INF2705 Infographie

Travail pratique 0

Configuration de l'environnement de développement

Département de génie informatique et génie logiciel
Polytechnique Montréal
Tristan Rioux, Hiver 2025

Table des matières

1 Description globale

1.1 But

Le but du TP est de permettre à l'étudiant de se familiariser avec l'environnement de développement des TPs du cours. Il devra s'assurer qu'il est capable de compiler un projet exemple. Un bref rappel des notions de programmation orientée objet en C++ est disponible pour éviter des erreurs courantes d'implémentation.

1.2 Travail demandé

Fonctionnement des TPs

Le système d'exploitation de l'environnement de développement des TPs du cours est Linux dû à l'utilisation d'une librairie partagée (.so) custom pour la correction automatique. Il sera donc obligatoire de travailler sur les postes de travail de l'école ou sur votre machine virtuelle. Rien ne vous empêche de reproduire un environnement de développement sur les autres OS, mais il ne sera pas possible d'exécuter de test automatique.

Les projets dépendent sur les librairies `SDL2`, `glm` et `glew`. Si votre machine ne les a pas d'installer, il est important d'installer les binaires, ainsi que le paquet de développement (souvent marqué "-dev" sur les gestionnaires de paquets) de ces 3 librairies. Des informations supplémentaires à propos de l'installation des librairies sont disponibles sur la FAQ du cours.

La librairie partagée vous permettra de vous auto-corriger et de consulter le corriger directement pour comparer votre résultat avec celui attendu. Il sera important de ne pas modifier les signatures des fonctions qui seront à compléter dans les prochains tps afin d'éviter des problèmes d'exécution.

Pour le TP0, la totalité du code est obfusqué dans la librairie, puisqu'on veut seulement mettre l'accent sur la compilation. Pour les prochains TPs, elle ne contiendra que les tests et le résultat final.

Il est recommandé de rejoindre le serveur Discord (disponible sur la page Moodle) afin d'obtenir du support à l'extérieur des heures de cours. N'oubliez pas de modifier votre nom sur le serveur par votre prénom et nom.

Rappel d'orienté objet C++

Puisque les TPs sont écrit en C++ et utilisent les principes de programmation orientée objet, il est recommandé de réviser les concepts de programmation orientée objet en C++ spécifiquement.

Entre autre, le principe de liste d'initialisation dans les constructeurs est important à maitriser, puisqu'une mauvaise utilisation de celui-ci est une erreur commune.

```
1
2
3  class AutreClasseExemple
4  {
5  private:
6      int m_var;
7      std::string m_text;
8  public:
9      AutreClasseExemple(int var, std::string text)
10     {
11         // À ÉVITER, NE PAS REPRODUIRE
12         m_var = var;
13         m_text = text;
14     }
15 };
16
17 class Exemple
18 {
19 private:
20     AutreClasseExemple m_classe;
21 public:
22     Exemple()
23     // Liste d'initialisation pour construire les classes membres
24     // et initialiser les attributs
25     : m_classe(3, "test")
26     {
27     }
28 };
```

FIGURE 1 – Exemple d'initialisation des classes.

Références

Vous allez sans aucun doute avoir à consulter le Wiki pour des explications approfondies des concepts et la documentation officielle de OpenGL pour comprendre les appels de fonctions :

- Wiki
- Documentation OpenGL

Il existe plusieurs documentations et tutoriels OpenGL sur Internet. Il est très recommandé de consulter les références suivantes tout au long de la session :

- LearnOpenGL.com
- Vidéos tutoriel de ThinMatrix (implémentation en java, mais il explique les concepts de façon claire)
- Vidéos tutoriel de Victor Gordan
- Cours universitaire de Cem Yuksel (le cours du printemps 2021 est recommandé)

Les références proposées contiennent la matière du cours, ainsi que de l'enrichissement pour poursuivre votre apprentissage en infographie.

Compilation et exécution du TP0

La compilation des projets se fait avec un makefile. Une simple commande `make` dans le répertoire `/src` est suffisante pour générer le binaire du TP. La commande `make run` permet de faire la compilation et l'exécution du binaire.

Bien que le binaire soit dans le répertoire `/src/build-Linux`, il est important de l'exécuter à partir du répertoire `/src` (avec la commande `./build-Linux/tp0.exe` à partir du répertoire `/src`), puisque

l'exécutable devra lire les shaders à cette emplacement (ce n'est pas le cas du TP0 par contre). Vous ne devriez pas l'exécuter directement en cliquant dessus.

Suite à la compilation, vous devriez être capable d'exécuter le binaire afin d'obtenir le résultat suivant :

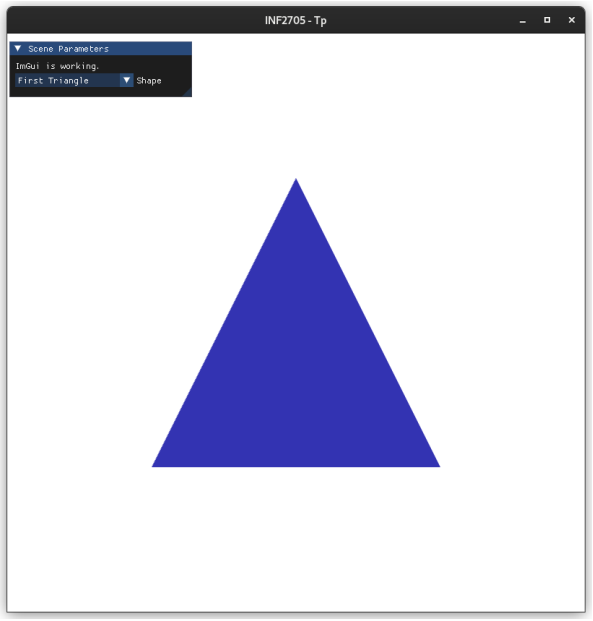

FIGURE 2 – Résultat du TP0.

Ceci complète le TP0 et votre environnement de développement devrait être configuré correctement pour les prochains laboratoires. Vous n'avez rien à remettre pour évaluation.

ANNEXES

A Liste des commandes

Touche	**Description**
`ESC`	Quitter l'application
`SPACE`	Cacher/Décacher la souris (pour le contrôle de la caméra)