HMIN 317 – Moteur de Jeux

INTRODUCTION

Rémi Ronfard

Remi.ronfard@inria.fr

https://team.inria.fr/imagine/remi-ronfard/

Ce cours est très largement inspiré des cours de ***Marc Moulis et de Benoit Lange***.

Le but de cette présentation est de fournir une vue globale des dix séances de cours et TP et de présenter les choix d'études documentaires et projets.

Qu’est-ce qu’un moteur de jeu ?

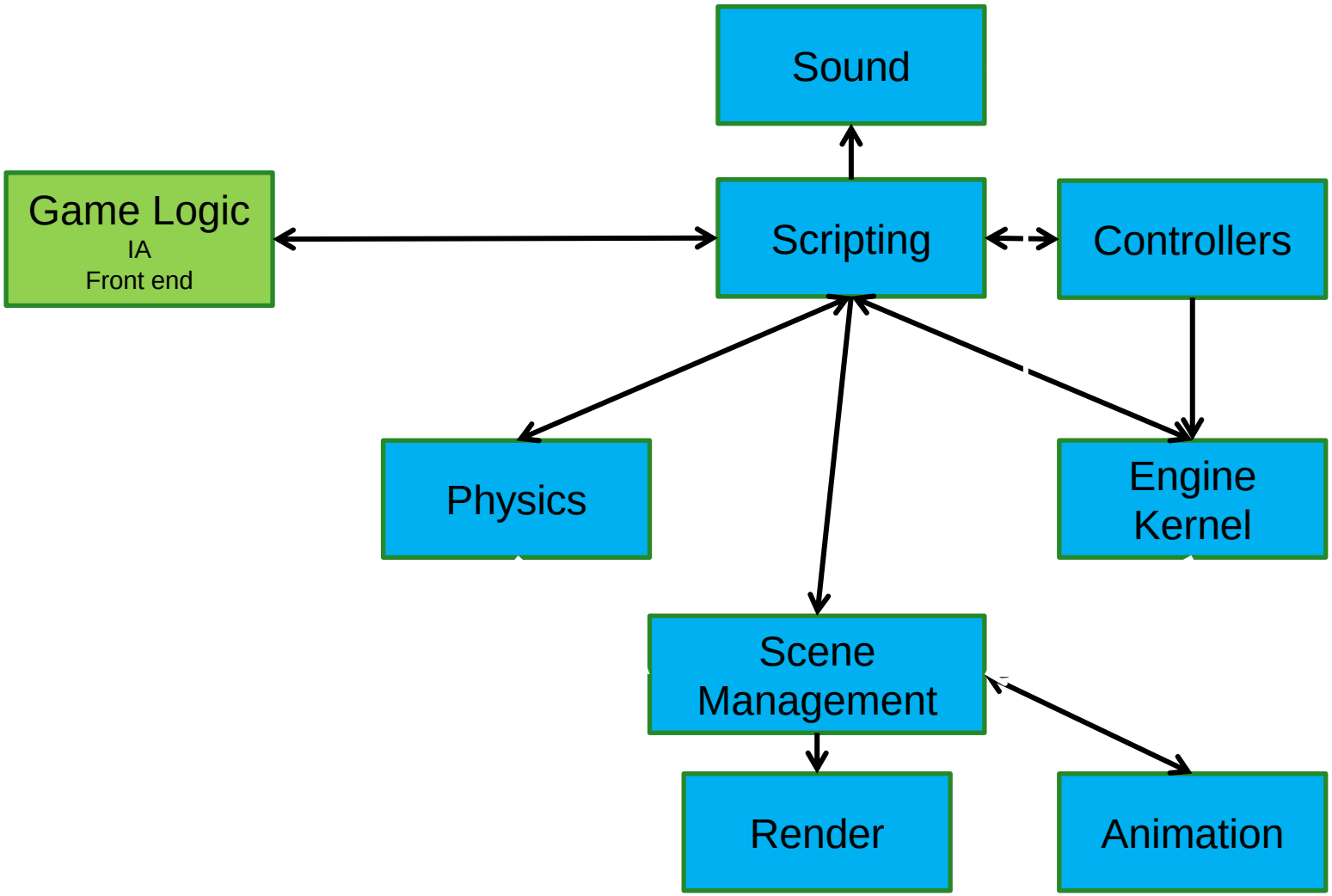
- On appelle "moteur de jeu" l'ensemble des composants logiciels qui fournissent tous les services nécessaires à l'évolution et l'affichage d'un univers interactif, à vocation ludique.
- On fera la distinction entre:
  - Moteurs first-party: le moteur est tout ou majoritairement développé en interne par le développeur du jeu
  - Moteurs third-party: le moteur est acquis auprès d'une société tierce qui l'a développé (Ex: Unreal Engine, Frostbite, Unity…)

