

Vendredi 2 décembre

# Rapport de projet Info3B

## Table des matières

1. Introduction.....	3
2. Comment jouer.....	3
3. Construction des objets.....	4
1.Construction de la piste.....	4
2. Construction du décor .....	4
3. Construction de la boule .....	5
4. Construction des quilles .....	5
4. Construction de la scène et mise en place .....	7
5. Animation .....	7
6. Collision et score .....	8
7.Menu GUI .....	8
8. Conclusion .....	9
9. Bibliographie.....	10

# 1. Introduction

Lors du troisième semestre, nous avons eu un projet en info3B qui consiste à simuler un bowling en utilisant le langage JavaScript et la bibliothèque Three.js qui lui est spécifique. Elle permet de créer des scènes 3D dans un navigateur web.

Le projet nous a été confié par Mr Lionel Garnier, professeur du module de Synthèse d'image. Nous venons donc avec ce rapport vous présenter notre projet.

## 2. Comment jouer

Notre jeu, est un bowling plutôt classique. Lors du lancement dans le menu GUI vous pourrez choisir la couleur de votre équipe dans les menus « Equipe ». Ensuite le jeu démarre. Pour cela le menu « Paramètres de lancer » vous permettra de modifier votre lancer rectiligne ou vos lancers qui suivent une courbe de Bézier.

Sur le côté il y a un tableau qui permet de compter le score à chaque manche pour chaque équipe. Vous pourrez suivre en temps réel le score de votre équipe ! Pas besoin de descendre en bas de la page pour le voir le tableau des scores est discret sur le côté gauche.

Vous pouvez à tout moment changer les couleurs de votre équipe si elle ne vous plaît pas, cela modifiera aussi la couleur des cellules du tableau.

Tableau des scores

	Manche 1		Manche 2	
Equipe 1				
Equipe 2				

*Figure 1 Tableau des scores*

### 3. Construction des objets

#### 1. Construction de la piste

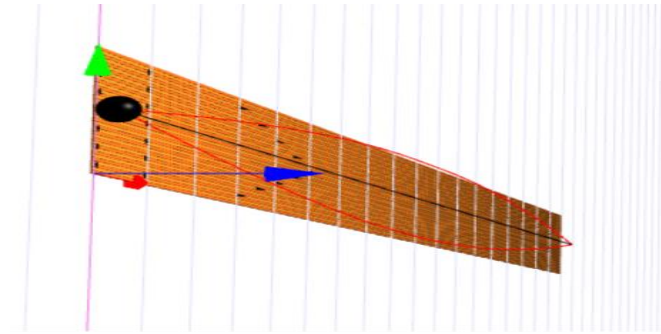


Figure 2 Piste de bowling

La piste a été créée avec des BoxGeometry qu'on est venu placer les uns à côté des autres. On a procédé ainsi car une piste de bowling est composée de plusieurs lattes (49), nous avons donc voulu reproduire à l'identique la piste de bowling. La longueur de la piste est faite à l'échelle d'une vraie piste.

Pour une meilleure immersion nous avons rajouté la ligne de début et de hors-jeu ainsi que les triangles pour la direction des tirs.

Les triangles ont été fait avec des points que l'on est venu ajouter pour former un triangle avec la fonction THREE.Triangle.

Les cercles ont été créés avec des CircleGeometry qu'on est venu poser sur la piste.

#### 2. Construction du décor

Nous avons fait un décor pour une meilleure immersion dans le jeu. Pour commencer on a rajouté des gouttières sur le côté de la piste

