

PROJET INFO3B

2022/2023

**Synthèse d'Images 3D & Animation
avec la librairie Three.js.**

**ABIDAT Jawad L2 IE3-1A
KUT KEMAL L2 IE3-3B**

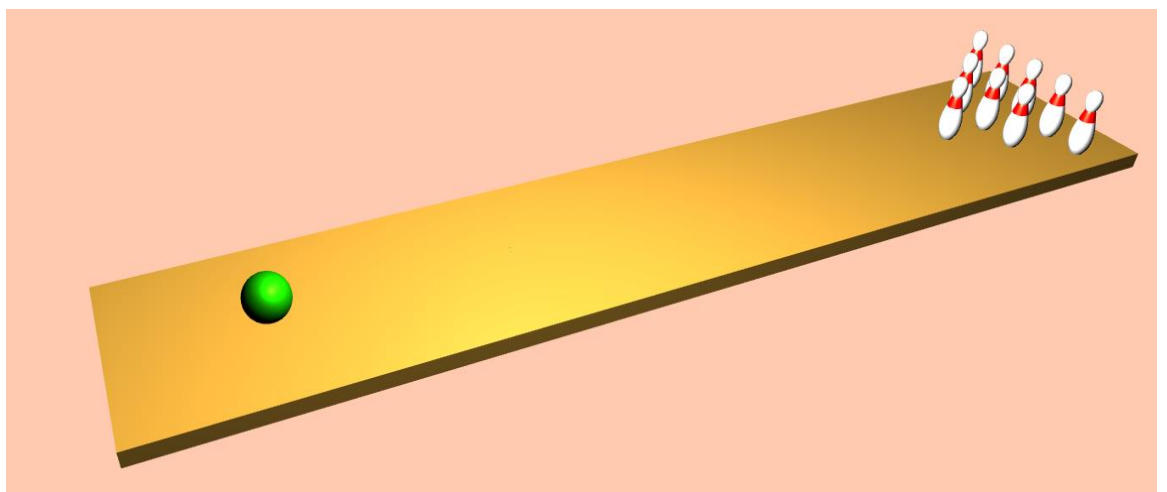
INTRODUCTION

Modélisation et animation d'une partie de bowling

Pour ce projet, nous devons réaliser 2 mènes par équipe d'une partie de bowling. Une mène est composée au plus de 2 lancers pour faire tomber toutes les quilles. Ce projet a pour but de mettre en œuvre nos connaissances en réalisant une partie de bowling.

Pour le bon déroulement de la partie nous avons besoin de deux équipes : équipe rouge et équipe bleue.

La partie se déroule sur une piste divisée en 2 parties, la partie qui contient la/les boule(s) et une deuxième partie qui se trouve à l'autre extrémité qui contient les quilles.



Capture d'écran de notre projet



INTRODUCTION

Nous avons pour but de faire un bowling en THREE JS sans que ce dernier ne ressemble à une simulation de la réalité, plusieurs éléments ne sont donc pas pris en compte dans cet exercice, tel que certaines forces, tel que la gravité, les frottements et les forces de contacts.

Voici les « objets » et les actions à modéliser/programmer pour le bon déroulement du projet :

- Une piste de bowling
- Des boules de bowling aux couleurs des équipes
- 10 quilles à disposer dans la piste en forme de « triangle »
- Des courbes de trajectoire pour les boules
- Des déplacements rectilignes (= les lancers) et les collisions boule/quilles
- L'affichage des scores

DIFFICULTÉ PRINCIPALE DU PROJET :

**Code trop lourd et complexe.
Alléger/faciliter le code, utilisation
de la POO – Programmation
Orientée Objet pour la
modélisation de certains objets.**

Construction des objets :

