

VIII – Illumination, éclairage et matériaux

116

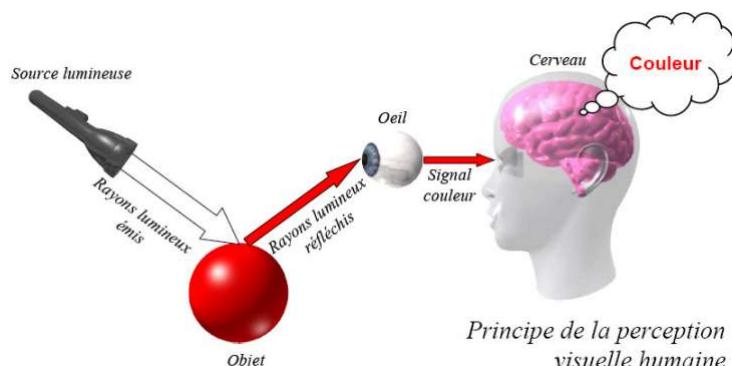
116

VIII.1 – Réalité Virtuelle... moins virtuelle !



- Objectif :

- Rendre les scènes plus réalistes...
- ... en utilisant un « nuanceur » (« shader ») : modification des couleurs
- Représenter l'apparence de l'objet sous l'influence de la lumière
- Simuler des effets lumineux sur les objets présents dans la scène



117

117

VIII.1 – Calcul de rendu graphique

- Principe du calcul du « Rendu graphique » :
 - Exploitation de la carte graphique !
 - ... car généralement des algorithmes complexes, de calcul :
 - d'absorption de la lumière
 - de diffusion,
 - de réflexion,
 - d'ombrage, etc.
- Paramètres principaux pour effectuer ces calculs :
 - types d'**éclairage**
 - diverses formes de sources lumineuses
 - diverses intensités lumineuses
 - caractéristiques des **matériaux** qui constituent les objets
 - capacités à réfléchir ou non la lumière (mat ou brillant, ...)
 - calcul de l'**illumination**
 - analyse de l'apport des sources lumineuses sur l'objet éclairé

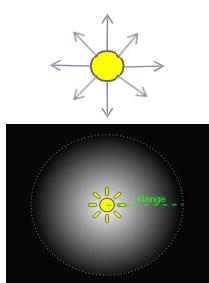
118

118

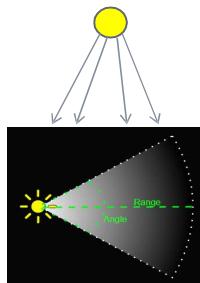
VIII.2 – Types d'éclairage

- Différentes formes de sources lumineuses :

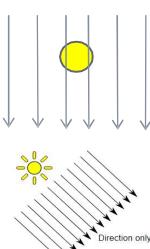
Point lumineux



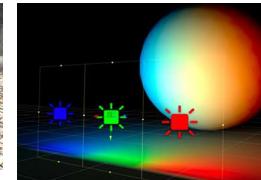
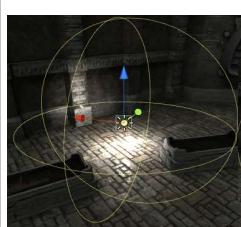
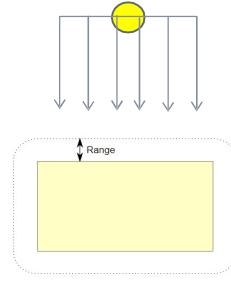
Spot



Source Directionnelle



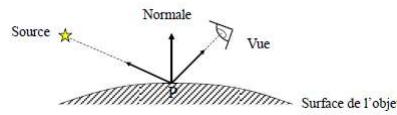
Aire / Source Ambiante



119

VIII.3 – Modèle de matériaux

- Utilité du modèle de matériaux
 - Un matériau est vu parce qu'il réfléchit la lumière !



- Caractérisation d'un matériau par ses capacités à réfléchir la lumière
 - Réflexion ambiante (K_a) : couleur intrinsèque de l'objet
 - Réflexion diffuse (K_d) : couleur émise par réflexion de l'objet
 - Réflexion spéculaire (K_s et N_s) : réflexion de la source et brillance localisée (« shininess »)

- Exemple : extrait du .MTL de l'objet « dragon »

```
newmtl skin
Ka 0.2286 0.0187 0.0187
Kd 0.1102 0.0328 0.0139
Ks 0.3000 0.3000 0.3000
illum 2          // = 2 : presence speculaire
Ns 17.8300
```



