

# DIAPORAMA DU PROJET DE RENDU 3D PAR LANCER DE RAYONS

ABBAD KAMEL - 21911536  
BOUSADIA LAHCENE - 21911132  
MARTIN MAXENCE - 21807030  
MEYER ARTHUR - 21805134

**Université de Caen Normandie**  
**PROFESSEUR : G. Bonnet, C. Alec**

04-MAI-2020

- 1 Introduction
  - Présentation du projet
  - Spécification minimal demandé
  
- 2 Structure générale du projet
  - Arborescence des packages
  - Les packages utilisés
  
- 3 Rendu final
  - Diagramme UML
  - Interface graphique
  
- 4 Conclusion

## Définition du Lancer de rayons

Le ray tracing (lancer de rayons) est une technique de rendu de graphiques en trois dimensions avec des interactions lumineuses très complexes. Cela signifie que vous pouvez créer des images pleines de miroirs, de surfaces transparentes et d'ombres, avec des résultats étonnants comme on peut le voir dans la figure.

# Spécification minimal demandé

## Objectif du projet

L'objectif de notre project est de faire une application qui prend un fichier décrivant une scène au format ".pov" et de créer l'image associé. Pour pouvoir afficher l'image on doit lire le fichier qui décrit la scène puis y appliquer un lancé de rayons pour créer une image où chaque rayons est un pixel qui change de couleurs en fonction de l'acteur touché, de l'éclairage etc ... Pour ce faire nous devons utiliser un peu d'algèbre et de géométrie pour colorer correctement rayons (et donc pixels).

# Arborescence des packages

## Hierarchie générale

Dans notre projet on a crée un grand dossier «Engine» qui est le principal package qui est constitué de trois autres répertoires aussi.

## Arbre des packages

Une photo qui résume les packages créés dans notre projet.

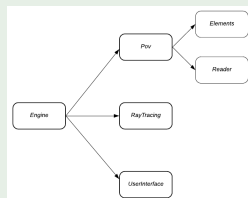


Figure – Arborescence des packages.

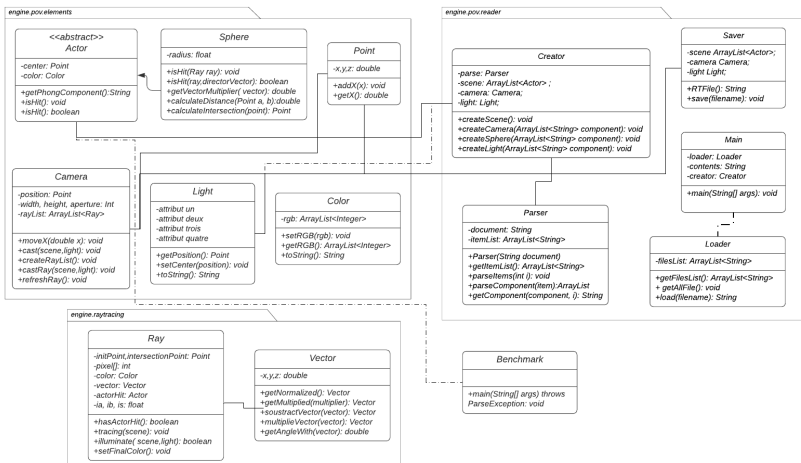
## Le role de chaque package

- Le package POV qui se divise en deux packages « Elements » et « Reader », on retrouve toutes les classes responsable des acteurs (formes géométriques), caméra et la lumière ainsi le chargement, sauvegarde et création des scènes.
- RayTracing est le deuxième package qui gère les rayons et toutes les équations des vecteurs qui y sont associés.
- Enfin le package UserInterface qui est responsable de l'affichage d'interface graphique (boutons de sauvegarde, update, vue...), qui contient aussi la classe exécutable Main.

# UML du projet

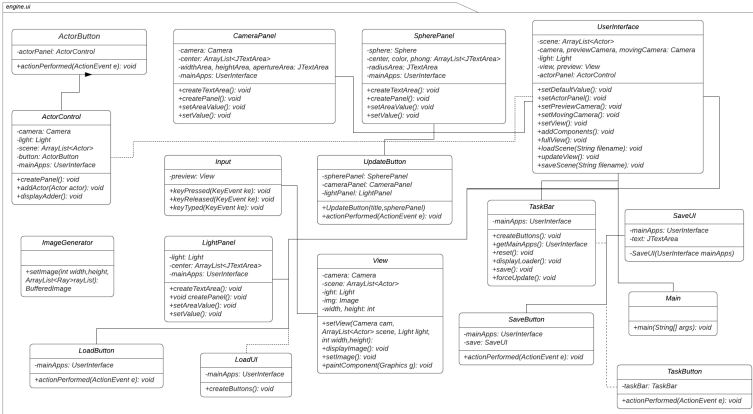
## UML du Core

Diagramme général des classes qui représente le Core.



# UML de l'interface

Diagramme général des classes qui représente l'interface graphique.





# Interface finale

## Version final

Image finale de l'interface graphique après l'exécution.

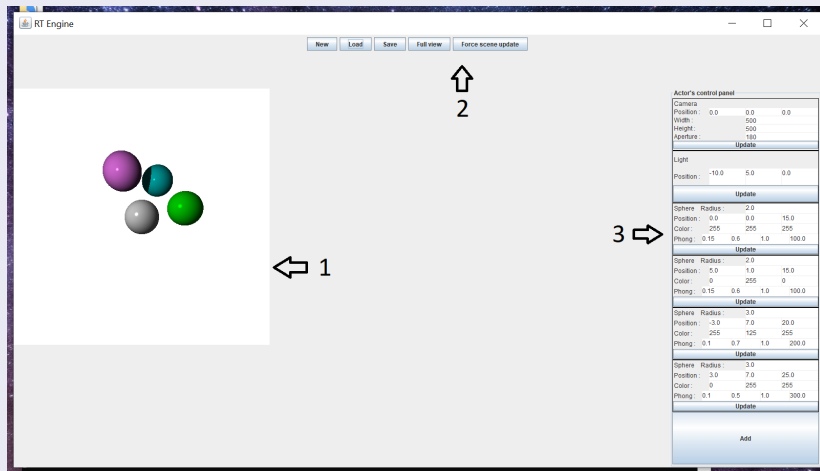


Figure Interface finale

## Conclusion générale

- L'objectif principal de ce projet est de réaliser une application en langage java en groupe permettant ainsi d'évoluer en travail d'équipe et de réaliser un moteur de rendu 3D.
- Nous avons un contenu limité avec l'affichage d'une forme unique qui est la sphère. D'autres formes auraient pu être implémenté tels que des cônes, des cylindres ou des cubes.
- De nombreuses fonctionnalités aurait pu être ajouté au projet tels que la rotation de la caméra et des acteurs, la réflexion des objets entre eux ou l'ajout de transparence des objets (pour recrée des matière comme le metal avec une multiple réflexion ou du verre pour la transparence.