IMN529 **Synthèse d'images**

Travail pratique #2
Montagne fractale

But:

Implémenter l'affichage d'une montagne fractale ainsi que l'intersection de celle-ci avec un rayon.

Directives:

Dans un fichier « scene.scn », le format d'un atome fractal est le suivant :

FR $n\ h_{min}\ h_{max}$
$x_1\ y_1\ z_1$
$x_2\ y_2\ z_2$
$x_3\ y_3\ z_3$

La valeur *n* spécifie le nombre de subdivision de la montagne fractale lors de l'intersection avec un rayon. Les valeurs $h_{min}\ h_{max}$ spécifient l'intervalle de la perturbation de la hauteur du fractal au niveau 1. La perturbation se fera dans la direction de l'axe des X (dans les coordonnées locales du fractal). Finalement, les valeurs $x_i\ y_i\ z_i$ spécifient la base triangulaire du fractal.

Dans un premier temps, vous devez compléter la méthode **Fractal ::Afficher**, qui est à la fin du fichier **aff.cpp**. Pour afficher un triangle, utilisez la procédure **Affiche_Trigone** (qui utilise OpenGL) qui est dans le fichier **aff.cpp**. Vous pouvez faire varier le nombre de subdivision du fractal, affiché avec ÒpenGL, dans la fenêtre « settings ». Regardez le fichier **fractal.h** afin de connaître les méthodes qui sont déjà définies sur un fractal.

Dans un deuxième temps, complétez la méthode **Fractal ::inter** qui est dans le fichier **fractal.cpp**. Afin de calculer l'intersection d'un rayon avec un triangle, utilisez la procédure **Inter_Triangle** qui est aussi dans le fichier **fractal.cpp**. Après l'appel de la méthode **Fractal::inter**, le membre **normal** (un vecteur) de la classe Fractal devra contenir la valeur du vecteur normal du triangle le plus proche du point **p** parmi les triangles intersectés par le rayon. Important : ne transformez pas le vecteur normal, il doit être dans les coordonnées locales du fractal. Vous pouvez vérifier que l'intersection d'un rayon avec un fractal fonctionne bien en cliquant sur l'image de celui-ci avec le bouton gauche de la souris. Pour faire du tracé de rayons, copiez le fichier **rayons.cpp** que vous avez fait lors du tp1 dans le répertoire. Je vous conseille au début d'enlever le calcul des ombres afin d'accélérer le traitement. Pour un même niveau de subdivision, l'image sans ombre obtenue avec le tracé de rayons, devra être pratiquement identique à l'image affichée en OpenGL. Lorsque la méthode **Fractal ::inter** fonctionnera pour des

niveaux de subdivision peu élevés (0, 1, 2 ou 3), vous devrez ajouter le concept de volume englobant. Pour ce faire, utilisez la procédure **Inter_Cube** dans le fichier **fractal.cpp**. Avec des volumes englobants, vous devriez pouvoir faire du tracé de rayons avec des fractals subdivisés 15 fois (cela sera un peu long par contre et compiler en mode « release »).

Remise : expédiez les fichiers **fractal.cpp, aff.cpp** par turnin web http://opus.dinf.usherbrooke.ca/ au plus tard le **1 novembre à 23 h 59**. Au début du fichier **fractal.cpp**, inscrivez vos noms et C.I.P.