

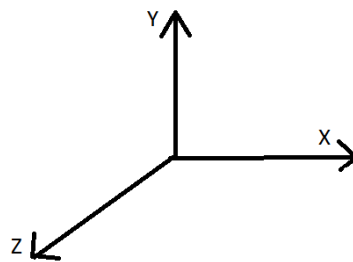
Rapport Projet OpenGL

I. Environnement de développement

Nous avons fait le choix de reprendre le code du projet depuis le début. Pour cela, nous avons utilisé CodeBlocks 10.05 tel que configuré en cours.

II. Graphe

Le repère utilisé pour tout le projet est tel que :



Le graphe utilisé est celui qui nous a été fourni dans l'API.

L'animation de la scène est totalement dissociée du graphe : on peut remplacer le graphe par un autre sans altérer le déplacement des trains. Cependant, le positionnement des éléments du décor (arbres, maisons) et de la surface a été pré-calculé dépendant du graphe et ne seront donc pas forcément dessinés comme il faut.

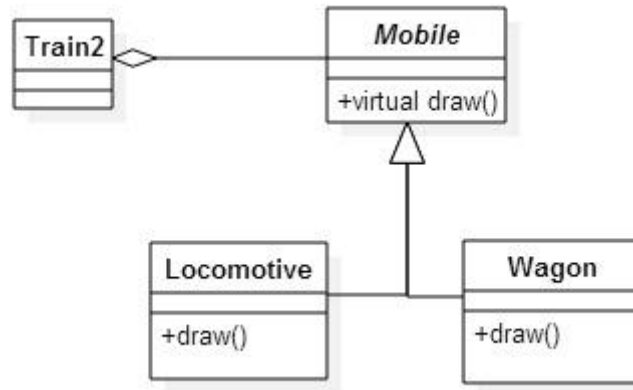
III. Trains

Les trains sont composés d'une locomotive, qui est en fait constituée de deux bonshommes de neige. Cependant, le déplacement du second bonhomme de neige sur le graphe n'est pas optimal avec ce modèle.

On tire au sort les arcs et le sommet sur lesquels on positionne les trains initialement.

La flotte permet de gérer tous les trains. Lorsque l'on veut agir sur le déplacement des trains ou les dessiner, on utilise les fonctions de Flotte.cpp.

C'est pour cela que nous avons tenté de faire un modèle de données tel que :



Cependant, nous n'avons pas réussi à le mettre en place car le polymorphisme a posé problème : le compilateur ne comprenait pas que la fonction `draw()` de `Mobile` n'était pas définie puisque abstraite, et donc définie dans les classes-mères

Déplacement des trains

Le déplacement des trains suit la topologie du graphe : ils vont du sommet initial au sommet final. L'arc suivant est choisi aléatoirement parmi les arcs sortants du sommet final de l'arc précédent.

La vitesse de déplacement des trains est modifiable dans `Flotte.cpp`. Le déplacement des trains, lui, est géré dans la fonction `move()` de `Train.cpp`. La direction est recalculée à chaque fois que le train approche d'un point (sommet ou point annexe). On a :

$$v = \sqrt{dx^2 + dy^2 + dz^2}$$

Le déplacement différentiel est calculé par :

$$\begin{cases} dy = ay * dx \\ dz = az * dx \end{cases}$$

Avec

$$ay = \frac{yp-ys}{xp-xs} \quad az = \frac{zp-zs}{xp-xs}$$

IV. Caméra

Le changement de caméra a été implémenté sur la touche « c ». Les caméras disponibles sont :

- La vue de dessus
- Une caméra sur chaque train (touche « n » pour changer de train)
- 2 caméras libres

V. Paysage

Surface interpolée

Pour obtenir la surface interpolée, nous avons utilisé la méthode d'interpolation linéaire à partir de la triangulation de Delaunay. Les points utilisés pour cela sont les sommets et les points annexes du graphe.

Éléments du décor

On trouve dans le décor deux types d'arbres : des feuillus, des conifères, des maisons et des nuages.

Nous avons géré l'emplacement de chaque objet de façon à ce qu'ils ne soient pas sur les voies.

Texture

Des textures de couleur ont été appliquées aux éléments du décor précédemment cités ainsi qu'au sol.

VI. Améliorations possibles

Si nous avons pu mettre en place le modèle de données comme souhaité, le déplacement du wagon derrière la locomotive aurait mieux suivi le graphe.

Les collisions entre les trains auraient pu être gérées.

Pour ce qui est de la caméra qui suit les trains, la rotation autour de l'axe z n'est pas gérée. On le remarque notamment dans la montée...