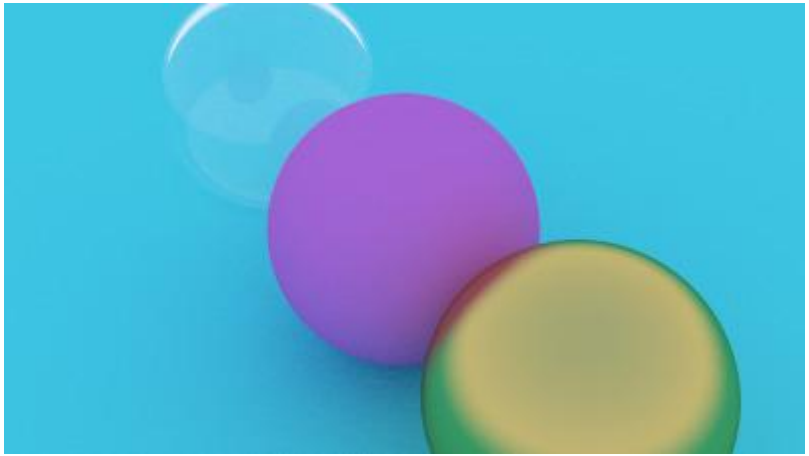
Soutenance Projet long

Ray Tracing

LI Ningyu & Wang Wenchi

09/05/2022



- Introduction
- Architecture
- Programmation
- Conclusion

Domaine transversal :

- Optique
- infographie
- mathématiques
- Algorithmes
- …

Problématique :

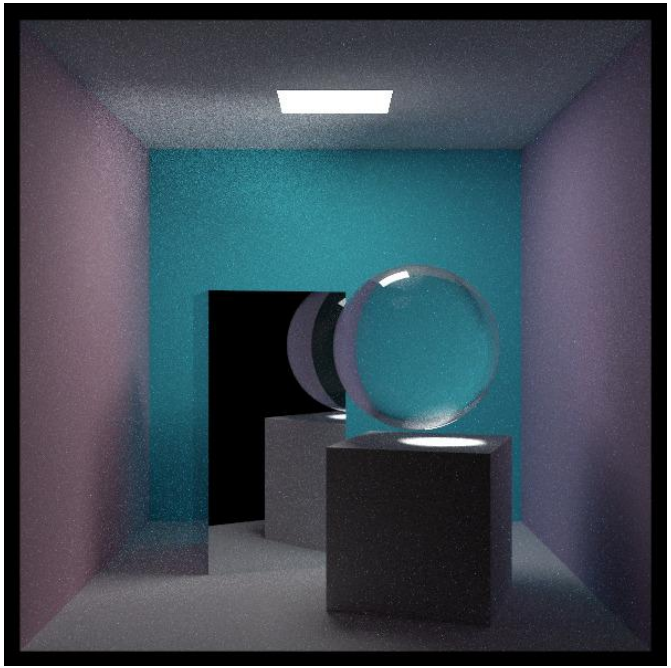
- Réalité
- vitesse d’exécution

Objectif initial :

créer une scène où les objets, la lumière et les ombres de la scène 3D sont affichés sur une image 2D au moyen de la méthode de Monte Carlo

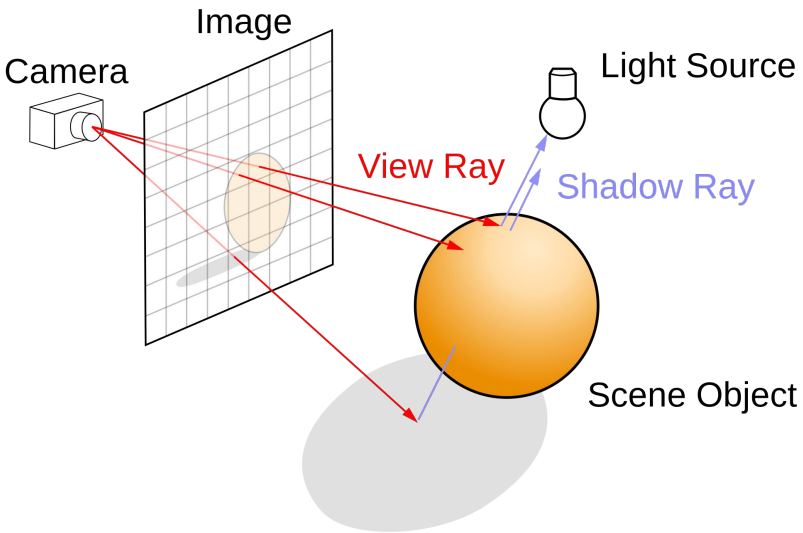
Notre produit :

