

Rapport d'IN55

Génération d'objets paramétriques

BALLET Nicolas & CORTIER Benoît & LAZARE Lucas

Le June 15, 2018

Contents

1	Introduction	2
2	Utilisation du logiciel	3
3	Conception du logiciel	4
3.1	Objet Paramétrique	6
3.2	Lumières	6
3.3	Fenêtre	6
4	Génération d'objet paramétrique : le diamant	7
5	Éclairage à l'aide de la méthode de Phong	9
6	Réflexions et réfractions sur le diamant	10
7	Conclusion	14

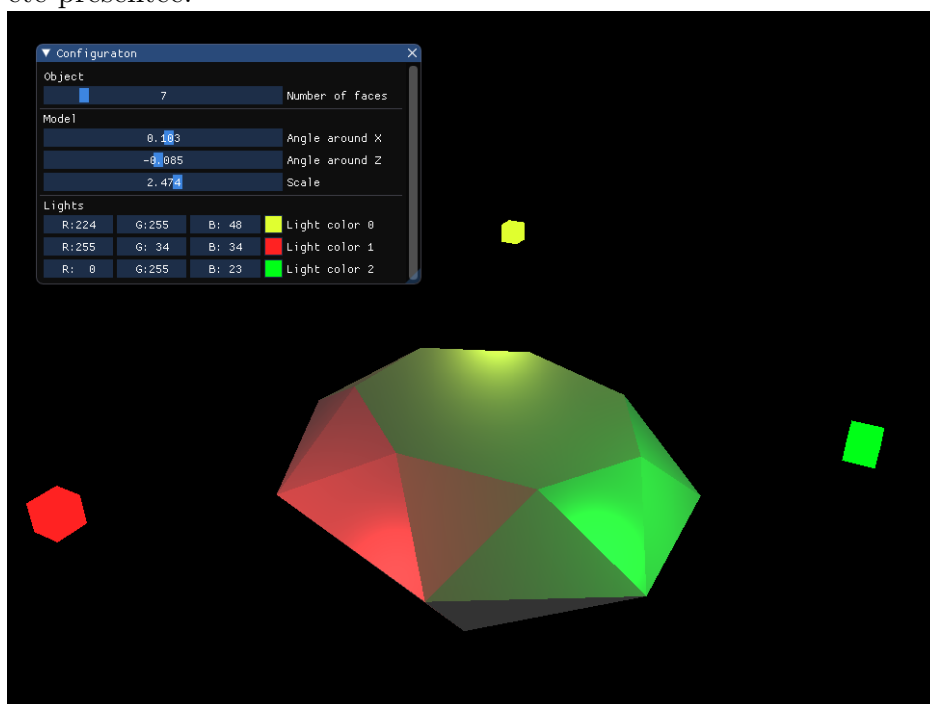
1 Introduction

Pour ce projet d'IN55, nous avons choisi le sujet de *génération d'objets paramétriques*.

Dans le cadre de ce sujet, nous avons fait le choix de nous concentrer sur le paramétrage et le rendu 3D d'un *diamant*. Dans un premier temps, le maillage du diamant et le système de lumière a été implémenté. Il est possible de configurer le nombre de faces inférieures du diamant ainsi que les lumières projetées sur le diamant qui est opaque.

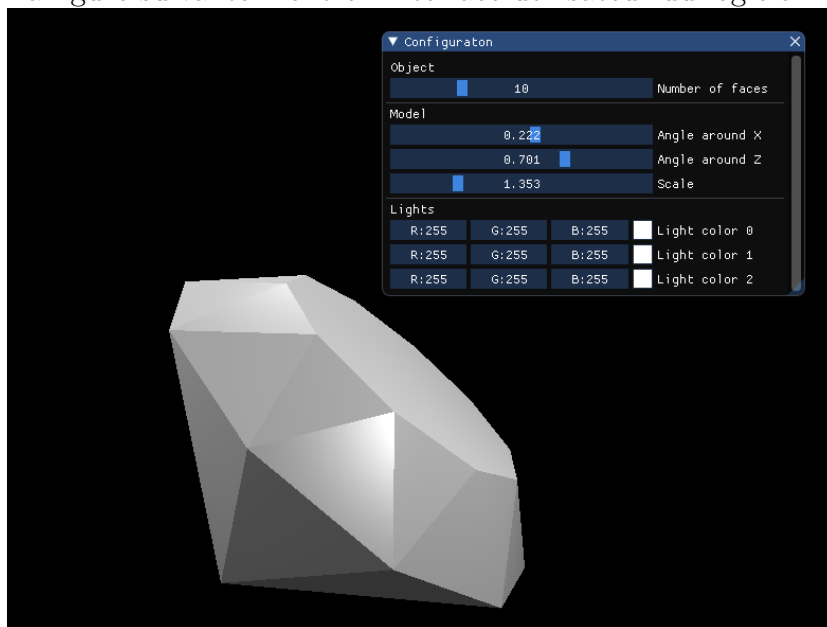
Dans un second temps, nous avons essayé de modéliser les réflexions en rendant le diamant transparent.

