

GMIN 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.

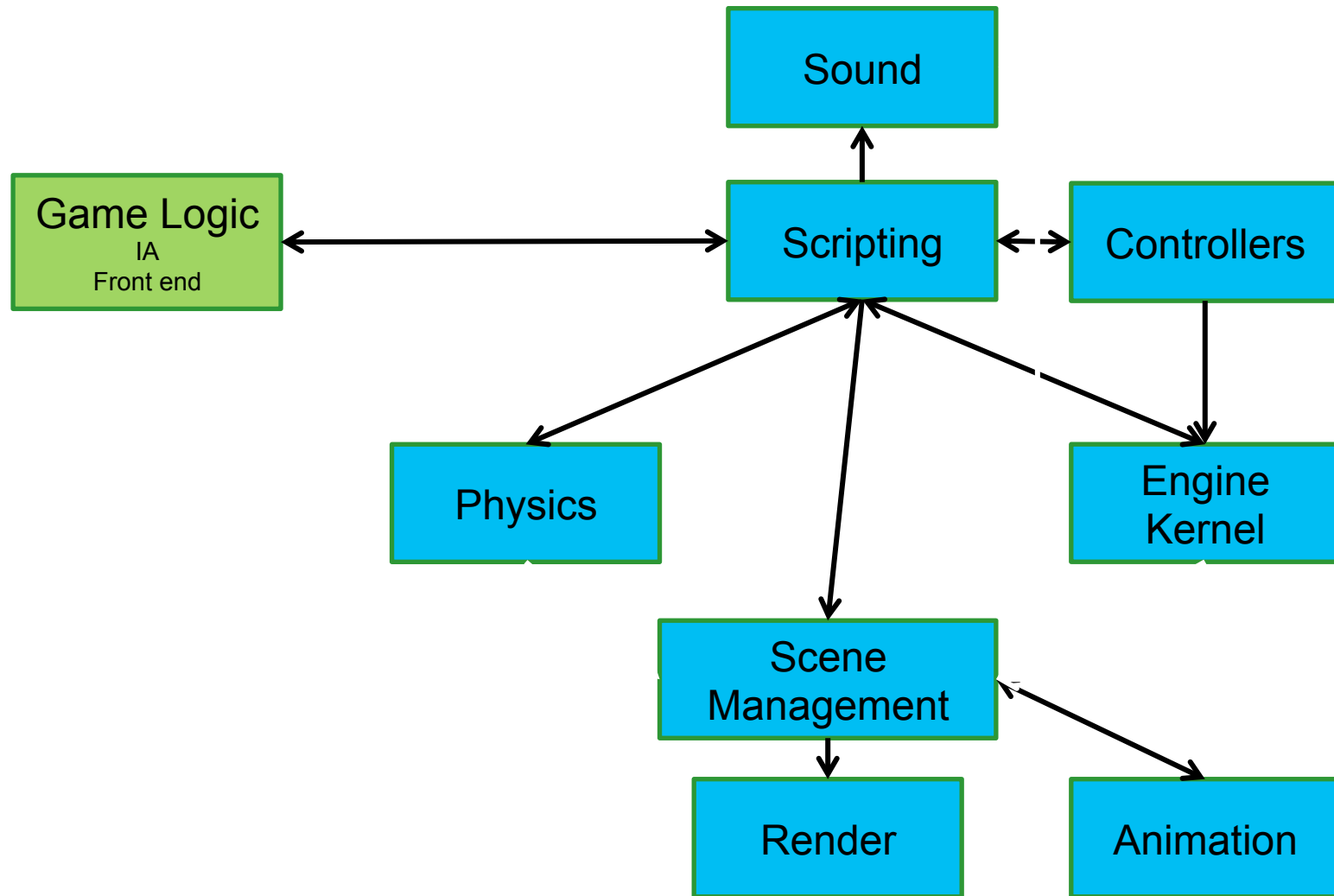
Qu'est-ce qu'un moteur de jeu ?

