

RAGE

Scenegraphs / Node Controllers

```
// imports go here, including import ray.rage.rendersystem.*;
```

```
public class MyGame extends VariableFrameRateGame
{
    ...
    // constructor, main(), setupWindow(), setupCamera(),
    // update(), and lights setup as before
    ...

    protected void setupScene(Engine eng, SceneManager sm) throws IOException
    {
        Entity dolphinE = sm.createEntity("myDolphin", "dolphinHighPoly.obj");
        dolphinE.setPrimitive(Primitive.TRIANGLES);

        SceneNode dolphinN = sm.getRootSceneNode().createChildSceneNode("DolphinNode");
        dolphinN.moveBackward(2.0f);
        dolphinN.attachObject(dolphinE);
        ...

        // lights set up as before

        ...

        StretchController sc = new StretchController(); // user-defined node controller
        sc.addNode(dolphinN);
        sm.addController(sc);
    }
}
```

Custom Node Controller

```
import ray.rage.scene.*;
import ray.rage.scene.controllers.*;
import ray.rml.*;
```

```
public class StretchController extends AbstractController
{
    private float scaleRate = .003f; // growth per second
    private float cycleTime = 2000.0f; // default cycle time
    private float totalTime = 0.0f;
    private float direction = 1.0f;

    @Override
    protected void updateImpl(float elapsedTimeMillis)
    {
        totalTime += elapsedTimeMillis;
        float scaleAmt = 1.0f + direction * scaleRate;

        if (totalTime > cycleTime)
        {
            direction = -direction;
            totalTime = 0.0f;
        }

        for (Node n : super.controlledNodesList)
        {
            Vector3 curScale = n.getLocalScale();
            curScale = Vector3f.createFrom(curScale.x()*scaleAmt, curScale.y(), curScale.z());
            n.setLocalScale(curScale);
        }
    }
}
```