Vaa3D , Update to QT6

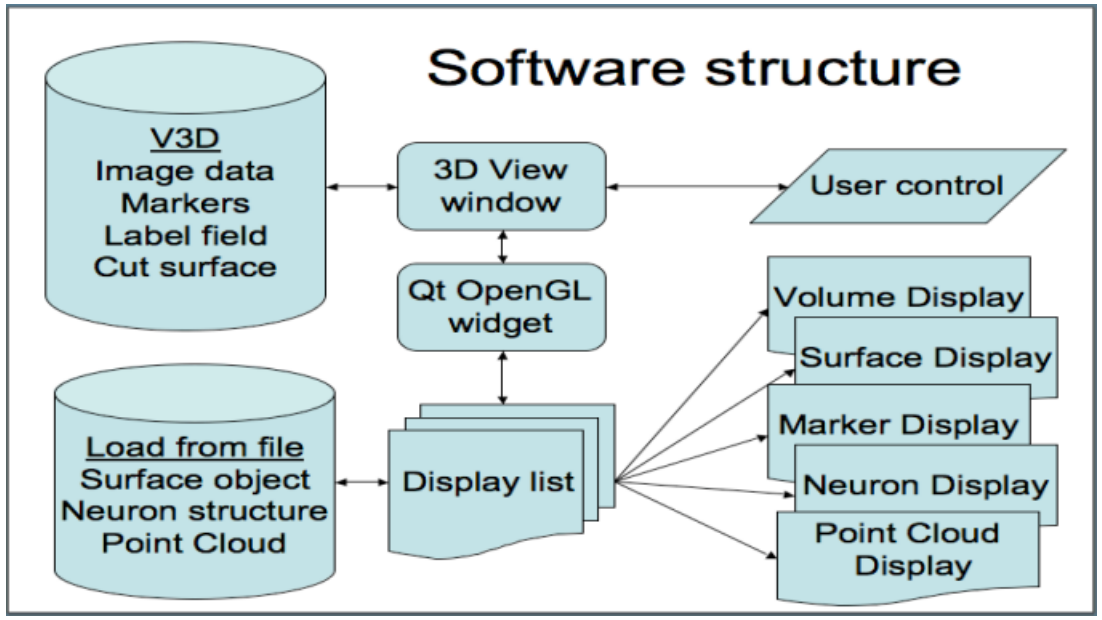
2021.11.23

Jiangshan Liang

Luchen Deng

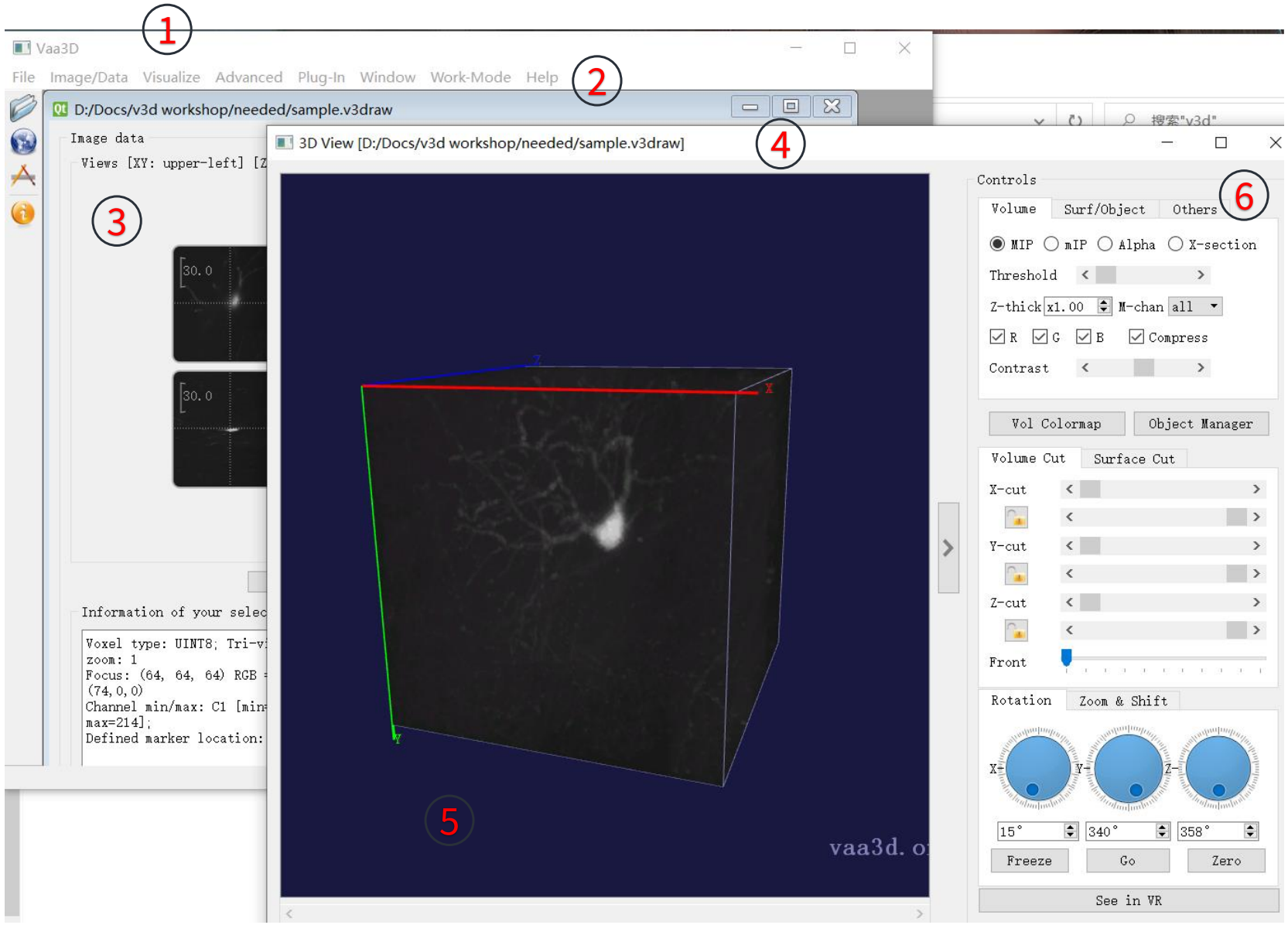
Contents

Vaa3D System Structure



- Feature
  - Cross-platform supported by Qt and OpenGL
  - Visualize volume image and surface object seamlessly
  - Integrated into the image analysis pipeline
  - Support 3D image marker visualization
  - Support SWC object visualization

