

Illumination

5 ETI – CPE Lyon

Majeure Image, modélisation et informatique

Noms, Prénoms : Simon Jeffrey, Rakotoarivony Aurore

Date : 02/12/2023

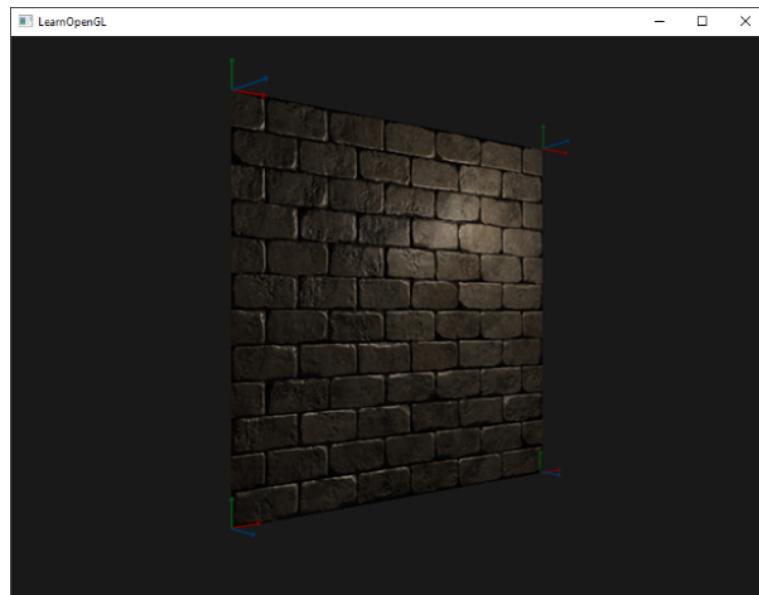


FIGURE 1 – <https://learnopengl.com/Advanced-Lighting/Normal-Mapping>

1 Introduction

Dans le domaine de l'informatique graphique, l'obtention d'images réalistes est importante. Au cours de ce TP, nous plongerons dans divers aspects du rendu moderne, incluant l'application de l'illumination de Phong, l'utilisation de plusieurs textures dans un même shader, la cartographie d'occlusion ambiante, l'utilisation de cartes d'élévation pour la modification de géométrie, et l'utilisation de l'espace tangent pour des effets tels que le "normal mapping" et, en option, le "parallax mapping".

Les objectifs principaux de ce TP sont de comprendre les subtilités de l'illumination de Phong, d'exploiter la puissance de plusieurs textures pour des détails visuels plus riches, d'utiliser des cartes d'occlusion ambiante pour simuler un ombrage réaliste, de tirer parti des cartes d'élévation pour modifier dynamiquement la géométrie, et d'explorer les transformations dans l'espace tangent pour des techniques de mappage avancées. À travers l'expérience pratique et l'expérimentation, nous visons à comprendre l'impact de ces techniques sur la qualité visuelle des scènes rendues.

Pour illustrer ces concepts, nous travaillerons avec une variété de textures, chacune portant des propriétés uniques. Ces propriétés seront utilisées de manière stratégique pour mettre en évidence la profondeur et les subtilités de nos objets rendus, veillant à ce que notre représentation maillée ne paraisse ni simpliste ni irréaliste.

L'utilisation des techniques avancées d'éclairage permettent d'éviter d'utiliser des maillages trop complexes dans lesquelles il faudrait faire des calculs sur des milliers de triangles.

2 Bind des Textures

L'étape de *binding* des textures dans un programme OpenGL est importante pour associer des images aux objets 3D, permettant ainsi un rendu réaliste. Voici une explication détaillée du processus :

1. **Activation de l'unité de texture :** Avant de charger une texture il faut l'activer. On commence donc par activer la texture *TEXTURE0* à l'aide de la commande `glActiveTexture`.
2. **Chargement des textures :** Ensuite on vient charger la texture à l'aide de la fonction `load_texture`.
3. **Association des textures au programme :** Une fois chargée, on envoie la texture aux différents shader en la passant sous la forme d'une variable uniforme
4. **Utilisation des textures dans le shader :** Dans le shader, on déclare une variable uniforme pour utiliser les textures associées aux unités de texture spécifiées.

Ensuite on répète ces étapes pour les différentes textures que l'on veux utiliser dans le programme.

3 Role des différentes textures

Notre boule n'est pas une surface lisse. Elle a des reliefs, des creux. Cette partie consiste à expliquer comment des textures peuvent exprimer différentes quantités physiques de la boule pour mettre en valeur son illumination , son réalisme et ses reliefs.