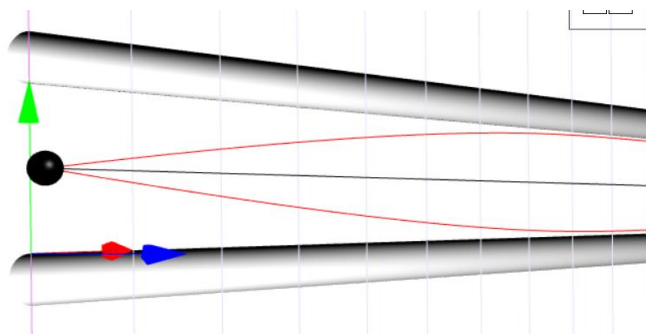


Figure 3 Gouttières

Elles ont été faites avec des CylinderGeometry c'est un choix que nous avons fait pour que cela soit plus simple à réaliser, nous avons donc créé le cylindre et on l'a coupé en deux lors de la création dans ses paramètres. On l'a cloné pour éviter de recréer l'objet. Ensuite on

est venu ajouter de l'ombre pour donner une profondeur a l'objet, il suffit juste de jouer avec la lumière et de permettre au matériel de la recevoir.

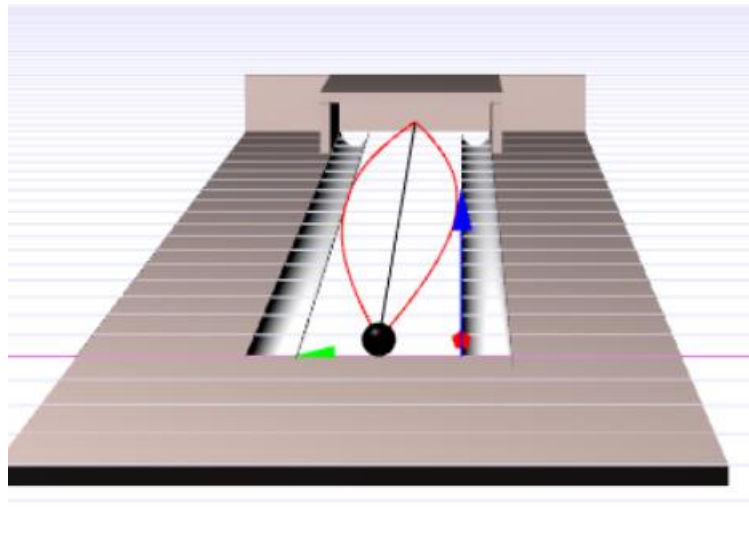


Figure 4 Décor du jeu

Nous avons créé un petit décor avec des BoxGeometry pour donner un coté réaliste. Un sol et un fond avec une boîte pour les quilles.

### 3. Construction de la boule

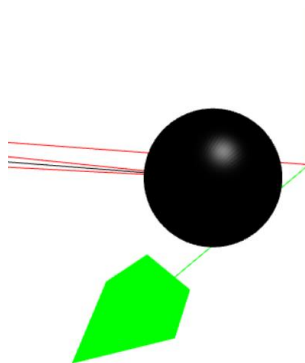


Figure 5 Boule

Nous avons créé la boule de bowling avec une SphereGeometry.

La boule de bowling est soumise à plusieurs contraintes dans le projet :

- a) Boule aux couleurs de l'équipe, on peut choisir la couleur de son équipe dans le menu GUI ce qui modifie dynamiquement la couleur de la boule
- b) La boule doit avoir une courbe aux couleurs de l'équipe adverse.

### 4. Construction des quilles

La construction des quilles n'a pas été facile car nous sommes soumis à plusieurs contraintes par le sujet :

- a) À l'aide d'au moins trois surfaces de révolution avec une jointure G1 entre chaque surface, l'utilisation d'au moins deux « lathe » lisses est obligatoire et celles-ci doivent se raccorder entre elles ;
- b) Pour chaque quille, la « lathe » intermédiaire a une couleur différente des deux autres surfaces de révolution ;

Nous avons déjà commencé par créer les lathes avec des courbes sans créer l'objet, nous avons pris des courbes de Bézier cubiques pour plus de liberté sur la courbe.

Petit rappel de cours sur la construction des courbes de Bézier

#### Définition (Courbe de Bézier polynomiale)

Soit  $P_0, P_1, P_2$  et  $P_3$  quatre points non alignés de  $\mathcal{P}$ .

