Fundamentos de Processamento Gráfico

Aula 7

Realidade Virtual

Criação de Mundos Virtuais 3D Interativos com Java3D

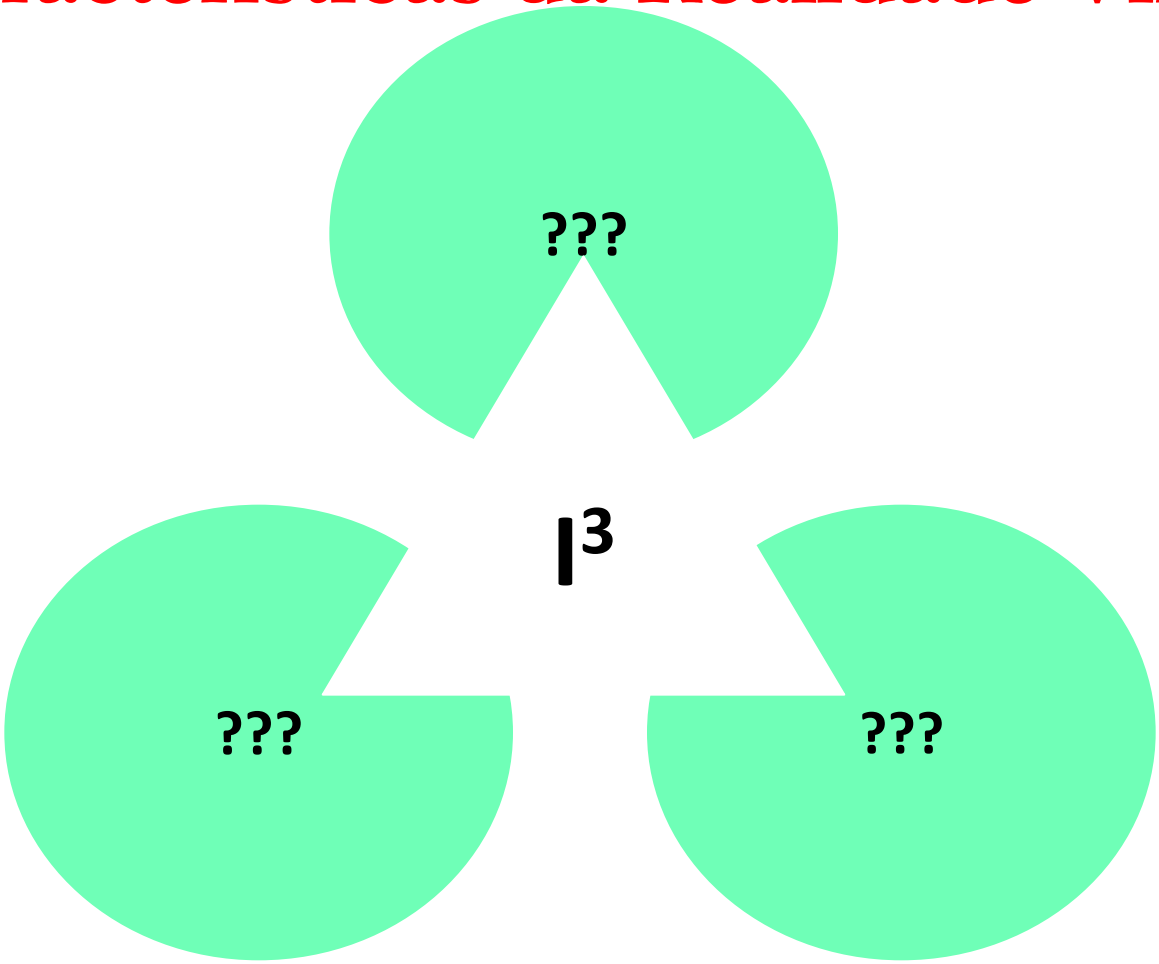
Profa. Fátima Nunes

Material baseado em:

NUNES, F. L. S. ; CORRÊA, C. G. . Interação com Java3D. In: José Remo Ferreira Brega; Judith Kelner. (Org.). Interação com Realidade Virtual e Realidade Aumentada. 1ed.Bauru (SP): Canal 6, 2010, v. 1, p. 105-118

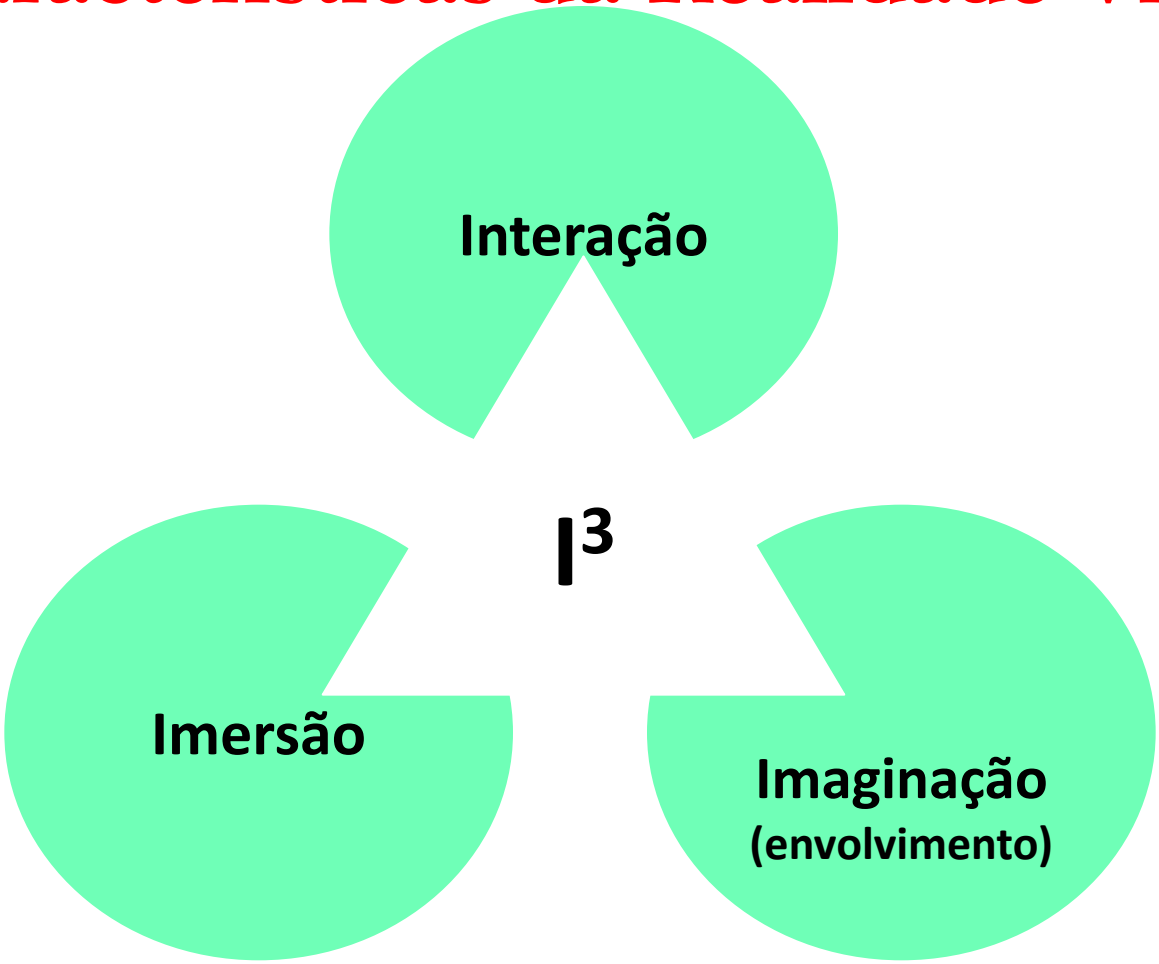
Relembrando

Características da Realidade Virtual



Relembrando

Características da Realidade Virtual



Introdução

- O que é a API Java 3D?

