<모바일응용 소프트웨어 설계>

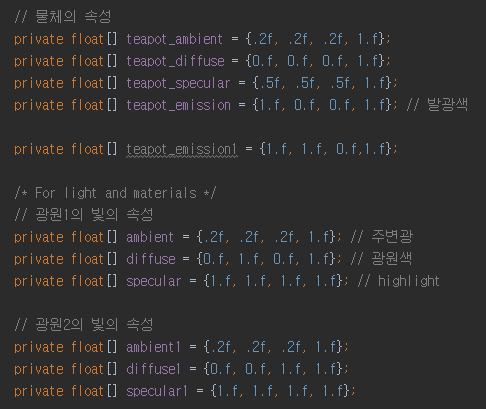
# 10주차 실습 과제

정보통신공학과

12161774 이 채 은

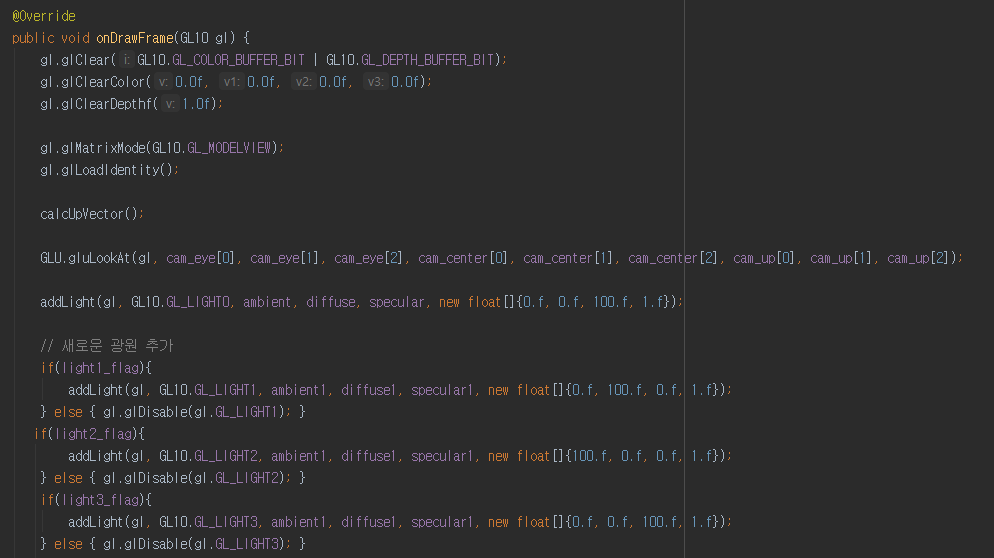
**[소스 코드]**

**1. MyTeapot**



**=>** 조명과 물체의 속성을 정의하는 부분이다. 조명에는 총 세가지 요소가 있는데 각각 ambient, diffuse, specular 이다. Ambient는 공간에 퍼져있는 조명으로 주변광이라고 할 수 있다. 이 빛이 000 이면 다 까맣게 보이고, 111 이면 다 하얗게 보인다고 할 수 있다. Diffuse는 물체에서 반사가 일어나서 우리 눈에 들어오는 빛의 색으로, 물체의 색을 결정한다. 광원의 색이라고 볼 수 있다. Specular 은 물체 표면에서 반사 되는 빛이 일정 방향으로 보이는 것으로, 물체에서 highlight 를 나타낸다. 이 세가지의 빛의 속성이 모두 합쳐져서 광원의 color를 지정하는 것이다.

