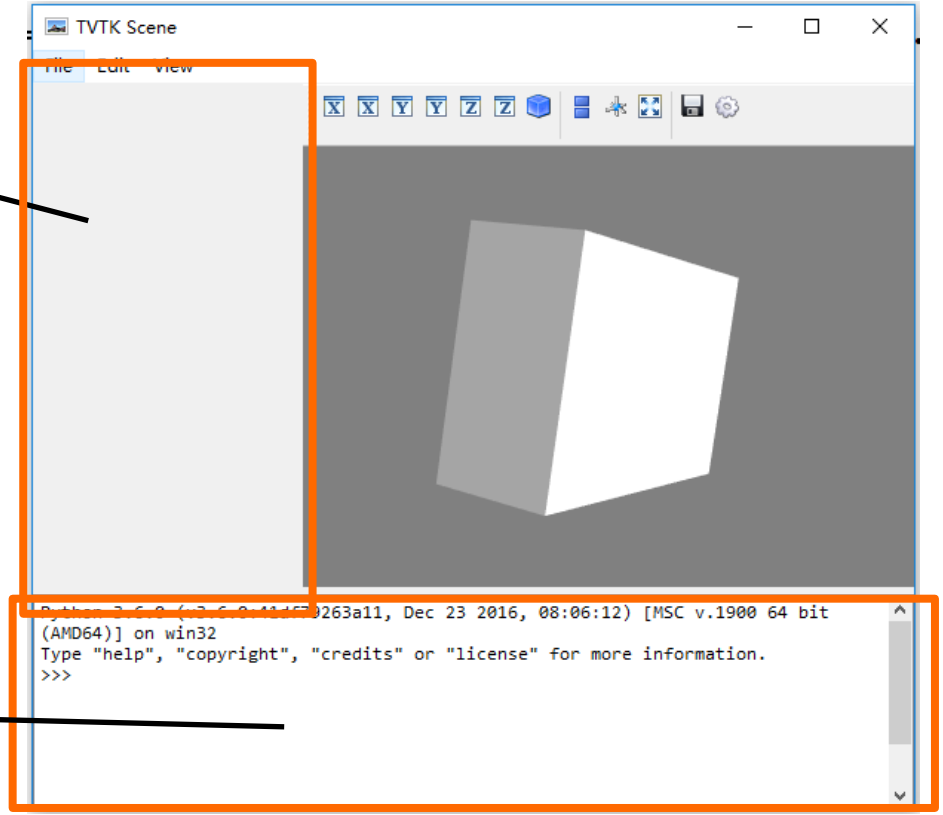
用ivtk工具观察管线

```
from tvtk.api import ivtk
```

使用ivtk显示立方体的程序

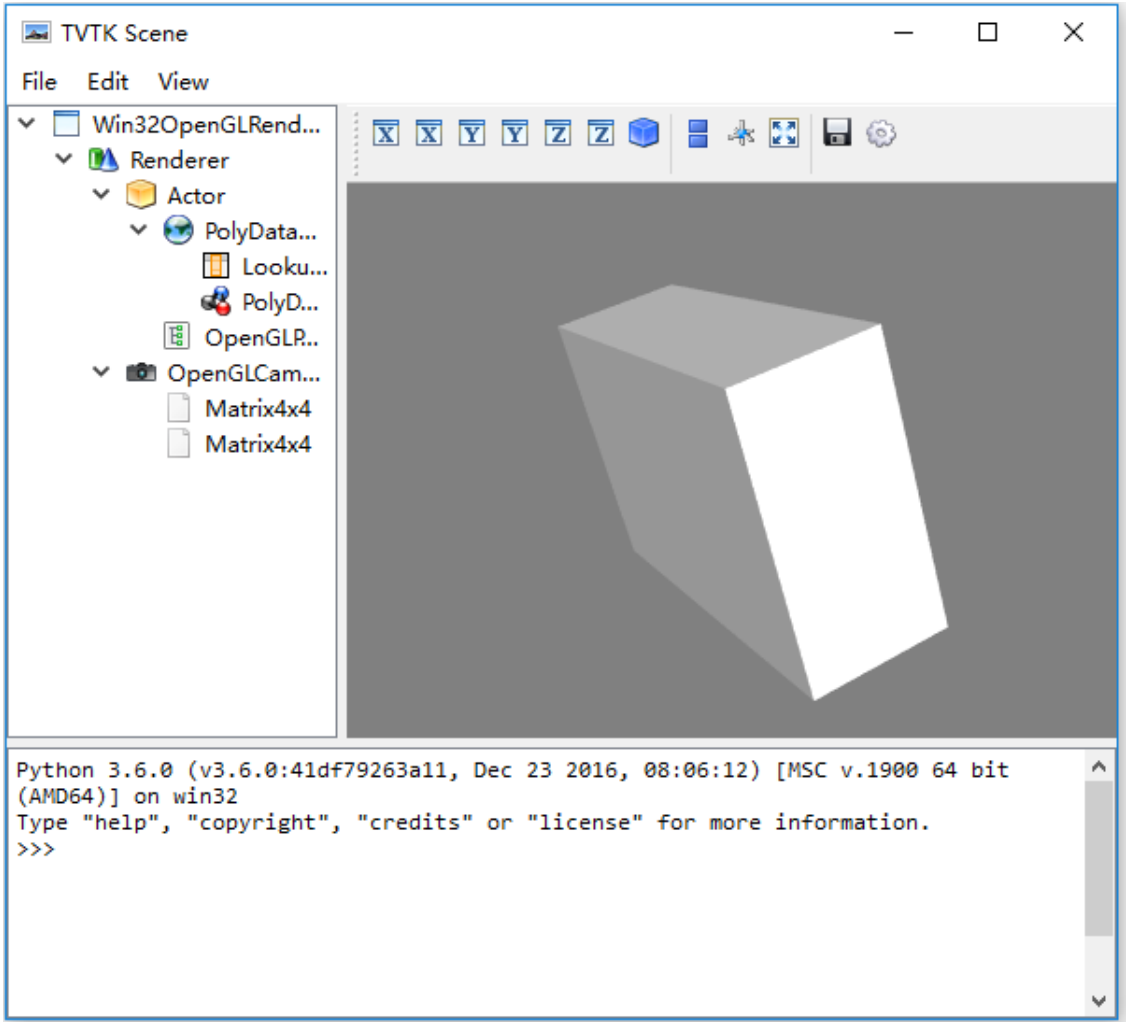
```
from tvtk.api import tvtk
from tvtk.tools import ivtk
from pyface.api import GUI

s = tvtk.CubeSource(x_length=1.0, y_length=2.0, z_length=3.0)
m = tvtk.PolyDataMapper(input_connection=s.output_port)
a = tvtk.Actor(mapper=m)

#创建一个带Crust（Python Shell）的窗口
gui = GUI()
win = ivtk.IVTKWithCrustAndBrowser()
win.open()
win.scene.add_actor(a)

#开始界面消息循环
gui.start_event_loop()
```