3.1 Sans ajout : illumination phong Classique

Le modèle d'illumination de Phong comprend trois composantes : ambiante, diffuse et spéculaire. La composante ambiante représente une lumière uniforme dans toutes les directions, éclairant les surfaces même dans les zones ombragées. La composante diffuse simule la réflexion uniforme de la lumière sur une surface, dépendant de l'angle entre la source lumineuse et la surface. Enfin, la composante spéculaire crée des reflets brillants sur une surface, en fonction de l'angle entre la caméra et la lumière réfléchie.

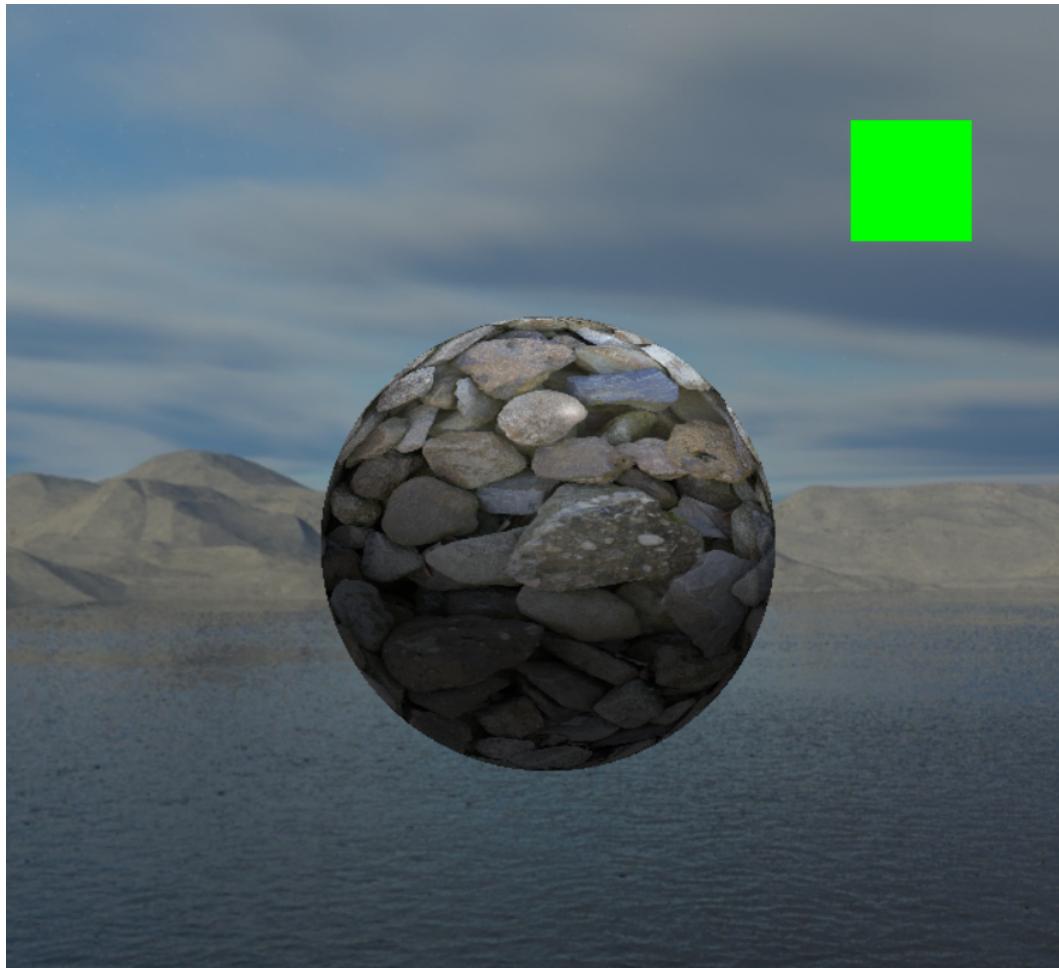


FIGURE 2 – texture illumination Phong

3.2 Ambient occlusion

Dans le contexte de l'illumination avec le modèle de Phong, l'occlusion ambiante (ambient occlusion) est une technique qui permet de simuler l'atténuation de la lumière ambiante en fonction de la proximité des objets dans une scène 3D. Cela crée des zones plus sombres là où il y a des trous, donnant ainsi une apparence plus réaliste à la scène.

On va donc modifier l'illumination ambiante en la multipliant par la carte d'occlusion ambiante que l'on a récupérée sous forme de variable uniforme depuis le fragment shader.

