



COMP4050 - Software Engineering Practices

User Manual

School of Computing
Macquarie University, Balacava Rd,
Macquarie Park NSW 2109

Project Description

This is the project, wherein we are supposed to make a testing tool for processing that can be used in education as well as for tutors and lecturers. Due to this reason, our project is intended to tackle the problem by developing a CLI or GUI program that enables users, mostly tutors, to submit assignment submissions. The program will deliver the individual tests that each student's tutor had prepared for. These tests can provide brief results that include student cumulative grades, error warnings, and a list of all tests that passed or failed while the code was being executed.

The program will run the following tests:

- Runtime Test
 - o The program will execute the program being tested to check for any runtime errors.
- Static Analysis Test
 - o A static analysis tool will be run on the program to ascertain programming correctness and quality.
- JUNIT Tests
 - o JUNIT tests can be passed in by the User which will then get run on the testable program.

Project Requirements

To run the application you must have the following programs installed and configured:

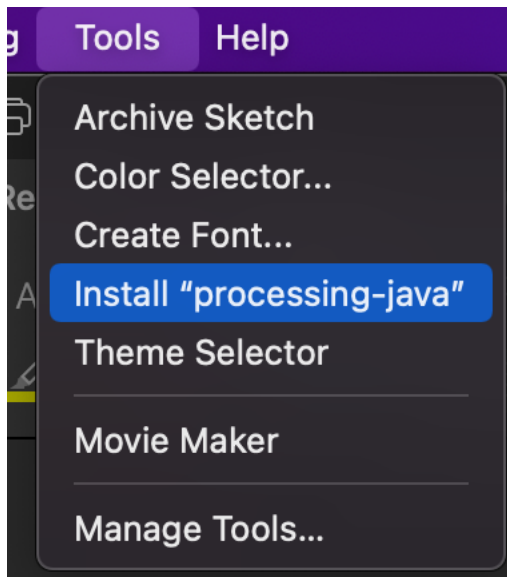
Operating System:

The current Operating Systems that the program supports is MAC OS and Windows.

Processing-java

Processing-java is a command line tool which is used to export processing code into its Java equivalent. To install processing-java you will need to download the Processing application from <https://processing.org/download>. Once installed you need to follow the below steps:

- Open Processing.
- Select Tools > "install processing-java"



Java Runtime

The program is packaged as a JAR file and so to run the program you will need to have a copy of the Java program which is capable of running the JAR file.

Java version 16.0.2 or higher is required.

Running the application

The application JAR file can be found in the releases folder in the parent directory of the program. The program takes a few arguments which are listed below:

Required arguments:

- -p <processing-java location>: The location of processing-java
- -t <location of project being tested>: The location of the project files being tested.

Optional arguments:

- -j: <JUNIT test file>: The location of the JUNIT test file which will be run against the testable program. An example file can be found in the test_artifacts directory.
- -r: A flag which when set will only run the runtime check test.
- -s: A flag which will only run the static analysis test.
- -m: letting the program know that the folder you are passing in the for -t will contain multiple projects to be tested. If this is the case the -t folder will need to be in the following format. You can have multiple student id folders:
 - o <parent_folder>/<student_id>/<project_folder>/<.PDE files>

If you run the program with no arguments, a help menu will be outputted showing all arguments that can be passed into the program.

Two example commands can be found below:

Testing one project:

```
java -jar hawaiian-0.1.0.jar -p "/Users/Jack/processing-java" -t  
"/Users/Jack/Documents/Processing/Flocking" -j  
"/Volumes/projects/Comp4050_Team2_PizzaCrew/test_artifacts/TestFile.java"
```

Testing multiple projects:

```
java -jar hawaiian-0.1.0.jar -p "/Users/Jack/processing-java" -t  
"/Users/Jack/Documents/Processing/multiple_projects" -j  
"/Volumes/projects/Comp4050_Team2_PizzaCrew/test_artifacts/TestFile.java" -m
```

Output File

The program will output the test results in a directory called Results in the same location as the JAR file. The format of the results files will be a comma delimited CSV.