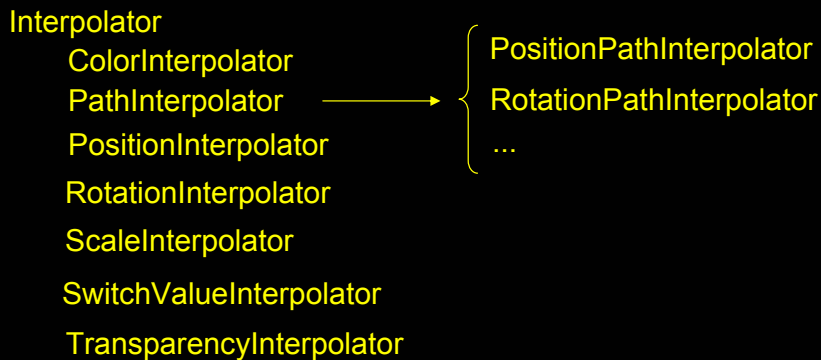


Animação

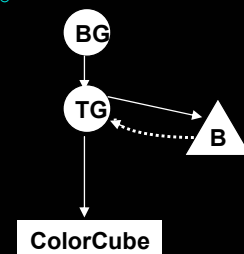
Interpolator class: classe abstracta que estende a classe **Behavior** e fornece vários métodos usados por subclasses de interpolação



1

Exemplo: HelloJava3Dc

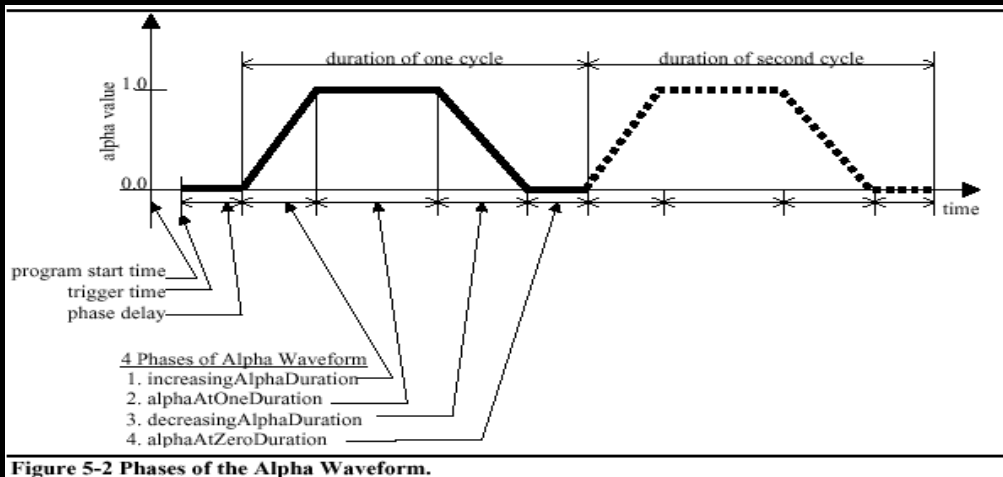
```
public BranchGroup createSceneGraph() {  
    // Create the root of the branch graph  
    BranchGroup objRoot = new BranchGroup();  
  
    // Create the transform group node and initialize it to the  
    // identity. Add it to the root of the subgraph.  
    TransformGroup objSpin = new TransformGroup();  
    objSpin.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);  
    objRoot.addChild(objSpin);  
  
    // Create a simple shape leaf node, add it to the scene graph.  
    // ColorCube is a Convenience Utility class  
    objSpin.addChild(new ColorCube(0.4));  
  
    // Create a new Behavior object that will perform the desired  
    // operation on the specified transform object and add it into  
    // the scene graph.  
    Alpha rotationAlpha = new Alpha(-1, 4000);  
  
    RotationInterpolator rotator =  
        new RotationInterpolator(rotationAlpha, objSpin);  
  
    // a bounding sphere specifies a region a behavior is active  
    // create a sphere centered at the origin with radius of 100  
    BoundingSphere bounds = new BoundingSphere();  
    rotator.setSchedulingBounds(bounds);  
    objSpin.addChild(rotator);  
  
    return objRoot;  
} // end of CreateSceneGraph method
```



2

ALPHA

- Objecto que produz um valor entre 0 e 1.
- Usado para controlar a evolução dos interpoladores



3

ALPHA

Características:

- StartTime de todos os Alpha → inicio do programa.
- Todos os tempos do ciclo são relativos ao StartTime.
- Objectos Alpha criados em momentos diferentes têm o mesmo StartTime → **Todos os interpoladores**, mesmo que baseados em diferentes Alpha, **estão sincronizados**.

4

ALPHA

Problema: Fazer com que um interpolador comece (ou recomece) num dado momento.

