Projet de session – gestionnaire de scènes 2D / 3D

TP2

IFT-3100 – Infographie
Hiver 2022

Présenté à Philippe Voyer

Équipe 6
Guillaume Côté (536 853 701)
Hugo Chiasson (111 249 565)
Keven Lessard (111 266 993)

UNIVERSITÉ LAVAL

Faculté des sciences et de génie

Table des matières

Table des Figures :

1. Sommaire

Dans le cadre du TP2, nous avons amélioré notre application en ajoutant des contrôles sur la qualité visuelle des éléments. Pour ce faire, nous avons travaillé principalement 4 volets soit les textures, l'illumination classique, la Topologie ainsi que des éléments de l'illumination moderne. Au total ce sont 15 nouveaux critères fonctionnels qui ont été implémentés.

1.1 Textures

En ce qui concerne l'ajout de textures aux modèles, nous avons ajouté la gestion des coordonnées de textures, quatre types de filtrage par convolution et le mappage tonal. Nous avons également amélioré le rendu de la scène en ajoutant un « Skybox » pour répondre au critère en lien avec le cubemap.

1.2 Illumination classique

Les différents modèles d'illuminations ont été implémentés soit Lambert Gouraud, Phong et Blinn-Phong. Quatre matériaux ont été ajoutés dans les choix à appliquer aux différents modèles 3D. La scène peut également être égaillée par quatre types de lumières soit ambiantes, directionnelles, ponctuelles et projecteur. Bien entendu plusieurs instances de ces lumières peuvent être ajoutées à la scène.

1.3 Topologie

Dans les objets que nous pouvons ajouter à la scène, il y a maintenant une courbe paramétrique de Catmull-Rom ainsi qu'une surface paramétrique de Bézier bicubique. Un algorithme de tessellation a été implémenté sur un objet nommé Quads. Il est également possible de faire une triangulation de Delaunay à partir d'un ensemble de sommets que l'on génère à l'écran.

1.4 Illumination moderne

L'illumination des modèles se fait en appliquant un modèle PBR. Celui-ci applique une texture par défaut et c'est possible de nuancer le résultat par l'ajout d'un paramètre de métallicité ainsi qu'un paramètre de rugosité qui gère l'application d'une simulation de microfacettes.

Le TP2 a donc permis d'ajouter de la qualité visuelle aux fonctions pratiques ajoutées lors du premier travail. Ce rapport fait état de l'ensemble des critères fonctionnels qui ont été développés dans le cadre de ce livrable.

2. Technologie

Comme pour le travail pratique 1, le langage principal utilisé pour la réalisation de ce projet est le C++. Les Shaders eux sont écrits en GLSL. L'outil technologique principal est Open Framework en travaillant avec la version 3.3 d'OpenGL. Des modules complémentaires ont été ajoutés aux librairies standard de OpenFramework soit ofxGui, ofxAssimpModelLoader, ofxDelaunay ainsi que ofxReflexionRefraction .

2.1 Module ofxGui

Ce module complémentaire a permis de mettre en place les différents panneaux présents dans l'interface ainsi que l'ensemble de l'interactivité qui en découle. C'est ce module qui permet également l'ajout des étiquettes d'identification, des boutons cliquables, le sélecteur de couleur et des glissières intégrées aux panneaux.

2.2 Module ofxAssimpModelLoader

L'importation de modèles 3D se fait principalement en passant par ce module. Les fonctions associées renseignent sur l'ensemble des attributs reliés aux formats supportés.

2.3 Module ofxDelaunay

Ce module permet d'obtenir les fonctions nécessaires à la réalisation de l'implémentation de la triangulation de Delaunay.

2.4 Module ofxReflexionRefraction

Ce module permet de créer l'effet de réflexion et de réfraction voulue à l'objet « Boite en verre ». Cet objet retourne les informations provenant du « Skybox » et ajoute une déformation calculée par le module ofxReflexionRefraction.

3. Compilation

La première étape est d'installer openFrameworks sur l'ordinateur. Par la suite, il suffit de copier le projet dans le dossier « myapps » de openFramworks. Une fois le dossier copier, il faut utiliser le « project generator » de openFrameworks et il faut ajouter les librairies ofxAssimpModelLoader, ofxGui, ofxReflexionRefraction, ofxVectorGraphics et ofxDelaunay. Une fois le projet ouvert, il faut s'assurer de supprimer les fichiers « ofApp.cpp » et « ofApp.h ». Pour s'assurer que tout marche bien, il serait préférable d'utiliser le même IDE que nous avons utilisé, c'est-à-dire, Visual Studio 2022.

Les liens github pour les librairies qui ne sont pas déjà incluses dans openFrameworks sont les suivantes :

- https://github.com/obviousjim/ofxDelaunay
- https://github.com/kashimAstro/ofxReflectionRefraction

4. Architecture

Le fichier main gère les paramètres de la fenêtre et la crée avec une résolution de 1024 par 768 pixels. La version de OpenGL est fixée à 3.3.

4.1 Fichiers nommés application

Les ficher application.h et application.cpp contiennent la classe ofApp, classe dérivée de ofBaseApp.

Comme mentionné dans le premier rapport, dans la classe ofApp, l'ensemble des objets reliés à l'interface graphique comme les panneaux de contrôles de propriétés, de création d'objets, de caméra et de hiérarchie y sont créés. Dans cette deuxième partie du travail, nous avons ajouté les interfaces pour la gestion du filtrage et du mappage tonale des images dans la scène 2D ainsi que les boutons pour ajouter les deux types de courbes paramétriques. Un panneau ponctuel a également été ajouté pour la gestion des points de contrôle des courbes. Dans la scène 3D, des boutons pour de nouveaux objets ont été ajoutés pour la surface de Bézier, le Quad, la triangulation de Delaunay ainsi qu'une boite de verre «GlassBox». Des panneaux pour le choix du modèle d'illumination., l'ajout de lumière, et le choix de certains matériaux ont été également ajoutés.

