RAGE / JAVA
MouseLook Controller / Custom Cursors

```
// Demonstrates using a Java "Robot" to recenter the mouse after each
// mouse move, keeping the mouse from ever reaching the screen edge.

(imports go here )

public class MyGame extends VariableFrameRateGame implements
                MouseListener, MouseMotionListener
{ private Robot robot;    // these are additional variable declarations
  private Canvas canvas;
  private RenderWindow rw;
  private RenderSystem rs;
  private float prevMouseX, prevMouseY, curMouseX, curMouseY;
  private boolean isRecentering;       //indicates the Robot is in action

//  constructor, main(), setupWindow(), and update() same as before
//  code to build ManualObject (pyramid) also the same as before
```

```
  protected void setupCameras(SceneManager sm, RenderWindow rw)
  { SceneNode rootNode = sm.getRootSceneNode();
    RenderSystem rs = sm.getRenderSystem();
    camera = sm.createCamera("MainCamera", Projection.PERSPECTIVE);
    rw.getViewport(0).setCamera(camera);

    camera.setRt((Vector3f)Vector3f.createFrom(1.0f, 0.0f, 0.0f));
    camera.setUp((Vector3f)Vector3f.createFrom(0.0f, 1.0f, 0.0f));
    camera.setFd((Vector3f)Vector3f.createFrom(0.0f, 0.0f, -1.0f));
    camera.setPo((Vector3f)Vector3f.createFrom(0.0f, 0.0f, 3.0f));

    SceneNode cameraN =
       rootNode.createChildSceneNode(camera.getName() + "Node");
    cameraN.attachObject(camera);
    camera.setMode('c');
    initMouseMode(rs, rw);
  }
```

```
  protected void setupScene(Engine eng, SceneManager sm)
                                            throws IOException
  { ManualObject pyr = makePyramid(eng, sm);
    SceneNode pyrN =
          sm.getRootSceneNode().createChildSceneNode("PyrNode");
    pyrN.attachObject(pyr);
    RotationController rc =
          new RotationController(Vector3f.createUnitVectorY(), .02f);
    rc.addNode(pyrN);
    sm.addController(rc);

    // set up lights (same as before)
  }
```

```
  private void initMouseMode(RenderSystem r, RenderWindow w)
  { rw = w;
    rs = s;
    Viewport v = rw.getViewport(0);
    int left = rw.getLocationLeft();
    int top = rw.getLocationTop();
    int widt = v.getActualScissorWidth();
    int hei = v.getActualScissorHeight();
    int centerX = left + widt/2;
    int centerY = top + hei/2;
    isRecentering = false;

    try  // note that some platforms may not support the Robot class
    { robot = new Robot(); } catch (AWTException ex)
    { throw new RuntimeException("Couldn't create Robot!"); }
    recenterMouse();
    prevMouseX = centerX;      // 'prevMouse' defines the initial
    prevMouseY = centerY;      //   mouse position

    // also change the cursor
    Image faceImage = new
                ImageIcon("./assets/images/face.gif").getImage();
    Cursor faceCursor = Toolkit.getDefaultToolkit().
          createCustomCursor(faceImage, new Point(0,0), "FaceCursor");
    canvas = rs.getCanvas();
    canvas.setCursor(faceCursor);
  }
```

```
  public void mouseMoved(MouseEvent e)
  { // if robot is recentering and the MouseEvent location is in the center,
    // then this event was generated by the robot

    if (isRecentering &&
          centerX == e.getXOnScreen() && centerY == e.getYOnScreen())
    { isRecentering = false; }     // mouse recentered, recentering complete
    else
    {  // event was due to a user mouse-move, and must be processed
      curMouseX = e.getXOnScreen();
      curMouseY = e.getYOnScreen();
      float mouseDeltaX = prevMouseX - curMouseX;
      float mouseDeltaY = prevMouseY - curMouseY;
      yaw(mouseDeltaX);
      pitch(mouseDeltaY);
      prevMouseX = curMouseX;
      prevMouseY = curMouseY;

      // tell robot to put the cursor to the center (since user just moved it)
      recenterMouse();
      prevMouseX = centerX;    //reset prev to center
      prevMouseY = centerY;
  } }
```

```
  private void recenterMouse()
  { // use the robot to move the mouse to the center point.
    // Note that this generates one MouseEvent.
    Viewport v = rw.getViewport(0);
    int left = rw.getLocationLeft();
    int top = rw.getLocationTop();
    int widt = v.getActualScissorWidth();
    int hei = v.getActualScissorHeight();
    centerX = left + widt/2;
    centerY = top + hei/2;
    isRecentering = true;
    robot.mouseMove((int)centerX, (int)centerY);
  }
```

```
  public void pitch(float mouseDeltaY)
  { float tilt;
    Vector3 f = camera.getFd();
    Vector3 r = camera.getRt();
    Vector3 u = camera.getUp();
    if (mouseDeltaY < 0.0) tilt = -1.0f;
      else if (mouseDeltaY > 0.0) tilt = 1.0f;
      else tilt = 0.0f;
    Vector3 fn = (f.rotate(Degreef.createFrom(0.1f * tilt), r)).normalize();
    Vector3 un = (u.rotate(Degreef.createFrom(0.1f * tilt), r)).normalize();
    camera.setFd((Vector3f)Vector3f.createFrom(fn.x(),fn.y(),fn.z()));
    camera.setUp((Vector3f)Vector3f.createFrom(un.x(),un.y(),un.z()));
  }
```

```
  public void yaw(float mouseDeltaX) { }    // not shown here
```

```
 // other mouse listener methods go here
}
```