修正窗口显示错误



Shell

```python
from tvtk.api import tvtk
from tvtk.tools import ivtk
from pyface.api import GUI

s = tvtk.CubeSource(x_length=1.0, y_length=2.0, z_length=3.0)
m = tvtk.PolyDataMapper(input_connection=s.output_port)
a = tvtk.Actor(mapper=m)

#创建一个带Crust（Python Shell）的窗口
gui = GUI()
win = ivtk.IVTKWithCrustAndBrowser()
win.open()
win.scene.add_actor(a)

#修正错误
dialog = win.control.centralWidget().widget(0).widget(0)
from pyface.qt import QtCore
dialog.setWindowFlags(QtCore.Qt.WindowFlags(0x00000000))
dialog.show()

#开始界面消息循环
gui.start_event_loop()
```


TVTK Scene
File
Edit
View
Win32OpenGLRend...
Renderer
Actor
PolyData...
Looku...
PolyD...
OpenGLP...
OpenGLCam...
Matrix4x4
Matrix4x4
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 08:06:12) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>

```python
from tvtk.api import tvtk

def ivtk_scene(actors):
    from tvtk.tools import ivtk
    #创建一个带Crust（Python Shell）的窗口
    win = ivtk.IVTKWithCrustAndBrowser()
    win.open()
    win.scene.add_actor(actors)
    #修正窗口错误
    dialog = win.control.centralWidget().widget(0).widget(0)
    from pyface.qt import QtCore
    dialog.setWindowFlags(QtCore.Qt.WindowFlags(0x00000000))
    dialog.show()
    return win

def event_loop():
    from pyface.api import GUI
    gui = GUI()
    gui.start_event_loop()

s = tvtk.CubeSource(x_length=1.0, y_length=2.0, z_length=3.0)
m = tvtk.PolyDataMapper(input_connection=s.output_port)
a = tvtk.Actor(mapper=m)
win = ivtk_scene(a)
win.scene.isometric_view()
event_loop()
```

cube_ivtk_func.py

```python
from tvtk.api import tvtk

def ivtk_scene(actors):
    from tvtk.tools import ivtk
    #创建一个带Crust（Python Shell）的窗口
    win = ivtk.IVTKWithCrustAndBrowser()
    win.open()
    win.scene.add_actor(actors)
    #修正窗口错误
    dialog = win.control.centralWidget().widget(0).widget(0)
    from pyface.qt import QtCore
    dialog.setWindowFlags(QtCore.Qt.WindowFlags(0x00000000))
    dialog.show()
    return win

def event_loop():
    from pyface.api import GUI
    gui = GUI()
    gui.start_event_loop()

s = tvtk.CubeSource(x_length=1.0, y_length=2.0, z_length=3.0)
m = tvtk.PolyDataMapper(input_connection=s.output_port)
a = tvtk.Actor(mapper=m)
win = ivtk_scene(a)
win.scene.isometric_view()
event_loop()
```

