Mardi 02 Décembre 2021

Rapport Projet Info3B

YILMAZ Rahman DUBOST Lucie

IE3-00-2b

Plan du rapport

1. **Introduction………………………………………………………………**
2. **Comment jouer…………………………………………………**
3. **Construction des objets…………………………………………………**

I - Construction de la piste…………………………………………………

II - Construction de la maison……………………………………

III - Construction des balais…………………………………………

IV - Construction des palets…………………………………………

1. **Construction et mise en place de la scène……………………………**
2. **Animations…………………………………………………**

I – Les déplacements……………………

1. Les déplacements rectilignes
2. Les déplacements selon des courbes de Béziers et leurs points de contrôle

II - Les chocs…………………………………………………

1. **Menus GUI…………………………………………………**
2. **Arbre C.S.G. …………………………………………………**
3. **Conclusion…………………………………………………**
4. **Bibliographie…………………………………………………**
5. Introduction

Nous avons eu à faire, durant notre troisième semestre de Licence en Informatique-Électronique, un projet portant sur une partie de curling, réalisé avec le langage JavaScript, plus précisément avec une bibliothèque spécifique, Three.js.

Three.js (ou ThreeJs) est une bibliothèque JavaScript utilisée pour créer des scènes 3D dans un navigateur web. Créée en 2010 par mrDoob, son code source est hébergé sur GitHub, dont vous trouverez le lien dans la Bibliographie.

Ce projet nous a été confié par notre professeur Lionel Garnier, achevant le module dont il a la charge : Info3B – Synthèse d’images.

Nous vous présentons donc aujourd’hui le résultat final d’un travail de plusieurs semaines, et espérons qu’il saura vous divertir.

Voici cependant quelques points à prendre en compte avant de vous lancer dans une partie.

1. Comment jouer

Inutile de vous le cacher, notre jeu n’est pas le plus minimaliste qui soit. Avec tous les paramètres qu’il offre, nous nous voyons dans l’obligation d’éclaircir vos lanternes.

Tout d’abord, sachez qu’il vous sera impossible de modifier les paramètres après le lancement de la partie.

Pour choisir la couleur de votre équipe, il vous faudra sélectionner votre équipe dans le menu « Partie », puis modifier les paramètres que vous souhaitez, comme la taille et la couleur de vos pierres et balais. Sachez que la couleur de votre équipe sera celle de la couleur du centre de vos palet, alors choisissez bien !

Avant de lancer une pierre, vous avez la possibilité de modifier sa trajectoire, qu’elle soit rectiligne ou qu’elle suive une courbe de Bézier. Adaptez la force de votre lancer et la force de frottement de vos balais pour tirer le plus proche de la maison.

Le tableau situé sous la piste contient la distance à la maison des palets lancés, selon l’équipe ayant tiré. Si votre équipe se retrouve mal en point vis-à-vis du score, n’hésitez pas à pousser les pierres de vos adversaires !

///////////////CAPTURE QUAND IL SERA OP//////////////

1. Construction des objets

I - Construction de la piste

Pour implémenter la piste, nous avons utilisé la classe CubeGeometry, car elle nous a permis de facilement créer le rectangle servant de piste. Vous pourriez vous demander pourquoi nous avons décidé d’utiliser CubeGeometry au lieu de BoxGeometry, une méthode moins désuète. C’est tout simplement car nous avons utilisé cette dernière dans la création du balai, et il nous a paru plus agréable de changer de méthode, nous adaptant à la polyvalence de Three.js.

Son matériel est un MeshPhongMaterial, Elle est centrée en (0, 0, 0).

Au départ, nous avions l’ambition d’apposer sur cette piste une texture de glace, pour plus de réalisme. Malheureusement, bien que cela fonctionnait à merveille sur l’ordinateur de l’un de nous, le second était bloqué par le « CORS policy », donc au lieu de nous prendre la tête avec ce contretemps, nous avons ôté la texture.



Figure – Piste finale (avec maison)

II - Construction de la maison

La maison est construite avec trois anneaux, tracés avec la classe RingGeometry, le premier, bleu, le second, blanc et le troisième, rouge. Le centre de la maison, lui, est construit avec un CircleGeometry, blanc.

