# 6 - Finalisation et bonus

## Objectif

Finaliser le projet, corriger les derniers bugs et éventuellement ajouter les bonus.

## Spécification

### Finalisation du projet

A ce stade, n'hésitez pas à refaire une passe sur votre code pour

- Vérifier sa propreté (nommage, clarté, respect des conventions)
- Que les méthodes soient lisibles et courtes
- Ajouter de la documentation
- Compléter vos tests unitaires et d'intégration
- **Vérifier que votre code compile** et n'a aucune dépendance hors librairie standard (excepté JUnit)
- Préparer et valider un exécutable fonctionnel que vous ajouterez à votre archive finale
- Préparer le rendu de la scene `final.scene`

> **Attention**
>
> **Un code qui ne compile pas n'est pas exploitable.** Je ne pourrai pas prendre le temps de corriger les bugs de compilation de toute la promo. Ce sera forcément pénalisant pour vous.

### Bonus - Autres formes

#### Intersection avec un plan

- Un plan est défini par un point $q$ et une normale $\vec{n}$.
- On cherche un point $p = o + \vec{d} * t$ tel que $(p - q)$ et $\vec{n}$ soient orthogonaux, soit $(p - q) \cdot \vec{n} = 0$.
- Il faut donc résoudre l'équation $(o + \vec{d} * t - q) \cdot \vec{n} = 0$
  - soit $(o - q + \vec{d} * t) \cdot \vec{n} = 0$
  - soit $(o - q) \cdot \vec{n} + \vec{d} * t \cdot \vec{n} = 0$.
- On trouve donc $t = \frac{-(o-q)\cdot\vec{n}}{\vec{d}\cdot\vec{n}}$.
- Comme $-(o - q) = (q - o)$, on peut simplifier en

**À implémenter**

$$t = \frac{(q - o) \cdot \vec{n}}{\vec{d} \cdot \vec{n}}$$

Si $\vec{d} \cdot \vec{n} = 0$, notre équation n'a pas de solution : $\vec{d}$ est perpendiculaire à la normale, donc est parallèle au plan.

## Intersection avec un triangle

- On dispose de trois points, donnés selon l'ordre anti-horaire : $a, b, c$.
- On cherche tout d'abord si le point se trouve dans le plan du triangle, en utilisant la même technique que pour le plan.
- Ensuite, on cherche si le point est à l'intérieur du triangle en recherchant l'expression du point à partir des coordonnées des points.

### Intersection avec le plan

- On cherche tout d'abord la normale du triangle $a, b, c$

$$\vec{n} = \frac{(b - a) \cdot (c - a)}{||(b - a) \cdot (c - a)||}$$

- On applique la technique ci-dessus pour calculer la valeur de $t$ si elle existe.

### Calcul des normales

Il suffit de vérifier les trois conditions suivante :

- $((b - a) \times (p - a)) \cdot \vec{n} \geq 0$
- $((c - b) \times (p - b)) \cdot \vec{n} \geq 0$
- $((a - c) \times (p - c)) \cdot \vec{n} \geq 0$

### Calcul du barycentre

Il s'agit maintenant de vérifier que le point est à l'intérieur du triangle.

Si c'est le cas, alors on peut exprimer $p$ en fonction de $a$, $b$ et $c$,
soit $p = \alpha * a + \beta * b + \gamma * c$ tel que

- $\alpha \geq 0, \beta \geq 0, \gamma \geq 0$
- $\alpha + \beta + \gamma = 1$

On peut définir $\alpha = 1 - \beta - \gamma$ et le remplacer dans la première équation :

$$p = (1 - \beta - \gamma) * a + \beta * b + \gamma * c = a + \beta * (b - a) + \gamma * (c - a) p - a = \beta * (b - a) + \gamma * (c - a)$$

#### Calcul implémenté

**À implémenter**

##### Intersection avec le plan

On calcule tout d'abord la normale $\vec{p}$ du plan formé par $\vec{d}$ et $(c - a)$.

$$\vec{p} = \vec{d} \times (c - a)$$

On vérifie ensuite la valeur du cosinus de l'angle entre $\vec{p}$ et $(b - a)$.

