RAGE
Simple example game
"DolphinClick"

```
package myGame;

import java.awt.*;
import java.awt.event.*;
import java.io.*;

import ray.rage.*;
import ray.rage.game.*;
import ray.rage.rendersystem.*;
import ray.rage.rendersystem.Renderable.*;
import ray.rage.scene.*;
import ray.rage.scene.Camera.Frustum.*;
import ray.rage.scene.controllers.*;
import ray.rml.*;
import ray.rage.rendersystem.gl4.GL4RenderSystem;

public class SimpleTest extends VariableFrameRateGame
{
   // to minimize variable allocation in update()
   GL4RenderSystem rs;
   float elapsTime = 0.0f;
   String elapsTimeStr, counterStr, dispStr;
   int elapsTimeSec, counter = 0;
-----------------------------------------------------
   public SimpleTest()
   {  super();
      System.out.println("press T to render triangles");
      System.out.println("press L to render lines");
      System.out.println("press P to render points");
      System.out.println("press C to increment counter");
   }
-----------------------------------------------------
   public static void main(String[] args)
   {  Game game = new SimpleTest();
      try
      {  game.startup();
         game.run();
      }
      catch (Exception e)
      {  e.printStackTrace(System.err);
      }
      finally
      {  game.shutdown();
         game.exit();
   }  }
-----------------------------------------------------
   @Override
   protected void setupCameras(SceneManager sm, RenderWindow rw)
   {  SceneNode rootNode = sm.getRootSceneNode();
      Camera camera =
            sm.createCamera("MainCamera", Projection.PERSPECTIVE);
      rw.getViewport(0).setCamera(camera);
      SceneNode cameraNode =
            rootNode.createChildSceneNode(camera.getName()+"Node");
      cameraNode.attachObject(camera);
   }
   @Override
   protected void setupWindow(RenderSystem rs, GraphicsEnvironment ge)
   {  rs.createRenderWindow(new DisplayMode(1000, 700, 24, 60), false);
   }
-----------------------------------------------------
   @Override
   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(dolphinE.getName()
                                                      + "Node");
      dolphinN.moveBackward(2.0f);
      dolphinN.attachObject(dolphinE);

      sm.getAmbientLight().setIntensity(new Color(.1f, .1f, .1f));
      Light plight = sm.createLight("testLamp1", Light.Type.POINT);
      plight.setAmbient(new Color(.3f, .3f, .3f));
      plight.setDiffuse(new Color(.7f, .7f, .7f));
      plight.setSpecular(new Color(1.0f, 1.0f, 1.0f));
      plight.setRange(5f);

      SceneNode plightNode =
         sm.getRootSceneNode().createChildSceneNode("plightNode");
      plightNode.attachObject(plight);

      RotationController rc = new
      RotationController(Vector3f.createUnitVectorY(), .02f);
      rc.addNode(dolphinN);
      sm.addController(rc);
   }
-----------------------------------------------------
   @Override
   protected void update(Engine engine)
   {  // build and set HUD
      rs = (GL4RenderSystem) engine.getRenderSystem();
      elapsTime += engine.getElapsedTimeMillis();
      elapsTimeSec = Math.round(elapsTime/1000.0f);
      elapsTimeStr = Integer.toString(elapsTimeSec);
      counterStr = Integer.toString(counter);
      dispStr = "Time = " + elapsTimeStr + "   Keyboard hits = " + counterStr;
      rs.setHUD(dispStr, 15, 15);
   }
-----------------------------------------------------
   @Override
   public void keyPressed(KeyEvent e)
   {  Entity dolphin=getEngine().getSceneManager().getEntity("myDolphin");
      switch (e.getKeyCode())
      {  case KeyEvent.VK_L:
            dolphin.setPrimitive(Primitive.LINES);
            break;
         case KeyEvent.VK_T:
            dolphin.setPrimitive(Primitive.TRIANGLES);
            break;
         case KeyEvent.VK_P:
            dolphin.setPrimitive(Primitive.POINTS);
            break;
         case KeyEvent.VK_C:
            counter++;
            break;
      }
      super.keyPressed(e);
} }
```