

Cours :	INF 707 Fondements d'infographie appliquée
Trimestre :	Automne 2020
Enseignant :	Mykel Leclerc Brisson

1. Mise en contexte

L'industrie du jeu vidéo a toujours placé les aspects infographiques au premier plan lors du développement d'applications. Même si, aujourd'hui, d'autres aspects tels l'intelligence artificielle, l'animation ou la physique sont tout aussi importants, les techniques infographiques sont centrales au développement d'un jeu vidéo. En plus d'avoir un impact direct sur la qualité visuelle et le niveau d'immersion, elles déterminent les techniques de production du contenu (textures, modèles, effets spéciaux, etc.).

L'un des plus importants défis de l'industrie est d'ailleurs de conserver un certain leadership quant à la maîtrise de technologies telles que le 3D accéléré, les langages intégrés sur les cartes graphiques et les effets graphiques toujours plus avancés. Parallèlement à toutes ces innovations, les développeurs doivent aussi conserver une certaine forme de compatibilité et de standardisation pour leurs solutions. L'utilisation de bibliothèques spécialisées telles DirectX et OpenGL est très répandue, et est devenue l'élément clé de l'infographie moderne.

2. Place du cours dans le programme

Le cours présume chez l'étudiant (étudiante) une connaissance de la programmation, et du langage C++ en particulier.

Concomitante : le cours Concepts mathématiques et physiques appliqués en jeu vidéo (INF 706).

Les notions mathématiques et physiques du 3D étant essentielles pour INF 707, le cours INF 706 est un complément tout aussi essentiel. Notez toutefois que certaines notions mathématiques et physiques seront reprises (souvent de façon simplifiée) de façon à couvrir les spécificités des implantations en environnement de développement.

Concomitante : le cours de Conception orientée objet avancée (INF 737).

Même si l'approche orientée objet n'est pas une priorité pour INF 707, elle sera fortement recommandée et prendra une importance croissante au cours des apprentissages. Plusieurs des techniques OO introduites dans le cours ci-haut mentionné seront intégrées dans les activités de développement.

Le cours INF 707 met en place les concepts et éléments pratiques qui permettront à l'étudiant (étudiante) de poursuivre de façon plus approfondie son apprentissage des techniques liées aux jeux vidéo.

3. Objectifs généraux

Ce cours permettra aux étudiants (étudiantes) du programme de « Diplôme de développement du jeu vidéo » d'acquies les bases conceptuelles et pratiques permettant l'intégration de l'infographie tridimensionnelle dans les applications informatiques.

À travers le développement d'un jeu vidéo en 3D, les étudiants (étudiantes) seront amenés à résoudre des problèmes spécifiques à l'infographie 3D. L'application devra être interactive et fluide, la qualité de l'image satisfaisante et les différentes tâches correctement synchronisées.

4. Objectifs spécifiques

À la fin de ce cours, l'étudiante ou l'étudiant sera capable de :

- Comprendre comment une trame est rendue dans un jeu vidéo.
 - Comprendre le pipeline de rendu et ses différentes étapes.
 - Comprendre comment une liste de sommets est transformée en pixels à l'écran.
 - Comprendre comment le pipeline de rendu est manipulé avec DirectX 11.
- Appliquer les concepts de bases en algèbre linéaire.
 - Afficher les objets au bon endroit et les animer.
 - Manipuler une caméra virtuelle.
- Programmer des *shaders* de complexité simple à intermédiaire.
 - *Vertex shader* et *pixel shader*.
- Comprendre comment les données sont structurées pour afficher un objet dans une scène 3D.
 - Textures
 - Maillages. (*Vertex buffer* et *index buffer*)
- Appliquer des techniques de base de gestion de scènes comme le *frustum culling*.
- Comprendre les bases de l'éclairage d'une scène virtuelle et appliquer un modèle simple d'éclairage. (Phong)
- Comprendre et appliquer des techniques de *blending*.
- Affichage des *sprites* — technique de *billboard*.
- Utiliser un débogueur graphique pour mieux comprendre le rendu et déboguer les *shaders*.
- Appliquer les concepts d'infographie 3D dans une application interactive.

