

REMARQUE : Je n'ai pas réussi à faire fonctionner le TP6 même après avoir essayé après le TP par conséquent j'ai emprunté le TP 6 de Maël LE GAL. J'espère que vous ne tiendrez pas compte de ce fait dans la notation.

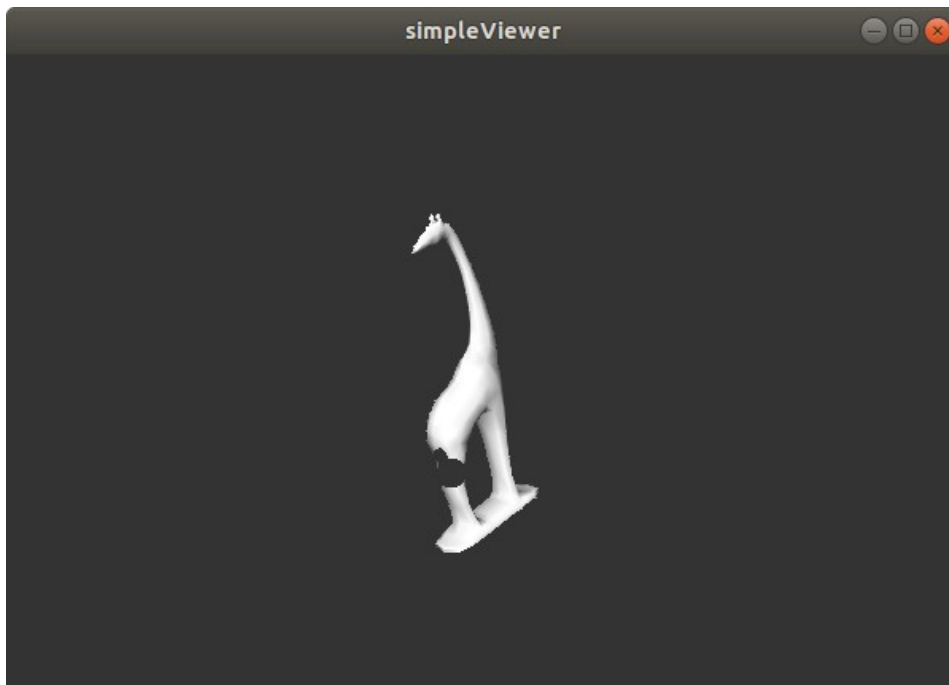
Question 1 :

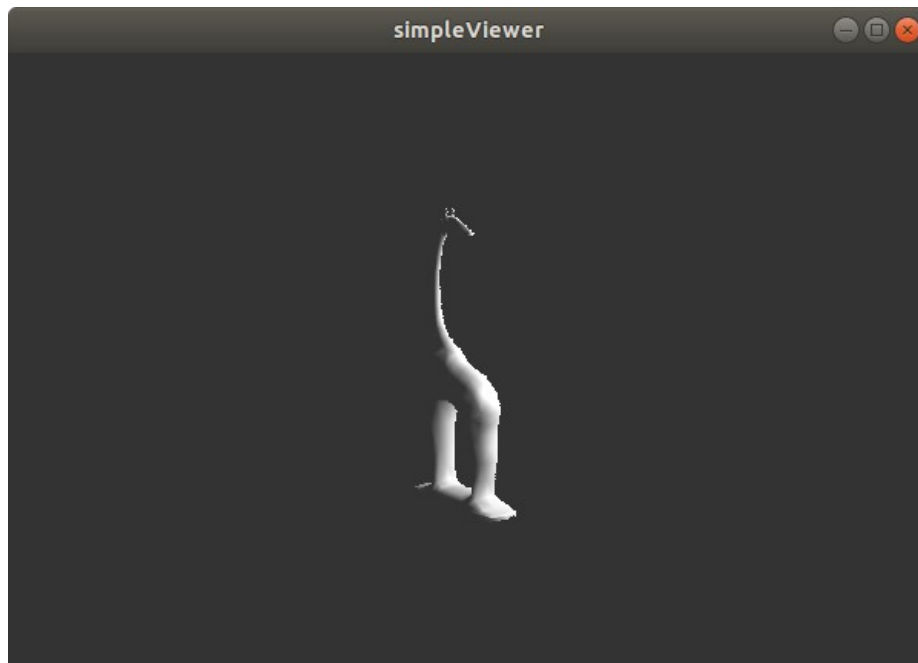
```
glEnable(GL_LIGHTING);
glEnable(GL_LIGHT0);
glEnable(GL_DEPTH_TEST);

GLfloat light_ambient[] = { 0.0, 0.0, 0.0, 1.0 };
GLfloat light_diffuse[] = { 1.0, 1.0, 1.0, 1.0 };
GLfloat light_specular[] = { 1.0, 1.0, 1.0, 1.0 };
GLfloat light_position[] = { 1.0, 1.0, 1.0, 0.0 };

glLightfv(GL_LIGHT0, GL_AMBIENT, light_ambient);
glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse);
glLightfv(GL_LIGHT0, GL_SPECULAR, light_specular);
glLightfv(GL_LIGHT0, GL_POSITION, light_position);
drawLight(GL_LIGHT0);
```

