# Odyssey Now! v0.2.4

(See v0.2.3 for explanation of Version Numbers)

# Launching Odyssey Now!

1. Double click BUILD.
2. Double click LAUNCH.

If this doesn’t work, you likely don’t have the Java Runtime Environment installed.

The Command Prompt may appear while this occurs. This is normal.

You may get security warnings from Windows saying it blocked the run. I can tell you they’re safe, but of course it’s your choice if you believe it or not.

# Running From the Command Line

I wanted to make sure that developers could still compile and run the code from the command line if they so choose, so I ensured this compatibility remained inside the code, but due to the JSSC problem outlined below, you’ll need to explicitly state the dependency during compilation and running, in the style of:

javac -cp "./jssc.jar;" ODSYRunner.java

java -cp "./jssc.jar;" ODSYRunner

# TO DO:

The compatibility of Odyssey Now! with Mac and Linux is under evaluation. Ideally, this project should run on all three operating systems. The operation of the program while the controller was plugged in was examined, and needs improved on.

# Modifications from v0.2.3

Arduino

We’ve begun using the Arduino code from the previous groups to modify the microcontroller to only send values when they have changed instead of constantly sending them. However, due to the nature of the jittering of the program (the hardware seems to be acknowledging movement when there is none) we’re worried about the efficiency of this attempt.

The professor suggested implementing a 1-2 pixel/voltage leeway that will also be detected as non-movement from the device.

# Other Concerns

I’ve noticed that the SerialInput class opens and closes the port each time input is requested. I’m not sure why it does this, and am wondering if maybe some efficiency could be garnered by opening and leaving the port open.

# Moving Forward

Personally, I would recommend moving a copy of the Arduino code to the GitHub folder for easy access and to keep it accessible when the project goes live.

I also have interest in running the program with the aid of Java Visual VM or some other tool to help narrow down where the lag is coming from.