

Forum:[Hack Forums](#)

Thread: [Java Random Walk](#)

WOLGA - July 2, 2015, 12:24 pm

Hello HackForums,

Hello! I messed around with LWJGL and I have made a random walk code. You can use the arrow keys to look/move around, and the left shift and left control will make you go faster and slower, respectively.

Please note that I am just beginning with Java, so if I moved out of convention, let me know! I enjoy feedback, too. I will also try to answer any questions.

Below is the code:

```
Code import org.lwjgl.LWJGLEException;
import org.lwjgl.input.Keyboard;
import org.lwjgl.opengl.Display;
import org.lwjgl.opengl.DisplayMode;
import org.lwjgl.opengl.GL11;

import java.util.Random;

public class randomWalk {

    public static void main(String[] args) {
        randomWalk display = new randomWalk();

        display.start();
    }
    public void start() {
        try {
            Display.setDisplayMode(new DisplayMode(800,600));
            Display.create();
            Display.setTitle("");
            Keyboard.create();
            GL11.glMatrixMode(GL11.GL_PROJECTION);
            GL11.glLoadIdentity();
            GL11.glOrtho(0, 800, 0, 600, 1, -1);
            GL11.glMatrixMode(GL11.GL_MODELVIEW);
        } catch (LWJGLEException e) {
            e.printStackTrace();
            System.exit(0);
        }

        int x = 400; int y = 300; int xd,yd;
        Random prng = new Random();
```

```

float toMoveY=0;
float toMoveX=0;
Line[] lines = new Line[25000];

while (!Display.isCloseRequested()) {

    GL11.glClear(GL11.GL_COLOR_BUFFER_BIT | GL11.GL_DEPTH_BUFFER_BIT);
    xd = prng.nextInt(9)-4;
    yd = prng.nextInt(9)-4;
    GL11.glTranslatef(-toMoveX, -toMoveY, 0);
    toMoveX = toMoveY = 0;

    lines[0] = new Line(x, y, (x+xd), (y+yd));
    for(int i=24999;i>0;i--) {
        lines[i] = lines[i-1];
    }
    for(Line l : lines) {
        if(l != null) {
            drawLine(l.x,l.y,l.x1,l.y1);
        }
    }

    x=x+xd;y=y+yd;

    Display.update();
    if(Keyboard.isKeyDown(Keyboard.KEY_UP)) {
        toMoveY=0.5f;
    }
    if(Keyboard.isKeyDown(Keyboard.KEY_DOWN)) {
        toMoveY=-0.5f;
    }
    if(Keyboard.isKeyDown(Keyboard.KEY_RIGHT)) {
        toMoveX=0.5f;
    }
    if(Keyboard.isKeyDown(Keyboard.KEY_LEFT)) {
        toMoveX=-0.5f;
    }
    if(Keyboard.isKeyDown(Keyboard.KEY_LSHIFT)) {
        toMoveX*=2;
        toMoveY*=2;
    }
    if(Keyboard.isKeyDown(Keyboard.KEY_LCONTROL)) {
        toMoveX*=0.5;
        toMoveY*=0.5;
    }

    Display.sync(960);
}
Keyboard.destroy();
Display.destroy();
}

```

```

public static void drawLine(double x, double y, double x1, double y1){

    GL11.glBegin(GL11.GL_LINES);
    GL11.glVertex2d(x,y);
    GL11.glVertex2d(x1,y1);
    GL11 glEnd();
}

public static class Line {
    double x;
    double y;
    double x1;
    double y1;

    public Line(double x, double y, double x1, double y1) {
        this.x = x;
        this.y = y;
        this.x1 = x1;
        this.y1 = y1;
    }
}
}

```

Sparta - July 2, 2015, 1:35 pm

Try creating classes in separate files instead of in other classes, if you're just learning. Also, I think LWJGL is updated to version 3, and if I'm not mistaken, the way windows and such are made are now different, but if you like sticking to the current version you're using, there really is no big issue about it; just a fair warning.

WOLGA - July 2, 2015, 1:57 pm

(Jul 2, 2015 01:35 PM)Sparta. Wrote: [\[CITE\]](#) Try creating classes in separate files instead of in other classes, if you're just learning. Also, I think LWJGL is updated to version 3, and if I'm not mistaken, the way windows and such are made are now different, but if you like sticking to the current version you're using, there really is no big issue about it; just a fair warning.

I never quite liked creating separate files. Is there a particularly compelling reason to do it?

JIM NASIUM'S GYMNASIUM - July 2, 2015, 2:07 pm

(Jul 2, 2015 01:57 PM)Qwertyuiopasdfghjklzxc Wrote: [\[CITE\]](#) I never quite liked creating separate files. Is there a particularly compelling reason to do it?

Organization. It might not seem important with such a small amount of code, but when you start working on real projects with thousands of lines of code, you really don't want them to all be in a single file. You might as well get used to it now.

MadZ - July 2, 2015, 2:09 pm

Multiple classes help you stay organized so you can work on specific things at once

Not necessary for small projects but if you are expanding into a game of sort then you should probably have separate classes depending on their functions

was sniped.