

# Plan de cours

|              |                                                |
|--------------|------------------------------------------------|
| Cours :      | INF709 – Concepts spécialisés de programmation |
| Trimestre :  | Hiver 2025                                     |
| Enseignant : | Patrice Roy                                    |

## 1. Mise en contexte

La compétitivité du marché dans lequel se trouve plongée l'industrie du développement du jeu vidéo est clairement un facteur par lequel cette industrie peut se définir. Les produits sont mis sur le marché à un rythme effréné, la compétition est féroce, et le caractère grand public de chaque produit le place face à une horde de critiques potentielles. Il y a, dans de marché, peu de place pour l'erreur.

Le cours INF709 – Concepts spécialisés de programmation, malgré toute la bonne volonté de son professeur, ne peut assurer à lui seul la réussite d'un projet de développement de jeu vidéo. Une telle activité dépend d'une multitude de facteurs, et l'expertise technique n'est que l'un d'entre eux. Cela dit, nous travaillerons tous de concert dans ce cours afin que les techniques de programmation qui y seront enseignées et mises en pratique profitent à chaque individu assistant au cours.

Un ensemble de stratégies y seront couvertes :

- Multiprogrammation
- Écriture de code résilient
- Techniques de métaprogrammation
- Écriture de code et de structures de données compactes
- Gestion efficiente des erreurs
- Identification et mesure des lieux stratégiques dans un programme
- Calibration des programmes
- Techniques d'optimisation diverses
- Diverses techniques de programmation sophistiquées

## 2. Place du cours dans le programme

L'étudiant(e) doit rencontrer les préalables d'inscription au Diplôme de développement du jeu vidéo (DDJV).

Il est présumé que l'étudiant(e) a au moins réussi le cours INF737 – Conception orientée objets avancée, du fait que le contenu du cours INF709 construit en partie sur les fondations mises en place dans le cours INF737.

Les cours INF737 et INF709 ont d'ailleurs en commun l'objectif spécifique suivant du DDJV : acquérir des connaissances sur les méthodes et outils utilisés pour spécifier, concevoir et implanter des jeux vidéo.

Il est probable que certaines applications du cours INF709 se fassent directement dans d'autres cours de la même session, par exemple pour faciliter la réalisation de certaines tâches numériques coûteuses en arrière-plan, de manière prédictive, pendant que la dynamique de jeu se poursuit normalement, ou pour simplifier l'application de certains algorithmes spécialisés sur des séquences.

De manière plus générale, il est attendu que l'étudiant(e) est *a priori* habile avec le langage C++, incluant quelques éléments-clés du standard le plus récent de ce langage.

## 3. Descripteur du cours

Le cours INF709 a pour objectif d'amener l'étudiant(e) à développer des techniques de programmation qui lui permettront de produire plus rapidement des programmes plus simples et de meilleure qualité. La visée est donc une connaissance appliquée et applicable à des problèmes réels.

Les jeux vidéo sont des systèmes dont la dynamique doit être particulièrement fluide. Ce cours abordera donc des techniques qui permettront à un système de tirer profit de stratégies de multiprogrammation pour réaliser des tâches en arrière-plan sans interrompre l'action en avant-plan.

L'optimisation et l'équilibre systémique peuvent être des objectifs antagonistes. Nous chercherons à définir un point de vue architectural, holiste, plutôt que de nous limiter à une approche d'optimisation locale (sans toutefois exclure cette dernière).

## 4. Objectifs spécifiques

À la fin de cette activité pédagogique, l'étudiant sera capable de :

- Rédiger une solution logicielle à la fois simple et rapide applicable à un large éventail de problèmes informatiques susceptibles d'apparaître dans un contexte de développement du jeu vidéo.
- Concevoir et implémenter un système réactif.
- Identifier les optimisations locales ou systémiques désirables.
- Procéder à des optimisations locales et systémiques.
- Respecter des contraintes de type temps réel en situation monolithique.
- Respecter des contraintes de type temps réel en situation multiprogrammée.

