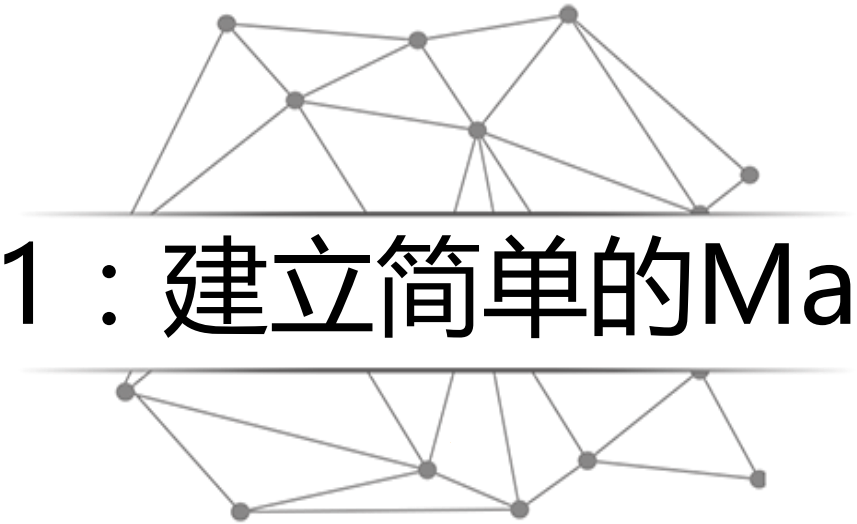
TraitsUI与Mayavi应用实例

黄天羽 嵩天
www.python123.org



实例1：建立简单的Mayavi窗口

建立mayavi窗口步骤

1、建立从HasTraits继承的类

1.1 建立MlabSceneModel场景实例scene

1.2 建立View视图

1.3 定义__init__函数，生成数据

2、建立类的实例，调用configure_traits()方法

建立简单的mayavi窗口框架

```
# 1.建立HasTraits继承类
class ActorViewer(HasTraits):
    # 1.1建立场景实例
    scene = Instance(MlabSceneModel, ())
    # 1.2 提供Mayavi视图窗口
    view = View(Item(name='scene',
                     editor=SceneEditor(scene_class=MayaviScene)
                     ......)
                )
    # 1.3 重载初始化函数
    def __init__(self, **traits):
        HasTraits.__init__(self, **traits)
        self.generate_data()#生成数据，并绘制
    def generate_data(self):

# 2. 显示窗口
a = ActorViewer()
a.configure_traits()
```

1.1建立场景实例

```
from traits.api import Instance
from mayavi.tools.mlab_scene_model import MlabSceneModel

scene = Instance(MlabSceneModel, ())
```

1.1建立场景实例

使用mlab进行可视化

```
from traits.api import Instance
from mayavi.tools.mlab_scene_model import MlabSceneModel

scene = Instance(MlabSceneModel, ())
def generate_data(self):
    ... ...# 生成数据
    self.scene.mlab.surf()#对数据进行可视化
```

1.2 提供Mayavi视图窗口

引入视图编辑器模块

```
from tvtk.pyface.scene_editor import SceneEditor
from mayavi.tools.mlab_scene_model import MlabSceneModel
```