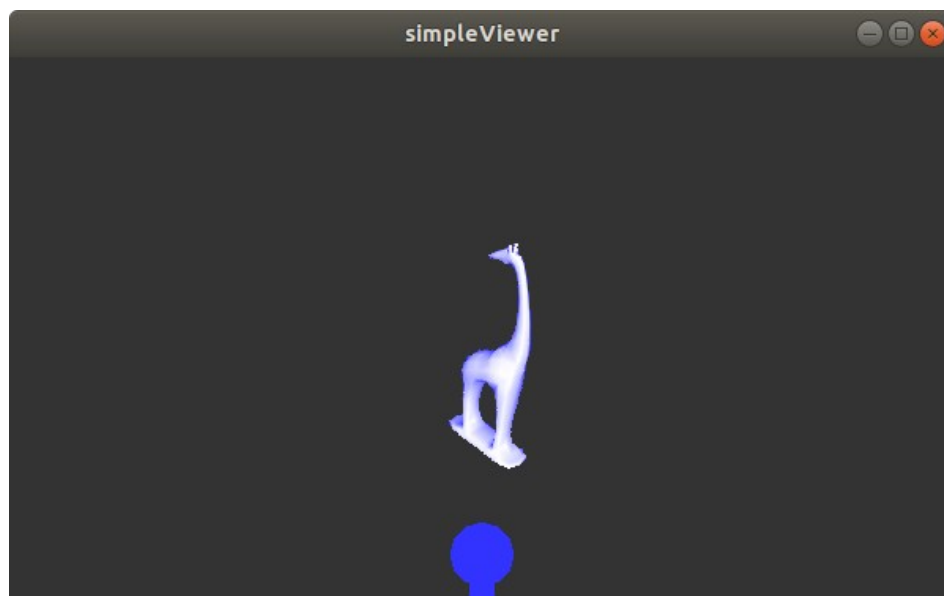
Lorsqu'on met une lumière émet à partir d'une position on peut apercevoir de face la lumière qui est émise en l'occurrence du blanc et le rendu sur la face caché sera l'ombre de l'objet par rapport à la position de la lumière.





LUMIERE DE FACE BLEU :

```
GLfloat light_ambient[] = { 0.0, 0.0, 1.0, 1.0 };
glLightfv(GL_LIGHT0, GL_AMBIENT, light_ambient);
```



Question 2 :

