

MAR - TP

Pierre Gaudichon

Hiver 2017

Abstract

Ce rapport présente le projet de MAR réalisé en Janvier 2018. Le but est d'animer une voiture sur un circuit, de créer un hélicoptère qui embarque une autre caméra et finalement de créer des nuages de particules.

Contents

1	Gestion des caméras	2
2	Réagir aux événements de la course	2
3	Enregistrement et replay des parties	2
4	Création d'hélicoptères	3
4.1	Corps de l'hélicoptère	3
4.2	Courbe de bézier	3
5	Génération de nuages de particules	3
5.1	L'émetteur en cône	4
5.2	La gravité	4
5.3	L'intégration sur l'hélicoptère	4
6	Conclusion	4

1 Gestion des caméras

La première partie a consisté en la création de caméras fixes le long du circuit.

Pour cela on créé des points autour du circuit ou seront posées les caméras. On récupère ensuite le *plane* sur lequel est la voiture. Enfin, on affiche la camera qui correspond a ce *plane*. Les cameras suivent la voiture du regard.

Pour *switcher* de camera, il faut utiliser la touche **(P)**. Les différentes caméras sont :

- La principale (donnée) qui suit la voiture.
- Une sur le dessus de la voiture (donnée aussi), dont la hauteur réagit a la vitesse.
- Celles le long du circuit (décrite au dessus).
- Une sur le dessus, qui regarde tout le *track*.
- Une sur l'hélicoptère, qui regarde vers le bas (décrite section 4.1).
- Une sur l'autre hélicoptère (décrite section 4.2).

2 Réagir aux événements de la course

Le circuit est séparé en différents *planes*. On récupère les changements de planes pour détecter trois événements : la fin d'un tour, la fin de la course et quand la voiture va dans mauvaise direction. On enregistre et affiche aussi les temps par tour.

C'est a ce moment du TP que j'ai créé un module de debug pour afficher différentes informations sur l'ensemble du système. Ce module de debug et l'affichage des temps de course sont gérés avec **Vue.js** et **jQuery**.

3 Enregistrement et replay des parties

Pour pouvoir afficher les ghosts des anciennes parties, la position de la voiture a chaque boucle de rendue. Ces positions sont enregistrées dans le *localStorage* puis réaffichées dans les parties suivantes.

Par contre, on n'enregistre la position que sur les axes x et y . A la première pente, la voiture continue tout droit et disparaît dans le décor. Ce bug n'a pas été corrigé, parce que le temps manquait après avoir fait les parties sur l'hélicoptère et le nuage de particules.

4 Création d'hélicoptères

4.1 Corps de l'hélicoptère

Le corps de l'hélicoptère a été assemblé à partir des différentes parties mises à la bonne position. Pour avoir un code plus clair, chaque partie (*Helico*, *Propeller*, *Blade*) est dans une classe spécifique.

Le premier hélicoptère (*HelicoKey*) est contrôlé par les touches du clavier. Celles ci sont déconnectées de la voiture pendant que l'hélicoptère à le contrôle.

Comme demandé dans le sujet, la vitesse de rotation des pales de l'hélicoptère dépend de la vitesse de l'appareil. De plus les moteurs (*Propellers*) tournent en avance de l'appareil.

4.2 Courbe de bézier

Le deuxième hélicoptère est contrôlé par une courbe de Bézier. Sa position est posé sur la courbe et sa vitesse est dirigé par la dérivé de la fonction qui créé cette courbe.

Au lieu d'utiliser le générateur de courbe de Bézier de **ThreeJS**, j'en ai créé un a la main. C'était une erreur en terme de productivité, mais c'était intéressant de manipuler des polynômes.

5 Génération de nuages de particules

La partie finale du TP est l'ajout d'un nuage de particules qui représente de la fumée aux turbines de l'hélicoptère.

5.1 L'émetteur en cône

La première version est un émetteur en cône en utilisant le moteur de particules fourni. On fait varier l'opacité et la couleur des particules avec les modules donnés.

5.2 La gravité

Pour intégrer la gravité, il faut attacher le système de particules à la scène directement pour qu'il n'y ait pas de rotation du vecteur qui représente le haut. Pour cela, la position initiale des particules est déterminé par un *Object3D*. Ces changements on impliqués beaucoup de modifications de la classe *ConeEmitter_Class* (renommé en *AbsoluteConeEmitter* dans le processus).

5.3 L'intégration sur l'hélicoptère

Finalement, le système de particules est intégré dans l'hélicoptère. On affiche de la fumée à la sortie des deux turbines latérales.

Il y a un problème dans la gestion de la couleur des particules. Quoi qu'on fasse, on ne peut pas mettre la couleur à noir ($\{r:0, g:0, b:0\}$ affiche du blanc). Pour contourner ce problème, au lieu d'afficher de la fumée, on affiche du feu (couleur orangée).

6 Conclusion

Ce TP d'animation 3D a permis de voir différents concepts essentiels dans l'animation : les jeux vidéos, notamment les nuages de particules, les courbes de Bézier et la composition des graphes de scène.