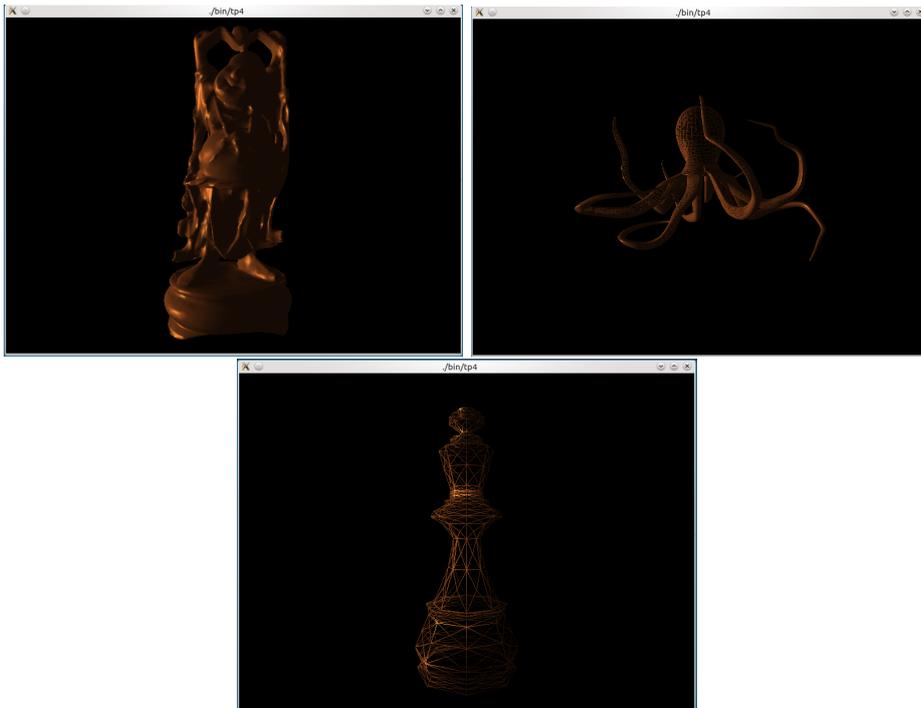


## Afficher un maillage 3D

Ce TP a pour but de comprendre comment charger un fichier représentant un objet 3D par des facettes, et d'afficher cet objet.



## Rappels

Les fichiers de maillage que nous allons utiliser sont au format .off et ils se présentent de la façon suivante :

```
OFF
8 12 0
0.256 0.365 0.569
2.365 2.654 8.145
...
...
3 0 1 2
3 2 5 9
3 6 0 4
...
...
```

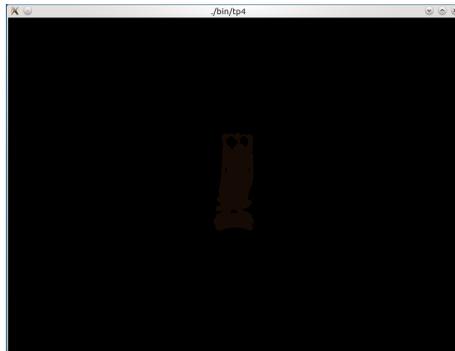
où :

- Tout d'abord, il s'agit de fichiers ASCII.
- Les trois premiers caractères rappellent l'extension du fichier. Ici : OFF
- Le premier entier est le nombre de sommets du maillage. Ici : 8
- Le deuxième entier est le nombre de facettes du maillage. Ici : 12
- Le troisième entier est le nombre d'arêtes et bien souvent cette valeur est à 0 : elle n'a pas été mise à jour lors de la création du fichier.
- Ensuite vient la liste des coordonnées des sommets du maillage sous forme de flottants. Les trois premiers flottants sont les coordonnées x, y, z du premier sommet (numéro 0), les trois suivants sont les coordonnées x, y, z du deuxième sommet (numéro 1) et ainsi de suite.
- Une fois tous les sommets stockés, vient la liste des facettes sous la forme suivante : La première valeur est le nombre de sommets de la facette (pour nous, il n'y aura que des facettes triangulaires, donc cette valeur sera 3). Cette valeur est suivie du numéro des sommets composant la facette. Ici, la première facette triangulaire est composée du sommet numéro 0, du sommet 1 et du sommet 2. La deuxième facette triangulaire est composée des sommets numéro 2, 5 et 9, et ainsi de suite.

## Introduction

- Récupérez le fichier *buddha.off* et vérifiez qu'il est bien conforme au format présenté précédemment.
- Récupérez les fichiers *mesh.h* et *mesh.c*, et observez la structure utilisée pour représenter un maillage.

