First steps

Wombacher Sascha

28. Oktober 2017

1 Introduction

This small guide will help you install and run all provided examples and provide some info about how to add you own projects.

Getting started:

- Introduction for Windows users (section 3)
- Introduction for macOS users (section 4)
- Introduction for Linux users (section 5)

Available documentation (located in *<ProjectDirectory>/Documentation*):

- HTML (recommended)
- PDF

2 General

This library uses some third Party libraries:

- GLM (GL mathematics for vector and matrix operations)
- OpenCV (ComputerVision library for 2D graphics)
- Glut/Freeglut (3D, currently only OpenGL 1.x is used)
- InfInt (Lib for very long int values, C# equivalent BigInt)

3 Windows

Installation:

1. Install Visual Studio 2015/2017, link: `https://www.visualstudio.com/downloads/`
   - Required should only be the default C++ development tools
   - if this doesn't work: Install all packages which are mentioning 'C++'
2. Run 'createVisualStudioSolution_<Version>.bat'
3. Open Visual Studio solution <ProjectDirectory>/VisualStudio/ALL_BUILD
4. Build all (Debug/Release libraries should both be linked)
5. Set a example Project: Right click on example → set as start Project
6. Run example

Create a new project:

1. Add a new *.cpp file in directory 'Solutions'
2. Run 'createVisualStudioSolution_<Version>.bat'

Remove a project:

1. Remove the corresponding *.cpp file from the 'Solutions' directory
2. Run 'createVisualStudioSolution_<Version>.bat'

3.1 Some helpful Visual Studio functionalities

Set a command line parameter

Some Examples require a command line argument(s). If you want to use those parameter you have three options:

- Set parameter in Visual studio:
  - Right click on project → preferences
  - Go to Debugging → command arguments
- Open program with command line and add a parameter
- If you only have 1 input parameter you can drag and drop a file onto the executable

4 Mac

Installation:

1. Install XCode, link: `https://developer.apple.com/download/`
2. Install dependencies by running 'mac_installDependencies.sh'
3. Run 'createXCodeProject.sh'
4. Open XCodeProject: <ProjectDirectory>/XCodeProject/*.xcodeproj
5. Build all examples
6. Choose a example and run it

Create a new project:

1. Add a new *.cpp file in directory 'Solutions'
2. Run 'createXCodeProject.sh'

Remove a project:

1. Remove the corresponding *.cpp file from the 'Solutions' directory
2. Run 'createXCodeProject.sh'

5 Linux

Currently three dependency commands are available:

- Apt-Get based (Ubuntu/Debian/...)
- Pacman based (Arch/Manjaro/...)
- Yum based (Fedora/RedHat/...)

If you own another distribution you should know how to install dependencies, required packages:

- GL
- GLU
- Freeglut
- GLM
- OpenCV (2.4 and 3.1 have been tested)
- libxmu
- libxi
- cmake

5.1 Linux installation

Installation:

1. Install dependencies by running 'linux_installDependencies.sh'
2. Run 'createMakeFileProject.sh'
3. Run 'cd <ProjectDirectory>/MakeFileProject' → 'make -j 4'
4. All compiled executables are located in: <ProjectDirectory>/bin
5. Choose a example and run it

Create a new project:

1. Add a new *.cpp file in directory 'Solutions'
2. Run 'createMakeFileProject.sh'

Remove a project:

1. Remove the corresponding *.cpp file from the 'Solutions' directory
2. Run 'createMakeFileProject.sh'

5.2 Linux IDEs

There are a couple of IDE's available for Linux. Possible IDE's with integrated cmake support are:

- QtCreator
- CLion (free for students)