TP Image

Synthèse de paysage

TP à réaliser en deux semaines.
Une évaluation par binôme sera organisée lors de la dernière séance de TP entre 16h et 18h.

I. Objectif

On désire tester divers algorithmes liés à la représentation graphique de scènes 3D. Il est clair que seules des primitives 2D pourront être utilisées, par exemple tracé de segments et de points en 2D, coloriage d'un polygone 2D … L'objectif est ici de générer un paysage réaliste à partir d'une description de départ très grossière (voir figure 1).

On commencera par créer les grandes lignes de ce terrain en se donnant un maillage (tableau NxM de points) avec les altitudes correspondantes. Un exemple de terrain vous est fourni dans le fichier **`terrain.m`**.
On affinera ensuite la définition de ce terrain par interpolation fractale (algorithme Diamond-Square), ce qui conduira à une augmentation du nombre de points dans le maillage.
Pour finir, on fera évoluer la couleur du terrain (en fonction de la position, de l'altitude, de l'orientation des zones …) afin de donner une impression visuelle plus réaliste.

On désire représenter en deux dimensions le terrain en respectant la perspective. On procèdera de la façon suivante.

1. On pourra commencer par représenter le terrain en utilisant la fonction **`surf`** de Matlab.
2. Pour construire une représentation 2D plus réaliste, on se donnera ensuite la position de l'observateur, la direction de son regard ainsi que la direction de son « chapeau ». A partir de ces informations, on va dessiner ce que voit cet observateur, par changement de repère.
3. Le rendu se fera à partir d'une décomposition du maillage en un ensemble de triangles avec élimination des parties cachées. Pour cela, deux algorithmes sont possibles :
   - Algorithme du peintre (le plus simple): on trie les triangles du plus éloigné au plus proche et on les affiche dans cet ordre.
   - Algorithme à base d'un Z-Buffer (plus ambitieux, mais permet de traiter les parties cachées et les ombres portées): on commence par construire un buffer identique à l'image 2D de rendu mais contenant, en chaque position, la distance au triangle visible.
4. On refera le même rendu en incluant cette fois la transformation perspective.

II. Quelques raffinements

Voila quelques raffinements possibles, sans ordre de préférence :

- On modifiera le paysage en positionnant un barrage qui, pour simplifier, pourra être supposé parfaitement plan et situé en bout de la vallée (*j*=1 pour le tableau représentant le

terrain). Son altitude max devrait être de 595m pour permettre une mise en eau atteignant 590m.

- On ajoutera les triangles qui représentent le niveau de l'eau. On désire représenter la surface de l'eau par une texture fractale. On subdivisera les triangles pour obtenir des triangles suffisamment petits et on appliquera une variation aléatoire de la teinte.
- On pourra rendre la couleur des différentes zones de l'image plus réaliste en tenant compte de la direction de l'illumination par le modèle de Phong (parties ambiante et diffuse).
- On pourra distinguer les zones d'eau, de prairie, de rocher (pour des pentes trop fortes) ou de neige.
- On pourra prévoir l'élimination des parties hors de la pyramide de visualisation, dans le cas où l'on choisit de ne reconstruire qu'une partie du paysage.
- On pourra ajouter des bords latéraux (jupe) : le terrain est le sommet d'un cube. Cela permet de visualiser plus nettement les effets de perspective.
- On pourra enfin également agrémenter le paysage d'arbres, d'une rivière ...

Une certaine liberté est accordée sur ce que vous réaliserez. Cependant l'affichage en perspective est nécessaire.

Figure 1. Un exemple de réalisation

Annexe : Calcul des transformations pour les perspectives

Soient deux points
P_O : Origine du plan de projection
P_R : le point regardé (direction du regard), qui donne la direction *w*.
On définira un troisième point à partir des deux précédents :
P_C : Position de la caméra sur la droite portée par le vecteur *w*, à une distance *d* de P_O.

Il faut opérer les transformations suivantes :

1. Translation T pour amener l'origine O en P_O
2. Rotation R de façon à passer de (x,y,z) à (u,v,w).
   (u,v,w) est un repère direct avec :
   - w est le vecteur normé de la direction P_RP_O
   - u est le vecteur normé horizontal orthogonal à w et orienté pour former avec v un repère direct
   - v est le vecteur normé orthogonal à u et w et orienté vers le haut (direction du chapeau de l'observateur)

   R se construit alors en mettant en colonne les coordonnées de u,v et w exprimées dans la base (x,y,z). On y ajoutera ce qu'il faut pour obtenir une matrice de rotation en coordonnées homogènes.
3. Transformation perspective P et projection sur le plan 2D (Po, u, v) pour obtenir l'image du point de vue de l'observateur/caméra situé en P_C. Vous réfléchirez en particulier aux notions et relations entre « focale » , « zoom » et « impression de perspective ».

Une fois ces matrices confectionnées, il s'agit d'appliquer, sur tous les points Pt, la transformation résultante P.inv(R).inv(T).Pt avec Pt vecteur colonne en coordonnées homogènes.
La gestion des coordonnées des points peut se faire en colonne ou en ligne. Le passage d'une notation à l'autre implique de transposer les relations.

Compléments :

- la fonction Matlab **`cross`** permet de calculer le produit vectoriel entre vecteurs
- la fonction Matlab **`fill`** permet de desiner des triangles
- la fonction Matlab **`sortrows`** permet de trier des tableaux de valeurs par lignes selon l'index d'une colonne
- la fonction Matlab **`axis`** permet de régler la dimension de l'image projetée dans le plan (Po, u, v).