La courbe de Bézier polynomiale cubique de points de contrôle  $P_0, P_1, P_2$  et  $P_3$  est l'ensemble des points  $M(t)$  défini par :

$$\overrightarrow{OM}(t) = B_{0,3}(t)\overrightarrow{OP_0} + B_{1,3}(t)\overrightarrow{OP_1} + B_{2,3}(t)\overrightarrow{OP_2} + B_{3,3}(t)\overrightarrow{OP_3} \quad (18)$$

#### Définition (Courbe de Bézier de degré 3)

Pour  $i \in \llbracket 0; 3 \rrbracket$ , soit  $P_i$  une instance stockant les coordonnées du point  $P_i$  de  $\mathcal{E}_3$ .  
La syntaxe :

```
new THREE.CubicBezierCurve3 (P0, P1, P2, P3 );
```

permet de définir la courbe de Bézier polynomiale cubique de points de contrôle  $P_0, P_1, P_2$  et  $P_3$ .

#### Rappel :

Soit  $P_0, \dots, P_{n-1}, P_n$  les points de contrôle d'une courbe de Bézier  $\gamma_P$  de degré  $n$ .

Soit  $Q_0, \dots, Q_{m-1}, Q_m$  les points de contrôle d'une courbe de Bézier  $\gamma_Q$  de degré  $m$ .

Pour réaliser une jointure  $G^1$  entre les courbes  $\gamma_P$  et  $\gamma_Q$ , il suffit que :

- 1)  $P_n = Q_0$
- 2) les points  $P_{n-1}, P_n = Q_0$  et  $Q_1$  sont alignés.

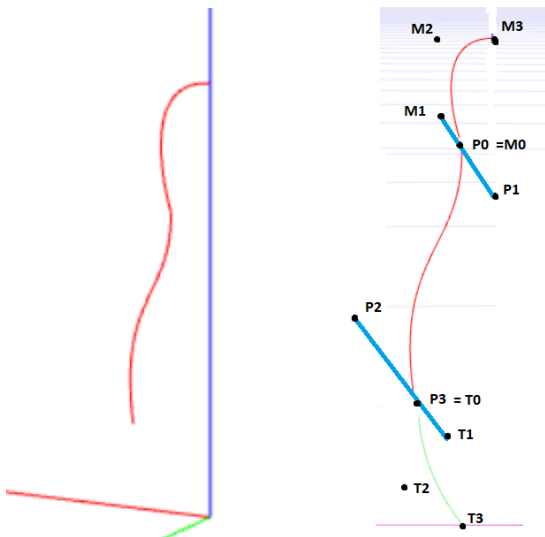


Figure 6 Quille en courbe de Bézier avec points

Figure 7 Quille en courbe de Bézier

Donc dans l'exemple ci-contre nous avons 3 lathes qui sont reliées les unes aux autres.

En utilisant les propriétés du cours ci-dessus, on est parti de la latte du milieu donc on a placé  $P_0$  point de début,  $P_3$  point de fin,  $P_1$  et  $P_2$  points de contrôle choisis nous-même, pour donner la forme de la quille.

Pour la construction de la deuxième latte, celle du bas, on a utilisé les propriétés de jointure et donc  $P_3$ , notre point de fin, on dit qu'il est égal à  $T_0$ , notre point de début de la deuxième latte. Ensuite on construit le vecteur  $P_2P_3 = P_2T_0$  qui est une tangente et sur cette tangente on vient ajouter  $T_1$ ,  $T_2$  sera notre point de contrôle qu'on choisira manuellement et  $T_3$  point de fin. Même procédé pour la latte du haut.

Une fois les courbes et la forme de la quille tracée, il est assez facile de modéliser l'objet, pour se faire on a utilisé `THREE.CubicBezierCurve3` (Nos 4 points). On stocke les points dans un tableau et on construit la latte avec `THREE.LatheGeometry` (Tableau de point), on fait la même chose pour les 3 lathes.

## 4. Construction de la scène et mise en place

La construction de la scène ne sera pas très longue car nous n'expliquerons pas le code dans le rapport qui n'est pas le lieu. Nous avons ajouté les quilles à la scène avec des boucles, de même pour les lattes au sol, le décor et la boule ont aussi été ajoutés dans la scène.