4.2 Fichiers nommés renderer

L'objectif de ces fichiers reste le même dans cette seconde partie du travail. C'est-à-dire, de recevoir l'ensemble des données des autres classes. Elle contient les objets 2D et 3D dans des vecteurs. Elle est responsable de la collecte d'information sur la position dans l'espace de la souris. Cette classe fait la gestion du temps. Elle reçoit ces paramètres en entrée, fait le traitement de l'information, applique les méthodes de transformations des objets concernées. Les résultats attendus lors de l'utilisation du logiciel sont fournis à la classe application ofApp pour l'affichage à l'écran. L'ensemble des méthodes comprenant la gestion des nouveaux critères fonctionnels, par exemple la gestion des lumières, le changement des filtres ainsi que l'ajout des nouveaux objets se font dans cette classe.

4.3 Fichiers nommés object2D

Une classe principale nommée Object2D est une classe virtuelle possédant des attributs pour la position, la rotation et la proportion le nom et la couleur des objets qui sont utiles pour l'ensemble des objets 2D. Ce sont maintenant neuf classes qui héritent de la classe Object2D. Il y a toujours les cinq classes pour la création de primitives vectorielles soit le cercle, le triangle, le rectangle, l'ellipse et la ligne. Les deux autres classes permettent la création de deux formes vectorielles, l'étoile et la maison. Une classe sert à la création d'un objet image et une nouvelle classe permet la création des courbes paramétriques.

4.4 Fichiers nommés object3D

Dans ces fichiers se trouve une grande classe pour gérer l'ensemble des objets 3D. L'ensemble de ces objets sont spécifiés par une énumération des types suivants : importation, primitive3D, sphere3D, box3D, cylinder3D, cone3D. À cette liste nous ajoutons ces types : surfaceBezier, quad3d, delaunayTriangle, GBox les attributs de la classe permettent la création de ces objets et de contenir les informations relatives aux paramètres de chaque objets3D (nom, taille, couleur, etc.) ainsi que les informations liées aux transformations (rotation, proportion et translation de position). Cette classe permet également la gestion des shaders, des textures et des animations des objets3D concernés.

4.5 Fichiers nommés bezierSurface

Ces fichiers contiennent le code permettant de créer un objet de type surface bicubique de Bézier permet de contenir l'ensemble des paramètres distinctifs à cet objet. Il est ensuite appelé dans les fichiers objets 3D.

4.6 Fichiers nommés Skybox

Ces fichiers permettent de générer la skybox et de faire intervenir les shaders du même nom. L'ensemble des paramètres de positionnement des images s'y retrouve. Elle est ensuite appelée dans la méthode setup () du renderer.

4.7 Fichiers nommés Quad

Ces fichiers permettent de générer la forme étrange sur laquelle s'applique l'algorithme de tessellation. Ils contiennent également le code permettant d'en faire la visualisation en fil de fer. Cette forme intervient dans la réalisation du critère fonctionnel sur la tessellation du module 9.

5. Fonctionnalités

L'ensemble des fonctionnalités du projet sont décrites dans cette section. Au total, ce sont 16 critères fonctionnels qui ont été implémentés pour arriver à ce résultat. Dans cette section, les critères sont détaillés en fonction de chacun des cinq modules correspondant à la deuxième partie du cours.

5.1 Textures

Les critères fonctionnels qui ont été choisis reliés au module sur les textures sont les coordonnées de textures, le filtrage, le mappage tonal, ainsi que le cubeMap. Chacun de ces critères sera montré dans les sous-sections suivantes.

5.1.1 Coordonnées de texture

Une texture peut être appliquée sur différentes formes géométriques 3D. Pour se faire, un mipMap de la texture est généré et ensuite elle est mise à jour par son application en fonction des coordonnées de l'objet. Cela permet d'habiller adéquatement par exemple, un cône (Figure 1). Pour se faire, il faut sélectionner. Un matériau existant dans le menu correspondant. Ensuite, en appuyant sur la touche F8, il est possible d'importer une nouvelle texture pour l'appliquer sur le modèle sélectionné.

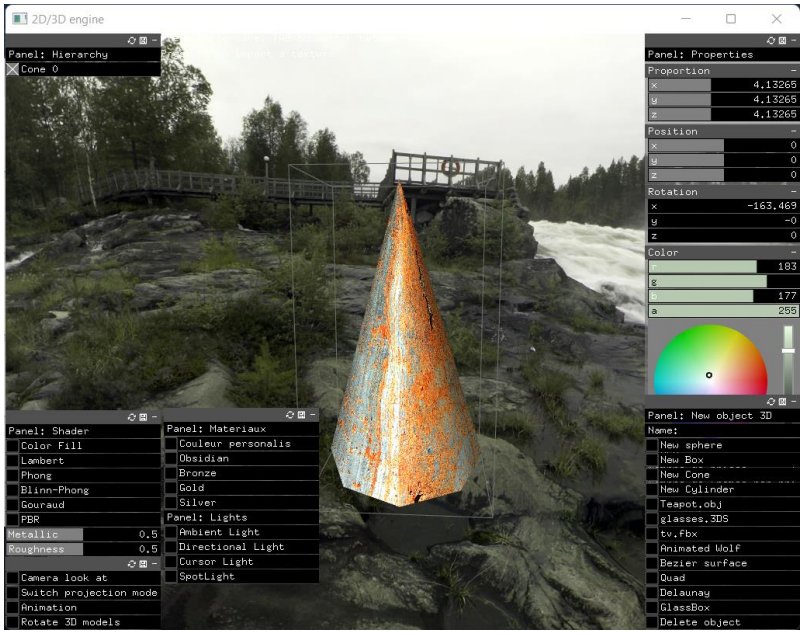

Figure 1 Cône sur lequel une texture rouillée est appliquée.