## 5. Contenu

Le contenu prévu pour le cours inclut :

- Considérations d'optimisation en temps et en espace
- Bases de programmation parallèle et concurrente
- Techniques d'organisation des données
- Accès efficace à la *Cache*
- Considérations d'effacement de types
- Exploration de l'implémentation de types sommes et de types produits
- Exploration de l'implémentation de types de pointeurs intelligents
- Exploration de structures de données hybrides mais utiles dans des systèmes complexes comme le sont les moteurs de jeux vidéo

Nous prendrons soin de voir en détail comment utiliser ces outils, mais aussi comment les implémenter, dans le but de mieux comprendre les enjeux dans chaque cas.

## 6. Approche pédagogique préconisée

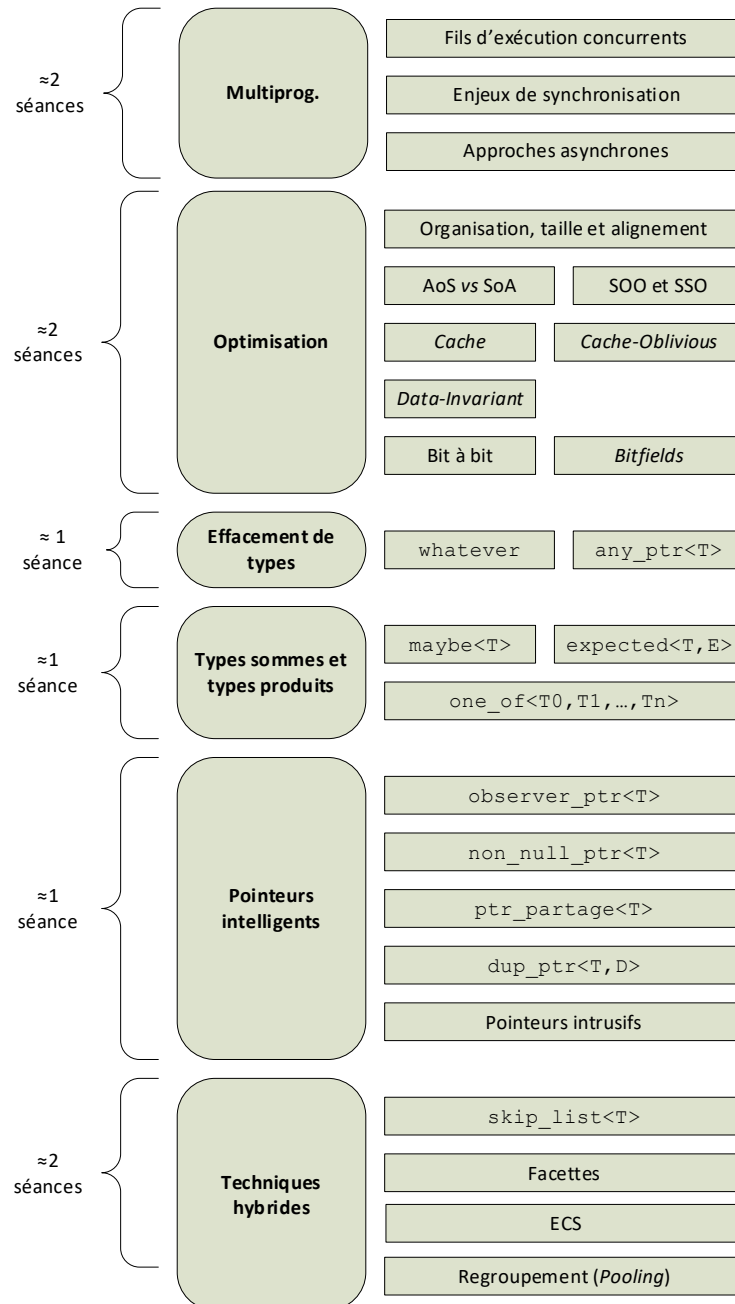
Pour favoriser l'intégration des nombreux concepts au menu, ce cours respectera essentiellement le modèle suivant :

- Exposés magistraux en classe, où les étudiant(e)s sont fortement encouragés à contribuer par leurs questions et commentaires
- Travaux pratiques et exercices à teneur formative, qui permettront aux étudiant(e)s de mesurer concrètement leur compréhension de la matière explorée, et qui pourront être corrigés par le professeur ou autocorrigés à l'aide d'une grille de vérification, selon le cas
- Travaux pratiques à teneur sommative, évalués par le professeur en fonction des mêmes critères que ceux appliqués dans le cadre des activités formatives. Ces travaux seront soumis à un échéancier de livrables et feront partie de votre projet intégrateur
- Des questions de réflexion (et parfois à saveur technique) sur une base quasi hebdomadaire, et
- Un contrôle théorique récapitulatif, en toute fin de parcours, vérifiant formellement l'atteinte des objectifs

Les questions et contrôles présumeront que chaque membre d'une équipe a contribué activement à la réalisation de chacun des travaux pratiques, et a bien compris les implications philosophiques et techniques de ces travaux.