La piste

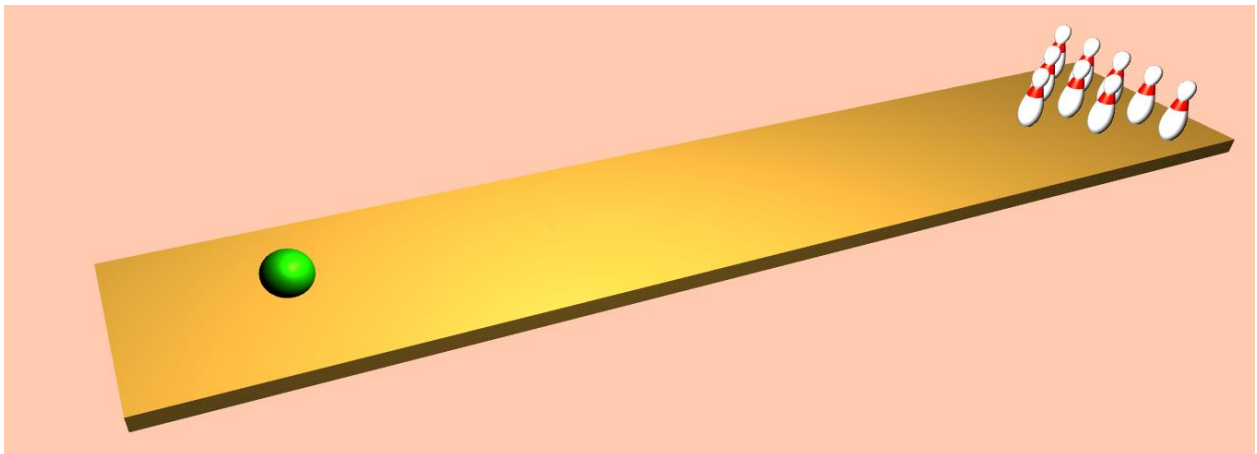
Dans notre jeu, il est nécessaire de créer une piste qui sera la base de notre jeu. La classe « Alley » contient un constructeur par défaut et une fonction « makeAlley » qui prend en compte toutes les constantes pour faire la piste.

Pour cela nous avons utilisé un parallélépipède de longueur 200, de largeur 40 et d'épaisseur/hauteur 3 puis nous lui avons attribué la couleur '0xAF822E'.



Nous avons utilisé le matériau MeshPhongMaterial pour bien avoir la 'brillance' d'une piste de bowling.

Notre Mesh : const main = new THREE.Mesh(mainGeometry, mainMaterial);



La boule

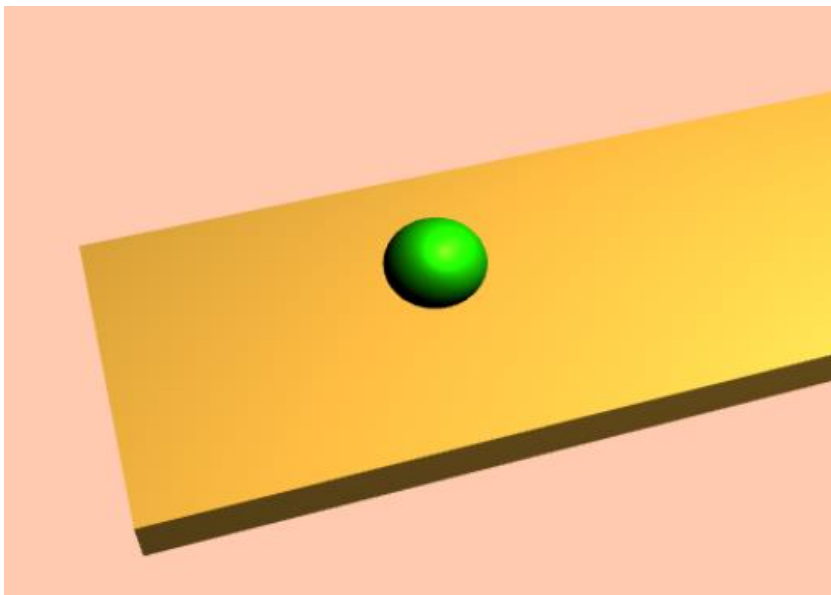
Dans notre jeu, il est nécessaire de créer une boule qui sera la base de notre jeu. La classe « Alley » contient un constructeur par défaut et une fonction « makeAlley » qui prend en compte toutes les constantes pour faire la piste.