1.2 提供Mayavi视图窗口

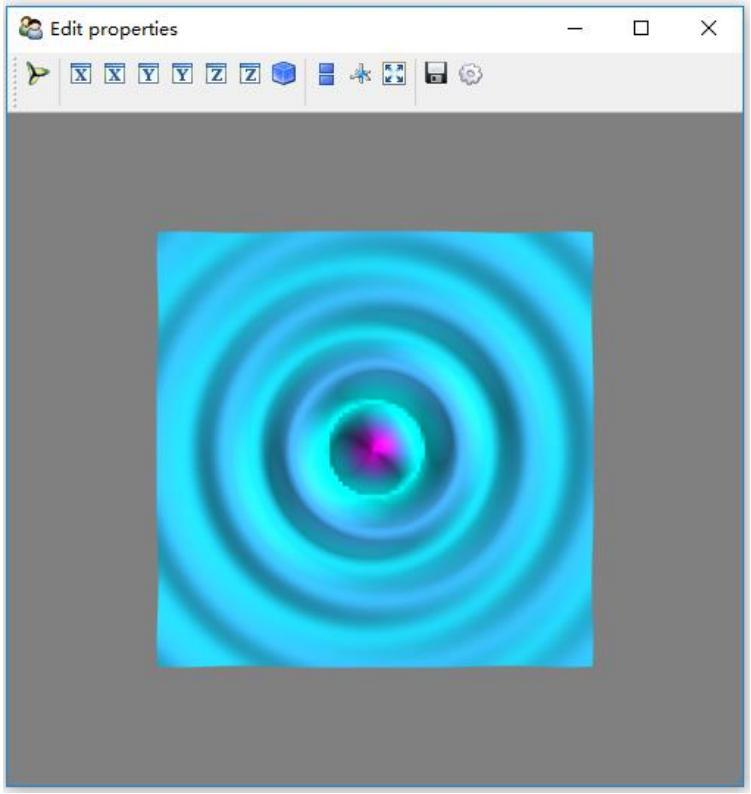
```
from traitsui.api import View, Item
from tvtk.pyface.scene_editor import SceneEditor
from mayavi.tools.mlab_scene_model import MlabSceneModel

view = View(Item(name='scene',
                 editor=SceneEditor(scene_class=MayaviScene),
                 show_label=False,
                 resizable=True,
                 height=500,
                 width=500),
            resizable=True)
```