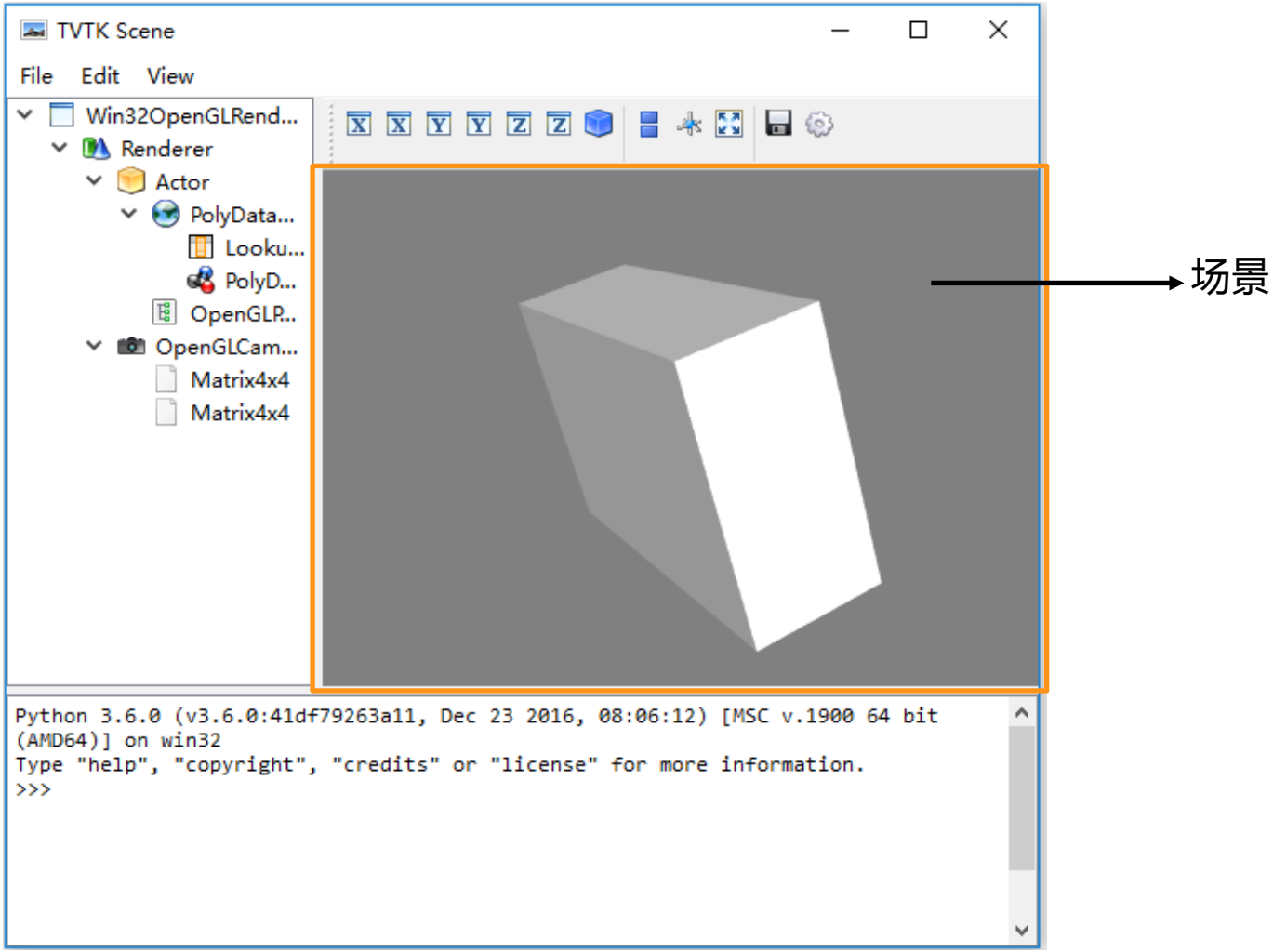
tvtkfunc.py

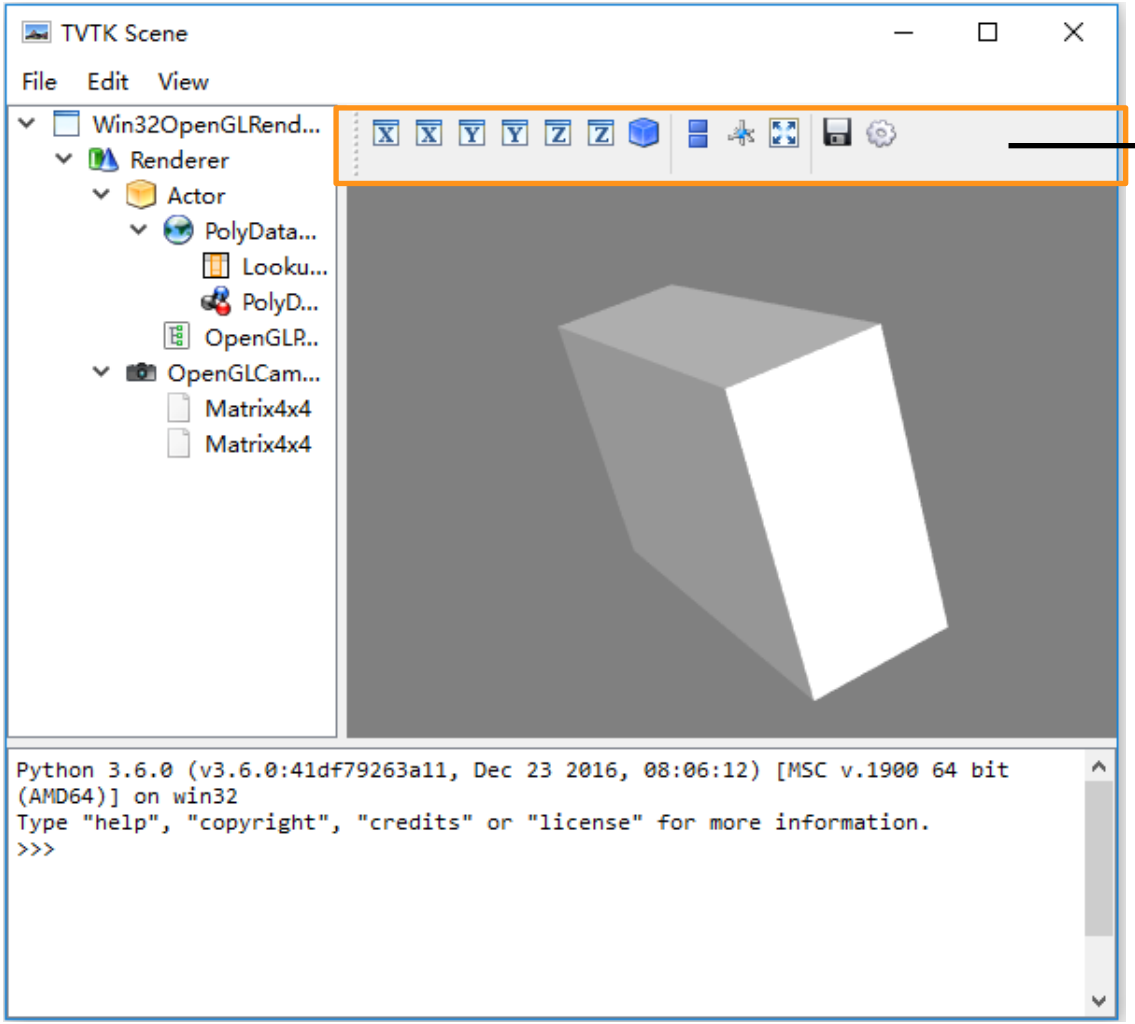
```python
def ivtk_scene(actors):
    from tvtk.tools import ivtk
    #创建一个带Crust（Python Shell）的窗口
    win = ivtk.IVTKWithCrustAndBrowser()
    win.open()
    win.scene.add_actor(actors)
    #修正窗口错误
    dialog = win.control.centralWidget().widget(0).widget(0)
    from pyface.qt import QtCore
    dialog.setWindowFlags(QtCore.Qt.WindowFlags(0x00000000))
    dialog.show()
    return win

def event_loop():
    from pyface.api import GUI
    gui = GUI()
    gui.start_event_loop()
```

