

Задание №3: Знакомство с WebGL/OpenGL (или каким-то другим API для GPU)

Задание:

- загрузить модель из файла формата OBJ/PLY или STL. Разобраться чем отличаются эти форматы. Выбрать тот, который будет удобно загрузить. Если Ваш компьютер позволяет, настоятельно рекомендуется использовать современную версию OpenGL (3.3 или 4.x) или WebGL.
- Показать покрашенную модель с простейшей доступной моделью освещения
- Реализовать изменение с помощью клавиатуры и/или мыши: перемещения камеры (точки наблюдения), ориентации камеры, параметров проектирования (ортогональная проекция или перспективная проекция с изменением поля зрения камеры (field of view/angle of view/frustum))
- Перемещение камеры с помощью клавиатуры/мыши должно быть достаточно удобно и предсказуемо, чтобы можно было достаточно просто разглядывать объект с разных сторон (возможный бонус перемещаться внутри и вокруг нарисованной трёхмерной сцены). В примерах из <http://www.opengl-tutorial.org/> навигация с помощью мыши и клавиатуры реализована довольно неудачно. Облететь модель с таким управлением довольно сложно. Плюс полный захват мыши меня пугает. Попробуйте понять как реализовано анимированное перемещение модели относительно камеры с помощью мыши в Google Earth, в 3d Studio max, <http://libgglviewer.com/examples/simpleViewer.html> и в других интересных и доступных примерах. Выберите самый удобный с Вашей точки зрения способ. Напишите инструкцию, объясняющую как это делать.
- Из стартового положения камеры визуализируемая сцена должна быть видна целиком и должна быть предусмотрена возможность вернуться в стартовое положение. Также будет хорошо, если будет предусмотрено быстрое переключение между несколькими наиболее важными положениями камеры (общий вид сверху, общий вид сбоку, забавный ракурс 1, забавный ракурс 2,...)
- Нарисовать рёбра поверх покрашенной модели.
- Найти и нарисовать силуэт (такие рёбра, что один из треугольников видим, а другой нет). Видимость определяется знаком скалярного произведения между нормалью к треугольнику и направлением на наблюдателя.
- Попробуйте реализовать отрисовку рёбер во фрагментном шейдере, передавая в качестве атрибутов расстояние до каждого из ребер в экранных координатах. Или же просто рисуйте рёбра как отрезки прямых, смещённые чуть-чуть к наблюдателю относительно триангулированной поверхности.
- *Нарисуйте несколько копий модели расположенных рядом, чтобы наглядно проиллюстрировать разницу между ортогональной и перспективной проекцией.

Рекомендуется не писать программу с нуля, а разобраться в каком-нибудь готовом примере/заготовке или даже нескольких (из Internet). В комментариях или в сопроводительном письме желательно указать источники Вашего вдохновения, и какие изменения Вы сделали. Язык программирования можно использовать любой из тех, в которых доступен OpenGL или WebGL и при этом программу можно было бы скомпилировать/выполнить на компьютере под Windows или Mac OS X. В прошлые годы студенты успешно использовали C++/C, Python, JavaScript (WebGL).

Онлайн учебники по WebGL:

- <http://webglfundamentals.org>
- <http://www.webglacademy.com>
- <https://sites.google.com/site/webglbook/>

Онлайн учебники по OpenGL

- <http://www.opengl-tutorial.org/beginners-tutorials/tutorial-1-opening-a-window/>
- <http://learnopengl.com>
- <http://opengl.datenwolf.net/gltut/html/index.html>
<https://bitbucket.org/alfonse/gltut/downloads>
- <https://moderengl.readthedocs.io/en/stable/ModernGL.html>
-

Для выполнения задания с помощью C/C++ Вам пригодятся:

- Одна из кроссплатформенных библиотек для создания окна и инициализации OpenGL контекста в нём: freeglut или glfw или даже монструозный Qt.
<http://libqglviewer.com/index.html>
<http://doc.qt.io/qt-5/examples-widgets-opengl.html>
- glew или glLoadGen (<https://bitbucket.org/alfonse/gloadgen/wiki/Home>). Заголовочные файлы для современных версий OpenGL доступны с помощью этих библиотек.
- Вспомогательная библиотека полезных математических функций/классов C++ эквивалентным функция/встроенным типам языка шейдеров (GLSL): glm

Коллекции примеров к популярным offline книжкам:

- <https://github.com/daw42/glscookbook>
- <http://www.opengl-redbook.com/>

С Python можно использовать что-нибудь вроде <https://pypi.python.org/pypi/omgl>