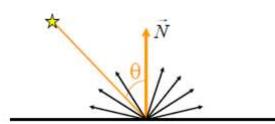
120

120

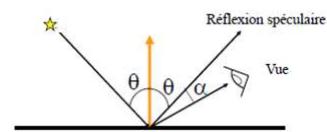
VIII.3 – Modèle d'illumination

- Illumination des matériaux
 - Calcul des réflexions du matériau exposé à une source lumineuse
 - Intensité de la source lumineuse : I_{sl}
- Intensités lumineuses résultantes, issues des réflexions
 - Ambiante :
 - réflexion proportionnelle, multi directionnelle
 - $I_a = I_{sl} \times K_a$

- Diffuse :
 - dépend de l'angle de la source lumineuse
 - $I_d = I_{sl} \times K_d \times \cos(\theta)$



- Spéculaire :
 - dépend des angles : de la source et du point de vue
 - $I_s = I_{sl} \times K_s \times (\cos(\alpha))^{N_s}$

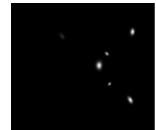


121

121

VIII.3 – Modèle d'illumination de « Phong »

- Aspect des diverses composantes :
 - Réflexion **Ambiante**
 - Coloriage uniforme avec la couleur de base
 - Pas d'ombrage sur les différentes parties de l'objet
 - Silhouette apparaissant de couleur uniforme, sans relief
 - Réflexion **Diffuse**
 - La surface diffuse de façon uniforme dans tout l'espace
 - L'aspect d'un point de la surface
 - dépend de l'angle de la source de lumière (ombrage)
 - mais ne dépend pas du point de vue de l'utilisateur
 - Réflexion **Spéculaire**
 - La surface diffuse de façon non uniforme, selon le point de vue
 - La source de lumière apparaît donc comme une tâche brillante
 - Cette tâche est de la couleur de la source (et non de l'objet)
 - Cette tâche se déplace à la surface de l'objet lorsque l'observateur bouge
 - Modèle d'illumination de « Phong »
 - La couleur RGB finale est la somme des 3 !
 - Couleur Finale = Ambiante + Diffuse + Spéculaire



122

122

VIII.4 – Illumination dans Processing

- Principe de mise en œuvre de l'illumination
 - Ne fonctionne qu'en mode 3D : donc `size(..., ..., P3D);`
 - Version « basique » possible :
 - Toutes les valeurs par défaut, Lumière blanche unidirectionnelle : `lights();`
 - Lance le calcul des ombres (sur carte graphique)
 - Attention : **moins** lumineux lorsque la lumière est allumée !

<https://processing.org/examples/onoff.html>
-
-
- #### ■ Configuration des caractéristiques des matériaux
- Fonctions à appeler **avant** le dessin de l'objet
 - `ambient()` : couleur intrinsèque de l'objet (remplace "fill")
 - `emissive()` : couleur renvoyée par l'objet sous un éclairage blanc (joue le rôle de "diffuse")
 - `specular()` et `shininess()` : brillance de type "gloss" ("specular" = Ks et "shininess" = Ns)
- 123
- 123
- Emmanuel MESNARD
- 4

VIII.4 – Illumination dans Processing

- Configuration des éclairages
 - Couleur de la source
 - généralement : lumière blanche...
 - mais possibilité de mettre une couleur RGB quelconque !
 - Ajout d'une composante spéculaire sur les sources : `lightSpecular()`
 - A mettre avant la définition des sources lumineuses
- Sources lumineuses
 - Ambiante : `ambientLight()`
 - Pas de direction privilégiée
 - Généralement utilisée en la combinant avec un autre type de lumière
 - Directionnelle : `directionalLight()`
 - Très dirigée
 - Effet d'ombre sur l'objet
 - Spot : `spotLight()`
 - Identique à la lumière directionnelle, mais avec plus d'options...
 - Choix de l'angle du cône d'éclairage
 - Point : `pointLight()`
 - Identique au spot, mais avec un cône à 180 degrés

124

124

VIII.5 – Exemple 15 : Illumination

```
...
void draw() {
    specular(255, 255, 255); // Matériaux avec reflet blanc pour le trace des objets

    // Dessin d'une petite boîte blanche en remplacement du curseur de
    // la souris, et simulant la présence d'une source lumineuse
    pushMatrix();
    translate(mouseX, mouseY, 120);
    box(5);
    popMatrix();

    lightSpecular(0, 55, 128); // Ajout d'une composante spéculaire en lumière bleue foncée
    // Source en lumière ponctuelle, en blanc gris, pilotée à la souris
    pointLight(200, 200, 200, mouseX, mouseY, 120);

    // Sphere à gauche avec faible shininess
    translate(width/3, height/2, 0);
    shininess(1.0);
    sphere(height/4);

    // Sphere à droite avec fort shininess
    translate(width/3, 0, 0);
    shininess(8.0);
    sphere(height/4);
}
```



125

125