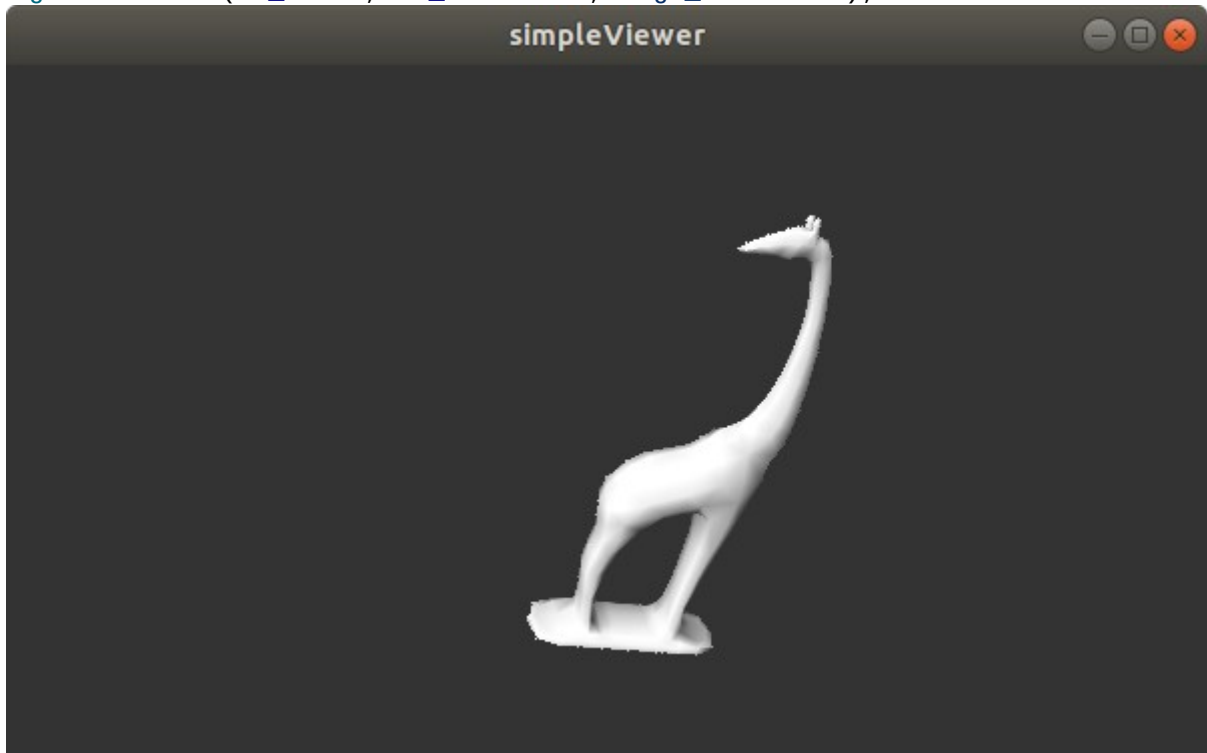
Haute Shininess (éclairage fort)

```
GLfloat red_light[] = { 1.0, 0.0, 0.0, 1.0 };
GLfloat white_light[] = { 1.0, 1.0, 1.0, 1.0 };
GLfloat low_shininess[] = { 5.0 };
GLfloat high_shininess[] = { 100.0 };
```

```

glLightfv(GL_LIGHT0, GL_DIFFUSE, white_light);
glLightfv(GL_LIGHT0, GL_SPECULAR, red_light);
glLightfv(GL_LIGHT0, GL_POSITION, light_position);
glMaterialfv(GL_FRONT, GL_SPECULAR, white_light);
glMaterialfv(GL_FRONT, GL_SHININESS, high_shininess);

```



Pour ce test on utilise de base une lumière rouge spéculaire, une lumière blanche diffuse et la position de la lumière se trouve à (1,1,1). On peut voir ici que la lumière blanche diffusé sur le matériel est plus forte que la lumière émise par la lumière.

