

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
4.1	Génération des points	7
4.2	Génération des faces	7
5	Éclairage à l'aide de la méthode de Phong	9
6	Réflexions et réfractions sur le diamant	10
6.1	Réflexions	10
6.2	Réfractions	12
6.3	Résultat actuel	13
7	Conclusion	15

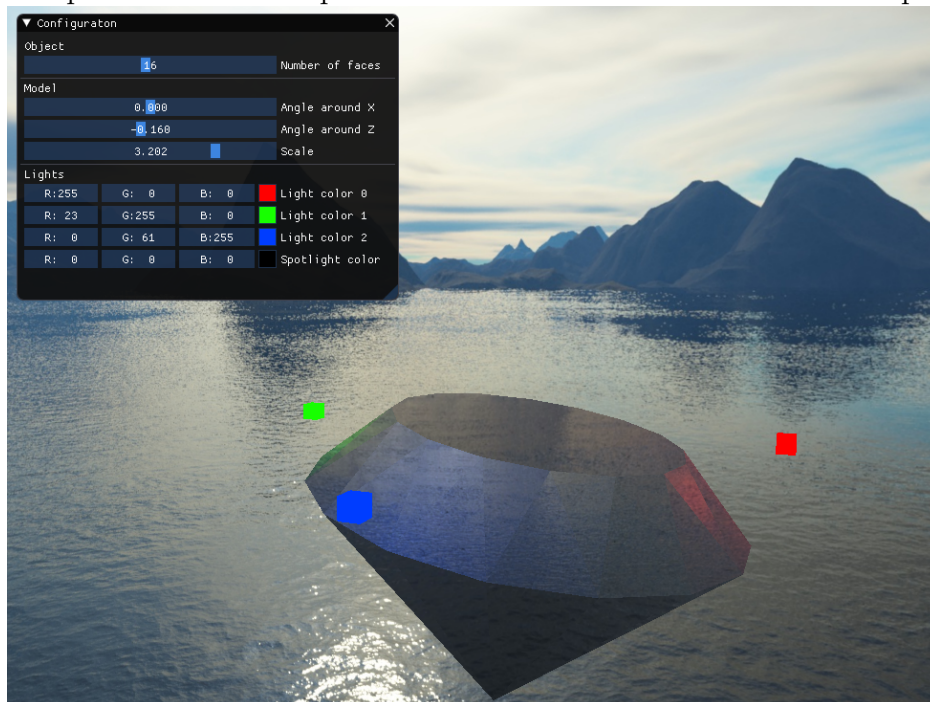
1 Introduction

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.

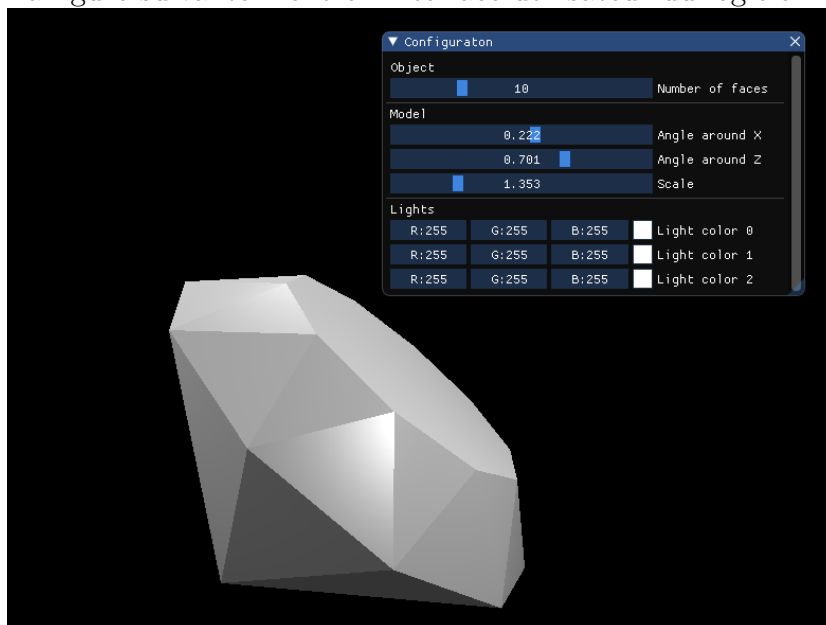
Dans un second temps, nous avons essayé de modéliser les réflexions en rendant le diamant transparent.