R: Criar outro Alpha não resulta uma vez que continuava no mesmo ponto do anterior. Pode ser efectuado através da seguinte instrução:

```
alpha.setStartTime(System.currentTimeMillis())
```

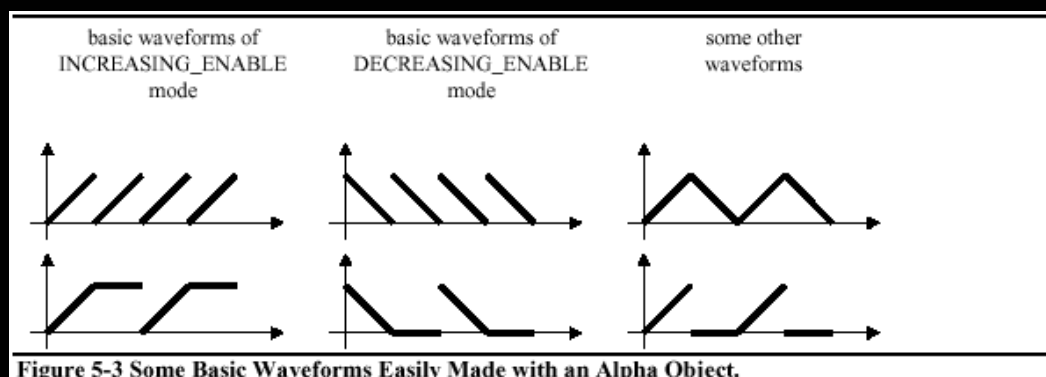
5

ALPHA

Construtores:

Alpha() loopCount : -1 mode : INCREASING_ENABLE triggerTime : 0
 phaseDelayDuration : 0 increasingAlphaDuration : 1000 increasingAlphaRampDuration : 0
 alphaAtOneDuration : 0 decreasingAlphaDuration : 0 decreasingAlphaRampDuration : 0
 alphaAtZeroDuration : 0

Alpha(int loopCount, long increasingAlphaDuration)



6

Exemplo de Interpolador: PositionPathInterpolator

`javax.media.j3d.PositionPathInterpolator`

Construtor:

```
PositionPathInterpolator(Alpha alpha, TransformGroup target,  
    Transform3D axisOfTranslation, float[] knots, Point3f[]  
    positions)
```

knots – vector com a indicação percentual da duração de cada segmento.

positions – vector com os pontos de controlo.

7

Exemplo de Interpolador: PositionPathInterpolator

```
Point3f[] pos = new Point3f[2];  
  
pos[0] = new Point3f (0.0f, 0.0f, 0.0f);  
pos[1] = new Point3f (4.0f, 0.0f, 0.0f);  
float[] knots = {0.0f, 1.0f};  
Transform3D axis = new Transform3D();  
  
long tk = 2000;  
  
Alpha alpha = new Alpha(1, Alpha.INCREASING_ENABLE, 0, 0, tk, 0, 0, 0, 0, 0);  
  
PositionPathInterpolator interp =  
    new PositionPathInterpolator(alpha, trgroup, axis, knots, pos);  
  
interp.setSchedulingBounds(new BoundingSphere());  
trgroup.addChild(ballInterp);
```

8

PositionPathInterpolator: redefinição da trajetória

```
// Redefinição dos vectores se tiverem tamanhos diferentes

pos = new Point3f[n];
knots = new float[n];

// atribuição dos novos valores
for(i=0; i<n; i++) pos[i] = new Point3f(. . . );
for(i=0; i<n; i++) knots[i] = (float). . . ;

// novo tk
tk = 1000;

// atribuição dos novos valores
interp.setPathArrays(pos, knots);

alpha.setIncreasingAlphaDuration(tk);
alpha.setStartTime(System.currentTimeMillis());
```

9

Comunicação entre *behaviors*: exemplo com interpolador

```
public class BallInterpolator extends PositionPathInterpolator {
    private boolean skipOnce = true;

    public BallInterpolator (Alpha alpha, TransformGroup target, Transform3D axis,
        float knots[], Point3f positions[]) {

        super (alpha, target, axis, knots, positions);
    }

    public void processStimulus (Enumeration criteria) {
        Alpha thisAlpha = getAlpha();
        if (skipOnce) { // primeira vez, posicao 0
            skipOnce = false;
            thisAlpha.setStartTime(System.currentTimeMillis());
        }

        if (thisAlpha.finished()) { // fim do ciclo
            if (!skipOnce)
                postId(SINAL); // envia mensagem, SINAL é uma constante inteira

            wakeupOn(defaultWakeupCriterion);
        }
        else
            super.processStimulus(criteria);
    }
}
```

10

Comunicação entre *behaviors*: *behavior* receptor

```
public void initialize() {
    WakeupCriterion myCriterion = new WakeupOnBehaviorPost(null, SINAL);
    wakeupOn (myCriterion);
}

public void processStimulus (Enumeration criteria) {
    WakeupCriterion wakeup;
    int id;

    while (criteria.hasMoreElements()) {
        wakeup = (WakeupCriterion) criteria.nextElement();
        if (wakeup instanceof WakeupOnBehaviorPost) {
            id = ((WakeupOnBehaviorPost)wakeup).getPostId();
            if (id == SINAL) {
                . . .
            }
        }
    }

    wakeupOn (myCriterion);
}
```