# PLAN CADRE

Mai. 2021

**Titre du cours** : Physique dans le jeu vidéo

**Code du cours** : 203-J28-BB

**Discipline** : Physique **Programme :** Programmeur de jeux vidéo

LEA.C7-H21

**Préalables** : Algèbre vectorielle appliquée(201-J25-BB), Programmation 2 (420-J26-BB)

**Préparatoire à** : Intelligence Artificielle 1(420-J32-BB)

**Corequis :** aucun

**Durée** : 60 heures

**Pondération :** 2-2-4

**Unités :**

|  |  |  |  |
| --- | --- | --- | --- |
| **Code de la compétence** | **Compétence** | **Entièrement** | **Partiellement** |
| BAYM | Appliquer des notions de mécanique classique et des méthodes numériques pour simuler des objets physiques en mouvement | ☒ | 🞎 |

**Éléments de compétence(s) concerné(s)** :

1. Décrire les lois de la physique mécanique et les propriétés des corps rigides.
2. Résoudre numériquement les équations du mouvement.
3. Utiliser l’algèbre vectorielle pour simuler des collisions entre des objets.
4. Expliquer le fonctionnement d’un moteur de physique

**NOTE PRÉLIMINAIRE**

### Contribution du cours au programme de formation

Le cours *Physique dans le jeu vidéo* (203-J28-BB) est donné à la troisième session du programme *Programmeur de jeux vidéo* car il a comme préalable le cours d’*Algèbre vectorielle appliquée* (201-J25-BB) et *Programmation 2* (420-J26-BB). Le cours de physique est en quelques sorte une mise en application des méthodes d’algèbres vectorielles. Les notions de physique sont à leur tour préalables au cours de d’*Intelligence d’artificielle* (420-J32-BB).

L’objectif de ce cours est de permettre à l’étudiant de s’approprier les notions de base d’un moteur de simulation physique utilisé dans les jeux vidéo. D’abord, l’étudiant devra se familiariser avec les notions de mécaniques classique afin de comprendre le mouvement d’un corps rigide. Ensuite, on présentera le fonctionnement général d’un moteur de physique ainsi que différents algorithmes et techniques utilisés dans les différentes phases de la simulation physique.

Tout au long du cours, l’étudiant sera amené à mettre en pratiques ses acquis théoriques et pratiques dans le cadre d’une implémentant un moteur simple de physique en deux dimensions (2D).

L’utilisation d’un moteur physique pour créer une mécanique de jeu ainsi que son interaction avec les autres modules du moteur de jeu sera davantage abordée dans le cadre du cours *Utilisation d’un engin de jeu vidéo 1* (420-J31-BB).

### Orientations pédagogiques

Le cours *Physique dans le jeu vidéo* est un cours qui présente la physique telle qu’elle est utilisée dans l’industrie du jeu vidéo. Il ne s’agit pas d’un cours de physique théorique où l’objectif serait de résoudre algébriquement des problèmes de physique, mais plutôt de présenter des techniques numériques permettant d’approximer à l’aide d’un ordinateur un mouvement physique. Ce cours ne vise pas à former un programmeur spécialisé en physique, mais bien un programmeur de jeu capable d’utiliser un module de physique afin de créer des mécaniques de jeu utilisant la physique. Par conséquent, on aura avantage à laisser de côté l’aspect purement théorique au profit d’une approche pratique basée sur l’implémentation de principes physiques dans un contexte de jeu.

## PARAMÈTRES PÉDAGOGIQUES

**Énoncé de compétences** : Appliquer des notions de mécanique classique et des méthodes numériques pour simuler des objets physiques en mouvement

