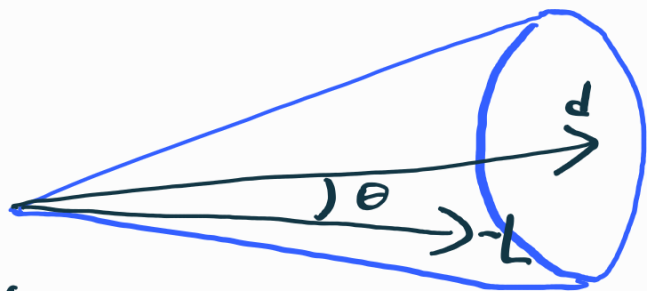
```
} // 조명에 강도 계산을 위해 사용한다.
```

lightStrength \times CalcAttenuation(렌더링 포인트에서 조명까지의 거리,
L.fallOffStart, L.fallOffEnd);

Point Light은 Raytracing에서와 같이 렌더링할 시점과 조명의 위치를 사용하여 Light의 Vector를 계산하고 FallOff를 사용해 렌더링한다.

Spot Light은 Point Light과 같은 빛 계산이지만 위치 계산을 한 번 더 해준다

$$\text{LightVector} = \text{Light.position} - \text{Point}$$



조명의 중심에서 빛이 날 수 있는 범위이다.

```
float SpotFactor = pow(
```

```
    max(dot(L.direction, -lightvector), 0.0f),  
    L.spotPower);
```

lightStrength = lightStrength \times SpotFactor