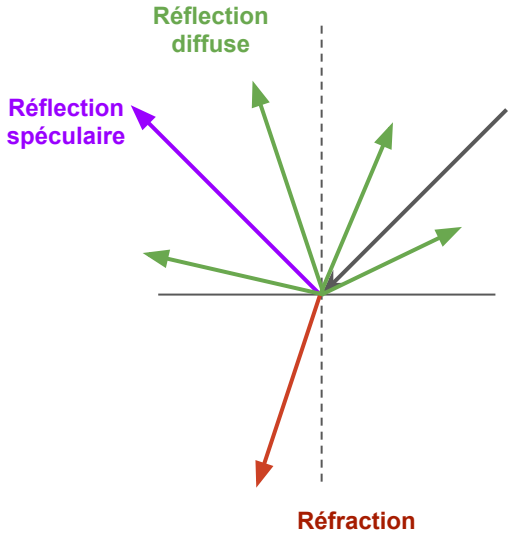
processus de calcul de la lumière ligne par ligne

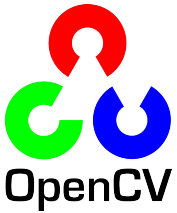


Taille : 600*600, 1000 exemplaires par pixel

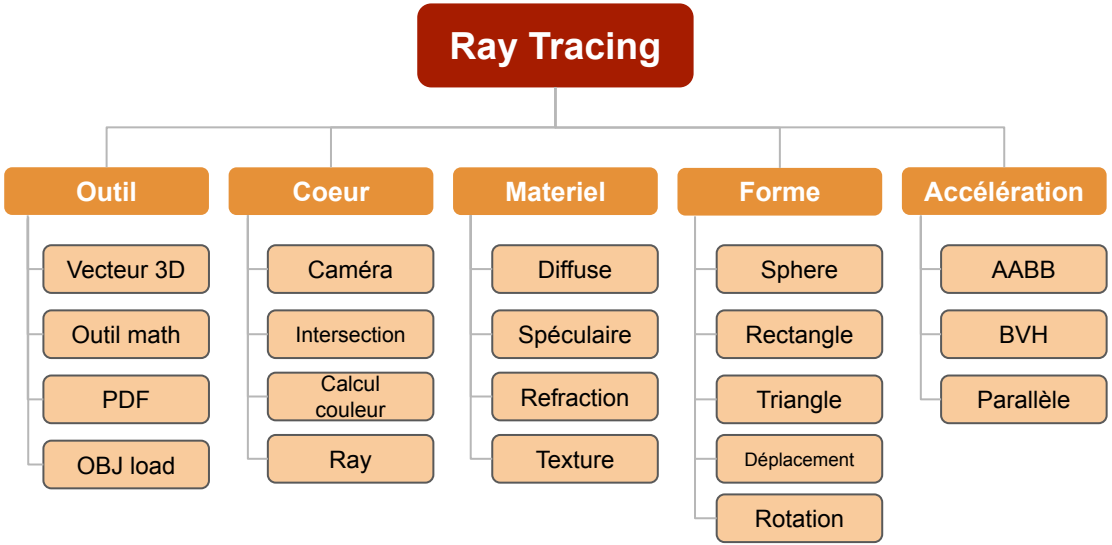
Fonctionnement :



