

École Polytechnique de Montréal
Automne 2005, 12 décembre 2005

Projet : Éditeur de *roller coaster*

INF6802, Réalité virtuelle

Nom	Matricule	Groupe
HERVE LOMBAERT	1075601	1
YVES CHOUINARD	41710	1

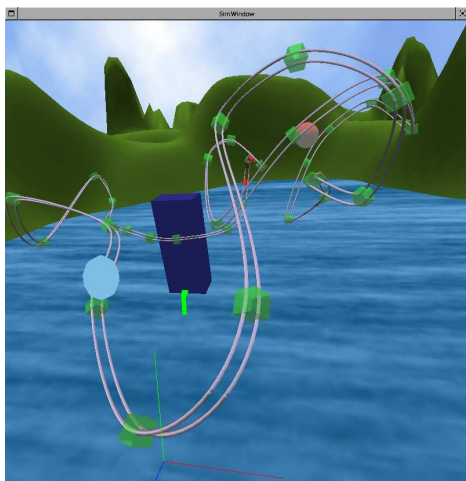


FIG. 1 – Édition du parcours d'un *roller coaster*

1 Exposé du projet

Il serait intéressant d'éprouver les sensations fortes d'un *roller coaster* en immersion. Ce projet veut répondre à cette question en permettant de créer rapidement un *roller coaster* et de l'essayer. Sa réalisation se fait dans l'environnement de la CAVE, 4 écrans géants de 3 mètres entourant l'utilisateur permettent de visualiser le *roller coaster*. Un joystick 3D permet d'éditer le parcours du *roller coaster*. Le programme possède deux modes : un mode d'édition et un mode de simulation. Les commandes du programme seront d'abord expliquées. La réalisation du projet sera ensuite décrite, pour terminer sur une discussion générale des résultats obtenus.

2 Mode d'utilisation du programme

2.1 Exécution

Le programme utilise les bibliothèques VRJuggler 2.0.1, et OpenGL 1.6.0. La compilation se fait en configurant correctement les variables d'environnement de VRJuggler et OpenGL (variables fournies en annexe). Sans l'environnement de la CAVE, VRJuggler permet l'exécution sur un ordinateur personnel en mode simulé :

```
$ ./rollercoaster standalone.jconf
```

Dans l'environnement de la CAVE, le programme doit être exécuté sur les ordinateurs contrôlant chaque écran. Un script fourni en annexe est utilisé :

```
$ ./startCAVE rollercoaster
```

2.2 Commandes

Lorsque le programme est lancé, il se trouve en mode d'édition. L'interaction se fait avec le joystick 3D qui comporte 3 boutons. Le premier bouton est une gachette sur le joystick 3D. Le second bouton est sur le dessus du joystick 3D, à gauche. Le troisième bouton se trouve également sur le dessus, à droite. Chacun de ces boutons réagit selon :

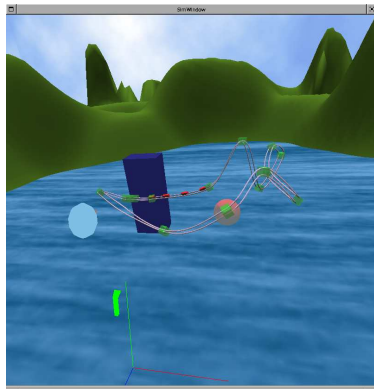
- 1er bouton : insère et déplace un point de contrôle,
- 2eme bouton : élimine un point de contrôle, ou déplace l'utilisateur,
- 3eme bouton : bascule l'utilisateur entre le mode d'édition et de simulation.

En mode de simulation, seul le troisième bouton est utilisé. En mode d'édition, le joystick 3D contrôle un curseur représenté par une sphère rouge transparente. Le curseur permet de manipuler les points de contrôle du *roller coaster* représentés par des cubes verts transparents.

