

8INF955 - Principes de conception et de développement de jeux vidéo ¹

Objectifs du cours

Amener l'étudiant à maîtriser les principes particuliers de conception et de développement logiciel dans le domaine des jeux vidéo. Sensibiliser l'étudiant aux pratiques de l'industrie, aux principes de design et de modélisation, aux outils et moteurs commerciaux utilisés dans le développement des jeux sur différentes plateformes. Initier l'étudiant aux défis liés à la recherche et au développement inhérent au domaine des jeux.

Contenu général

Architecture logicielle d'un jeu vidéo (gameloop, jetons, états, moteurs, etc.). Processus de développement dans l'industrie des jeux vidéo et gestion d'un projet de développement d'un jeu. Approches de développement Agile et SRCUM. Outils technologiques, bibliothèques et langages utilisés en industrie (ex. Unity 3D, UDK, etc.). Les moteurs de jeux. Théorie des jeux. Design de jeux. Étude sur la jouabilité « Gameplay ». Équilibre dans la structure des éléments d'un jeu (pointages, options, défis versus habilités, ajustement dynamique, chances, etc.). Ergonomie des interfaces de jeux. Les « serious games » et autres jeux à objectifs non ludiques. Algorithmes de base de l'intelligence artificielle appliquée aux jeux vidéo. Recherche et développement dans le domaine des jeux vidéo.

Approche du cours

Ce cours s'adresse à une clientèle de professionnels des technologies de l'information possédant un diplôme de premier cycle universitaire en informatique et voulant se spécialiser dans la conception et le développement de jeux vidéo. Il s'effectuera en deux volets parallèles. Le premier volet, couvert par les cours magistraux, permettra aux étudiants de bien comprendre tous les aspects théoriques et algorithmiques du processus de conception et de développement d'un jeu vidéo, de l'idée originale à sa distribution. Le second volet du cours, couvert par un ensemble de travaux pratiques, aura pour objectif de familiariser l'étudiant aux outils technologiques (moteurs de jeux, techniques de programmation, etc.) utilisés en industrie. Le but visé est de développer une expertise technique suffisante afin d'être apte à réaliser en équipe un projet intégrateur de développement de jeux dans la deuxième partie du cours. Lors de ces travaux, l'étudiant sera amené à programmer de nouvelles fonctionnalités à travers des canevas de jeux préconçus, à travailler avec des technologies et moteurs 2D et 3D utilisés en industrie, comme c'est le cas en entreprise. Enfin, comme dans la majorité des cours gradués, la portion finale du cours portera sur la réalisation d'un projet intégrateur des connaissances ainsi que sur une étude d'un article portant sur la recherche scientifique liée au domaine des jeux vidéo.

¹ Ce cours est optionnel dans les programmes 3017 (Maîtrise en informatique) et 3775 (D.E.S.S. en informatique appliquée). Il est obligatoire dans la concentration « développement de jeux vidéo » de la maîtrise en informatique.

Au terme du cours l'étudiant aura :

- Une connaissance claire du processus de développement, des documents et des pratiques en industrie.
- Une expérience de base en programmation de jeux avec des outils utilisés en industrie.
- Une initiation aux pratiques de design, d'équilibrage et de conception d'interfaces de jeux.
- Réalisé un projet de développement d'un jeu en utilisant des outils industriels.
- Un aperçu des problématiques et enjeux liés à la recherche scientifique dans le domaine.

Sommaire concernant le contenu des cours magistraux

Le plan suivant donne un aperçu de l'organisation temporelle des activités d'apprentissages liées aux séances magistrales. Bien que nous respecterons en gros cette structure générale, les contenus particuliers peuvent changer pour répondre aux besoins. Les laboratoires auront normalement lieu au local (P3-3100).

- **Section 1**

Introduction, présentation du cours, des objectifs, du syllabus.

Présentation du professeur et de ses recherches.

Table ronde sur les appréhensions étudiantes vis-à-vis du cours.

Initiation à la problématique : qu'est-ce qu'un jeu?

Processus de production d'un jeu en entreprise: de l'idée à la distribution.