5. Planification hebdomadaire

Note : l'organisation hebdomadaire est donnée à titre indicatif. Les deux ou trois premières séances devraient respecter la planification, mais par la, suite certains sujets pourront être écourtés ou étirés et occasionnellement changés d'ordre, selon les besoins du cours et des projets.

Séance 1 Introduction

Introduction au pipeline de visualisation. Applications interactives temps réel. Survol des technologies matérielles. Terminologie 3D. Survol des technologies logicielles. Design d'une application 3D. Présentation de DirectX. DirectX 11 versus DirectX 12. Présentation du squelette de PetitMoteur3D.

Références : [RTR] Chapitre 2
[ADX] Introduction et chapitre 1

Séance 2 Techniques de visualisation dans une application interactive

Modèles d'applications. Outils de développement. Scène et mise en scène. Transformations géométriques de base. Introduction à l'animation. Présentation de la bibliothèque Direct3D.

Atelier : Un modèle de base d'animation 3D [ADX] Chap. 2

Références : [RTR] Chapitre 4
[DXD] Programming Guide - Direct3D Architecture
[ADX] Chapitres 2 et 3

Séance 3 Paramètres 3D

Caméra et matrice de vision. Projection et matrice de projection. Volume de vision. Attributs de vision. Dispositif de rendu. Dispositifs sous DirectX.

Atelier : Premiers pas avec Direct 3D. [ADX] Chap. 3

Références : [RTR] Chapitre 4
[DXD] Programming Guide - Direct3D Devices
[ADX] Chapitres 3 et 4
[CG] Chapitre 6

Séance 4 Affichage d'objets 3D

Composition des objets. Points, vecteurs, sommets, normales et autres attributs. Matrice de transformation dans le monde. Construction d'une primitive. Introduction au retrait des surfaces cachées (*backface culling*). Introduction aux *shaders*. Premiers *shaders*.

Références : [RTR] Chapitre 12
[DXD] Programming Guide - Direct3D Resources-Vertex and Index Buffers
[ADX] Chapitre 4

Séance 5 Cartes graphiques et retrait des surfaces cachées

Énumération et sélection spécialisée d'un dispositif. Sélection des paramètres. Caractéristiques des cartes graphiques. Gestion des dispositifs. Introduction aux états de rendu (*render states*). Configuration et sélection d'un tri de profondeur. Algorithme du peintre. Z-Buffers et W-Buffers. Autres algorithmes et impacts sur le design de l'application.

Atelier : Contrôle de l'affichage. [ADX] Chap. 5

Références : [RTR] Chapitre 3
[DXD] Programming Guide — Direct3D Devices — States
[ADX] Chapitres 4 et 5

Séance 6 Éclairage

Sources d'éclairage. Matériaux. Éclairage vs lissage. Algorithmes de lissage (plat, Gouraud, Phong). Aliasing et anti-aliasing. Transparence. Brume.

Atelier : Éclairage. [ADX] Chapitre 6

Test #1 : Chapitres 1-5 (45 minutes)

Références : [RTR] Chapitre 5
[DXD] Programming Guide – Lights and materials
[DXD] Programming Guide – Fog types
[ADX] Chapitre 6

Séance 7 Textures et dispositifs d'entrées

Application des textures. Format des textures. Transparence. Paramètres de rendu des textures. *Mipmaps*. Dispositifs d'entrée.

Atelier : Textures. [ADX] Chap. 7

Références : [RTR] Chapitre 7
[DXD] Programming Guide – Lights and materials
[ADX] Chapitre 7 et 10

Séance 8 Importation d'objets (maillages ou *meshes*)

Formats d'objets. Volume d'un objet (*bounding box* et *bounding sphere*). Introduction aux hiérarchies d'objets.