- Moteurs dédiés: Les composants du moteur de jeu sont spécialisés pour un type de jeu précis: FPS, course, aventure, plateforme, RTS, etc...
- Moteurs généralistes: Les composants fournissent tous les services utiles pour la mise en œuvre de virtuellement n'importe quel type de jeu (c'est la tendance des moteurs third party : Renderware, Unreal Engine, Unity, Ogre, etc...)
- **NB:** *Dans le cadre du cours, nous nous intéresserons principalement à la mise en place d'un framework qui pourrait servir de base commune tant à un moteur généraliste que dédié. Les spécificités techniques de chaque catégorie de jeu ne seront donc pas ou peu abordées.*

Structure d'un moteur de jeu

- Notifications
- Gestion mémoire
- Game loop
- Gestion des controleurs
- Gestion des donnés disque
- Gestion du temps
- IA & Comportements
- Interactions avec la scène
- Gestion du son
- Gestion du front end
- Gameplay

Structure d'un moteur de jeu



Processus de développement d'un jeu vidéo

Avant d'étudier en détail les composants d'un moteur de jeu, il est important de se familiariser avec le processus de développement d'un jeu.

Selon l'ampleur et le type de projet, tous les corps de métiers ci-après peuvent ne pas être représentés, ou certaines personnes peuvent endosser plusieurs casquettes

- Artistique
  - Directeur artistique
  - Concept artist
  - Graphiste (2D, 3D)
  - Graphiste technique
  - Animateur
  - Designer sonore
  - Musicien
- Technique
  - Directeur technique
  - Développeur
- Production
  - Producteur
  - Game designer, Réalisateur
  - Level designer
  - Opérateur mocap
  - Testeur
  - Chargé de production

Workflow : étapes de la création d'un jeu

- Brainstorming
  - Propositions de concepts
  - Feu vert
- Pré-production
  - Ecriture du game design document (GDD)
  - Ecriture du technical design document (TDD)
  - Recherches artistiques
  - Mise en place de la chaîne d'outils (export, éditeurs, ...)

Workflow

- Production
  - Ecriture du jeu
  - Création des données (assets)
  - Mise en place du gameplay
  - Test panels
  - Polishing
- Testing & validation
  - Tests & débuggage
  - Soumission pour les TRC
  - Production du gold master

- Post-production
  - Action commerciale
  - Contenus additionels
  - Ni versions, ni débugs !
- Post-Mortem
  - Retours sur expérience

Plan du cours

- Cours 1: Introduction et API OpenGL
- Cours 2: Structure d'un moteur de jeu
- Cours 3: Programmation temps réel
- Cours 4: Gameplay (UBISOFT)
- Cours 5: Mathématiques du jeu vidéo
- Cours 6: Gestion de scène
- Cours 7: Physique du jeu vidéo (UBISOFT)
- Cours 8: Animation temps réel
- Cours 9: Intelligence artificielle pour le jeu
- Cours 10: Présentation des projets

Cours 1 : Introduction

Les outils de ce cours:

- API Rendu OpenGL
- Gestion de version Git
- Programmation Qt

Cours 2 : Structure d'un moteur de jeu

- Game loop
- Game objects

Cours 3 : Programmation temps réel

- Gestion du temps réel
- Interactions utilisateur
- Controleurs de jeu
- Workflow

Cours 4: Gameplay

- Fiction interactive
- Jeux de plateau
- Jeux d'aventures
- First-person shooters
- Third-person shooters

Cours 5 : Mathématiques pour le jeu vidéo

- Vecteurs
- Matrices
- Quaternions
- Géométrie 3D
- Distances
- Collisions
- Chemins

Cours 6 : Gestion de scène

- Modélisation surfacique
- Modélisation voxellique
- Mise à jour
- Performance
- Niveaux de détails

Cours 7: Physique du jeu vidéo

- Cinématique
- Dynamique
- Physique du solide
- Systèmes de particules
- Simulation