from tvtkfunc import ……

cube_ivtk.py

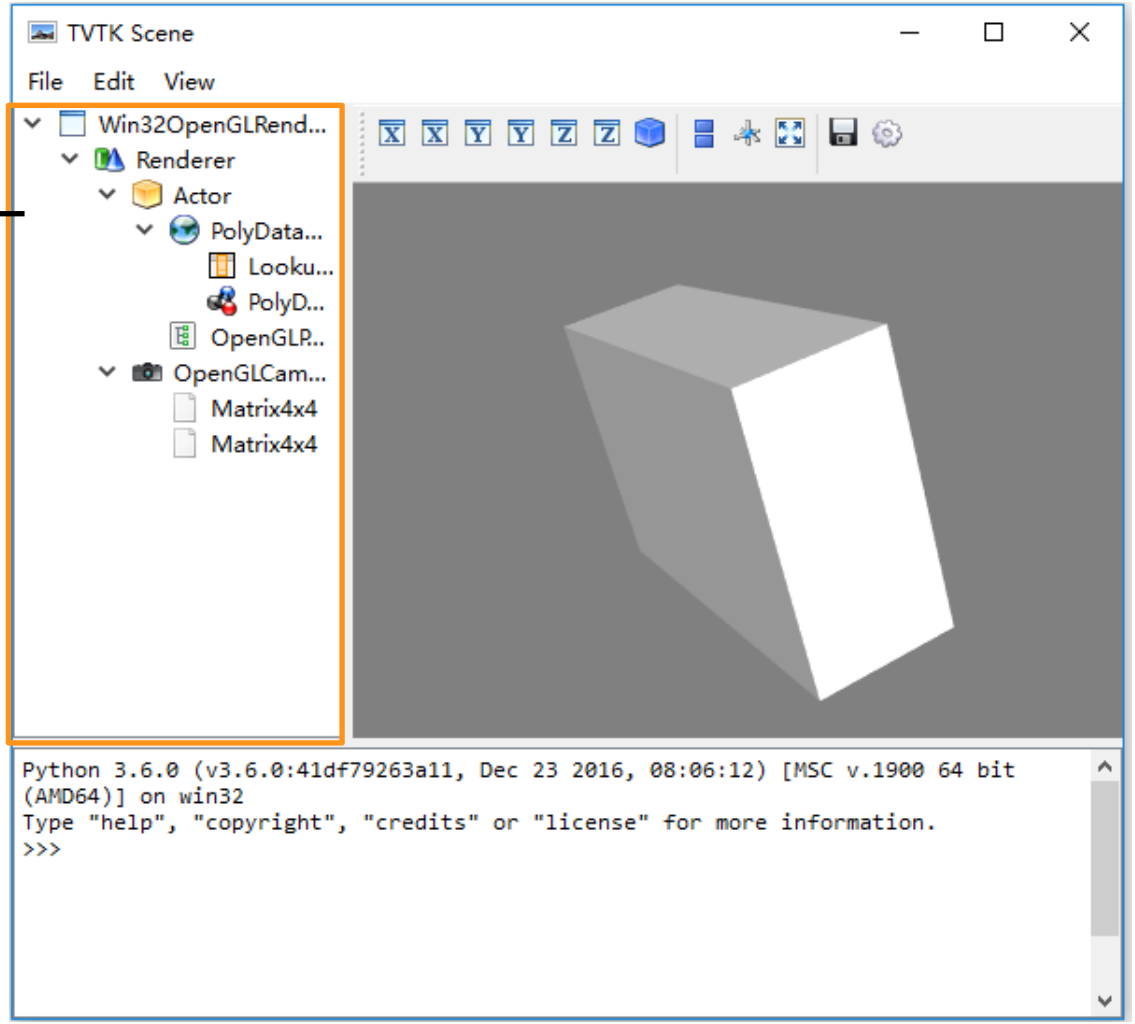
```python
from tvtk.api import tvtk
from tvtkfunc import ivtk_scene,event_loop

s = tvtk.CubeSource(x_length=1.0, y_length=2.0, z_length=3.0)
m = tvtk.PolyDataMapper(input_connection=s.output_port)
a = tvtk.Actor(mapper=m)
win = ivtk_scene(a)
win.scene.isometric_view()
event_loop()
```