- 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 de cette présentation, 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



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 débbugs !
- Post-Mortem
 - Retours sur expérience

Plan des cours

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

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
- Contrôleurs de jeu
- Workflow

Cours 4 : Mathématiques pour le jeu vidéo

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

Cours 5 : Rendu temps-réel

- Vertex shaders
- Pixel shaders
- Geometry shaders
- Effets Spéciaux

Cours 6: Gameplay

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

Cours 7 : Gestion de scène

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

Cours 8 : Physique du jeu vidéo

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

Cours 9 : Animation temps-réel

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

Cours 10 : Intelligence Artificielle pour le jeu vidéo

- Path-finding
- Machines d'états finis
- Arbres de comportements
- Personnages non-joueurs (NPC)
- Sims AI
- Apprentissage

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-fenêtres

TP3: GESTIONNAIRE DE RESSOURCES

- Parallelisme
- Calcul intensif
- Programmation client/serveur

TP4: GESTIONNAIRE DE RESSOURCES (SUITE)

- Parallelisme
- Calcul intensif
- Programmation client/serveur

TP5: TEXTURES ET GPU

- Programmation GPU
- Shaders
- Textures
- Transformations

TP6: GAMEPLAY

- Etudes de cas avec Ubisoft

TP7: GESTION DE SCENES

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

TP8: PHYSIQUE

- Etudes de cas avec Ubisoft

TP9: ETUDES DOCUMENTAIRES

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

TP10: RAPPORT D'ETAPE MINI-PROJETS

- Conception (design) et plan de travail des mini-projets
 - 10 minutes par binome
 - 5 minutes de questions /réponses

ETUDES DOCUMENTAIRES

- Former des binômes, et choisir un des sujets de la liste
- Chaque sujet ne peut être traité que par UN SEUL binôme
- 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

1.GENERATION 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. Génération procédurale de villes

- La présentation s'intéressera aux différentes méthodes qui permettent de générer l'ensemble des éléments d'une ville virtuelle, à savoir:
 - Le découpage parcellaire en fonction de la topologie du site et de divers paramètres (obstacles naturels, typologie de la cité, ...)
 - Les bâtiments (emprise au sol, géométrie, textures)
 - Les infrastructures de communication (routes, voies ferrées, ...)
- La présentation conclura sur l'intérêt et les limites de l'utilisation de telles technologies dans le contexte du jeu vidéo

3.GENERATION DE PLANTES

- L'étude doit dresser un état des lieux des techniques existantes de génération procédurale de plantes (réalistes ou non). On s'attachera notamment à détailler:
 - Le fonctionnement des différents algorithmes (L-systems, fractales, surfaces de subdivision, ...)
 - Les méthodes de génération des textures associées (feuillage, écorce, éclairage, ...)
 - Les méthodes de simplification possibles des modèles polygonaux
- L'étude conclura sur les technologies qui semblent le plus adaptées dans le cas de l'utilisation dans un jeu vidéo moderne.

4.SIMULATION DE TISSUS

- La présentation s'intéressera aux algorithmes utilisables en temps réel pour représenter des tissus
 - Organisation de la structure de données
 - Contraintes et résultats
- La présentation devra clairement identifier les technologies utilisables dans le cadre d'une utilisation dans les jeux vidéo

5.GENERATION PROCEDURALE DE CIEL

- Les étudiants présenteront les différentes techniques qui peuvent être utilisées pour générer procéduralement un ciel:
 - Génération des textures
 - Eclairage
 - Animation

6.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

7.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)

8.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.

9.High Dynamic Range

- L'étude devra décrire le fonctionnement et les cas d'utilisation d'images de type HDR dans le domaine du jeu vidéo:
 - Effets spéciaux
 - Eclairage
 - Tone mapping
- Les étudiants présenteront 1 ou 2 algorithmes mettant en avant des images HDR.