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, et concerne donc principalement la création algorithmique d'un maillage 3D, mais une piste de recherche pour faire des réflexions réalistes est aussi présentée.



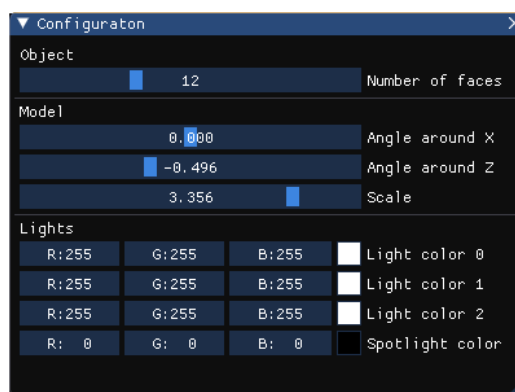
2 Utilisation du logiciel

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.



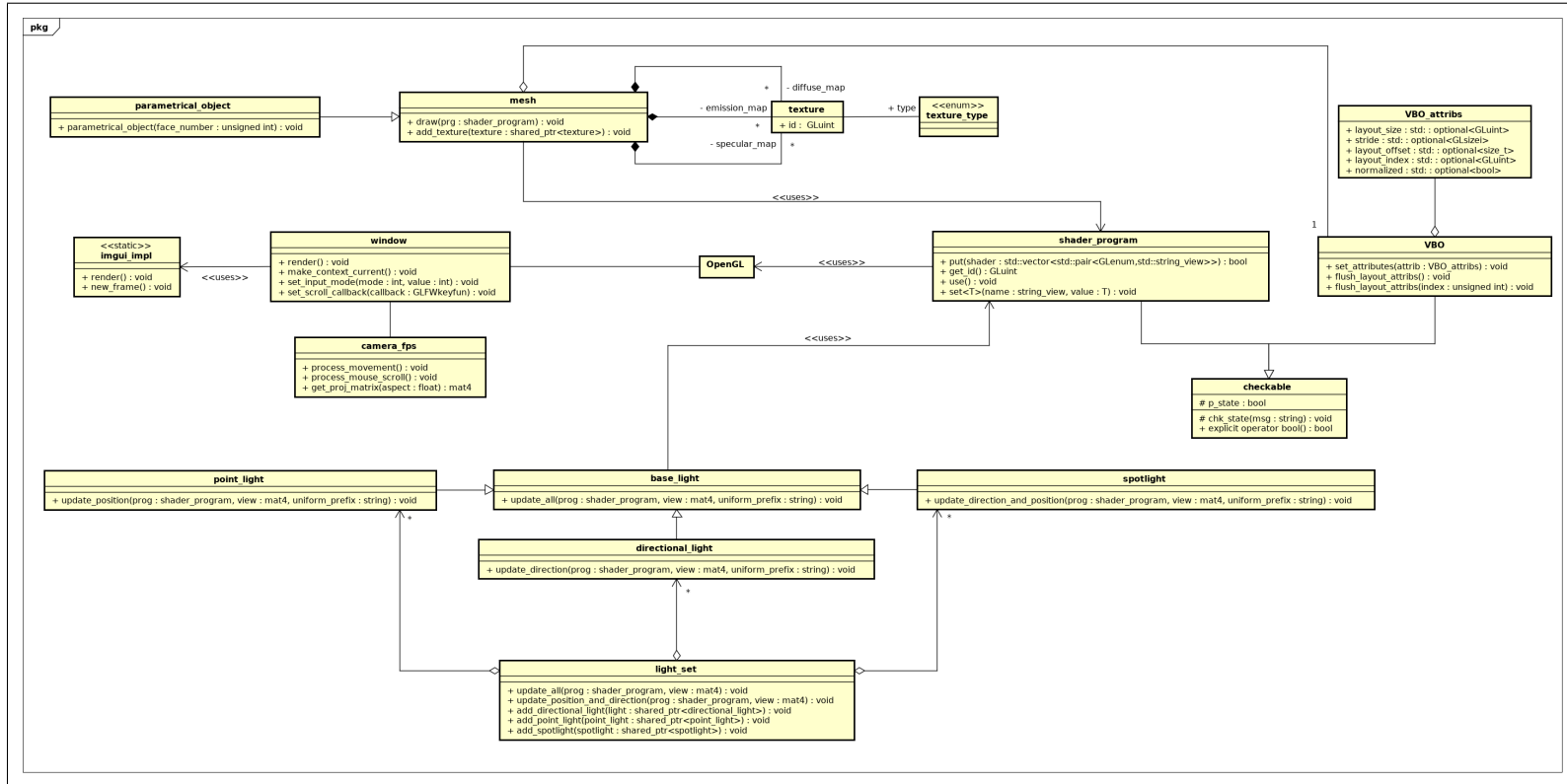
Lorsque le curseur n'est pas sur la fenêtre de configuration, cliquer et déplacer la souris permet d'orienter la caméra.