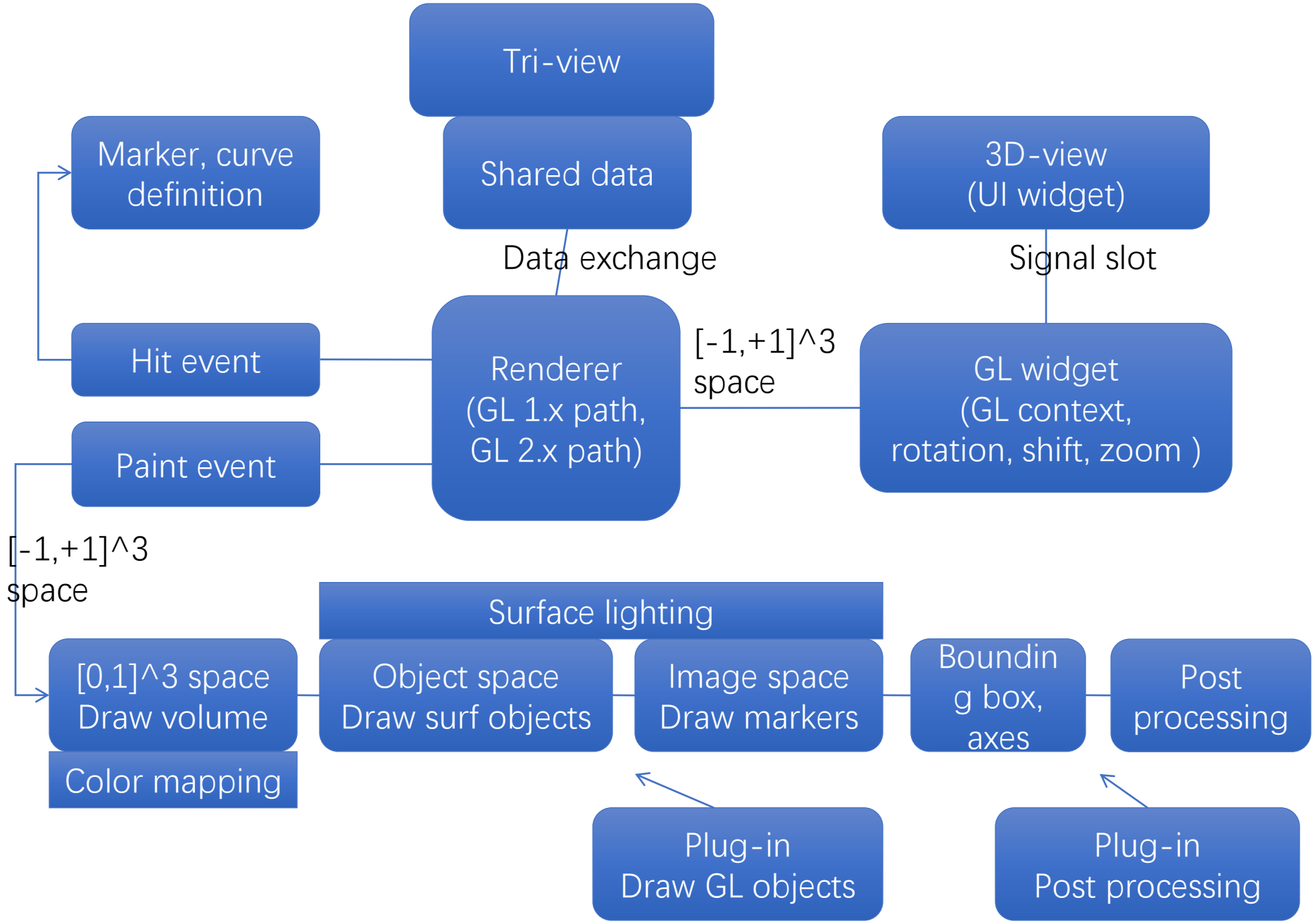
Vaa3d Widgets



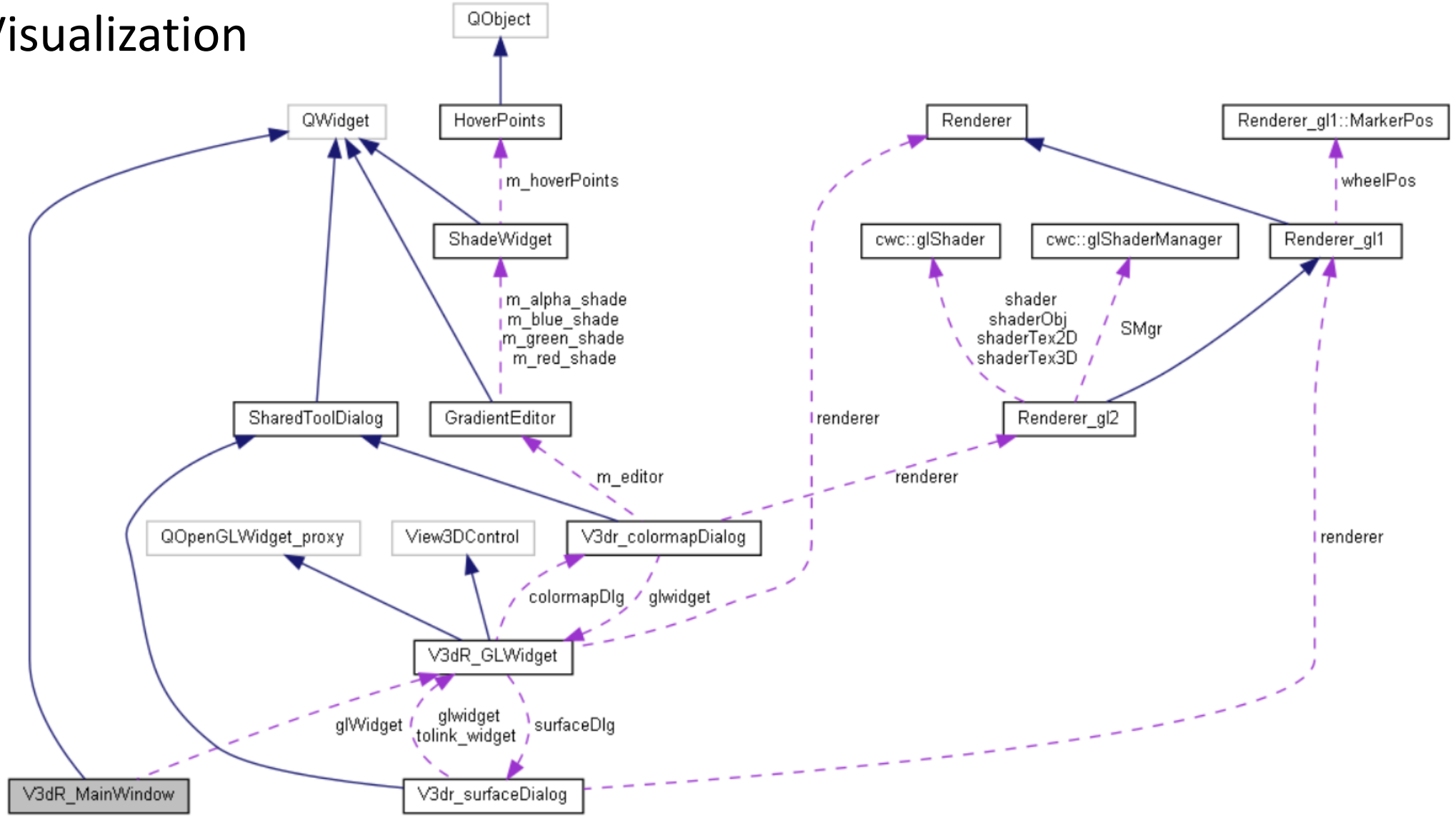
1. Vaa3D MainWindow
2. Menu Bar
3. Tri-view Window
4. V3dr_MainWindow(3D)
5. V3dr_GLWidget
6. V3dr_Control_Signal