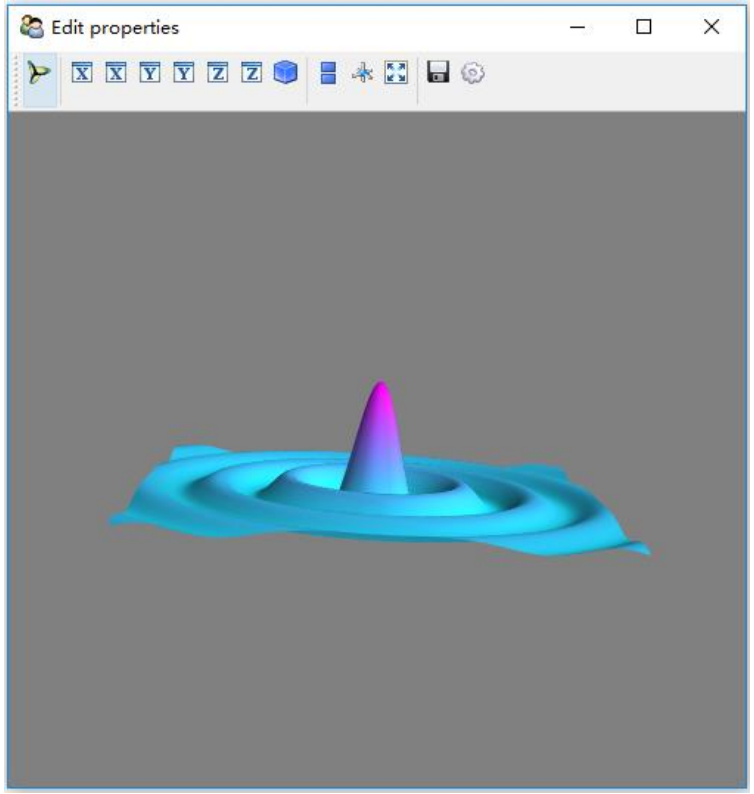
1.3初始化生成数据

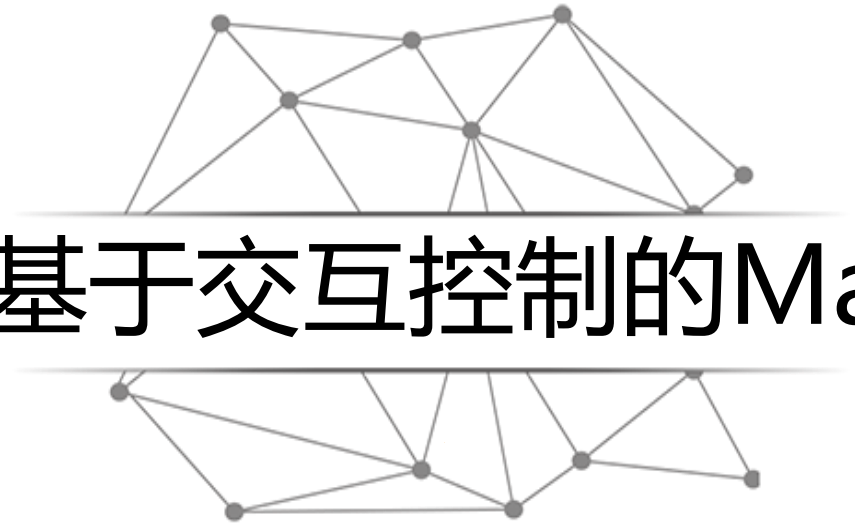
```
from numpy import sqrt, sin, mgrid

def generate_data(self):
    # 建立数据
    X, Y = mgrid[-2:2:100j, -2:2:100j]
    R = 10*sqrt(X**2 + Y**2)
    Z = sin(R)/R
    # 绘制数据
    self.scene.mlab.surf(X, Y, Z, colormap='cool')
```

```python
from numpy import sqrt, sin, mgrid
from traits.api import HasTraits, Instance
from traitsui.api import View, Item
from tvtk.pyface.scene_editor import SceneEditor
from mayavi.tools.mlab_scene_model import MlabSceneModel
from mayavi.core.ui.mayavi_scene import MayaviScene

class ActorViewer(HasTraits):
    # 场景模型
    scene = Instance(MlabSceneModel, ())
    # 建立视图
    view = View(Item(name='scene',
                     editor=SceneEditor(scene_class=MayaviScene),
                     show_label=False,
                     resizable=True,
                     height=500,
                     width=500),
                resizable=True)

    def __init__(self, **traits):
        HasTraits.__init__(self, **traits)
        self.generate_data()

    def generate_data(self):
        # 建立数据
        X, Y = mgrid[-2:2:100j, -2:2:100j]
        R = 10*sqrt(X**2 + Y**2)
        Z = sin(R)/R
        # 绘制数据
        self.scene.mlab.surf(X, Y, Z, colormap='cool')

a = ActorViewer()
a.configure_traits()
```