La boule contient une courbe (cf tp ex balle de tennis). La boule est de la même couleur que l'équipe qui joue la manche. La courbe dans la boule est de couleur 'opposée' à celle de l'équipe qui joue la manche.

Nous avons eu un soucis de dernière minute dans la compilation du script et nous avons remplacé notre code de la boule contenant la courbe par une simple sphère faite avec la méthode `SphereGeometry()`.

Mais nous allons quand même vous expliquer la modélisation d'une boule qui contient une courbe :

Nous affectons à l'enveloppe de la sphère un certain nombre de points, grâce à une projection de ces derniers on obtiendra la création d'une courbe de Bézier.



Les quilles

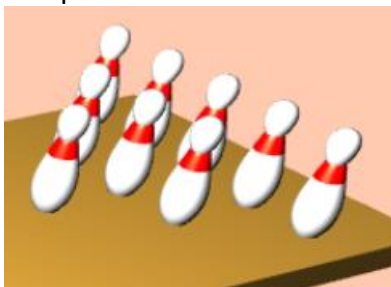
Pour modéliser une quille nous devons utiliser au moins 2 lathes. Nous avons d'abord créé 3 courbes de Bézier avec ensuite un calcul qui permet grâce à 3 points qui seront alignés peu importe le changement de taille/angle de la courbe de Bézier, et ceci à 2 reprises.



Quille de Bowling du projet

Pour la disposition des quilles sur la piste nous avons eu des soucis pour positionner chaque quille à son emplacement. Nous avons trouvé la solution de stocker les coordonnées de chaque quille dans un tableau :

Nous avons disposé les 10 quilles présentes sur la piste dans un tableau qui comporte dans chaque case les coordonnées (x,y,z) de chaque quille.



Collision boule / quille

Pour la collision entre la boule et les quilles nous avons implanté des méthodes qui considèrent les corps comme des parallélogrammes rectangles.

Comme énoncé précédemment on utilisera pour les quilles un tableau qui nous permettra à tout instant de savoir si la boule à l'instant t est de passage sur les coordonnées d'une quille, ce qui annoncera la collision entre la boule et une voir plusieurs quilles.

Animation

Le code se limite à un envoi de la balle répétitive sur la piste de manière rectiligne. Avec la prise en compte des ombre émise grâce à la lumière diffusée sur la scène

Point d'équipe

Si l'équipe arrive à renverser toutes les quilles au premier lancer, l'équipe marque 30 points.

Sinon l'équipe marque 15 points si toutes les quilles sont renversées au second lancer. Pour finir, si le l'équipe n'arrive pas à renverser toutes les quilles durant les deux lancers, alors, les points marqués par l'équipe correspondent au nombre de quilles renversées.

Le menu GUI

Pour le menu GUI nous n'avons pu intégrer que le changement d'angle de lumière afin de mettre plus ou moins de lumière, et aussi l'activité de la caméra afin de dynamiser la simulation et facilité l'utilisateur.

Malheureusement certaines fonctionnalités n'ont pas pu être intégré au menu, tel que la variation de la courbe de lancement (angle, taille...)

Conclusion

Notre jeu ne fonctionne malheureusement pas les difficultés furent nombreuses même si certaines ont pu être confrontées. La création des objets et la dynamisation nous a pris plus de temps que prévu malheureusement et peut-être que d'éventuelles méthodes moins énergivores et plus simples existent et auraient pu nous faciliter la tâche. Le sujet est en tout cas très intéressant, et amusant à exercer.

CITE ET DOCUMENT UTILISÉS

GOOGLE.COM

YOUTUBE.COM

GITHUB :

- **ARCHIVE D'ANCIEN ETUDIANT DE L2 A L'UNIVERSITE DE BOURGOGNE**
- **ARCHIVE CREATION DE BOWLING (SIMULATION)**