EI3-Semestre 2/IHM-RV/Sujet n°4 (2 séances) ISTIA

Objectif: notion de texture.

Eléments fournis :

- Vidéo du programme à réaliser (lecture avec VLC par exemple)
- Code source du programme de départ : on utilisera la documentation en ligne pour réaliser le travail

Ce TD porte sur l'utilisation des textures (voir figure 1 pour le principe de plaquage d'une texture sur un objet géométrique). On partira du caneva fourni: un carré avec une texture (parmi 3). Les fonctions opengl à utiliser lors de ce TD sont (voir la documentation en ligne) :

- Pour la définition de textures, on s'inspirera de l'exemple fourni, utilisant les fonctions glGenTextures(...), glBindTexture(...), glTexImage2D() et glTexParameterf(...)
- Pour l'utilisation de textures (e.g. activation, plaquage, sélection de la texture courante), on s'inspirera de l'exemple fourni, utilisant les fonctions glBindTexture(...), glTexCoord2f(...), glGetTexEnviv(...), glTexEnvf(...), glEnable(...) et glDisable(...) ; ainsi que les paramètres/valeurs GL_TEXTURE_2D, GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_REPLACE, GL_MODULATE.

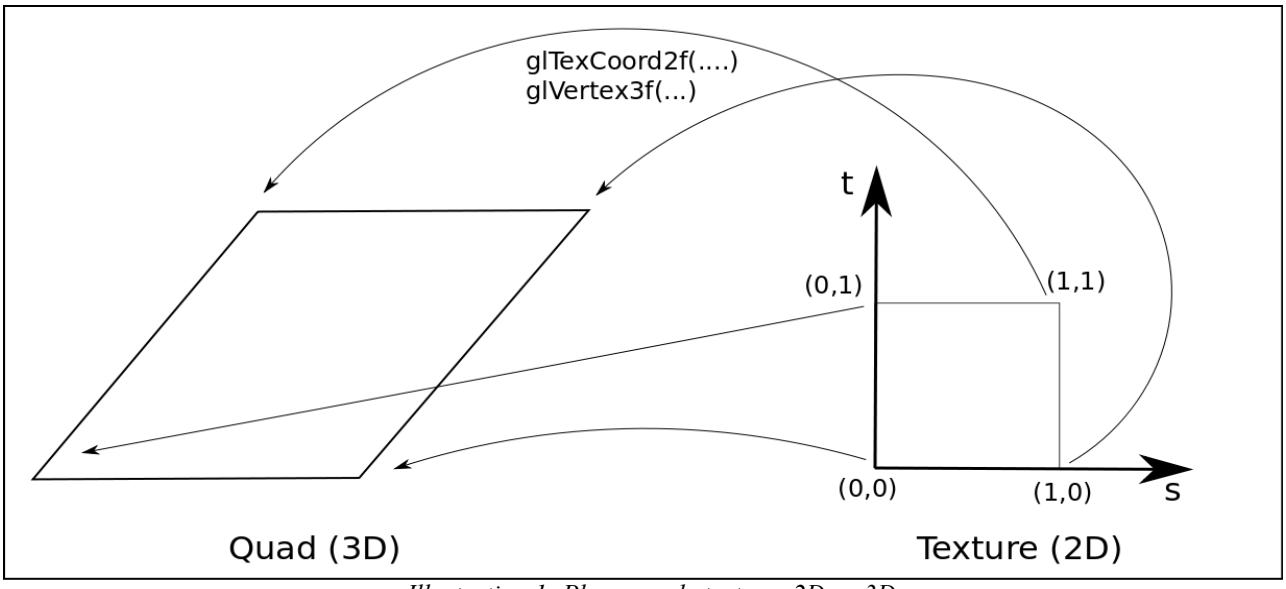

Illustration 1: Plaquage de textures 2D en 3D.

1) Ajouter le sol à la scène, en ajustant le mode de projection en perspective adapté (voir illustration 2). Cette question reprend l'idée du sol en damier considéré lors des sujets précédents. La projection et les caractéristiques de la caméra sont identiques (penser à bien positionner glLookAt(), et glFrustum()). Par contre, le sol est défini par un seul et unique quadrilatère (centré dans le plan (0,x,z) et de taille 12x12), sur lequel est plaqué la texture. La texture est définie comme un tableau de taille 6x6 correctement initialisé (les « cases » seront blanches ou « grises ») afin d'obtenir le rendu attendu. Les bandes blanches matérialisent des routes.

