

컴퓨터 그래픽스 과제 #2

1. Lighting 이해와 응용을 위한 과제입니다.
2. 제출은 kedilms 페이지의 과제 제출 시스템을 사용하여 제출하십시오.
 - ① 제출 파일은 shader.vert/shader.frag 파일입니다. 이외 다른 파일을 수정하였을 경우 사유와 함께 수정한 모든 파일을 제출해주세요.
 - ② 제출 파일은 표절 검사 프로그램을 거칩니다. 표절로 판단될 경우 0점 처리합니다.
3. 첨부한 예제 프로그램을 주의 깊게 테스트 해 보시고, 동일하게 동작하는지 확인 후 제출하십시오.
4. 제출 기한: 시스템에 제출 시간이 자동으로 기록됩니다. **11월 25일 23:55 분 이전까지 제출하셔야 합니다.**
 - ① 늦게 제출하실 경우 1시간당 1점씩 감점합니다.
 - ② 시스템에 기록된 시간은 업로드 완료 시점입니다. 감안하시고 여유 있게 제출해주세요.
5. 기타 문의사항은 이메일(hk.kim@jbnu.ac.kr)로 연락 바랍니다.

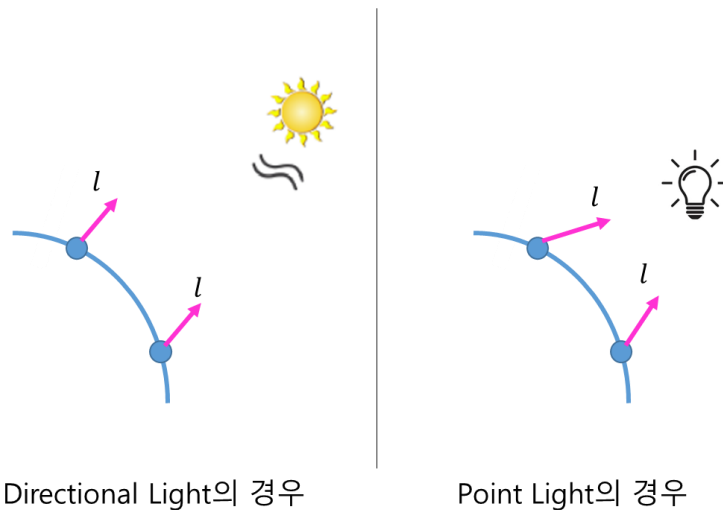
채점 기준

- ① 요구 조건대로 프로그램 정상 동작 (10점)
 - ◆ 표준적인 조건에서 채점 가능하도록 shader에 이미 정의된 변수명은 변경하지 말 것.

문제

- Directional Light에 대해 배운 내용을 바탕으로 Point Light를 사용하는 Lighting Shader를 구현하시오.
- 첨부한 Assignment2 Project 코드를 참고하여, shader.vert 및 shader.frag를 수정하여 구현한다.
- Point Light와 물체의 상호작용 방식의 특징은 아래와 같다.

1. 아래 그림과 같이, Directional Light의 경우 빛을 향하는 벡터 l 이 fragment의 위치와 관계없이 동일한 벡터이나, Point Light는 per-fragment별로 다르다.



2. Point Light의 경우 근접 광원이기 때문에, 거리에 비례하여 빛이 감쇄 (attenuation)하는 효과를 고려하여 색상을 계산한다.
3. 감쇄 정도는 아래와 같다.

$$\text{Attenuated Color} = \frac{\text{color}}{\text{attenuationFactor} * r^2}$$

where r is a distance between fragment and point light source

- Assignment2 Project에 포함된 fragment shader는, Lighting Project에서부터 Point Light 구현을 위해 약간의 수정을 가하였다.
 1. Point Light 관련 정보를 입력받기 위한 PointLight 구조체와 pointLights uniform을 추가함
 - ◆ PointLight는 광원의 color, ambientIntensity, diffuseIntensity 이외에 광원의 위치(position)와 광원의 감쇄 정도(attenuationFactor)를 멤버로 갖는다.
 2. Directional Light에 의한 색상 계산 로직을 CalcDirectionalLight로 분리함
 - ◆ color, ambientIntensity, diffuseIntensity에 의한 색상 계산 방법은 directional

light와 point light에서 동일함

3. Texturing으로부터 얻어지는 Md, Ma, Ms는 Directional Light와 Point Light에서 모두 동일하므로 최종 색상 계산에서만 곱해지도록 함
- 구현이 올바르게 이루어졌는지 확인하기 위해 프로그램의 Main 함수에 아래와 같은 기능을 추가함. 동작 결과는 함께 첨부한 예제 프로그램에서 확인 가능함.
 1. 화살표 키를 이용해 point light의 위치 변경
 2. 페이지 업/다운 키를 이용해 point light의 attenuation 정도 변경