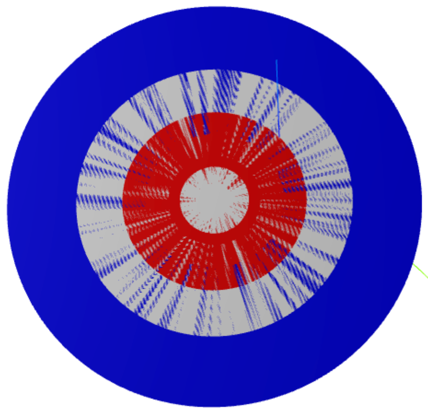
 

Figure 1 - Première tentative de la maison Figure 2 -Maison finale

Comme vous pouvez le constater dans la Figure 1, nous avons tout d’abord eu quelques problèmes avec la texture de la maison. Nous avions en effet de prime abord choisi de la coder en utilisant uniquement des CircleGeometry, c’est pourquoi les matériaux, se superposant, créaient des interactions entre eux, sans que nous ne sachions gérer ce genre de problème.

Nous avons alors modifié la méthode utilisée, nous penchant davantage sur des RingGeometry (voir Figure 2). Nous avons alors réuni tous ces anneaux et le cercle dans un unique groupe, ce qui nous a permis de manipuler l’ensemble bien plus facilement. Ainsi notre maison fut créée.

III - Construction des balais

Afin de créer les balais nécessaires au curling, nous avons décidé de coder chaque élément un par un. En effet, le manche, les poils, ainsi que le porte-poils de la brosse ont vu le jour séparément. C’est par la suite que nous les avons regroupés dans un seul et même objet : le balai. Il nous a fallu pour cela ajuster les emplacements de chaque élément vis-à-vis des autres.

Le manche, ainsi que les poils, sont créés par un CylinderGeometry ; la brosse par un BoxGeometry. Encore une fois, le choix des matériaux est un MeshPhongMaterial, car il offre plus de paramètres qu’un MeshBasicMaterial.

Tout comme pour la piste, l’idée de départ était de donner au manche un aspect plus réaliste en passant par une texture, mais comme vous vous en doutez, un même problème mène au même résultat, c’est-à-dire un abandon de cette idée.

Nous avons par la suite tout rassemblé dans un unique groupe, afin de créer un unique objet, et ceci grâce à la méthode groupe.add(élément).

Petit détail : l’utilisateur peut choisir le nombre de poils figurant sur son balai, de 10 à 1 000. Sachez que cela ne change strictement rien à la partie.

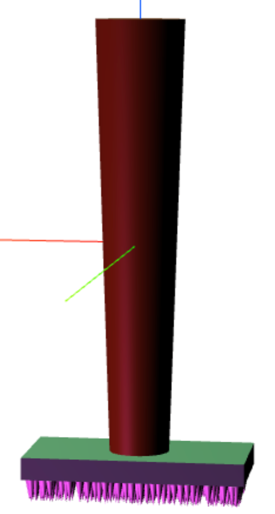
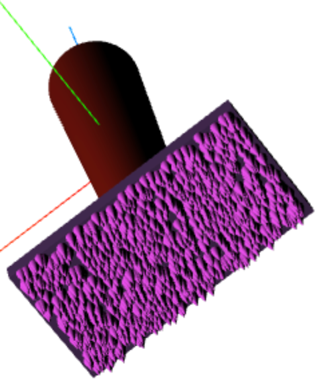
 

Figure – Balai final Figure – Répartition des poils

IV - Construction des palets

Tout comme pour les balais et la maison, chaque élément fut codé à part, mais quelques divergences fondamentales existent entre ces groupes. En effet, la construction des palets fut soumise à de nombreuses consignes :

1. utilisation de lathes et d’au moins trois surfaces de révolution lisses raccordées par un raccord G1;
2. la couleur de la pierre de chaque équipe est différente et la lathe intermédiaire arbore une couleur différente de celles qui l’entourent.

Nous avons décidé de laisser la liberté au(x) joueur(s) de choisir la couleur du palet (retrouver comment procéder dans la section « Comment jouer » ///+ lien jusqu’à la section). Ainsi, chaque équipe pourra jouir du palet de la couleur qu’il souhaite.

