

IFT3100H20
Projet de session

Sommaire

L'objectif du projet de session est de développer une application qui permet de construire, éditer et rendre des scènes visuelles.

Votre travail pratique sera principalement évalué selon votre implémentation de différents critères fonctionnels en lien avec la matière des modules du cours.

Un total de **50** critères fonctionnels organisés en **10** catégories vous est proposé dans les pages suivantes. Chaque catégorie est en lien avec la matière du module théorique du même nom.

La première partie du projet de session (**TP1**) porte sur les critères fonctionnels des **5** premières catégories. La seconde partie du projet de session (**TP2**) porte sur les critères fonctionnels des **5** dernières catégories.

Tous les critères fonctionnels sont optionnels et ont une valeur de **1** à **3** points. Vous devez implémenter environ **3** critères sur **5**. Vous pouvez faire plus de critères dans certaines catégories et moins d'en d'autre, selon vos préférences.

Le langage de programmation recommandé pour le développement de la couche application est le **C++**.

La librairie graphique recommandée est **OpenGL 3.3** ou plus, avec **GLSL** comme langage de programmation pour les shaders.

Tous les autres choix en lien avec les technologies, le design logiciel et la thématique du projet sont laissés à la préférence des membres des équipes.

Évaluation

Voici les informations par rapport à l'évaluation du projet de session.

Le projet de session est séparé en deux livraisons : travail pratique 1 (**TP1**) et travail pratique 2 (**TP2**).

L'évaluation de chacune des livraisons sera faite en fonction de ces **3** éléments :

- ▷ Critères fonctionnels (**40** points)
- ▷ Document de design (**5** points)
- ▷ Vidéo (**5** points)

Voici comment la valeur de l'évaluation sera calculée :

- ▷ Chaque livraison sera évaluée sur **50** points, soit le total des points pour les critères fonctionnels, le document de design et la vidéo.
- ▷ Ce total sera ramené sur **100%**, puis les bonus et les malus seront ajoutés, si applicables.
- ▷ Le pourcentage total incluant les bonus et malus sera ensuite ramené sur **20** points, soit la pondération du travail pratique sur l'ensemble de la session.

Voici l'échelle de notation pour les critères fonctionnels :

- ▷ Tous les critères seront évalués de **0** à **3** points.
- ▷ **★★★ 3** points, excellent, au-delà des attentes.
- ▷ **★★ 2** points, bon, au niveau des attentes.
- ▷ **★ 1** point, correct, mais en bas des attentes.
- ▷ **0** point, critère non évalué ou insuffisant par rapport aux attentes.

La grille d'évaluation (un fichier excel) se trouve sur le site web du cours et elle est à livrer à chacune des deux livraisons avec le pointage visé pour les critères fonctionnels à évaluer.

1. Critères fonctionnels

Voici les informations sur l'évaluation des critères fonctionnels :

- ▷ Tous les critères fonctionnels sont optionnels.
- ▷ Vous devez implémenter au minimum **3** critères fonctionnels sur **5** dans n'importe quelle catégorie (**1** à **5** pour **TP1** et **6** à **10** pour **TP2**).
- ▷ Implémenter plus de critères fonctionnels peut contribuer à obtenir une meilleure note.
- ▷ Une implémentation de bonne qualité d'un des critères fonctionnels vaudra généralement **2** points.
- ▷ Le nombre maximal de points pour la somme des critères fonctionnels est de **40** points par livraison.

2. Document de design

Voici les informations sur l'évaluation du document de design :

- ▷ Le document de design a une valeur de **5** points à chacune des **2** livraisons.
- ▷ Pour le **TP1**, l'évaluation est par rapport à l'information présentée sur les **5** premières catégories de critères fonctionnels et sur les autres sections du document.
- ▷ Pour le **TP2**, l'évaluation est par rapport à l'information présentée sur les **5** dernières catégories de critères fonctionnels et sur l'amélioration du document en général.
- ▷ Le document de design peut être rédigé avec le logiciel de votre choix (ex. *Word*, *Google Docs*, *LaTeX*).
- ▷ La version à évaluer doit être exportée en format **.pdf**.

3. Vidéo

Voici les informations sur l'évaluation de la vidéo :

- ▷ La vidéo a une valeur de **5** points à chacune des **2** livraisons.
- ▷ L'évaluation est par rapport à la qualité de la démonstration des critères fonctionnels et du montage de ses différents segments.
- ▷ La vidéo peut avoir pour sources des images générées par votre application ou des captures d'écran.
- ▷ Le montage vidéo peut être fait avec le logiciel de votre choix (ex. *iMovie* avec *macOS*, *Microsoft Photos* avec *Windows*).
- ▷ Si la vidéo est livrée sous forme de fichier, elle doit être exportée dans un format (ex. *.mp4*, *.mov*) et un codec (ex. *H.264*, *HEVC*) qui fonctionnent sur la plupart des ordinateurs.
- ▷ Idéalement, la vidéo est livrée sur le web (ex. *YouTube*, *Vimeo*).

