

M4105 – Moteurs 3D
TP 4

Nous allons voir dans ce TP comment afficher avec OpenSceneGraph un terrain 3D texturé, comment le recouvrir d'une forêt en affichant les arbres au moyen de billboards et comment positionner correctement un objet 3D à sa surface.



1. Créez un nouveau programme.
2. Écrivez une méthode **creation_terrain()** permettant de charger une image en niveaux de gris et de l'utiliser pour créer les hauteurs d'un terrain 3D de type **osg::HeightField**. Cette méthode retournera un **osg::Node***. Dans votre programme principal, vous ferez appel à **creation_terrain()** et vous ajouterez le nœud retourné dans le graphe de votre scène (voir chapitre 2, diapo 98). Vous pourrez utiliser le fichier **terrain.tga** comme image de hauteurs.
3. Ajoutez un matériau au terrain ayant pour ambiant (0.5, 0.5, 0.5, 1.0), pour diffus (0.9, 0.9, 0.9, 1.0), pour spéculaire (0.0, 0.0, 0.0, 1.0) (voir chapitre 2, diapo 28).
4. Modifiez la source de lumière **GL_LIGHT0**, avec pour ambiant (0.5, 0.5, 0.5, 1.0), pour diffus (0.9, 0.9, 0.9, 1.0), pour spéculaire (1.0, 1.0, 1.0, 1.0).
5. Modifiez la méthode **creation_terrain()** de manière à plaquer sur le terrain une image reçue en paramètre (voir chapitre 2, diapo 99). Vous pourrez utiliser le fichier **herbe.tga** comme image de texture.

6. Par défaut, la texture plaquée sur le terrain est étirée sur toute sa surface. Modifiez la méthode **creation_terrain()** de manière à modifier les coordonnées de texture pour que la texture soit répétée plusieurs fois (voir chapitre 2, diapo 100).

7. Écrire une méthode :

```
bool intersection_terrain( float x, float y,
                          osg::Node* terrain,
                          osg::Vec3& position,
                          osg::Vec3& normale)
```

qui calcule la position et la normale d'un point à la surface du terrain étant donné des coordonnées (x,y), en calculant l'intersection d'un segment de droite avec le terrain (voir chapitre 2, diapo 104).

8. Écrire une méthode :

```
osg::Group* creation_foret( osg::Node* terrain,
                            int nb_arbres)
```

Dans cette méthode, vous créerez un `osg::Group*` qui sera retourné à la fin de la méthode. Vous ajouterez à ce groupe **nb_arbres** billboards (voir chapitre 2, diapo 109) dont la position (x,y) sera tirée aléatoirement et dont la position précise à la surface du terrain sera obtenue grâce à la méthode **intersection_terrain()**. Vous pourrez utiliser le fichier **arbre.tga** comme texture de ces billboards. Pour plus de réalisme en ayant des arbres de tailles différentes, tirez aléatoirement la taille du billboard entre 5 et 20.

9. Dans **main()**, faites appel à **creation_foret()** et ajoutez le groupe retourné par la méthode au graphe de scène.

10. Écrire une méthode :

```
osg::Group* creation_foret( osg::Node* terrain,
                            std::string imageForet)
```

Dans cette méthode, vous créerez un `osg::Group*` qui sera retourné à la fin de la méthode. Cette fois, la position des billboards des arbres sera déterminée en fonction d'une image en niveaux de gris passée en paramètre. Quand le pixel d'une image est noir, vous ne placerez pas d'arbre. Si le pixel est blanc, créez un arbre que vous ajouterez au groupe. Pour plus de réalisme en évitant que les arbres ne soient tous alignés sur la matrice de pixels de l'image, ajoutez un peu d'aléatoire à la position (x,y) du billboard.

Dans **main()**, faites appel à cette nouvelle méthode **creation_foret()** plutôt qu'à la précédente et ajoutez le groupe retourné par la méthode au graphe de scène.

11. Téléchargez l'archive **t72-tank.zip** et décompressez la, vous obtiendrez le fichier **t72-tank_des.flt** et un dossier de textures. Placez l'objet 3D **t72-tank_des.flt** sur le terrain. Utilisez la méthode **intersection_terrain()** pour calculer précisément sa position à la surface du terrain (voir chapitre 2, diapo 106), et modifiez son orientation pour suivre celle de la normale du terrain (voir chapitre 2, diapo 107).