Cours 8: Animation temps-réel

- Animation de personnages
- Rigging, skinning
- Animation faciale
- Blendshapes
- Capture de mouvement
- Principes d’animation

Cours 9: Intelligence Artificielle pour le jeu vidéo

- Path-finding
- Machines d'états finis
- Arbres de comportements
- Personnages non-joueurs (NPC)
- Apprentissage
- Etude de cas: IA des Sims

TP1: PROGRAMMATION QT ET OPENGL

- Prise en main des outils Qt, Git et OpenGL
- Gestion des événements
- Rendu d'une scène 3D

TP2: TIMERS

- Gestion du temps
- Applications multi-fenetres

TP3: GESTIONNAIRE DE RESSOURCES

- Optimisation
- Parallelisme
- Calcul intensif

TP4: GAMEPLAY

- Etudes de cas avec UBISOFT

TP5: TEXTURES ET GPU

- Programmation GPU
- Shaders
- Textures

TP6: GESTION DE SCENES

- Octrees
- Niveaux de détails
- Imposteurs
- Physique des sphères

TP7: PHYSIQUE

- Etudes de cas avec UBISOFT

TP8: ETUDES DOCUMENTAIRES

- Présentation des études documentaires
  - 10 minutes par binome
  - 5 minutes de questions /réponses

TP9: FINALISATION MINI-PROJETS

- Répétition des démos
- Rédaction des docs techniques
- Test et debug

TP10: PRESENTATION MINI-PROJETS

- Présentation des projets
  - 20 minutes par binome
    - 10 minutes de présentation
    - 5 minutes de démo
    - 5 minutes de questions /réponses

ETUDES DOCUMENTAIRES

- Former des binômes, et choisir un des sujets de la liste
- Le sujet fera l'objet d'une présentation orale (avec slides projetés) d'une dizaine de minutes par TOUS les membres du groupe devant l'ensemble des étudiants
- L'examen final porte sur ces sujets (au choix)

1.GENERATION PROCEDURALE DE TERRAINS

- On s'intéressera aux différentes possibilités de génération et affichage en temps réel d'un terrain 3D virtuel:
  - Méthodes existantes (ROAM, height maps, fractales, …), en détaillant leurs avantages et leurs limites
  - Problématiques d'affichage (LOD, culling, occlusions, …)
  - Contraintes d'application des textures (problèmes de projection parallèle, …)

2.CALCUL DES OMBRES

- La présentation dressera un inventaire des différentes méthodes de génération temps réel des ombres portées:
  - Algorithmes existants (textures projetées, shadow volumes, shadow mapping, …)
  - Avantages et limitations de chacun de ces algorithmes

3.METHODES D'ECLAIRAGE

- L'étude présentera une comparaison, en indiquant clairement le fonctionnement et les limites des méthodes de calcul d'éclairage suivantes, dans le cadre d'une utilisation pour les jeux vidéos:
  - Light maps
  - Photon maps
- L'exposé devra présenter des algorithmes détaillés.

4.CINEMATOGRAPHIE ET EFFETS SPECIAUX

- La présentation devra présenter au moins une méthode de mise en œuvre (algorithme détaillé) pour CHACUN des effets spéciaux temps réel suivants, que l'on trouve actuellement dans la plupart des jeux modernes:
  - Profondeur de champ (Depth Of Field)
  - God rays (light scattering)
  - Glows (blooming)

5.CINEMATIQUE INVERSE

- La présentation dressera un inventaire des différentes méthodes de cinématique inverse utilisés dans le jeu vidéo
  - Algorithmes existants (Jacobien, CCD, FABRIK, Triangulation,…)
  - Exemples de jeux mettant en œuvre ces méthodes
  - Avantages et limitations

6.CINEMATIQUES

- La présentation dressera un inventaire des outils proposés par les moteurs de jeux pour la réalisation de cinématiques
  - Exemples de jeux mettant en œuvre ces outils
  - Evolution et perspectives
  - Avantages et limitations

