Realidade Virtual e Interfaces Modernas

Prof. Carlos Henrique Q. Forster
IEC-ITA
Julho/2005

Exemplos do Java3D

Tópicos

- Usando Jython
- Construindo um grafo de cena básico
- Inserindo elementos
  - Forma, aparência, fonte de luz, transformação

Jython

- Interpretador Python para máquina virtual Java
- Utiliza as classes Java em Python
- Modo interativo
- Compilador
- Ver documentação do Python

Jython

- Importando pacotes do Java

```
import java.awt
from java import awt, applet
from java.awt import *
```

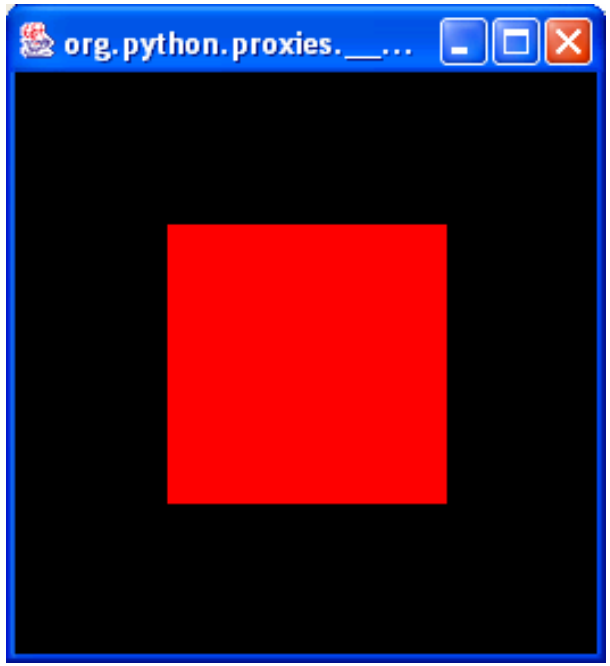
- Definindo funções à moda Python

```
def hello_world():
    return "Hello, world!"
```

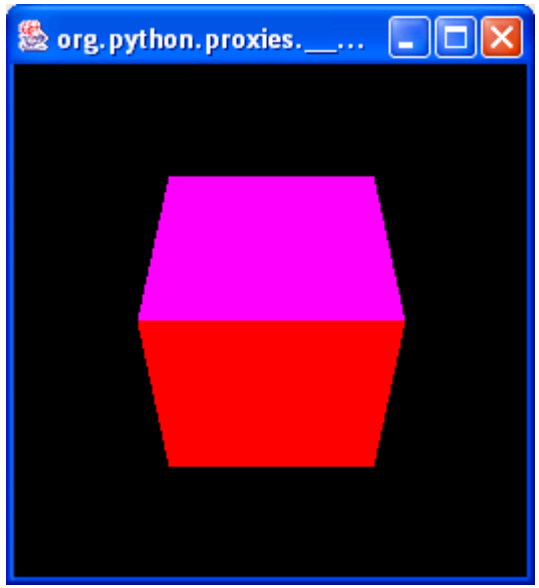
Jython

- Sair do jython
  - raise SystemExit
- Usando classes Java
  - Import java.lang.String
  - Dir(java.lang.String)
  - S=java.lang.String("Hello World")
  - Print S.toLowerCase()
  - Type(S)

