

GUIBARD Théo

GIE Jimmy

L2IE-03

L2 : Rapport projet Info3B Bowling



Objectif du projet et règle :

L'objectif du projet est de créer un jeu de bowling avec deux équipes de couleurs différentes qui s'affrontent le tout en 2 mènes chacune.

- si toutes les quilles sont renversées l'équipe rentre un strike et gagne 30 points.
- si toutes les quilles sont renversées au deuxième tour l'équipe rentre un spare et gagne 15 points.
- sinon les points marqués correspondent au nombre de quilles tombées.

Le projet permettra une approche de Three.js, une librairie qui permet d'ajouter du contenu 3D sur une page web.

Pour modéliser la piste ainsi que les différents éléments nous avons dû nous documentés sur les différentes caractéristiques du bowling comme les dimensions de la piste, des gouttières, le diamètre de la boule, des quilles et leurs positions.

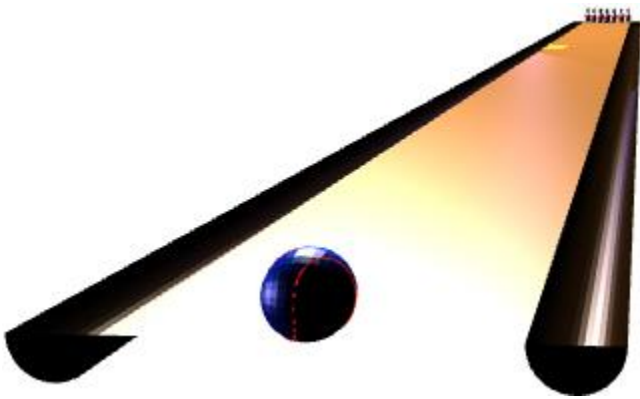
Source :

[https://www.ffbsq.fr/ffbsq/CNB/arbitrage/Livre III Caracteristiques techniques revision 2020 04 29.pdf](https://www.ffbsq.fr/ffbsq/CNB/arbitrage/Livre_III_Caracteristiques_techniques_revision_2020_04_29.pdf)

Création des objets à l'échelle :

Dans un premier temps nous avons créé la piste et les gouttières :

-La **piste** est tout simplement un plan de dimensions 1x20 de côté.

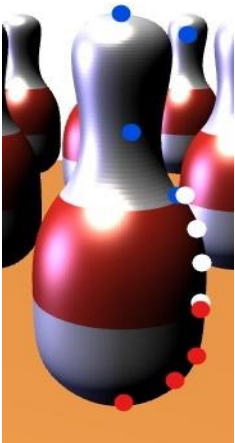


-Les **gouttières** sont deux cylindres d'angle $\theta_{\text{début}} = 0$ à $\theta_{\text{fin}} = \pi$ de dimensions 0,235x20 de côté (un peu plus large que la boule).

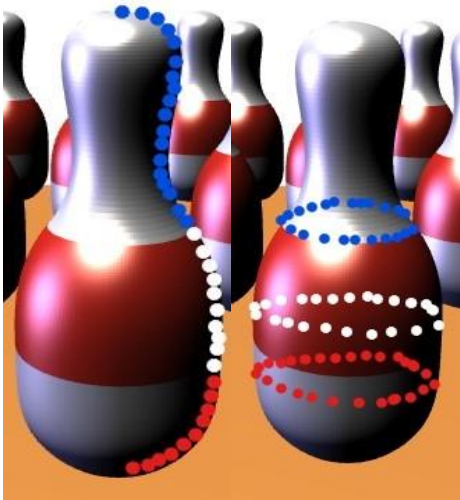
Pour créer les **quilles** nous avons besoin de savoir créer des lathes ainsi que des jointures G1. Une latte est une rotation autour de l'axe Y d'une courbe formant une sorte de vase.

Ici nous avons divisé la quille en 3 lathes avec une jointure G^1 les reliant par deux.

La jointure G^1 permet d'avoir une continuité entre les courbes permettant ainsi de ne pas avoir de « cassure » sur l'enchaînement des lathes.



La première étape pour modéliser une quille est de définir les points pour dessiner la courbe de chaque latte, ici chaque lathes possède 4 points, soit 12 points au total.