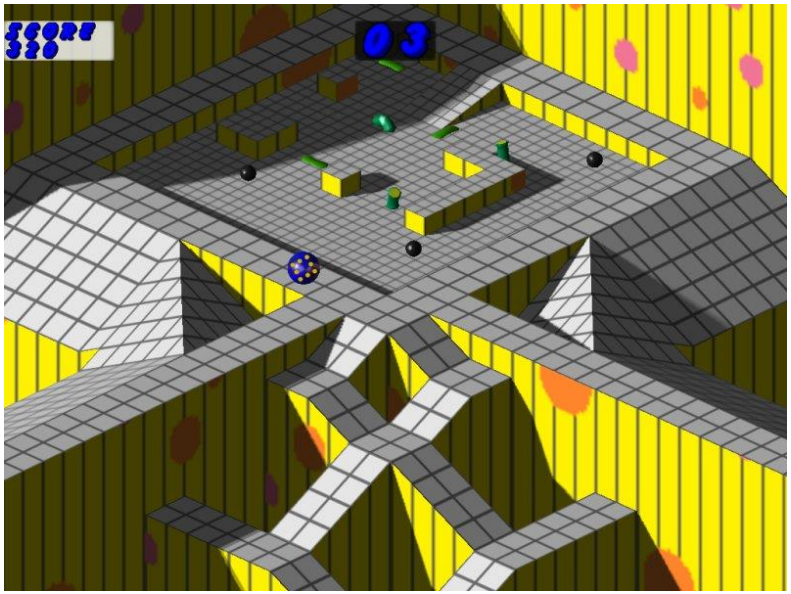
MINI PROJETS DE PROGRAMMATION

- Former des binômes, et choisir un sujet
- Le sujet fera l'objet d'une présentation orale :
  - 15 minutes de présentation (avec Slides)
  - 5 minutes de démonstration
  - 10 minutes de questions et réponses
- Travail à rendre :
  - Le code source (compilable)
  - Une documentation de quelques pages
  - La présentation

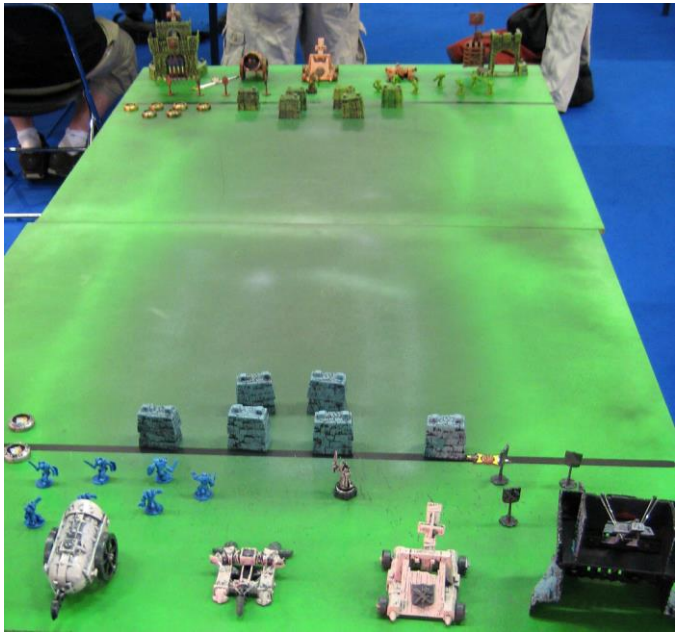
1. MARBLE MADNESS

Le projet devra recréer un niveau jouable 3D inspiré du jeu « Marble Madness ».

Une grande attention sera portée au gameplay.



2. JEU DE STRATEGIE



Le projet devra recréer une démo jouable d'un jeu de stratégie orienté combat.

Dans ce projet, deux équipes s'affronteront : une piloté par le joueur et une piloté par une IA.

Chaque unité devra posséder une conscience propre.

La gestion des déplacements sera effectuée par des algorithmes de recherche de type A*.

Une grande attention devra être porté sur la partie IA. Le rendu pourra être effectué en 2D.

3. MINECRAFT

Le projet devra recréer un niveau jouable 3D inspiré du jeu « Minecraft ».

La carte devra être générée aléatoirement et posséder plusieurs écosystèmes.

Un certain nombre d'actions devront être possibles par le joueur (creuser, construire)

Il sera nécessaire d'animer les NPC.

Une grande attention sera portée à la gestion de la scène.