Usine de développement (pipeline). Penser avec les tokens, jalons et dates limites.

Méthode de développement utilisée en industrie Agile, le SCRUM, XP programming.

- **Section 2**

Anatomie des composantes d'un jeu vidéo : jetons, la boucle de jeu, collisions, etc.

Développement logiciel lié au domaine des jeux vidéo.

Principaux algorithmes et structures de données liés à la gestion des jeux.

Laboratoire 1 (normalement *cours 3 – mardi 27 janvier) portant sur la matière vue en classe.

- **Section 3**

Ergonomie des interfaces servant pour l'interaction avec le joueur.

Laboratoire 2 (*cours 6 – mardi 17 février) portant sur la matière vue en classe.

- **Section 4**

Jouabilité « gameplay » : étude de cas dans différents jeux.

Gestion de l'incertitude et utilisation du hasard dans les jeux vidéo.

Équilibre dans les jeux et répartition balancée des éléments.

Laboratoire 3 (*cours 8 – mardi 10 mars) portant sur la matière vue en classe.

- **Section 5**

Méthodologie d'assurance qualité : tests automatisés, tests unitaires, régressions, la réutilisation, etc.
Maintenance post-développement: aspects fondamentaux (support, mises à jour, expansions, etc.).

Bilan (*Post-mortem*) et processus d'apprentissage au terme d'un projet de jeu.

Laboratoire 4 (cours 10 – lundi 10 novembre) portant sur la matière vue en classe.

- **Section 6**

Conférencier invité en provenance de l'industrie du jeu vidéo.

Temps alloués pour le développement de votre projet intégrateur.

- **Section 7**

Présentation orale des projets de jeux réalisés en équipe.

Chaque équipe devra réaliser un « pitch » professionnel de son projet comme en entreprise.

Table ronde et critique constructive des différents articles présentés.

Objectifs spécifiques des séances pratiques en laboratoire

- Renforcer les bases de programmation en matière de jeux vidéo.
- Introduire l'étudiant à l'exploitation de bibliothèques et moteurs graphiques 2D et 3D.
- Prendre contact avec des outils de programmation utilisés en industrie et relativement complexes.
- Rappeler les notions mathématiques fondamentales qui sont utiles pour la programmation de jeux.
- Interagir en synergie avec différents moteurs et bibliothèques.
- Familiariser l'étudiant à l'utilisation d'un système de contrôle de version.
- Préparer l'étudiant à mener de front un projet intégrateur en utilisant des outils de haut niveau.

Matériel requis

Il n'existe aucun livre couvrant entièrement la matière vue dans ce cours qui est plutôt unique et qui est en partie basé sur l'expérience. Par conséquent, aucun livre n'est obligatoire, des notes ciblées de cours ainsi que des références Web et littéraires seront suggérées. Par contre, les livres contenus dans la bibliographie constituent des références solides sur des portions spécifiques du contenu. Par exemple, le livre de Andrew Rollings constitue une référence clé en matière de conception de jeux vidéo. Le livre intitulé « Core Techniques and Algorithms in Game Programming » constitue une bonne introduction au niveau de la programmation. Les trois dernières références sont des incontournables pour tous les développeurs.

Utilisation des TIC : usage permis si non perturbant.

Bibliographie

A. Rollings & E. Adams, *Fundamentals of game design*: 2nd edition, Berkeley, 2009.

D. Dalmau, *Core Techniques and Algorithms in Game Programming*, 2003.

Rules of Play: Game Design Fundamentals, MIT press, 2003.

Raph Koster, *A Theory of Fun for Game Design*, O'Reilly & Associates, 2004.

E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns*, Addison-Wesley, 1994.

M. Fowler, et al., *Refactoring, Improving the design of existing code*, Addison-Wiley, 2000.

Robert C. Martin, *Clean Code: A Handbook of Agile Software Craftsmanship*, 2008.

Références Web

Game Developers: <http://www.gdmag.com/homepage.htm>

GameDev.net: <http://www.gamedev.net/>

Techno compétences : <http://www.technocompetences.qc.ca/>

Gamasutra: <http://www.gamasutra.com/>