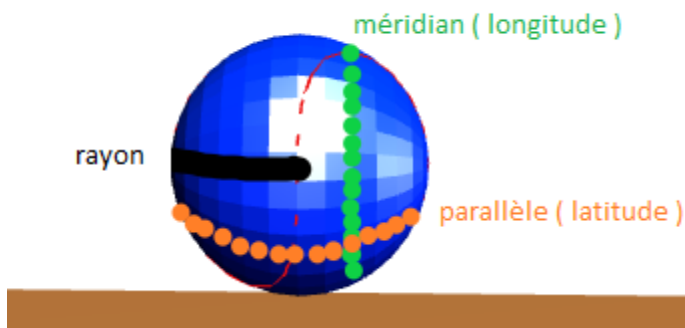
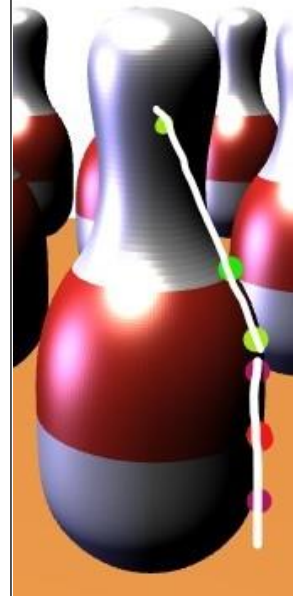
Nos lathes auront 2 paramètres important:

- le nombre de points tracés sur la courbe de Bézier cubique
- le nombre de rotation sur l'axe Y

Plus les valeurs seront grandes plus nos quilles seront détaillés.

Pour réaliser nos **jointures G1** il faut que les points de jointure soient confondus et qu'ils soient alignés sur leurs points de contrôle adjacents.

Pour placer nos quilles, nous avons utilisé la documentation au-dessus et mesuré chacun des emplacements pour chacune des quilles (fonction placerQuille)



La **boule** quant à elle est une sphère de rayon 0.108 et une courbe paramétrique la compose. Sa fonction de construction prend en compte la couleur de l'équipe qui joue pour que la courbe soit de couleur opposée. Un nombre de parallèles et de méridiens permettent le "lissage" de la courbe.

Nous avons précédemment présenté la géométrie des objets de notre scène. Mais sans **matériaux** elle serait complètement vide. C'est la fonction surfPhong qui nous sert alors de référence. Elle permet, à partir d'une géométrie, d'une couleur de l'objet, l'opacité, la transparence et la couleur de réflexion, de créer une texture à nos objets.

Les différentes trajectoires :

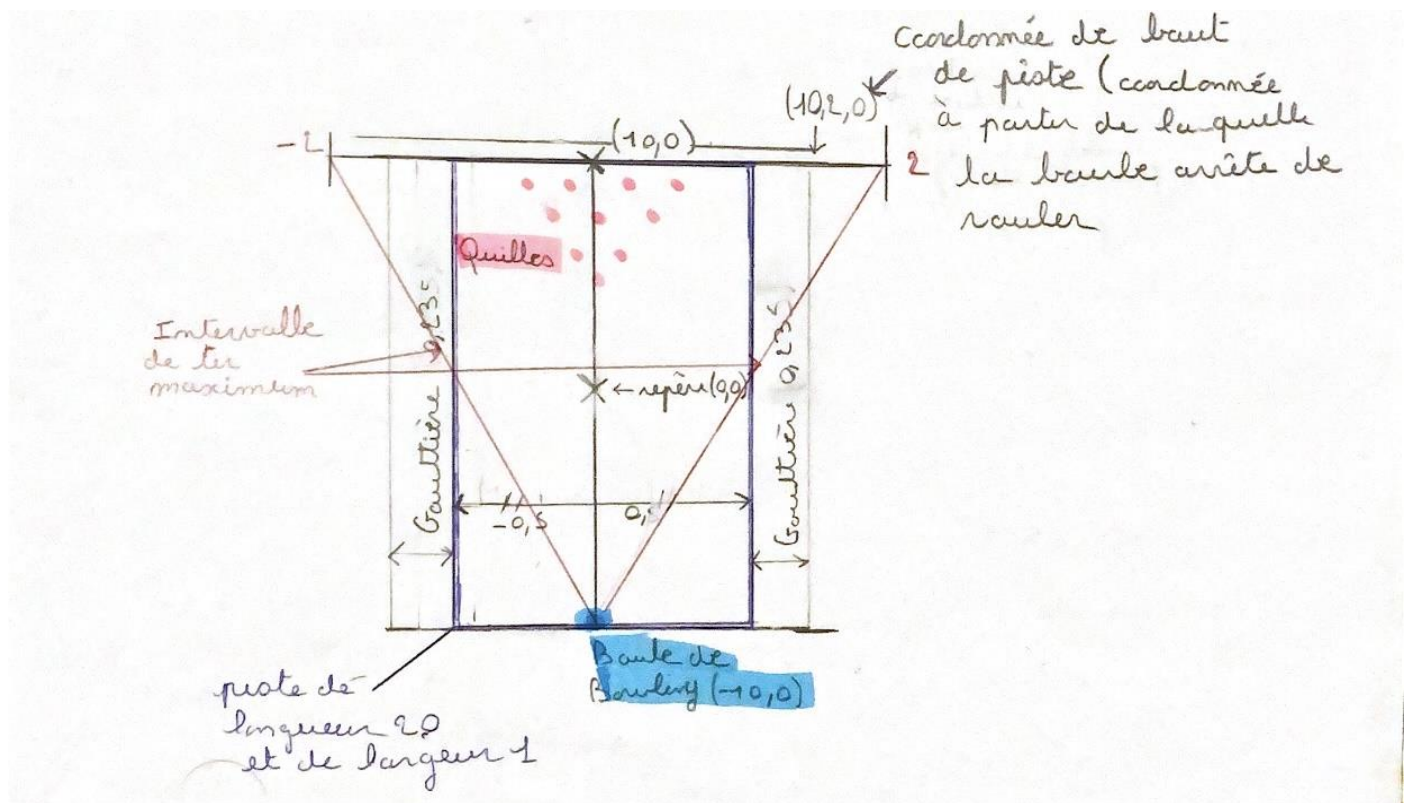
Aides : les trajectoires sont affichées par des droites ou des courbes/points pour les visualiser mais supprimées dès le début du lancer.