Atelier : Maillages et objets 3D. [ADX] Chapitre 8

Test #2 : Chapitres 1-7 (45 minutes)

Références : [DXD] Programming Guide – D3DX
[ADX] Chapitre 8

Séances 9 Rendu d'images 2D & techniques avancées d'éclairage et de lissage, introduction

Surfaces de rendu avancées. Panneaux. Affichage du texte. Sprites. Particules (*point sprite*). Ciels et autres effets. Orientation des panneaux. Ombres et réflexions. Types de lissage avancés. Post-effects.

Atelier : Affichage 2D. [ADX] Chapitre 9

Références : [RTR] Chapitres 7, 8, 9 et 10
[DXD] Programming Guide – D3DX
[ADX] Chapitre 9, 10, 11, 12

Séance 10 Examen final

Introduction aux langages de lissage (*shading langages*). Types de lissage avancés. Support pour les *shaders* sur le matériel. Introduction au HLSL. Différentes façons d'intégrer des *shaders* dans les applications.

Références : [RTR] Chapitre 3
[DXD] Programming Guide – HLSL Shaders

6. Approche pédagogique préconisée

L'objectif principal de ce cours étant l'intégration des concepts d'infographie dans une application interactive de type jeu vidéo, les séances s'organiseront autour de trois pôles d'activité :

- Présentation des concepts théoriques en classe avec liens avec les ateliers.
- Présentation de certains éléments clés de la bibliothèque DirectX.
- Présentation et révision des ateliers pratiques. Ceux-ci sont habituellement donnés à l'étudiante (l'étudiant) sous forme de travail pratique formatif. Les ateliers pratiques se présentent sous deux formes : les tutoriels d'apprentissage et les travaux complémentaires permettant de raffiner certaines techniques. L'étudiant (étudiante) doit normalement réaliser le tutoriel dans la semaine précédant le cours, et le travail complémentaire dans la semaine suivante.

De plus, un travail pratique plus avancé, comptant dans l'évaluation, sera effectué par l'étudiante (étudiant). Ce travail sera effectué en trois volets et permettra à l'étudiant (étudiante) de réaliser une application complète, de dimension modeste mais possédant tous les concepts de base d'un jeu 3D (au niveau de l'infographie).

7. Évaluation de l'apprentissage

Les méthodes d'évaluations sont les suivantes :

- a) Un examen individuel de 15 %. L'objectif est de réviser tout particulièrement les matières acquises dans les trois cours précédents, mais l'examen peut référer à l'ensemble de la matière couverte. Il s'agira de quelques questions (de 3 à 5) à développement. Une ou plusieurs courtes sections de code (programmation) peuvent y être demandées. Le temps alloué pour cet examen est d'environ quarante-cinq minutes.

- b) Un examen individuel de récapitulation de 10 %. Cet examen a pour but de réviser la matière avant l'examen final et de préparer l'étudiant à certains types de problèmes pratiques.
- c) Un travail pratique (55 % de la note finale) sera réalisé en trois volets plus un certain nombre d'éléments (« rushes ») implantés en cours de session à partir des ateliers. Chaque volet est corrigé aux dates de livraison. Le travail d'équipe est fortement recommandé pour ce travail. Les modalités complètes du travail pratique ainsi que la grille d'évaluation seront données dans le devis du travail **IMPORTANT** : la note allouée pour ce travail demeure une note **individuelle**.
- d) Un examen final récapitulatif et individuel (20 % de la note finale) se tiendra à la dernière séance.