3D-view pipeline

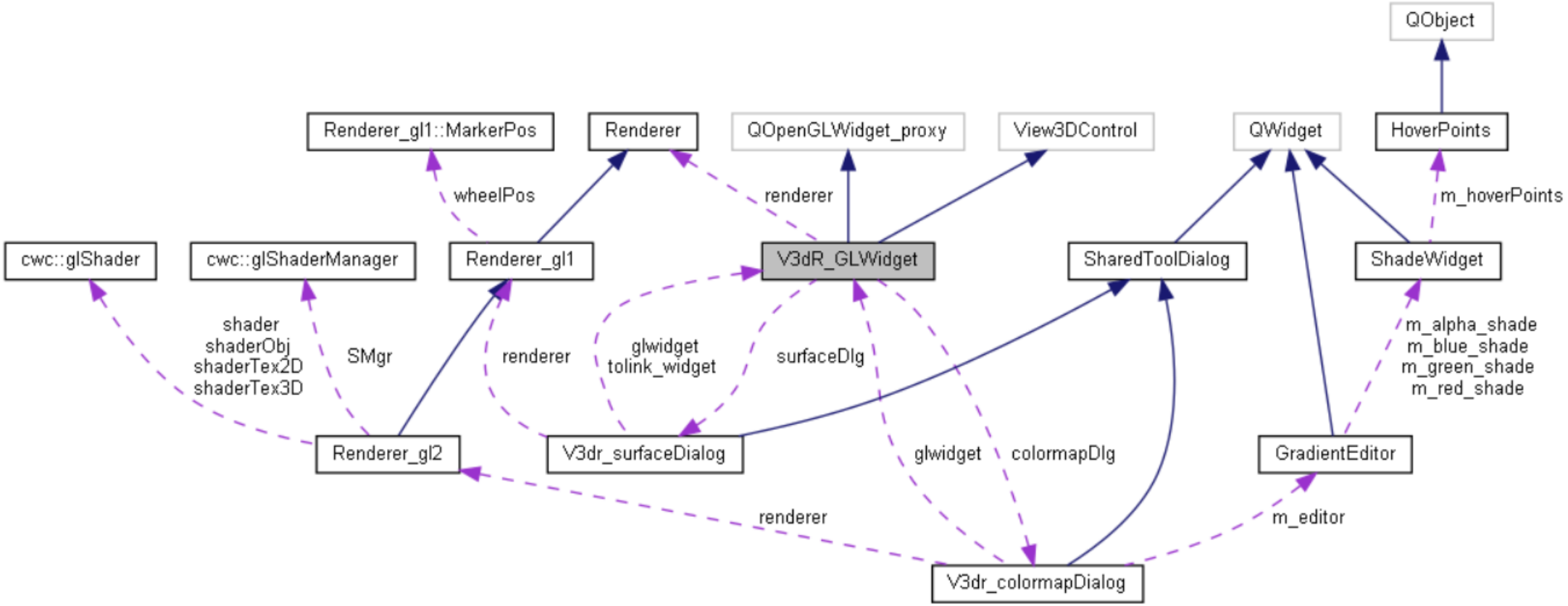
- Renderer acts as a paintbrush
- Functions like maker or drawing curves are accomplished through Hit event
- Surface objects and Plug-in base on Paint event



Code Visualization



Pay attention to the 'Path' : V3dR_MainWindow-->V3dR_GLWidget-->RendererX



- Renderer is base for rendering
- Renderer_gl1 inherited from Renderer, is the drawing body, which can be subdivided into:
  - Renderer_tex : Volume rendering
  - Renderer_obj : Surface object rendering;
  - Renderer_hit: object selection and the functions of object's context menu
- Renderer_gl2 is used for subsequent additions to some functions

