



# Mini-projet d'Infographie

## SCOP

42 staff [staff@42.fr](mailto:staff@42.fr)

*Résumé: “So Corporate OpenGL Program”  
“Self Congratulation OpenGL Program”  
Ce mini-projet est une première incursion dans l'utilisation d'OpenGL.*

# Table des matières

<b>I</b>	<b>Préambule</b>	<b>2</b>
<b>II</b>	<b>Sujet</b>	<b>3</b>
II.1	Il n'y a pas de mal à se faire du bien . . . . .	3
II.2	Ce qu'il faut réaliser . . . . .	3
<b>III</b>	<b>Bonus</b>	<b>5</b>
<b>IV</b>	<b>Zoli</b>	<b>6</b>

# Chapitre I

## Préambule



C'est vraiment parce qu'on ne l'avait pas encore faite, celle-là.

# Chapitre II

## Sujet

### II.1 Il n’y a pas de mal à se faire du bien

De temps en temps, il n’est pas pour déplaire de flâter son ego. Nous allons donc réaliser une petite application pour se brosser dans le sens du poil.

C’est l’occasion d’attaquer tout doucement un peu d’OpenGL. Ce projet est organisé pour vous faire mettre les mains dans le cambouis. Il y a donc quelques restrictions dans ce but.

### II.2 Ce qu’il faut réaliser

Votre objectif est de réaliser un petit programme qui affiche un objet 3D conçu au moyen d’un logiciel de modélisation, comme Blender par exemple. L’objet 3D est stocké dans un fichier au format *.obj*. Vous devrez vous charger du parsing à minima pour obtenir le rendu demandé.

Dans une fenêtre, votre objet 3D doit s’afficher, en perspective (c’est à dire que ce qui est loin doit être plus petit), en tournant sur lui-même autour de ce qui semble être son axe de symétrie principal (en gros, le milieu de l’objet quoi...). Par un jeu de couleur, il doit être possible de distinguer les différentes facettes. L’objet peut être déplacé sur les trois axes, dans les deux directions.

Enfin, une texture doit pouvoir être appliquée basiquement sur l’objet lorsqu’on appuie sur une touche dédiée, et la même touche permet de revenir à l’affichage des couleurs. Une transition douce est demandée entre les deux.

Les contraintes techniques sont les suivantes :

- Coder sur les dumps à la norme, en C : l’API première OpenGL est en C, frottez-vous y. Certaines surcouches de certains langages font plein de choses automatiquement, ça serait trop facile.
- Avoir un Makefile normal (tout ce que vous avez l’habitude d’y mettre)
- Utiliser de l’OpenGL MODERNE : en version 4.0 minimum, avec des shaders obligatoirement.

- N'utiliser des bibliothèques externes (autre qu'OpenGL, libm et la lib C) QUE pour la gestion de la fenêtre et des événements. Vous avez le droit d'utiliser votre libft. Dans le doute, utilisez la MinilibX avec son extension OpenGL. Pas de bibliothèque pour loader l'objet 3D, ni pour vous faire les matrices, ou charger les shaders.

Comme il s'agit d'un programme d'auto-congratulation, il est indispensable que vous puissiez présenter en soutenance au moins le logo de 42 fourni avec ce sujet, tournant autour de son axe central (attention, pas autour de l'un des bords), avec des niveaux de gris comme couleurs pour les facettes, et une texture de poneys, licornes ou chatons, au choix.

Lors de la soutenance, d'autres objets 3D seront bien sûr testés.

## Bonus

Quelques bonus sont proposés :

- Gestion correcte des fichiers *.obj* ambigus, non coplanaires, concaves. La th  i  re fournie avec le sujet est en deux versions : la premiere est l'originale, avec des effets de bords   tranges. La seconde est un import-export dans Blender, sans retouche humaine, mais un minimum normalis  e par le logiciel. Il s'agit d'afficher correctement la premiere.
- Application plus subtile de la texture. Elle n'est   tir  e sur aucune des faces.
- Il y a s  urement d'autres bonus que vous pouvez impl  menter.

Bon projet !

# Chapitre IV

## Zoli

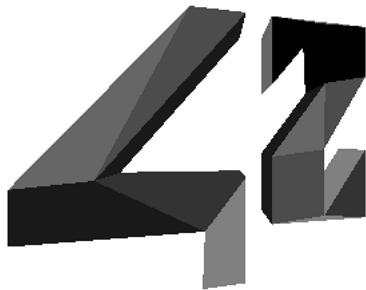


FIGURE IV.1 – Le logo 42 avec des couleurs différentes par facette



FIGURE IV.2 – Vue de dos, avec une texture

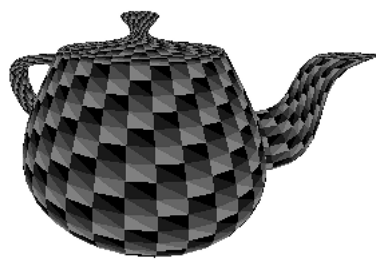


FIGURE IV.3 – La théière