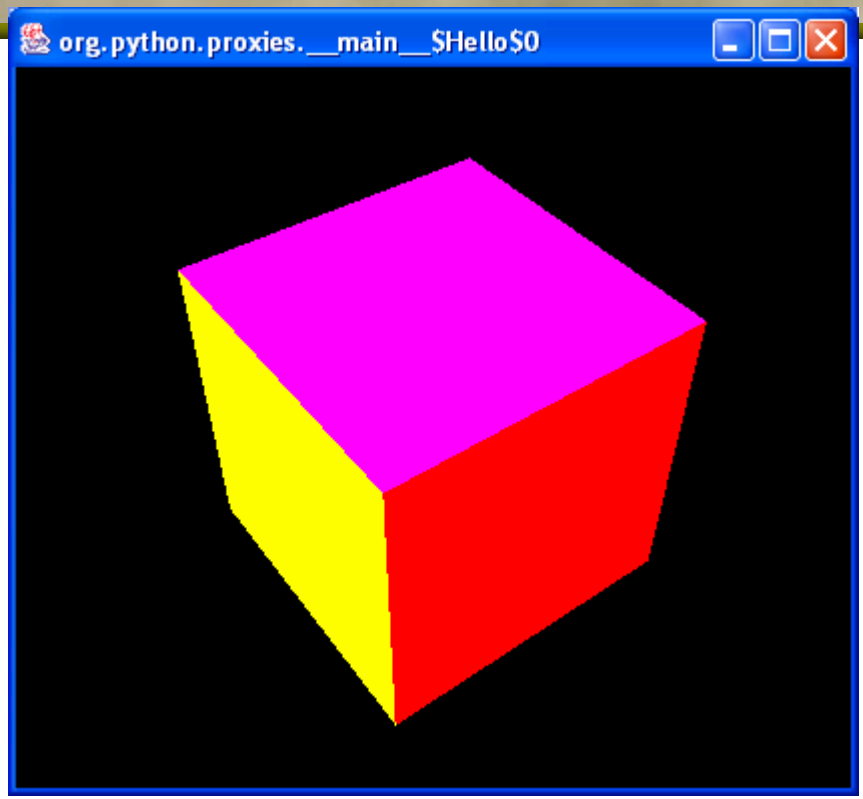
```
from java.applet import Applet.
from java.awt import *.
from com.sun.j3d.utils.universe import SimpleUniverse.
from com.sun.j3d.utils.geometry import ColorCube.
from com.sun.j3d.utils.applet import MainFrame.
from javax.media.j3d import *.
.
class Hello(Applet):.
        def __init__(self):.
                Applet.__init__(self).
                self.setLayout(BorderLayout()).
                self.config=SimpleUniverse.getPreferredConfiguration().
                self.canvas3d=Canvas3D(self.config).
                self.add("Center", self.canvas3d).
                self.scene=create_scene_graph().
                self.scene.compile().
                self.universe=SimpleUniverse(self.canvas3d).
                self.universe.getViewingPlatform().setNominalViewingTransform()
                self.universe.addBranchGraph(self.scene).
                .
def create_scene_graph():.
        objroot=BranchGroup().
        objroot.addChild(ColorCube(0.4)).
        return objroot.
.
mf=MainFrame(Hello(),256,256).
```


org.python.proxies.__...

```
from java.applet import Applet.
from java.awt import *.
from com.sun.j3d.utils.universe import SimpleUniverse.
from com.sun.j3d.utils.geometry import ColorCube.
from com.sun.j3d.utils.applet import MainFrame.
from javax.media.j3d import *.
from java.lang import Math.
.
class Hello(Applet):.
        def __init__(self):.
                Applet.__init__(self).
                self.setLayout(BorderLayout()).
                self.config=SimpleUniverse.getPreferredConfiguration().
                self.canvas3d=Canvas3D(self.config).
                self.add("Center", self.canvas3d).
                self.scene=create_scene_graph().
                self.scene.compile().
                self.universe=SimpleUniverse(self.canvas3d).
                self.universe.getViewingPlatform().setNominalViewingTransform().
                self.universe.addBranchGraph(self.scene).
                .
def create_scene_graph():.
        objroot=BranchGroup().
        rotate=Transform3D().
        rotate.rotX(Math.PI/4.0).
        objrotate=TransformGroup(rotate).
        objrotate.addChild(ColorCube(0.4)).
        objroot.addChild(objrotate).
        return objroot.
.
mf=MainFrame(Hello(),256,256).
```