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

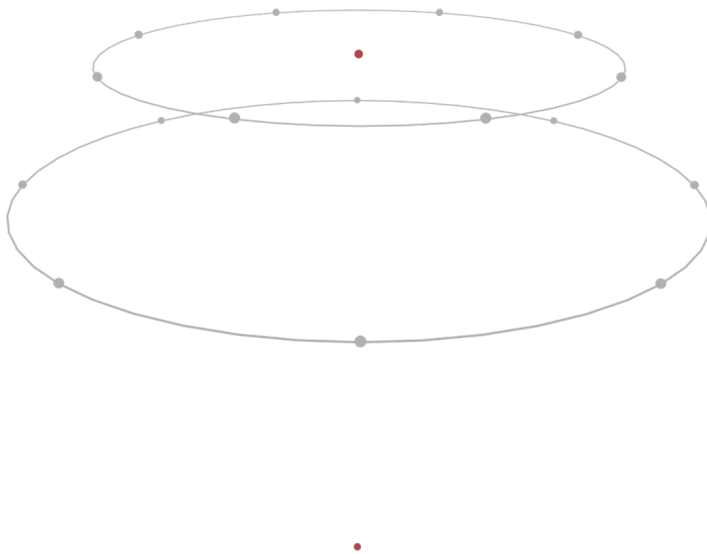
4 Génération d'objet paramétrique : le diamant

Dans un premier temps, nous nous sommes attaqué au maillage des diamants. Pour se faire, nous avons découpé l'algorithme en deux étapes principales:

- Génération des points dans l'espace
- Génération des faces

4.1 Génération des points

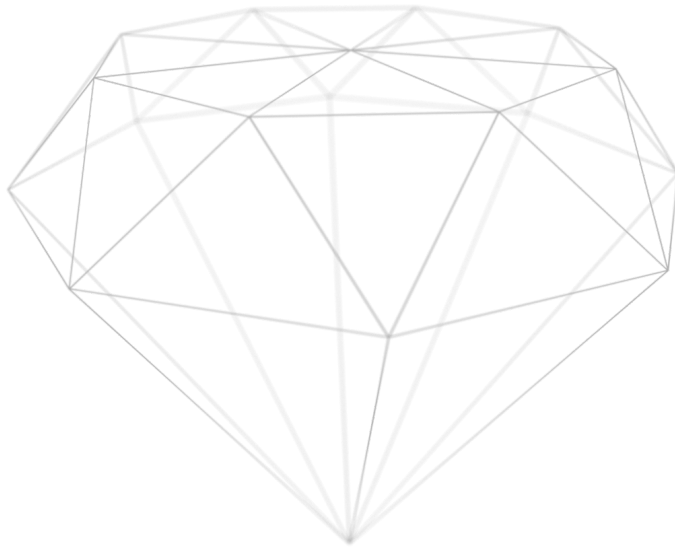
Pour générer les points du diamant, on considère deux sommets aux extrémités du diamant, ainsi que 2 cercles sur lesquels les points seront répartis, tout cela en fonction du nombre de facettes désiré.



Une fois les points placés, il ne reste plus qu'à les relier.

