COSC363 IDE Options

Last compiled: 2019/02/23

Abstract

In past years we have recommended students use Geany as there editor while completing labs and assignment material. We have opted for this because students have experience using Geany from completing ENCE260, and it is quite simple to start using.

You are still welcome to use tools like Geany, that said using an IDE has some strong advantages, like code completion and syntactic analysis. This reduces the chances of making small, hard to spot errors.

A lot of modern C/C++ IDE's use applications to manage building a project. Here we will introduce cmake as it is a cross-platform system that works well with a wide variety of IDE's like:

- Qtcreator: on the CSSE computers
- Kdevelop
- Clion: Students can get a free license see `www.jetbrains.com/student/`

We will cover the basic use of Qtcreator.

Cmake

We will provide cmake files for all labs that you are assigned, and you can use these to learn the very rudimentary cmake required for COSC363. An example cmake file is:

```
1  cmake_minimum_required(VERSION 2.8)
2
3  project(lab1)
4  add_executable(Teapot.out Teapot.cpp)
5
6  find_package(OpenGL REQUIRED)
7  find_package(GLUT REQUIRED)
8  target_include_directories(Teapot.out PRIVATE
9      ${OPENGL_INCLUDE_DIRS} ${GLUT_INCLUDE_DIRS})
10
11 target_link_libraries(Teapot.out ${OPENGL_LIBRARIES} ${GLUT_LIBRARY})
```

Important points:

Going through line by line in the listing above

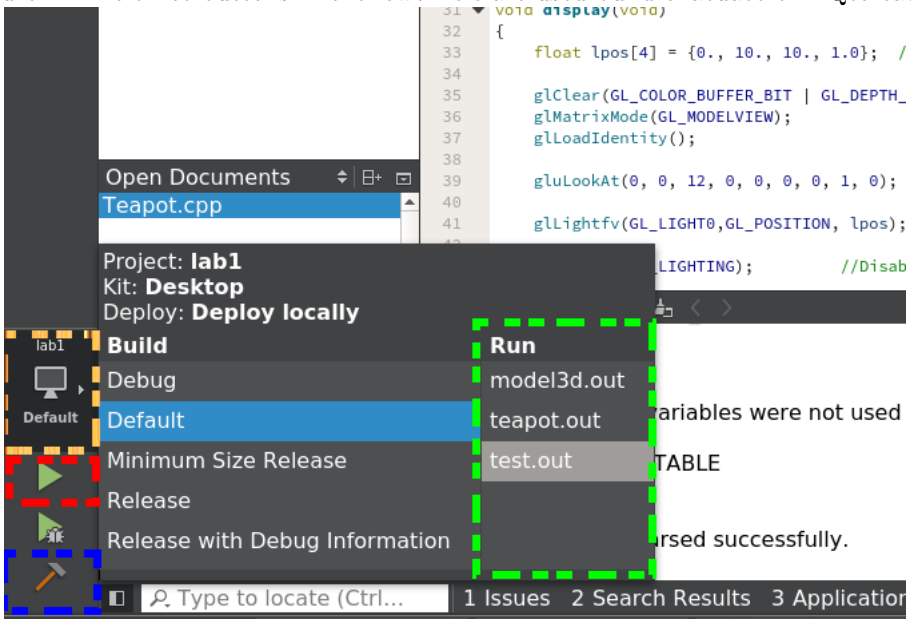
- `cmake_minimum_required` means that if a cmake that meets minimum version requirement is not installed then it will not attempt to build the project. This is because it is assumed that *CmakeLists.txt* uses features that require at least this cmake version.
- `project` gives the overall project a name
- `add_executable` names an executable that our project knows how to generate, and states the files that will be used to compile it
- `find_package` seeks out the libraries that our project will use, states that if cmake can't find it then stop building the project

- `target_include_directories` tells cmake to find and make available to the compiler the headers for OpenGL and glut. The PRIVATE argument means that the headers will only be made available to this target and not its dependencies, this is not a super important point.
- `target_link_libraries` actually links the libraries to the specific executable

Qtcreator

- Open Qtcreator
- goto: File →Open File or Project... →navgate to your project file →select *CMaleLists.txt*
- Select the Desktop kit →Configure Project

Figure 1: The three buttons in the lower left are used build exacuatble in Qtcreator.



I will refer to the sections of the figure by there colored borders.

- Using the orange button we can select the build settings and then in the green section we can select the executable to build.
- The red button will run the program that has been selected, if the project is not built then it will also build it.
- The blue button will build the current selected executable

Cool Tips

I will put things here that are neat tricks as they become apparent.

- **This is important for lab 1, to locate the *.off files.**
  To change the current working directory of the project click on projects in the left column select run and change the path of Working directory.