

Compte-rendu Projet

Synthèse d'images

Arbres de scènes en résolution adaptative

COMMANDES

Commande de compilation : (dans le même répertoire du Makefile)

make → Compile les fichiers sources ainsi que les scripts.
make clean → Supprime le dossier « obj » ou ce situe tous les fichier .o suite a la compilation.
make cleanall → Supprime le dossier « obj » ou ce situe tous les fichier .o suite a la compilation ainsi que les fichier d'exécution des scripts.

Commande d'exécution :

./<nom_script> → « nom_script » correspond au nom du fichier du script compilé par le Makefile.

AJOUTER UN SCRIPT

Pour pouvoir ajouter un script au projet il faut arriver à le faire compiler. Pour cela il faut ajouter le fichier a l'intérieur du Makefile.

Il y a trois choses a ajouter :

Premièrement, ajouter une ligne dédié au script à l'intérieur de la section « all » du Makefile.

« @make --no-print-directory <nom_script> »

Ensuite, créer une section « <nom_script> : \$(struct) \$(shape) » en dessous de la section « all » du Makefile.

Pour finir, il faut ajouter le nom du nouveau script a l'intérieur de la section « cleanall ».

Lieux :

Le Makefile est configuré pour que les scripts soit dans le fichier « src » mais cela peut être modifié si jamais la librairie graphique est ailleurs.

DESCRIPTION DU PROJET

Base :

Le projet possède 5 formes de bases possèdent leurs surfaces ainsi que les vecteurs normaux : Cube, Sphère, Cylindre, Tore, Cône. Toutes fichiers de formes sont dans « src/shape ».

Le nombre de point maximale des différentes formes sont définies dans le fichier « include/structure/Shape.h ». Le projet possède donc des formes qui possède 128 méridiens, 128 parallèles et 64 cercle sur la base (NBM, NBP, NBC).

Chaque formes de bases possède un affichage par faces (GL_QUADS) et par points (GL_POINTS) adapté a l'échantillonnage. L'affichage possède un raccordement adaptatif en fonction de la taille de la forme ainsi qu'une adaptation aux homothéties.

Arbre :

Le projet possède un arbre de scène avec des transformations géométriques permettant de construire un arbre puis de l'afficher en 3D sur la fenêtre d'affichage.

Pour ce qui est de la construction de l'arbre tout ce passe dans le fichier « src/structure/Tree.c » qui liste toutes les fonctions de l'allocation jusqu'à la créations de différents nœuds.

Il y a plusieurs fonction pour créer un nœud :

- creat_node_empty → Créer un nœud vide.
- creat_node_shape → Créer un nœud avec une forme.
- creat_node_property → Créer un nœud avec tous les propriétés.
- creat_node → Créer un nœud avec tout.

Il est possible d'ajouter n'importe quel éléments avec les fonction « add.. » (on ne peut ajouter un enfant au nœud unique avec une fonction add).

Toutes les propriétés d'un nœuds sont : Une matrice, une couleur, un matériaux, un scale factor.

/!\ On considère qu'un propriété est null si jamais elle possède des valeurs négative car elles sont susceptible de causer problème.

Concernant l'affichage de l'arbre tout ce passe dans le fichier « src/structure/Display.c » qui liste toutes les fonctions qui permette l'affichage totale d'un arbre.

Il y a deux fonction d'affichage :

- display_tree_faces → Affiche les formes avec « draw_faces ».
- display_tree_points → Affiche les formes avec « draw_points ».

L'affichage ce fait ensuite grâce a une fonction récursif qui permet la transmission des informations du père a un fils.

Si le nœud possède un frère après lui dans la liste il lui donne les informations du père et attend qu'il est terminé d'afficher pour commencer. Une fois terminé il distribue ses informations a son première fils qui continuera le cycle jusqu'à ce que tout le monde est été affiché.

C'est ici que ce passe la transmission d'information d'un père a son fils si jamais l'une des propriété est considéré comme null. Le scale factor s'adapte en fonction de la matrice.

Caméra :

Le calcul de la distance d'un entre un objet et la caméra ce fait dans la fonction d'affichage et est transmit grâce aux fonctions d'affichage qui adapte leur échantillonnage par rapport à cette distance.