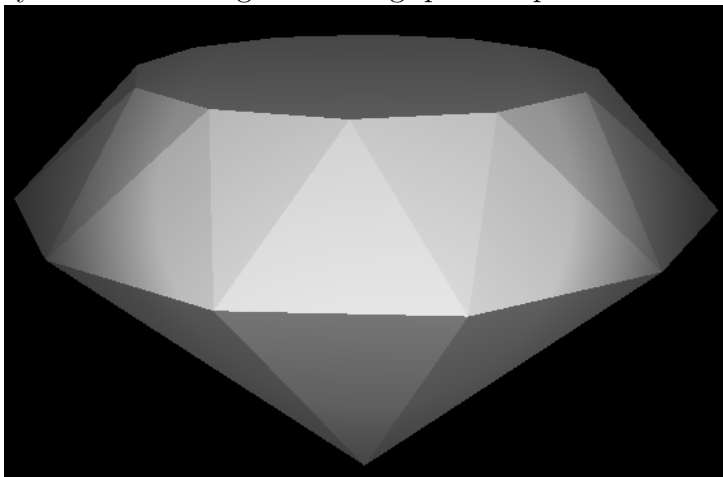
4.2 Génération des faces

Pour relier les dits points, on pourra se contenter de simples triangles, étant donné qu'en dehors des triangles de la face supérieure, tous les autres triangles auront des normales différentes.



Il ne nous reste alors plus qu'à calculer les normales de chaque face, ce que l'on peut faire à l'aide d'un produit vectoriel, en choisissant bien ses vecteurs.

La figure suivante est une petite démonstration du résultat final, avec le système d'éclairage de Phong qui sera présenté dans la prochaine section :



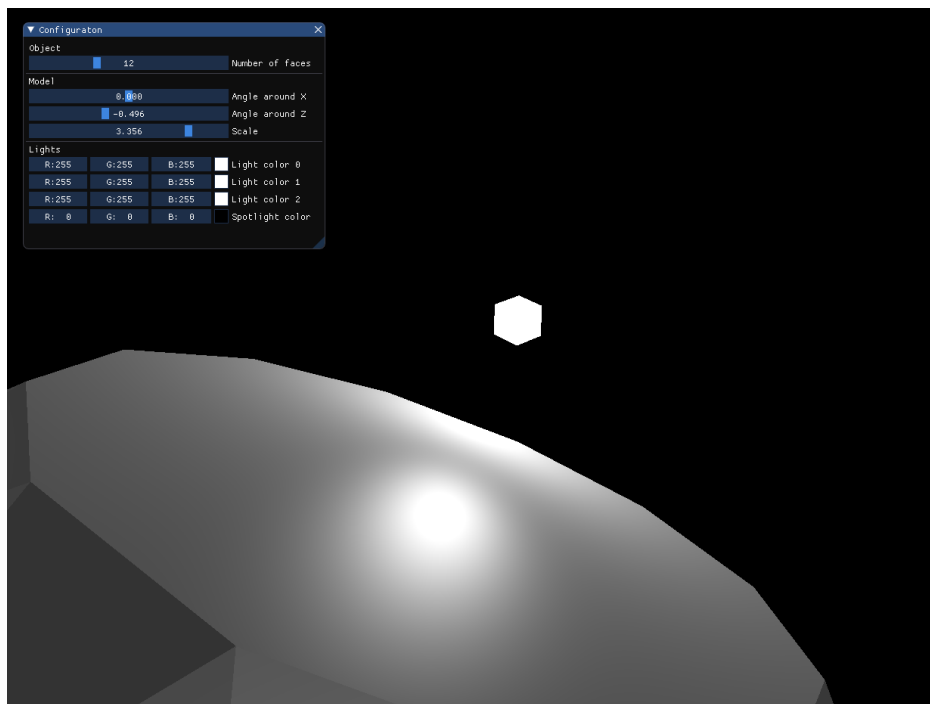
5 Éclairage à l'aide de la méthode de Phong

Pour gérer l'éclairage nous utilisons la méthode de Phong, qui divise les lumières en trois catégories :

- composante ambiante
- composante diffuse (gestion de la diffusion de la lumière au sein d'une face)
- composante spéculaire (gestion des reflets).

Au sein de ce système, nous avons implémenté trois différents types de lumières :

- lumière directionnelle
- lumière localisée
- projecteur.



Notre implémentation nous permet de gérer un nombre arbitraire de lampes, le chiffre exact étant défini par des constantes se trouvant dans le *fragment shader* (configuré de manière arbitraire sur 5 lumières directionnelles, 15 lumières localisées et 15 spots au maximum).