建立简单的mayavi窗口

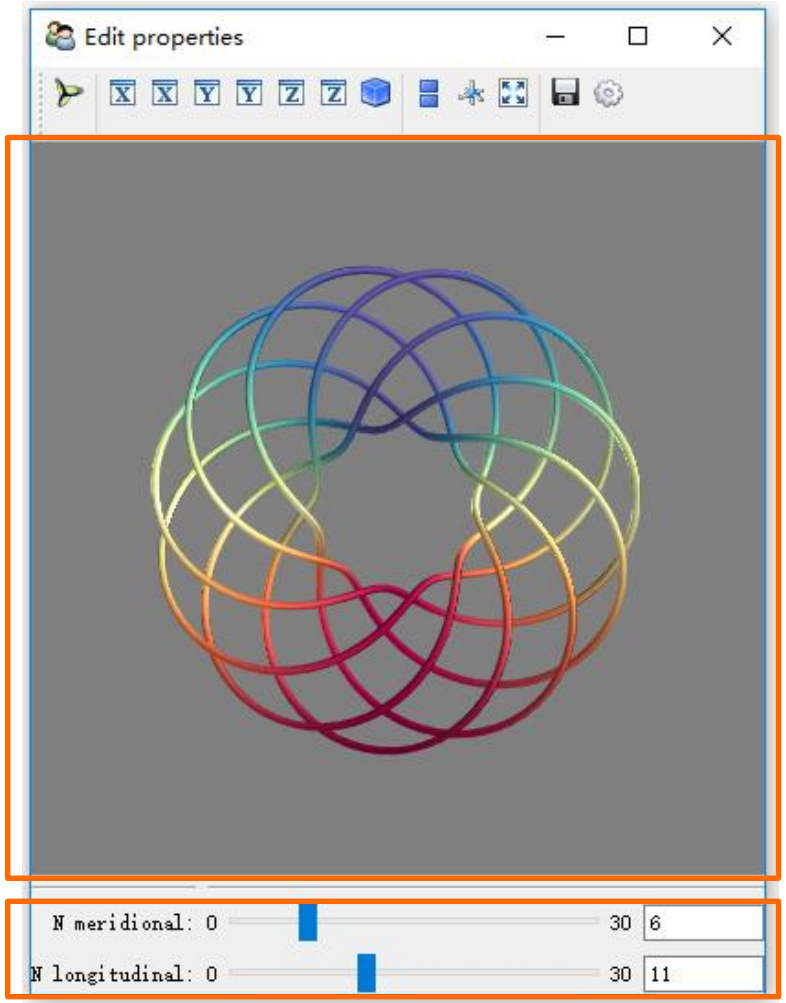






实例2：基于交互控制的Mayavi窗口

Scene变量



Item对象

Range变量

Group对象

框架步骤

1、定义从HasTraits继承的类

    1.1 定义窗口中的变量

    1.2 定义监听函数、更新视图绘制

    1.3 定义视图的布局

2、调用configure_traits()

程序框架

```
# 1.定义从HasTraits继承类
class MyModel(HasTraits):
    # 1.1定义窗口中的变量
    n_meridional
    n_longitudinal
    scene
    # 1.2更新视图绘制
    @on_trait_change()
    def update_plot(self):
        ... ...
    # 1.3建立视图布局
    view = View()

#2. 显示窗口
model = MyModel()
model.configure_traits()
```

框架步骤

1、定义从HasTraits继承的类

1.1 定义窗口中的变量

1.2 定义监听函数、更新视图绘制

1.3 定义视图的布局

2、调用configure_traits()

1.1定义窗口变量

```
from traits.api import HasTraits, Range, Instance
from mayavi.core.ui.api import MlabSceneModel

class MyModel(HasTraits):
    # 1.1定义窗口中的变量
    n_meridional    = Range(0, 30, 6) #滑动条控件
    n_longitudinal  = Range(0, 30, 11) #滑动条控件
    scene = Instance(MlabSceneModel, ())# 场景模型实例
```