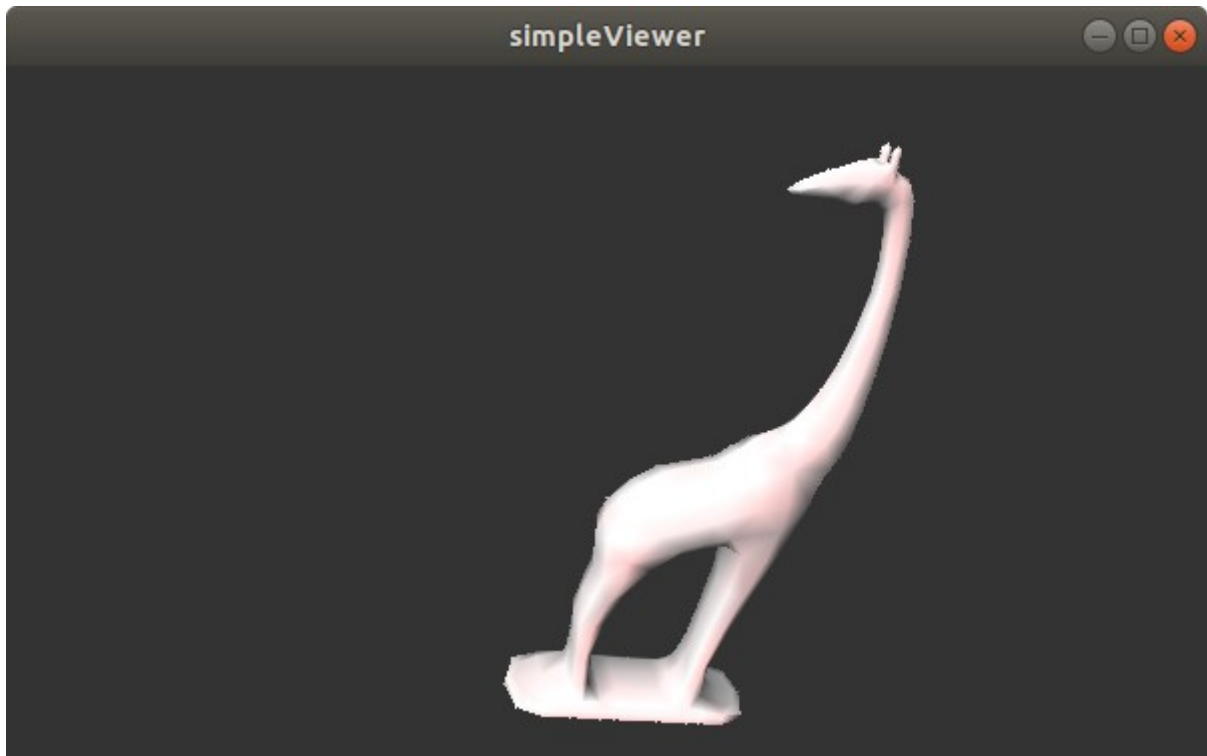
Basse Shininess (éclairage brillant faible)

```

GLfloat red_light[] = { 1.0, 0.0, 0.0, 1.0 };
GLfloat white_light[] = { 1.0, 1.0, 1.0, 1.0 };
GLfloat low_shininess[] = { 5.0 };

glLightfv(GL_LIGHT0, GL_DIFFUSE, white_light);
glLightfv(GL_LIGHT0, GL_SPECULAR, red_light);
glLightfv(GL_LIGHT0, GL_POSITION, light_position);
glMaterialfv(GL_FRONT, GL_SPECULAR, white_light);
glMaterialfv(GL_FRONT, GL_SHININESS, low_shininess);