10.SIMULATION DE FOULE

- L'étude devra détailler les méthodes de fonctionnement d'une simulation de foule:
 - Flocking
 - Approches multi-échelles
 - Agents
- La présentation conclura sur les usages possibles et les limitations dans le contexte du jeu vidéo

11.CINEMATIQUE INVERSE

- L'étude devra présenter les divers algorithmes utilisés pour calculer une cinématique inverse:
 - Utilisation dans les jeux
 - Jacobiens, méthodes incrémentales,...
 - Algorithmes

12. GRAPHES DE MOUVEMENTS

- L'étude devra présenter les méthodes de « motion graphs » et leur utilisation dans les jeux
 - Capture de mouvements
 - Organisation temporelle
 - Organisation spatiale
 - Algorithmes offline
 - Algorithmes temps réels

MINI PROJETS DE PROGRAMMATION

- Former des binômes, et choisir un des sujets de la liste
- Un effort devra être fait sur la compatibilité (Linux, Windows, MacOS)
- 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.GENERATEUR DECHÂTEAU FORT

Le projet s'attachera à la génération procédurale de modèles 3D représentant des châteaux-forts, en se basant sur une étude préliminaire des caractéristiques architecturales de telles bâtisses.

Le projet devra implémenter:

- La génération aléatoire du modèle 3D en fonction de divers paramètres (à définir suite à l'étude architecturale)
- La visualisation du modèle 3D
- Bonus :
 - Texturer le château
 - Générer les textures
 - Animer le Château



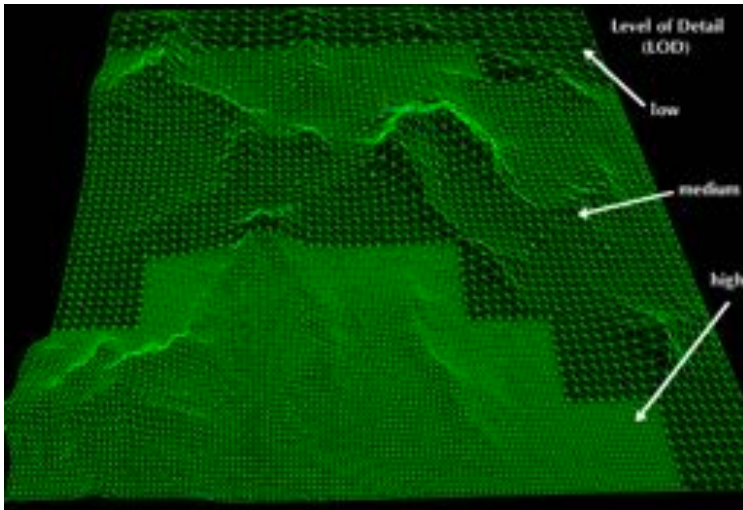
2.GENERATEUR DE TERRAIN



On souhaite modéliser automatiquement un environnement extérieur réaliste qui pourrait servir de base pour le décor d'un jeu.

Le projet devra implémenter:

- La génération (aléatoire selon paramètres) du modèle 3D de terrain à partir d'une méthode de génération choisie
- La visualisation, navigation et la mise en place d'une stratégie de subdivision spatiale
- Bonus :
 - Affichage de textures
 - Effets spéciaux : nuages, eau, météo
 - Génération d'une flore



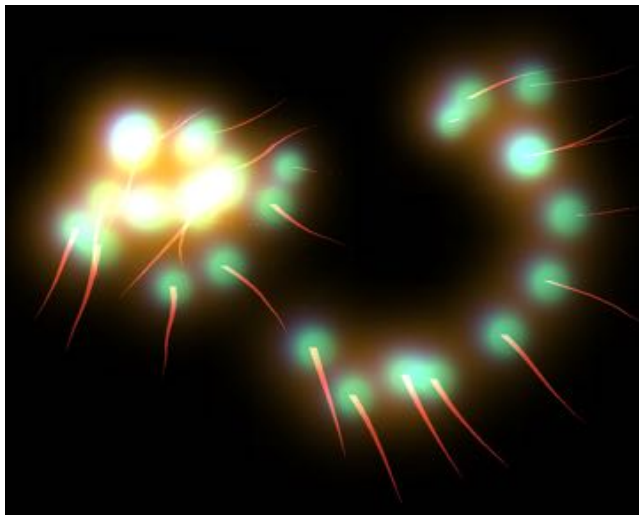
3.SIMULATEUR DE VOL D'OISEAU



On souhaite visualiser le comportement de groupe d'un ensemble d'acteurs soumis à différents stimuli (à définir par les étudiants) dans un environnement contraint (obstacles, topologie, etc...)

