

Introducció

En aquests exercicis us demanen que implementeu plugins (en C++) pel visualitzador bàsic de l'assignatura. Alguns exercicis requereixen escriure shaders (en GLSL).

Farem servir aquesta nomenclatura:

- **effect plugin** -> cal que desenvolueu un plugin que implementi la interfície **EffectInterface**.
 - **draw plugin**: cal que desenvolueu un plugin que implementi la interfície **DrawInterface**. Teniu un exemple a plugins/draw-immediate.
 - **action plugin** -> cal que desenvolueu un plugin que implementi la interfície **ActionInterface**. Teniu un exemple a plugins/navigate-default
 - **render plugin** -> cal que desenvolueu un plugin que implementi la interfície **RenderInterface**. Teniu un exemple a plugins/render-default
-
- VS, GS i FS fan referència a vertex shader, geometry shader i fragment shader, resp.

Warming up

Desplega les fonts del visualitzador (viewer) en un directori teu i prova construir els binaris, seguint les instruccions que us hem donat. Executa el viewer i prova de carregar algun model (format .obj).

Framerate

Escriu un **Effect Plugin** que sobreescrigui en la cantonada superior dreta de l'àrea de dibuix el *Frame Rate* actual de l'aplicació. Pots fer servir un `QElapsedTimer` per a calcular el temps usat en dibuixar els *frames*, i el mètode `renderText` del `QGLWidget` per a pintar el resultat. Proveu-lo mesurant els temps de dibuixar diferents models.

ModelInfo

Escriu un **Effect Plugin** que sobreescrigui en la cantonada inferior dreta de l'àrea de dibuix la següent informació sobre l'escena: número total d'objectes carregats, número total de polígons, número total de vèrtexs, i el percentatge de polígons que són triangles. Feu una implementació que tingui un impacte negligible en el frame rate.

Animate vertices

Escriu un **effect plugin** que activi un VS per tal d'obtenir el mateix efecte de l'exercici **Animate Vertices (1)**.

El mètode **onPluginLoad** haurà de carregar, compilar i muntar el shader. El mètode **preFrame()** els haurà d'activar i donar un valor apropiat a l'**uniform float time**; el mètode **postFrame()** els haurà de desactivar.

Podeu fer servir l'API d'OpenGL per carregar els shaders, i les classes `QGLShader` i `QGLShaderProgram`, que faciliten la càrrega de shaders des de fitxers, (veure l'*assistant*, o <http://doc.qt.digia.com/qt/qglshaderprogram.html>)

El resultat serà diferent que amb ShaderMaker perquè les normals s'envien per cara, no per vèrtex.

Il·luminació per fragment amb shaders

Escriu un **effect plugin** que activi un VS i un FS per tal de tenir il·luminació Blinn-Phong per fragment.

El mètode `onPluginLoad` haurà de carregar, compilar i muntar els shaders. El mètode `preFrame()` els haurà d'activar, i el mètode `postFrame()` els haurà de desactivar.

Podeu fer servir l'API d'OpenGL per carregar els shaders, i les classes `QGLShader` i `QGLShaderProgram`, que faciliten la càrrega de shaders des de fitxers, (veure l'*assistant*, o <http://doc.qt.digia.com/qt/qglshaderprogram.html>)

El resultat serà diferent que amb ShaderMaker perquè les normals s'envien per cara, no per vèrtex.

Spherize

Escriu un **effect plugin** que activi un VS per tal d'obtenir el mateix efecte de l'exercici **Spherize**.