4. Bonus

Des points bonus peuvent être accordés si par exemple le projet se démarque d'un point de vue qualitatif ou s'il va au-delà de ce qui est demandé dans cet énoncé.

Voici quelques exemples de bonus :

- ▷ Versions compilées pour plusieurs plateformes.
- ▷ Qualité visuelle du projet.
- ▷ Originalité.
- ▷ Partage.

5. Malus

Des points peuvent être retranchés en malus lorsqu'entre autres :

- ▷ Les consignes de l'énoncé ne sont pas respectées.
- ▷ Le code source de l'application ne compile pas.
- ▷ Des erreurs sont affichées dans la console pendant d'exécution.
- ▷ L'application plante au démarrage.
- ▷ L'application plante en cours d'exécution.
- ▷ L'application est difficile à utiliser.
- ▷ Le projet manque de finition.
- ▷ Le projet est livré en retard.

Livraison

Voici le contenu de chaque livraison (**TP1** et **TP2**) :

- ▷ Tout le contenu de la livraison doit se trouver à l'intérieur d'un répertoire racine qui à cette nomenclature : **IFT3100H20_TP2_E01** pour le **TP2** de l'équipe **1**.
- ▷ Le répertoire racine organise le contenu du projet dans des sous-répertoires (ex. */doc*, */src*, */bin*).
- ▷ Code source de l'application.
- ▷ Dépendances nécessaires à la compilation de l'application.
- ▷ Version compilée pour au moins un système d'exploitation
- ▷ Version **.pdf** du document de design.
- ▷ Vidéo sous forme de fichier ou de liens vers le web.
- ▷ Compresser le répertoire racine dans une archive de format **.zip** avant la livraison dans la boîte de dépôt.

Voici les informations générales par rapport aux livraisons :

- ▷ La livraison des travaux pratiques doit se faire dans la boîte de dépôt de l'équipe sur le site web du cours.
- ▷ La limite de poids permise pour l'archive du travail pratique est la même que celle du système de boîte de dépôt du portail des cours (**250 MO**).
- ▷ Le système de boîte de dépôt permet de faire plusieurs livraisons, mais seulement la dernière version livrée sera évaluée.
- ▷ Éviter les caractères spéciaux, les caractères avec des accents et les espaces dans tous les noms de fichiers et de répertoire.
- ▷ Il est fortement recommandé de tester la version compilée du projet sur d'autres ordinateurs pour valider que tout fonctionne comme prévu.
- ▷ La date de livraison du **TP1** est le dimanche **8 mars** avant minuit.
- ▷ La date de livraison du **TP2** est le dimanche **3 mai** avant minuit.
- ▷ La pénalité de retard est de **1%** par heure.

Document de design

Cette section présente la structure proposée pour le document de design qui accompagne le projet de session.

1. Sommaire

Description sommaire du projet (1 page).

2. Interactivité

Description en détail de toutes les formes d'interactivité possibles ainsi que les principales entrées et sorties de l'application (1 page).

3. Technologie

Présentation des principaux outils technologiques utilisés pour le développement de l'application (de 1/2 page à 1 page).

4. Compilation

Présentation de la procédure et des dépendances pour réussir à compiler l'application sur un autre ordinateur (de 1/2 page à 1 page).

5. Architecture

Présentation de l'architecture logicielle de l'application sous forme textuelle et/ou de diagrammes (1 page).

6. Fonctionnalités

Description des critères fonctionnels implémentés dans votre projet.

Pour chacun des critères fonctionnels :

- ▷ Faire une description sommaire de comment le critère fonctionnel a été réalisé dans votre projet.
- ▷ Si pertinent, présenter une ou des images où le critère fonctionnel est mis en évidence.
- ▷ Si pertinent, présenter l'essentiel du code qui a permis d'implémenter le critère fonctionnel.

Prévoir environ 1 à 3 critères fonctionnels par page si seulement du texte ou 1 à 2 par page si accompagnés d'images et/ou d'extraits de code).

7. Ressources

Liste des ressources originales produites pour le projet (ex. image, géométrie, animation, audio, vidéo, texte) et citation des références pour les ressources qui n'ont pas été produites par l'équipe (1 page).

8. Présentation

Présentation de l'équipe et de ses membres, ainsi qu'un sommaire de qui a fait quoi dans le projet (1 page).