O que é a API Java 3D?

- **Objetivo**: construção de Ambientes Virtuais
- **Funcionalidades e características**: som 3D, animação e interações, renderização paralela e otimizada.
- **Produtividade** (reuso de código) e programação de alto nível (grafo de cena)

Recursos de Instalação

- Download gratuito

  http://java.sun.com/javase/technologies/desktop/java3d/

- Requisito: JDK (*Java Development Kit*)
- Suportes:

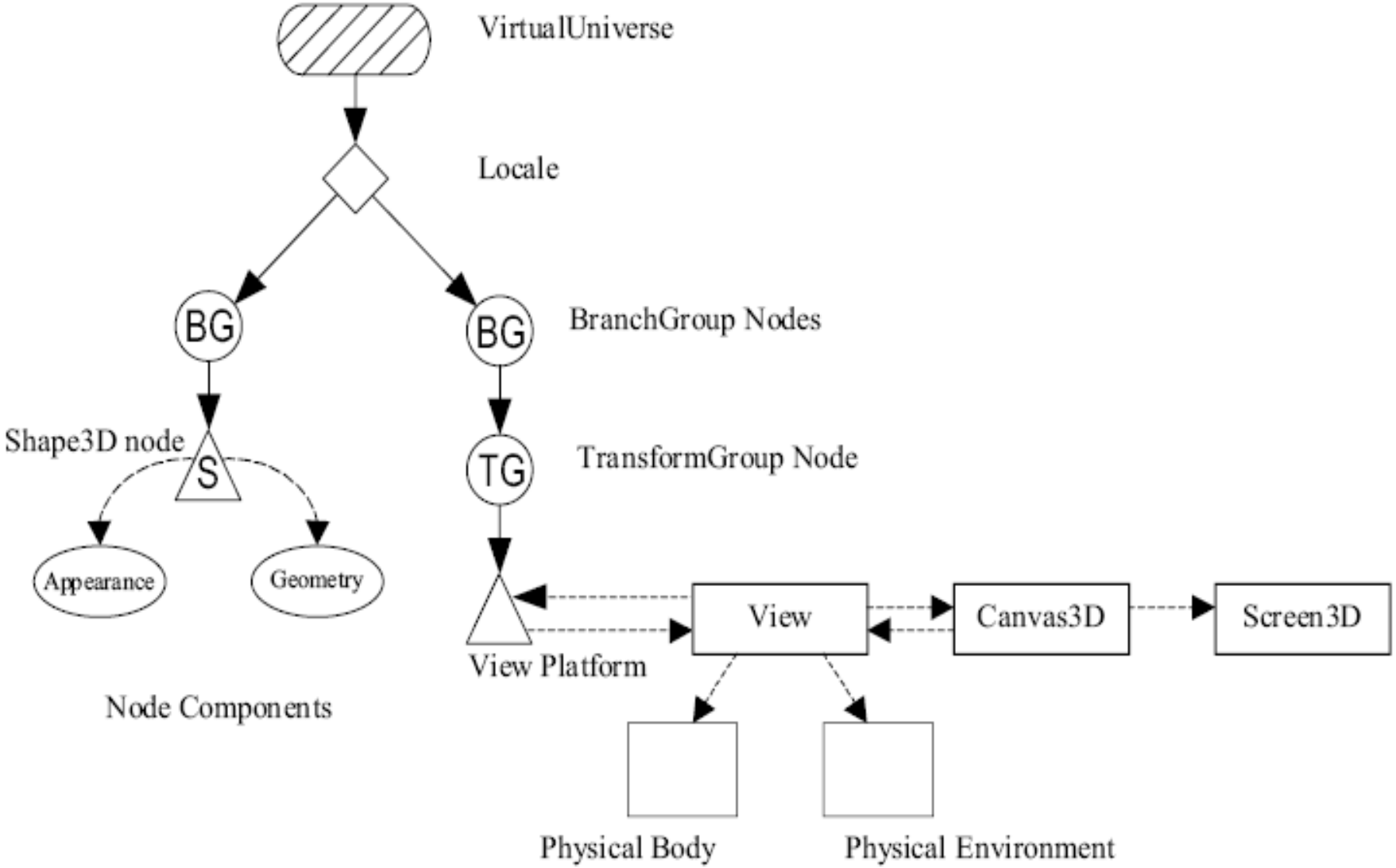
  - OpenGL

  - DirectX

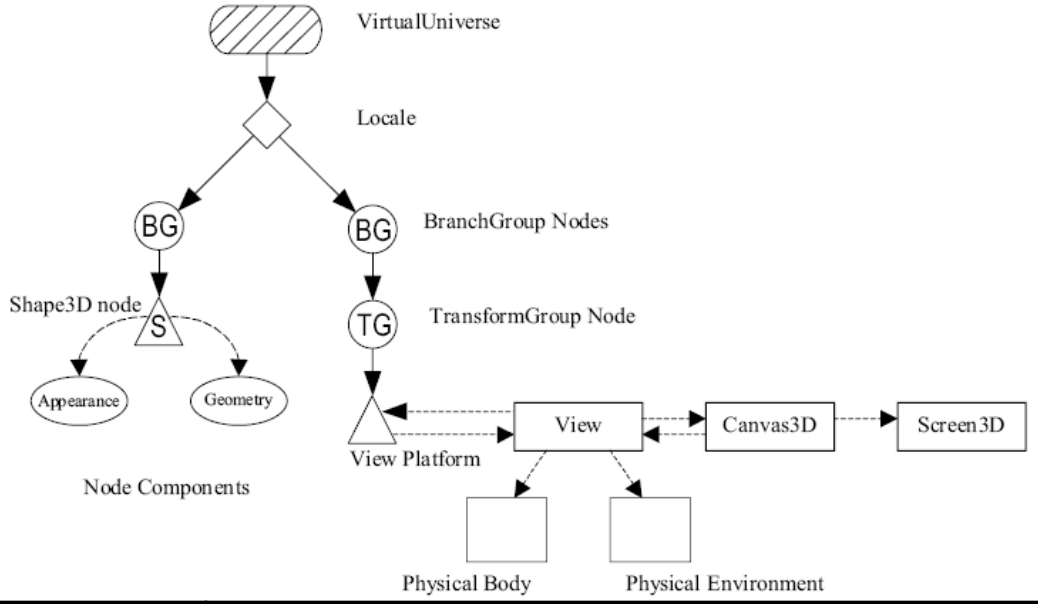
Estrutura da API Java 3D

- **Classes e pacotes:** mesmos conceitos da Linguagem de programação Java
- **Grafo de cena:** determina as relações entre os objetos da cena e as propriedades de tais objetos (textura, cor e geometria, por exemplo)
- **Objetos:** instâncias das classes ou nós na representação do grafo de cena

