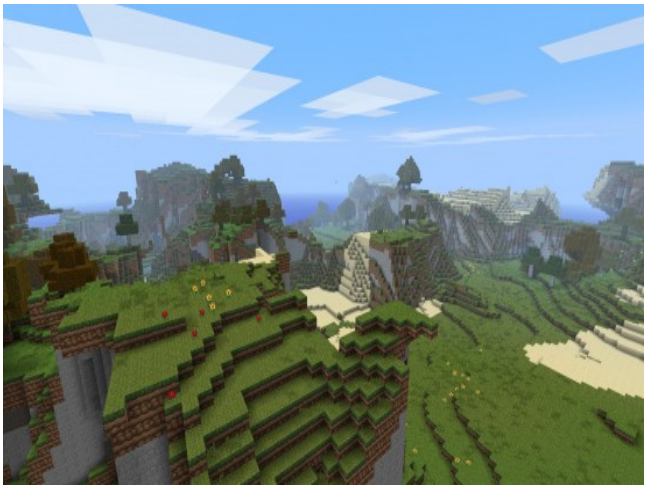
Projet de synthèse d'images

Master Informatique Science de l'Image – 2ème année

Master Craft

L'objectif de ce projet est la création d'un environnement représenté par un ensemble de cubes texturés. Ce projet se réalise par monôme ou binôme.

Le but de ce projet est de réaliser un monde 3D représenté uniquement par des cubes 3D dans l'esprit de « Minecraft ». Il vous permettra de mettre en œuvre un moteur de rendu simple.



1. Le Master Craft : la scène

Vous devrez créer un univers représenté par des cubes. Cet univers se caractérise par une « base rectangulaire » sur laquelle se placent différents cubes représentant le terrain. Sur ce terrain, peuvent se déplacer des personnages (cubiques) et peuvent être posés des éléments de décors (tels des arbres, cubiques eux aussi). La scène est également éclairée par le soleil et dispose d'une « environnement map » qui représente le monde « lointain ». Cet univers a néanmoins une taille finie et on peut atteindre son bord. Le monde aura deux ambiances lumineuses, une de « jour » et une de « nuit ».

1.1. Le terrain

Le terrain est représenté, informatiquement, par une image appelée « height map ». La résolution de cette image (*W*,*H*) définit donc la taille rectangulaire [*W*res_w ; H*res_h*] du terrain à représenter, *res_w* et *res_h* constituant un facteur prédéfini de mise à l'échelle. Chaque pixel de cette image, définie en niveau de gris, représente l'élévation du terrain. Chaque pixel étant codé sur un octet, l'élévation est donc définie entre 0 et 255**res_v*, *res_v* étant là encore un facteur de mise à l'échelle. Cette valeur entre 0 et 255 fixe le nombre de cube à placer à cet endroit, en colonne. Le terrain est donc uniquement constitué de cube.

En complément de cette image, une autre image, de même résolution mais codée cette fois en couleur, indique pour chaque pixel, la nature du terrain à représenter. Au moins 3 types de terrain différent devront être définis : le sable, le gazon, la roche. Vous définirez pour chaque type de terrain, un code couleur correspondant. Par exemple, le sable pourrait être représenté par la couleur (200,155,40). Je vous laisse le choix des couleurs à associer à différents types. A chaque type de terrain est associé, visuellement, une texture différente à appliquer sur les cubes de ce type (lors de la construction de votre application, vous pourrez commencer par utiliser une couleur différente au lieu des textures).

Coté technique, nous vous conseillons d'utiliser le format ppm pour le chargement de ces images. C'est un format simple et non compressé qui permet donc de conserver les bonnes « couleurs ».

1.2. Les éléments de décors

Le terrain est « agrémenté » d'éléments de décors. Ceux-ci sont positionnés dans des pixels spéciaux du terrain, mais l'élément de décor, qui devra être constitué de cubes texturés, peut s'étendre sur plus d'un pixel. Vous définirez au minimum un type d'élément de décor qui sera un arbre.

Afin de positionner ces éléments de décor, vous ajouterez, dans l'image représentant les différents types de terrain, +X au canal rouge de la couleur représentant le terrain, X indiquant le type d'élément de décor à mettre sur ce pixel. Ainsi, si un pixel représentant du sable, codé par le triplet (200,155,40), possède un élément de décors de type 5, alors le pixel sera codé par le triplet (205,155,40).