Le mini-projet devra implémenter:

- La visualisation 2D ou 3D des acteurs et de l'environnement
- L'animation des stimuli et des acteurs dans l'environnement
- Bonus :
 - édition interactive des stimuli
 - Modélisation de plusieurs comportements



4. FEU D'ARTIFICE



Le projet s'attachera à la simulation/visualisation réaliste en temps réel d'un feu d'artifice.

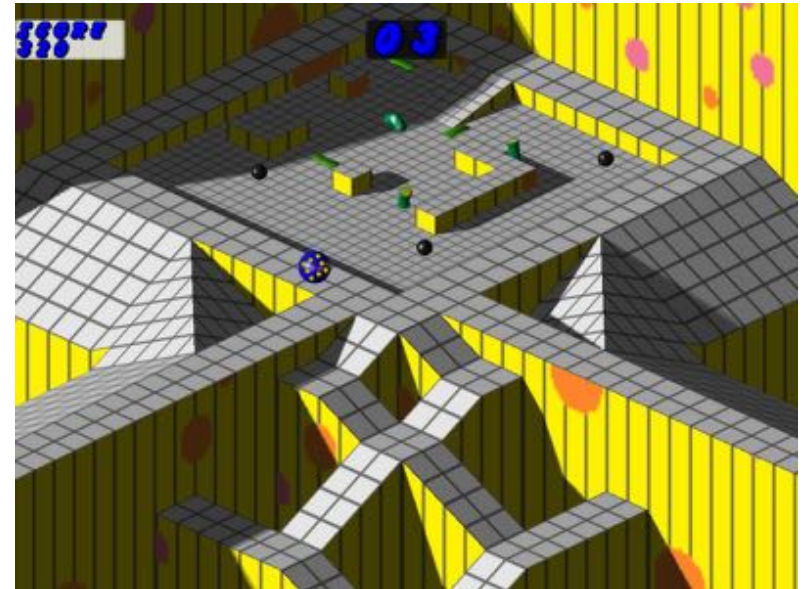
L'implémentation devra comprendre:

- Le rendu d'un script prédéfini (environ 10 secondes)
- Au moins 3 types différents de comportement réaliste (physique) des fusées
- Un système simple de script
- Bonus:
 - Générer les effets sonores
 - Adapter l'éclairement de la scène en fonction des bombes

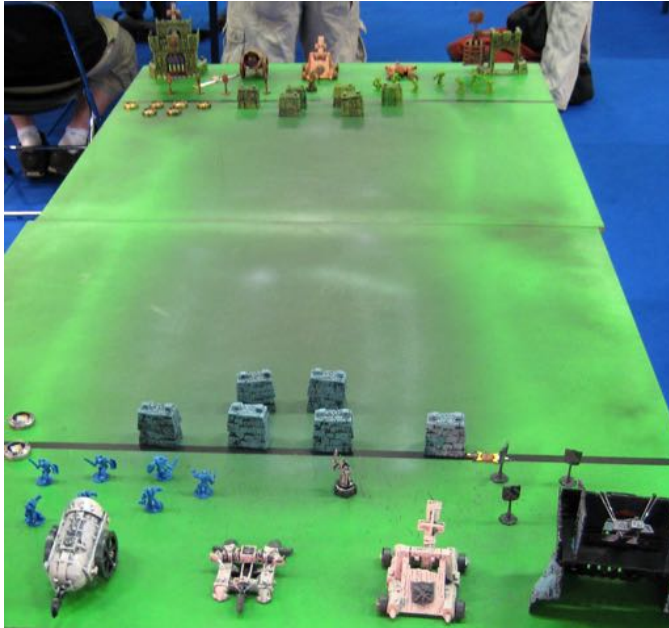
5. MARBLE MADNESS

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

Une grande attention sera portée au gameplay.



6. 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 pilotée par le joueur et une pilotée 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 portée sur la partie IA. Le rendu pourra être effectué en 2D.

7. 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.



8. 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.



9.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



10. 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.



CONCLUSION

- Et maintenant