Ce rapport contient la notice d'utilisation du logiciel, en présente son architecture et détaille les différentes méthodes que nous avons mises en place. Le cœur de ce projet étant la génération paramétrique d'un diamant, cela concerne donc principalement la création algorithmique d'un maillage 3D, mais une piste de recherche pour faire des réflexions réalistes est aussi été présentée.



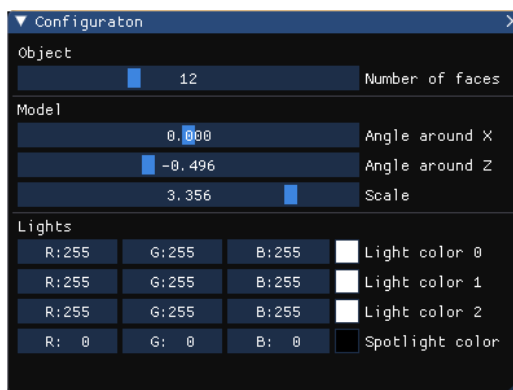
2 Utilisation du logiciel

La figure suivante montre l'interface utilisateur du logiciel.



Il est possible de déplacer la caméra à l'aide des touches WASD sur un clavier qwerty ou ZQSD sur un clavier azerty (le mappage est physique et ne dépend donc pas de la disposition du clavier). La touche espace permet de faire s'élever la caméra tandis que la touche contrôle permet l'abaisser.

Une fenêtre de configuration flottante permet de configurer la scène. Comme on peut le voir sur la figure, il est possible de configurer le nombre de faces inférieures du diamant, de changer son orientation et sa taille, mais aussi de configurer les lumières. Il est possible « d'éteindre » les lampes en réglant leur couleur sur noir. Par défaut, la torche accrochée à la caméra est éteinte.



