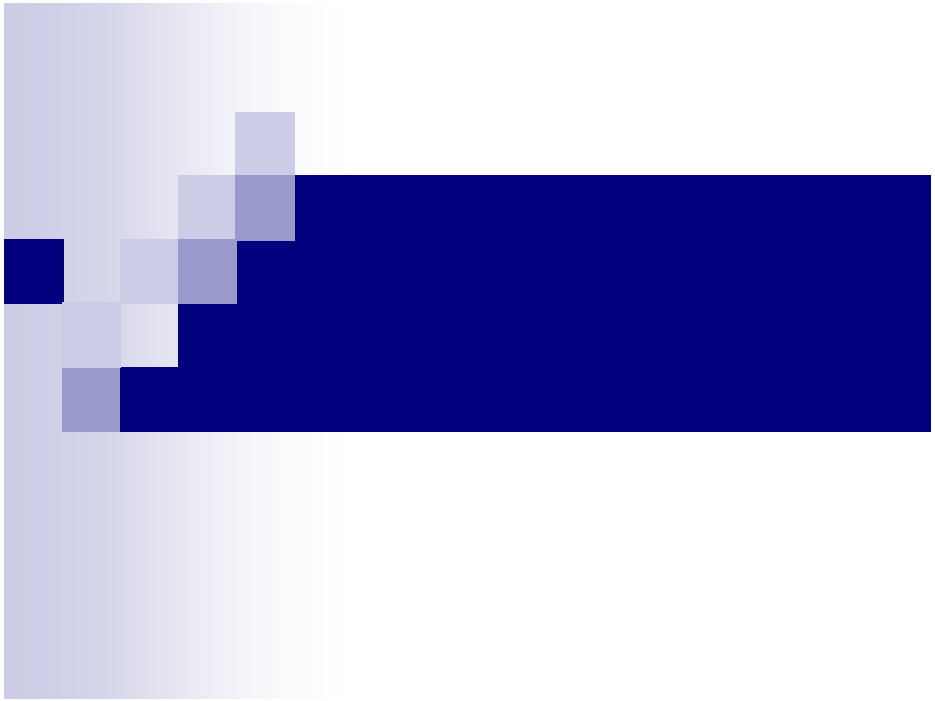
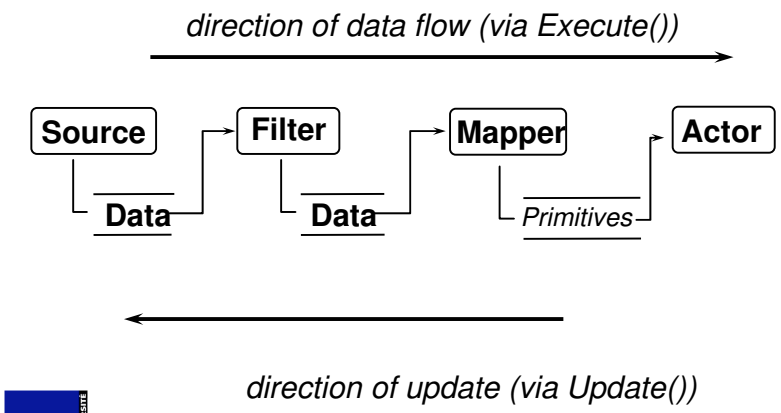
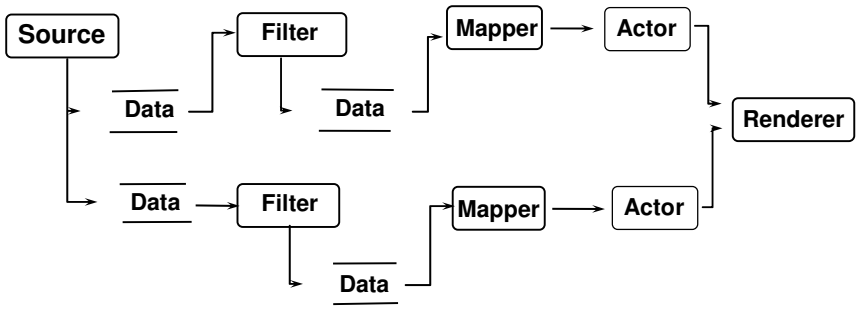