4. SIMULATEUR DE VOL

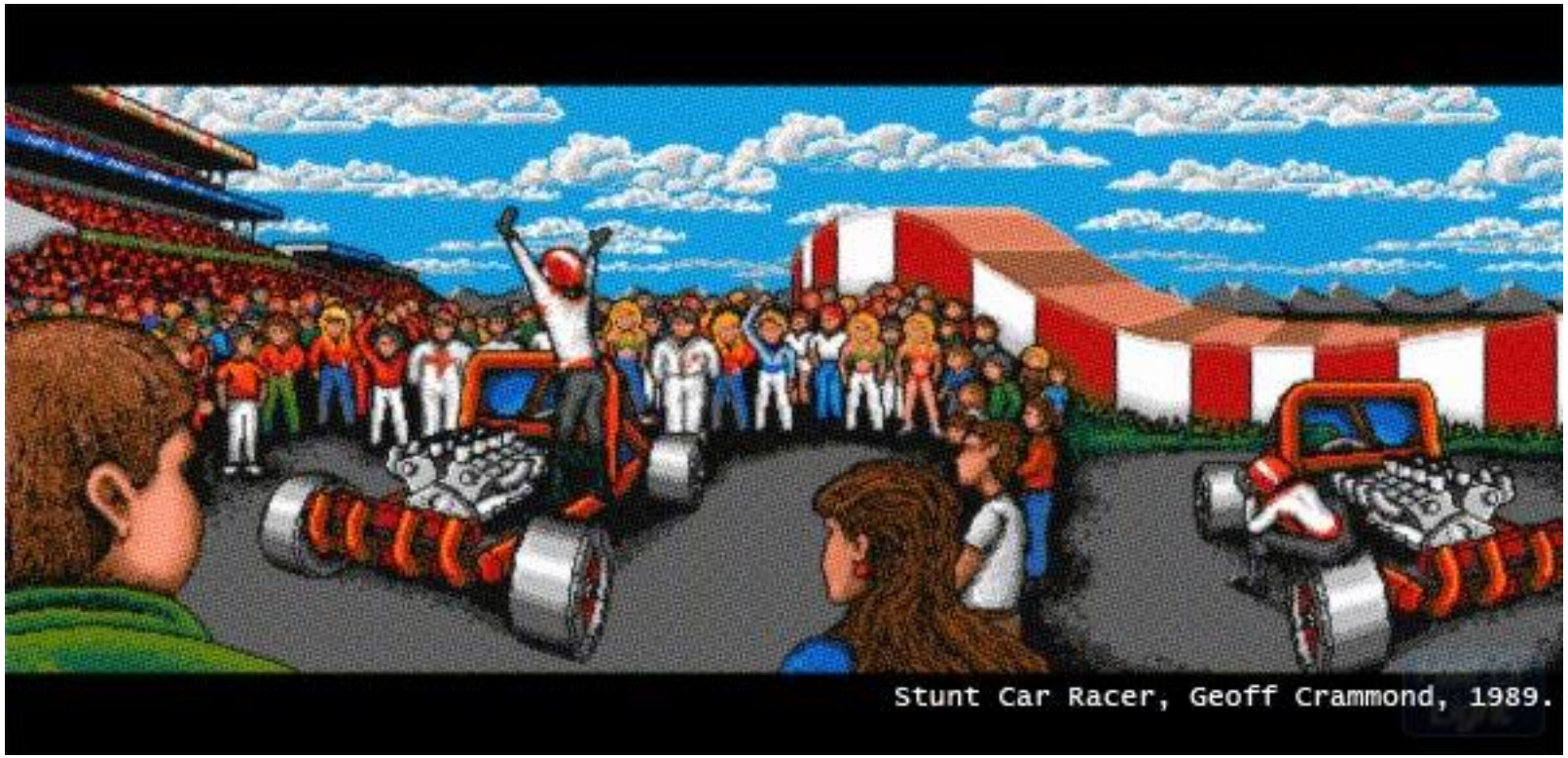
Réaliser un simulateur de vol en avion dans un univers infini.

Il faudra proposer des méthodes intuitives pour le contrôle des vitesses et des accélération dans les trois directions.

Il faudra gérer simultanément les défilement du décor, et les mises à jour du tableau de bord.

5.COURSE DE VOITURES

Réaliser un jeu de course de voiture en caméra subjective, dans le style du jeu « stunt car racer », en gérant les déplacements, vitesses et accélérations du véhicules. Une attention particulière devra être apportée à la gestion des collisions entre le véhicule et le décor



6.JEU DE PLATEFORME

Réaliser un jeu de plateforme dans le style de Super Mario 64 ou Galaxy.

Il faudra en particulier mettre en œuvre les animations du personnage principal et leur contrôle par le joueur.