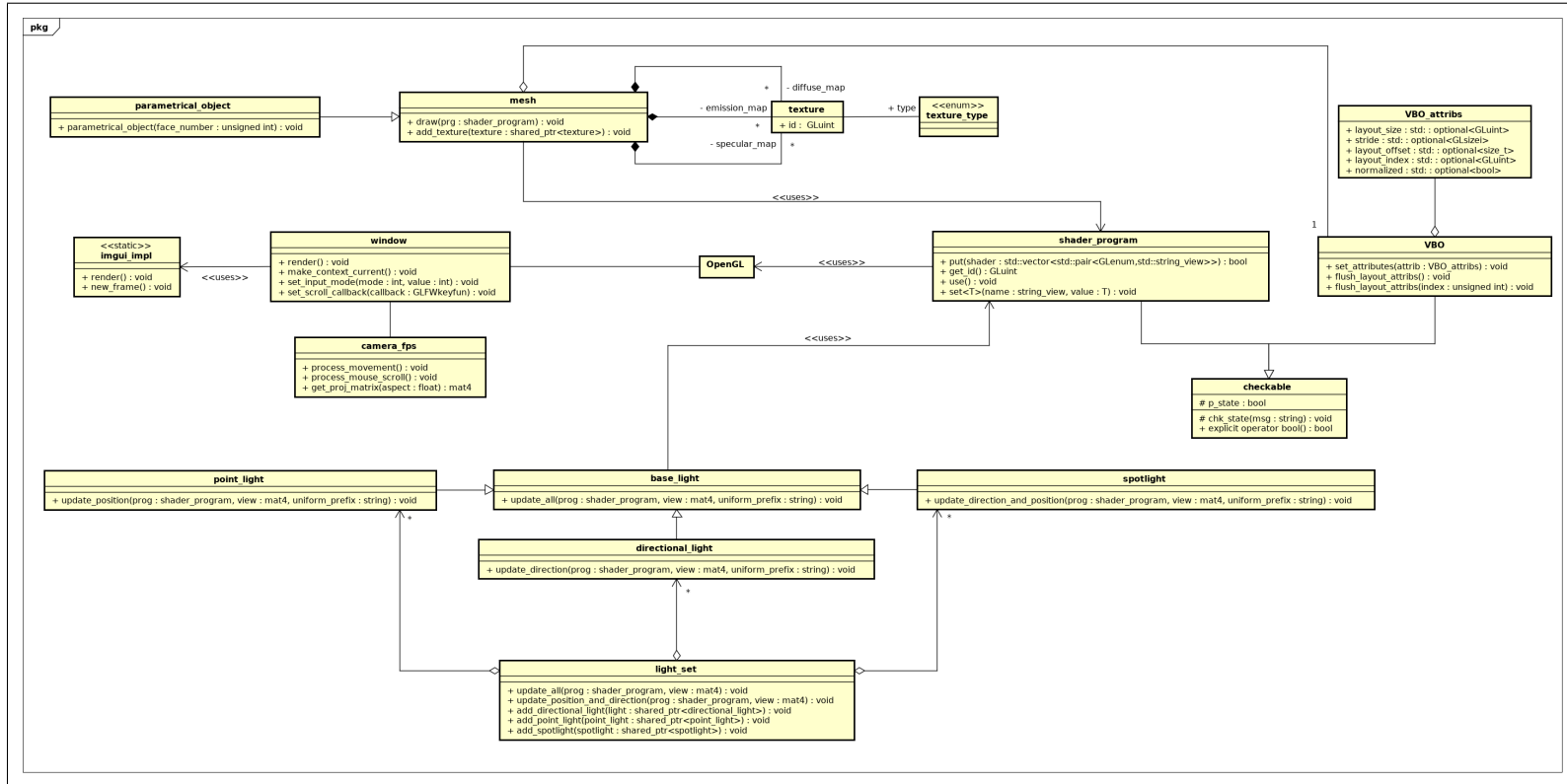
Lorsque le curseur n'est pas sur la fenêtre de configuration, cliquer et déplacer la souris permet d'orienter la caméra.

3 Conception du logiciel

Le logiciel se décompose en trois blocs principaux :

- L'objet paramétrique
- Les lumières
- La gestion des entrées/sorties (fenêtre)

On notera également l'existence des classes *VBO* et *shader_program*, qui sont de simples wrappers autours d'OpenGL pour faciliter son utilisation (en gérant par exemple les vérifications d'erreur et la libération des ressources)
[voir UML page suivante]



3.1 Objet Paramétrique

L'objet paramétrique est construit à partir d'un entier n représentant le nombre de faces sur la partie inférieure d'un diamant. À l'instanciation, il génère un tableau de vertices et un d'indices représentant le-dit diamant.

3.2 Lumières

Les lumières sont découpées en types :

- *point_light* qui diffuse une lumière autours d'un point
- *directional_light* qui génère une lumière directionnelle unie (façon soleil)
- *spotlight* qui émet une lumière en cône à partir d'un point

La classe *light_set* est une simple classe utilitaire utilisée pour gérer tous les éclairages à la fois

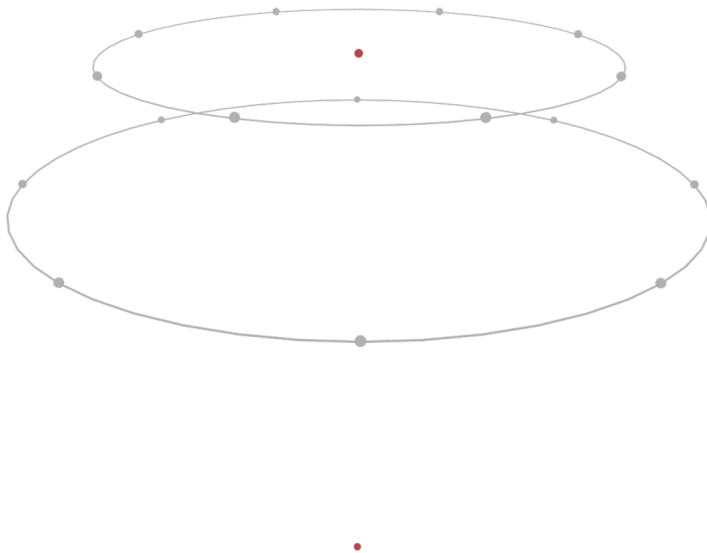
3.3 Fenêtre

La classe *window* est une classe qui nous permet d'initialiser les différentes bibliothèques gérant l'affichage :