$$det = (b - a) \cdot \vec{p}$$

Si ce cosinus est proche de 0, alors $\vec{p}$ et $(b - a)$ sont perpendiculaires, donc $\vec{d}$ est parallèle au plan du triange.

$$\text{si } |det| < \epsilon \text{ alors pas d'intersection}$$

##### A l'intérieur du triangle

- $\vec{t} = (lookFrom - a)$
- $\beta = \frac{\vec{t} \cdot \vec{p}}{det}$
- si $\beta < 0$ alors pas d'intersection
- $\vec{q} = \vec{t} \times (b - a)$
- $\gamma = \frac{\vec{d} \cdot \vec{q}}{det}$
- si $\gamma < 0$ ou $\beta + \gamma > 1$ alors pas d'intersection
- $t = \frac{(c-a) \cdot \vec{q}}{det}$
- si $t < \epsilon$ alors pas d'intersection
- retourner $t$

## Bonus - Réflexion

### Prise en compte de l'illumination indirecte

Jusqu'à présent, la couleur d'un point était calculée à partir des sources de lumières directes : la lumière d'ambiance et les sources de lumière ponctuelles ou directionnelles.

Nous allons maintenant prendre en compte la lumière réfléchie.

Soit $c$ la couleur d'un point $p$ comme calculée au Jalon 5 à l'aide des modèles d'illumination de Lambert et Blinn-Phong.

Si la valeur de la couleur $specular$ de l'objet est différente de zéro (noir), il s'agit d'une surface réfléchissante.

**Il s'agit d'une convention dans notre représentation des scènes, pour simplifier la description des scènes. Vous trouverez dans la littérature des couleurs spécifiques pour la lumière réfléchie, différente de la couleur specular.**

Dans ce cas, il est nécessaire d'ajouter à la couleur courante la couleur reçue indirectement via réflexion.

- On note $\vec{r}$ le vecteur qui indique la direction "réfléchie" du regard, c'est à dire ayant le même angle avec la normale que $\vec{d}$, sur le même plan que $\vec{n}$ et $\vec{d}$.
- On calcule $\vec{r}$ de la manière suivante :

$$\vec{r} = \vec{d} + 2 * (\vec{n} \cdot (-\vec{d})) * \vec{n}$$

Le calcul de cette contribution se fait en trois étapes :

1. On vérifie tout d'abord qu'il existe un objet renvoyant de la lumière, c'est à dire qu'il existe une intersection avec un objet dans la direction $\vec{r}$ de la lumière réfléchie.
2. Si c'est le cas, on calcule la couleur $c'$ du point $p'$ d'intersection avec cet objet à l'aide d'un appel récursif à la méthode de calcul de couleur. **Pour arrêter la récursion, il faut utiliser un paramètre de profondeur `maxdepth` et s'arrêter quand `maxdepth` est atteint.**
3. Enfin on ajoute la contribution de la lumière réfléchie en appliquant la couleur de réflexion : $c = c + specular * c'$.

On peut représenter cette approche à l'aide d'une suite arithmétique, en considérant que $p$ est le point d'intersection courant et $p'$ est le point d'intersection depuis $p$ selon $\vec{r}$.

- $u_i(p) = couleur_{directe}(p)$ si $specular_p = (0,0,0)$ ou si pas de point $p'$
- $u_i(p) = couleur_{directe}(p) + specular_p * u_{i+1}(p')$
- $u_{maxdepth}(p) = (0,0,0)$

## Attention aux couleurs calculées

Jusqu'à présent, par construction des scènes, les sources de lumière directes ne peuvent pas résulter en une couleur dont une composante dépasse 1 (100%).

En prenant en compte les sources de lumière indirectes, il est possible d'obtenir des couleurs dont l'une des composantes dépasse 1. Si c'est le cas, cette composante sera ramenée à 1.

## Le fichier de description de scènes

Afin de gérer dans le fichier de scène la limite du calcul récursif de la lumière, on ajoute un nouveau paramètre `maxdepth`.

Si `maxdepth` vaut 1, alors on ne prend pas en compte la lumière réfléchie, on se limite à la contribution des lumières directes.

Quand `maxdepth` augmente, le nombre de reflets dans les objets est susceptible d'évoluer.

Généralement, on utilisera une valeur de `maxdepth` comprise entre 5 et 10.