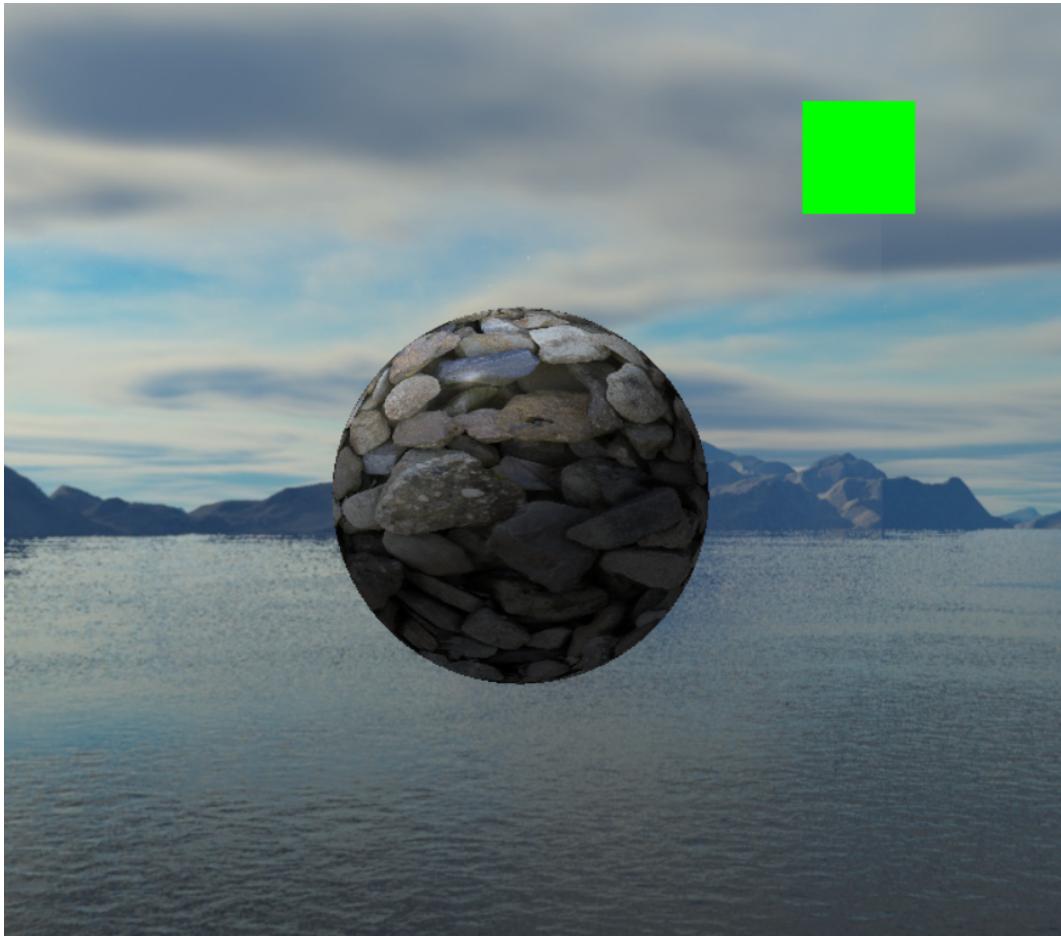


FIGURE 3 – occlusion ambiante

On observe bien plus de contraste sur cette image que la précédente, notamment au niveau des zones d'ombres.

3.3 Carte des hauteurs

On souhaite simuler des reliefs sur notre surface. La carte des hauteurs (texture2) contient des informations sur la variation de la hauteur des surfaces en fonction de la vue de la caméra. On l'utilise dans le vertex shader pour déplacer les sommets en direction de la normale selon la carte des hauteurs.

On applique ensuite l'illumination de Phong sur les nouvelles coordonnées pour calculer l'éclairage du fragment. Les différentes composantes de l'illumination de Phong sont influencées par la texture de couleur (texture1) et la normale obtenue à partir de la carte des normales (texture3).