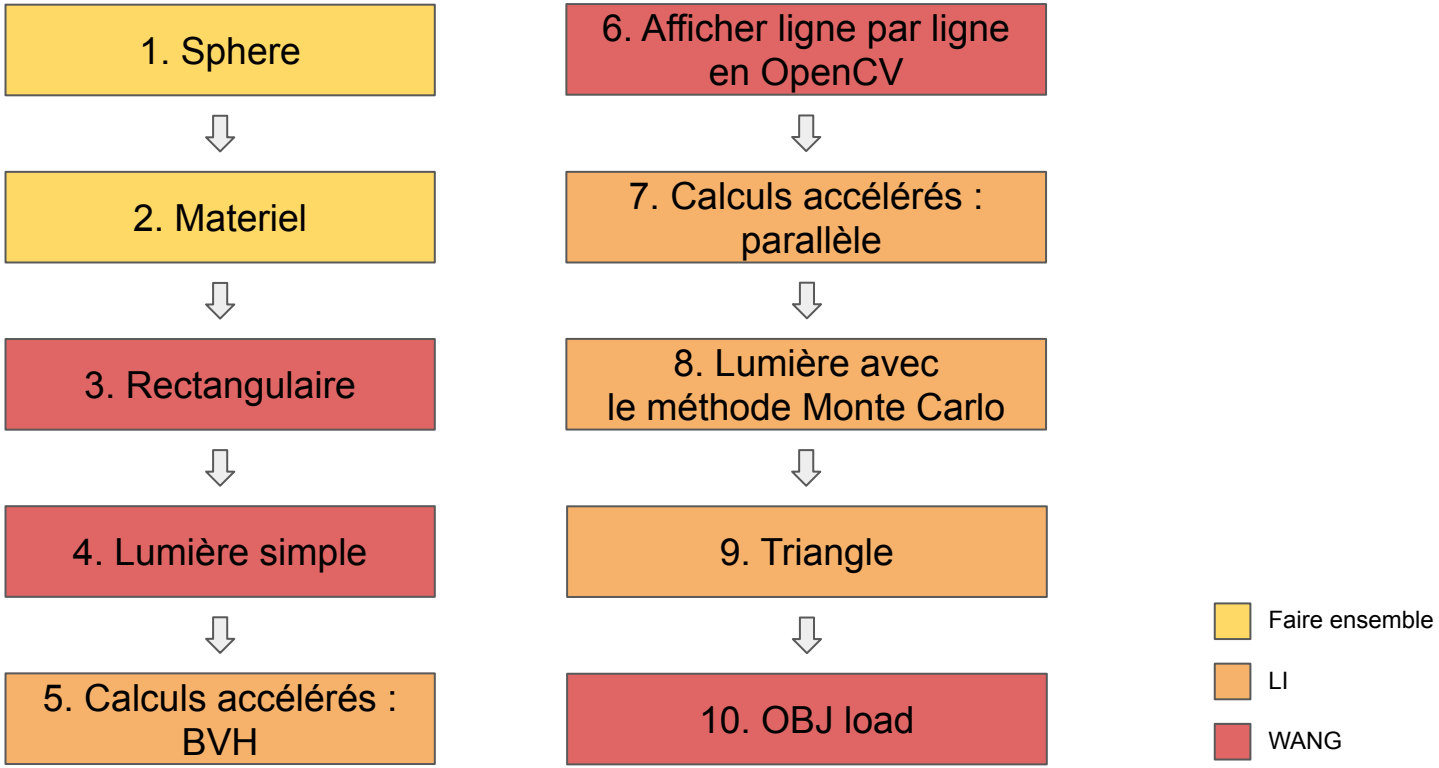




Modules principaux :



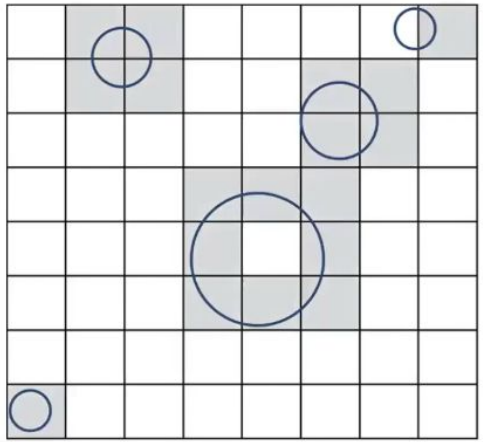
Décomposé le problème :



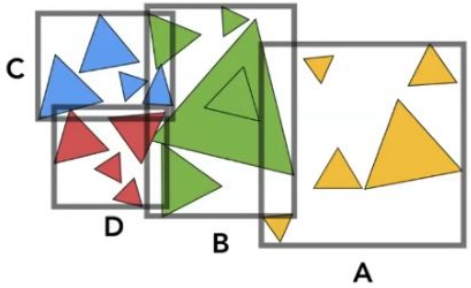
Principales difficultés :

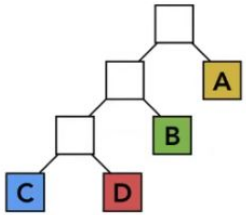
Trouver l'objet le plus proche du rayon	⇨	structure pour enregistrer les informations d'intersection
Affichage dynamique du processus de scannage	⇨	OpenCV (problème en plusieurs threads)
Image de sortie a beaucoup de bruit	⇨	Échantillonnage de la lumière avec le méthode Monte Carlo
Calcul avec le GPU	➡	OpenGL Pas encore fait

Algorithme : Arbre - BVH (bounding volume hierarchy)



Partition Objets - BVH






```
Intersect ( Ray ray, BVH node ) {
   if ( ray misses node.bbox) return ;

   if (node is a leaf node )
      Test intersection with all objs ;
      Return closest intersection ;

   hit1 = Intersect ( ray, node.child1 ) ;
   hit2 = Intersect ( ray, node.child2 ) ;

   Return the closer of hit1, hit2 ;
}
```

Nouvelle domaine :

infographie

Regrouper les cours qu'on a appris :

- Théorie et pratique de la concurrence
- Langages à objet avancés
- Outils formels pour la science des données
- Algorithmique avancée et complexité
- Algèbre linéaire

Version 2.0

- Calcul avec GPU
- Programmer avec OpenGL
- Scènes complexes

Merci

09/05/2022