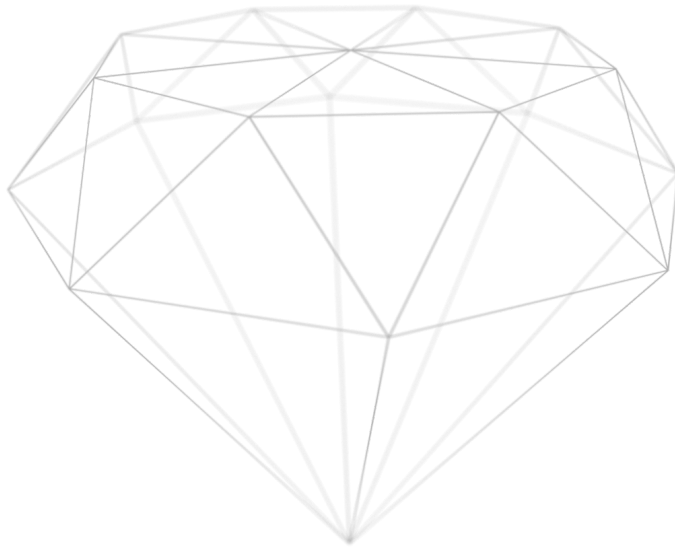
- *imgui*, qui gère les boutons, les barres de défilement, ...
- *glad*, qui s'occupe de charger OpenGL
- *glfw*, qui permet d'ouvrir la fenêtre et de récupérer les entrées clavier/souris

4 Génération d'objet paramétrique : le diamant

Premièrement nous nous sommes attaqué au maillage des diamants. L'algorithme se fait donc en deux étapes, la première étant la génération des points dans l'espace. On considère pour cela deux sommets aux extrémités du diamant, ainsi que 2 cercles sur lesquels seront répartis un certain nombre de points, dépendamment du nombre de facettes désiré.

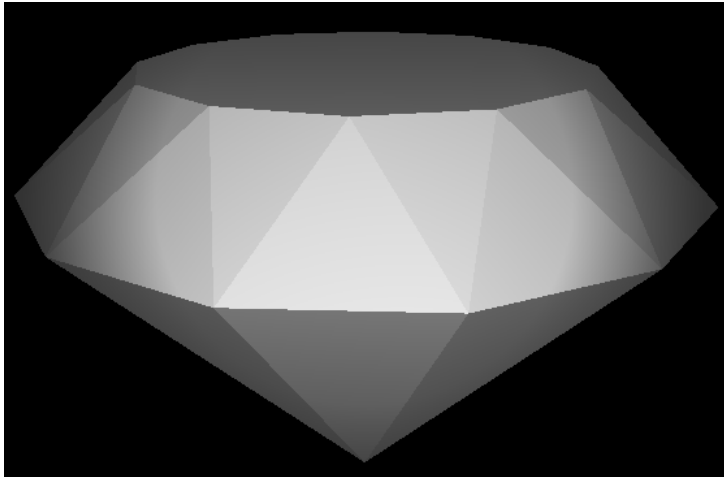


Une fois les points placés, il ne reste plus qu'à les relier avec des triangles simples, étant donné qu'hormis les triangles de la face supérieure, tous les autres triangles auront des normales différentes.



Il ne reste plus qu'à calculer les normales de chaque face à l'aide d'un produit vectoriel et en choisissant bien les vecteurs.

La figure suivante montre résultat final avec le système d'éclairage de Phong que nous allons présenter dans la section précédente :