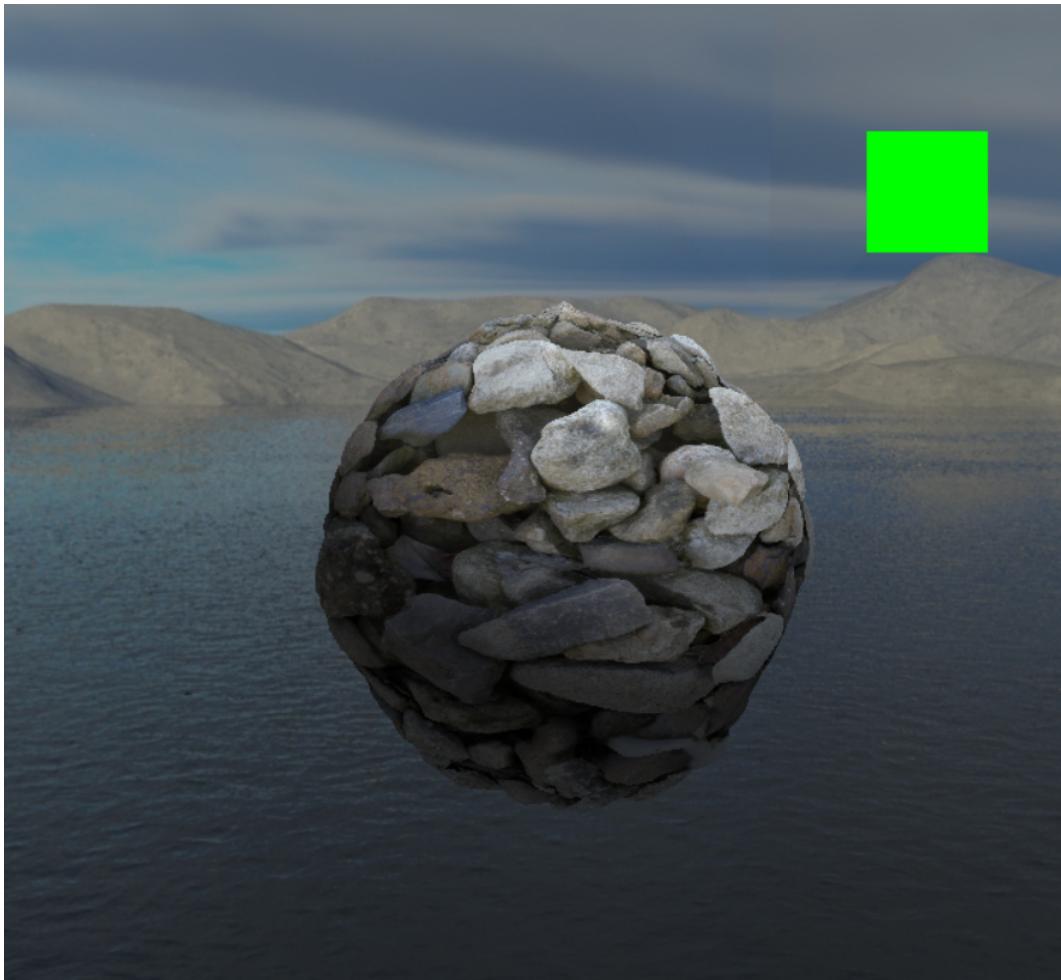


FIGURE 4 – Carte des hauteurs

Donc la carte des hauteurs permet de modifier les coordonnées des textures pour améliorer le rendu visuels et l'illumination des surfaces.

3.4 Mapping

La surface n'est pas une boule parfaire. Elle possède des creux, des discontinuités, des bosses. Ceux ci influencent les normales. En effet, un creux va voir ses normales vers une certaine direction. Voyons les techniques de Mapping.

3.4.1 Espace tangent

Dans la carte de normales, les vecteurs normaux que nous avons utilisés pointent tous approximativement dans la direction positive de l'axe z. Cependant, ces vecteurs ne tiennent pas compte de l'orientation réelle de l'objet. Ainsi, pour remédier à cela, nous définissons nos normales dans l'espace tangent. Dans cet espace, les vecteurs normaux sont toujours orientés approximativement dans la direction positive de l'axe

z. L'espace tangent, étant local à la surface d'un triangle, permet d'exprimer les normales par rapport au repère local des triangles individuels.

Pour formaliser cette transformation, nous utilisons une matrice TBN (Tangent, Bitangent, Normal). Cette matrice est cruciale car elle établit le lien entre les vecteurs dans l'espace tangent et les vecteurs dans l'espace global.

Ainsi, la matrice TBN agit comme une sorte de pont entre les deux espaces, assurant une cohérence dans l'orientation des normales par rapport à la surface réelle de l'objet.