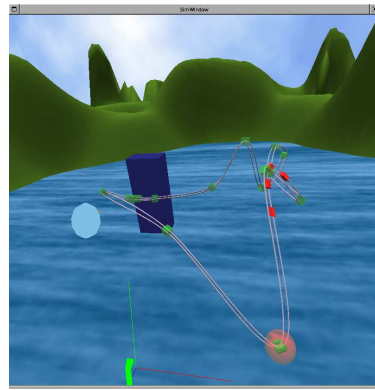
Lorsque le curseur ne pointe sur aucun point de contrôle, l'utilisateur peut se déplacer dans le monde en maintenant le second bouton enfoncé et en penchant le joystick 3D vers la direction qu'il souhaite se diriger. En utilisant le premier bouton, il insère un nouveau point de contrôle après le point de contrôle le plus proche. En maintenant le premier bouton enfoncé, le nouveau point de contrôle est déplacé jusqu'au relâchement du bouton.

Lorsque le curseur se trouve relativement proche d'un point de contrôle, l'utilisateur peut l'effacer avec le second bouton, ou bien le déplacer en maintenant le premier bouton enfoncé.

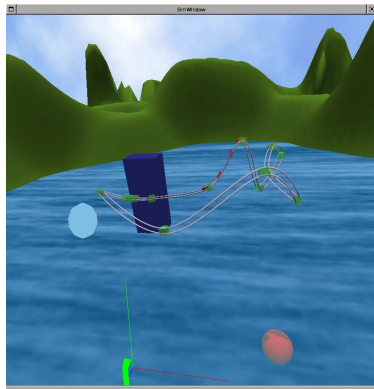
La figure 2 illustre un scénario classique d'utilisation lors de l'édition. La figure 3 montre quelques vues prises lors du parcours illustré à la figure 1.



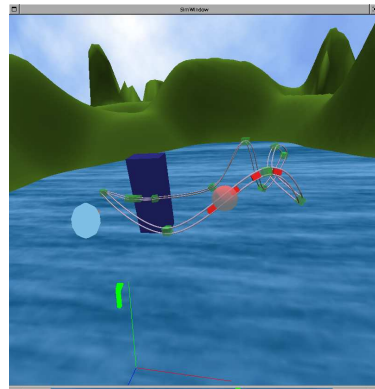
(1) Placer le curseur sur un point de contrôle



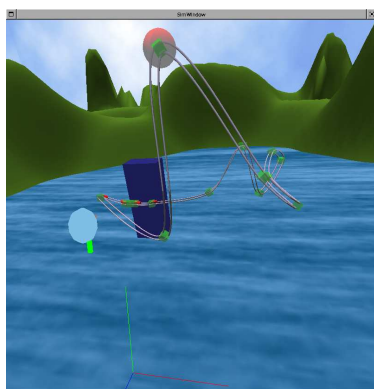
(2) Déplacer le point de contrôle en maintenant le 1er bouton enfoncé



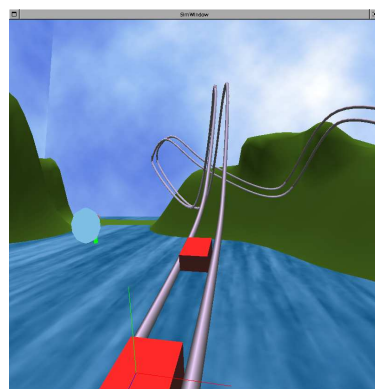
(3) Effacer le point de contrôle avec le 2nd bouton



