

# **INF2705 Infographie**

## **Spécification des requis du système**

### **Travail pratique 0**

#### ***Les primitives***

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	But . . . . .	2
1.2	Portée . . . . .	2
1.3	Remise . . . . .	2
<b>2</b>	<b>Description globale</b>	<b>3</b>
2.1	But . . . . .	3
2.2	Travail demandé . . . . .	3
2.3	Fichiers fournis . . . . .	3
<b>3</b>	<b>Exigences</b>	<b>4</b>
3.1	Exigences fonctionnelles . . . . .	4
3.2	Exigences non fonctionnelles . . . . .	4
<b>A</b>	<b>Liste des commandes</b>	<b>5</b>
<b>B</b>	<b>Figures supplémentaires</b>	<b>5</b>
<b>C</b>	<b>Apprentissage supplémentaire</b>	<b>6</b>

# 1 Introduction

Ce document décrit les exigences du TP0 « *Les primitives* » du cours INF2705 Infographie.

## 1.1 But

Le but des travaux pratiques est de permettre à l'étudiant de directement appliquer les notions vues en classe.

## 1.2 Portée

Chaque travail pratique permet à l'étudiant d'aborder un sujet spécifique.

## 1.3 Remise

Faites la commande « `make remise` » afin de créer l'archive « **INF2705\_remise\_TPn.zip** » que vous déposerez ensuite dans Moodle. (Moodle ajoute automatiquement vos matricules ou le numéro de votre groupe au nom du fichier remis.)

Ce fichier zip contient le fichier Rapport.txt et tout le code source du TP (`makefile`, `*.h`, `*.cpp`, `*.gls1`, `*.txt`).

## 2 Description globale

### 2.1 But

Le but de ce TP est de permettre à l'étudiant de se familiariser avec les outils de travail mis à sa disposition et de mettre en pratique le dessin en projection orthogonale avec les différentes primitives de la librairie OpenGL.

### 2.2 Travail demandé

La Figure 1 représente un objet à l'aide de sept sommets. On demande de réaliser un programme dessinant cet objet de six façons différentes dans six cadres délimités par des lignes blanches. On utilisera la primitive `GL_LINES` pour les lignes et les primitives suivantes pour l'objet :

<code>GL_QUADS</code>	<code>GL_TRIANGLES</code>	<code>GL_POLYGON</code>
<code>GL_QUAD_STRIP</code>	<code>GL_TRIANGLE_STRIP</code>	<code>GL_TRIANGLE_FAN</code>

### 2.3 Fichiers fournis

La fenêtre initiale a un rapport de 3/2 et il est facile de la subdiviser en six carrés dans lesquels sont dessinés les objets en utilisant chacune des primitives, comme illustré à la Figure 2.

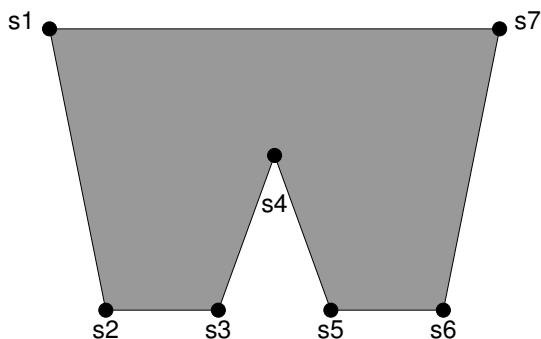


FIGURE 1 – Objet tracé avec les sept sommets donnés

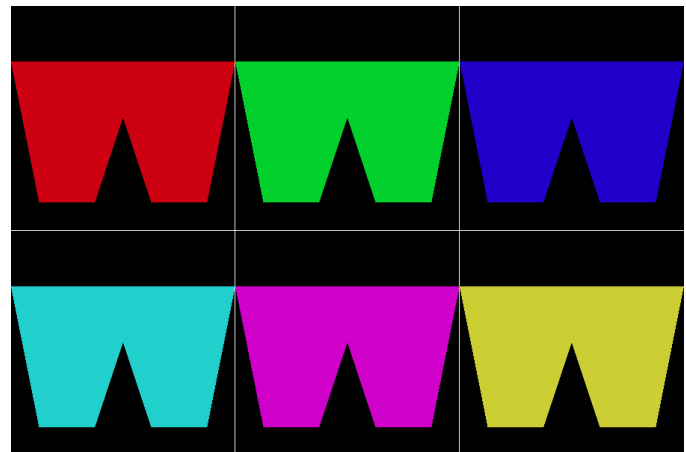


FIGURE 2 – Les six objets demandés

## 3 Exigences

### 3.1 Exigences fonctionnelles

- E1. La fenêtre initiale est subdivisée en 3x2 carrés.
- E2. Chacun des carrés contient un objet.
- E3. Les six carrés sont encadrés par des lignes blanches.
- E4. Chaque objet est dessiné avec une primitive différente, de manière à utiliser les six primitives demandées.
- E5. La fonction `glTranslatef()` est utilisée pour traduire les différents objets à partir du centre de l'écran vers les six carrés.
- E6. (L'utilisateur peut passer en mode fil de fer ou en mode plein.)
- E7. (Le logiciel utilise correctement les touches listées à l'annexe A.)