Donc :

<i>Examens</i>	<i>Pondération</i>	<i>Références</i>	<i>Notes et détails</i>
Test #1	15 %	Séance 6 (de 1 à 5)	Individuels, documentation ¹
Test #2	10 %	Séance 8 (Préparation)	
Examen final	20 %	Séance 10 (Récapitulatif)	

<i>Travail</i>	<i>Pondération</i>	<i>Références</i>	<i>Type d'évaluation</i>
Volet 1	(15 %)	Design et prototypage	Travail en équipe, objectifs, qualité et délais ²
Volet 2	(30 %)	Modélisation	
Volet 3	(40 %)	Rendu et effets	
« <i>Rushes</i> »	(15 %)	Activités d'intégration au projet.	Activités individuelles ou en équipe en cours de projet, objectifs, qualité et délais ²

Note 1 : À cause de la nature de la matière, et parce que les questions sont construites en conséquence, la documentation de DirectX 11 est permise.

Note 2 : Pour le travail pratique, les pénalités relatives aux retards s'appliqueront de la façon suivante : -12 points (/100) par jour ouvrable (lundi au vendredi) pour un retard non-signalé, -5 points (/100) par jour ouvrable (lundi au vendredi) pour un retard justifié et négocié à l'avance.

Toute modification reliée à une date de remise doit avoir été acceptée par le groupe et la direction du CeFTI dans un délai plus grand qu'une semaine avant l'échéance de la remise.

8. Plagiat

Un document dont le texte et la structure se rapportent à des textes intégraux tirés d'un livre, d'une publication scientifique ou même d'un site Internet, doit être référencé adéquatement. Lors de la correction de tout travail individuel ou de groupe une attention spéciale sera portée au plagiat, défini dans le Règlement des études comme « le fait, dans une activité pédagogique évaluée, de faire passer indûment pour siens des passages ou des idées tirés de l'œuvre d'autrui. “Le cas échéant, le plagiat est un délit qui contrevient à l'article 8.1.2 du Règlement des études : ‘tout acte ou manœuvre visant à tromper quant au rendement scolaire ou quant à la réussite d'une exigence relative à une activité pédagogique.’”. À titre de sanction disciplinaire, les mesures suivantes peuvent être imposées : a) l'obligation de reprendre un travail, un examen ou une activité pédagogique et b) l'attribution de la note E ou de la note 0 pour un travail, un examen ou une activité évaluée. Tout travail suspecté de plagiat sera référé à la vice-doyenne à l'enseignement de la Faculté des sciences.

9. Adresse électronique

mykel.leclerc.brisson@usherbrooke.ca

10. Bibliographie

Références

[RTR] Möller, Haines, **Real time rendering** (fourth edition), A K Peters/CRC Press, 2018, ISBN-13 : 978-1138627000

Référence théorique sur les concepts actuels de l'infographie, la notion de rendu en temps réel en rend aussi l'approche comparable à celle que nous vivons au niveau du jeu vidéo. Le site Web du livre (www.realtimerendering.com) est très dynamique et aussi une excellente source de liens sur l'actualité de l'infographie.

[ADX] Jean, François, **Ateliers Infographie avec DirectX 11**

Ce document contient les tutoriels nécessaires aux travaux pratiques formatifs ainsi que la correspondance entre les notions théoriques et la bibliothèque graphique (Direct3D).

[DXD] Documentation en ligne **DirectX 11 ++ Documentation**

Il s'agit d'une référence indispensable, incluse avec le SDK de DirectX (SDK Windows Platform). L'étudiant se doit d'être familier avec cette documentation.

- [1] Marschner, Steve ; **Fundamentals of Computer Graphics** (Fourth Edition), A K Peters/CRC Press, 2015, ISBN-13 978-1482229394
- [2] Luna, Frank D., **Introduction to 3D Game Programming with DirectX 11.0**, Mercury Learning, ISBN 978-1-9364202-2-3
- [3] Kink, Jason, **Practical Rendering and Computation with Direct 3D 11**, CRC Press, ISBN 978-1-56881-720-0
- [4] Engel, Wolfgang, **ShaderX3: Advanced Rendering with DirectX and OpenGL**, Delmar Thomson Learning, ISBN 1-58450-357-2