Stage C++

Master 2 informatique – Université Bordeaux I

1 Présentation du projet C++

L'objectif de ce projet est d'écrire en C++ un moteur de simulation de particules.

2 Gestion des particules

Une particule est un petit élément physique qui se déplace dans un espace à trois dimensions. Sa trajectoire est calculée en fonction des forces qui s'exercent sur elle. Dans un premier temps nous ne prendrons en compte que l'effet de la pesanteur. Puis nous ajouterons au modèle les forces de frottements.

2.1 Rappel simple de la mécanique du point

- Vitesse : $\overrightarrow{v} = \frac{d\overrightarrow{OM}}{dt}$.
- Accélération : $\overrightarrow{a} = \frac{d\overrightarrow{v}}{dt} = \frac{d^2\overrightarrow{OM}}{dt^2}$
- Relation fondamentale de la dynamique : $\sum \overrightarrow{F} = m\overrightarrow{a}$
- Sur la terre l'accélération de la pesanteur G est égale à $9.81m.s^{-2}$.
- Poids : $\overrightarrow{P} = m.\overrightarrow{g}$ (m est la masse)
- Force de frottement visqueux : $\overrightarrow{F_v} = -\lambda.\overrightarrow{V}$, (pour les faibles vitesses)
- Dans le cas d'une sphère on a $\lambda = 6\pi\eta r$ avec :
  - η : coefficient de viscosité (dans l'air $\eta = 2.10^{-5}Kg.m^{-1}.s^{-1}$ dans l'huile $0.36Kg.m^{-1}.s^{-1}$).
  - r : Rayon de la sphère.

3 Visualisation 3D des particules

Afin de visualiser l'évolution du système de particules il nous faudra aussi implementer un moteur d'affichage de particules en trois dimensions. Ce programme devra aussi permettre d'afficher une grille en trois dimensions qui servira de sol dans le monde où les particules évolueront.

3.1 Coordonnées homogènes

En infographie, les points sont représentés par un vecteur en 4 dimensions. Les trois premières coordonnées représentent la position 3D du point et la quatrième un rapport de mise à l'échelle. Dans ce cas, on dit que les vecteurs sont en coordonnées homogènes. Cette représentation permet de simplifier les opérations sur les points (rotation, translation, mise à l'échelle). En effet, elle permet d'effectuer par une seule multiplication de matrices ces trois opérations simultanément.

3.2 Passage coordonnées homogènes / cartésiennes

Soit un vecteur 3D $[x, y, z]$ ses coordonnées homogènes sont $[x, y, z, 1]$.
Soit un vecteur 3D en coordonnées homogènes $[x, y, z, w]$ ses coordonnées 3D sont $\left[\frac{x}{w}, \frac{y}{w}, \frac{z}{w}\right]$.

3.3 Matrices de transformations

Matrice de translation :

$$T_{xyz} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ dx & dy & dz & 1 \end{bmatrix}$$

Matrice de changement d'échelle :

$$E_{xyz} = \begin{bmatrix} E_x & 0 & 0 & 0 \\ 0 & E_y & 0 & 0 \\ 0 & 0 & E_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matrice de rotation (axe x) :

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) & 0 \\ 0 & -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matrice de rotation (axe z) :

$$R_z = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

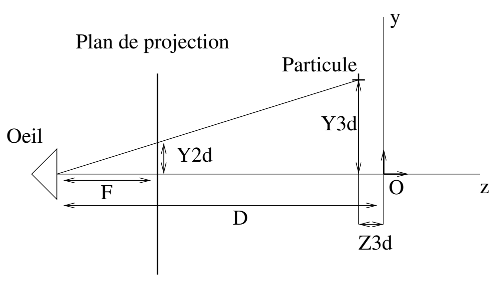
Matrice de rotation (axe y) :

$$R_y = \begin{bmatrix} \cos(\alpha) & 0 & -\sin(\alpha) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3.4 Projections en perspective

Les projections en perspective ne préservent pas la forme de l'objet et sont caractérisées par la convergence des lignes parallèles et une diminution de la taille avec l'éloignement ce qui provoque un raccourcissement des dimensions non uniforme. Seules des lignes parallèles au plan de projection restent parallèles lorsqu'elles sont vues en perspective. Toutes autres parallèles convergent vers un point appelé point de fuite. Le point de fuite d'un ensemble E de droites parallèles est le point où la droite passant par le centre de projection, parallèle à l'ensemble des droites de E, coupe le plan de projection. Les distances et les angles changent lors de la projection sauf pour les faces de l'objet parallèles au plan de projection.

Dans notre cas la solution retenue sera d'utiliser un plan de projection fixe parallèle au plan xOy. Ainsi deux paramètres sont à prendre en compte : la distance de l'observateur au plan de projection noté F comme focale, et la distance de la scène à l'observateur notée D.



Pour obtenir une projection en perspective il suffit donc d'utiliser les résultats suivants :

$$x_{2D} = \frac{x_{3D} * F}{z_{3D} + D} \qquad y_{2D} = \frac{y_{3D} * F}{z_{3D} + D}$$

Matrice de projection :

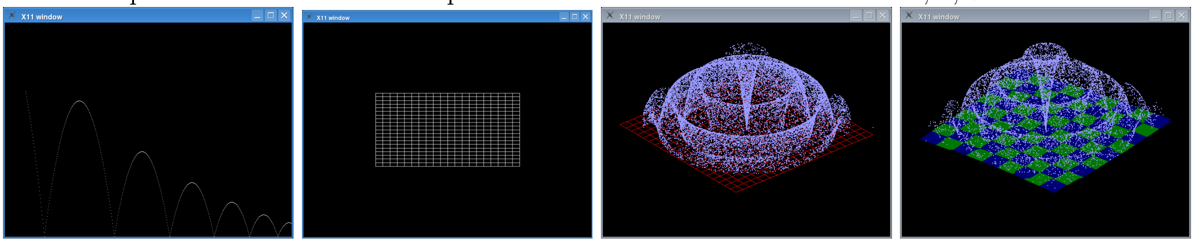
$$R_{proj} = \begin{bmatrix} F & 0 & 0 & 0 \\ 0 & F & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & D \end{bmatrix}$$

4 Quelques sites internet utiles

- Référence de la bibliothèque standard du C++ : `http://www.sgi.com/tech/stl/`
- Site de référence sur C++ : `http://www.cplusplus.com/reference/`
- FAQ C++ : `http://www.parashift.com/c++-faq/`
- Support de cours C++ (Henri Garreta) : `http://h.garreta.free.fr/polys/PolyCpp.pdf`

5 Résultats souhaités

Voici des captures d'écran du résultat que nous voulons obtenir sur les feuilles 1, 2, 3 et 4.



En modifiant légèrement ce code nous pouvons obtenir ce genre d'effet (non demandé mais donnée en extra).