

# Инструкция по установке и сборке проектов лабораторных работ по OpenGL

А. С. Болдарев

14 сентября 2017 г.

С целью обеспечения переносимости программы (т. е. возможности ее сборки и выполнения лабораторной работы с использованием широкого класса средств разработки — Visual Studio любой версии, Qt Creator, Eclipse, утилита make и т. д.), а также для демонстрации способа обеспечения такой мультиплатформенности, шаблон лабораторной работы сделан для сборки системой CMake. Здесь приведена инструкция, как собирать проект с помощью CMake.

## 1 Установка CMake на компьютере

Если на Вашем компьютере не установлен CMake (в компьютерных классах 2-го этажа Станкина он не установлен), следует зайти на сайт поддержки <http://www.cmake.org/download> и установить актуальную версию, подходящую для Вашей операционной системы, следуя инструкциям установщика. Для системы Windows, однако, есть одно ограничение — если Вы не имеете доступа к системе с правами администратора, установщик не сработает правильно. В этом случае следует скачать ZIP архив (вместо установщика MSI) и распаковать его в какой-либо каталог Вашего профиля пользователя, тогда для запуска CMake Вам придется запускать файл `bin\cmake-gui.exe` из папки, извлеченной из архива. При полноценной установке ярлык для запуска CMake окажется в общем меню системы (для Windows), либо для запуска нужно будет вводить в командной строке команду `cmake-gui` (работает и для Windows, и для Linux). Кроме `cmake-gui`, есть еще `cmake` (получает управляющие параметры из командной строки) и `ccmake` (для Linux предоставляет псевдографический интерфейс вместо графического), в принципе они предоставляют ту же самую функциональность, но для первого ознакомления `cmake-gui` проще всего.

## 2 Сборка проекта

Исходный код проекта обязательно должен содержать хотя бы один файл с именем `CMakeLists.txt`, каталог файловой системы, в котором этот файл находится, должен быть задан в CMake в строке “Where is the source code:”. Этот файл `CMakeLists.txt`, собственно, и служит первичным описанием проекта — в нем указано, какие файлы с исходными текстами должны быть в проект включены, какие нужны дополнительные библиотеки и т. д.

Вторая строка, “Where to build the binaries:”, должна содержать путь к тому каталогу, где CMake создаст все файлы, в частности, файл для Visual Studio, `*.sln`, или `makefile` для Linux’а. В принципе, можно задать тот же каталог, что и для исходного кода, тогда все файлы, связанные с проектом, будут сосредоточены в одном месте. Я,

однако, предпочитаю не мусорить в каталоге исходного кода, и задаю в этой строке путь к какой-нибудь пустой папке (можно даже несуществующей — CMake в этом случае ее создаст)<sup>1</sup>.

Для более удобной группировки управляющих переменных я рекомендую отметить галочками “Grouped” и “Advanced”. Затем нажимаем “Configure”. Соглашаемся на создание нового каталога (если путь к binaries указывал на несуществующую папку), затем выбираем нужную версию Visual Studio, нужно посмотреть, какая установлена на Вашем компьютере и обычно используется. Вместо Visual Studio при необходимости можно указать и другую среду разработки, принципиально последовательность действий от этого не изменится.

Через некоторое время, после того как CMake проверит наличие соответствующей среды разработки, протестирует компиляторы и попытается найти нужные внешние библиотеки, в окне CMake появится список управляющих переменных. Переменные, появляющиеся в первый раз, отмечены красным цветом фона. Значения переменных можно (и в некоторых случаях нужно) редактировать, после чего снова нажимать “Configure”. Этот процесс повторять до тех пор, пока все переменные не станут белыми, и пока не будут внесены все необходимые модификации в их значения. После чего нужно нажать “Generate”, и открывать проект кнопкой “Open Project” или двойным кликом по файлу \*.sln, появившемуся в каталоге сборки.

Решение (solution), которое создано CMake’ом, содержит кроме содержательного проекта (или проектов) еще и пару служебных — ALL\_BUILD и ZERO\_CHECK. Поэтому для дальнейшей работы с проектом (конкретно, для того чтобы команды запуска программы через отладчик или без отладчика приводили бы к запуску нужного проекта), нужно в списке проектов правой кнопкой мыши щелкнуть по содержательному проекту, и в появившемся контекстном меню выбрать “Set as StartUp Project” (или русскоязычный аналог). Имя проекта должно отобразиться жирным шрифтом.

### 3 Внешние библиотеки

Для выполнения лабораторных работ по OpenGL нужны две внешние библиотеки — сама OpenGL и GLUT. В принципе, если они корректно установлены в системе, то CMake сама их обнаруживает, и в этом случае сборка проекта CMake’ом не требует какого-либо редактирования управляющих переменных. Библиотека OpenGL обычно установлена и успешно обнаруживается, так как является частью среды разработки (так же, как библиотеки поддержки времени выполнения). С библиотекой GLUT довольно часто возникают проблемы, и тогда придется ее сначала установить (если ее нет вообще), а потом, возможно, вручную задать каталог включаемых файлов и путь к самой библиотеке через соответствующие управляющие переменные CMake.

Для установки библиотеки GLUT в среде Linux я отсылаю читателя на сайт

---

<sup>1</sup>Собранная программа, будучи запущенной через среду Visual Studio, также будет считать этот каталог своим рабочим каталогом, и искать именно там все файлы, которые она должна открыть, если они заданы только именем. Это существенно для второй лабораторной работы, где программа должна загрузить файл с изображением для текстуры. В этом случае придется проделать одну из трех вещей:

- Скопировать файлы с изображением \*.bmp из каталога исходных текстов в каталог сборки;
- Средствами Visual Studio задать для проекта другой каталог по умолчанию, для этого в свойствах проекта во вкладке “Debugging” задать правильный “Working Directory”;
- Использовать для сборки каталог исходных кодов.

<http://www.opengl.org/resources/libraries/glut/> за дистрибутивами и инструкциями по установке. Для Windows, к сожалению, поддержка GLUT в настоящее время не осуществляется, поэтому я рекомендую воспользоваться архивом, распространяемым вместе с лабораторными работами. В этом случае нужно распаковать архив `glut-3.7.6-src.zip`, открыть файл `glut.dsw` с помощью Visual Studio, переключить конфигурацию сборки на Release и собрать проект. Затем создать где-либо у себя в профиле пользователя каталог GLUT, скопировать в него из `glut-3.7.6` содержимое каталогов `include` и `lib\glut\Release`. После этого следует включить каталог GLUT в переменную окружения PATH через Control Panel — System and Security — System — Advanced system settings — Environment Variables...

Если включить каталог GLUT в переменную окружения PATH не получится, то, во-первых, CMake не сможет найти библиотеку GLUT. В этом случае в ходе CMake-сборки придется задать управляющие переменные `GLUT_INCLUDE_DIR = GLUT` и `GLUT_glut_LIBRARY = GLUT/glut32.lib` (указывать полные пути). Во-вторых, в ходе выполнения программы она не сможет найти и загрузить динамическую библиотеку `glut32.dll`, и ее придется скопировать в каталог, где лежит выполняемый файл проекта, т. е. Debug или Release подкаталог каталога сборки.