(4) Placer le curseur entre deux points de contrôle

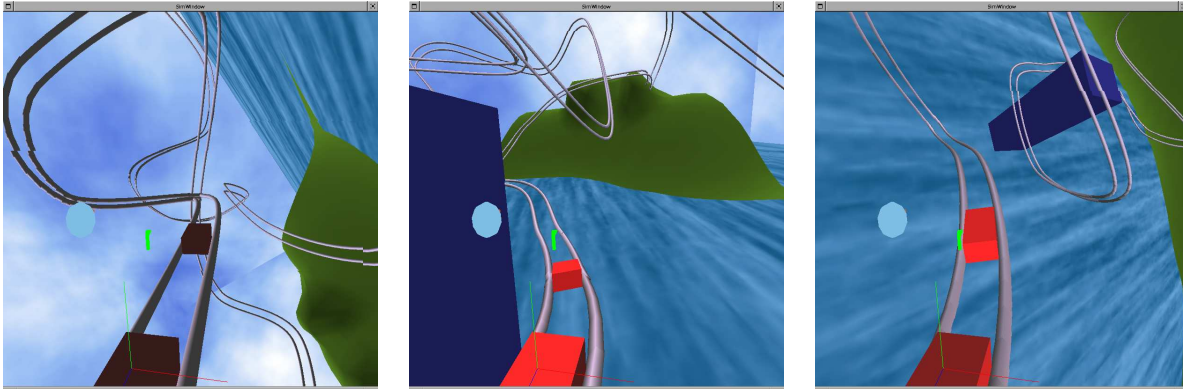


(5) Insérer d'un point de contrôle avec le 1er bouton



(6) Basculer en mode de simulation avec le 3ème bouton

FIG. 2 – Scénario classique d'utilisation de l'éditeur de *roller coaster*

FIG. 3 – Simulation du *roller coaster*

3 Structures de données et algorithmes

Le mouvement du *roller coaster* fait appel à quelques notions de mécanique. Également, sa position nécessite l'utilisation de fonctions d'interpolation. De plus, une méthode originale est utilisée pour le choix de la direction du train, notamment de son *up vector*.

Le mouvement du train fait appel à la conservation de l'énergie. Ainsi, en tout temps, il y a un équilibre entre l'énergie cinétique E_c et l'énergie potentielle E_p . En sachant la hauteur maximale h_{max} du *roller coaster*, et la vitesse du train v_{max} à cette hauteur :

$$\begin{aligned} E_{c_t} + E_{p_t} &= E_{c_{max}} + E_{p_{max}} \\ \frac{1}{2}mv_t^2 + mgh_t &= \frac{1}{2}mv_{max}^2 + mgh_{max} \\ v_t^2 &= v_{max}^2 + 2g(h_{max} - h_t) \end{aligned}$$

L'expression précédente permet de calculer la vitesse du train v_t en fonction de sa hauteur h_t . Ainsi, à chaque nouvel incrément de temps dt , le train est avancé d'une distance $l_{t+1} = l_t + v_t * dt$.

En annexe, la classe `speed` est en charge de calculer la vitesse du train, ainsi que sa position à une distance donnée.

Le train passe par les positions des points de contrôles. L'interpolation utilisée est celle de Catmull-Rom. La distance l_t parcourue par le train permet de connaître les points de contrôle voisins au

train. Avec la succession de points contrôle p_0 , p_1 , p_2 et p_3 , sachant que le train se situe entre p_1 et p_2 , à une distance s , où $s = 0$ correspond à la position de p_1 , et $s = 1$ correspond à la position de p_2 , il est possible d'interpoler la position du train $p(s)$ par :

$$p(s) = \frac{1}{2} \begin{pmatrix} 2p_1 + (-p_0 + p_2)s + (2p_0 - 5p_1 + 4p_2 - p_3)s^2 + (-p_0 + 3p_1 - 3p_2 + p_3)s^3 \end{pmatrix}$$

L'orientation du train est également interpolée par cette fonction. L'orientation est représentée par des quaternions, il n'y a alors qu'un vecteur et un angle de rotation à interpoler.

Lors de l'insertion de nouveaux points de contrôle, le choix de leurs orientations doit tenir compte du mouvement du train. Le *up vector* du train est mis à jour en parcourant tout le chemin du *roller coaster* à petit pas. Chaque mise à jour utilise la direction du *roller coaster*, et un produit en croix est appliqué depuis l'orientation initiale du train.