1.3. Les personnages

Dans cet univers, vous placerez, aléatoirement, des « personnages » (humains, animaux, autres créatures…). Ceux-ci sont au minimum représentés par un cube texturé, sachant que la modélisation peut être plus complexe. Ces personnages devront se déplacer aléatoirement dans l'univers, tout en restant « collés » au sol. Ces déplacements ne sont pas nécessairement orientés sur les quatre axes.

Vous définirez au minimum un personnage, qui se déplacera continûment sans sortir de la carte.

1.4. L'environnement

L'univers présentera deux ambiances lumineuses, une de « jour », caractérisée par une source de lumière directionnelle (comme le soleil) et exploitant le modèle de Lambert (réflexion diffuse), et une de « nuit » qui présentera un éclairage relativement sombre. Pour l'ambiance de nuit, vous êtes libre d'utiliser n'importe quel style de rendu que vous souhaitez (éclairage constant, rendu non photoréaliste …) tant que celui-ci est **original**.

L'environnement constitue le reste de l'univers. Tout d'abord il conviendra de définir un « ciel » pour notre univers 3D. Le ciel sera constituée de 2 « skybox ». Une skybox est un cube texturé qui englobe la scène. Traditionnellement, il est dessiné avant tout autre objet, autour de la caméra et en désactivant l'écriture dans le Z-buffer. Cela permet, ensuite de dessiner tout le reste par dessus. Dans notre cas, il peut également être définie comme un véritable cube au bord du monde. Quelle que soit votre choix, vous implémenterez deux skybox différentes pour les deux ambiances lumineuses de jour et de nuit.

2. L'application Master Craft

2.1. Visualisation et IHM

L'application elle-même devra se représenter sous la forme d'une fenêtre OpenGL qui visualise la scène. Idéalement, cette fenêtre devrait recouvrir l'ensemble de l'écran (donc proposer une application « full screen »). Enfin, l'application permettra de visualiser l'univers suivant le point de vue d'un observateur contrôlable par l'utilisateur avec des commandes de type FPS (vous utiliserez donc une caméra de type « Freefly »).

Les interactions entre l'utilisateur et l'application seront simples. On veut au minimum les interactions suivantes :

- Piloter l'observateur virtuel à l'aide des touches 'z' et 's' pour avancer et reculer, 'q' et 'd' pour se décaler sur la gauche et la droite. L'application permettra de pouvoir regarder, grâce à la souris, tout autour de la scène.
- A l'appui de la touche 'e', passer du mode « jour » au mode « nuit ».
- A l'appui de la touche 'Escape' sortir de l'application,

2.2. Travail à réaliser

Vous devrez réaliser une application en C/C++ utilisant de l'OpenGL 3.3 ou plus (ou OpenGL ES version 3.0). Vous pouvez exploiter les bibliothèques vues en TD (SDL, GLEW, glimac…) ainsi que d'autres bibliothèques, par exemple de chargement de modèle 3D (Assimp) ou d'image. Si vous souhaitez utiliser une autre bibliothèque, merci de vérifier auprès de moi si cela est possible. **Votre**

application devra fonctionner sous linux, et en particulier, tourner sur les ordinateurs de l'université. Vous pouvez utiliser cmake pour la compilation.

Vous devrez ensuite réaliser les spécifications suivantes (dans l'ordre de « difficulté ») :

1. La lecture des deux images définissant l'univers
2. L'affichage du terrain avec au minimum 3 type de terrain
3. Le déplacement de l'observateur virtuel
4. L'affichage des décors avec au moins 1 type de décor
5. L'affichage et l'animation des personnages avec au moins un type de personnage
6. L'affichage de la skybox et des ambiances lumineuses
7. Faire en sorte que les personnages et l'observateur virtuel ne sorte pas du « monde »

De plus, les consignes de rendu seront spécifiées sur l'elearning du cours.

2.3. Bonus supplémentaires

La réalisation des spécifications demandées vous conférera une note honorable (autour de 14). Pour aller plus loin, vous pouvez implémenter les différentes extensions suivantes :

- Types supplémentaires de terrains, décors et personnages
- Déplacement plus intelligent des personnages
- Animation de la lumière directionnelle de « jour »
- Placement d'une lumière attachée à l'observateur virtuel de « nuit »
- Plus compliqué, la possibilité de retirer ou d'ajouter un type de terrain sur une « case » donnée