

# Polygon 3D

---

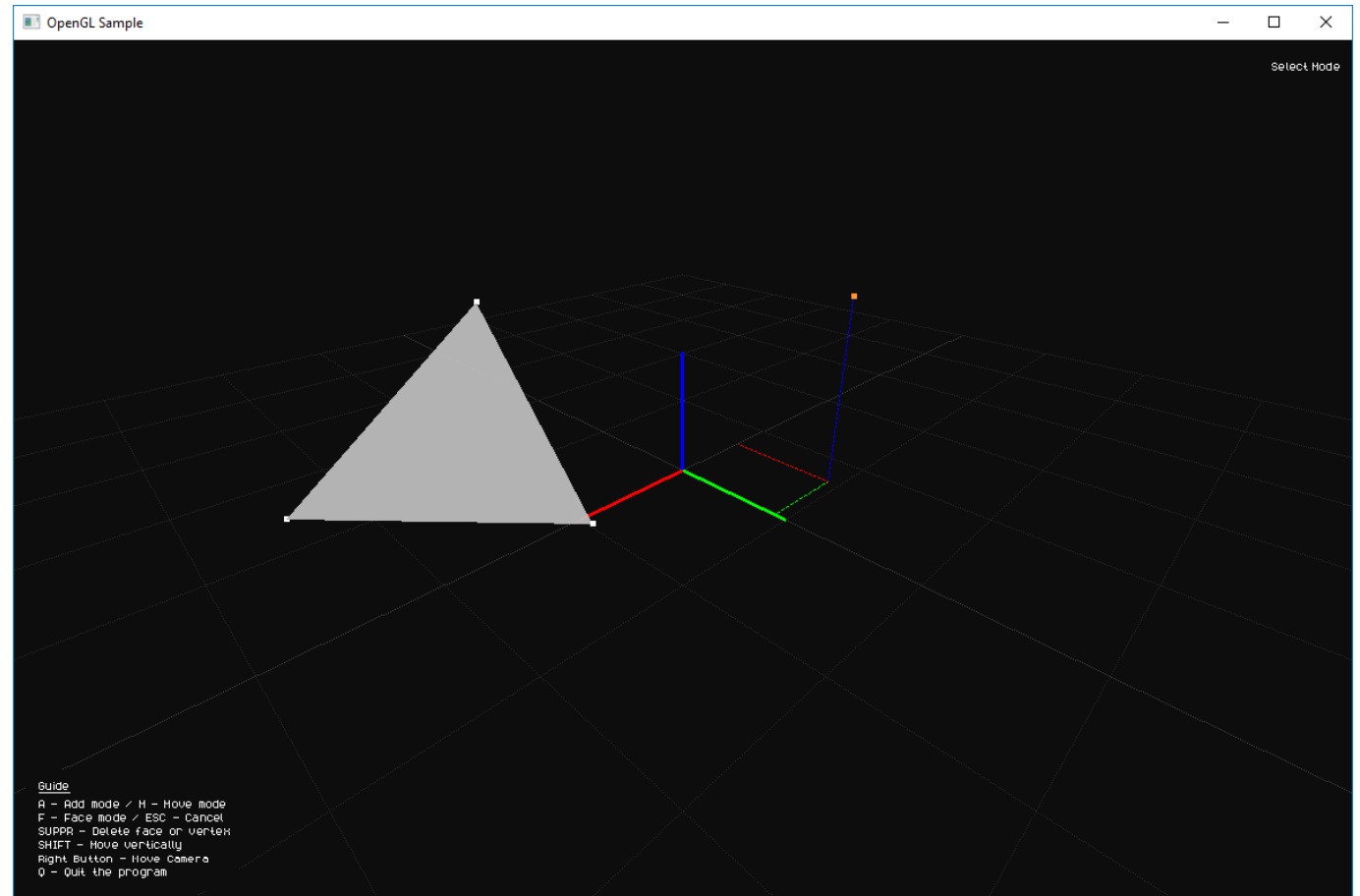
PROJET D'IN204

# Descriptif du Logiciel

Logiciel rudimentaire de manipulation de points et de faces (3 points) en 3D.

- Ajout et Suppression de Points
- Ajout et Suppression de Faces
- Déplacement de Points
- Déplacement de la caméra 3D

Dépendance : OpenGL et Windows



# Réflexions initiales

---

## OBJECTIFS

Inspiré par Blender (logiciel open source de 3D)

Manipulation de points, faces et arrêtes : transformations affines (rotation, zoom, translation)

Déplacement de la caméra

## CHOIX TECHNIQUES

Logiciel en C++

Utilisation de OpenGL

Utilisation de CodeBlock comme IDE pour la compilation et la gestion des librairies

# La structure du programme

---

Un fichier *main.cpp* fournit en grande partie par un modèle CodeBlock

Un fichier *system.cpp* gérant les interactions avec l'utilisateur principalement.

Un fichier *scene.cpp* gérant les objets 3D.

Un fichier *graphics.cpp* qui s'occupe de l'affichage

Le fichier *main.cpp* appelle une fois la fonction l'initialisation des 3 autres « modules » et régulièrement les fonctions de mise à jour de système et graphisme.

La scène contient un vecteur de Points et un vecteur de Faces, ainsi qu'une Caméra.

Pour gérer la 3D, il y a une *struct vec3* qui correspond à un vecteur en 3 dimensions et à l'ensemble des opérations algébriques associées.

Il y a de plus une *struct ray* qui correspond à un rayon avec une origine et une direction.

La classe parentale Selectable correspond à l'ensemble des objets que l'utilisateur peut sélectionner.

# La structure du programme

---

La classe parentale *Selectable* correspond à l'ensemble des objets que l'utilisateur peut sélectionner. Il s'agit d'une classe abstraite qui a les méthodes de sélection (*select*, *deselect*, etc.) et la méthode virtuelle *intersect*.

La classe *Vertex* est une classe enfant de *Selectable*, elle représente les points de l'espace que l'utilisateur peut manipuler.

La classe *Face* est une classe enfant de *Selectable*, elle représente les faces que l'utilisateur peut manipuler. Elle a trois *Vertex* comme propriétés.

La classe *Camera* est une classe qui gère la camera et ses mouvements.

# Problèmes et Solutions

---

## INTERACTIONS

Par soucis d'efficacité, interface minimale basée essentiellement sur des raccourcis claviers (comme dans Blender)

Difficulté des mouvements en 3D dans un écran 2D :

- La caméra garde une distance fixe à l'origine (diminution à 2 dimension donc on peut utiliser la position de la souris)
- Utilisation de *shift* pour le déplacement des points selon d'axe Z.

## AFFICHAGE

Par simplicité, on n'utilise pas de shaders. En effet, il y a à la fois des objets matériels (points et faces) et de l'interface (axes, sol) en 3D.

Un système d'ombres rudimentaire est cependant nécessaire à la visualisation 3D (quelques bugs rares)

Mélange de 2D et de 3D pour l'affichage des aides mémoires.