En annexe, la classe `path` est en charge d'interpoler la position et l'orientation du train à une distance donnée. Le choix de l'orientation des nouveaux points de contrôle est également déterminé par la classe `path`.

Le graphisme du projet se fait en utilisant un graphe de scène. Au sommet du graphe se trouve une transformation générale. Cette transformation permet à l'utilisateur de se situer face au *roller coaster*

de s'y déplacer en mode d'édition, ou dans le train en mode de simulation. Suite à cette transformation se place une lumière qui éclairera tous les éléments de la scène qui se trouve en dessous dans le graphe. Les éléments sont le train et la structure du *roller coaster*, mais également l'environnement comme les montagnes, le lac, le ciel, et des repères visuels. Dans le graphe, le train contient les différents wagons, et la structure contient une série de rails et les points de contrôles.

Lorsque le parcours du *roller coaster* est édité, les éléments de la structure modifiés sont recréés entre les points de contrôle mis à jour. Au courant de l'exécution du programme, différentes transformations sont mises à jour afin de bouger les wagons du train en fonction du temps, la position du curseur en fonction de la position du joystick 3D.

En annexe, la classe `rc` gère le graphe de scène. À chaque modification du système, cette classe mettra à jour les différents éléments du graphe permettant ainsi aux éléments de bouger, à l'utilisateur de se déplacer, ou de déplacer les éléments de la scène.

4 Discussion

L'essai du *roller coaster* en immersion permet de recréer des sensations fortes. Les sensations de vitesse, de vertige, et d'étourdissements ont été ressentis. Cela même si le réalisme du projet n'a pas été poussé. Des formes géométriques simples permettent à l'utilisateur de se laisser emporter dans le monde virtuel. L'édition du *roller coaster* proposée offre une manipulation simple et efficace à l'utilisateur.

Le déplacement dans le monde virtuel pour l'édition et la visualisation du *roller coaster* s'est avéré être un atout important. Il permet à l'utilisateur de ne plus être restreint à l'espace physique de l'environnement de la CAVE. En plus de se déplacer physiquement dans le cube immersif, il peut se déplacer dans le monde virtuel.

Les interactions avec le joystick 3D n'ont pas été contraignantes. Contrairement à une souris, il permet de contrôler facilement un curseur 3D. Pour l'édition du *roller coaster*, les boutons du joystick 3D permettent de manipuler correctement les points de contrôle. Nous avons opté pour l'utilisation intuitive d'une commande par bouton, mais des alternatives

intéressantes sont envisageables. Par exemple, au lieu d'utiliser le second bouton pour éliminer un point de contrôle, un double click économiserait un bouton. Le basculement du mode d'édition au mode de simulation peut se faire en lâchant le joystick 3D. Ainsi, avec un seul bouton utilisé pour toutes les manipulations, l'utilisateur peut avoir une interaction plus intuitive.

Afin d'augmenter l'immersion dans le *roller coaster*, des effets sonores ont été ajoutés. Sans ces effets, il était difficile de s'imaginer dans le train en mode simulation. Les effets sonores à des moments critiques aident à s'imaginer des sensations fortes dans les courbes et bosses du *roller coaster*. En lançant un son de cri à ces moments, l'utilisateur s'imaginer plus facilement qu'il se trouve dans le train virtuel.