Nous avons ajouté des spots de lumières, pour éclairer les objets qui eux-mêmes vont recevoir la lumière et la renvoyer. Comme on peut le voir sur la figure 5 et 3 on a des ombres et pour ça on joue avec la lumière. Les quilles aussi reçoivent la lumière mais très faiblement car elles se trouvent dans une boîte.

## 5. Animation

### a) Déplacement rectiligne

Le déplacement rectiligne ne fut pas le plus dur.

On a défini les points de départ et d'arrivée et on a créé la ligne à partir des deux points avec `THREE.Line`.

On crée une ligne curve avec le même point de début et d'arrivée et on récupère un tableau de 10 points. Ce tableau de points va nous permettre de faire avancer la boule sur ses points en fonction du temps.

### b) Déplacement courbe de Bézier

Comme vu dans la « construction des quilles » on va construire de la même façon deux courbes de Bézier mais notre premier point sera notre boule et le point de fin la fin de la piste. De la même façon que pour les quilles nous avons réalisé des jointures avec des tangentes pour rendre le tout propre comme on peut le voir dans la figure 1.

Cette partie sur la construction des courbes ne nous a pas posé un problème, car nous avons déjà fait les quilles auparavant.

Maintenant la partie difficile que nous avons eu du mal à gérer et qui nous a pris le plus de temps, le déplacement de la boule sur les courbes de Bézier, je ne vais pas expliquer le code mais je vais expliquer notre procédé. Comme pour la droite rectiligne nous récupérons sur notre première courbe les points avant et après la jointure et on fait parcourir la boule et quand elle arrive un peu avant la jointure elle passe sur la deuxième partie de la courbe et fini sa course. Si seulement c'était si simple, les ralentissements de la machine qui font revenir en arrière la boule, la boule qui ne passe jamais sur la deuxième partie car les points qu'elle suit n'arrive pas suffisamment au bout et si on y change elle se téléporte... Nous avons stabilisé le problème mais la balle retourne en arrière encore et fini par s'arrêter correctement.

## 6. Collision et score

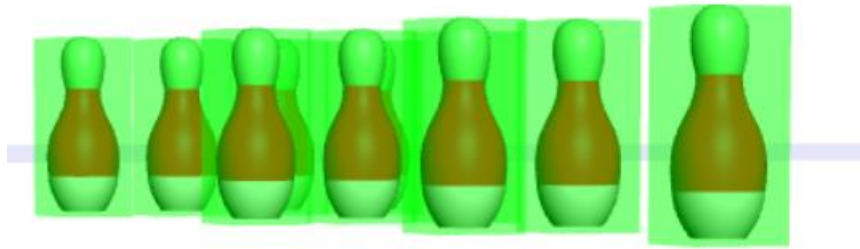


Figure 8 Box des quilles

Nous avons créé des BoxGeometry que nous avons placés sur chaque quille, pour que lorsque la balle touche une quille, elle disparaît. Il ne reste plus que la box de taille réduite. Mais dans la réalité nous n'avons pas réussi à faire cela, donc nous avons improvisé, les quilles ne voulant absolument pas disparaître même en changeant leur opacité et en le vérifiant dans la console, nous avons décidé de les laisser visible. Ducoup les box ne servent à rien mais vu qu'elles étaient demandées elles sont présentes.

Le comptage des quilles est compliqué vu que l'on ne peut pas savoir que la balle est passée à travers la quille vu qu'elle ne disparaît pas car il est impossible de récupérer les infos de la quille malgré le temps qu'on y a passé. Nous avons donc changé de façon de compter les points, à chaque lancer on va stocker dans le tableau une valeur aléatoire entre 0 et 5, ce qui nous a bien simplifié la tâche.

## 7. Menu GUI

Le menu GUI est la partie la plus complexe du projet car tout passe par le menu GUI dans le jeu. Ce fut la partie la plus difficile à gérer.

Le menu gérant la caméra est assez simple, mais il n'en reste pas moins complet. Il permet de modifier le point de vue de la caméra à tout moment.