**Cours :** 203-J28-BB

|  |  |  |  |
| --- | --- | --- | --- |
| **OBJECTIFS**  **TERMINAUX** | **ÉLÉMENTS DE COMPÉTENCE** | **BALISES DE CONTENU** | **CRITÈRES**  **D’ÉVALUATION** |
| 1. Résoudre des problèmes simples de cinématiques 2. Décrire les principes de la dynamique Newtonienne 3. Décrire les propriétés des corps rigides 4. Énoncer l’équation du mouvement d’un corps rigides |  | Cinématique 3D:   * Vitesse moyenne et instantanée * Accélération moyenne et instantanée * Vélocités linéaires & angulaires * Accélération linéaires & angulaires   Dynamique Newtonienne:   * Masse * Forces * Trois lois de Newton * Impulsions et quantité de mouvement * Moment de force   Corps rigides :   * Centre de masse * Moment d’inertie * Équations du mouvement * Contraintes entre corps rigides (Contact, rotule, pivot, glissière)   Corps déformables :   * Systèmes de contraintes * Système masse-ressort | *Évaluation formative*  *Évaluation sommative*  Définition précise des trois lois de Newton ;  Description juste des notions mathématiques reliées à la physique mécanique (vélocités linéaires et angulaires, masse, forces, impulsions, friction, etc.).  Description juste des concepts de *Rigid* *Body* et *Soft* *Body*.  Identification correcte des principaux types de contraintes physiques. |
| 1. Formulation mathématique de différents types de force 2. Implémentation adéquate du mouvement d’une masse ponctuelle soumise à plusieurs forces 3. Identification sommaire des enjeux reliés aux calculs numériques. |  | Types de forces :   * Force gravitationnelle * Force de friction * Force d’un ressort * Force d’amortissement visqueux * Champ de force   Implémentation d’un système de particules :   * Équation du mouvement d’une masse ponctuelle * Accumulation vectorielle de forces externes   Exemples d’utilisation : Attracteur gravitationnel, Pendule, Système masse-ressort    Enjeux de l’intégration numérique :   * Intervalles de temps * Convergence et stabilité * Précision numérique | *Évaluation formative*  *Évaluation sommative*  *Travaux individuels et travaux à réaliser en équipes*  Description et implémentation correcte des différentes forces  Implémentation efficace du mouvement d’une particule  Identification des principaux problèmes reliés à l’intégration numérique |
| 1. Reconnaître adéquatement les types de contacts 2. Implémenter une détection de collision 3. Implémenter une réponse à une collision en mouvement | 3. | * Contact statique vs collision * Friction statique vs dynamique * Collision élastique et inélastique * Restitution * Collision oblique * Détection de collision entre polygones convexes : Théorème des axes séparateurs (*SAT*) * Conservation de la quantité de mouvement * Utilisation de la normale et du point de contact d’une collision. | *Évaluation formative*  *Évaluation sommative*  *Travaux individuels et travaux à réaliser en équipes*  Décrire correctement le comportement d’objets physique en contact  Utilisation juste des algorithmes simples de détection de collisions entre primitives géométriques  Implémentation correcte d’une fonction de réponse à la suite d’une collision entre un objet et l’environnement. |
| 1. Décrire les différentes étapes de traitement d’un moteur de physique. 2. Utiliser adéquatement les différentes requêtes de collision | 4. | Présentation des différentes librairies utilisées par l’industrie du jeu vidéo sur le marché (*Box2D, Bullet*, *Havok*, *PhysX*, *ODE*, etc…)  Description du fonctionnement interne d’un moteur de simulation physique :   1. Intégration des vélocités 2. Détermination des paires d’objets 3. Phase d’élagage 4. Algorithmes d’élagage 5. Algorithmes de détection de collisions discrètes et continues (*CCD*) 6. Algorithmes de résolutions d’interpénétrations   Identification et utilisation adéquate des fonctionnalités d’un API physique :  - Lancer de rayons (*Raycasts*)  - Balayage de primitives (*Shapecasts*)  - Calcul de distance minimale  - Intersection de primitives | *Évaluation formative*  *Évaluation sommative*  *Travaux individuels et travaux à réaliser en équipes*  Description adéquate des différentes étapes de la simulation physique dans un jeu (intégration des vélocités, détection des collisions, résolutions des interpénétrations).  Classification juste des différents algorithmes d’élagage et de détection de collisions.  Identification précise des fonctionnalités physiques fournies par un moteur de jeu.  Utilisation adéquate des fonctionnalités de l’API physique afin de représenter un objet rigide en mouvement.  Utilisation des résultats de l’algorithme de détection des collisions dans le cadre du développement d’une mécanique de jeu. |

**ÉVALUATION FINALE**

##### Pondération : 30%

#### Formulation de la performance finale attendue de l’élève au terme de ce cours

À la fin de ce cours l'étudiant devra être capable d’expliquer le fonctionnement général d’un moteur de simulation physique et d’exploiter une librairie existante afin d’implémenter des mécaniques de jeu.

#### Forme de l’épreuve terminale

* Problèmes portant sur les concepts théoriques de la physique d’un objet en mouvement. (partie théorique)
* Modélisation sommaire d’un moteur de simulation physique dans son ensemble. (partie théorique)
* Réalisation de programmes simples en langage orienté objet dans l’environnement de développement intégré utilisant la physique pour résoudre un problème typique lié à la programmation d’un jeu vidéo 2D ou 3D (partie pratique)

#### Contexte de réalisation de l’épreuve terminale

* Individuellement dans un laboratoire d’informatique
* D'une durée minimale de 3 heures.
* Dans la partie pratique, les étudiants peuvent utiliser une documentation autorisée par le professeur.

Contexte de réalisation pour la partie pratique :

* + Individuellement.
  + À partir de spécifications établies par l’enseignant.
  + En utilisant l’environnement de développement et la librairie de simulation physique choisis pour le cours.

#### Critères d’évaluation propres à l’épreuve terminale

* + Respect des consignes.
  + Justesse des calculs. (partie théorique)
  + Utilisation appropriée des fonctionnalités d’une librairie de simulation physique. (partie pratique)
  + Fonctionnalité des programmes. (partie théorique)

**MÉDIAGRAPHIE DE BASE**

**Médiagraphie pour les élèves :**

* Game Physics Engine Development: How to Build a Robust Commercial-Grade Physics Engine for your Game, 2nd E., Ian Millington (2010), ISBN-10 : 0123819768, IBSN-13: 978-0123819765 (*facultatif*)

#### Médiagraphie pour les enseignants :

* Game Physics, 2nd Ed., David H. Eberly (2010), ISBN-10: 0123749034, ISBN-13: 978-0123749031
* Real-Time Collision Detection, 1st Ed, Christer Ericson (2004), ISBN-10 : 1558607323, IBSN-13: 978-1558607323
* Game Engine Architecture, 1st Ed., , Matt Whiting (2009), ISBN-10: 1568814135, IBSN-13: 978-1568814131