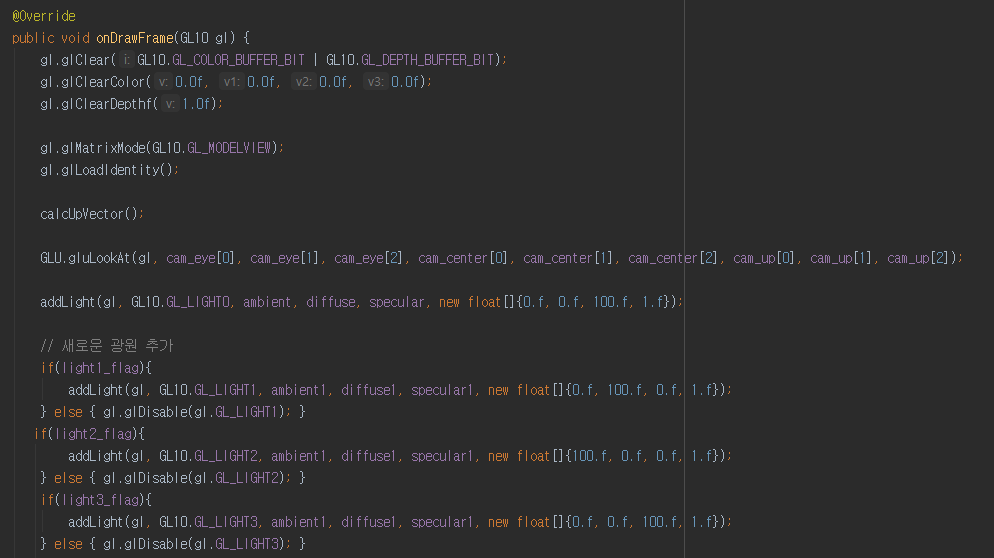
그리고 물체의 속성은, 위의 세가지의 속성에 emission이 추가되는데 물체 자체가 발광하는 색을 나타낸다. 물체의 속성은 물체의 재질과 색을 결정하는 원리이다.

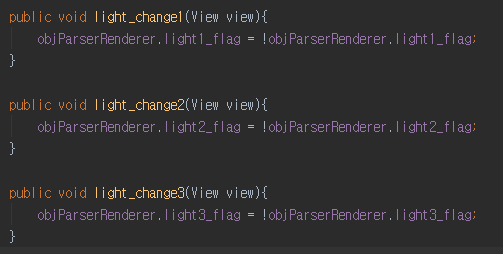


=> 앱 상에서, 광원 효과가 추가 된 후 물체가 보이는 색상을 앞/뒤/옆면 모두 확인해보기 위해서 gluLookAt을 먼저 실행하고 addLight를 해서 물체의 위치를 움직여도 빛의 효과가 그대로 보이도록 설정했다.

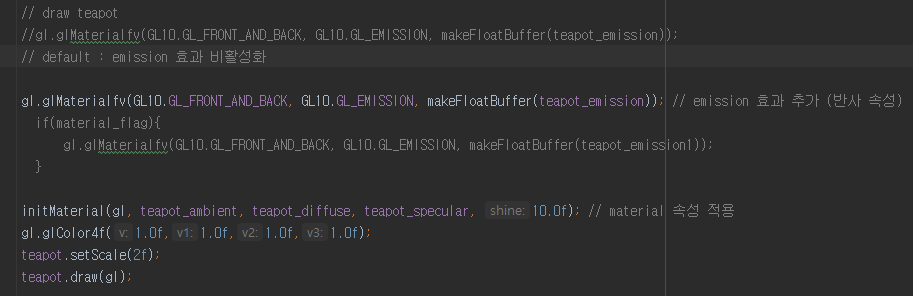


=> addLight 는 위와 같이 광원을 추가해주는 함수이다. 맨 위에서 설정해준 광원의 속성 값 ambient, diffuse, specular의 값을 그대로 대입한다. 그리고 마지막 인자는 광원의 위치이다. 광원은 맨 처음에 local 좌표계에서 원점에 위치해 있다. 따라서 이 코드의 경우에는 첫 번째 광원의 위치를 z 축으로 100.f 만큼 이동시켜서 물체에 조명을 가했다.



