

# TD 3 - Sous le capot

Xavier Décoret

## Résumé

Le but de ce TD est de se familiariser avec les fonctionnalités du framebuffer

- alpha blending
- stencil buffer

## 1 Alpha blending

Le but de cet exercice est d'afficher 3 sphères semi-transparentes et que la gestion de la transparence soit correcte, comme dans la figure 1-a. Vous pouvez vous

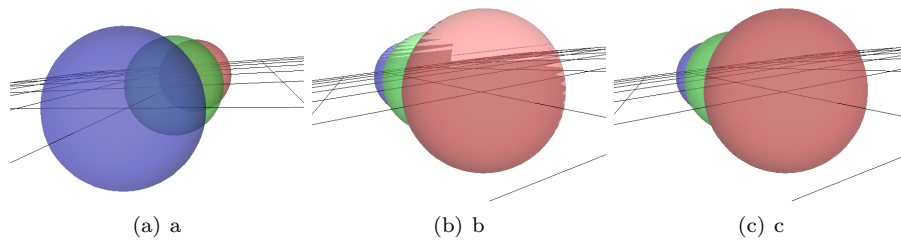


FIG. 1 – Les cas que vous rencontrerez pendant l'exercice (a) affichage correct (b) au sein d'un même objet, l'ordre de rendu des faces pose problème (c) l'ordre de rendu des objets pose problème.

procurer un squelette de programme qui affiche les sphères avec une mauvaise transparence en tapant

```
wget http://artis.imag.fr/~Xavier.Decoret/teaching/inf583/td3/template/A/main.cpp
wget http://artis.imag.fr/~Xavier.Decoret/teaching/inf583/td3/template/A/viewer.cpp
wget http://artis.imag.fr/~Xavier.Decoret/teaching/inf583/td3/template/A/viewer.h
wget http://artis.imag.fr/~Xavier.Decoret/teaching/inf583/td3/template/A/A.pro
```

Comme indice :

- rendez les sphères de la plus lointaine à la plus proche
- utilisez le fait que les sphères sont convexes et pensez au *culling*

## 2 Stencil buffer

Le but de cet exercice est de trouver une méthode pour visualiser l'overdraw d'un rendu en 3D, c'est à dire le nombre de fragments qui tombe dans chaque pixel. La figure 2 montre un exemple.

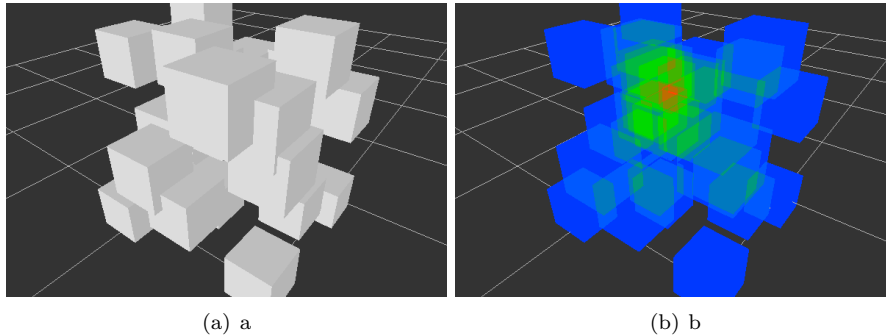


FIG. 2 – (a) Une vue de la scène (b) Une vue de l'*overdraw*

Vous partez d'un squelette de code qui affiche plusieurs théières. Récupérez-le avec :

```
wget http://artis.imag.fr/~Xavier.Decoret/teaching/inf583/td3/template/B/main.cpp
wget http://artis.imag.fr/~Xavier.Decoret/teaching/inf583/td3/template/B/viewer.cpp
wget http://artis.imag.fr/~Xavier.Decoret/teaching/inf583/td3/template/B/viewer.h
wget http://artis.imag.fr/~Xavier.Decoret/teaching/inf583/td3/template/B/B.pro
```

Ensuite, à vous de jouer. Plusieurs stratégies possibles :

- On lit le contenu du stencil avec `glReadPixels()`, on construit soi-même une texture, et on l'affiche sur un quad qui couvre l'écran.
- On affiche une série de quads de couleurs différentes qui couvrent tout l'écran, et on utilise des stencils test astucieux.

Pour afficher facilement un quad qui couvre l'écran, allez voir la fonction `startScreenCoordinatesSystem()` de `libQGLViewer`