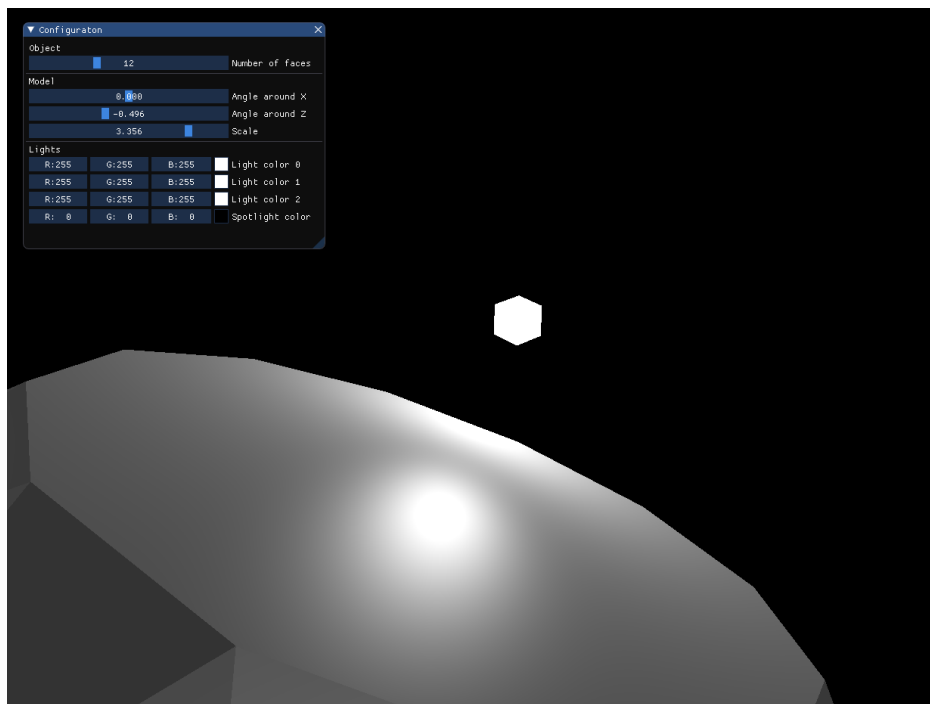
5 Éclairage à l'aide de la méthode de Phong

Pour gérer l'éclairage nous utilisons la méthode de Phong :

- Composante ambiante.
- Composante diffuse.
- Composante spéculaire.

De plus différents types de lumières sont également implémentés :

- Lumière directionnelle.
- Lumière localisée.
- Spot.



Notre implémentation permet de gérer un nombre arbitraire de lampes selon des constantes définies dans le fragment shader (actuellement, arbitrairement configuré sur 5 lumières directionnelles, 15 lumières localisée et 15 spots au maximum).

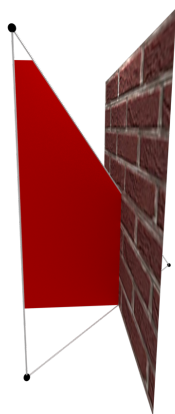
6 Réflexions et réfractions sur le diamant

Ici, le but était de trouver différentes méthodes de calculs de réflexion afin d'être le plus proche de la réalité possible.

Une première approche simpliste mais rapide à mettre en œuvre était l'utilisation d'une texture cubique. Cette méthode est très rapide et permet un rendu de qualité raisonnable pour des objets simples. Or, les diamants comportant des réflexions internes, cette méthode n'était pas très convaincante. Un rendu cubique autour de chaque face pouvant apporter des artefacts visuels sur des faces adjacentes et proches.

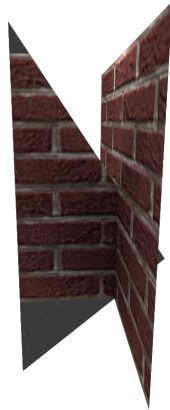
Nous avons donc creusé dans une autre direction. Le but est de générer, pour chaque triangle de la scène, un rendu de son monde « miroir » dans un buffer commun à toute la scène.

Voici le but recherché :



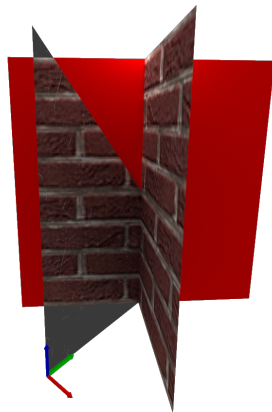
avec en rouge la zone où le rendu de la réflexion sera effectué.

Nous avons dans cet exemple, un carré, et un miroir triangulaire tourné à 45° .