框架步骤

1、定义从HasTraits继承的类

1.1 定义窗口中的变量

1.2 定义监听函数、更新视图绘制

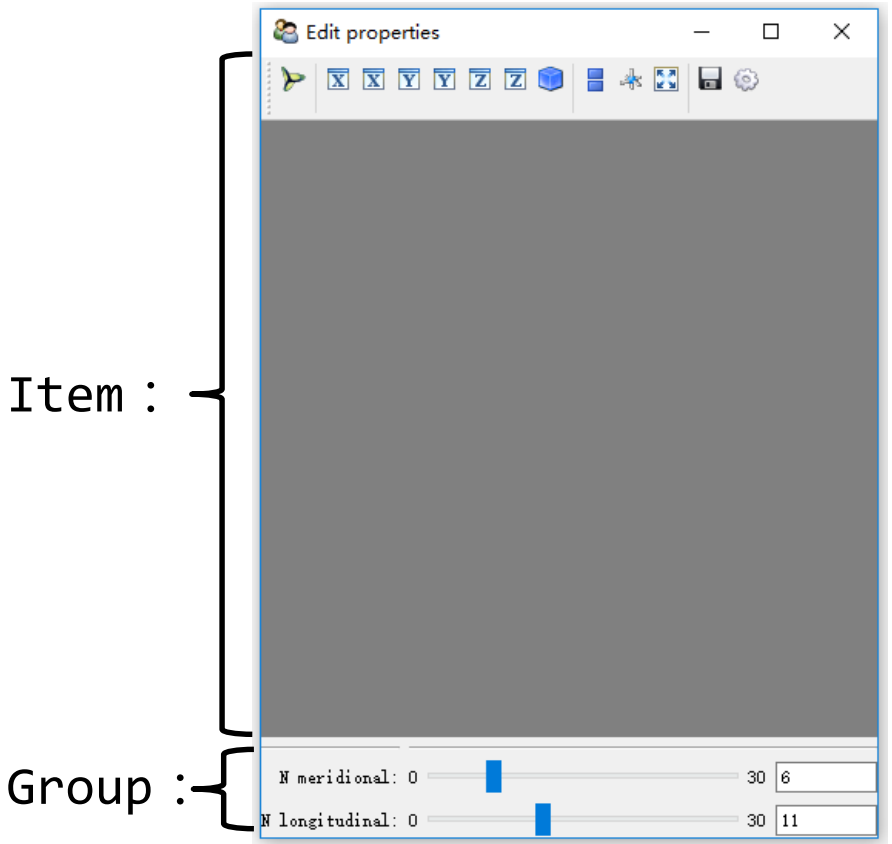
1.3 定义视图的布局

2、调用configure_traits()

1.3 建立视图布局

```
from traitsui.api import View, Item, Group
from mayavi.core.ui.api import MayaviScene, SceneEditor

# 1.3建立视图布局
view = View(
    Item('scene', editor=SceneEditor(scene_class=MayaviScene),
         height=250, width=300, show_label=False),
    Group('_', 'n_meridional', 'n_longitudinal'),
    resizable=True
    )
```

建立视图布局


Item :
Group :
Edit properties
N meridional: 0
30
6
N longitudinal: 0
30
11

框架步骤

1、定义从HasTraits继承的类

1.1 定义窗口中的变量

1.2 定义监听函数、更新视图绘制

1.3 定义视图的布局

2、调用configure_traits()

1.3 定义监听函数，更新视图绘制

```
from traits.api import on_trait_change

@on_trait_change('n_meridional,n_longitudinal,scene.activated')
def update_plot(self):
    ... ...#生成数据，并更新视图
```

1.3 定义监听函数，更新视图绘制

```
# 获得管线实例
from mayavi.core.api import PipelineBase
plot = Instance(PipelineBase)

# 当场景被激活或参数发生改变，更新图形
from traits.api import on_trait_change
@on_trait_change('n_meridional,n_longitudinal,scene.activated')
def update_plot(self):
    x, y, z, t = curve(n_meridional,n_longitudinal)
    if self.plot is None:
        self.plot = self.scene.mlab.plot3d(x, y, z, t)
    else:
        self.plot.mlab_source.set(x, y, z, t )
```