## Afficher le maillage sans les normales



- Ecrivez la fonction *pMesh creerMesh (char \*nomFich) {...}* qui alloue l'espace mémoire et affecte les champs *nbSommets*, *nbIndex*, *sommets* et *index* à partir du fichier dont le nom est passé en paramètre, et retourne le maillage stocké dans un pointeur de type *pMesh*. Cette fonction permet aussi l'initialisation du matériau. Lors de la création de votre maillage pour le bouddha, affectez lui le matériaux suivant :

```
float laiton_ambient [] = {0.4,0.2,0.08,1.0};  
float laiton_diffuse [] = {0.4,0.2,0.08,1.0};  
float laiton_specular [] = {1.,0.5,0.2,1.0};  
float laiton_shininess = 90.;
```

### Rappels / aides :

```
// déclaration d'une variable de type fichier  
FILE *f;  
// tableau de 3 caractères  
char c[3];  
// compteur  
int i;  
// variable de type pointeur sur une structure de maillage  
pMesh maillage;  
  
...  
  
// ouverture du fichier en lecture  
f = fopen(nomFich, "r");
```

```

...

// allocation du pointeur sur la structure de maillage
maillage = (pMesh)malloc(sizeof(tMesh));

...

// lecture des trois premiers caractères du fichier
fread(c, sizeof(char), 3, f);
// lecture du nombre de sommets
fscanf(f, "%d", &maillage->nbSommets);

...

// allocation mémoire pour le tableau de sommets
maillage->sommets = (float*)malloc(sizeof(float)*3*maillage->nbSommets);

...

// lecture des coordonnées x,y,z des sommets du maillage
for (i=0; i<nbSommets*3; i++) fscanf (f, "%f", maillage->sommets+i);

...

fclose(f); // fermeture du fichier
return maillage; // renvoi du pointeur sur la structure de maillage

```

- Dans votre fichier *main.c*, dans la fonction *main*, créez un maillage à l'aide de la fonction précédente. Une façon simple d'appeler cette fonction de chargement d'un maillage est la suivante : soit la variable globale static *pMesh* *\_mesh* dans votre fichier *main.c*, l'appel de la fonction peut être *\_mesh = creerMesh ("Maillages/buddha.off")* (si le fichier *buddha.off* est dans le répertoire *Maillages*).
- Dans la fonction *display* du fichier *main.c*, appelez la fonction *afficheMesh* qui permet de visualiser le maillage à l'écran. Dans la fonction *main*, après la création de la fenêtre de rendu et l'activation du Z-buffer, appelez la fonction *creerMeshVBO*.
- Observez la position du Buddha dans la scène ainsi que sa taille. En fait, quand on charge un maillage à partir d'un fichier *.off*, on ne sait jamais quelle va être la taille de l'objet, ni si il va être centré. Pour éviter ce problème, dans la fonction *creerMesh (...)*, vous allez calculer le centre de gravité du maillage (la somme des sommets divisée par leur nombre) et vous allez centrer le maillage en (0,0,0) (en soustrayant le centre de gravité à chaque sommet). Ainsi, tous les maillages chargés seront centrés.

Ensuite, vous allez calculer la plus grande coordonnée en valeur absolue (pour la valeur absolue, vous pouvez utiliser la fonction *float fabs (float f)* qui est dans la librairie *math.h*) et vous allez diviser toutes les coordonnées par cette valeur positive. Ainsi, le maillage aura toutes ses coordonnées comprises entre -1 et 1. Tous les maillages chargés auront grossièrement la même taille.

- Le maillage est maintenant affiché avec son centre au niveau du centre de gravité. Ceci n'est pas très pratique si le maillage représente un objet que l'on veut poser sur un autre (par exemple poser le buddha sur le sol). Toujours dans la

fonction *creerMesh (...)*, ajoutez le calcul de la coordonnée en Y la plus petite et stockez la dans le champs *minY* de la structure de maillage. Ajoutez une translation dans la fonction *afficheMesh (...)* pour que le bas de l'objet se trace toujours en  $y=0$  dans le repère objet.

- Ajouter deux champs à votre structure : *Vector4 position* et *float scale*, permettant de positionner le maillage ainsi que de choisir sa taille. Ces champs sont initialisés dans la fonction *creerMesh (... , Vector4 pos, float scale)* et le positionnement et la mise à l'échelle du maillage sont effectués dans la fonction *afficheMesh (...)* (avec les fonctions *glTranslatef* et *glScalef*).

Affichez le bouddha en (4.,-3.,-9.,1.) avec une taille de 4.

## Afficher le maillage avec les normales

- Les normales aux sommets du maillage se calculent de la façon suivante :

Dans la fonction *creerMesh (...)*, après avoir alloué l'espace mémoire pour le tableau normales de la structure de maillage, on initialise les normales en chaque sommet à (0,0,0) : chaque normale a 3 coordonnées x,y,z. Puis, à chaque fois qu'une facette est chargée dans le tableau d'index, on calcule sa normale, on la norme et on l'ajoute à la normale de chaque sommet de la facette. Quand toutes les facettes du fichier ont été parcourues, le tableau des normales est complet et correct.

Il ne reste plus qu'à décommenter les lignes de code nécessaires à la prise en compte du tableau de normales dans les fonctions *afficheMesh (...)* et *creerMeshVBO (...)*.

- Maintenant vous avez tout ce qu'il vous faut pour charger d'autres maillages. Placez à votre convenance les objets suivants dans la scène. Si vous ajoutez des fonctions pour contrôler la taille et la position d'un maillage, vous pourrez utiliser plusieurs fois le même maillage à différentes tailles et différentes positions sans avoir à le recharger, simplement en faisant plusieurs appels de la fonction d'affichage.

