



# 멀티라이팅

사 업 명 : 2022년 경기과학고등학교 정보과학세미나 1  
제 작 자 : 박건호(3학년)  
지 도 교 사 : 채상미

## 개요

프로세싱 등 많은 것을 해본 사람으로서 2D화면에서는 명암을 구현하기 너무 어려워 단색으로만 프로그램을 디자인해야한다는 아쉬운 경험이 있었다. 특히 프로그램에서 명암을 넣는다는 것을 고민하는 것보다 사진을 봤을 때 너무 멋있어서 예쁘게 디자인하고 싶다는 사람도 많을 것이라 생각한다. 프로그램에서 조작하여 이미지를 만든다는 개념보다는 이미지를 실제로 불러와서 조명을 반영한 이미지를 내보내는 프로그램을 만들 것이다. 물론 그냥 색깔만 툭 건드리는 단순한 수식이 아닌 lambert 조명 기법이라는 유명한 조명 기법이다. 이 조명 기법은 그냥 모든 프로그램에 쓰인다. obj mtl 이라는 파일 모두 ambient diffuse color를 쓰는 lambert 기법을 기반으로 할 정도로 유명하다.

## 목표 및 필요성

목표는 위에서 설명했듯이 어떠한 이미지에 조명을 추가하여 그 이미지를 저장하는 것이다. 이 프로젝트를 통해 우리가 원할 때 사진의 조명을 많이 추가할 수 있을 것이다. 물론 조명을 여러개 추가했을 때 각각의 조명에 대한 특성이 모두 나타나도록 하는 것이 중요한 점이 될 것이다. 개요에서 말한 프로그램의 측면에서는 프로그램으로 이를 만드는 것은 불가능하지 않다. 물론 api에 있을 정도로 유명한 조명 기법이지만 api를 제공해도 다루기 어렵다. 그냥 일반적인 프로젝트와 다르게 이 프로젝트는 무엇보다도 lambert light를 사람들이 쉽게 사용할 수 있게 하는 것이 매우 중요하다.

## Flowchart

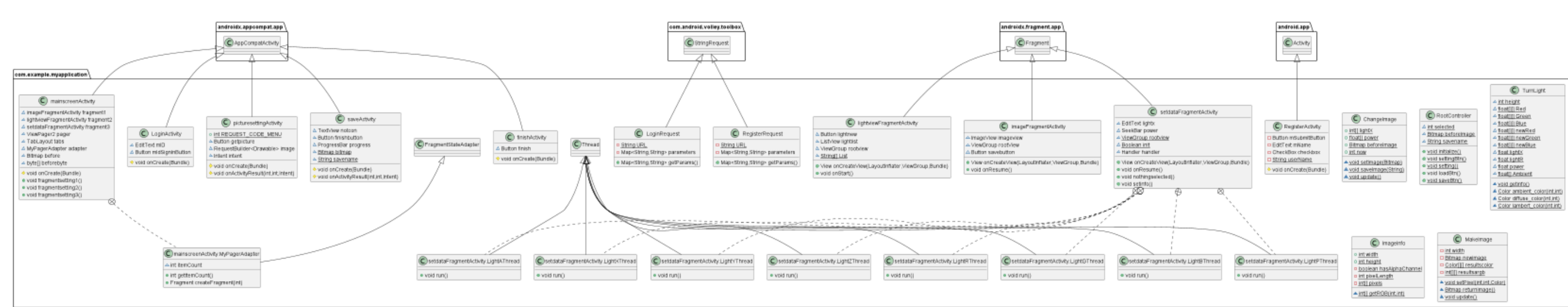


Fig. 1: uml of project

## 기능

빛의 효과를 받은 물체의 색깔은 전체적으로 3가지 모습의 합쳐진 모습으로 나타난다.



Fig. 2: ambient color, diffuse color, specular color의 효과를 보여준다. 여기서 ambient color는 평균적인 색상을 의미하는데 전체적으로 어두워 잘 보이지 않지만 잘보면 평균적인 색상을 나타낸다는 것을 알 수 있다.

- ▶ 전체적인 평균 색상을 알려주는 ambient 색깔
- ▶ 물체의 조명의 위치에 따라 부분마다 다른 색깔이 나타날 것이다. 이에 대한 효과는 diffuse 색깔로 나타난다.
- ▶ 그리고 마지막으로 그 물체의 특수한 재질때문에 생기는 효과로 인한 색깔이 더해진다. 제일 대표적인 예시로 광택이 있으며 이를 specular 색깔이라고 한다.

각 ambient 색깔, diffuse 색깔, specular 색깔은 각각  $K_a$ ,  $K_d$ ,  $K_s$ 라 하는 ambient 반사도, diffuse 반사도, specular 반사도에 영향을 받는다.

이에 대한 식은 다음과 같다.

1. ambient 색깔만 반영하면 최종 색깔은 다음과 같은 식으로 나타난다.

$$ambientcolor = materialcolor * ambient,$$

2. diffuse 색깔까지 반영하면 최종 색깔은 다음과 같은 식으로 나타난다.

$$lightvector = lightposition - objectposition$$
$$cosine = \text{dotproduct}(objectnormalvector(0), normalizedlightvector)$$

## 기능

$$lambertfactor = \max(\cosine, 0)$$

$$luminosity = \frac{1}{1 + distance * distance}$$

$$diffusecolor = materialcolor * lightcolor * lambertfactor * luminosity$$

$$finalcolor = ambientcolor + diffusecolor$$

3. specular 색깔까지 반영하면 최종 색깔은 다음과 같은 식으로 나타난다.

$$lightvector = lightposition - objectposition$$

$$lightReflected = -lightvector.x, -lightvector.y, lightvector.z$$

$$spec = \text{pow}(\max(\text{dot}(\text{viewDir}, lightReflected), 0), specularpower)$$

$$specularcolor = K_s * spec * lightcolor * objectcolor$$

$$finalcolor = ambientcolor + diffusecolor + specularpower$$

실제로 위와 같은 식을 재귀적으로 만들어서 앱을 만들어 실행해보면 이와같은 다중조명이 나타난 것을 알 수 있다.



Fig. 3: before



Fig. 4: after

## 기대효과 및 활용 방안

이 과정을 통해 우리는 원하는 수 만큼의 독립된 조명을 설치 할 수 있게 되며 이런 것들은 사진 편집 프로그램이 익숙하지 않을 때 대신 사용할 수 있는 편리함을 제공한다. 그리고 다중 조명을 한번에 놓게 해줌으로서 차별점을 보이기도 한다

## 결론

우리는 멀티라이팅이라는 앱을 통해 어떤 사진에 조명을 넣을 수 있게 된다. 물론 다중조명이라는 점 자체 만으로 다른 앱과 차별화되지만 spotlight처럼 일정 범위에 더 빛을 강하게 쬔어주는 기능이나 mtl 파일에 있는 재질도 모두 생각하여 이 조명 모델을 시각화한다면 더 발전된 앱이 될 수 있을거라 생각한다. 그리고 이 사진 뿐만이 아니라 애니메이션에도 계속 이런 조명을 넣을 수 있다면 더 발전될 것이라고 생각한다.

## REFERENCES

- ▶ android api reference by developers
- ▶ lambert light by google