1.3 定义监听函数，更新视图绘制

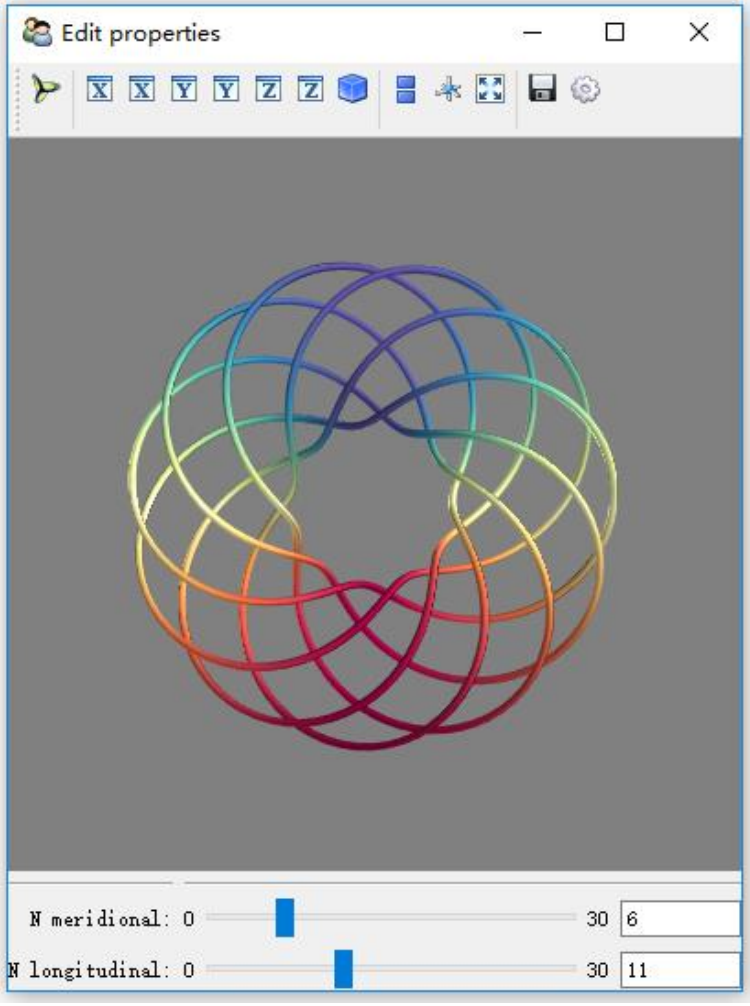
完善绘制函数：

```
def update_plot(self):
    x, y, z, t = curve(self.n_meridional, self.n_longitudinal)
    if self.plot is None:
        self.plot = self.scene.mlab.plot3d(x, y, z, t,
                    tube_radius=0.025, colormap='Spectral')
    else:
        self.plot.mlab_source.set(x=x, y=y, z=z, scalars=t)
```

定义Curve生成数据

```
from numpy import arange, pi, cos, sin
dphi = pi/300.
phi = arange(0.0, 2*pi + 0.5*dphi, dphi, 'd')
def curve(n_mer, n_long):
    mu = phi*n_mer
    x = cos(mu) * (1 + cos(n_long * mu/n_mer)*0.5)
    y = sin(mu) * (1 + cos(n_long * mu/n_mer)*0.5)
    z = 0.5 * sin(n_long*mu/n_mer)
    t = sin(mu)
    return x, y, z, t
```

```
from mayavi.core.api import PipelineBase
from mayavi.core.ui.api import MayaviScene, SceneEditor, MlabSceneModel

dphi = pi/300.
phi = arange(0.0, 2*pi + 0.5*dphi, dphi, 'd')
#建立数据
def curve(n_mer, n_long):
    mu = phi*n_mer
    x = cos(mu) * (1 + cos(n_long * mu/n_mer)*0.5)
    y = sin(mu) * (1 + cos(n_long * mu/n_mer)*0.5)
    z = 0.5 * sin(n_long*mu/n_mer)
    t = sin(mu)
    return x, y, z, t

class MyModel(HasTraits):
    n_meridional    = Range(0, 30, 6)
    n_longitudinal  = Range(0, 30, 11)
    # 场景模型实例
    scene = Instance(MlabSceneModel, ())
    # 管线实例
    plot = Instance(PipelineBase)
    #当场景被激活，或者参数发生改变，更新图形
    @on_trait_change('n_meridional,n_longitudinal,scene.activated')
    def update_plot(self):
        x, y, z, t = curve(self.n_meridional, self.n_longitudinal)
        if self.plot is None:#如果plot未绘制则生成plot3d
            self.plot = self.scene.mlab.plot3d(x, y, z, t,
                        tube_radius=0.025, colormap='Spectral')
        else:#如果数据有变化，将数据更新即重新赋值
            self.plot.mlab_source.set(x=x, y=y, z=z, scalars=t)

    # 建立视图布局
    view = View(Item('scene', editor=SceneEditor(scene_class=MayaviScene),
                     height=250, width=300, show_label=False),
                Group('_', 'n_meridional', 'n_longitudinal'),
                resizable=True)

model = MyModel()
model.configure_traits()
```


Edit properties
N meridional: 0
30
6
N longitudinal: 0
30
11