1 Image

1.1 Importation d'images

Après le lancement de l'application, il est possible d'importer de manière interactive des fichiers images et de les afficher dans une scène sous une forme ou une autre.

□

1.2 Exportation d'images

En cours d'exécution, il est possible d'exporter de manière interactive des séquences de fichiers images qui correspondent au rendu d'une scène pendant un intervalle de temps.

□

1.3 Échantillonnage d'images

L'application permet de générer une nouvelle image à partir d'un agencement d'échantillons de blocs de pixels en provenance d'une ou plusieurs autres images.

□

1.4 Espaces de couleur

L'application permet de manière interactive de choisir et de convertir des couleurs dans l'espace RGB et HSB pour au moins 2 éléments distincts.

□

1.5 Histogramme

L'application permet de calculer et de dessiner l'histogramme d'une image.

□

2 Dessin vectoriel

2.1 Curseur dynamique

Il existe au moins 5 représentations visuelles différentes du curseur qui sont affichées en fonction d'état courant de l'application.



2.2 Outils de dessin

Il est possible de modifier de manière interactive la valeur des outils de dessin vectoriel tels que l'épaisseur des lignes de contour, la couleur des lignes de contour, la couleur des zones de remplissage et la couleur d'arrière-plan de la scène.



2.3 Primitives vectorielles

Il est possible de créer de manière interactive des instances d'au moins 5 types des primitives vectorielles parmi cet ensemble : point, ligne, carré, rectangle, triangle, quadrilatère, polygone régulier, polygone irrégulier, cercle, ellipse et arc.



2.4 Formes vectorielles

Il est possible de créer des instances d'au moins 2 types de formes vectorielles composées d'un ensemble d'instances de primitives vectorielles.



2.5 Interface

Un ou des éléments d'interface graphique offrent de la rétroaction informative visuelle à l'utilisateur et des contrôles interactifs pour influencer les états de l'application.



3 Transformation

3.1 Graphe de scène

Tous les éléments visuels présents dans une scène sont organisés dans une ou des structures de données qui permettent l'ajout, la suppression et la sélection d'éléments.

□

3.2 Sélection multiple

Il est possible de sélectionner plus d'une instance des éléments visuels présents dans une scène et de modifier sur chaque élément de la sélection la valeur de certains attributs qu'ils ont en commun.

□

3.3 Transformations interactives

Il est possible de modifier de manière interactive la translation, la rotation et la proportion des éléments visuels présents dans une scène.

□

3.4 Historique de transformation

Il est possible d'annuler ou de refaire (*undo / redo*) les dernières actions interactives qui ont un impact sur la transformation des éléments visuels présents dans une scène.

□

3.5 Système de coordonnées

Il est possible de dessiner des éléments visuels qui sont transformés dans un système de coordonnées non cartésien.

□

4 Géométrie

4.1 Boîte de délimitation

Une option permet de dessiner les arêtes d'une boîte d'une taille juste assez grande pour envelopper tous les sommets d'un modèle 3D pour chaque type de modèle qu'il est possible d'utiliser avec l'application.

□

4.2 Primitives géométriques

Il est possible de dessiner au moins 2 types de primitives géométriques 3D générées à partir d'un algorithme qui n'utilise aucune donnée externe au programme.

□

4.3 Modèles 3D

Il est possible de dessiner des instances d'au moins 3 types de modèles 3D importés à partir d'un fichier externe.

□

4.4 Animation

Il est possible de faire le rendu d'au moins 2 animations sur 1 ou plusieurs maillages géométriques.

□

4.5 Shader de géométrie

L'application fait le rendu d'au moins 1 élément visuel qui utilise un shader de géométrie.

□

5 Caméra

5.1 Point de vue

Il est possible de transformer une caméra par rapport à une ou des entités géométriques et l'utilisateur peut manipuler interactivement la caméra pour voir la position centrale de la sélection de différents points de vue, en plus de pouvoir s'en approcher et s'en éloigner.

□

5.2 Mode de projection

Au moins deux types de mode de projection (ex. perspective et orthogonale) sont supportés par l'application et offrent à l'utilisateur un point de vue cohérent de la scène.

□

5.3 Agencement

Il est possible de voir une scène de plusieurs points de vue différents, en pleine fenêtre ou avec des sous-fenêtres d'affichage dans la fenêtre principale de l'application où le rendu est fait en simultané dans chaque fenêtre.

□

5.4 Occlusion

L'application utilise une technique d'occlusion autre que celles de base pour tenter de minimiser le nombre d'éléments du graphe de scène à rendre du point de vue d'une caméra (ex. BVH, quadtree, octree, etc).