```

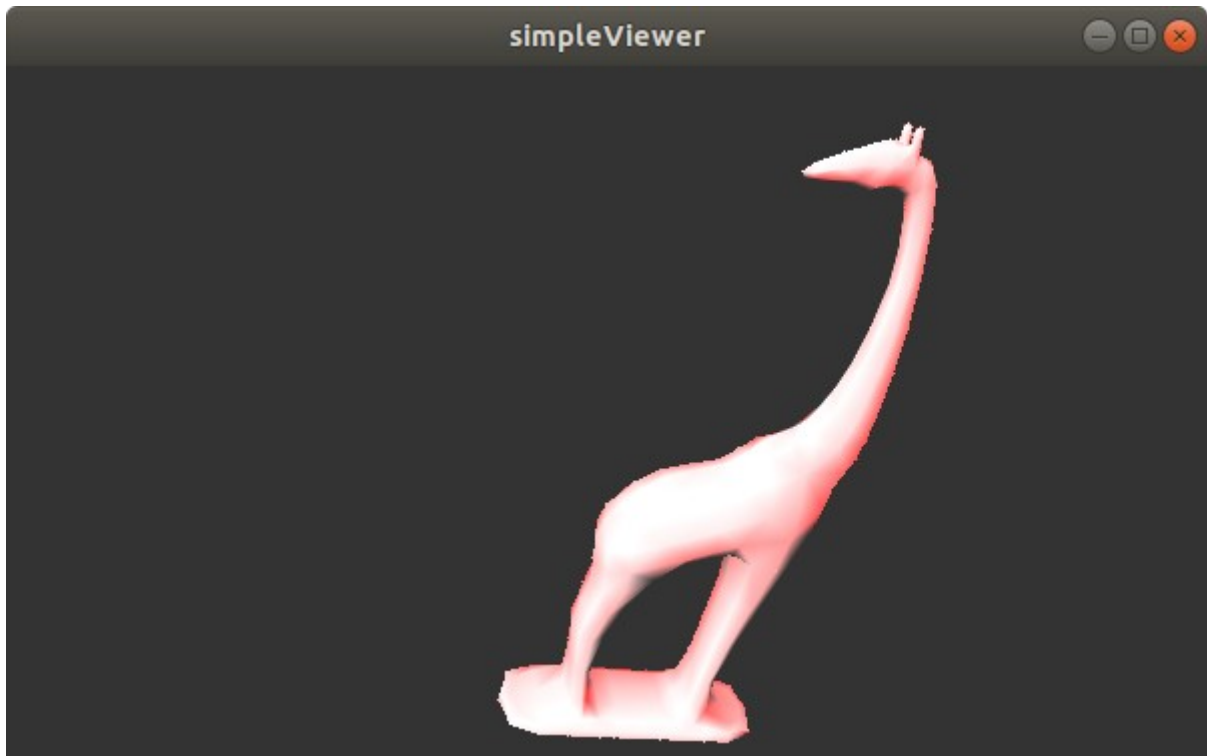


On utilise les mêmes propriétés que l'exemple précédent en modifiant la force de la « brillance », on peut ici apercevoir que la lumière rouge contenu dans la lumière qui éclaire peut être vue sur l'objet signifiant que l'éclairage du matériel est moins fort.

Pas d'éclairage brillant (no Shininess)

```
GLfloat red_light[] = { 1.0, 0.0, 0.0, 1.0 };
GLfloat white_light[] = { 1.0, 1.0, 1.0, 1.0 };
GLfloat no_shininess[] = { 0.0 };

glLightfv(GL_LIGHT0, GL_DIFFUSE, white_light);
glLightfv(GL_LIGHT0, GL_SPECULAR, red_light);
glLightfv(GL_LIGHT0, GL_POSITION, light_position);
glMaterialfv(GL_FRONT, GL_SPECULAR, white_light);
glMaterialfv(GL_FRONT, GL_SHININESS, no_shininess);
```



Ici on enlève encore la plus la brillance (« shininess ») ce qui fait apparaître encore plus le rouge.

Question 3 :