## 7. Planification

Notez que la planification qui suit est illustrative, et veut donner un aperçu du contenu prévu du cours. Le tout sera ajusté en fonction des événements.



## 8. Évaluation de l'apprentissage

De petits tests, souvent d'une seule question, auront lieu sur une base relativement régulière, soit une par semaine, la plupart des semaines.

**Minitests**  
(40%)

Chaque question portera sur le thème de la semaine précédente ou des deux semaines précédentes. Les questions seront surtout orientées sur la réflexion, mais la technique à proprement dit s'y glissera à l'occasion.

Le poids de chacun de ces petits tests sera une fraction de la note finale. En général, le temps alloué pour y répondre sera d'environ 15 minutes, au début du cours.

Les travaux pratiques de ce cours seront directement intégrés au projet de session, activité transversale qui vous occupera toutes et tous jusqu'à la toute fin. Vous devrez toutefois livrer deux « travaux pratiques » faisant état de votre progression. Les exigences techniques de ces travaux sont indiquées à même le document décrivant les consignes pour ce projet, et les règles de remise seront indiquées sur le site du cours.

**Travaux** (30%)

Un examen final récapitulatif valant 30% de la note finale aura lieu lors de la dernière séance de la session.

**Examen** (30%)

Aucun retard ne sera toléré dans la remise des travaux pratiques.

Tout travail devra être produit dans un français jugé de bonne qualité. Une pénalité allant jusqu'à 5% des points pourra être appliquée à un travail produit dans un français ne rencontrant pas les standards de qualité de la faculté des sciences.

Les règles de qualité des programmes qui seront distribuées en cours de session seront applicables aux travaux pratiques et au code rédigé dans le cadre des contrôles.

**Un pourcentage de chaque extrant sera consacré à la qualité du français et au format (présentation).**

**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.**

## 9. 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 idées ou des passages 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 transmis au vice-doyen à l'enseignement de la Faculté des sciences.

## 10. Contact électronique

Adresse de courriel : [Patrice.Roy@USherbrooke.ca](mailto:Patrice.Roy@USherbrooke.ca)

## 11. Médiagraphie suggérée

Les notes de cours qui vous seront distribuées constitueront votre référence principale pour la session qui s'annonce. D'autres références vous sont proposées sous la forme d'une *médiagraphie* commentée sur le site suivant :

<http://h-deb.ca/Liens/Suggestions-lecture.html>

Parmi les quelques suggestions que vous trouverez sur ce site, portez particulièrement attention aux volumes suivants :

- **C++ 17 The Complete Guide**, de *Nicolai Josuttis*
- **C++ 17 Templates, The Complete Guide**, de *Daveed Vandevoorde, Nicolai Josuttis et Douglas Gregor*
- **Effective C++**, **Effective Modern C++** et **Effective STL**, de *Scott Meyers*
- **Exceptional C++**, de *Herb Sutter*
- **Modern C++ Design**, par *Andrei Alexandrescu*
- **C++ Coding Standards**, par *Herb Sutter* et *Andrei Alexandrescu*
- **C++ Concurrency in Action, Practical Multithreading**, par *Anthony Williams*
- **The C++ Programming Language**, par *Bjarne Stroustrup*
- **The C++ Standard Library 2<sup>nd</sup> Edition**, par *Nicolai M. Josuttis*

### Site Web du cours et autres références

Le site Web du cours devrait être, avec les notes de cours en tant que telles, votre référence principale :

<http://h-deb.ca/UdeS/CSP>

Je déposerai entre autres sur ce site des références électroniques me semblant pertinentes. Je vous invite donc fortement à le consulter sur une base régulière.

Mon site principal est :

<http://h-deb.ca/>

Dans tous les cas, faites attention aux majuscules et aux minuscules puisque l'adresse est sensible à la casse.