

# A Simple OpenCASCADE Qt Demo-occQt

[eryar@163.com](mailto:eryar@163.com)

**Abstract.** OpenCASCADE have provided the Qt samples in the samples directory, but they are a little complicated. So I decide write a simple OpenCASCADE Qt demo for the OpenCASCADE beginners.

**Key Words.** OpenCASCADE6.8.0, Qt5.4

## 1. Introduction

OpenCASCADE is a software development platform providing services for 3D surface and solid modeling, CAD data exchange, and visualization. Most of OCCT functionality is available in the form of C++ libraries. OCCT can be best applied in development of software dealing with 3D modeling(CAD), manufacturing/measuring(CAM) or numerical simulation(CAE).

OpenCASCADE Technology is free software; you can redistribute it and or modify it under the terms of the GNU Lesser General Public License(LGPL) version 2.1, with additional exception. You can get the OpenCASCADE from here: <http://www.opencascade.org/>



Figure 1.1 OpenCASCADE6.8.0 Release Notes

Qt is a cross-platform application framework that is widely used for developing application software with graphical user interface(GUI). Qt is free and open source software distributed under the terms of the GNU Lesser General Public License.

### Qt Services

Qt is a powerful framework that lets you create stunning user interfaces with amazing performance. We want to help you to achieve optimal results by using Qt building blocks in the best possible way, which is why we offer an extensive range of Qt advisory services.

Code Less, Create More. Easily create connected devices, UIs and applications that run anywhere

on any device.



Figure 1.2 Qt Ads

You can develop your program in the following manner:

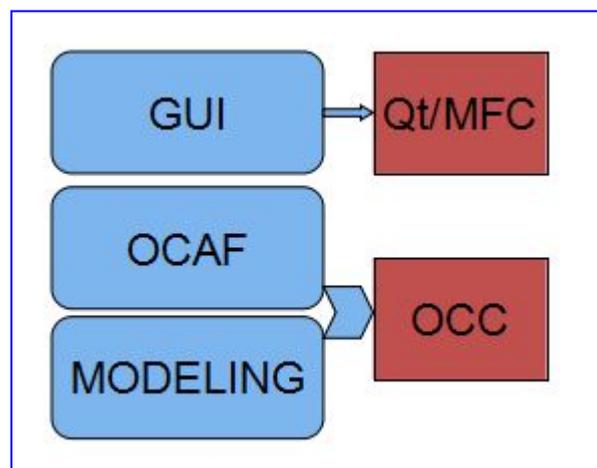


Figure 1.3 A application based on Qt and OpenCASCADE

You can use Qt for the GUI development, and for data management such as restore data from file or undo/redo support, you can choose OCAF, and for modeling algorithms, you can use OpenCASCADE.

You can also use the Model/View/Controller of Qt to manage the data and undo/redo mechanism, to accelerate the developing.

OpenCASCADE have provided many samples to show its functions, but there are a lots codes, it is a little complicated for the beginner. I write a simple program to try OpenCASCADE in a straightforward way. I hope the demo can help you to understand the usage of OpenCASCADE.

The demo is based on OpenCASCADE6.8.0 and Qt5.4. Because of there are many bug fixes in OpenCASCADE new version6.8.0, especially the visualization module. So I update the code for the previous demo Qt with OpenCASCADE and add something new:

<http://www.cppblog.com/eryar/archive/2013/08/18/202617.aspx>

## 2. View Operations

In occQt there are three operations to manipulate the view: pan, zoom and rotate, the following picture is the main window of occQt:

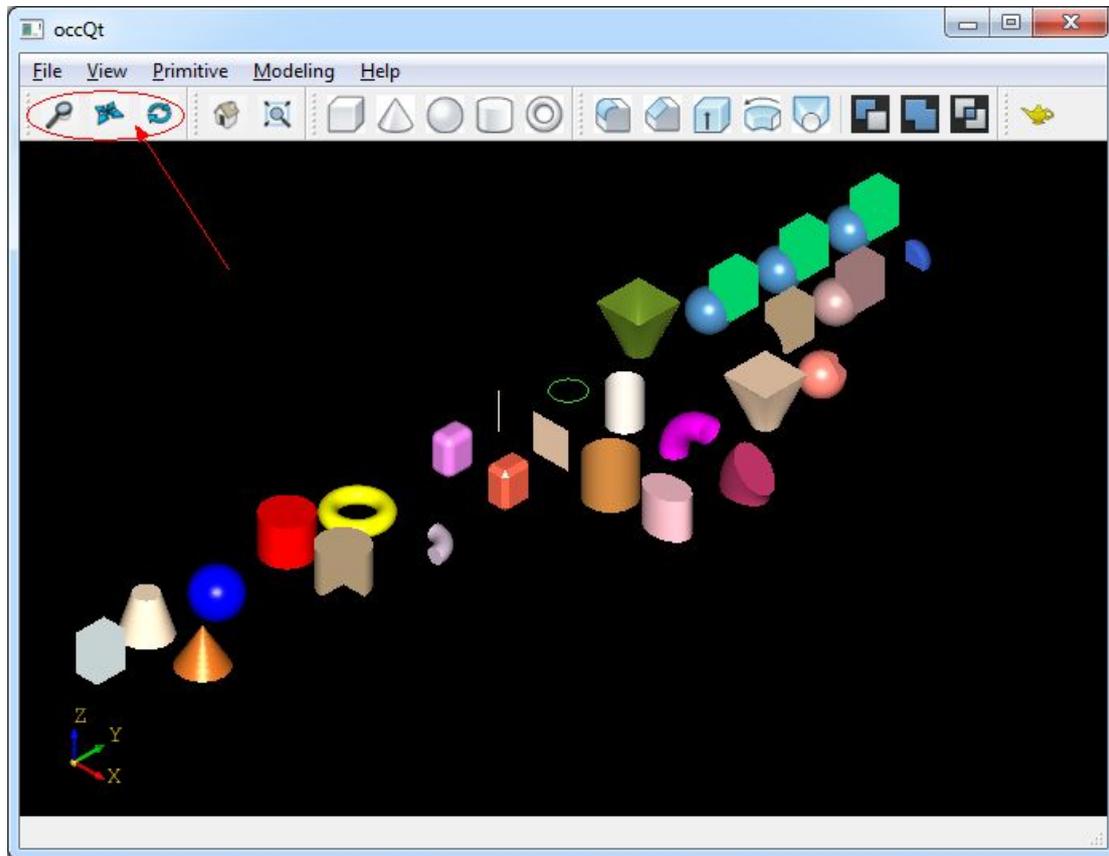


Figure 2.1 occQt GUI

The options is for the middle button of the mouse, you can dragged the middle button of the mouse to pan, zoom and rotate the view. And the mouse left button is used for the pick and multi-select operation, single click you can pick a shape; drag left button will select all the shapes in the rectangle.