Exemplo de Grafo de Cena



Exemplo de Grafo de Cena

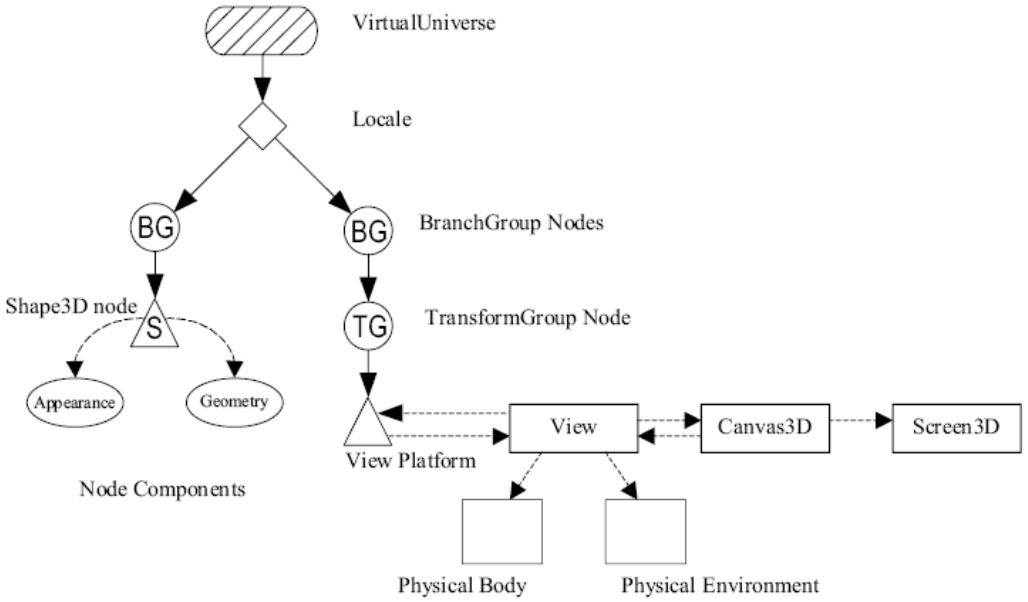


Símbolo	Nome	Finalidade
	VirtualUniverse	Representar o universo virtual
	Locale	Definir uma área com tamanho e localização no universo virtual
	Group	Organizar os dados, controlar a ordem de renderização e alterar a posição, orientação e tamanho dos objetos visíveis no Ambiente Virtual
	Leaf	Representar os objetos que compõem a cena gráfica
	NodeComponent	Especificar as propriedades de um determinado objeto
	Outros objetos	Especificar as propriedades de um determinado objeto
	Relacionamento entre pai e filho	
	Referência	

Relacionamentos no Grafo de Cena

- Criação do nó do tipo *Locale*:

```
Locale myLocal = new Locale (this);
```



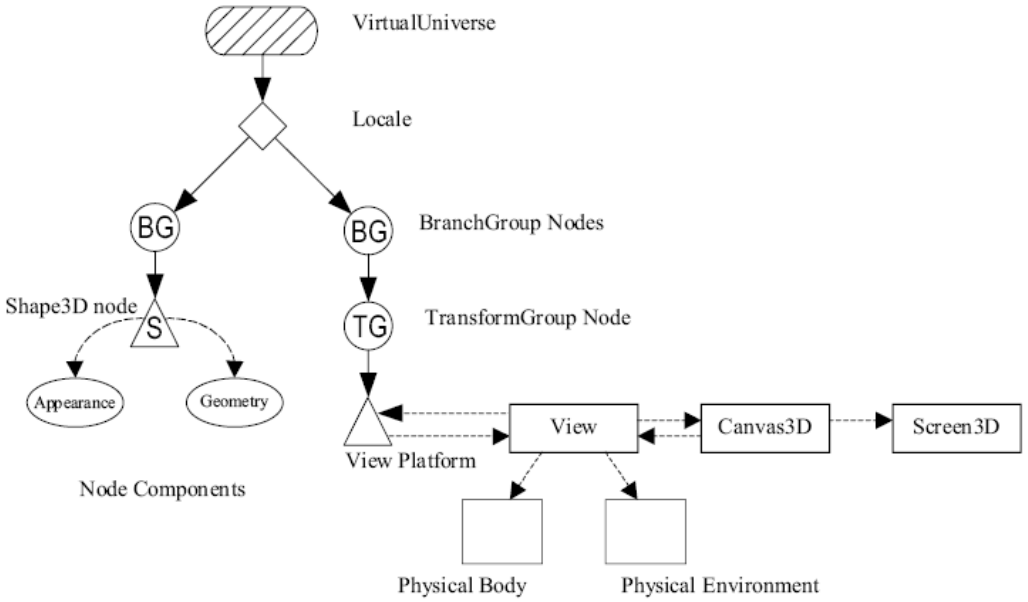
Relacionamentos no Grafo de Cena

- Criação do nó do tipo *Locale*:

```
Locale myLocal = new Locale (this);
```

- Criação do nó do tipo *BranchGroup*:

```
BranchGroup myBG = new BranchGroup;
```



Relacionamentos no Grafo de Cena

- Criação do nó do tipo *Locale*:

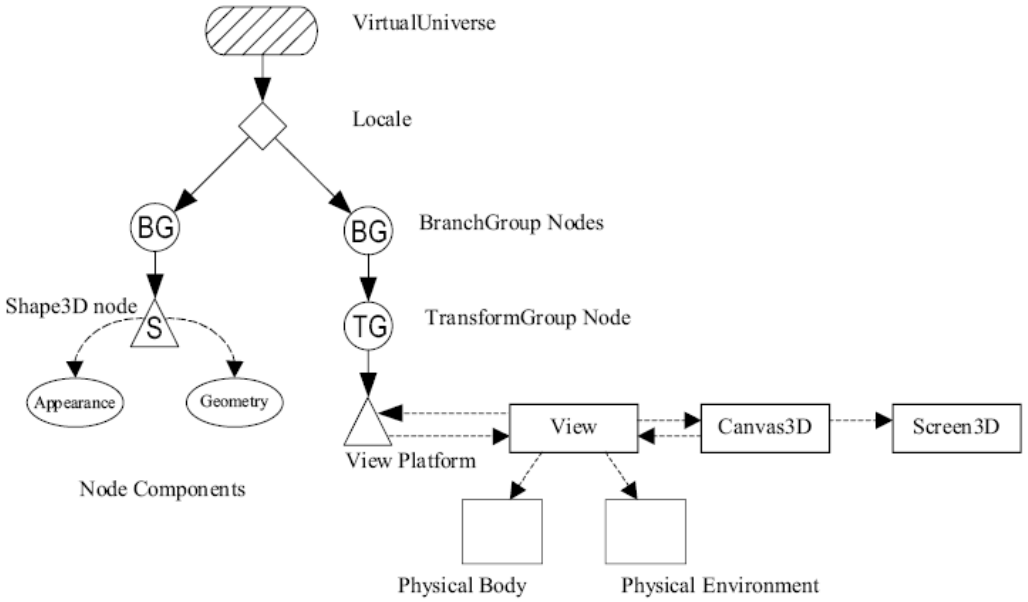
```
Locale myLocal = new Locale (this);
```

- Criação do nó do tipo *BranchGroup*:

```
BranchGroup myBG = new BranchGroup;
```

- Ligação do nó *BranchGroup* ao nó *Locale*:

```
myLocal.addBranchGraph(myBG);
```



Dispositivos na Interação

- Convencionais
  - Mouse
  - Teclado
- Não convencionais
  - Luva de dados
  - Dispositivo háptico

Luva de Dados

- Material sintético e sensores ópticos ou mecânicos
- Identificar movimentos da mão:

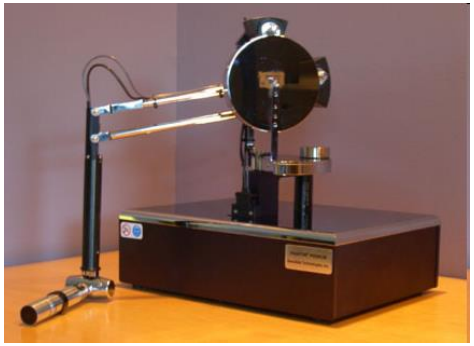
  - ângulo de cada dedo (falanges)