5.1.2 Filtrage

Lorsque l'on charge une image dans le moteur de rendu 2D, un nouveau menu s'affiche dans lequel nous avons la possibilité d'ajouter quatre filtres à l'image en question ainsi que la possibilité de revenir à l'identité originale de l'image (Figure 2). Les types de filtrages sont par embossage, par aiguisage, par détection des bordures et par flou.

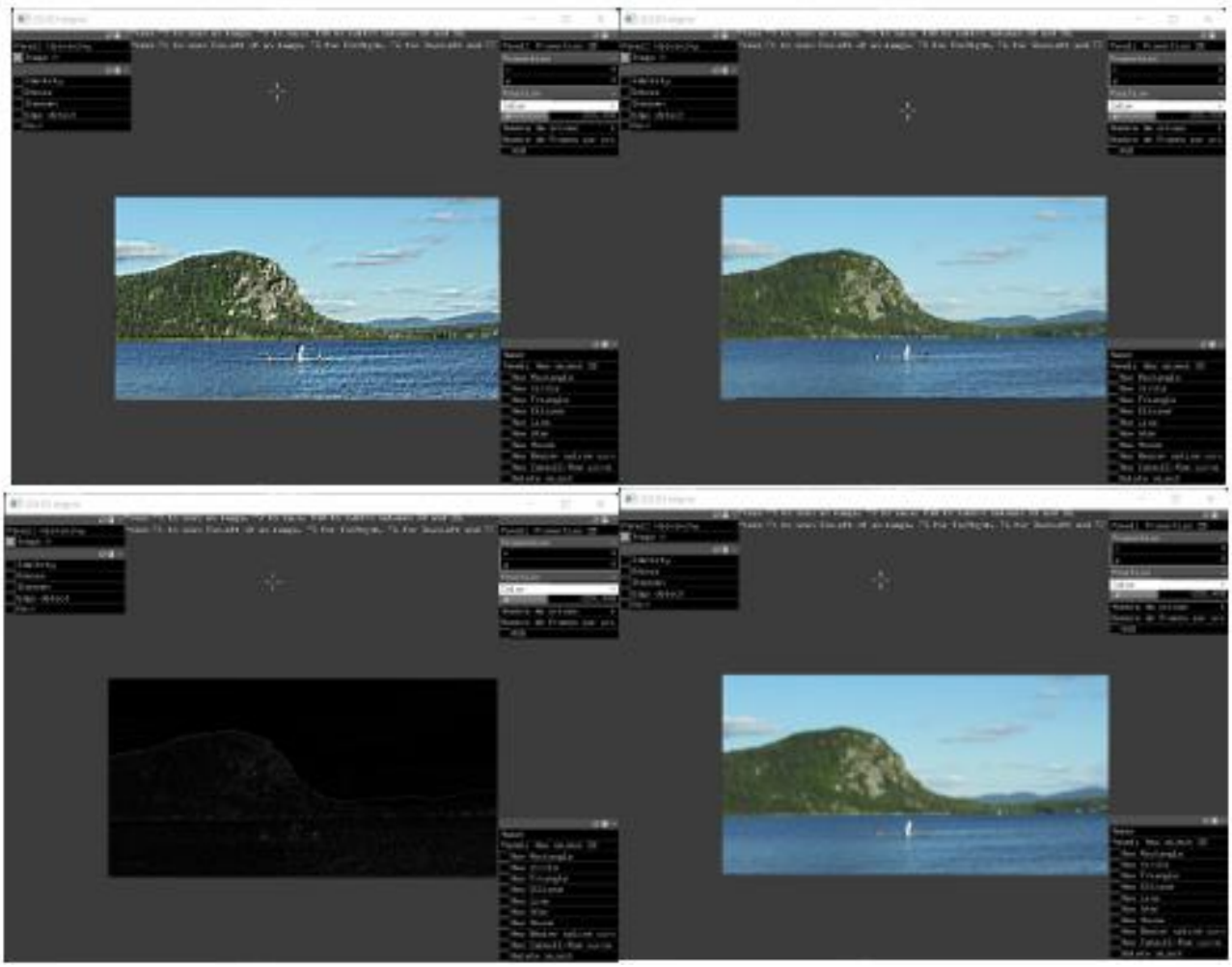

Figure 2 Filtrage embossée, aiguisée, par détection des bords et floue sur une image chargée préalablement.

5.1.3 Mappage tonal

Dans le même menu que les options de filtrage qui apparait au chargement d'une image, nous y trouvons deux glissières permettant de modifier l'exposition ainsi que le facteur de correction gamma. Ces glissières sont positionnées à l'origine sur les valeurs de 1.00 et de 2.2 respectivement (Figure 3).

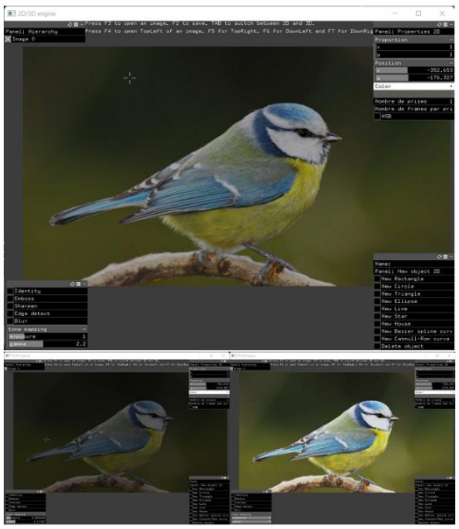

Figure 3. Sous l'image d'origine, l'image de gauche correspond à une diminution de moitié du facteur gamma et de l'exposition tandis que, dans l'image de droite, les deux paramètres ont une valeur de 5.