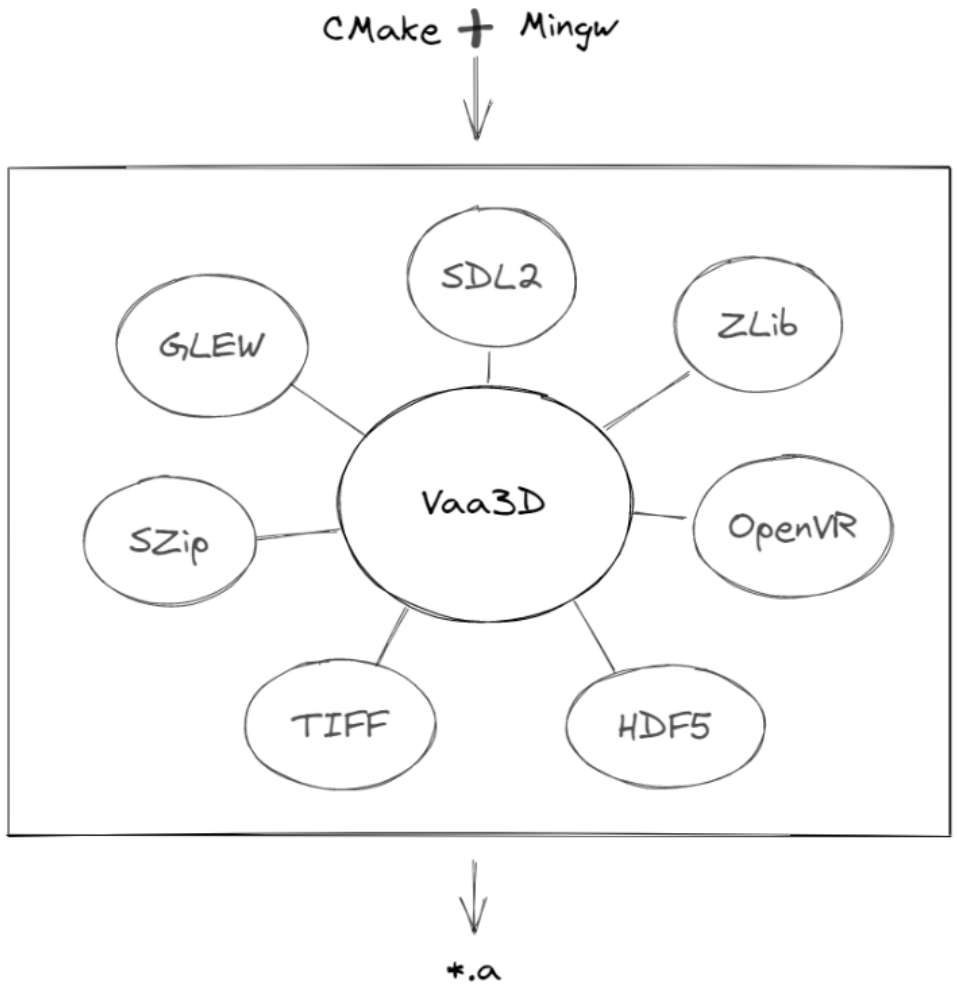
Some previously existing problems for building Vaa3D

- On any platform: building Vaa3D relies on Qt4, which is very much outdated
- On Windows: building Vaa3D relies on outdated Visual Studio environment (e.g. VS2013)
- On latest versions of macOS: Vaa3D cannot be built (root cause is the using of Qt4)

Replacing Visual Studio with Mingw on Windows platform

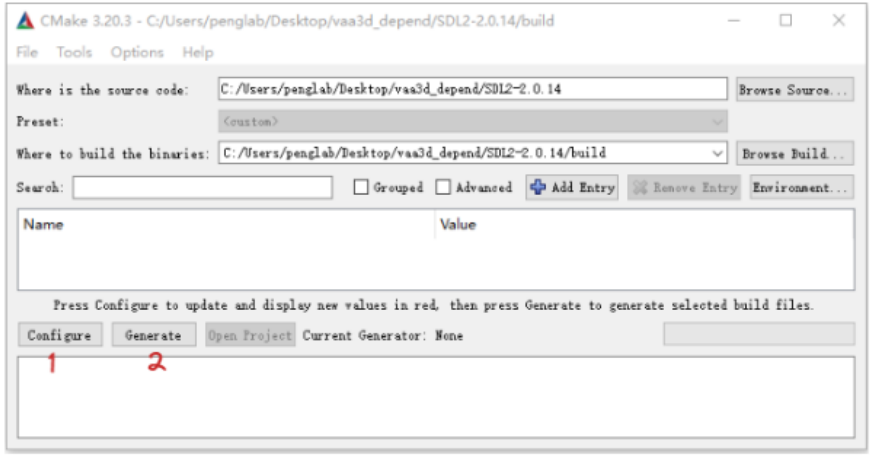
- Status: completed
- Now Vaa3D compilation on Windows is more simplified, forward compatible, and no longer needs Visual Studio
- In order to build the Vaa3D core program using Mingw, we have:
  - Built all the 3rd party libs in Mingw
  - Fixed OpenVR in Mingw environment