### 3.2 Exigences non fonctionnelles

Quelques règles à respecter :

- Vous ne devez pas utiliser d'autres sommets que ceux existants.
- Vous ne devez pas remplir ou tracer par-dessus ce qui est déjà rempli ou tracé.
- Vous devez utiliser un nombre minimal de paires `glBegin() ... glEnd()`, mais vous devez bien sûr les répéter lorsque nécessaire.

La totalité du dessin peut se faire dans la fonction `afficherScene()`. Vous pouvez cependant ajouter des fonctions supplémentaires ainsi que d'autres classes et/ou fichiers si vous le jugez nécessaire.

## ANNEXES

### A Liste des commandes

Touche	Description
q	Quitter l'application
x	Activer/désactiver l'affichage des axes
v	Recharger les fichiers des nuanceurs et recréer le programme
g	Permuter l'affichage en fil de fer ou plein

### B Figures supplémentaires

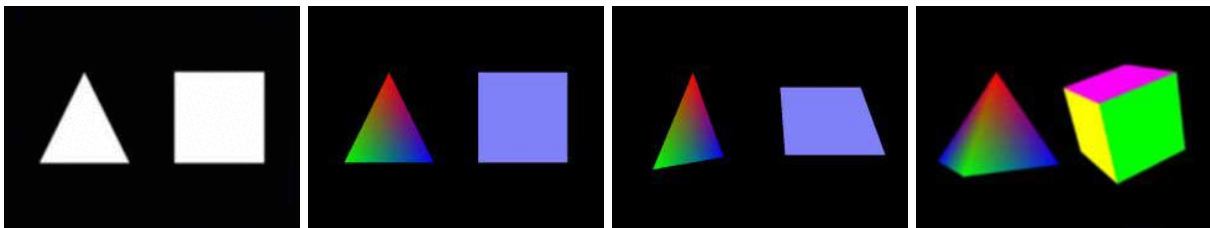


FIGURE 3 – Premiers pas

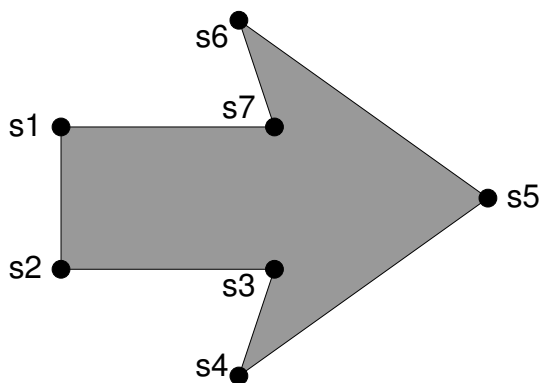


FIGURE 4 – Objet alternatif : une flèche

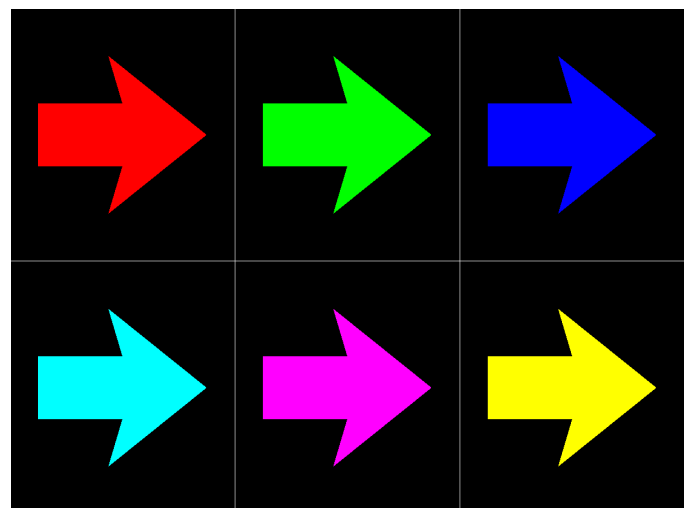


FIGURE 5 – Les six flèches

## C Apprentissage supplémentaire

1. Afin de bien vous familiariser avec les primitives d'OpenGL, modifiez l'ordre et les coordonnées des points afin de tracer des figures différentes.
2. Que voit-on à l'écran lorsque tous les points d'une primitive sont alignés sur une même ligne ? En mode plein ? En mode fil de fer ? (touche 'g')
3. Quelle primitive est la plus facile à utiliser ? Quelle est la plus efficace ?
4. Utiliser la version OpenGL 2.x ou OpenGL 4.x (en modifiant le `#if 0` au début de `afficherScene()`). Voit-on un changement à l'affichage ?
5. Changez d'objet et afficher plutôt la flèche de la Figure 4. Utilisez les coordonnées fournies pour afficher les objets en utilisant chacune des primitives, comme illustré à la Figure 5.
6. Modifier les coordonnées pour afficher un objet en 3D.