2) Ajouter les fonctionnalités d'activation/désactivation des textures : menu, manipulation des fonctions glEnable(GL_TEXTURE_2D) et glDisable(GL_TEXTURE_2D). Indication : pour consulter une variable d'activation en OpenGL, on peut utiliser glIsEnabled(...).

3) Ajouter la fonctionnalité de mélange des couleurs. L'objectif est ici de permettre ou non que la couleur de la primitive graphique (glColor3f(..)) soit ou non multipliée à la valeur des textels (éléments/points de texture).
Il s'agira d'ajouter le menu approprié et de manipuler les fonctions :
- glTexEnvf(...) , avec les paramètres approprié s :
  → glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_REPLACE) : ignore la couleur affectée à la primitive graphique.
  → glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE) : déclare que la couleur de la texture sera modulée (multipliée) par celle de la primitive graphique.
- glGetTexEnviv(...), permettant d'obtenir l'état d'une variable de configuration de l'affichage des textures.

Jean-Baptiste Fasquel

Illustration 2: Illutration du résultat attendu pour la question 1 : sol texturé.

4) Ajout d'un mur texturé : définition d'une seconde texture et plaquage approprié sur un quadrilatère parallèle au plan (0,x,y). La texture sera définie par un tableau de taille 5x5 dont le contenu sera limité à deux valeurs (noir pour le mur, blanc pour les fenêtres).

5) Ajout d'un bâtiment, composé de 4 murs texturés et d'un toit plat non texturé. La taille élémentaire d'un bâtiment est 1x1x1. Pensez à invoquer plusieurs fois la fonctionnalité précédente (mur texturé) pour construire le bâtiment, en appliquant à chaque fois la transformation géométrique appropriée (glTranslate(), glRotate()).

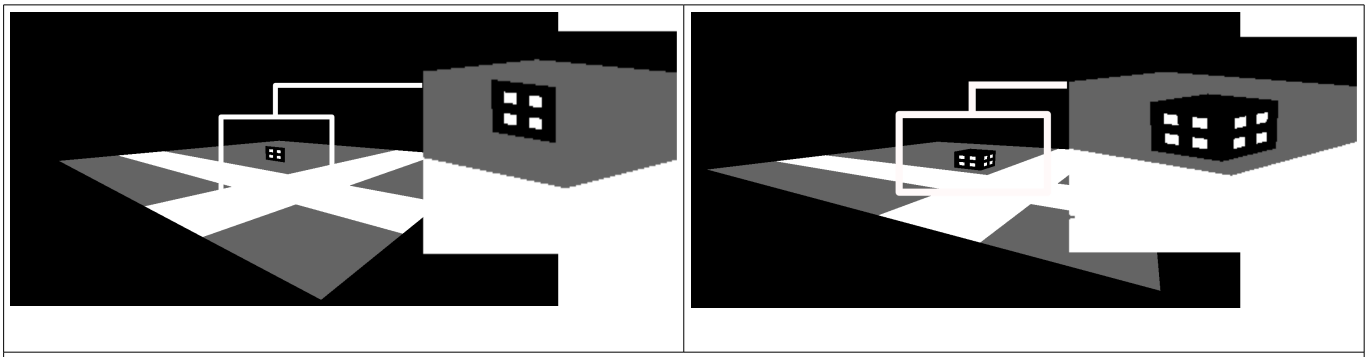

Illustration 3 : gauche : un « mur » texturé (question 4). Droite : un bâtiment composé de 4 murs texturés et d'un toit non texturé (question 5).

6) Ajouter un ensemble de bâtiments, de positions et de tailles variables. Au sujet de la taille, ne pas utiliser la transformation glScalef(), qui étirerait les textures : penser à assembler des bâtiment élémentaires pour construire des bâtiments plus haut (bâtiments empilés) ou plus longs (bâtiments accolés).