5.1.4 Cubemap

En ce qui concerne le critère du « cubemap », nous avons opté pour un rendu de type Skybox pour ajouter une mise en contexte des éléments ajoutés à la scène (Figure 4).

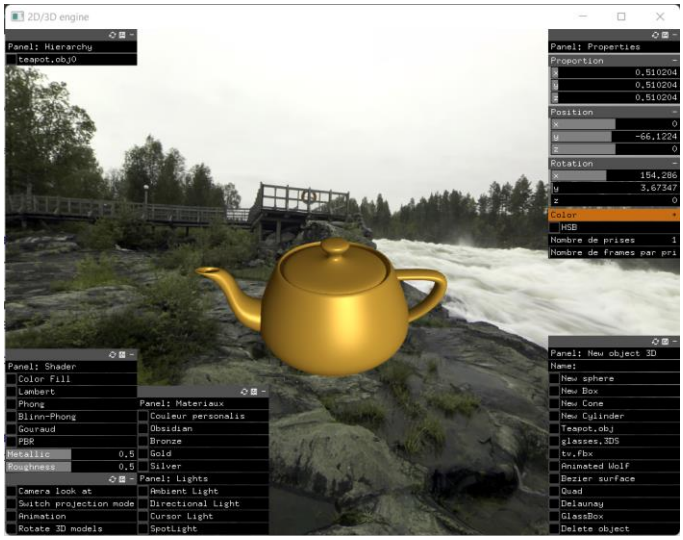

Figure 4 Présence d'un paysage de type Skybox dans la scène

5.2 Illumination classique

Des critères fonctionnels en lien avec l'illumination classique ont été ajoutés à notre moteur de rendu 3D. On y retrouve des modèles d'illuminations, des matériaux, des lumières de différents types et la possibilité d'avoir des lumières multiples.

5.2.1 Modèles d'illumination.

Les modèles d'illumination sont contenus dans des shaders et chaque modèle est applicable en le sélectionnant dans le panneau de contrôle des shaders. On y retrouve une simple couleur de remplissage, un modèle de Lambert, un modèle de Phong, un de Blinn-Phong ainsi que celui de Gouraud. Lorsque l'on sélectionne le bouton apparenté dans le menu des shaders, on obtient l'illumination correspondante (Figure 5).

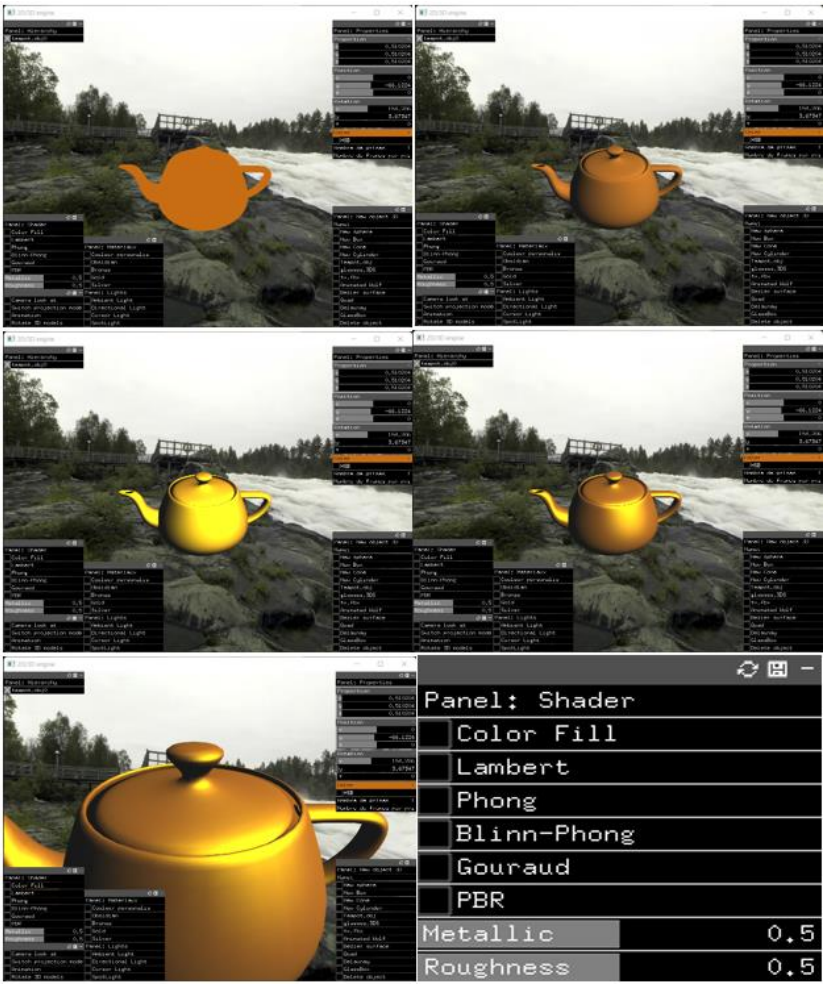

Figure 5 Résultat des modèles d'illumination 1. Couleur de remplissage 2. Modèle de Lambert 3. Modèle de Phong 4. Modèle de Blinn-Phong 5. Modèle de Gouraud 6. Panneau de sélection des shaders

5.2.2 Matériaux

Quatre matériaux ont été mis à disposition dans le panneau du même nom. Les choix sont obsidienne, bronze, or et argent (Figure 6). Lorsqu'on sélectionne un objet, il est possible de choisir un de ces matériaux et de les mettre en valeur avec les options du panneau d'éclairage. Il y a aussi la possibilité d'appliquer une simple couleur personnalisée comme matériau.

