Antoine Pizon

Informatique Graphique : TP8

Partie I : Wrapping

Il existe plusieurs façons d'incorporer une texture sur une surface en fonction de l'effet que l'on souhaite obtenir, ces différentes options sont :

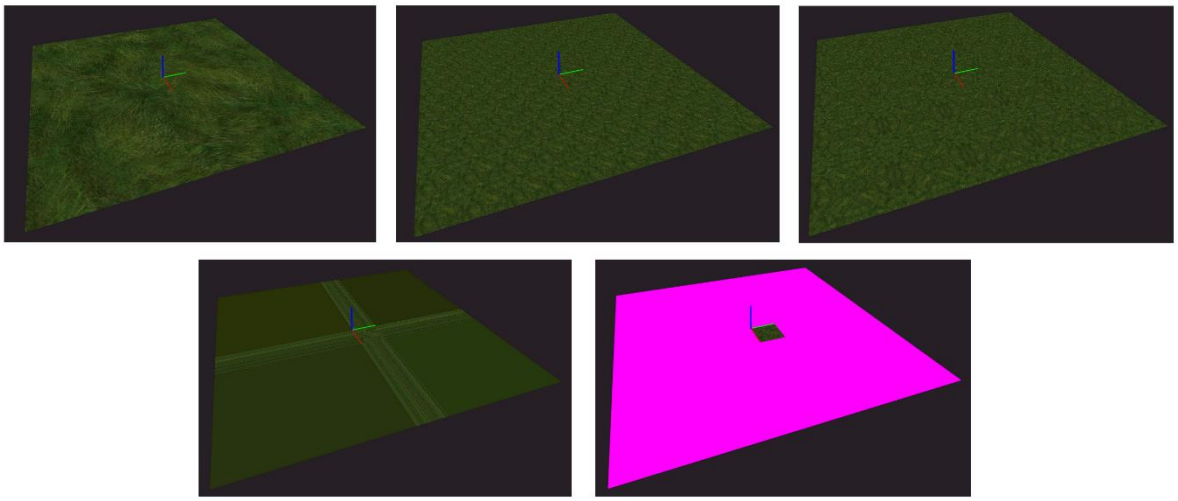
Le mode **sans wrapping** : La texture est accrochée à différents points de la surface, elle est étendue ou diminuée et déformée pour s'aligner sur ces points.

Le mode **REPEAT** : La texture n'est pas déformée, elle garde sa taille et forme d'origine, et est répétée de façon linéaire autant de fois que nécessaire dans les deux directions afin de remplir la surface.

Le mode **MIRRORED_REPEAT** : Même principe que précédemment mais cette fois au lieu d'être répétée à l'identique elle est miroitée.

Le mode **CLAMP_TO_EDGE** : La texture n'est pas déformée, elle garde sa taille et tous les points à l'extérieur prennent la valeur des points aux extrémités de la texture.

Le mode **CLAMP_TO_BORDER** : La texture n'est pas déformée et les points à l'extérieur ne sont pas texturés

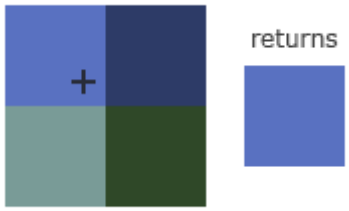


De gauche à droit en commençant par la ligne du haut : mode sans wrapping, REPEAT, MIRRORED_REPEAT, CLAMP_TO_EDGE et CLAMP_TO_BORDER.

Partie II : Filtering

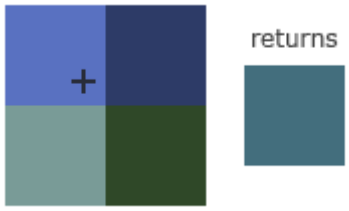
Il existe aussi plusieurs moyens de **filtrer** les textures, c'est-à-dire de choisir la couleur d'un point en fonction de sa position sur la texture.

Le mode **GL_NEAREST** : Le mode par défaut d'OpenGL, la couleur d'un point est choisie en fonction du texel dans lequel il se trouve, peu importe s'il est proche des bords ou non. Ce mode résulte en un rendu plus pixelisé.

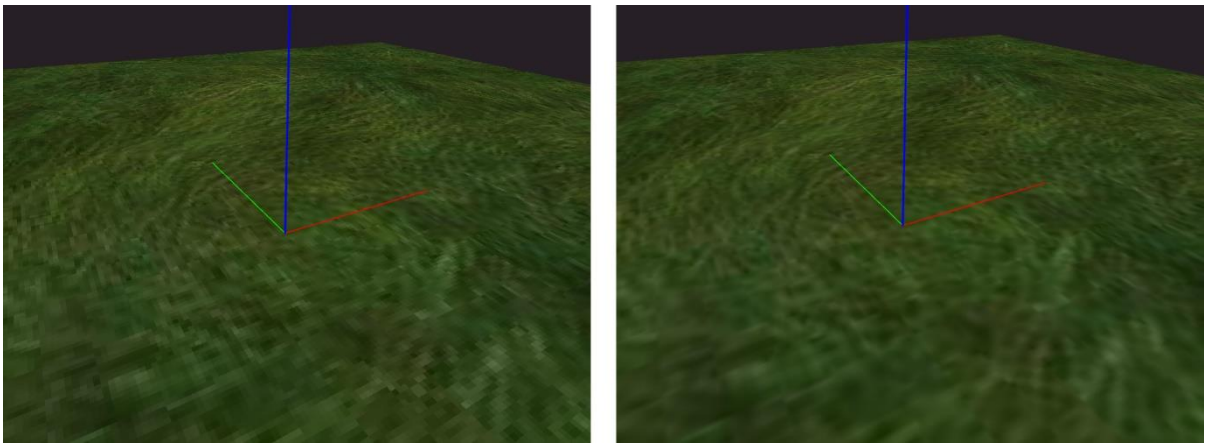


Fonctionnement du mode GL_NEAREST (Source : learnopengl.com)

Le mode **GL_LINEAR** : La couleur du point est une moyenne des différents texels (les pixels composants la texture) voisins à ce point, plus le point est proche du centre d'un texel, plus il sera proche de sa couleur. Ce mode résulte en un rendu plus lissé (et donc parfois plus flou).