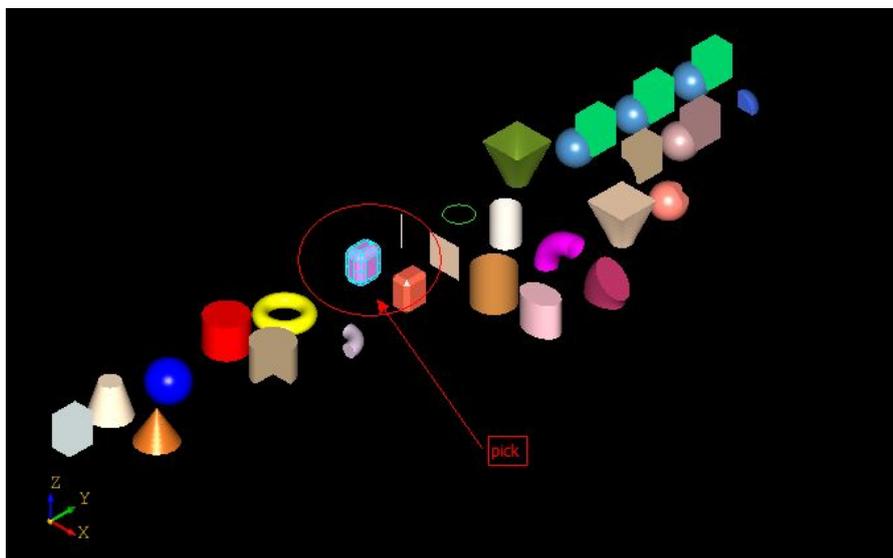


Figure 2.2 Click mouse left button to pick

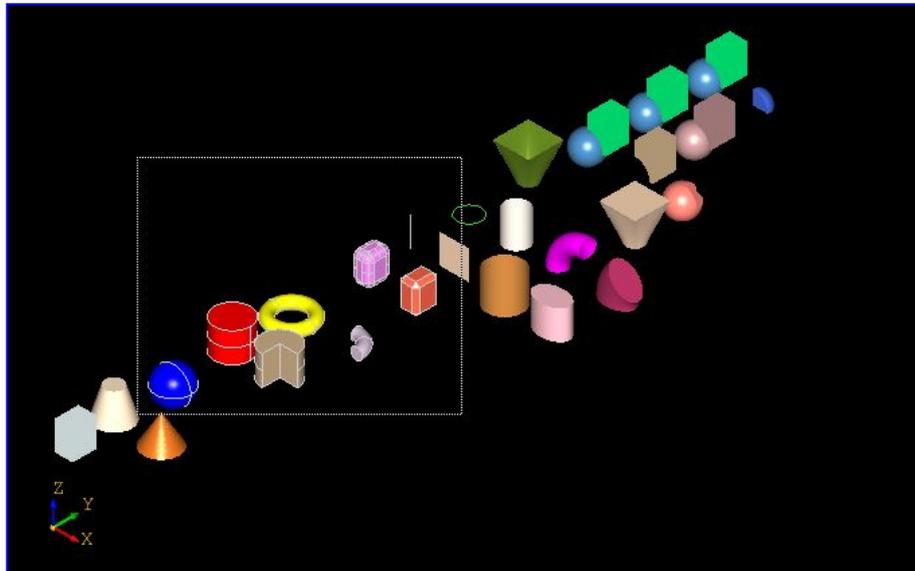


Figure 2.3 Drag mouse left button to multi-select

The selected shapes will be highlighted, but now do not provide any operations for the selected shapes.

And also provide the Reset and Fit All function for the view:

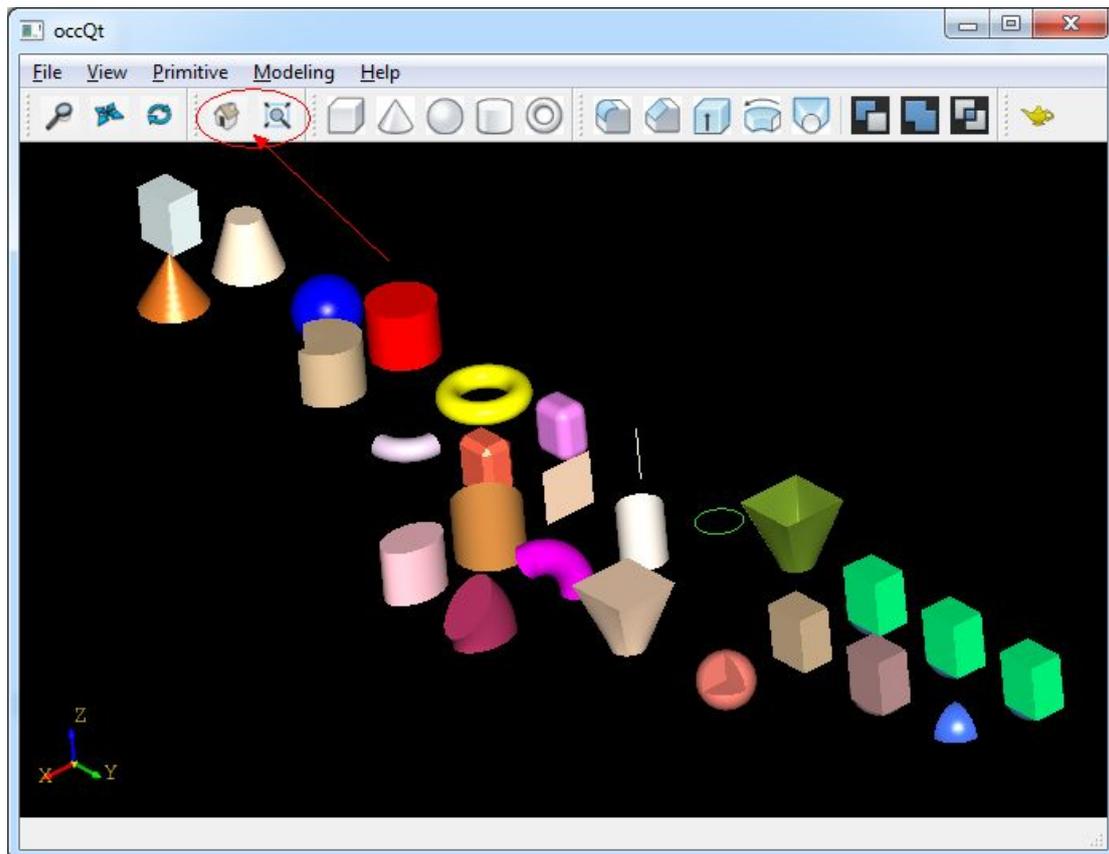


Figure 2.4 Reset and Fit all for the view

### 3. Make Primitives

OpenCASCADE provides simple API for the construction of primitives, such as box, cone, sphere, cylinder, and torus, .etc. You can make a primitive by a simple class in OpenCASCADE, the sample code as follows:

```
TopoDS_Shape aTopoBox = BRepPrimAPI_MakeBox(3.0, 4.0, 5.0);  
TopoDS_Shape aTopoCone = BRepPrimAPI_MakeCone(3.0, 0.0, 5.0);  
TopoDS_Shape aTopoSphere = BRepPrimAPI_MakeSphere(3.0);  
TopoDS_Shape aTopoCylinder = BRepPrimAPI_MakeCylinder(3.0, 5.0);  
TopoDS_Shape aTopoTorus = BRepPrimAPI_MakeTorus(3.0, 1.0);
```

And the shapes in the viewer is in the following picture:

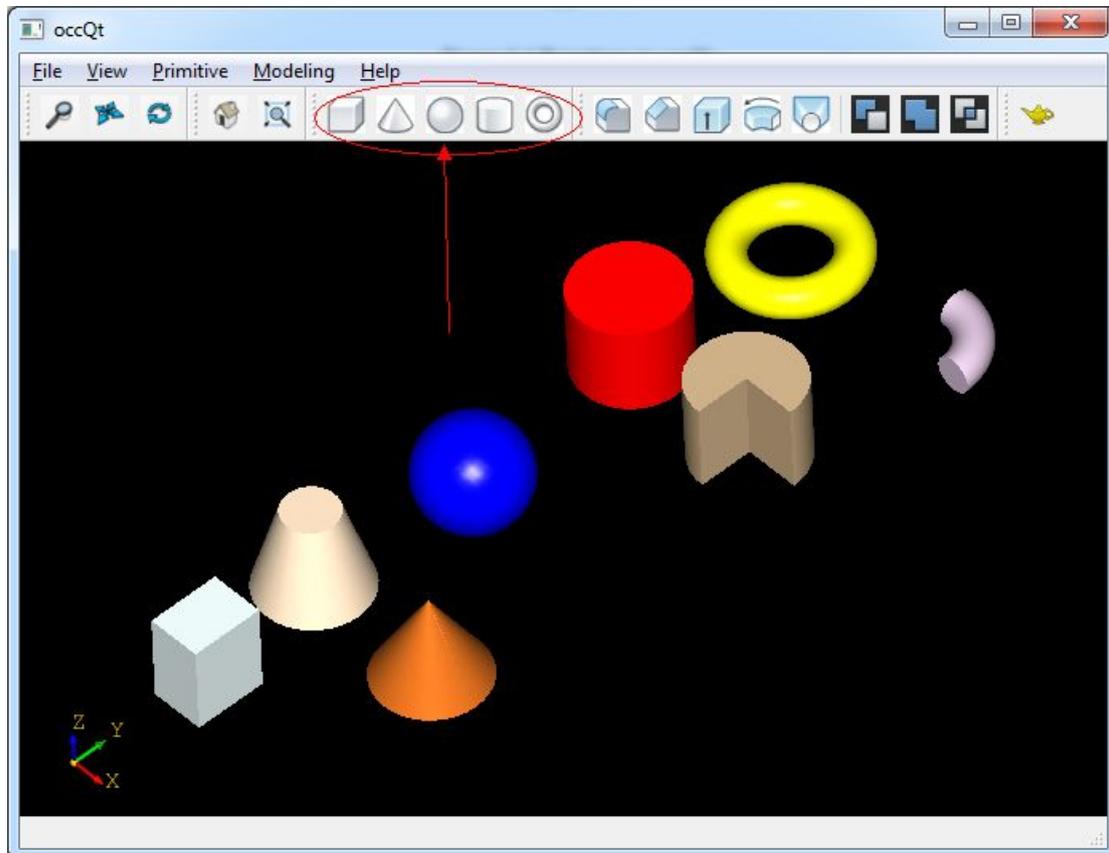


Figure 3.1 Primitives in occQt

#### 4. Modeling Algorithms

Modeling algorithms is the outstanding advantage of OpenCASCADE. But in occQt just give an example for the usage, you can check the documents and OpenCASCADE reference manual to have a look. The modeling algorithms in the occQt are:

- ❖ Fillet;
- ❖ Chamfer;
- ❖ Extrude;
- ❖ Revol;
- ❖ loft.

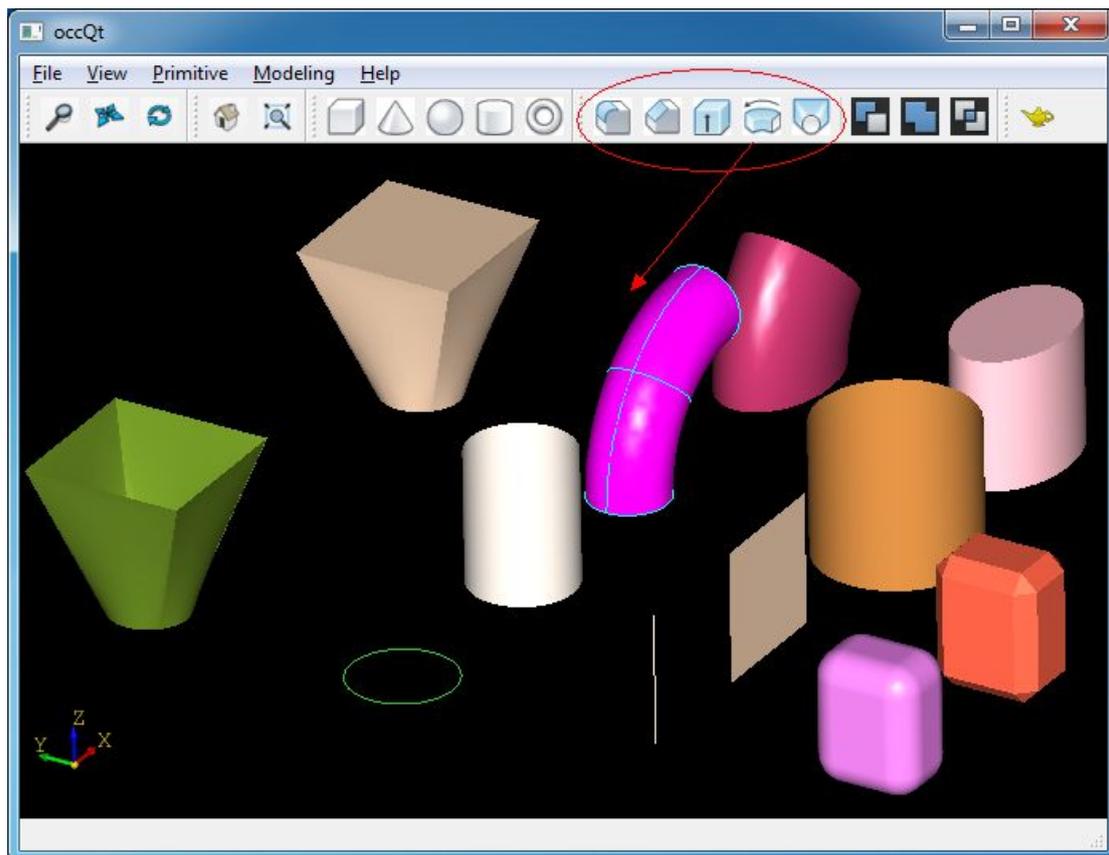


Figure 4.1 Modeling Algorithms in occQt

## 5. Boolean Operations

Boolean operations are used to create new shapes from the combinations of two shapes,