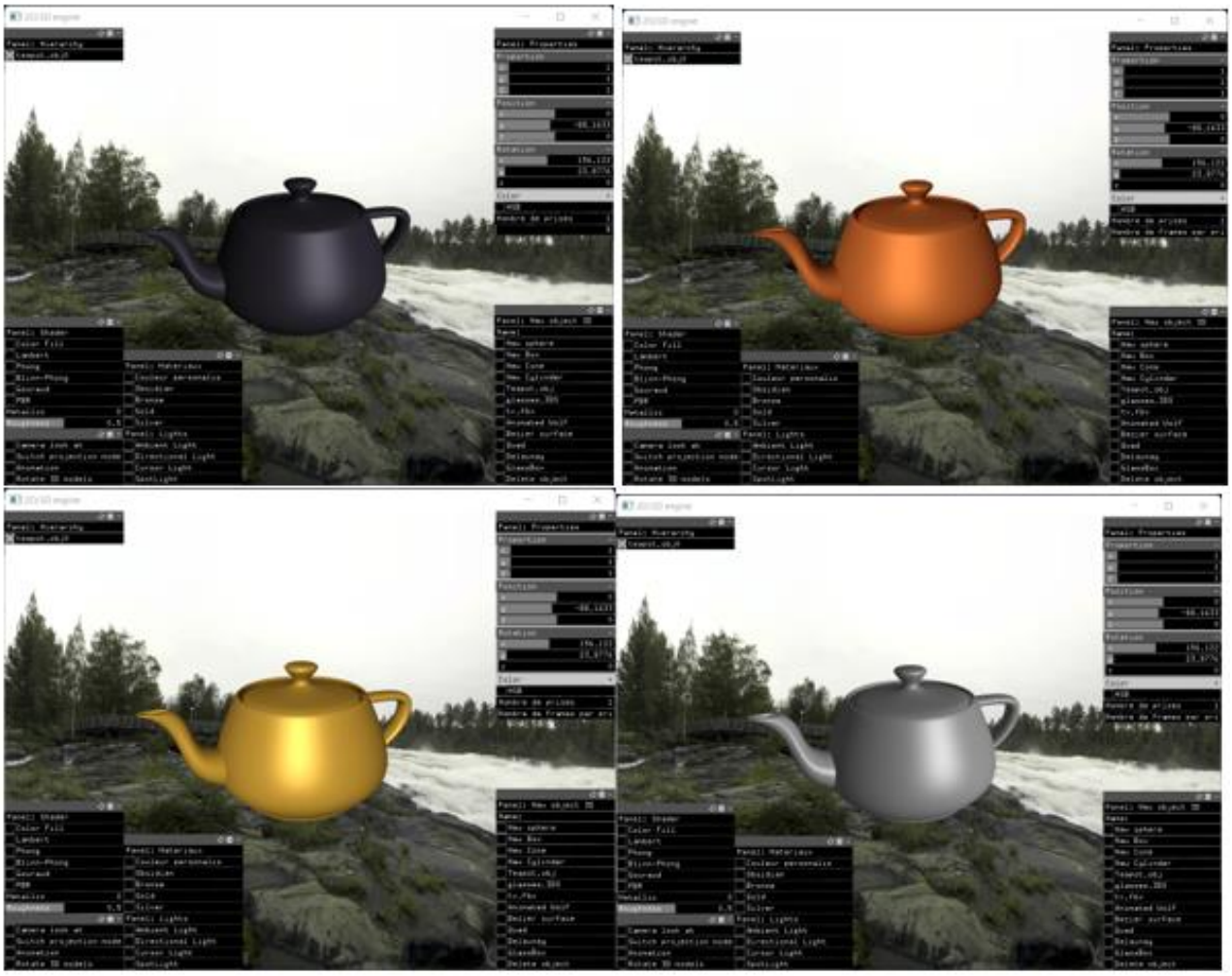

Figure 6 Matériaux obsidienne, bronze, or et argents appliqués à une théière

5.2.3 *Types de lumières*

Dans le Panneau des matériaux, quatre types de lumières peuvent être ajoutés pour augmenter la qualité du rendu visuel de l'objet. La lumière ambiante, la lumière directionnelle, la lumière ponctuelle et la lumière projecteur sont disponibles dans le panneau de sélection des lumières (Figure 7).

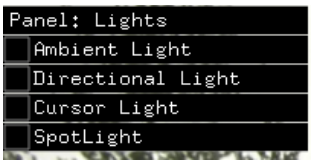

Figure 7. Menu concernant l'ajout des types de lumières dans la scène

5.2.4 *Lumières multiples*

Sur l'exemple suivant, trois reflets de lumières se distinguent sur la théière. Celui de gauche provient de la lumière directionnelle (cette lumière est dynamique et observable dans le vidéo. Celui en bas à droite est une lumière ponctuelle reliée au curseur et le troisième reflet provient d'une lumière projecteur (Figure 8). Une lumière ambiante permet également de bien percevoir le matériau.

Figure 8 Observation de 3 reflets de lumières principaux reliés aux lumières présentes dans la scène.

5.3 Lancer de Rayon

Deux critères fonctionnels ont été intégrés au projet soit la Réflexion et la Réfraction. Nous avons implémenté un cube qui en fonction de la rotation de celui-ci offre les qualités de réflexion d'un miroir ou de réfraction d'un cube de verre.