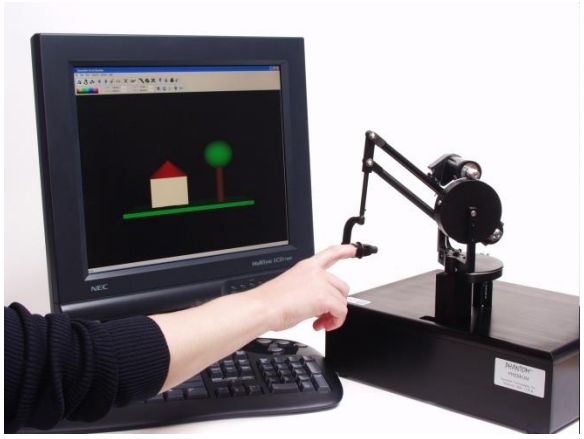
  - posição e orientação do pulso

Dispositivo Háptico

- Conceitos:
  - *feedback* tátil: ação aplicada à pele que indica alguma sensação;
  - *feedback* de força: retorno de sensação de peso ou resistência de algo;
  - *feedback* cinestésico: percepção de movimentos por estruturas existentes em músculos, tendões, juntas;
  - *feedback* proprioceptivo: movimentos definidos por informações oferecidas de acordo com a postura do corpo;

Dispositivos Hápticos





Implementando a Interação

- Classe *Behavior*
- WakeupOn_____
- TransformGroup
- Transform3D:

  - set(Matrix)

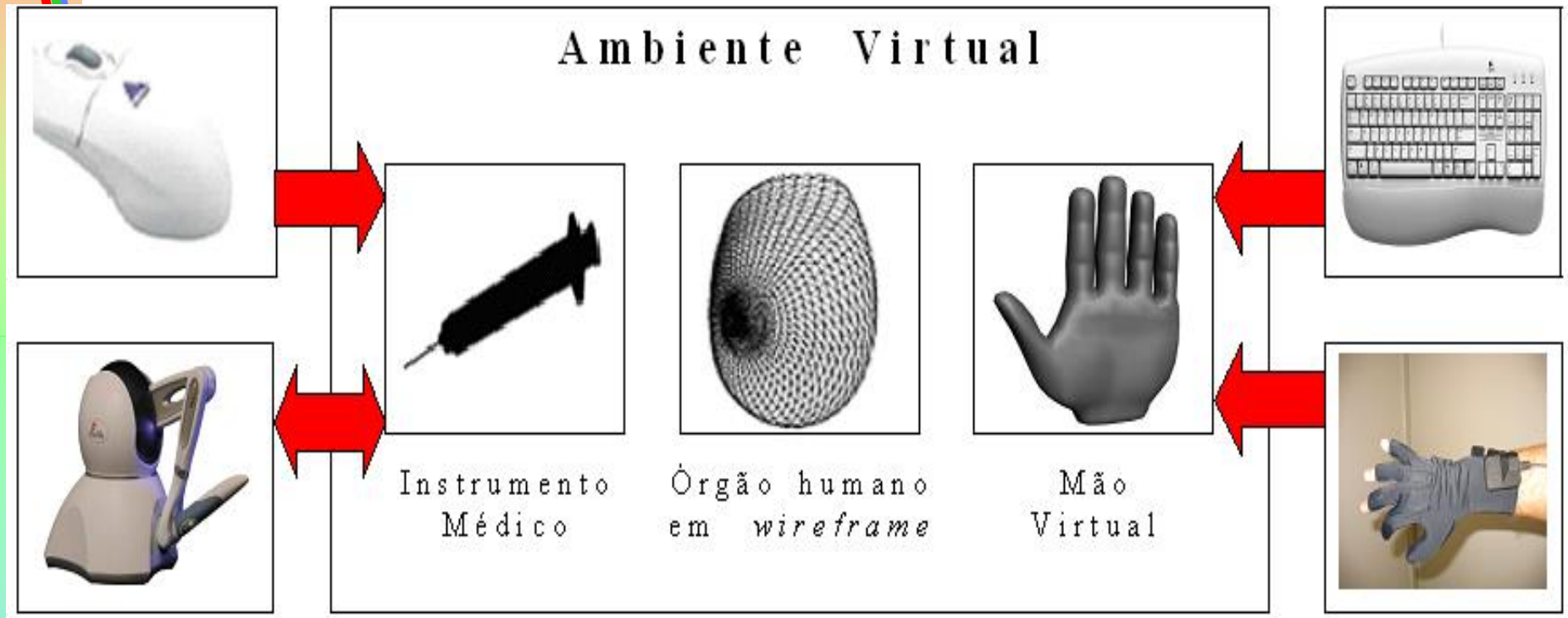
  - setRotation(x, y, z, angle)

  - setTranslation(x, y, z)

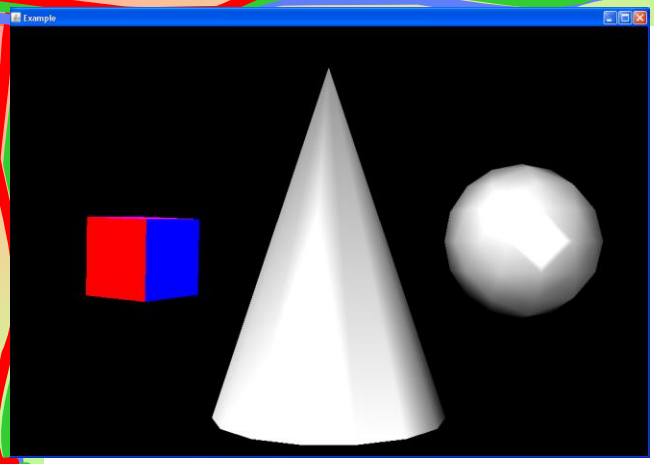
Exemplo: ViMeT

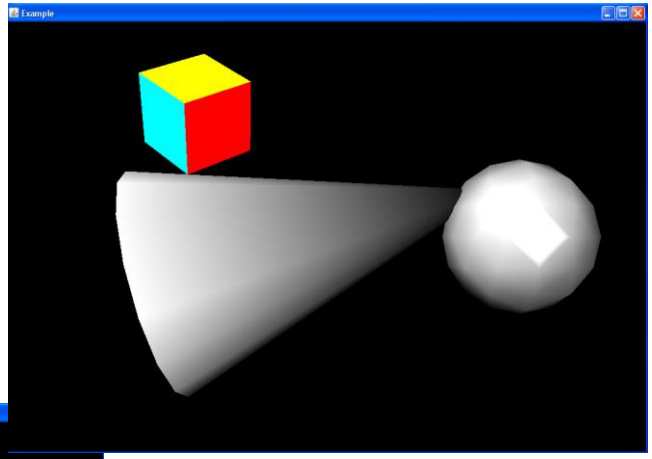
- ***ViMeT = Virtual Medical Training***
  - classes Java para desenvolvimento de aplicações com funcionalidades e características no domínio de procedimentos médicos (biópsia);
  - produtividade na implementação de aplicações;
  - treinamento de qualidade com benefícios;
  - interação com dispositivos convencionais e não convencionais.