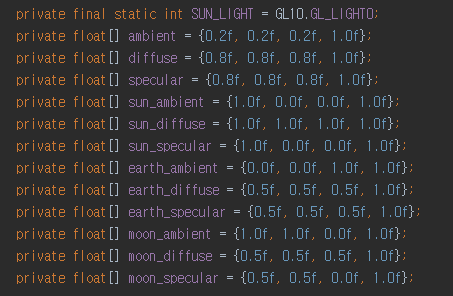
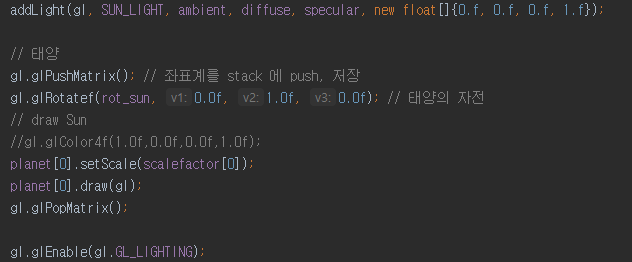


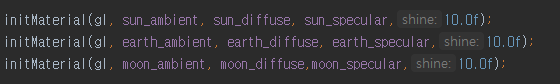
=> 물체에 새로운 광원을 하나 더 추가해서 나타나는 결과를 보기 위해서 flag를 추가하고, flag별로 새로 추가한 광원의 위치를 다르게 했다. Light1\_flag이면 광원을 y축으로 100.f 에 위치시켰고, lignt2\_flag이면 광원을 x축 100.f, light3\_flag이면 광원을 z축 100.f에 위치했다.



=> 위에서 광원의 material 을 속성에 따라 추가했다면, 이 코드는 물체 자체의 속성을 추가해주는 코드이다. initMaterial 함수를 이용해서 맨 위에서 설정해준 속성 값들을 직접 대입한다. 이 때 가장 마지막 인자는 shine을 나타내는 것으로, shine 값이 높을수록 더 매끄럽고, shine 값이 낮으면 울퉁불퉁한 표면을 갖게 된다. 또한 물체 자체의 방출 색상을 나타내는 emission 효과를 추가하기 위해서 glMaterialfv 함수를 써준다. 이는 광원에 의한 물체의 반사 속성을 나타내는 것이고, 마지막에 teapot\_emission 을 마지막 인자로 넣어준다.