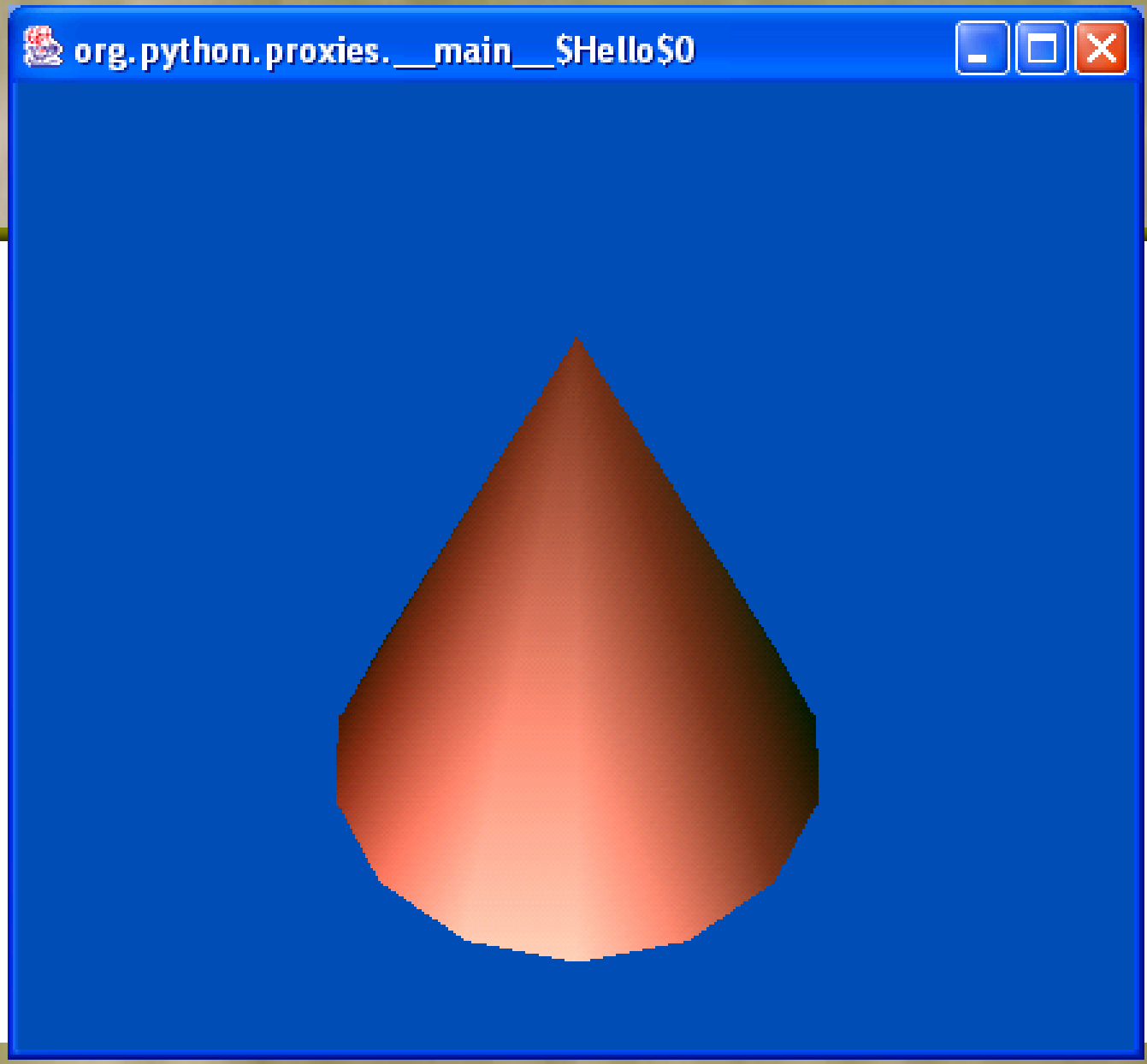

org.python.proxies.__...

```
.
def create_scene_graph():.
        objroot=BranchGroup().
        rotate=Transform3D().
        temprotate=Transform3D().
        rotate.rotX(Math.PI/4.0).
        temprotate.rotY(Math.PI/5.0).
        rotate.mul(temprotate).
        objrotate=TransformGroup(rotate).
        objrotate.addChild(ColorCube(0.4)).
        objroot.addChild(objrotate).
        return objroot.
.
mf=MainFrame(Hello(),256,256).
```


org.python.proxies.__main__$Hello$0

```
def create_scene_graph():.
        objroot=BranchGroup().
        .
        transf=Transform3D().
        temp=Transform3D().
        transf.rotX(Math.PI/4.0).
        temp.rotY(Math.PI/5.0).
        transf.mul(temp).
        objrotate=TransformGroup(transf).
        .
        defbounds=BoundingSphere(Point3d(0,0,0),80).
        .
        ap=Appearance().
        ap.setMaterial(Material( Color3f(0, 0.1, 0), Color3f(0, 0, 0),.
                                 Color3f(0.7, 0, 0 ),  Color3f(1, 1, 1), 4 )).
        .
        light=DirectionalLight(1 , Color3f(1, 1, 1), Vector3f(0.3, 0.2, -1)).
        light.setInfluencingBounds(defbounds).
```

```
        amblight=AmbientLight(1, Color3f(1,1,1)).
        amblight.setInfluencingBounds(defbounds).
        .
        #primitive=Box(0.4, 0.3, 0.2, ap).
        primitive=Cone(0.5,1.0,ap).
        .
        backg=Background(Color3f(0, 0.3, 0.7)).
        backg.setApplicationBounds(defbounds).
        .
        objrotate.addChild(primitive).
        objroot.addChild(objrotate).
        objroot.addChild(light).
        objroot.addChild(amblight).
        objroot.addChild(backg).
        return objroot.
.
mf=MainFrame(Hello(),256,256).
```


org.python.proxies.__main__$Hello$0