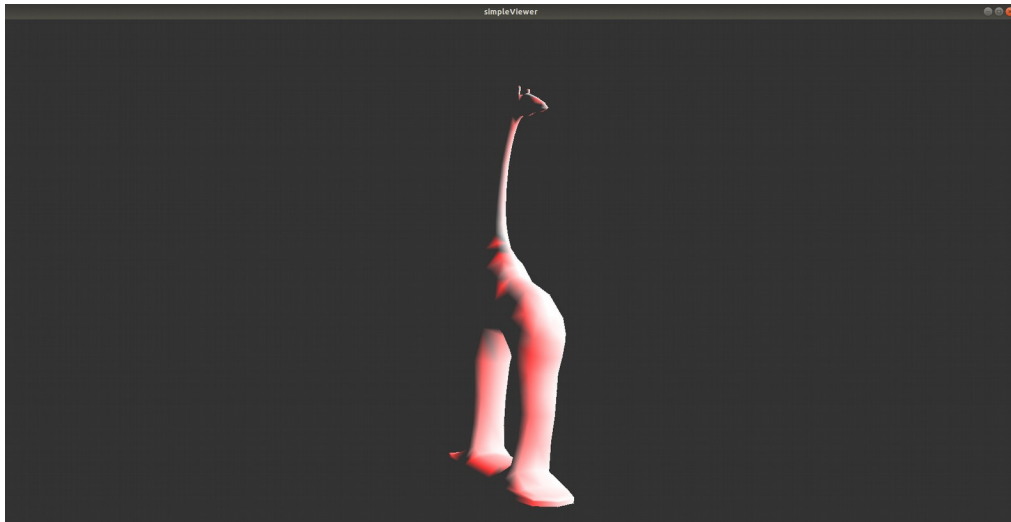
Model Ambient :

```
glEnable(GL_LIGHTING);
glEnable(GL_LIGHT0);
glEnable(GL_DEPTH_TEST);
GLfloat light_ambient[] = { 0.0, 0.0, 0.0, 1.0 };
GLfloat light_diffuse[] = { 1.0, 1.0, 1.0, 1.0 };
GLfloat light_specular[] = { 1.0, 1.0, 1.0, 1.0 };
GLfloat light_position[] = { 1.0, 1.0, 1.0, 0.0 };
glLightfv(GL_LIGHT0, GL_AMBIENT, light_ambient);
glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse);
glLightfv(GL_LIGHT0, GL_SPECULAR, light_specular);
glLightfv(GL_LIGHT0, GL_POSITION, light_position);
drawLight(GL_LIGHT0);

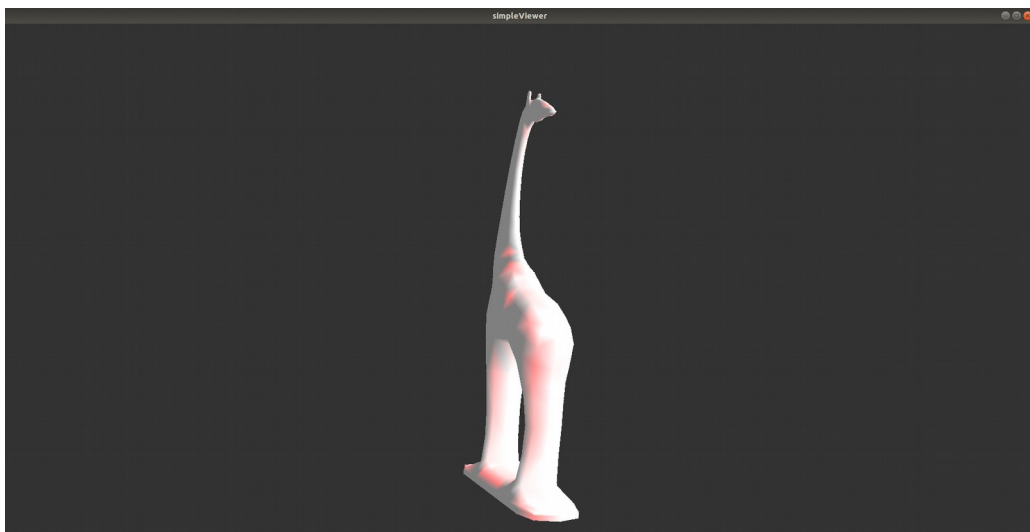
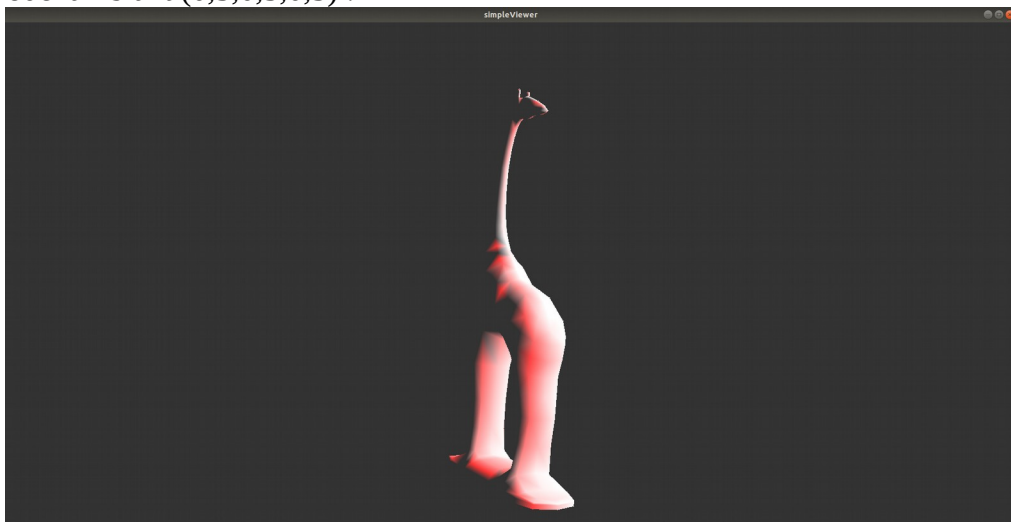
GLfloat red_light[] = { 1.0, 0.0, 0.0, 1.0 };
