Illumination et interpolation

TD n°4

Dans ce TP, vous allez repartir du projet précédent et vous allez y ajouter les fonctions nécessaires pour effectuer l'illumination de votre scène avec un modèle de LAMBERT puis de GOURAUD.

Dans ce projet, il vous faudra exclusivement compléter les parties du code source où le commentaire suivant est marqué

```
// TODO => TP04 //
```

1 L'illumination selon LAMBERT

L'illumination selon LAMBERT effectue un calcul pour déterminer la couleur de chaque face en fonction de la position de la lumière et de l'orientation de la face. Chaque face est donc plus ou moins éclairée en fonction de son exposition à la lumière (voir figure 1).

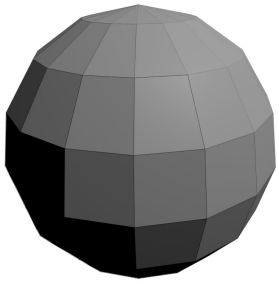

Figure 1 – Illumination selon LAMBERT

Pour calculer l'intensité lumineuse de chaque face, on utilisera le produit scalaire entre la normale de la face et la direction de la lumière soit $\vec{n} \cdot \vec{L}$ selon la figure 2.

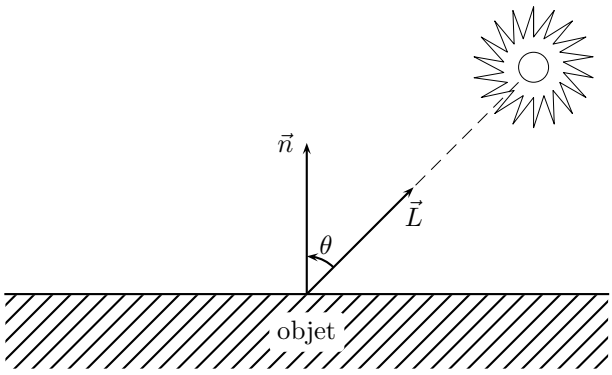

Figure 2 – Calcul de l'intensité lumineuse

Exercice 1 *Complétez le mode de dessin* `DRAW_LAMBERT` *de la fonction* `draw()` *dans le fichier* `object.cpp`. *Ce mode de dessin doit permettre de dessiner l'ensemble des polygones en les remplissant (rasterization) selon le modèle d'illumination de* LAMBERT.

2 L'illumination selon GOURAUD

L'illumination selon GOURAUD ne colore pas les faces de manière uniforme mais tient compte de la couleur de chacun des sommets puis effectue une interpolation sur l'ensemble de la face (voir figure 3).

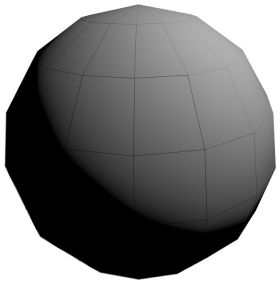

Figure 3 – Illumination selon GOURAUD

Jusqu'à présent, nous avons seulement développé des outils pour dessiner des faces de couleur unie (voir les paramètres que prend la fonction `draw_quad()`). Nous allons devoir créer une fonction `draw_quad()` qui prend en paramètre une couleur différente pour chaque sommet puis qui génère l'interpolation nécessaire. Cela va entraîner également une modification des fonctions `draw_horizontal_line()` et `raster_buffer_insert()`.

Exercice 2 *Complétez la fonction `draw_horizontal_line()` qui prend à présent deux couleurs en entrée et qui dessine une ligne avec une interpolation de la couleur. Le prototype de la fonction est le suivant.*

```
void draw_horizontal_line(int y, int x1, int x2, vec3 c[2]);
```

Exercice 3 *Complétez la fonction `raster_buffer_insert()` qui rempli un tableau de couleur en plus du tableau `raster_buffer`. Le prototype de la fonction est le suivant.*

```
void raster_buffer_insert(int x, int raster_buffer[2], vec3 color, vec3\
    color_buffer[2]);
```

Exercice 4 *Complétez la fonction `draw_quad()` qui dessine un polygône à quatre sommets et qui interpole les couleurs. Le prototype de la fonction est le suivant.*

```
void draw_quad(vec2 p[4], vec3 c[4]);
```

Exercice 5 *Complétez le mode de dessin `DRAW_GOURAUD` de la fonction `draw()` dans le fichier `object.cpp`. Ce mode de dessin doit permettre de dessiner l'ensemble des polygones en les remplissant (rasterization) selon le modèle d'illumination de* GOURAUD.