HAI719I – Programmation 3D
TP GLSL

Noura Faraj *noura.faraj@umontpellier.fr*

Objectif

Le but de ce TP est d'afficher une scène simple. Paramétrer le pipeline OpenGl. C'est à dire configurer la manière dont openGl affiche les objets en chargeant et parametrant les programmes appelés shaders dans le dossier shaders.

Décompresser le fichier, dans le dossier resultant :
make -j
./tp

Etudier le code de load_shader() dans le fichier src/Shader.cpp afin de charger et compiler le vertex shader et fragment shader. Comprendre tout le code

Question 1

Un triangle est défini dans la structure TriangleVArray, completer les fonctions de cette structure : afficher le à l'aide des shaders en complétant les parties indiquées du code.

Question 2

Passer une variable uniform au shader afin d'appliquer une mise à l'échelle uniforme sur le triangle (variable globale scale definie dans tp.cpp).
Permettre de contrôler ce paramètre à l'aide du clavier (+/-).
Ajouter une translation contrôlée à l'aide du clavier (q/d en x et z/s en y).

Question 3

Ajouter un attribut couleur, mettre à jour le code CPU et GPU. Définir comme couleur = normal du sommet et interpoler la valeur pour les fragments (piste, variable in et out).

Question 4

Mettre à jour la structure de mesh pour afficher le triangle en tant que liste indexée de sommets.

Question 5 : bonus

Definir une nouvelle structure dans laquelle le maillage est affiché en utilisant un VAO et des attributs entrelacés positions, couleurs :

```
std::vector<Vec3> g_vertex_buffer_data {
     Vec3(-1.0f, -1.0f, 0.0f), Vec3(1.0f, 0.0f, 0.0f), //Position, couleur
     Vec3(1.0f, -1.0f, 0.0f), Vec3(0.0f, 1.0f, 0.0f),
     Vec3(1.0f,  1.0f, 0.0f), Vec3(0.0f, 0.0f, 1.0f),
};
```