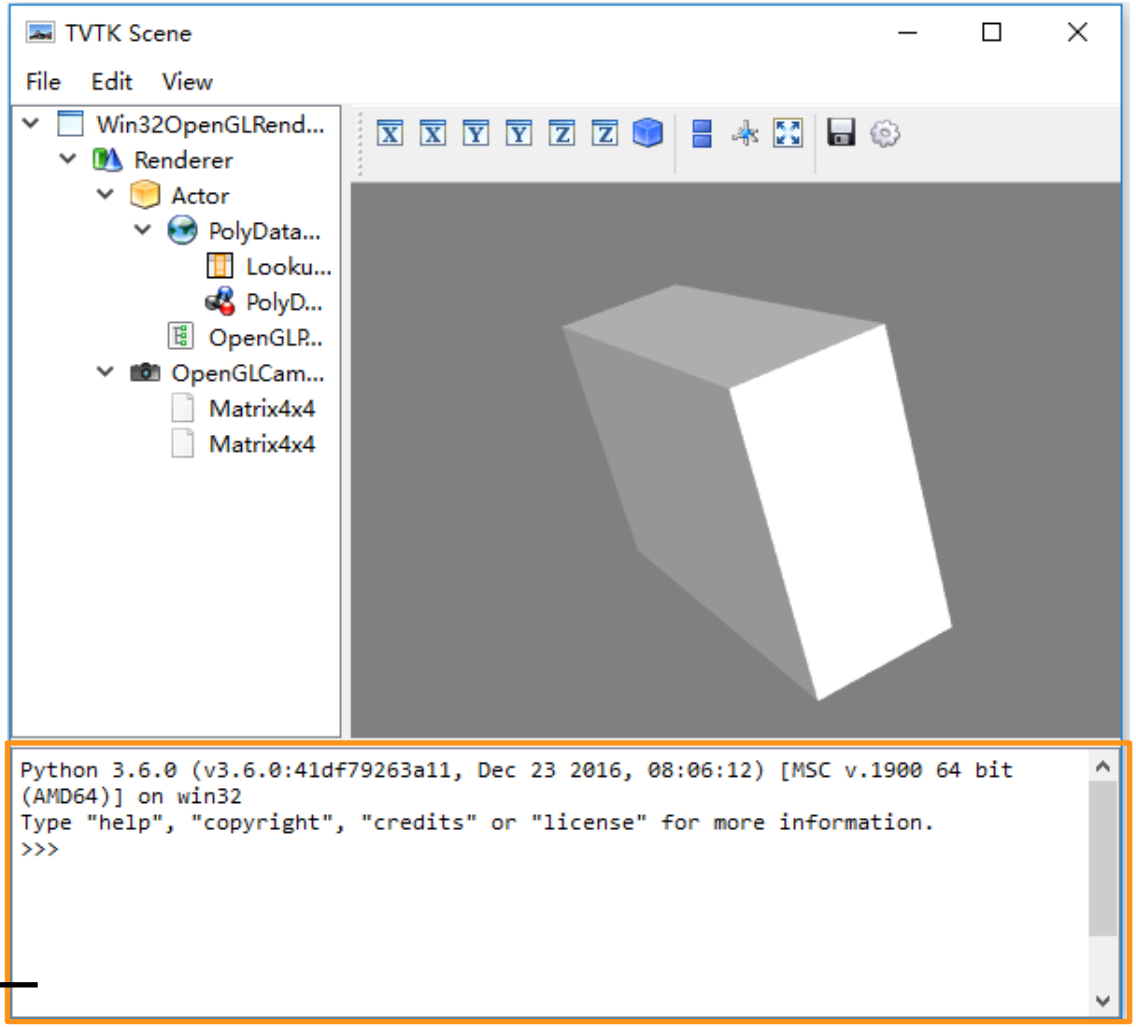

TVTK Scene
File
Edit
View
Win32OpenGLRend...
Renderer
Actor
PolyData...
Looku...
PolyD...
OpenGLP...
OpenGLCam...
Matrix4x4
Matrix4x4
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 08:06:12) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
场景


TVTK Scene
File
Edit
View
Win32OpenGLRend...
Renderer
Actor
PolyData...
Looku...
PolyD...
OpenGLP...
OpenGLCam...
Matrix4x4
Matrix4x4
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 08:06:12) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>