5.3.1 Réflexion

Lorsque le cube de verre montre les faces ayant les propriétés de réflexion, il est clair que l'image perçue sur ces faces est celle provenant du Skybox se trouvant derrière la caméra (Figure 9). Dans ce cas-ci. Des faces du cube reflètent les pierres au sol lorsque la caméra est orientée vers le ciel.

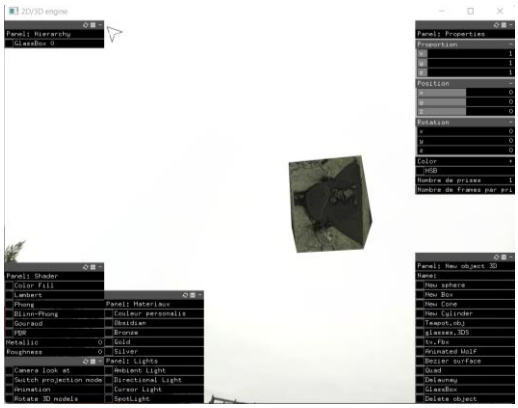

Figure 9. Réflexion des pierres au sol par le cube lorsque la caméra est orientée vers le ciel.

5.3.2 Réfraction

La réfraction est démontrée par l'entreprise du cube de verre qui est en mesure de déformer la lumière qui provient du Skybox vers la caméra (Figure 10)

Figure 10. Effet de la réfraction des rayons lancés vers la caméra par un cube de verre.

5.4 Topologie

Quatre critères fonctionnels ont été implémentés en ce qui concerne la topologie. Il est possible d'ajouter des courbes et des surfaces paramétriques. Il est également possible de modifier la tessellation d'un objet nommé « Quad » et de réaliser une triangulation de Delaunay.

5.4.1 Courbe paramétrique

Deux types de courbes paramétriques ont été ajoutées au moteur de rendu 2D. La première est une spline de Bézier (B-spline) et la seconde est une courbe de Catmull-Rom. En ce qui concerne la B-Spline, nous avons la possibilité de la modifier à l'aide de 7 points de contrôles. Un menu spécial pour la gestion de ces points s'ouvre à gauche de l'écran lorsque la B-Spline est sélectionnée dans la hiérarchie (Figure 11).

Figure 11 B-spline et à gauche de l'écran, son menu pour la gestion des positions de ses 7 points de contrôles

Une courbe de Catmull-Rom est également accessible dans le moteur 2D. Encore une fois elle est modifiable à l'aide de sept points de contrôles, tous accessibles par un menu qui apparait à gauche lorsque l'objet est sélectionné dans le panneau de hiérarchie (Figure 12).

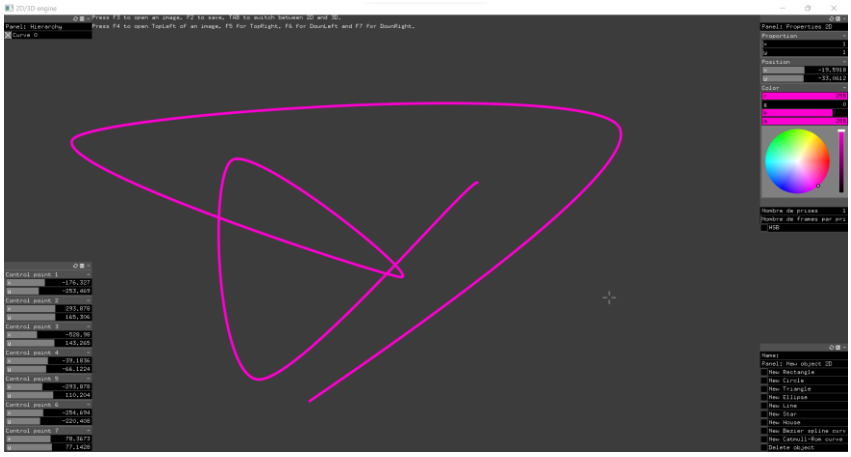

Figure 12 Courbe de Catmull-Rom et ses points de contrôles dans le menu de gauche.

5.4.2 *Surface paramétrique*

En ce qui concerne le moteur 3D, une surface paramétrique de Bézier bicubique a été ajoutée aux modèles que nous pouvons ajouter à la scène (Figure 13). Il est possible de voir son modèle en fil de fer à la sélection de la surface dans la hiérarchie des objets.

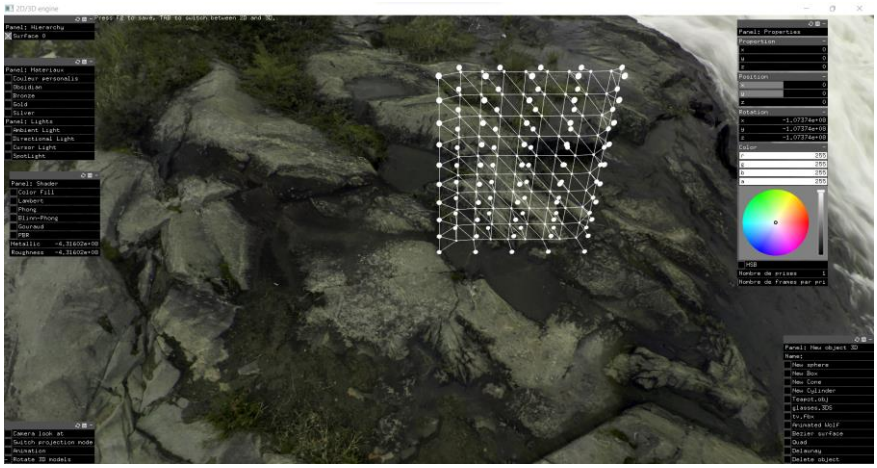

Figure 13 Surface de Bézier bicubique ajoutée à la scène avec ses points de contrôles

5.4.3 *Tessellation*

Un algorithme de tessellation a été implémenté ce qui permet en ajoutant un objet nommé quads avec une forme étrange, voire l'effet sur sa structure en fil de fer (Figure 14). Le nombre de sommets diminue en fonction de la distance de la caméra.