3.4.2 Normal Mapping

La normal mapping consiste à déformer l'orientation des normales afin d'inclure des effets de lumières tel que des creux pour donner l'illusion d'une surface moins lisse. Notons que dans les textures, les normales sont comprises entre 0 et 1. Nous devons les transformer dans le shader pour qu'elles soient comprise entre -1 et 1. Après avoir associé la bonne normale de la texture map au fragment, nous l'utilisons pour calculer l'illumination de phong. Nous devons convertir toute les autres grandeurs dans l'espace tangent pour assurer la consistance. Nous faisons ces opérations dans le vertex shader.



FIGURE 5 – normal mapping

Nous voyons que les effets étranges de l'illumination de Phong, tels que la grosse valeur spéculaire, ont disparu donnant l'impression d'une texture réelle et non d'une boule de plastique. On a l'impression de voir les imperfections de la roche.

3.4.3 Parallax Mapping

Le "parallax mapping" est une technique de rendu qui, au lieu de simplement prendre en compte le déplacement des sommets d'une surface basée sur une carte de hauteurs, ajuste également les coordonnées de texture pour tenir compte de la carte de hauteurs par rapport à l'orientation de la caméra.

En d'autres termes, cette technique simule l'illusion d'une géométrie complexe en modifiant les coordonnées de texture en fonction de la perspective de la caméra par rapport à la surface texturée. Cela crée l'illusion d'un relief plus prononcé et détaillé, même si la géométrie réelle reste relativement simple. Cette technique marche bien pour les textures structurées tel que les briques. Mais nous n'avons pas réussi à l'implémenter.



FIGURE 6 – parallax mapping

Nous voyons des effets étranges sur notre parallax.

```
vec2 ParallaxMapping(vec2 texCoords, vec3 viewDir)
{
    float height = texture(depthMap, texCoords).r;
    vec2 p = viewDir.xy / viewDir.z * (height * height_scale);
    return texCoords - p;
}
```

Nous effectuons la transformation des coordonnées textures selon les explications du tuto de learnopengl. Les résultats sont incohérents.

3.5 Ajout lumière

La fonction Phong dans le shader fragment a été modifiée pour prendre en compte deux sources lumineuses distinctes, chacune avec sa propre direction et couleur. Les contributions ambiantes, diffuses et spéculaires de chaque source lumineuse sont calculées séparément et additionnées. Les couleurs des deux sources lumineuses sont différencierées, permettant une illumination diversifiée des surfaces. Cette modification enrichit le modèle d'ombrage Phong en offrant une représentation plus réaliste des interactions lumineuses sur les objets, avec des teintes variées provenant de deux sources lumineuses distinctes.

