Scott Riddle

CPSC 334, Spring 2025

30 April 2025

Final Project Reflection

Techniques / tools used

Some techniques and tools I used to convert the project were make, Maven, GitHub actions, Debian packaging, bash scripting, and jar packaging.

Key steps to release

To release the game, I first used Maven to package my project into a jar file. Specifically, I had to use the maven shade plugin in order to package the game with its dependencies, such as the Dyn4j library. Once I had a jar file in the target/ directory, I aimed to package it in a Debian package. I used a bash script (pkg.bash) to create a temporary directory and create the opt/plinko and DEBIAN directories inside. It then copies my jar file into the opt/plinko directory as well as copies in the DEBIAN directory. It also changes the ownership of the opt directory to root so it can be installed with correct ownership on users' machines. Lastly, it builds the deb package and deletes the temp directory when it is finished. In my package job, it uploads the final deb package as an artifact.

Challenges

One of the first challenges I encountered was making the game actually run. I had to learn a lot about Maven and how it handles / packages dependencies, as well as where to put media files so Maven can package them into the jar. Once the game was able to run from the jar package, the next challenge was writing tests for the game, since we neglected to do this when we initially made the game a year ago. The tests were tricky because I had to test Java Swing components, as well as testing if an exception is correctly thrown on one method. Compared to these steps, actually converting the project to DevOps was the easy part. It only took me one try to get all my jobs to pass, which I am pretty proud of.

DevOps and future projects

Taking this course and making the counter project, I realized just how important unit testing, workflows, and packaging really are. In the past, the small and simple nature of my projects made testing and packaging seem like too much of a hassle. However, now that my projects are getting larger and more collaborative, the need for testing and packaging is apparent. I am now realizing that, if these tools are used consistently and appropriately, they give developers much more peace of mind on whether a project is in a working state or not. Especially if my future projects involve a team (like senior design), passing unit tests and actions will be a very important metric to check if a team member's branch is in a main-worthy state.

How this course has changed my perspective

I can honestly say that this course was not only one of the coolest classes I've ever taken, but it was also one of the most eye-opening. Coming into this course, I had used Linux before but would never consider using it as my daily driver OS. Now, my perspective has completely changed. As soon as I get the chance, I am switching my desktop OS to Fedora. This class made me realize the simple beauty of the Linux design as well as how easy (and powerful) it is to use over Windows. And, with gaming support significantly increasing for Linux in the past couple years, it's finally in a good state to where I can use it for everything and completely dump Windows.