Figure 14. Diminution du nombre de sommets en fonction de la distance de la caméra

5.4.4 *Triangulation*

En ce qui concerne l'intégration d'un algorithme de triangulation à partir d'un nombre d'un ensemble de sommets, nous avons choisi la triangulation de Delaunay. Lorsque la triangulation est ajoutée à la scène et qu'elle est sélectionnée dans la hiérarchie, il suffit de cliquer dans l'écran pour générer des sommets. À partir de trois sommets, les arrêtes s'affiches et se recalcules en fonction des sommets voisins (Figure 15).

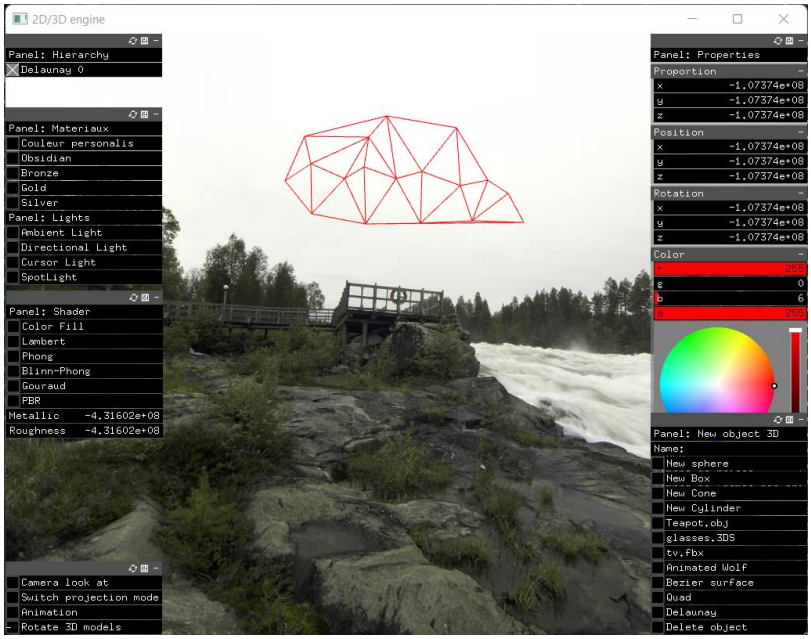

Figure 15. Maillage triangulaire généré par l'algorithme de Delaunay à la suite de la création par clics de souris de plus d'une quinzaine de sommets.

5.5 Illumination moderne

Les critères fonctionnels suivants relèvent du module portant sur l'illumination moderne. Deux critères ont été ajoutés soit la possibilité de modifier la métallicité d'un matériau ainsi que sa rugosité un coup que le PBR est activé.

5.5.1 Métallicité

Lorsqu'on active le rendu basé sur la physique (PBR), un matériau par défaut est ajouté à l'objet 3D. Il est possible ensuite de modifier ses caractéristiques, passant de métallique à diélectrique à l'aide d'une glissière dans le menu des Shaders (Figure 16). Il est possible de l'impact en temps réel en positionnant la glissière sur 0 pour un matériau totalement diélectrique, sur 1 pour un matériau totalement métallique ou entre les deux pour un résultat personnalisé.

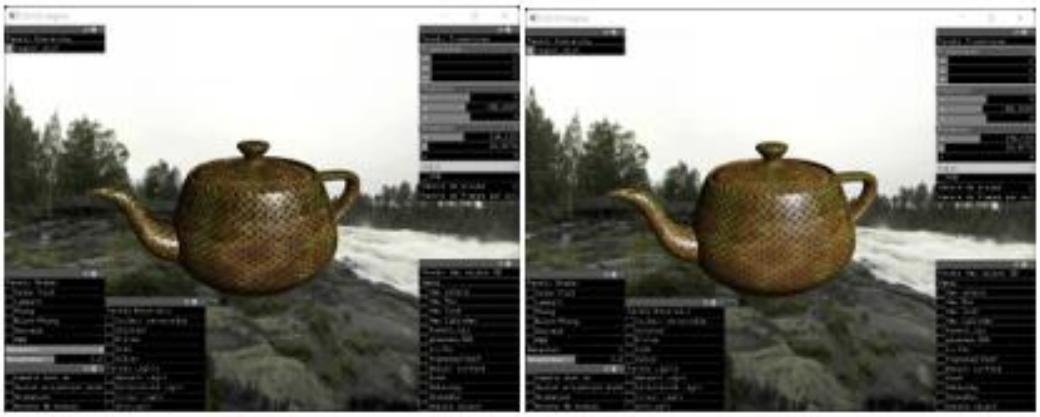

Figure 16 gauche: Effet du caractère métallique du matériau. Droite: Effet du caractère diélectrique du matériau la rugosité dans les 2 cas est à une valeur de 0,5.

5.5.2 Rugosité

Lorsque le mode PBR est sélectionné, il est possible de modifier la simulation de la rugosité à l'aide d'une glissière paramétrée entre 0 et 1. Lorsqu'elle se rapproche de 0, la rugosité devient presque inexistante ce qui a un effet sur la réflexion spéculaire qui devient très précise (Figure 17). Lorsqu'elle se rapproche de 1, la rugosité devient très importante en multipliant le nombre de micro facettes dans la simulation. Cela a un impact sur la réflexion spéculaire qui semble plus diffuse.