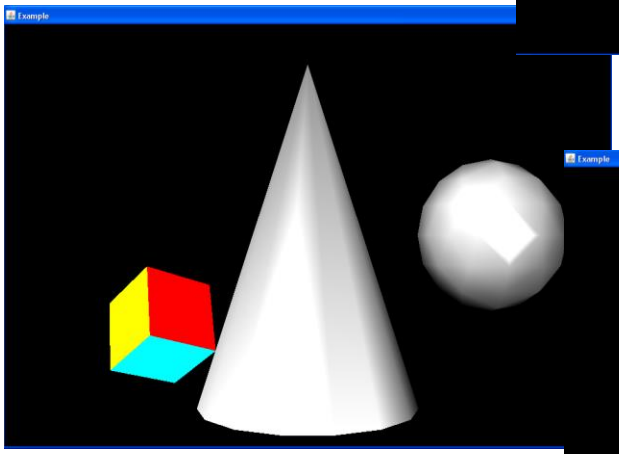
Módulo de Interação do ViMeT

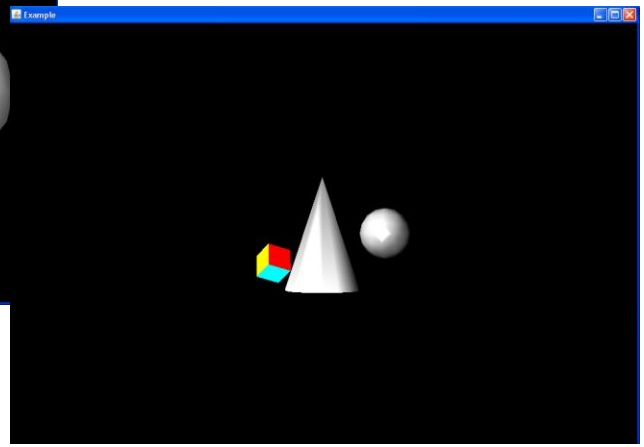


Exemplo Simplificado





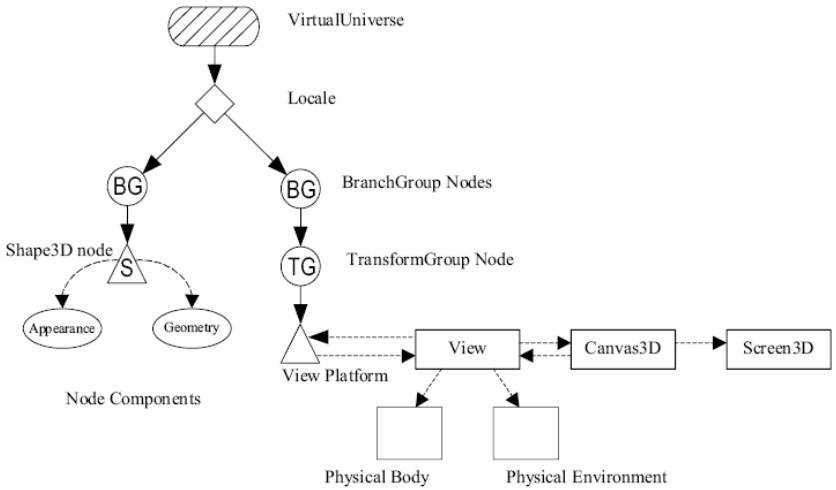




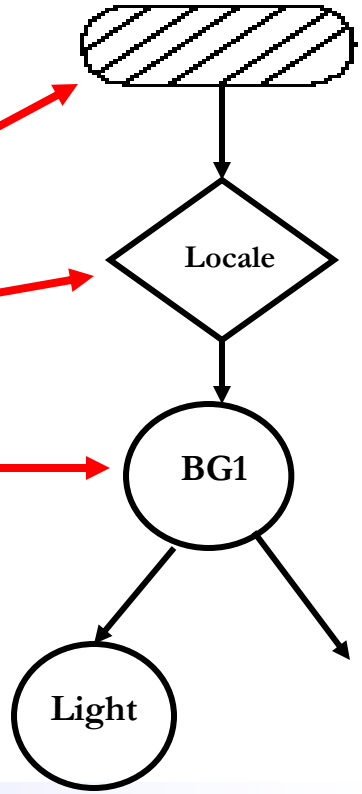
Exemplo Simplificado

```
0019
0020    public Example(Canvas3D canvas)
0021    {
0022
0023        VirtualUniverse universe = new VirtualUniverse();
0024
0025        Locale locale = new Locale(universe);
0026
0027        BranchGroup branchgroup = new BranchGroup();
0028
0029        branchgroup.addChild(createLight(0.0f, 0.0f, -1.0f));
0030        branchgroup.addChild(createLight(0.3f, -0.3f, -0.3f));
0031
0032        TransformGroup transformgroupCone = new TransformGroup();
0033        transformgroupCone.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
0034
0035        Cone cone = new Cone(0.3f, 0.9f, createAppearance());
0036
0037        transformgroupCone.addChild(cone);
0038
0039        branchgroup.addChild(transformgroupCone);
0040
0041        TransformGroup transformgroupCube = new TransformGroup();
0042        transformgroupCube.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
0043
0044        ColorCube colorcube = new ColorCube(0.1f);
0045        transformgroupCube.addChild(colorcube);
0046        transformgroupCone.addChild(transformgroupCube);
0047
0048        TransformGroup transformgroupSphere = new TransformGroup();
0049        transformgroupSphere.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
0050
0051        Transform3D transform3dSphere = new Transform3D();
0052        transform3dSphere.setTranslation(new Vector3f(0.5f, 0.0f, 0.0f));
0053        transformgroupSphere.setTransform(transform3dSphere);
0054
0055        Sphere sphere = new Sphere(0.2f);
0056        transformgroupSphere.addChild(sphere);
0057        branchgroup.addChild(transformgroupSphere);
0058
0059        locale.addBranchGraph(branchgroup);
0060
0061
0062
0063
0064        Observer observer = new Observer(canvas, locale);
0065
0066        Mouse mouse = new Mouse(transformgroupCube, locale);
0067
0068        keyboard = new Keyboard(transformgroupCone, locale);
0069
0070        Glove glove = new Glove(transformgroupSphere, locale);
0071    }
```

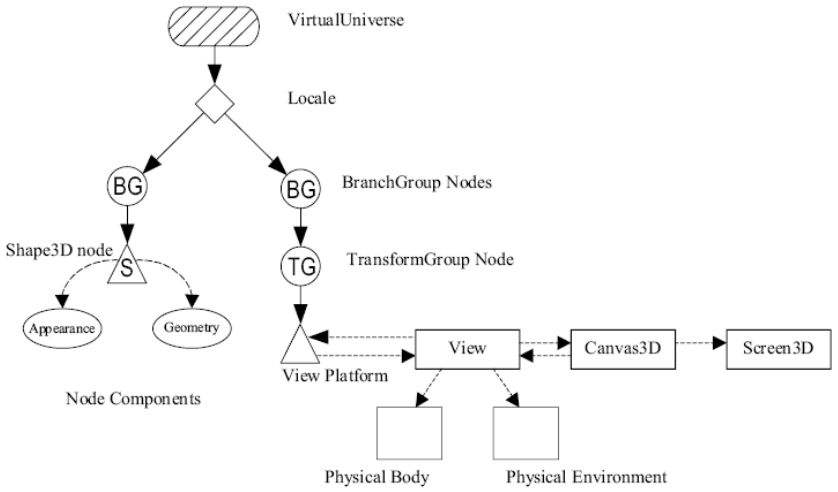
Exemplo Simplificado

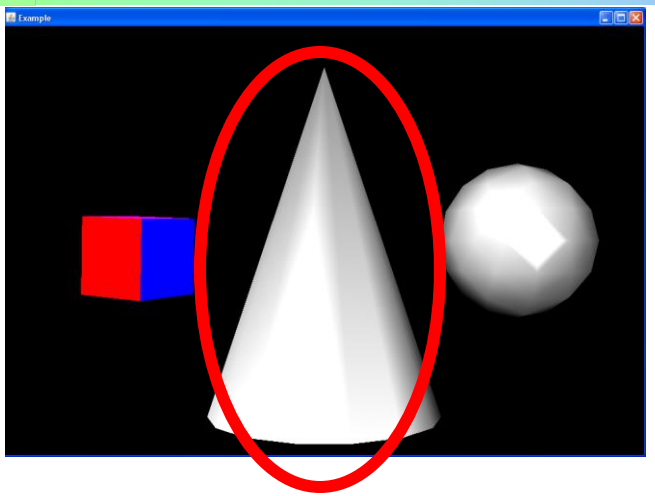



```
0019
0020    public Example(Canvas3D canvas)
0021    {
0022
0023        VirtualUniverse universe = new VirtualUniverse();
0024
0025        Locale locale = new Locale(universe);
0026
0027        BranchGroup branchgroup = new BranchGroup();
0028
0029        branchgroup.addChild(createLight(0.0f, 0.0f, -1.0f));
0030        branchgroup.addChild(createLight(0.3f, -0.3f, -0.3f));
```