Vaa3D Dependency Library



Step

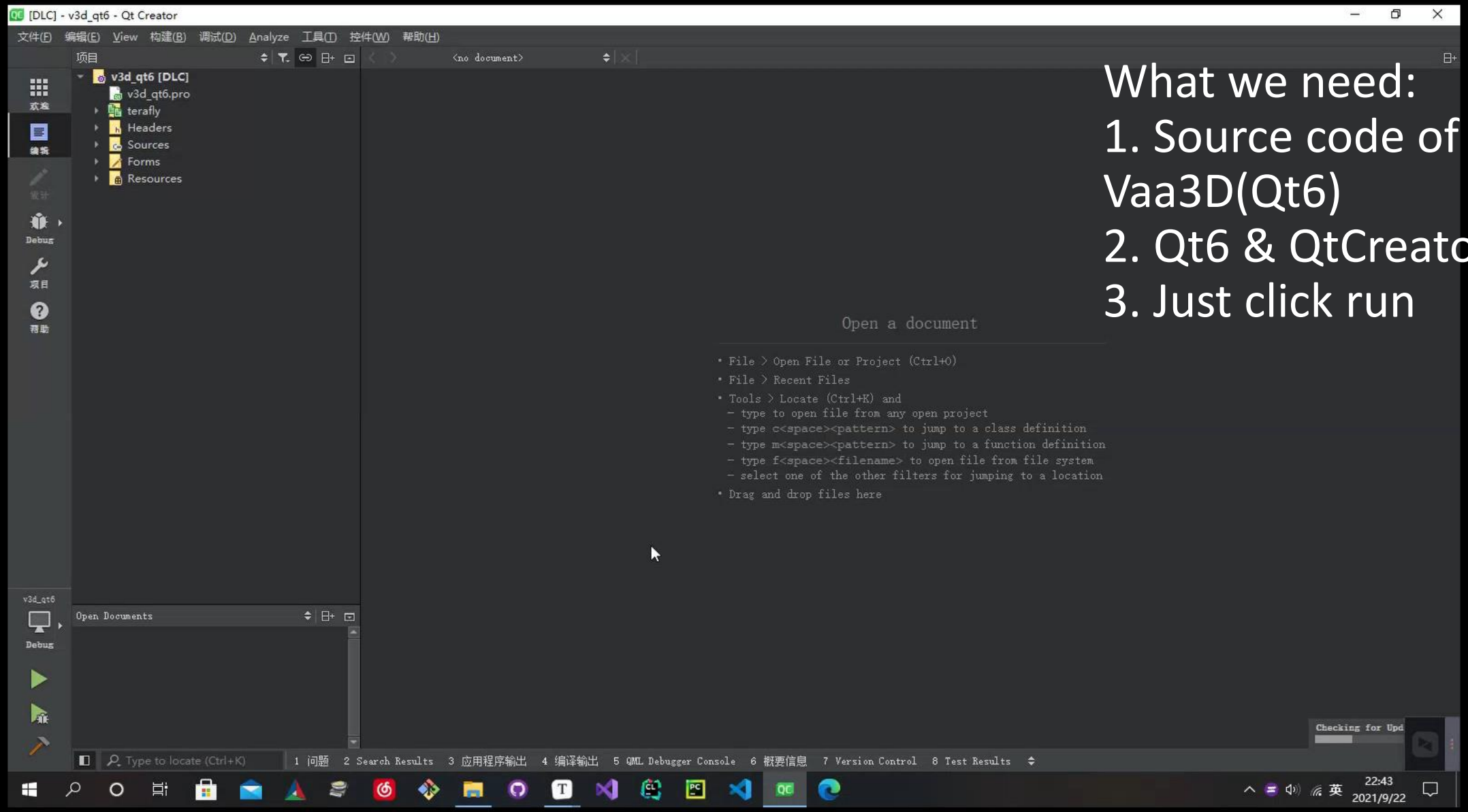
- Open Cmake gui, select the file path of source code and determine the path of the generated file
- Click the configure and generate, the makefile file will be built
- Open the CMD, go to the appropriate Build file and execute the following command:
  - make or mingw32-make



Upgrading Qt from Version 4 to Version 6

- 3drendered,terafly comment
- Qt4 - Qt6 syntax update, some functions under qt4, in qt6 is deprecated
- Un-comment
- Recompile all plugins

Compile qt6 version of vaa3d on Windows environment



What we need:

1. Source code of Vaa3D(Qt6)
2. Qt6 & QtCreator
3. Just click run

Mac Vaa3d

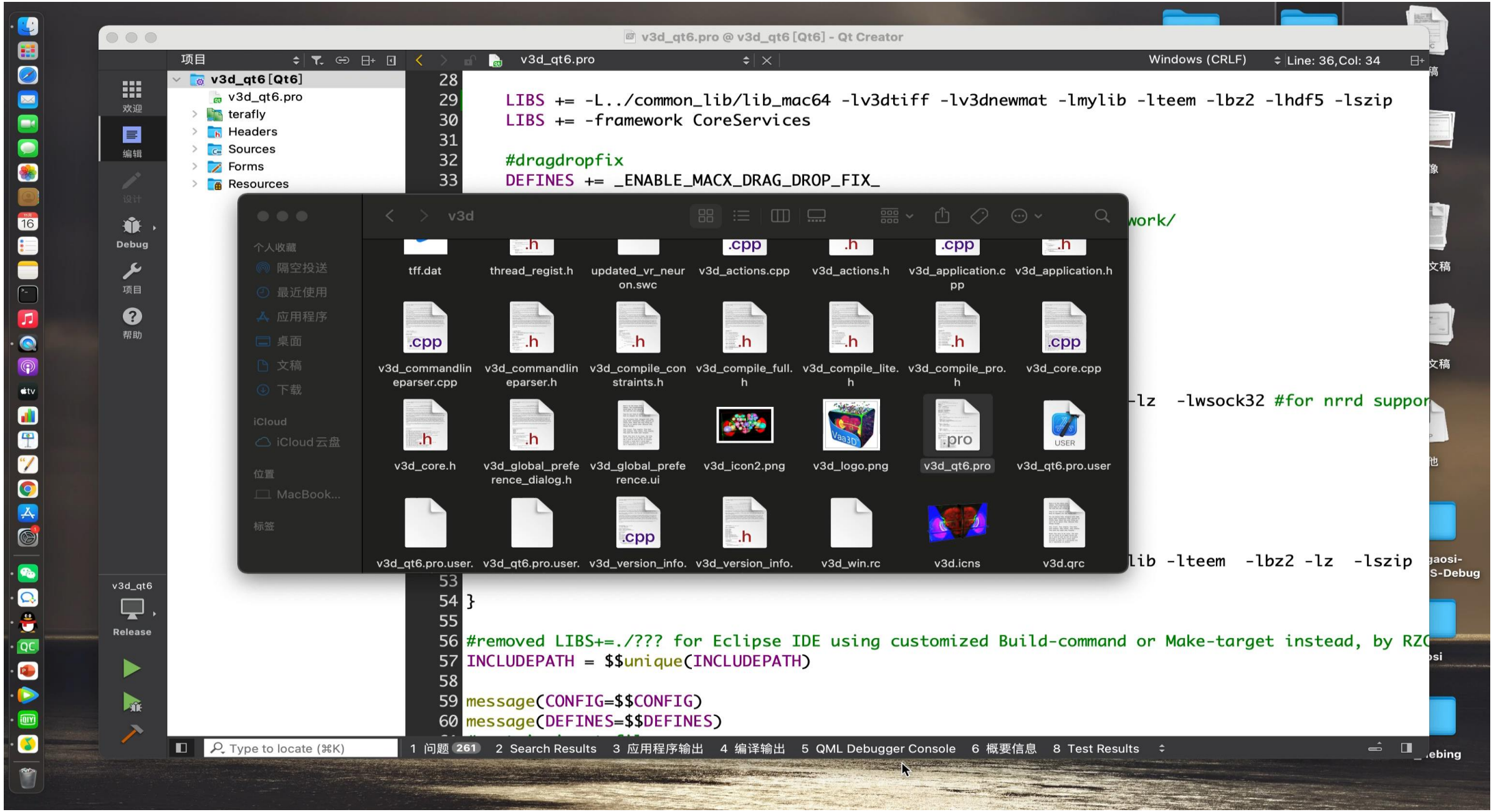

lib_mac64


mingw64


lib_ubuntu

```
macx{
    DEFINES += MACOS_SYSTEM

    LIBS += -L../common_lib/lib_mac64 -lv3dtiff -lv3dnewmat -lmylib -lteem  -lbz2 -lhdf5 -lszip
    LIBS += -framework CoreServices

    #dragdropfix
    DEFINES += _ENABLE_MACX_DRAG_DROP_FIX_
    OBJECTIVE_SOURCES += yosemiteFileURLfix.mm
    QMAKE_LFLAGS += -F /System/Library/Frameworks/Foundation.framework/
    LIBS += -framework Foundation
}

win32{
    DEFINES += WINDOWS_SYSTEM
    DEFINES += __ALLOW_VR_FUNCS__
    LIBS += -lm -lv3dtiff -lv3dnewmat
    LIBS += -lOpengl32  -lglu32
    LIBS += -L../common_lib/mingw64  -lhdf5 -lszip -lzlib  -lteem  -lz  -lwsock32 #for nrrd suppo

}

unix:!macx {
    DEFINES += LINUX_SYSTEM
    DEFINES += __ALLOW_VR_FUNCS__
    LIBS =  -lGLU -lglut
    LIBS += -L../common_lib/lib_ubuntu   -lv3dtiff -lv3dnewmat -lmylib -lteem  -lbz2 -lz  -lszip

}
```

Compile qt6 version of vaa3d on Mac environment



Running qt4 version of vaa3d on Mac

Vaa3d Principle of the plug-in implementation

- The new plugins all inherit from the V3DPluginInterface2_1 class, and V3DPluginInterface2_1 inherits from the V3DPluginInterface2 class, which contains several virtual functions that we need

- Take the *domenu* function for example, the virtual function has the V3DPluginCallback2 data type inside

```
virtual void domenu(const QString & menu_name, V3DPluginCallback2 & v3d, QWidget * parent) = 0;
```

- Then, the V3DPluginCallback2 class is defined in v3d_interface.h and inherits from the V3DPluginCallback class, there are various virtual functions inside that can be called when writing plugins

```
class V3DPluginCallback2 : public V3DPluginCallback
{
public:
    virtual ~V3DPluginCallback2() {}

    virtual View3DControl * getView3DControl(v3dhandle image_window) = 0;
    virtual View3DControl * getLocalView3DControl(v3dhandle image_window) = 0;
    virtual TriviewControl * getTriviewControl(v3dhandle image_window) = 0;

    //added PHC 20120406. add a main window handle, to allow access everything in Vaa3D

    virtual MainWindow * getVaa3DMainWindow() = 0;
    virtual QList <V3dR_MainWindow *> getListAll3DViewers() = 0;
    virtual V3dR_MainWindow * find3DViewerByName(QString fileName) = 0; //the name can be partially matched

    //added PHC 20120406 to allow uses to access the surface data objects in a 3D viewer, but based on a tri-view
    virtual QList <NeuronTree> * getHandleNeuronTrees_3DGlobalViewer(v3dhandle image_window) = 0;
    virtual QList <CellAPO>    * getHandleAPOCellList_3DGlobalViewer(v3dhandle image_window) = 0;
    virtual QList <LabelSurf> getListLabelSurf_3DGlobalViewer(v3dhandle image_window) = 0;
    virtual bool setListLabelSurf_3DGlobalViewer(v3dhandle image_window, QList <LabelSurf> listLabelSurfinput) = 0;

    //added PHC 20120406 to allow uses to access the surface data objects in a 3D viewer
    virtual QList <NeuronTree> * getHandleNeuronTrees_Any3DViewer(V3dR_MainWindow *w) = 0;
```

Vaa3D(Qt6) plugin build method

1. In pro file, it is necessary to add QT += widgets. When this module is included, qmake (makefile) will set the include path and lib path, also set the libs when linking.

2. The plug-in class written needs to inherit from the V3DPluginInterface2_1 class and also declare " Q_PLUGIN_METADATA(IID"com.janelia.v3d.V3DPluginInterface/2.1") " inside the class

3. Macro ‘Q_EXPORT_PLUGIN2’ is obsolete in the cpp file, just comment it out

e.g. Plugin: Gaussian_Filter

.pro file

```
5
6  QT += widgets
7  INCLUDEPATH     += $$V3DMAINPATH/basic_c_fun
8  INCLUDEPATH += $$V3DMAINPATH/common_lib/include
9  INCLUDEPATH += main
10 HEADERS += $$V3DMAINPATH/basic_c_fun/basic_memory.h \
11     gaosi.h
12 HEADERS += gaosi.h
13 SOURCES  = gaosi.cpp
14
```

.h file

```
class GaussianFilterPlugin : public QObject, public V3DPluginInterface2_1
{
     Q_OBJECT
     Q_INTERFACES(V3DPluginInterface2_1);                    extra ';' inside a cl
     Q_PLUGIN_METADATA(IID"com.janelia.v3d.V3DPluginInterface/2.1")
public:
    float getPluginVersion() const {return 1.1f;}    'getPluginVersion' overrides a membe

    QStringList menulist() const;    'menulist' overrides a member function but is not ma
    void domenu(const QString &menu_name, V3DPluginCallback2 &callback, QWidget *parent);

    QStringList funclist() const;    'funclist' overrides a member function but is not ma
     bool dofunc(const QString &func_name, const V3DPluginArgList &input, V3DPluginArgList &
};
```