6 Réflexions et réfractions sur le diamant

6.1 Réflexions

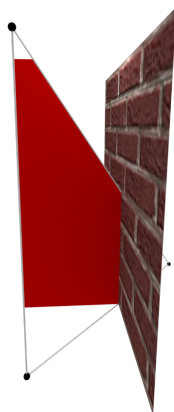
L'objectif de cette partie était de trouver différentes méthodes de calcul de réflexion, afin d'avoir le meilleur réalisme visuel possible.

Dans une première approche, certes simpliste, mais rapide à mettre en œuvre, nous avons utilisé une simple texture cubique. Cette méthode est extrêmement peu coûteuse en performance, mais permet tout de même un rendu de qualité raisonnable pour des objets simples.

Malheureusement, les diamants comportant des réflexions internes, cette méthode ne permet pas de rendu convaincant, un rendu cubique autour de chaque face pouvant apporter des artefacts visuels sur des faces adjacentes et proches.

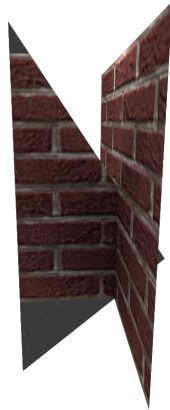
Nous avons donc creusé dans une autre direction, en générant, pour chaque triangle de la scène, un rendu de son monde « miroir », dans un buffer commun à toute la scène.

Le but final est d'obtenir quelque chose ressemblant à ça :



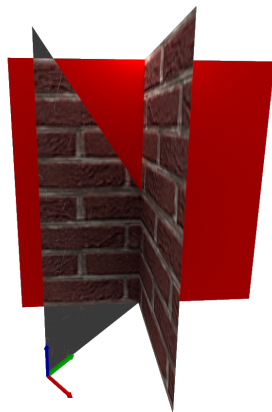
avec, en rouge, la zone où le rendu de la réflexion sera effectué.

Nous avons dans cet exemple un carré avec un miroir triangulaire tourné à 45° par rapport au mur.

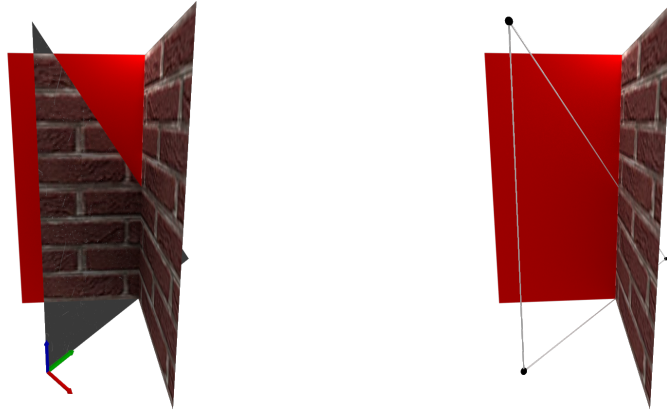


Pour chaque triangle se comportant comme un miroir on calcul la réflexion de la manière suivante :

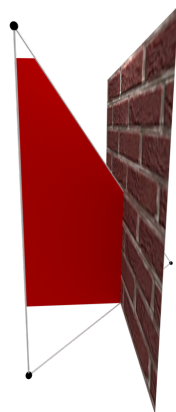
- On récupère sa matrice Tangente – Bitangente – Normale (TBN), qui nous permet d'effectuer la réflexion de la scène par rapport au triangle.



- On supprime ce qui se trouve de l'autre côté du mur ; en effet, une fois la réflexion effectuée, c'est uniquement la partie à l'avant qui représente le reflet.



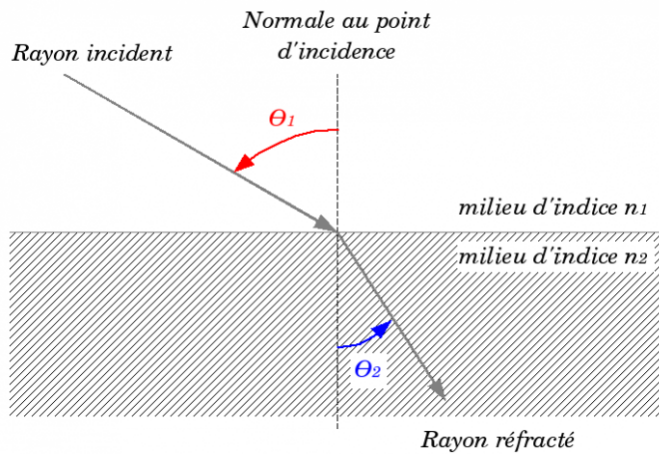
- Et pour finir, on ne garde que les pixels se trouvant à l'intérieur du triangle que l'on est en train de rendre.