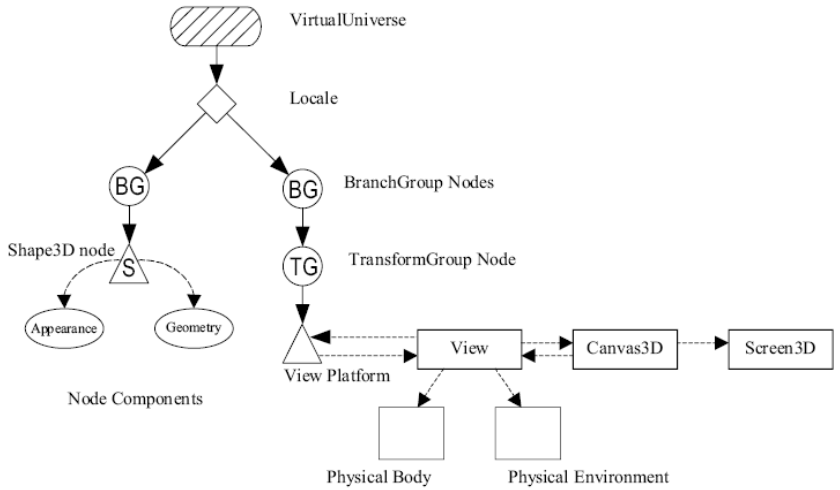
Exemplo Simplificado

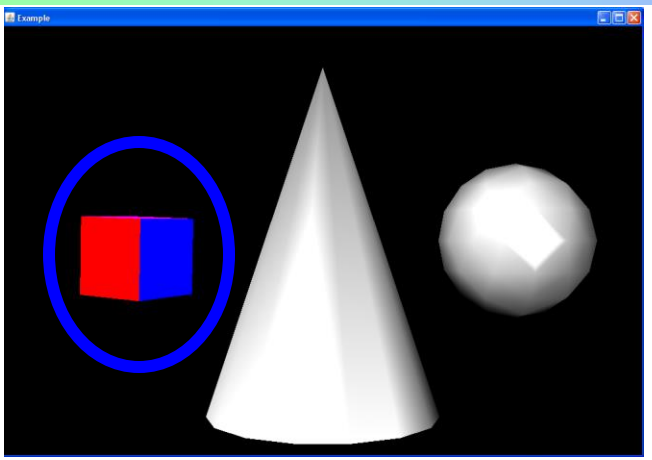





```
0032        TransformGroup transformgroupCone = new TransformGroup();
0033        transformgroupCone.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
0034
0035        Cone cone = new Cone(0.3f, 0.9f, createAppearance());
0036
0037        transformgroupCone.addChild(cone);
0038
0039        branchgroup.addChild(transformgroupCone);
0040
0041        TransformGroup transformgroupCube = new TransformGroup();
0042        transformgroupCube.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
```

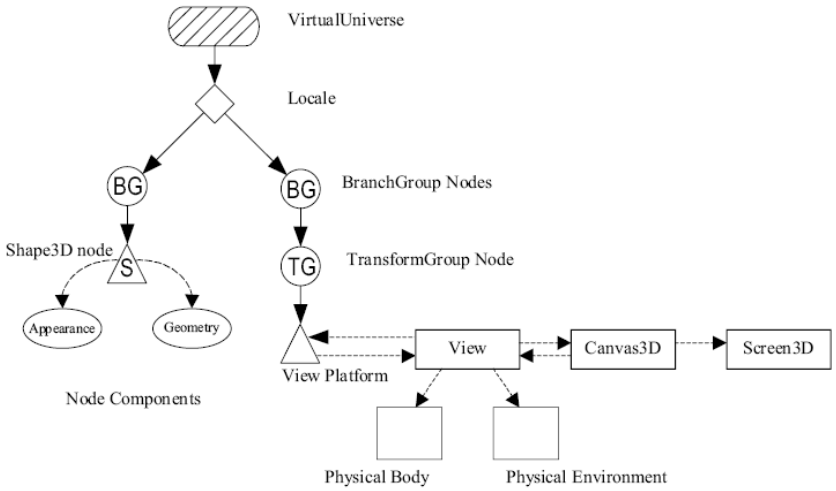
Exemplo Simplificado

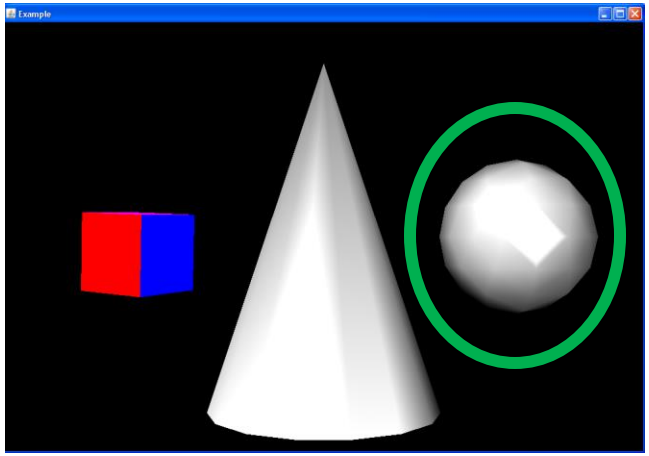





```
0039        branchgroup.addChild(transformgroupCone);
0040
0041        TransformGroup transformgroupCube = new TransformGroup();
0042        transformgroupCube.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE)
0043
0044        ColorCube colorcube = new ColorCube(0.1f);
0045        transformgroupCube.addChild(colorcube);
0046        transformgroupCone.addChild(transformgroupCube);
0047
0048        TransformGroup transformgroupSphere = new TransformGroup();
```