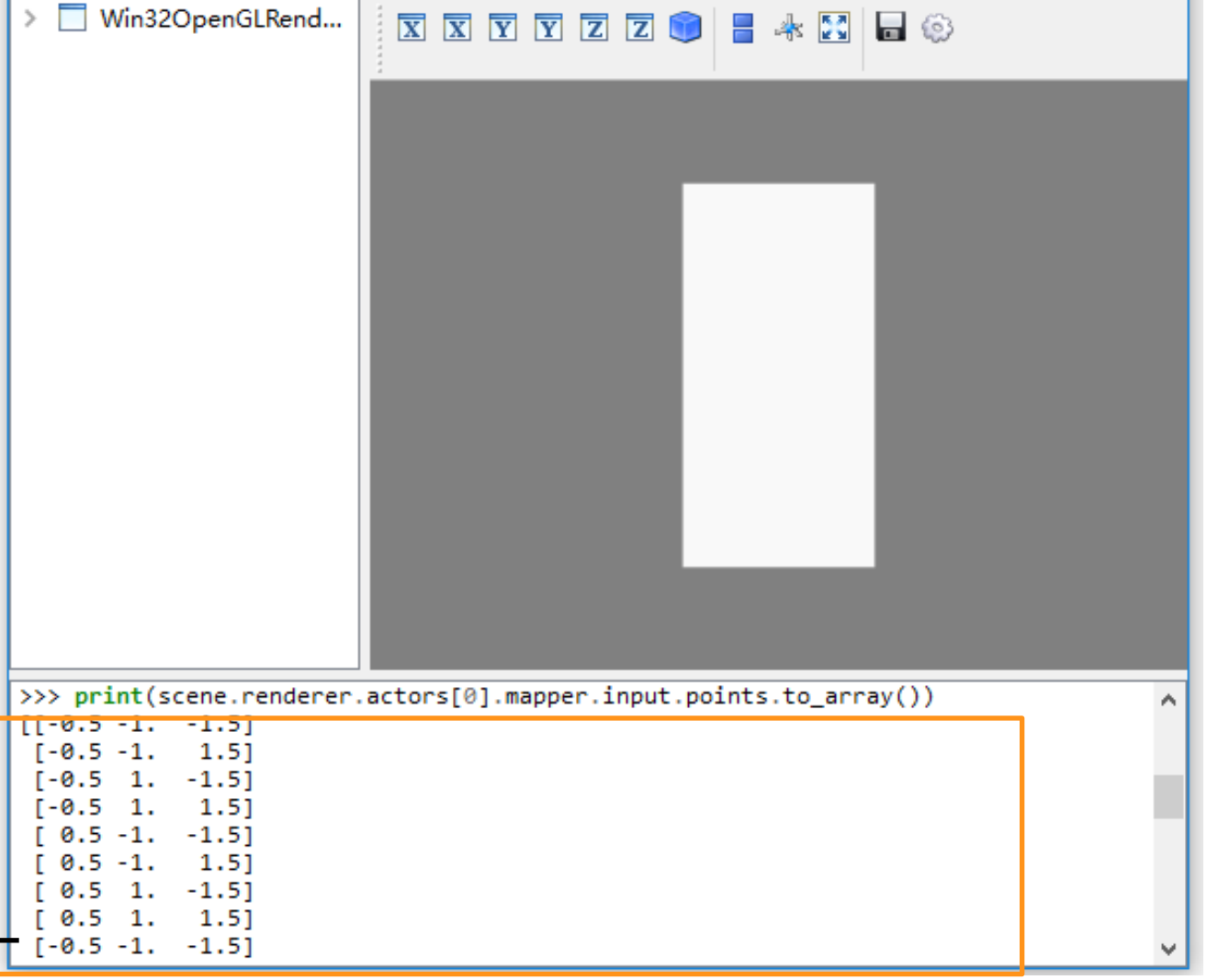
场景工具条

管线浏览器


TVTK Scene
File
Edit
View
Win32OpenGLRend...
Renderer
Actor
PolyData...
Looku...
PolyD...
OpenGLP...
OpenGLCam...
Matrix4x4
Matrix4x4
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 08:06:12) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>

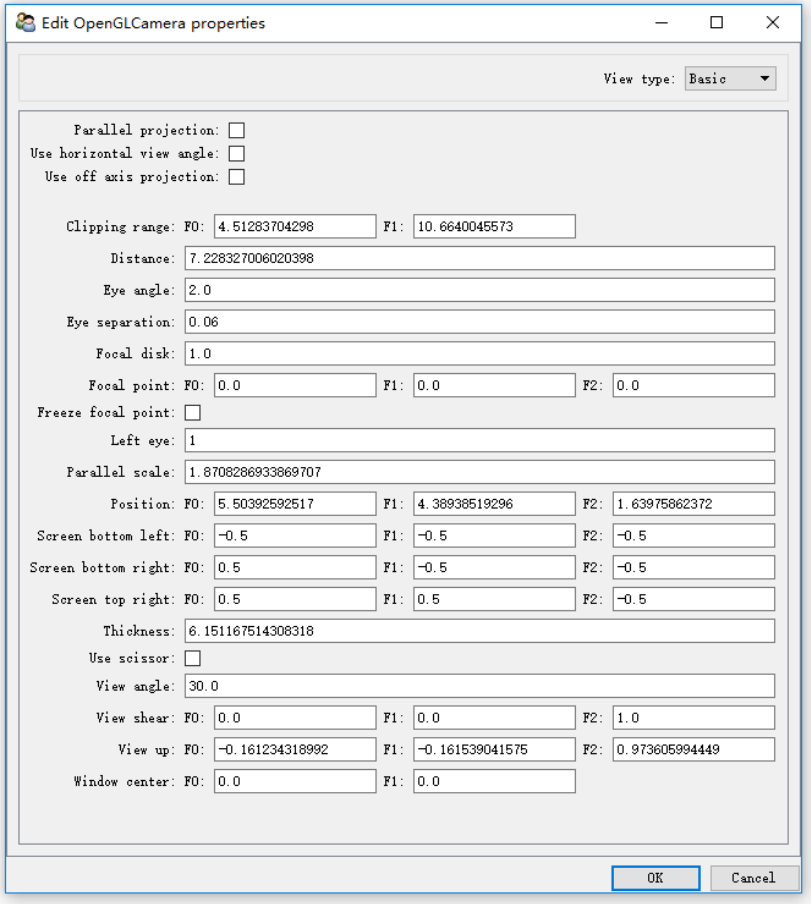

TVTK Scene
File
Edit
View
Win32OpenGLRend...
Renderer
Actor
PolyData...
Looku...
PolyD...
OpenGLP...
OpenGLCam...
Matrix4x4
Matrix4x4
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 08:06:12) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>