□

5.5 Portail

Un moins 1 effet visuel présent dans une scène est réalisé à partir d'une caméra qui rend dans une texture du contenu qui est ensuite affiché dans une scène rendue par une autre caméra (ex. portail, miroir, effet de réflexion, écran de surveillance, etc.).

□

6 Texture

6.1 Coordonnées de texture

Il existe au moins un modèle texturé avec des coordonnées de mapping adéquatement distribuées sur la surface du maillage géométrique, à l'exception de ceux-ci : triangle, plan, quad, cube.

□

6.2 Filtrage

L'application permet d'appliquer au moins 3 types d'algorithmes de traitement d'image ou de filtrage qui affecte l'apparence d'une texture utilisée dans le rendu d'au moins 1 élément visuel d'une scène.

□

6.3 Mappage tonal

L'application permet d'appliquer un algorithme de mappage tonal (*tone mapping*) sur le contenu de la fenêtre d'affichage ou sur une image.

□

6.4 Cubemap

Il existe au moins 1 texture de type cubemap qui est utilisée dans le rendu d'au moins 1 élément visuel d'une scène.

□

6.5 Texture procédurale

Au moins 1 texture utilisée dans le rendu d'au moins 1 élément visuel d'une scène a été de manière procédurale par un algorithme, sans recours à un fichier image.

□

7 Illumination classique

7.1 Modèles d'illumination

Le rendu d'au moins 3 éléments visuels est fait à partir d'au moins 3 modèles d'illumination différents (ex. Lambert, Gouraud, Phong, Blinn-Phong ou autre). ☐

7.2 Matériaux

Au moins 3 éléments visuels d'une scène ont une surface avec un matériau sélectionné parmi un ensemble d'au moins 3 matériaux différents. ☐

7.3 Types de lumière

Il est possible d'avoir au moins une instance de 4 types de lumières différents (ambiante, directionnelle, ponctuelle, projecteur ou autre). ☐

7.4 Lumières multiples

Possibilité d'avoir au moins 4 différentes instances de lumière dynamique dont la couleur, la position et l'atténuation sont prises en compte lors des calculs d'illumination. ☐

7.5 Réflexion anisotropique

Au moins 1 matériau a une propriété de réflexion anisotropique. ☐

8 Lancer de rayon

8.1 Intersection

L'application est capable de calculer le point d'intersection entre un rayon et au moins 3 types de primitives géométriques.

□

8.2 Réflexion

Une technique de rendu inspirée des principes du lancer de rayon est utilisée pour rendre au moins 1 effet de réflexion (ex. une surface miroir).

□

8.3 Réfraction

Une technique de rendu inspirée des principes du lancer de rayon est utilisée pour rendre au moins 1 effet de réfraction. (ex. une surface en verre)

□

8.4 Ombrage

Une technique de rendu inspirée des principes du lancer de rayon est utilisée pour rendre au moins 1 effet d'ombrage.

□

8.5 Illumination globale

Une technique de rendu inspirée des principes de l'illumination globale est utilisée pour rendre au moins 1 scène.

□

9 Topologie

9.1 Courbe paramétrique

Il est possible de rendre au moins 1 type de courbe paramétrique avec plus de 4 points de contrôle, par exemple une spline de Bézier ou de Catmull-Rom.

□

9.2 Surface paramétrique

Il est possible de rendre au moins 1 surface paramétrique, par exemple une surface de Bézier bicubique ou une surface de Coons.

□

9.3 Shader de tessellation

Un shader de tessellation permet de sous-diviser la géométrie d'au moins 1 modèle, par exemple un plan ou une surface paramétrique.

□

9.4 Triangulation

Un algorithme permet de générer au moins 1 maillage triangulaire à partir d'un ensemble de sommets, par exemple une triangulation de Delaunay ou un diagramme de Voronoï.

□

9.5 Effet de relief

Il est possible de rendre au moins 1 modèle où une ou des textures sont utilisées pour simuler un effet de relief sur la surface, par exemple les techniques du *displacement mapping* ou du *normal mapping*.

□

10 Illumination moderne

10.1 HDR

Les calculs d'illumination sont faits dans un espace de couleur linéaire et HDR.



10.2 IBL

L'application utilise au moins un cubemap HDR pour simuler une source d'éclairage qui affecte la surface d'un matériau.



10.3 Métallicité

Au moins un matériau utilise un facteur de métallicité pour simuler le niveau de réflexion entre diélectrique et métallique.



10.4 Microfacettes

Au moins un matériau utilise avec des microfacettes pour simuler la rugosité de sa surface.



10.5 Rendu en différé

L'application utilise une architecture de rendu en différé.

