TAGE / 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
{  private Robot robot;   // these are additional variable declarations
   private float curMouseX, curMouseY, centerX, centerY;
   private float prevMouseX, prevMouseY; // loc of mouse prior to move
   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
```

```
@Override
Public void initializeGame()
{  . . .
   initMouseMode();
}
```

```
private void initMouseMode()
{  RenderSystem rs = engine.getRenderSystem();
   Viewport vw = rs.getViewport("MAIN");
   float left = vw.getActualLeft();
   float bottom = vw.getActualBottom();
   float width = vw.getActualWidth();
   float height = vw.getActualHeight();

   centerX = (int) (left + width/2);
   centerY = (int) (bottom - height/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/textures/face.gif").getImage();
   Cursor faceCursor = Toolkit.getDefaultToolkit().
         createCustomCursor(faceImage, new Point(0,0), "FaceCursor");
   canvas = rs.getCanvas();
   canvas.setCursor(faceCursor);
}
```

```
@Override
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())
   {  // mouse recentered, recentering complete
      isRecentering = false;
   }
   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.

   RenderSystem rs = engine.getRenderSystem();
   Viewport vw = rs.getViewport("MAIN");
   float left = vw.getActualLeft();
   float bottom = vw.getActualBottom();
   float width = vw.getActualWidth();
   float height = vw.getActualHeight();
   int centerX = (int) (left + width/2.0f);
   int centerY = (int) (bottom - height/2.0f);

   isRecentering = true;
   robot.mouseMove((int)centerX, (int)centerY);
}
```

```
public void yaw(float mouseDeltaX)
{  float tilt;
   Camera c = engine.getRenderSystem()
                        .getViewport("MAIN").getCamera();
   Vector3f rightVector = c.getU();
   Vector3f upVector = c.getV();
   Vector3f fwdVector = c.getN();
   if (mouseDeltaX < 0.0) tilt = -1.0f;
   else if (mouseDeltaX > 0.0) tilt = 1.0f;
   else tilt = 0.0f;
   rightVector.rotateAxis(0.01f*tilt, upVector.x(),
                                 upVector.y(), upVector.z());
   fwdVector.rotateAxis(0.01f*tilt, upVector.x(),
                                 upVector.y(), upVector.z());
   c.setU(rightVector);
   c.setN(fwdVector);
}

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