6.2 Réfractions

Pour calculer les réfractions, il y a une étape supplémentaire car la lumière est dispersée, mais est également décomposée, en fonction de l'angle d'incidence et du matériau.

On a choisi ici de traiter l'image comme le ferait une vraie caméra, c'est à dire avec trois canaux de couleur. On peut alors soit effectuer chaque rendu trois fois, soit rendre de manière séparée le bleu, le vert, et le rouge. Chaque réfraction dépendra donc de la matrice TBN de la surface, et de son indice de réfraction (le rapport entre l'angle d'incidence et l'angle réfracté), mais également de la longueur d'onde de la lumière en train d'être rendue.



Les étapes suivantes pour cette partie sont les mêmes que pour la réflexion.

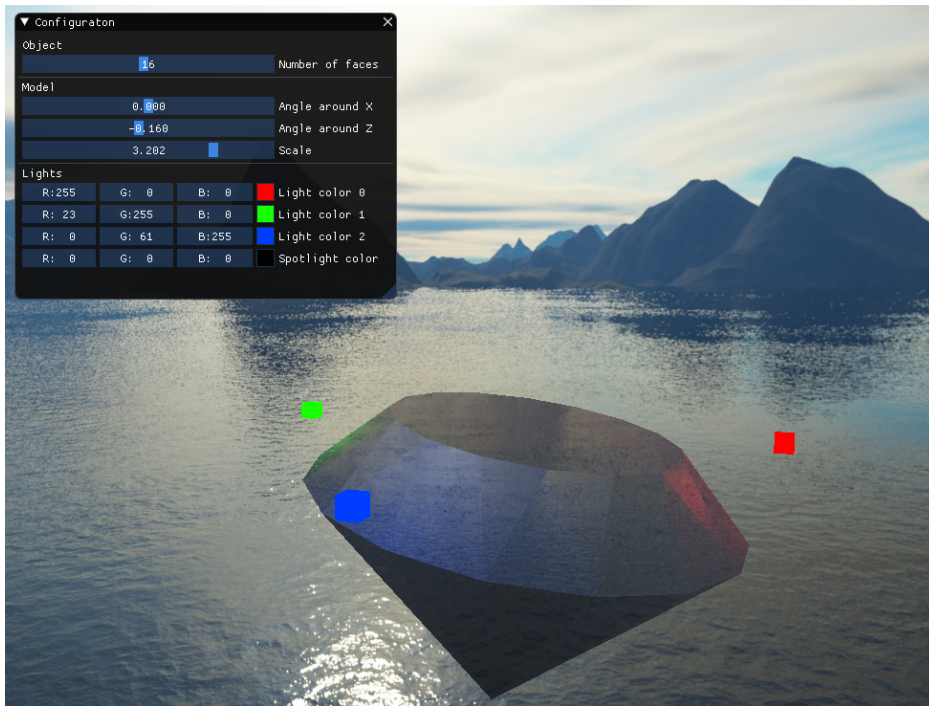
Il suit ensuite des étapes plus complexes, qui n'ont malheureusement pas été implémentées. En effet, il reste les étapes de récurrences, qui permettraient de voir les réflexions/réfractions se propager sur plusieurs faces, la profondeur de réflexion étant paramétrable.

Ces rendus supplémentaires ne sont pas triviaux, car un rendu de réflexion et de réfraction supplémentaire est nécessaire pour chaque niveau de récursivité, combinant les textures lors d'un reflet double (comme par exemple un miroir qui se reflète dans un autre miroir), le tout en prenant en compte l'angle d'incidence, via l'utilisation du Fresnel.

Cette méthode est très coûteuse en termes de performance, mais notre but étant le réalisme, elle pourrait s'avérer satisfaisante.

6.3 Résultat actuel

Nous n'avons pas eu le temps d'implémenter la réfraction et la récursivité, mais la réflexion est bien fonctionnelle dans le logiciel. La figure suivante montre le résultat.



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 »).