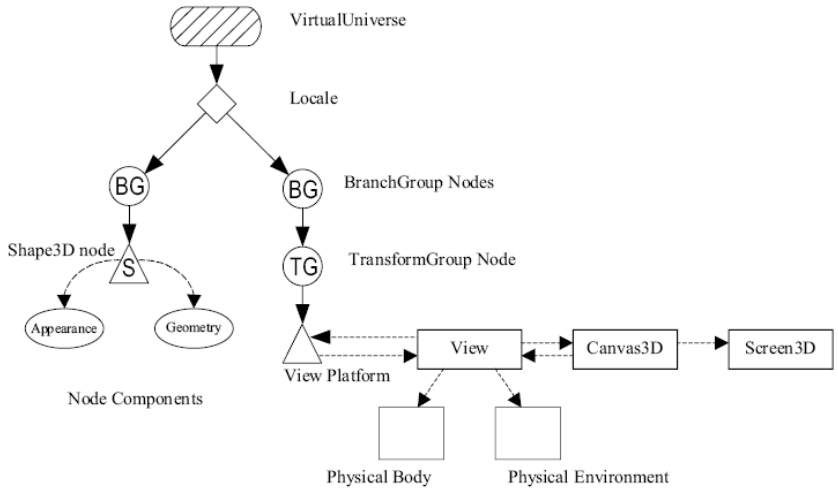
Exemplo Simplificado



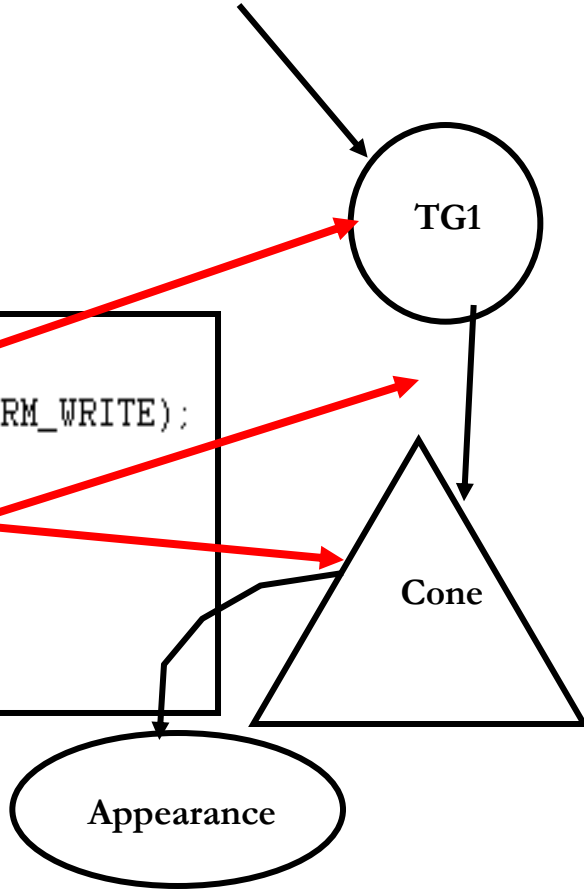



```
0048        TransformGroup transformgroupSphere = new TransformGroup();
0049        transformgroupSphere.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE)
0050
0051        Transform3D transform3dSphere = new Transform3D();
0052        transform3dSphere.setTranslation(new Vector3f(0.5f, 0.0f, 0.0f));
0053        transformgroupSphere.setTransform(transform3dSphere);
0054
0055        Sphere sphere = new Sphere(0.2f);
0056        transformgroupSphere.addChild(sphere);
0057        branchgroup.addChild(transformgroupSphere);
0058
0059        locale.addBranchGraph(branchgroup);
```

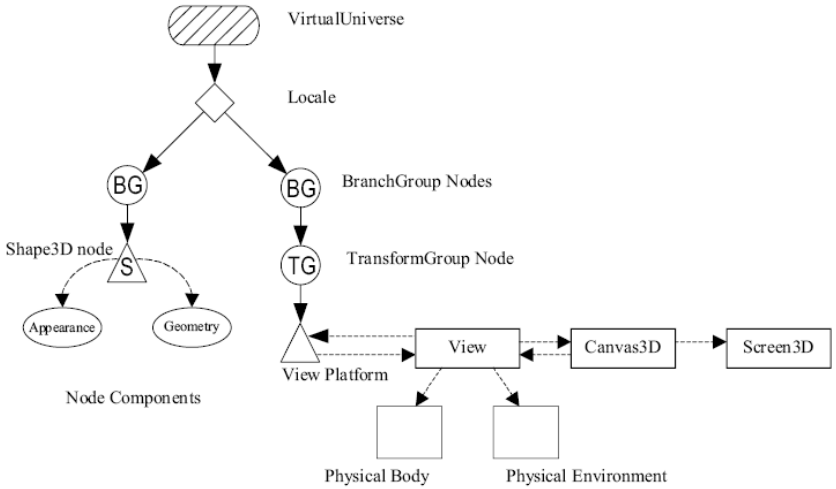
Exemplo Simplificado



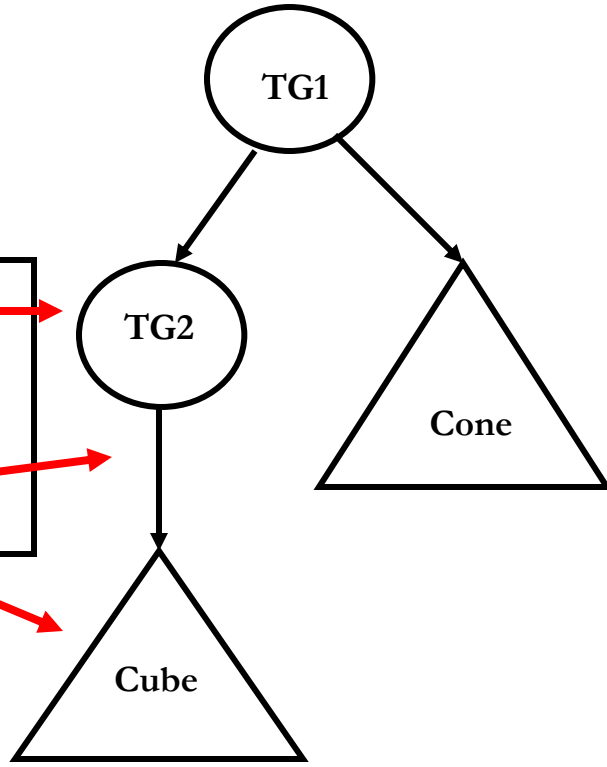

```
31
32      TransformGroup transformgroupCone = new TransformGroup();
33      transformgroupCone.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
34
35      Cone cone = new Cone(0.3f, 0.9f, createAppearance());
36
37      transformgroupCone.addChild(cone);
38
39      branchgroup.addChild(transformgroupCone);
40
```



Exemplo Simplificado




```
0040
0041      TransformGroup transformgroupCube = new TransformGroup();
0042      transformgroupCube.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
0043
0044      ColorCube colorcube = new ColorCube(0.1f);
0045      transformgroupCube.addChild(colorcube);
0046      transformgroupCone.addChild(transformgroupCube);
0047
```



Exemplo Simplificado

```
0065
0066        Mouse mouse = new Mouse(transformgroupCube, locale);
0067
0068        keyboard = new Keyboard(transformgroupCone, locale);
0069
0070        Glove glove = new Glove(transformgroupSphere, locale);
```

Teclado – Recursos da linguagem Java

- Interface *KeyListener*, implementa os métodos:

  -***KeyPressed:*** indica o pressionamento de uma tecla;

  -***KeyReleased***: indica a liberação de uma tecla;

  -***KeyTyped***: indica o pressionamento de uma tecla que não seja de ação.

- Método *getKeyChar()*: retorna string que identifica uma tecla

Exemplo Simplificado - Teclado

```
0015public class Keyboard implements KeyListener
0016{
0017
0018        public char tecla1 = ' ';
0019        public char tecla2 = ' ';
0020
0021
0022        public void keyPressed(KeyEvent event)
0023        {
0024           if(event.getKeyChar() == 'l')
0025           {
0026              tecla1 = 'l';
0027           }
0028           else if (event.getKeyChar() == 'k')
0029           {
0030              tecla2 = 'k';
0031           }
0032        }
0033
0034        public void keyReleased(KeyEvent event)
0035        {
0036           if (event.getKeyChar() == tecla1)
0037           {
0038              tecla1 = ' ';
0039           }
0040           else if (event.getKeyChar() == tecla2)
0041           {
0042              tecla2 = ' ';
0043           }
0044        }
0045
0046        public void keyTyped(KeyEvent event) {
0047
0048        }
```

Exemplo Simplificado - Teclado

```
0049
0050        private class KeyboardBehavior extends Behavior
0051        {
0052            private TransformGroup transformgroup;
0053            private Transform3D transform3d = new Transform3D();
0054
0055            ...
0056
0057            KeyboardBehavior (TransformGroup transformgroup)
0058            {
0059                this.transformgroup = transformgroup;
0060
0061            }
0062
0063            public void initialize()
0064            {
0065
0066                wAct = new WakeupOnElapsedFrames(0);
0067                wakeupOn(wAct);
0068            }
0069
0070            public void processStimulus(Enumeration criteria)
0071            {
0072                  if (tecla2 == 'k')
0073                  {
0074                     transform3d.setTranslation(new Vector3f(0.2f, 0.0f, 0.0f));
0075                     transformgroup.setTransform(transform3d);
0076                  }
0077                  else
0078                  {
0079                     transform3d.setTranslation(new Vector3f(0.0f, 0.0f, 0.0f));
0080                     transformgroup.setTransform(transform3d);
0081                  }
0082
0083            wakeupOn(wAct);
0084            }
0085
0086        }
0087    }
0088
```