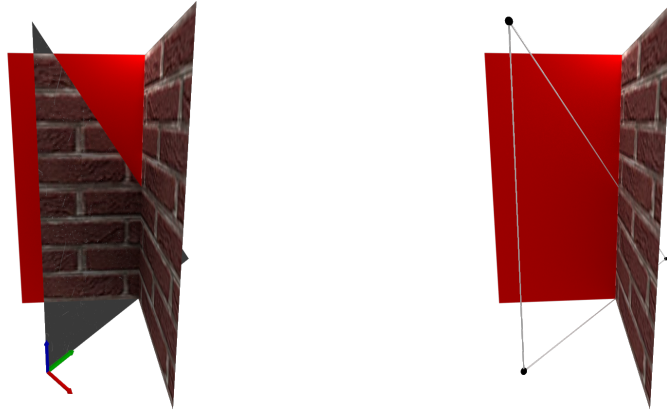


Pour chaque triangle se comportant comme un miroir :

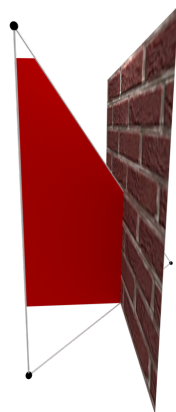
- On récupère sa matrice Tangente - Bitangente - Normale (TBN) afin d'effectuer la réflexion de la scène par rapport a ce triangle.



- On supprime ce qui se trouve derrière le carré de mur, étant donné qu'une fois la réflexion effectuée, c'est uniquement ce qui se trouve devant qui représente le reflet.



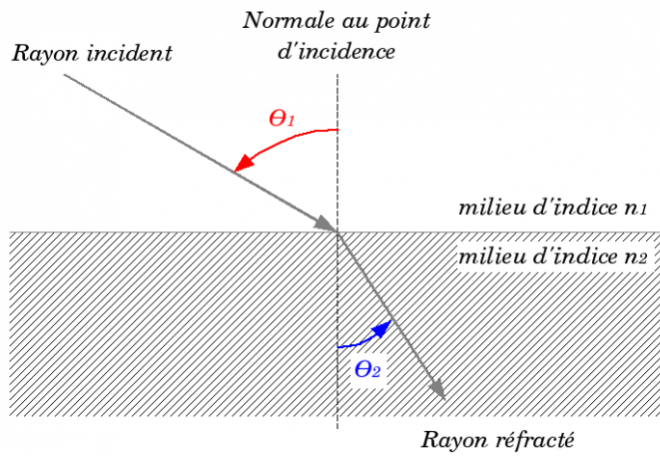
- Et pour finir, on ne garde que les pixels se trouvant à l'intérieur du triangle que l'on est en train de rendre.



Il reste les réfractions.

Ici, il y a une étape en plus, car la lumière est dispersée, mais aussi décomposée suivant l'angle d'incidence et le matériau.

On a choisi ici de traiter l'image comme le ferait une vraie caméra : en trois canaux de couleur. On effectue chaque rendu trois fois, ou on rendra de manière séparée, le bleu, le vert, et le rouge. Chaque réfraction dépendra donc de la matrice TBN de la surface, de son indice de réfraction, qui est le rapport entre l'angle d'incidence et l'angle réfracté, et aussi de la longueur d'onde de la couleur en train d'être rendue.



Les étapes suivantes sont les mêmes que pour la réflexion.

Ceci étant l'étape triviale de notre algorithme, car il faut maintenant faire la récurrence afin de voir les réflexions/réfractions sur plusieurs faces (la profondeur de réflexion étant paramétrable).

Ce n'est pas une étape à prendre à la légère, il faut effectuer un rendu des réflexions/réfractions pour chaque niveau de récursivité, et ainsi combiner les textures lors d'un double reflet (Ex: un miroir étant reflété par un autre) en prenant l'angle d'incidence en compte via l'utilisation du Fresnel.

Cette méthode est très coûteuse en termes de performance, mais notre but étant le réalisme, cela est satisfaisant.

7 Conclusion

Dans ce projet, nous avons généré le maillage d'un diamant, implémenté un système d'éclairage et nous avons tenté de pousser plus loin avec un système de réflexions et de réfractions bien que ces dernières ne soient pas encore fonctionnelles. Nous pensons avoir atteint les objectifs imposés par le sujet, cependant il reste encore de nombreuses pistes d'améliorations possibles dont :

- plus de paramètres sur la génération de l'objet (taille du cercle, ajout de couches, etc).
- rendre fonctionnel le système de réfraction et implémenter la récursion réflexion/réfraction.
- faire un système de lumière plus poussé (par exemple avec le PBR, « physically based rendering »).