Trajectoire lancer rectiligne :

Notre trajectoire rectiligne possède deux points, un de départ et un d'arrivée. Pour le point de départ la boule de bowling peut être déplacée sur l'axe des Y, allant de -0.49 à 0.49, empêchant le joueur de lancer la boule de bowling dans les gouttières au moment où la boule est lancée. Le deuxième point est le point d'arrivée, où un angle de tire maximal est défini à la fin de la piste $x=10$ avec y compris entre -2 et 2, et le point de départ initial de la boule en $(-10,0)$.

Pour définir une trajectoire dans cet intervalle on calcule le coefficient directeur de la droite et l'ordonnée à l'origine (fonction droite). $Y = c \cdot X + o$

Avec la formule suivante : $\frac{y_b - y_a}{x_b - x_a} = c$ et l'ordonnée $y_a - c \cdot x_a$



Animation lancer rectiligne :

Maintenant que nous avons l'équation de notre droite, nous allons faire avancer notre boule dessus avec x qui incrémente selon la constante $\text{pas} = 20/100$ soit 100 points sur toute la piste de 20m (fonction roule).

Trajectoire lancer Bézier :

La trajectoire de Bézier, tout comme la trajectoire rectiligne possède un point de départ pouvant varier de -0.49 à 0.49 sur l'axe des y , pour ne pas tomber dans les gouttières au lancement de la boule.

La trajectoire Bézier est composée de **deux courbes de Bézier cubique** ayant chacune 3 points, un point de départ, un point de contrôle et un point d'arrivée. Les deux courbes sont reliées entre elle grâce à une jointure G^1 , où le point d'arrivée de la première est confondu avec le point de départ de la deuxième courbe, et où les points de contrôle adjacents aux point centraux sont alignés.

Désormais il faut permettre à l'utilisateur **des déplacements sans casser la jointure G^1** :

