**Author: John Shows**

**Software Design and Engineering**

**Lab Document**

|  |  |
| --- | --- |
| **High Level Purpose Statement:** | I’ll be using Maven to handle dependencies for external libraries. The goal is to learn how to use Maven to package my project so that it can be distributed to others as an easy to use executable file. |
| **Experimental Design:** | My project will be an experimental game created with the LibGDX framework. Because this framework as many other dependencies, I think it will be a perfect case to study for Maven. It’s typically recommended to use Girdle, but it can be used for Maven if you are targeting only Desktops. |
| **Resources Available:** | Documentation for Maven. Documentation for IntelliJ. YouTube will likely have plenty of videos outlining out to use Maven with LibGDX. Stack Overflow will also likely be useful to finding solutions to common problems that will arise. |
| **Time Estimate:** | Because the underlying project idea is substantial, I expect this project to take about 15-20 hours. Possibly more. I will learn about packing projects using Maven and double checking the executable file works as intended. |
| **Experiment Notes:** | Maven makes packaging your project very simple. However, I ran into issues with LibGDX and the way it handles paths for files. Maven, by default, moves all resource assets into the top level directory in the Jar file. If you hardcode the paths as they are in the project directory, your paths won’t be correct when Jar is created. To get around this, I had to use LibGDX’s own internal file handling class `Gdx.files.internal` to resolve this issue.  Other issues I had was that my Jar file initially did not contain dependencies from Gdx and others.  In my Pom file, I had to specify the use of the `maven-assembly-plugin` which insures that all necessary dependencies are packaged with the Jar file. |
| **Results:** | On my Linux devices, the results are a success. I can play game, however basic it is, just fine.  On my Windows machine, it states the Java version is not the correct version. I think this could be resolved by either updating the JRE on the Windows machine or target an older version of Java on the project build. However, I ran out of time and was unable to finish testing this. |
| **Consequences for the Future:** | I found Maven to pretty easy to use for the most part. The syntax in the pom file is pretty cryptic at first glance and I can tell will take a while before I am comfortable working with it on my own without copious amounts of online resources. However, the benefits it provides to packaging your software for distribution and handling dependencies is very useful. |