GLfloat white_light[] = { 1.0, 1.0, 1.0, 1.0 };
GLfloat no_shininess[] = { 0.0 };

glLightfv(GL_LIGHT0, GL_DIFFUSE, white_light);
glLightfv(GL_LIGHT0, GL_SPECULAR, red_light);
glLightfv(GL_LIGHT0, GL_POSITION, light_position);
glMaterialfv(GL_FRONT, GL_SPECULAR, white_light);
glMaterialfv(GL_FRONT, GL_SHININESS, no_shininess);

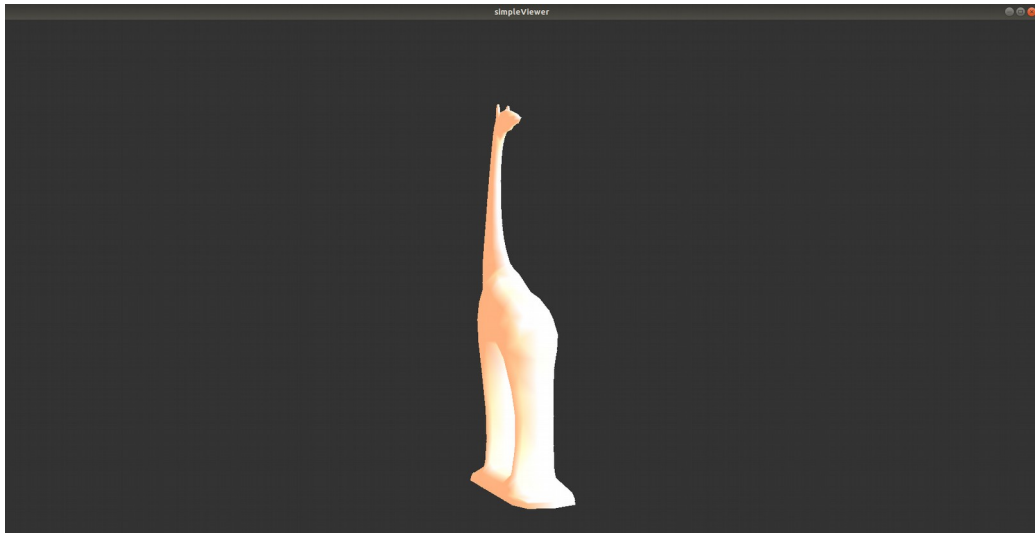
GLfloat model_ambient[] = { 0.5, 0.5, 0.5, 1.0 };
glLightModelfv(GL_LIGHT_MODEL_AMBIENT, model_ambient);
```



Avec un model ambiant (0,5,0,5,0,5) :



Avec un model ambiant (1, 0,7, 0,5) :



Le model ambient diffuse sur la face caché à la lumière la couleur qu'on lui donne en paramètre