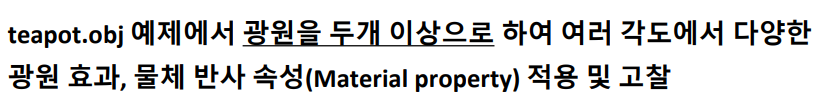
**2. MySolar**

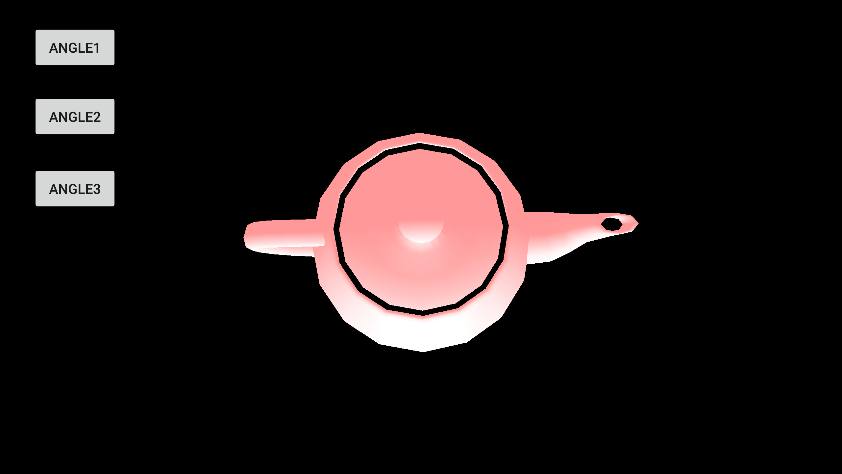




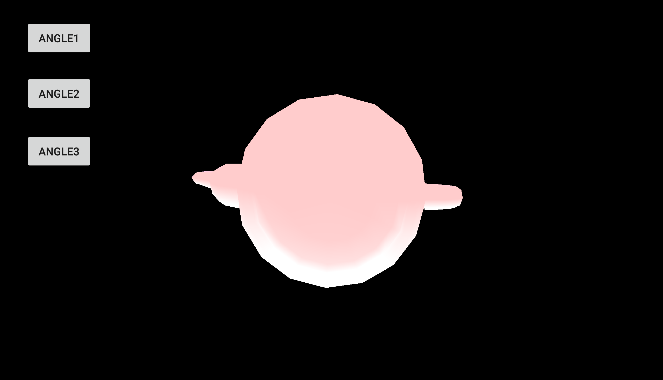
=> 각 행성의 반사 속성 값을 설정했다. 그리고 기존에 설정된 color4f 의 값을 지워주고, light 속성과 material 속성을 teapot 예제와 같은 순서로 추가해준다.

**[ 실습 결과 및 고찰 ]**

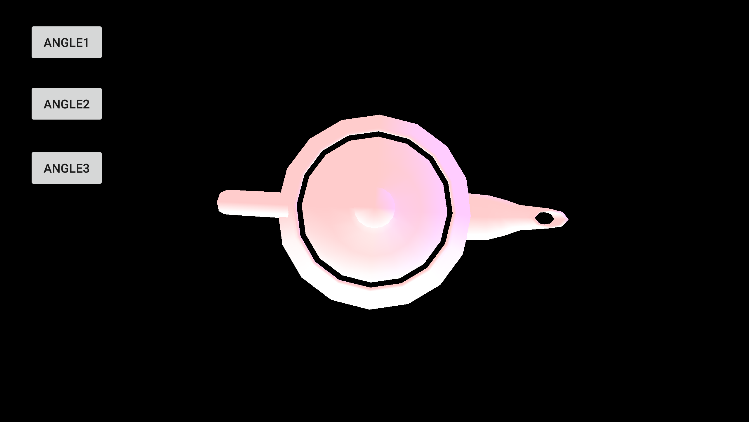




=> 가장 먼저 앱을 실행시키면 맨 처음에 추가한 조명 하나만 물체에 영향을 미친다. 따라서 처음 추가해준 광원은 z축 방향에서 Green 색으로 쏘고 있기 때문에 시야 방향만 색이 바뀐다. 광원의 효과를 확실하게 관찰하기 위해서 초기에 물체의 emission 값은 111(흰색)으로 설정했다. 또한 실행 시킨 결과 주변광 (ambient) 때문에 완벽한 광원의 색인 빨강색이 나오지 않고, 분홍색 빛을 띄는 것을 확인했다.



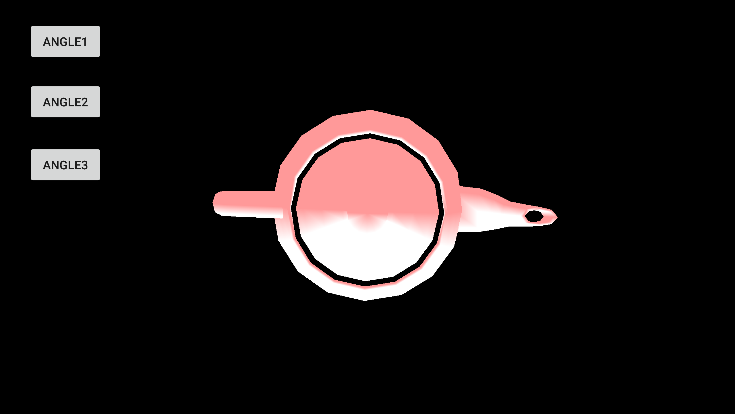
=> Angle1 버튼을 눌렀을 때는, y축 방향으로 Blue 색의 광원을 추가해준다. 따라서 새로 추가된광원의 영향을 받아 색이 바꼈다. 이 때 뒷부분을 보면 y축에 추가된 광원의 영향으로 윗 부분만 보라색으로 변하고 바닥으로 갈수록 새 광원의 영향을 덜 받았다. 그리고 아예 바닥은 새 광원의 영향을 받지 않아서 원래 설정해준 색이 나타나게 된다.



