

Maillages

M2 rpi ICAO, UJF-Grenoble

1 Prise en main et compréhension

Copiez le programme `main.cpp` ainsi que les fichiers `argumentParser.h`, `matrix.h`, `traqueboule.h` et `Makefile`. Compilez-le et exécutez-le. La scène représente une fonction de Rosenbrock sur l'intervalle $[-1; 1] \times [-1; 1]$, maillée en 3D.

Utilisez l'option `-h` pour de l'aide sur la ligne de commande. Essayez les différentes options (pour l'instant, seule la première méthode de tracé est implémentée). Vous pouvez déplacer le point de vue avec la souris. Pour sortir, tapez sur la touche `Escape`.

2 Affichage des polygones

question 1 *Pourquoi les triangles sont-ils visibles ? Que faut-il changer pour obtenir des couleurs plus continues ?*

question 2 *En regardant par en-dessous, on voit ne voit que les arêtes des triangles. Expliquez quelle commande en est responsable.*

question 3 *Observez l'effet produit en décommentant les lignes concernant le "culling". Que font-elles ?*

3 Précalculs

Nous nous intéressons à la méthode `tabuler_la_fonction()` qui échantillonne la fonction et affecte des coordonnées et des couleurs.

question 4 *Quelle est la dimension des tableaux utilisés pour mémoriser les valeurs ? À quoi sert la méthode `ind(int,int,int)` ? Faites un schéma pour expliquer sa formule.*

4 Triangle_Strip

La méthode 1 consiste à afficher le même maillage sous forme de `Triangle.Strip`.

question 5 *Complétez la procédure `afficher_par_triangle_strips()` pour que le tracé s'effectue.*

5 Vertex Arrays

La méthode 2 consiste à utiliser des vertex array pour profiter de l'accès indexé. Elle nécessite de remplir le tableau `point_numero` puis d'utiliser la méthode `glDrawElements` qui trace en lisant dans les tableaux de données. Deux paramètres manquent. Le premier est le nombre de points de chaque bande. Le deuxième est l'adresse du début de la liste d'indices appropriée.

question 6 *Complétez le programme pour utiliser un vertex array.*

6 Performances

Nous désirons comparer approximativement les performances des différentes méthodes de tracé. Nous y en ajoutons une, les `display list`, qui permettent en principe de précalculer un maximum de choses pour optimiser. Pour évaluer la vitesse de réaffichage nous regardons la vitesse de rotation de la scène, pour un grand nombre de polygones.

question 7 *Quelle est la méthode la plus rapide sur votre machine ?*

Pour estimer l'importance relative des calculs et des transferts de données dans les performances, ralentissez la méthode la plus rapide en lui faisant refaire tous les précalculs à chaque affichage.

question 8 *Qu'observez-vous, et qu'en concluez-vous ?*