Fonctionnement du mode GL_LINEAR (Source : learnopengl.com)



Un zoom sur une texture d'herbe avec à gauche le mode GL_NEAREST, à doite le mode GL_LINEAR

Partie III : MipMapping

Afin d'éviter les problèmes de crénelages qui apparaissent quand on s'éloigne trop d'une surface ou qu'on est à un angle trop aigu par rapport à la normal, on peut générer des **MipMap**, c'est-à-dire des versions de taille différente de nos textures.

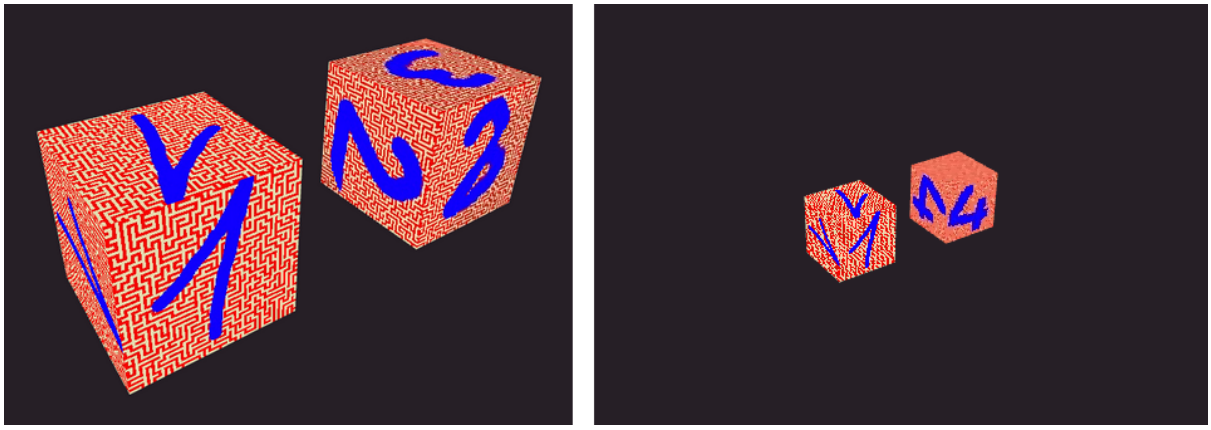
De la même manière que pour les textures, il est aussi possible d'utiliser différents modes de filtrages pour les MipMap. Il existe donc quatre variations du filtrage d'une texture MipMap, chacune correspondant à un filtrage de texture et un filtrage de MipMap : GL_NEAREST_MIPMAP_NEAREST, GL_NEAREST_MIPMAP_LINEAR, GL_LINEAR_MIPMAP_NEAREST, GL_LINEAR_MIPMAP_LINEAR.

Un filtrage **NEAREST** sur les MipMap fait que la MipMap utilisée est celle qui correspond le mieux à la distance/l'angle selon lesquels on observe l'objet.

Un filtrage **LINEAR** sur les MipMap fait que l'on utilisera une moyenne des MipMap pour calculer la texture correspondante.

Dans l'exemple ci-dessous on utilise le mode **GL_NEAREST_MIPMAP_LINEAR** pour le cube de droite (et **GL_NEAREST** pour le cube de gauche qui lui n'est pas MipMappé). On remarque que plus on s'éloigne plus on utilise les MipMaps de petite taille, même chose lorsqu'on est à un angle prononcé avec la normale de la surface.

On remarque aussi que le chiffre présent sur l'une des faces est un mélange de 2 et de 3, c'est grâce au filtrage linéaire qui fait que l'on utilise une moyenne des deux.



Rendu à différentes distances de deux cubes, celui de gauche a une texture classique, celui de droite est MipMappé

Partie IV : Multitexturing

Certains objets nécessitent l'utilisation de **plusieurs textures** pour obtenir le rendu désiré. Pour cela il est possible de charger plusieurs textures différentes et d'appliquer à chacune un coefficient indiquant comment elle se mélange (le coefficient de **blending**). Chacune de ces textures doit être mappée sur la surface de la même manière qu'une texture classique. On calcule ensuite la couleur d'un point en moyennant ces textures avec le coefficient de blending.

Le reste du rendu est réalisé comme pour n'importe quelle texture, les différentes techniques de wrapping, filtering et MipMapping s'appliquent de la même manière.

Sur l'exemple ci-dessous, on crée un cube qui utilise deux textures pour ses surfaces, la première est une texture de caisse en bois, la seconde le logo du jeu-vidéo half-life, qui est transparent sur les parties non colorées.

On remarque qu'on peut voir apparaître la texture de caisse en bois sur les parties en orange du logo grâce au blending des deux textures.

Exemple de cube utilisant deux textures pour ces faces