Python命令行



构成长方体顶点的三维坐标。

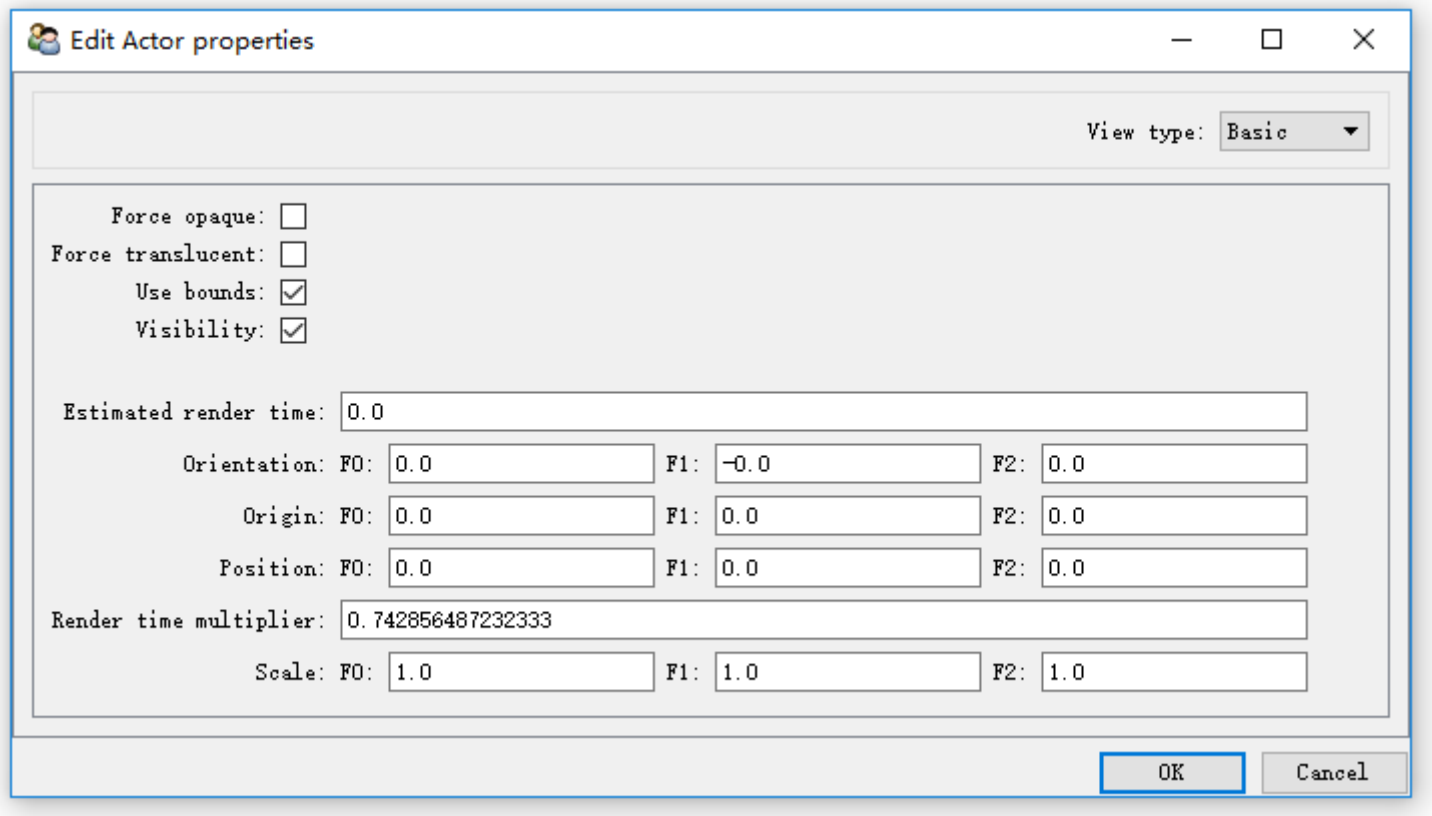
照相机


Edit OpenGLCamera properties
View type: Basic
Parallel projection:
Use horizontal view angle:
Use off axis projection:
Clipping range: F0: 4.51283704298 F1: 10.6640045573
Distance: 7.228327006020398
Eye angle: 2.0
Eye separation: 0.06
Focal disk: 1.0
Focal point: F0: 0.0 F1: 0.0 F2: 0.0
Freeze focal point:
Left eye: 1
Parallel scale: 1.870828693386970 7
Position: F0: 5.50392592517 F1: 4.38938519296 F2: 1.63975862372
Screen bottom left: F0: -0.5 F1: -0.5 F2: -0.5
Screen bottom right: F0: 0.5 F1: -0.5 F2: -0.5
Screen top right: F0: 0.5 F1: 0.5 F2: -0.5
Thickness: 6.151167514308318
Use scissor:
View angle: 30.0
View shear: F0: 0.0 F1: 0.0 F2: 1.0
View up: F0: -0.161234318992 F1: -0.161539041575 F2: 0.973605994449
Window center: F0: 0.0 F1: 0.0
OK
Cancel