Passons à leur construction.

Le bas et le haut de la pierre sont construits avec deux CircleGeometry, placés en (0, 0, 0) quant au groupe. L’anse du palet est créée par deux CylinderGeometry. Les surfaces de révolution /////////////////////////////////////////////

Comme nous le savons, pour qu’une jointure G1 soit réalisée, il faut que :

- le dernier point de la première courbe soit égal au premier point de la seconde ;

- ces même deux points sont alignés avec l’avant-dernier point de la première courbe.

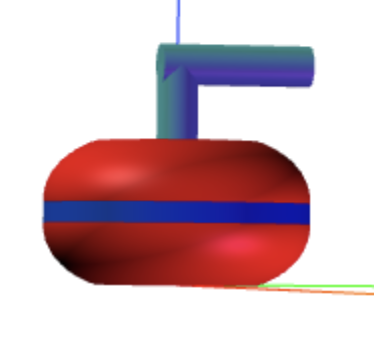
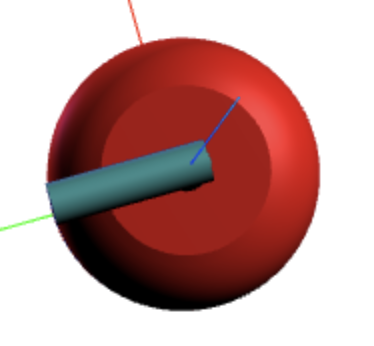
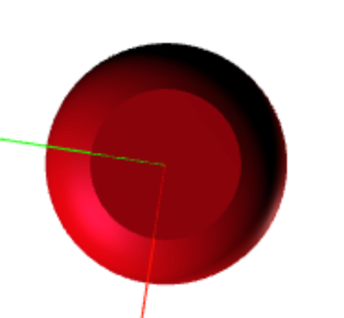
  

Figure – Palet final Figure – Palet vu de haut Figure – Palet vu du bas

1. Construction et mise en place de la scène

La création de notre scène ne fut guère différente de celles utilisées tout au long du semestre, et ce rapport n’ayant lieu d’être une explication du code, nous serons brefs.

1. Animations

Les animations ont constitué un point particulièrement retors dans la création de ce projet.

I – Les déplacements

1. Les déplacements rectilignes

Les déplacements en ligne droite ne nous ont posé aucun problème. Il nous a simplement fallu indiquer que les palets se déplaceraient selon l’axe x, avec une force variant selon le choix des utilisateurs (voir section « Menu G.U.I. » ////////+lien).

1. Les déplacements selon des courbes de Béziers et leurs points de contrôle

Fini de rigoler, passons à Bézier.

Une image contenant texte

Description générée automatiquement

Figure – Rappel concernant les jointures G1

En

II - Les chocs

Nous avons eu un type de chocs à gérer : les chocs entre les différentes pierres lors des lancers.

Pour se faire, nous avons créé une fonction détectant les chocs, nommée « chocDetected », et une autres les animant, « chocanime ».

chocDetected : Un choc est détecté dès que le rayon de la première pierre ajouté au rayon d’une seconde pierre avec laquelle la première va rentrer en collision est plus petit ou égal ///////////à la distance entre les deux ?////////////

chocanime : L’animation des chocs

Vous observerez un léger temps de latence entre le moment de la collision, et celui ou la pierre touchée se fait projeter.

Nous avons fait le choix de ne pas implémenter les collisions entre les balais et les pierres, car cela est interdit dans les règles du jeu, donc bien qu’il puisse vous sembler étrange de voir un balai traverser une pierre, sachez que c’est un choix que nous assumons.

1. Menus GUI

Nous avons pris de nombreuses libertés quant à l’implémentation du menu de notre projet, afin de permettre une totale personnalisation des parties à chaque utilisateur.

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, excepté lors d’un lancer (voir Figure 3).