Les outils utilisés pour la description physique du *roller coaster* ont amené quelques problèmes inattendus. En particulier, l'interpolation de Catmull-Rom possède un désavantage qui nous était inconnu. Elle estime une position entre deux points en sachant la distance normalisée s du point interpolé. À $s = 0$, la position du point interpolé correspond à la même position que le premier point de contrôle. À $s = 1$, elle correspond au dernier point de contrôle. À $s = 0.5$, le point interpolé se trouve bien à mi chemin entre les deux points de contrôle. Par contre, à une distance différente, par exemple, $s = 0.1$, la position interpolée ne se trouve pas exactement au dixième de la distance entre les deux points de contrôle, elle se trouvera plutôt à une distance légèrement inférieure à 0.1. En faisant alors varier un train à une vitesse fixée constante sur une courbe interpolée, le train ira plus vite à mi-chemin entre deux points de contrôle, et ralentira légèrement lorsqu'il est proche d'un point de contrôle. Ce phénomène n'est pas perceptible avec un unique wagon, mais en utilisant plusieurs wagons, cela provoque un effet accordéon sur le train. Les wagons se rapprochent les uns aux autres lorsqu'ils sont proches d'un point de contrôle. Pour éviter ce problème, il faudrait alors utiliser d'autres fonctions d'interpolation. Les perturbations de la fonction de Catmull-Rom étant quasiment négligeables à notre projet, nous n'avons pas poussé plus loin la résolution de ce problème. Au contraire, l'effet accordéon sur le wagon est un phénomène intéressant. Il ajoute une touche d'imprévisibilité au graphisme, aidant ainsi à mieux s'imaginer deux wagons qui peuvent se cogner

en réalité.

La stratégie de choix des orientations des points de contrôle a été délicate. Si par exemple pour une *loop* formée par 4 points de contrôle, les *up vectors* étaient fixés arbitrairement vers le haut, le train ne se retrouverait pas à l'envers au sommet de la *loop*. Il se retrouverait même à changer brusquement de sens en montant et descendant la *loop*. Seule la direction du chemin du *roller coaster* est connue (la soustraction des positions des points suivant et précédent sur la courbe). Il est possible de fixer arbitrairement le *right vector* toujours vers la droite, et de retrouver le *up vector* en effectuant un produit en croix. Le train effectuera alors correctement une *loop* en se retournant à son sommet. Par contre, le même problème survient lors d'un virage horizontal. Le *right vector* étant arbitrairement fixé à droite, le train se retrouverait à changer brusquement de direction à mi-chemin du virage. Il faut donc faire un suivi de l'orientation du train à partir d'une orientation fixée initialement. Chaque nouveau point de contrôle a donc une orientation calculée automatiquement à partir de l'orientation du pas précédent. L'utilisateur se retrouve alors avec peu de contrôle sur l'orientation du *roller coaster*. Une alternative consiste à fixer arbitrairement les points de contrôle avec une orientation fixe, et demander à l'utilisateur de la mettre à jour tout le long de l'édition. Il a été jugé que le choix automatique plutôt que manuel permet de faciliter l'édition du *roller coaster*. Il y a donc eu un compromis entre la rapidité d'édition et le contrôle total de l'édition.

L'environnement de la CAVE a des avantages d'ordre pratique. Il est en effet difficile d'avoir accès à une manipulation 3D avec un ordinateur personnel et sa souris. Les projecteurs de la CAVE et le joystick 3D permettent une manipulation directe des objets dans le monde virtuel. Il serait possible d'envisager un joystick 3D pour un ordinateur personnel avec une visualisation sur un moniteur stéréoscopique, mais se trouver face à un écran ne permet pas le même degré de perception que l'environnement de la CAVE : marcher sur le train et avoir la totalité du champ de vision disponible n'est pas possible avec un écran.

5 Conclusion

La réalisation du *roller coaster* dans l'environnement de la CAVE a permis de recréer des sensations fortes. Bien que n'étant pas du même ordre que les sensations éprouvées dans un *roller coaster* réel, la vitesse, le vertige et l'étourdissement sont au rendez-vous en immersion. Le joystick 3D permet une édition simple et efficace du *roller coaster*. Il est par contre possible d'envisager une édition 3D efficace sur un ordinateur personnel équipé d'un joystick 3D.