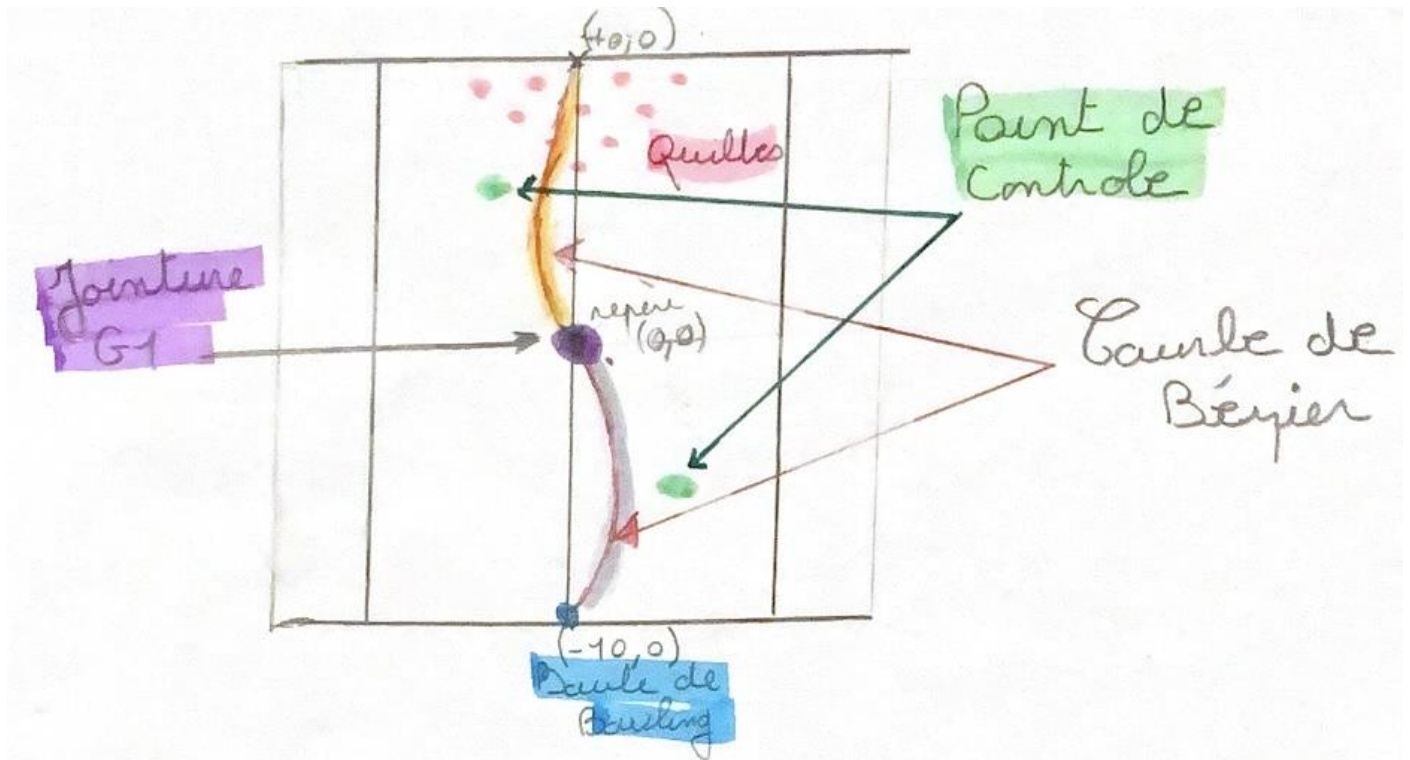
Les points centraux confondus :

- Lorsque l'on déplace ces point (x_{p2}, y_{p2}) les points de contrôle adjacents suivent exactement les mêmes déplacements.

Les points de contrôle adjacents :

- On applique une symétrie centrale sur les points centraux confondus. Alors, les points se suivent avec des déplacements similairement opposés.

Finalement, on garde une jointure G^1 avec des points alignés peu importe les déplacements effectués.



Animation lancer Bézier :