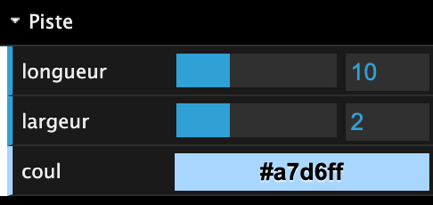
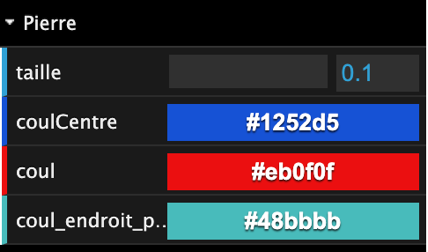
Figure 3 – Menu « Caméra »

Abordons à présent les menus maniant les différents objets.

Le menu régissant la piste vous donne l’occasion de jouer sur une piste de curling assez différente des autres : vous pouvez en modifier la longueur et la largeur à souhait, et, bien entendu, la couleur de la piste si celle par défaut ne vous plaît guère. À vous de voir si vous préférez jouer sur une très longue piste ou non (voir Figure 4).

Le menu gui nommé « balais » permet de modifier la taille des composantes des balais, que ce soit la largeur du manche ; sa longueur ; sa hauteur ; le nombre de poils le constituant, la couleur de tous ses composants, et j’en passe (voir Figure 5).

Le menu nommé « Pierre » vous permet de gérer la taille de vos palets, la couleur de l’anse, de la lathe centrale, et des surfaces environnantes (voir Figure 6).

  Une image contenant texte, capture d’écran, tableau de points

Description générée automatiquement

Figure 4 – Menu « Piste » Figure 5 – Menu « Pierre » Figure 6 – Menu « balai »

Voyons les deux dernières parties de ce long menu, celles contrôlant le jeu.

Dans le menu « Partie », il vous est proposé trancher quelle sera votre équipe, entre la 1 et la 2. Nous avons déjà évoqué la particularité de cette option dans la partie « Comment jouer ».

Viennent ensuite les options permettant de commencer ou recommencer la partie (voir Figure 7).

Le dernier menu est le plus important de tous, car c’est celui qui vous amènera peut-être la victoire : j’ai nommé le menu « lancer ». Avec lui, gérer les différentes forces de vos lancers, et choisissez leur trajectoire entre une ligne droite et une courbe de Bézier. Cliquez sur l’option « lancer » pour que les palets se mettent à glisser (voir Figure 8).

Une image contenant texte

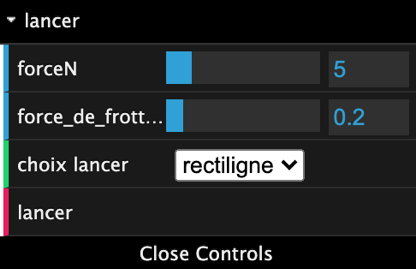
Description générée automatiquement 

Figure 7 – Menu « Partie » Figure 8 – Menu « lancer »

1. Arbre C.S.G.

L’arbre C.S.G. avec explications :

Le C.S.G. (pour Constructive Solid Geometry) est une technique de modélisation utilisant des opérations binaires sur les solides. Il nous permet alors d’unir, de soustraire, d’intersectionner deux solides entre eux, donnant par la suite de nouvelles formes.

Il ne nous est plus demandé dans le sujet de l’utiliser, mais il nous faut en parler.

Un arbre C.S.G. ressemble à cela, imaginons pour créer un palet :

Manche (création des deux cylindres de l’anse)

CSGCylinder1

CSGCylinder2

Pierre (création d’une

CSGSphere

CSGBox

Palet = Manche.union(Pierre) (union des deux éléments)

1. Conclusion

Ce projet nous a demandé beaucoup de réflexion, une grande autonomie, ainsi qu’un bon travail d’équipe. Nous sommes fiers d’être parvenu au bout de ce que nous voulions vous présenter, bien qu’il ne soit pas parfait. Nous avons certes quelques regrets (comme la texture de la piste ou des balais), mais ils sont moindres lorsque nous visualisons le résultat final de nos nombreuses semaines de travail. Les tracas furent nombreux, mais surmontables, ce qui nous a permis de grandement comprendre les notions apportées par le module d‘Info3B.

Nous espérons qu’il saura vous divertir, et surtout vous rendre fiers de vos élèves.

À présent, place au jeu !

1. Bibliographie

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

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

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

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

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

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

<https://threejsfundamentals.org/threejs/lessons/threejs-primitives.html>