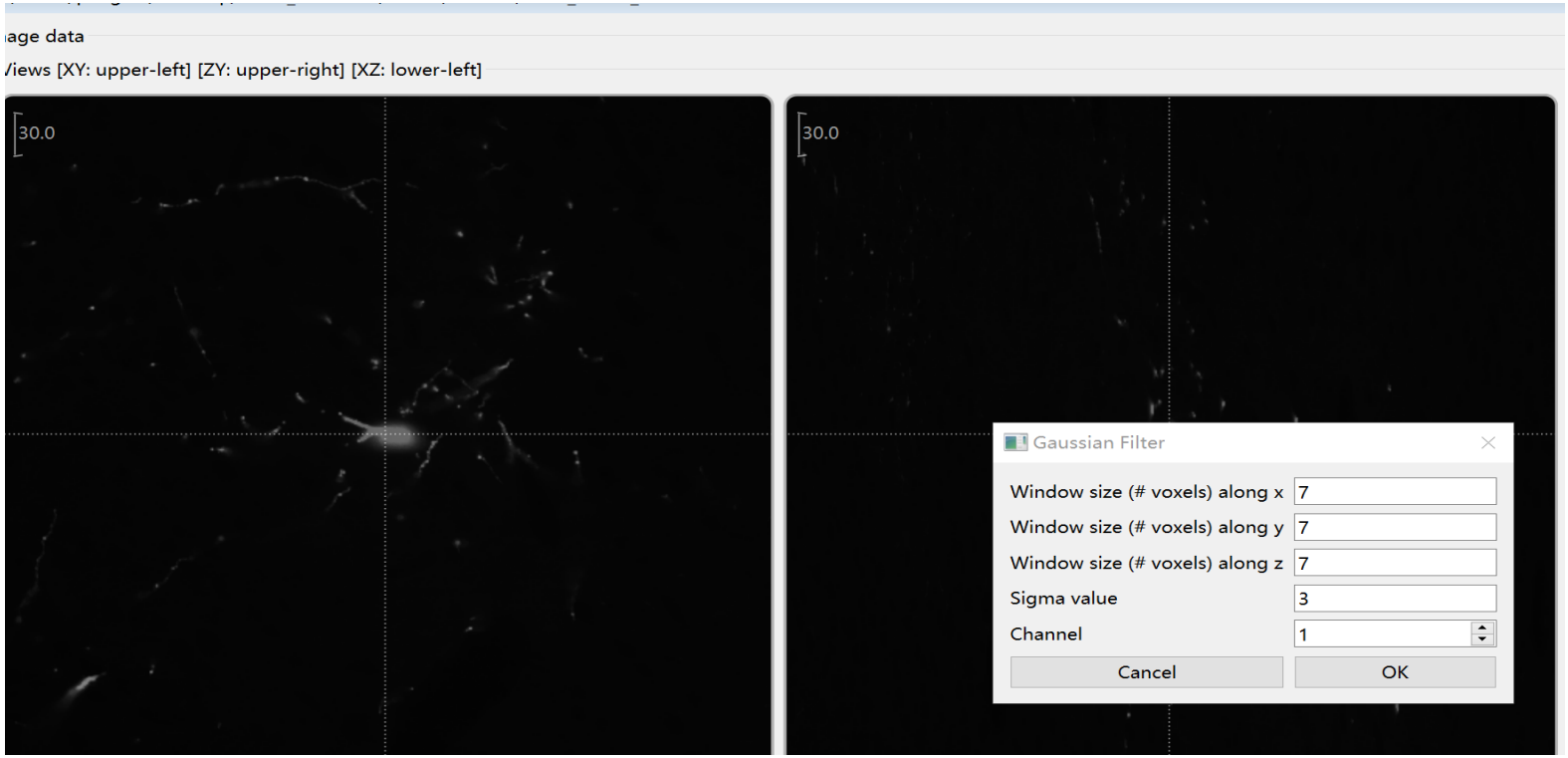
.cpp file

```
//Q_EXPORT_PLUGIN2(gaussianfilter, GaussianFilterPlugin)

void processImage(V3DPluginCallback2 &callback, QWidget *parent);
bool processImage(V3DPluginCallback2 &callback, const V3DPluginArgList & input,
template <class T> void gaussian_filter(T* data1d,
```

Windows:

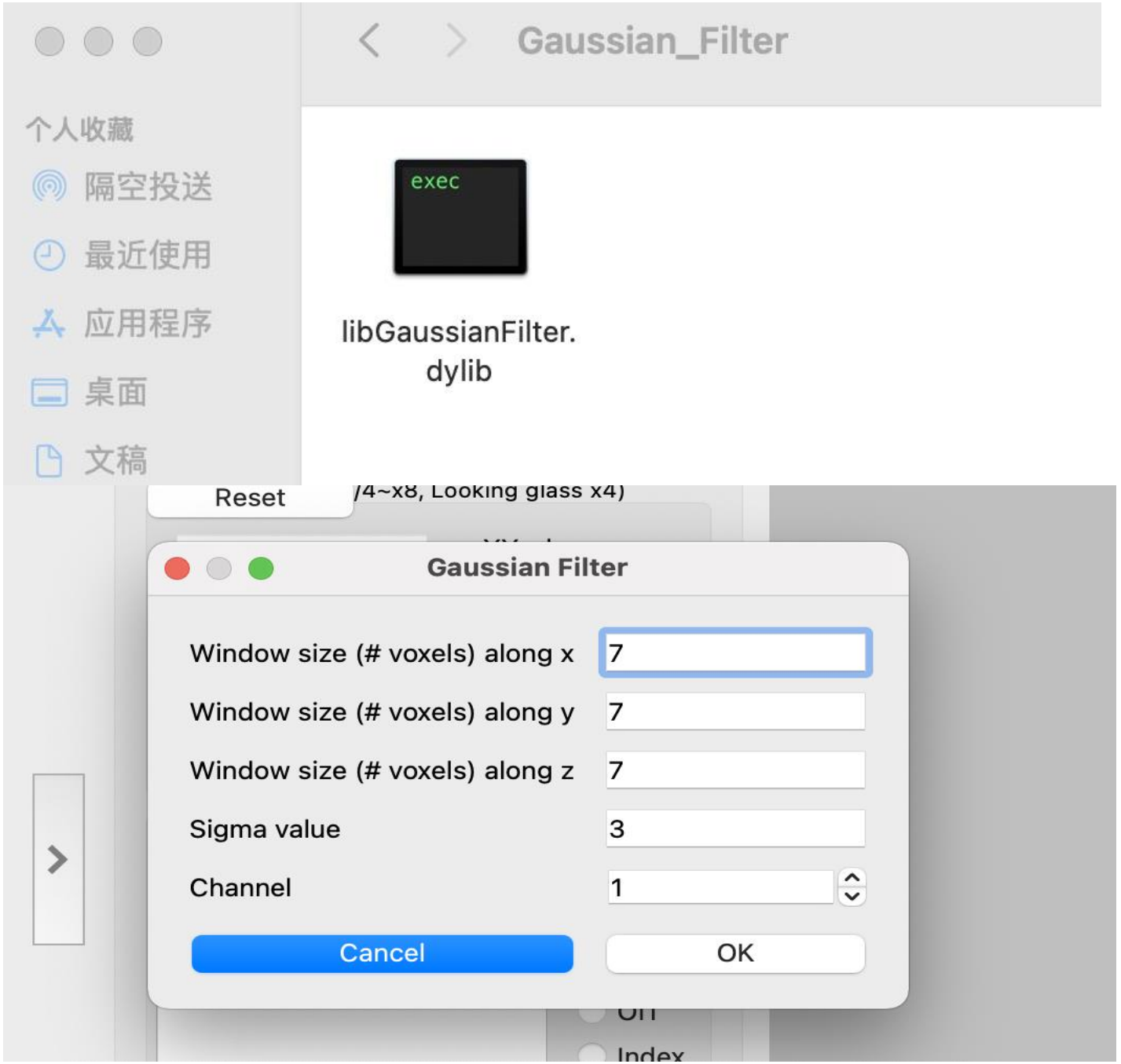
After build, the .dll and .a files will be automatically generated in the *bin* directory