Operation	Result
Fuse	all points in S1 or S2
Common	all points in S1 and S2
Cut S1 by S2	all points in S1 and not in S2

The class to perform this is simple to use, they are:

```
TopoDS_Shape aFusedShape = BRepAlgoAPI_Fuse(S1, S2);  
TopoDS_Shape aCommonShape = BRepAlgoAPI_Common(S1, S2);  
TopoDS_Shape aCuttedShape = BRepAlgoAPI_Cut(S1, S2);
```

You can try this in occQt:

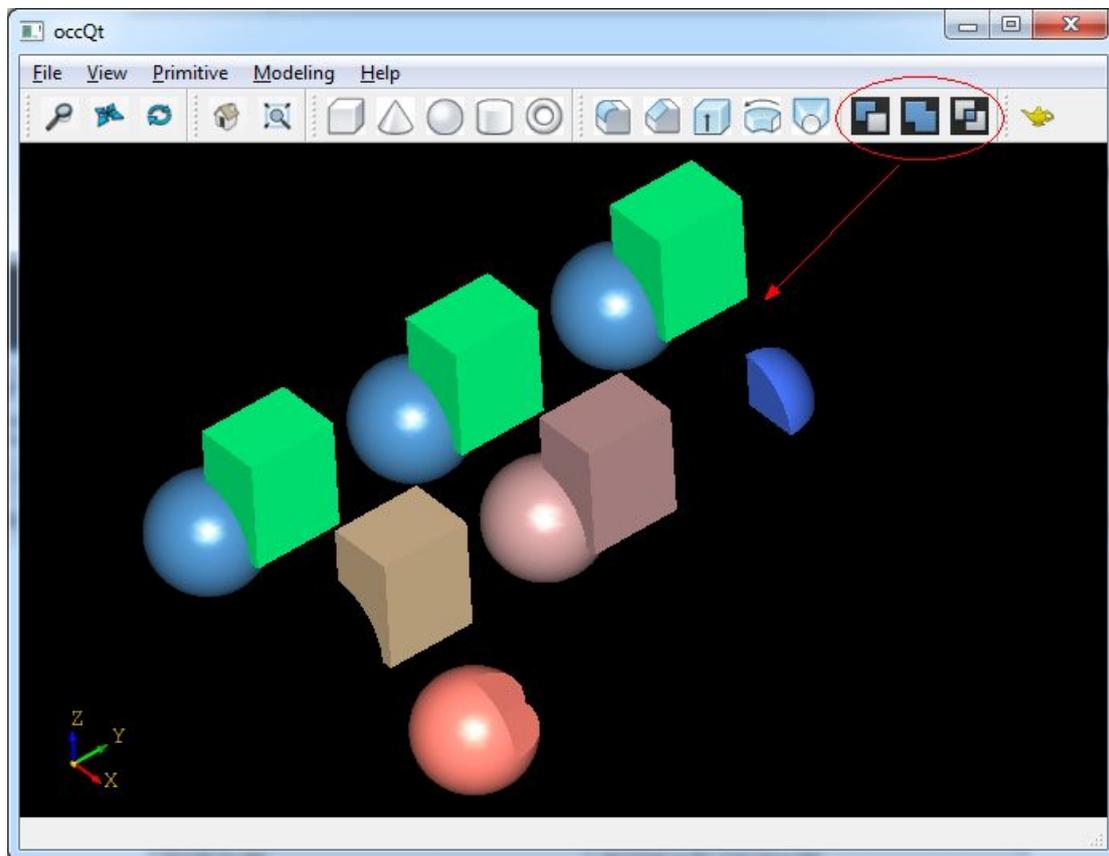


Figure 5.1 Boolean operations in occQt

## 6. Conclusion

From the simple code demo, I hope it will help you to understand OpenCASCADE easily. Thanks for OpenCASCADE and Qt the amazing work.

I put the code to GitHub: <https://github.com/eryar/occQt/>. You can download and have a try, then you can feedback or track the modification of occQt.

Enjoy!