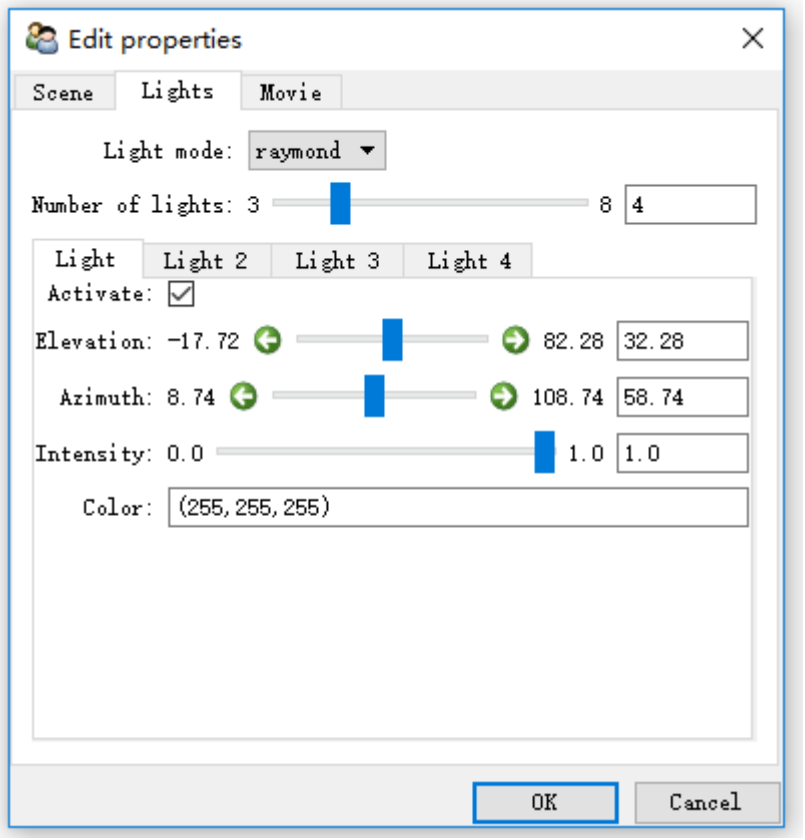
照相机属性

属性	说明
clipping_plane	它有两个元素，分别表示照相机到近、远两个裁剪平面的距离。在这两个平面范围之外将不会显示
position	照相机在三维空间中的坐标
focal_point	照相机所聚焦的焦点坐标
view_up	照相机的上方向矢量

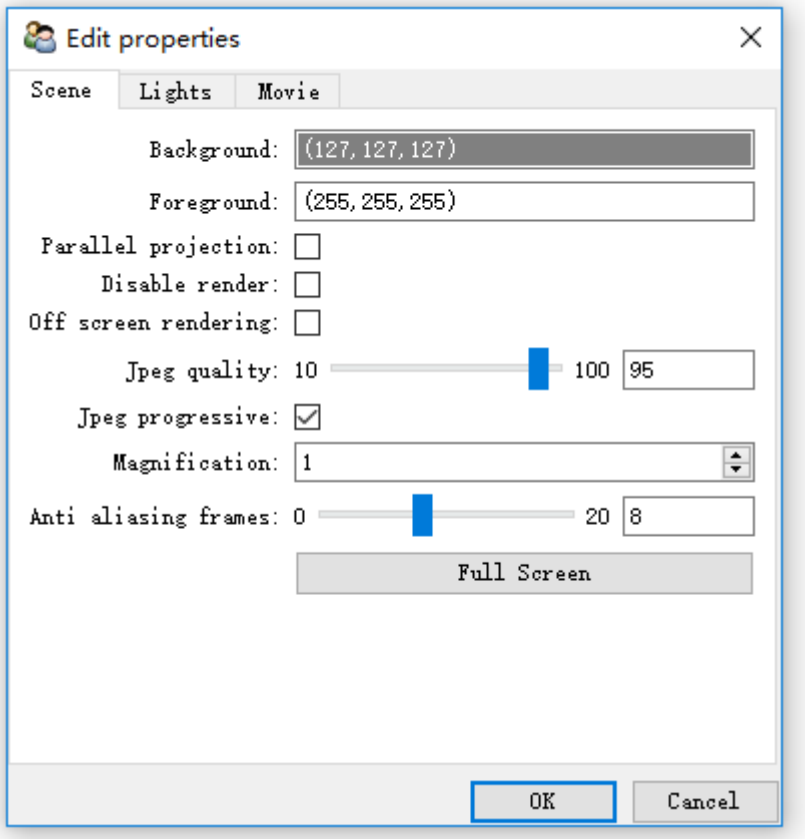
实体Actor


Edit Actor properties
View type: Basic
Force opaque:
Force translucent:
Use bounds:
Visibility:
Estimated render time: 0.0
Orientation: F0: 0.0 F1: -0.0 F2: 0.0
Origin: F0: 0.0 F1: 0.0 F2: 0.0
Position: F0: 0.0 F1: 0.0 F2: 0.0
Render time multiplier: 0.742856487232333
Scale: F0: 1.0 F1: 1.0 F2: 1.0
OK
Cancel

光源


Edit properties
Scene
Lights
Movie
Light mode:
raymond
Number of lights: 3
8
4
Light
Light 2
Light 3
Light 4
Activate:
Elevation: -17.72
82.28
32.28
Azimuth: 8.74
108.74
58.74
Intensity: 0.0
1.0
1.0
Color:
(255,255,255)
OK
Cancel

场景


Edit properties
Scene
Lights
Movie
Background:
(127, 127, 127)
Foreground:
(255, 255, 255)
Parallel projection:
Disable render:
Off screen rendering:
Jpeg quality:
10
100
95
Jpeg progressive:
Magnification:
1
Anti aliasing frames:
0
20
8
Full Screen
OK
Cancel