The plugin will be ready for use



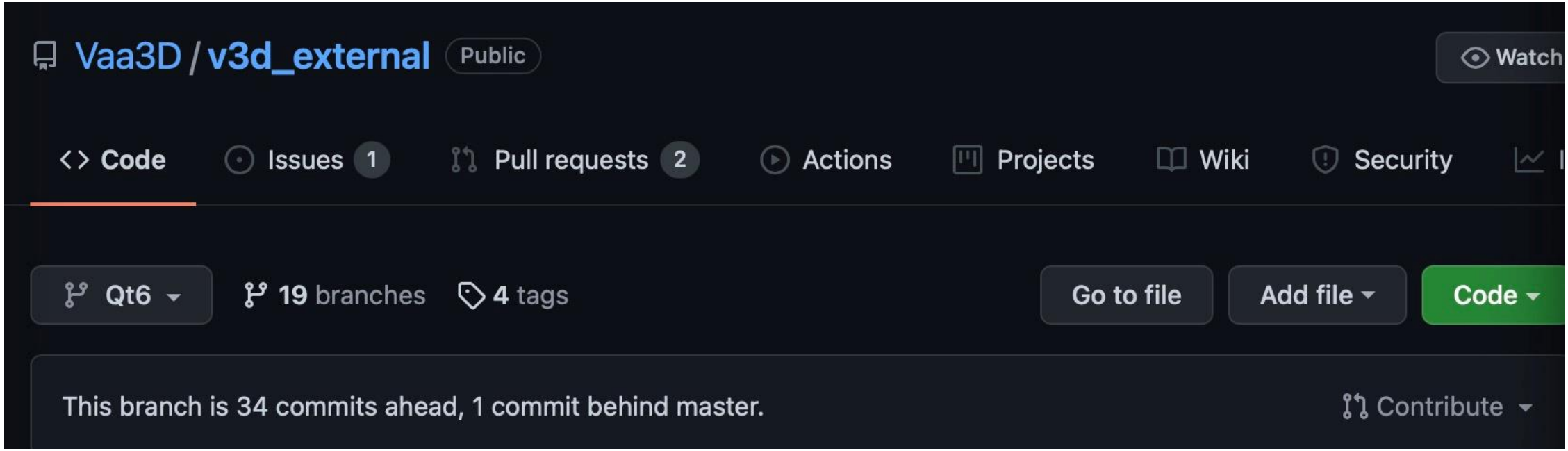
Mac

The steps to compile the plugin for Mac are exactly the same as for windows, generating a dylib type dynamic library.



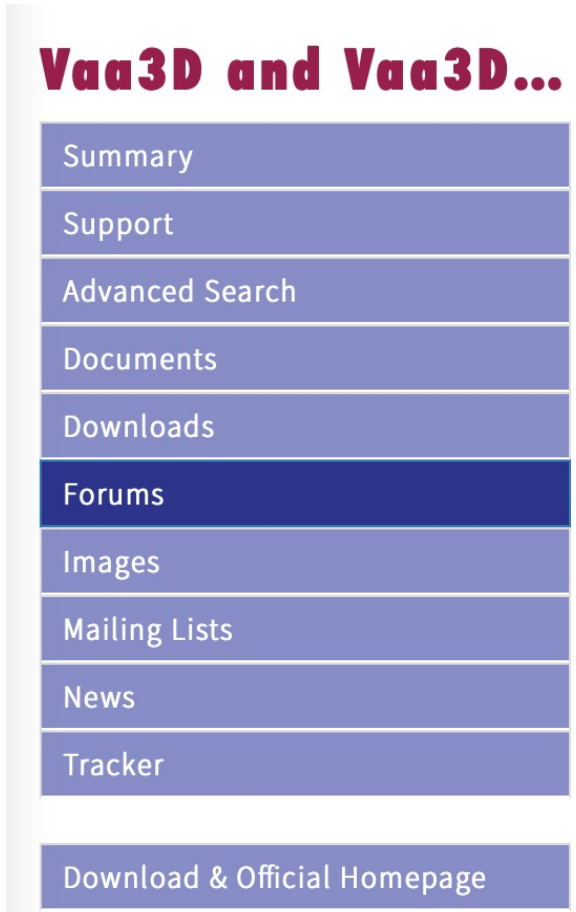
Qt6 Vaa3d

https://github.com/Vaa3D/v3d_external/tree/Qt6



Forum

https://www.nitrc.org



BROWSE FORUMS HELP

help

66 Subscribers

Showing 1-25 of 618 topics

Start New Topic

Display: Active Topics

Results per page: Show 25

1 2 ... 25 Next >

Topic	Author	Replies	Last Post
Failed Start on Windows	emanuels	1	Nov 4, 2021

Thanks !