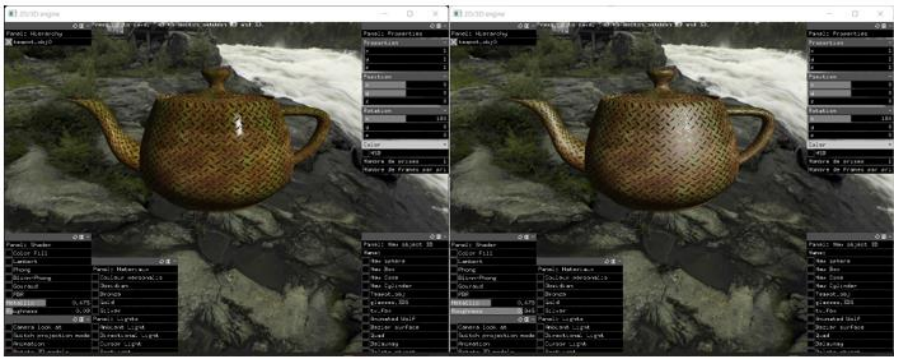

Figure 17 gauche: rugosité presque nulle avec un taux de 0,09. Droite: Rugosité importante avec un taux de 0,845

6. Ressources

Cette section fait état des ressources externes qui ont été utilisées pour réaliser le livrable.

6.1 Shaders

Plusieurs Shaders ont été récupérés dans les exemples de cours et ajoutés au projet par l'équipe comme ressource externe :

- Shader d'illumination de Lambert (lambert_330_fs , lambert_330_vs)
- Shader d'illumination de Gouraud (Gouraud_330_fs , Gouraud_330_vs)
- Shader de couleur ambiante (color_fill_330_fs , color_fill_330_vs)
- Shader d'illumination de Phong (Phong_330_fs , Phong_330_vs)
- Shader d'illumination de Blinn-Phong (blinn_phong_330_fs , blinn_phong_330_vs)
- Shader d'illumination moderne PBR (pbr_330_fs , pbr_330_vs)
- Shader pour le Skybox (Skybox_330_fs , Skybox_330_vs)

6.2 Models 3D

En ce qui concerne les modèles 3D, ils sont tous d'origine externe. Voici la liste des ressources :

- Teapot.obj
- Tv.fbx
- Glasses.3DS
- Wolf_dae.dae

À l'exception de teapot.obj, qui a été pris dans les exemples du cours, ces ressources gratuites ont été prises sur https://www.cgtrader.com/

6.3 Images du Skybox

Le Skybox provient d'images récupérées sur un site de Skybox gratuit à l'adresse suivante :

http://www.humus.name/index.php?page=Textures&start=0

6.4 Textures

Les textures pour le PBR ont été récupérées dans les exemples du cours

6.5 Addons pour openFrameworks

Les différents «addons» ont été téléchargés sur les pages GitHub correspondantes à partir de la page de référence d'openFramework à l'adresse suivante : https://ofxaddons.com/categories

Ces deux modules ont été récupérés de cette façon :

- ofxDelaunay
- ofxReflexionRefraction

7. Présentation de l'équipe

Hugo Chiasson

Mon nom est Hugo Chiasson, j'ai 30 ans et je suis natif de Trois-Rivières. Avant l'université, j'ai été 10 ans dans les forces armées comme technologue en système de communication terrestre. À 27ans, j'ai décidé de tout lâcher pour retourner à l'université et me voilà à la moitié du BAC en informatique. J'ai choisi ce cours, car j'ai toujours été intéressé par le milieu du divertissement, en particulier celui des jeux vidéo et de l'animation par ordinateur.

Contribution au projet :

Dans le cadre du TP2, j'ai surtout travaillé à implanter les critères liés à l'illumination classique, le lancer de rayon et l'illumination moderne. J'ai aidé mes coéquipiers pour améliorer l'implantation des critères des autres modules. J'ai réalisé également la vidéo de présentation.

Keven Lessard

Mon nom est Keven Lessard, j'ai 24 ans et je suis natif de Victoriaville. Je me suis inscrit au baccalauréat en informatique pour ma passion pour la programmation et les jeux vidéo. J'en suis maintenant à plus de la moitié dans mon parcours et j'ai choisi ce cours, car l'infographie est une partie très importante dans le développement de jeux vidéo. Ultimement, mon but est de travailler dans l'industrie du jeu donc ce projet est une excellente opportunité pour moi d'acquérir de l'expérience.

Contribution au projet :

J'ai développé surtout les critères liés à la topologie et les textures. J'ai également contribué à l'intégration des différentes parties en un projet final.

Guillaume Côté

Je fais le microprogramme en jeux vidéo par pur intérêt personnel. Je travaille comme conseiller pédagogique au cégep de Sherbrooke à temps plein. Je fais donc les cours à temps partiel en asynchrone. J'ai développé un intérêt pour le développement de jeux vidéo en explorant Blender, Unreal Engine et en participant à un Game jam. Mon envie de développer mes compétences m'a poussé à m'inscrire au programme pour ajouter de la structure dans mon parcours d'apprentissage. J'ai en tête de pouvoir réaliser un projet personnel en jeux vidéo à moyen terme.

Contribution au projet :

Ma contribution a été amoindrie à la suite de problèmes de santé. Je me suis principalement concentré sur les critères en lien avec le module sur les textures, surtout le CubeMap. Pour les autres critères. L'équipe en a pris beaucoup sur leurs épaules pour pallier la situation. Je me suis concentré par la suite à la rédaction du document Word.