Exemplo Simplificado - Teclado

```
0089
0090    public Keyboard (TransformGroup transformgroup, Locale locale)
0091    {
0092          BranchGroup branchgroup = new BranchGroup();
0093
0094          KeyboardBehavior myBehavior = new KeyboardBehavior(transformgroup);
0095          myBehavior.setSchedulingBounds(new BoundingSphere());
0096          branchgroup.addChild(myBehavior);
0097
0098          locale.addBranchGraph(branchgroup);
0099    }
```

Mouse

- Método *MouseRotate*: identifica movimentos de rotação (botão esquerdo pressionado acompanhado de movimento do dispositivo)
- Método *MouseTranslate*: identifica movimentos de translação nos eixos *x* e *y* (botão direito pressionado acompanhado de movimento do dispositivo)
- Método *MouseZoom*: identifica movimentos de translação no eixo *z* (terceiro botão pressionado acompanhado de movimento do dispositivo)

Exemplo Simplificado - Mouse

```
0006
0007public class Mouse
0008{
0009
0010    public Mouse(TransformGroup transformgroup, Locale locale)
0011    {
0012
0013        BranchGroup branchgroup = new BranchGroup();
0014
0015        MouseRotate mouserotate = new MouseRotate();
0016        mouserotate.setTransformGroup(transformgroup);
0017        mouserotate.setSchedulingBounds(new BoundingSphere());
0018        branchgroup.addChild(mouserotate);
0019
```

Behavior

Exemplo Simplificado - Mouse

```
0019
0020        MouseTranslate mousetranslate = new MouseTranslate();
0021        mousetranslate.setTransformGroup(transformgroup);
0022        mousetranslate.setSchedulingBounds(new BoundingSphere());
0023        branchgroup.addChild(mousetranslate);
0024
0025        MouseZoom mousezoom = new MouseZoom();
0026        mousezoom.setTransformGroup(transformgroup);
0027        mousezoom.setSchedulingBounds(new BoundingSphere());
0028        branchgroup.addChild(mousezoom);
0029
0030        locale.addBranchGraph(branchgroup);
0031    }
```

Exemplo Simplificado – Luva de Dados

```
0006
0007public class Glove
0008{
0009    NativeGlove nativeglove = new NativeGlove();
0010
0011    private class GloveBehavior extends Behavior
0012    {
0013        private TransformGroup transformgroup;
0014        private Transform3D transform3d = new Transform3D();
0015
0016        ...
0017
0018        GloveBehavior (TransformGroup transformgroup)
0019        {
0020            this.transformgroup = transformgroup;
0021        }
0022
0023
0024        public void initialize()
0025        {
0026            wAct = new WakeupOnElapsedFrames(0);
0027            wakeupOn(wAct);
0028        }
0029
```

Integração de Linguagens de Programação

- **Problema**: aplicação implementada em Java e *drivers* e bibliotecas de dispositivos não convencionais construídos usando linguagem de programação C/C++
- **Solução**: JNI (*Java Native Interface*)
  - Interoperação entre códigos escritos em diferentes linguagens de programação
  - Métodos em Java trocam informações com funções C/C++

Exemplo Simplificado - Integração

```
0003
0004public class NativeGlove
0005{
0006    public native int openGlove();
0007
0008    public native void closeGlove();
0009
0010    public native short[] getRawSensorData();
0011
0012    public native float[] getScaledSensorData();
0013
0014    public native void setCalibration(short upper, short lower);
0015
0016    public native int getGesture();
0017
0018    public native int getSensorRaw(int number);
0019
0020    public native float getSensorScaled(int number);
0021
0022    protected void finalize()
0023    {
0024        closeGlove();
0025    }
0026
0027    static
0028    {
0029        try
0030        {
0031            //Load DLL
0032            System.loadLibrary("5DTGlove");
0033
0034        }
0035        catch (Exception e) {
0036            System.out.println("Erro no Java Nativo: " +e.getMessage());
0037            e.printStackTrace();
0038        }
0039    }
0040}
```

Métodos Nativos

Exemplo Simplificado - Luva de Dados

```
0029
0030        public void processStimulus(Enumeration criteria)
0031        {
0032              if (verify == 0)
0033              {
0034                 verify = nativeglove.openGlove();
0035              }
0036
0037              if (verify == -1)
0038              {
0039                 System.out.println("Falha na conexao com a Luva de Dados");
0040              }
0041
0042              else if (verify == 1)
0043              {
0044
0045                 rawsensors = nativeglove.getRawSensorData();
0046
0047                 angle = rawsensors[0];
0048                 transform3d.setRotation(new AxisAngle4d (1.0f,-1.0f,-1.0f,angle);
0049                 transformgroup.setTransform(transform3d);
0050
0051                 wakeupOn(wAct);
0052               }
0053        }
0054     }
0055
```

Método Nativo

Exemplo Simplificado - Luva de Dados

```
0055
0056    public Glove (TransformGroup transformgroup, Locale locale)
0057    {
0058
0059          BranchGroup branchgroup = new BranchGroup();
0060
0061          GloveBehavior myBehavior = new GloveBehavior(transformgroup);
0062          myBehavior.setSchedulingBounds(new BoundingSphere());
0063          branchgroup.addChild(myBehavior);
0064
0065          locale.addBranchGraph(branchgroup);
0066    }
0067
0068}
```

Avaliação da Interação

- Aspectos computacionais (desempenho da aplicação)
  - ✓ tempo de resposta
  - ✓ *frames* por segundo (fps)
- Aspectos humanos (profissionais da área médica)
  - ✓ facilidade de uso
  - ✓ conforto com dispositivos
  - ✓ facilidade de aprendizado

Desempenho – fps

- Cálculo do número de fps

```
0001
0002        public void initialize()
0003        {
0004            wAct = new WakeupOnElapsedFrames(0);//Execução a cada frame
0005            wakeupOn(wAct);
0006        }
0007
0008        public void processStimulus(Enumeration criteria)
0009        {
0010        ...
```

Desempenho – fps

- Cálculo do número de fps

```
0001
0002
0003    //Verifica se o tempo decorrido foi de 1 segundo
0004    if (timeend >= 1000.0)
0005    {
0006          //Exibe a quantidade de frames por segundo
0007          System.out.println("Numero de frames - Haptico: " + framenumber);
0008
0009          timebegin = System.currentTimeMillis();
0010          framenumber = 0;
0011    }
0012
0013    //Incrementa a variável a cada iteração
0014    framenumber++;
0015    timeend = System.currentTimeMillis() - timebegin;
```

Conclusões

- Java3D
  - ✓ Gratuidade
  - ✓ Portabilidade
- Interação
  - ✓ Dispositivos convencionais
  - ✓ Dispositivos não convencionais (integração de linguagens de programação)

Exercícios (para entregar)

Para os exercícios a seguir, gere um único arquivo no formato PDF, com a solução dos exercícios na sequência solicitada.

1) Execute o programa exemplo detalhado nesta aula.

2) Altere o programa exemplo para:
   a) alterar a cor do objeto cone
   b) alterar a textura do objeto esfera
   c) incluir um objeto com formato .obj sempre que a tecla “Q” do teclado for acionada. O programa deve permitir rotacionar individualmente este objeto com o mouse (sem rotacionar os demais objetos da cena).
   Para este exercício: (i) gere um arquivo PDF que inclua o código fonte, destacando em cor diferente o trecho de código que foi alterado ou acrescentado; (ii) imprima a tela antes de acionar a tecla “Q” (sem o novo objeto), após acionar a tecla “Q” (com o novo objeto) e com o novo objeto rotacionado (após rotacionar com o mouse).

Fundamentos de Processamento Gráfico

Aula7

Realidade Virtual

Criação de Mundos Virtuais 3D Interativos com Java3D

Profa. Fátima Nunes

Material baseado em:

NUNES, F. L. S. ; CORRÊA, C. G. . Interação com Java3D. In: José Remo Ferreira Brega; Judith Kelner. (Org.). Interação com Realidade Virtual e Realidade Aumentada. 1ed.Bauru (SP): Canal 6, 2010, v. 1, p. 105-118