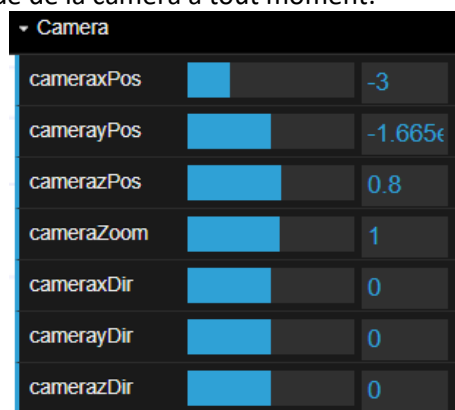


Figure 9 Déplacement camera



Il est possible de se déplacer avec la souris grâce à la bibliothèque « OrbitControls » que l'on a rajouté pour une question d'aisance.

Nous avons ajouté différents menus qui correspondent à la demande du sujet.

Pour commencer :



Figure 10 Menu GUI Equipe

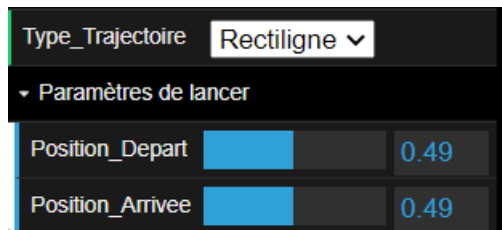


Figure 11 Trajectoire Rectiligne

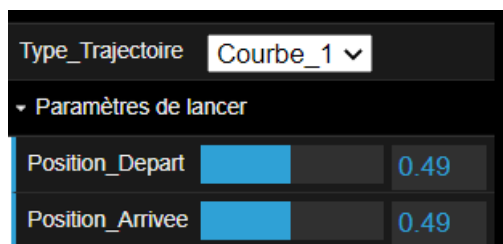


Figure 12 Trajectoire Courbe de Bézier



Figure 13 Bouton lancer

1) Possibilité de changer le couleur de l'équipe de façon dynamique durant la partie

2) On peut modifier le point de départ et d'arrivée de la ligne pour modifier la trajectoire.

3) Idem on peut aussi modifier la trajectoire de la courbe de Bézier

4) Nous avons un bouton lancer, qui comme son nom l'indique lance la boule qui suit la trajectoire sélectionnée en haut.

## 8. Conclusion

Ce projet nous a pris beaucoup de temps, et demandé beaucoup d'efforts, ainsi qu'une grande autonomie. Travailler sur un projet en binôme est une découverte et demande une bonne synchronisation des tâches et une bonne repartition du travail, ce que nous avons découvert avec le temps. En parlant de temps je pense que nous avons sous-estimé le projet qui était bien plus difficile et important qu'il n'y paraissait. Bien que nous avons les compétences avec un peu plus de temps notre projet aurait pu être mieux, ce qui indique que nous aurions pu mieux gérer notre temps de travail. Malgré toutes ces difficultés rencontrées et la console, nous avons acquis de nouvelles compétences et nous avons pu prendre conscience de nos difficultés et surtout s'amuser à faire un jeu de bowling.

## 9. Bibliographie

### Bibliothèque three.js :

RingGeometry :

<https://threejs.org/docs/index.html?q=ring#api/en/geometries/RingGeometry>

CircleGeometry :

<https://threejs.org/docs/index.html?q=circ#api/en/geometries/CircleGeometry>

BoxGeometry :

<https://threejs.org/docs/#api/en/geometries/BoxGeometry>

CubicBezierCurve :

<https://threejs.org/docs/?q=bezier#api/en/extras/curves/CubicBezierCurve>

QuadraticBezierCurve :

<https://threejs.org/docs/?q=bezier#api/en/extras/curves/QuadraticBezierCurve>

CylinderGeometry :

<https://threejs.org/docs/?q=cyli#api/en/geometries/CylinderGeometry>

Déplacement de la boule :

[https://gayerie.dev/docs/js/threejs/animation\\_curve.html](https://gayerie.dev/docs/js/threejs/animation_curve.html)