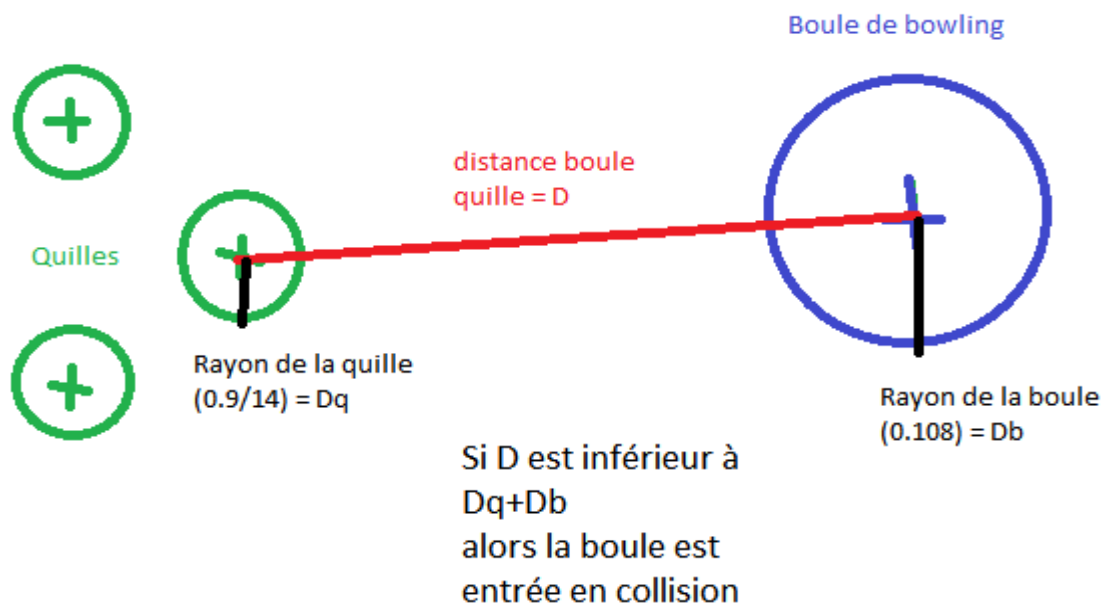
Afin de réaliser notre animation, notre boule va parcourir point par point les courbes de Bézier. La fonction `getPoints(nombre de points)` nous renverra un **tableau de points** de la courbe (ici des `vector3`). 2 tableaux seront déclarés et actualisés à chaque changement des courbes. Ensuite ils seront concaténés dans la fonction `tireBezier` pour que notre boule se déplace point par point sur la trajectoire complète. Dans tous les cas la boule s'arrête si elle dépasse la distance maximum d'un lancer qui est de 0,2x après la fin de la piste.

Animation des gouttières :

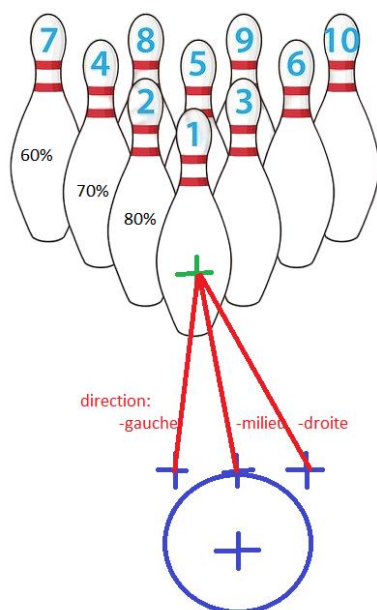
Si la boule sort de la piste et tombe dans une gouttière, il faut déterminer laquelle. On récupère alors la coordonnée en Y de la boule, si elle est positive alors delta est positif (gouttière gauche), inversement si elle est négative alors delta est négatif (gouttière droite). Pour finir la trajectoire, la boule parcourt en ligne droite la gouttière avec Y constant et X incrémenté par le pas.

Chocs entre quilles (fonction `toucheQuille`) :

Tout d'abord, pour savoir si notre boule a touché une quille on calcul la distance entre le centre de la boule et le centre la quille, si celle-ci est inférieur à leur rayon cumulé alors la boule est entrée en collision avec la quille.



Ensuite, nous avons défini 3 points à l'avant de la boule. Ceux-ci nous permettent de connaître la direction de l'impact afin de **déterminer la direction de la chute** de la quille (droite, gauche, centre). En fonction de cette direction, on renverra un tableau des quilles concernées. Exemple si je percute la quille 1 par la gauche de ma boule, alors ce sont les quilles 2, 4 et 7 (fonction quillesVoisine et schéma au-dessous) qui ont une chance de tomber. En effet, on parle de chance, car un élément aléatoire permettra une chute plus probable à une quille si elle est proche où lointaine de celle percutée à l'origine. (80% -> 70% -> 60% de chance)



Déroulement du jeu (fonction jeu) :

Pour suivre en direct, on compte le nombre de lancers effectués en permanence. Si les deux lancers d'une équipe ou qu'un strike est réalisé alors il faut changer l'équipe en redéfinissant la couleur de la boule, ajouter les points à l'équipe correspondante et replacer les quilles sur la piste (fonction reset). Afin de savoir si les 4 mènes se sont joués, on compte le nombre total de lancer, si celui-ci atteint 8 (un strike compte pour 2 lancers) alors la partie est finie (fonction finDuJeu).

Merci d'avoir lu notre rapport sur ce projet de bowling en webgl.

GUIBARD Théo GIE Jimmy
L2IE-03