# Pipeline Execution Model



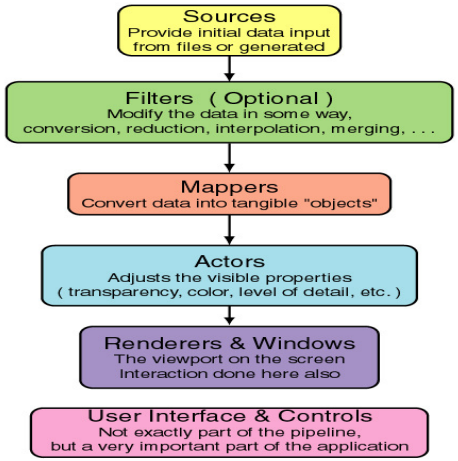

# VTK Visualization Pipeline



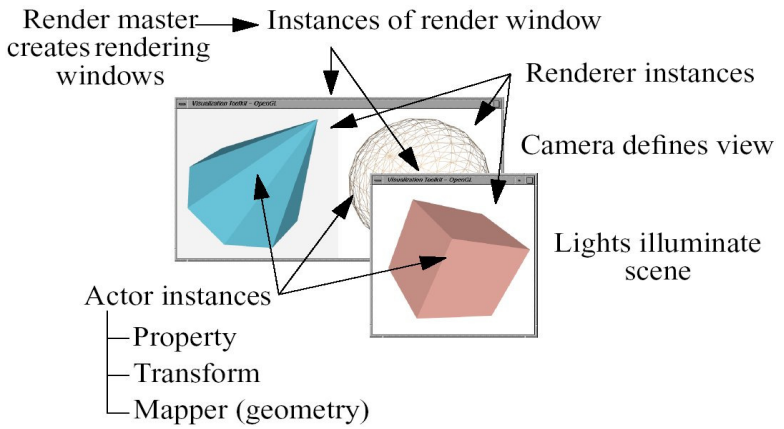

# Main Graphics Objects in VTK

- Conversion of a data structure into graphical object in VTK performed by a **mapper**
- Graphics objects in VTK are **actors**
  - Controls graphics properties such as colour and shading
  - Position, rotation and surface properties also specified by actor methods
  - Transformation from object to world co-ordinates
- Actors rendered in the scene by the **renderer**
  - Controls camera and lighting properties
- Renderer draws to a **render window**
  - Controls window size
  - Can display or capture to an image file

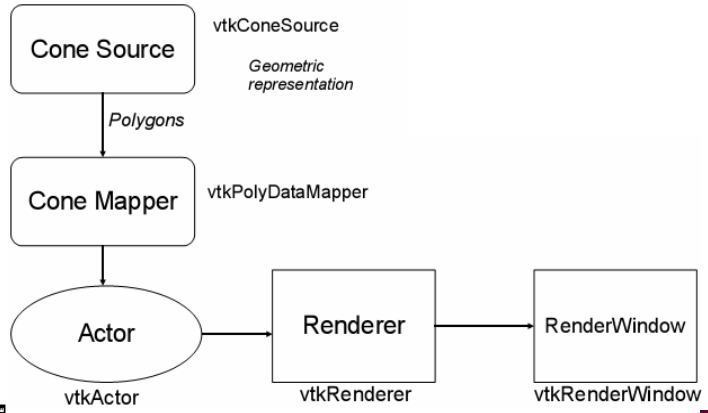
# VTK Pipeline : summary



# VTK Graphics Model




# Drawing a cone in VTK




# Drawing a cone : Tcl

```
package require vtk

# create a cone geometry source object
vtkConeSource cone
cone SetResolution 8

# create mapper object and map cone
    geometry
vtkPolyDataMapper coneMapper
coneMapper SetInput [cone GetOutput]

# create an actor object and set mapper
vtkActor coneActor
coneActor SetMapper coneMapper
```

```
# create a renderer
vtkRenderer ren1
ren1 SetBackground 0.1 0.2 0.4

# assign actor to the renderer
ren1 AddActor coneActor

# create a rendering window
vtkRenderWindow renWin

# assign renderier to window
renWin AddRenderer ren1

# render scene
renWin Render

# no interaction window
wm withdraw .
```

# Interactor

- Button 1 – rotate
- Button 2 – translate (<Shift> Button 1 on PC)
- Button 3 – zoom
- Keypress e or q – exit
- Keypress f – “fly-to” point under mouse
- Keypress s/w – surface/wireframe
- Keypress p – pick
- Keypress r – reset camera
- Keypress 3 – toggle stereo

***Switch styles:***
***Keypress j – joystick; t - trackball style***

# Interacting with a cone

```
package require vtk

# create a cone geometry source object
vtkConeSource cone
cone SetResolution 8

# create mapper object and map cone geometry
vtkPolyDataMapper coneMapper
coneMapper SetInput [cone GetOutput]

# create an actor object and set mapper
vtkActor coneActor
coneActor SetMapper coneMapper

# create a renderer
vtkRenderer ren1
ren1 SetBackground 0.1 0.2 0.4
# assign actor to the renderer
ren1 AddActor coneActor
# create a rendering window
vtkRenderWindow renWin
# assign renderier to window
renWin AddRenderer ren1
# render scene
renWin Render

# attach Interactor to Window
vtkRenderWindowInteractor iren
iren SetRenderWindow renWin
iren Initialize

# no interaction window
wm withdraw .
```

# Accessing Actors Properties

- The appearance of each prop (e.g. vtkActor) is controlled by its property member (of type vtkProperty)
- To modify the appearance of the actor we use methods associated with the property member e.g.

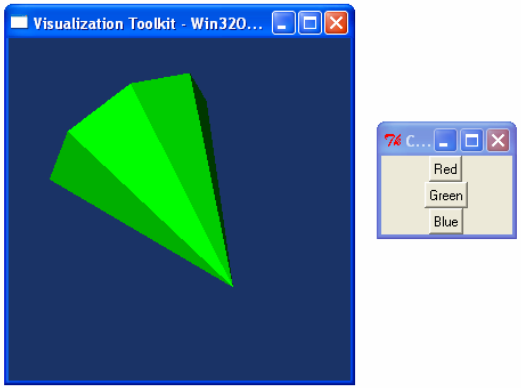
**Step 1 – get a reference to the property member**

```
set property [ coneActor GetProperty ]
```

**Step 2 – do something e.g. change the color etc.**

```
$property SetColor  1 0 0
$property SetAmbient 1.0      (Default for surface 0.0)
$property SetDiffuse 0.0      (Default for surface 1.0)
$property SetSpecular 0.0
$property SetRepresentationToWireframe
```

# Exercise: change cone color




# LegendSource

```
vtkLegendBoxActor lBar
lBar SetNumberOfEntries 1
lBar SetEntryString 0 CONE
lBar SetWidth 0.8
lBar SetHeight 0.06
lBar SetPosition 0.1 0.92
lBar BorderOff

vtkRenderer ren1
ren1 AddActor lBar
```

# PolyDataSource

- vtkConeSource is a vtkPolyDataSource. Other include:
  - vtkPolySphereSource
  - vtkPolyDiskSource
  - vtkPolyCylinderSource
  - vtkArrowSource

. . . . . .