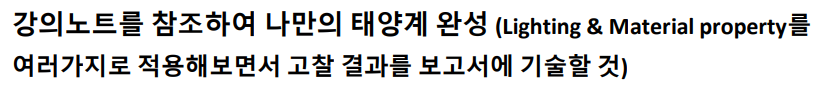
=> Angle2 버튼을 누르면, x축 방향으로 광원이 추가된다. 따라서 윗부분에서 봤을 때 x 축 방향으로만 광원의 영향을 받아 보라색으로 변한 것을 확인할 수 있었다.



=> Angle3 버튼을 누르면, z축 방향으로 광원이 추가되기 때문에 역시 색이 바뀐 것을 확인할 수 있었다. 또한 hightlight 색을 111로 설정해서 하얗게 했고, 물체에서 하얗게 highlight 가 나타나는 것을 확인했다.

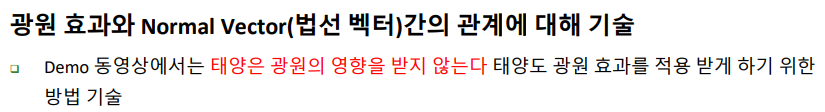


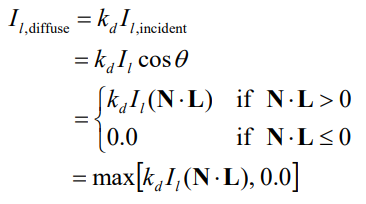
=> 추가로 material 속성 중 shine 값을 바꿔봤다. 10에서 200으로 바꿨을 때 위와 같이 더 매끈하게 물체의 속성이 바뀌는 것을 확인할 수 있었다.





=> 구현 결과는 위와 같다. 태양을 기준으로 태양의 가운데에 광원이 추가되어 있기 때문에 태양을 도는 행성에서 태양쪽은 광원의 효과를 받아 빛나고, 태양의 뒷부분은 조명의 효과를 받지 못했다. Ambient, diffuse, specular 값에 따라서 광원의 효과가 나타난다. 그리고 각 행성의 ambient, diffuse, specular 값에 따라서 행성의 재질이 결정된다. 추가로 shine 값을 바꾸면 재질이 좀 더 매끄럽게 변한다.





=> Diffuse Light 는 광원에 따라 물체 표면에서 분산되어 바로 눈으로 들어오는 빛이다. 이 빛은 빛의 입사 방향과 표면의 법선 벡터에 따라서 결정된다. 따라서 태양 obj 내부에 광원이 있는 MySolar의 경우에는 태양에서 빛이 나지 않는다. 그 이유는 태양의 법선 벡터(N)가 표면에서 밖으로 나가는 방향이기 때문이다. 따라서 L은 to light source 이기 때문에 태양의 법선 벡터와 이루는 각도가 90~180도가 된다. 따라서 위의 식에서 코사인 값이 음수가 되어서 총 값이 0이 되고, 빛이 나지 않게 되는 것이다. 이를 해결하기 위해서 법선 벡터의 방향을 반대로 설정했다. 그렇게 되면 각도가 항상 0~90도가 되고, 마찬가지로 위의 식에서 총 값이 양수가 되어 빛이 나는 것이다. 따라서 법선 벡터의 방향을 바꿔 내적한 값이 양수가 되도록 바꿔주면 광원의 영향을 받게 되는 것이다.