

Objets texturés

Familiarisation avec le code

Examinez le code dans les fichiers `cube.js` et `texture.html` et visualisez le résultat.

Cube.js contient une classe permettant d'initialiser et de dessiner un cube. Observez les sommets et les coordonnées de texture. Comment a-t-on fait pour plaquer la texture sur chacune des faces du cube ? Combien de sommets ont été utilisés pour créer le cube ? Combien de triangles ? Pensez-vous que l'on aurait pu obtenir la même chose avec le cube contenant 8 attributs (8 sommets / 8 couleurs / etc) que l'on avait utilisé lors du TP précédent ? Avec quelles coordonnées ?

Texture.html contient le reste du code, équivalent à ce que l'on a déjà vu. Observez (et comprenez) le code de chargement d'une texture, à quel moment elle est initialisée et envoyée à la carte graphique, comment on y accède dans le shader, etc.

Filtrage

Repérez la ligne à laquelle l'image contenant la texture est choisie. Modifiez la texture avec une image de votre choix. Attention : pour la suite, il est important que l'image que vous choisirez ait une taille en puissance de 2 (redimensionnez votre image si nécessaire avec un logiciel de dessin quelconque).

Repérez les lignes que déterminent comment la texture est filtrée (avec `gl.texParameteri`). Testez le mode de filtrage `gl.LINEAR` pour la minification et la magnification. Comparez avec le résultat précédent.

Ajoutez ensuite la ligne `gl.generateMipmap(gl.TEXTURE_2D);` pour générer automatiquement les mipmaps associés à la texture. Il faudra alors utiliser une mode de filtrage adapté aux mipmaps pour la minification, comme `gl.LINEAR_MIPMAP_LINEAR`. Testez, comparez avec les rendus précédents. Quelles sont les différences entre les modes `gl.NEAREST_MIPMAP_LINEAR` et `gl.LINEAR_MIPMAP_NEAREST` ? Que représentent les 2 options ?

Manipulation des coordonnées de texture

Utilisez la texture « `de.jpg` » pour rendre le cube. Observez le résultat. On souhaite que chaque chiffre de la texture constitue une des faces du cube pour former un dé. Dans le fichier `cube.js`, modifiez les coordonnées de texture de chacune des faces pour obtenir le résultat souhaité.

Chargement d'objets

Il est souvent difficile de fabriquer les objets à la main comme on l'a fait jusqu'à maintenant. La plupart du temps, les objets sont modélisés dans un logiciel spécialisé puis exportés dans un format adéquat pour être intégré dans le moteur de rendu. Le format « json » est couramment utilisé pour les applications web : <http://www.json.org/>. Observez la structure du fichier « `singe.json` ». Il contient les tableaux d'attributs indexés à des positions particulières. Un objet a d'abord été modélisé dans blender (`singe.blend`) puis exporté au format « ply » (`singe.ply`) pour enfin être convertie en « json » sur le site suivant : <http://www.greentoken.de/onlineconv/>.

En vous basant sur « `cube.js` », complétez le fichier « `model.js` » pour charger et initialiser les données du fichier dans les buffers GPU correspondants (les indices des tableaux utiles sont donnés en début

de fonction). Attention à bien initialiser le nombre d'éléments et leur taille pour chaque tableau. Chargez ensuite le modèle du singe dans texture.html au lieu du cube.

Effets de bordure

Dans le vertex shader, multipliez les coordonnées de textures par 4. Qu'observez-vous ? Testez et comparez les modes de wrapping `gl.REPEAT` et `gl.MIRRORED_REPEAT` pour les 2 dimensions (S et T).

Seconde texture

Créez et initialisez une seconde texture que vous récupérerez dans le fragment shader (de la même manière que la première). Cela implique des ajouts dans `initTexture`, `initShaders`, et dans le fragment shader. Combinez vos 2 textures pour le rendu avec une fonction de votre choix. Vous pouvez éventuellement les combiner en fonction d'un autre élément de la scène comme par exemple la profondeur (qu'il faut calculer dans le vertex shader puis faire passer au fragment avec une variable de type `varying`).

Bonus

Créez votre propre objet dans Blender (initiez-vous avec la doc et les tutoriels en ligne). Ajoutez des coordonnées de texture en mode édition (tab) puis touchez « u ». Exportez le ensuite en fichier « ply » puis convertissez ce fichier en « json » avec le convertisseur en ligne <http://www.greentoken.de/onlineconv/>. Chargez ensuite votre objet dans votre page web.