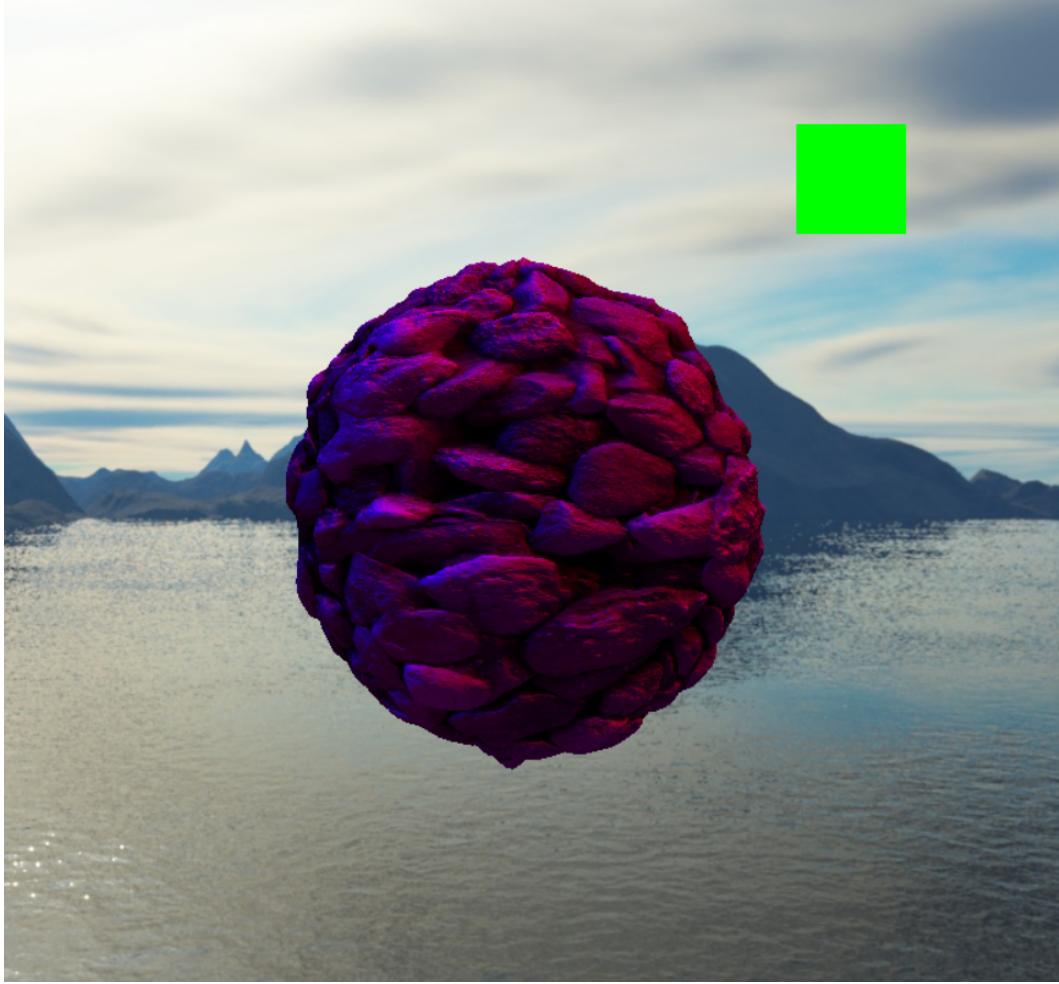


FIGURE 7 – 2 lumières rouge et bleue

3.6 Texture Cube pour décor

Nous souhaitons visualiser des environnements en utilisant des textures cube. Une texture cube est un ensemble de six textures associées à chaque face d'un cube. En plaçant la caméra à l'intérieur de ce cube texturé, nous créons l'illusion d'être entourés par un paysage tridimensionnel. La notion de cube map se distingue des textures 2D, bien que des paramétrisations similaires puissent être observées. Dans le shader, une cube map est traitée comme une texture conventionnelle. Cependant, OpenGL possède un mécanisme pour interpréter les coordonnées de texture spécifiques à cette cube map, qui contient un ensemble de six images au format PNG.

En termes pratiques, chaque face du cube est associée à une direction particulière dans l'espace tridimensionnel. Les coordonnées de texture dans le shader sont alors interprétées en fonction de ces directions spécifiques, permettant à la cube map de fournir des informations visuelles cohérentes à partir de n'importe quel point à l'intérieur du cube.

L'utilisation d'une cube map dans le contexte OpenGL offre une solution immersive pour la représentation d'environnements 3D, en particulier lorsqu'il est souhaitable de simuler des paysages complets ou des ciels panoramiques. Cela permet de créer des scènes où l'utilisateur semble être au centre d'un monde tridimensionnel, améliorant ainsi l'expérience visuelle globale.

3.7 Ajout bouton

On symbolise le bouton à l'aide d'un rectangle de manière classique avec un VAO et un VBO. On peut changer sa couleur en passant une variable uniform dans son fragment shader. On souhaite vérifier l'événement "la souris passe sur le bouton". Cependant les coordonnées OpenGL diffèrent des coordonnées glut on utilise donc une fonction pour inverser la transformation des coordonnées de la souris à l'écran en

coordonnées OpenGL. Cela nous permet d'utiliser la position de la souris. On vérifie la zone dans laquelle se situe le curseur. Si il est au bon endroit et que l'utilisateur clique sur le bouton on rentre dans nos conditions d'où on peu appeler diverse fonction.

4 Conclusion

A travers ce TP nous avons pu explorer divers aspects du rendu moderne pour obtenir des images réalistes. En se concentrant sur l'illumination de Phong, l'utilisation de plusieurs textures, la cartographie d'occlusion ambiante, l'intégration de cartes d'élévation pour modifier la géométrie, et l'exploration des espaces tangents pour des effets avancés tels que le "normal mapping" et le "parallax mapping". L'illumination de Phong a également servi à intégrer une deuxième source lumineuse avec des couleurs distinctes. Ce qui a permis une représentation plus réaliste des interactions lumineuses.

L'activation et le chargement des textures montrent l'importance du *binding* des textures dans un programme OpenGL pour associer des images aux objets 3D, améliorant ainsi le réalisme du rendu. L'utilisation de différentes textures a été mise en place pour exprimer des quantités physiques telles que l'illumination, les ombres, et les reliefs sur la surface des objets.

Enfin, l'introduction de textures cube a été abordée pour créer des environnements immersifs et la gestion d'un bouton interactif a été implémentée.