Unit Tests

Product Name: ONI Code Visualization
Team Name: GraalVM UCSC
Date: April 11 2020

GraalVM Static Analysis Report to JSON:

- Navigate to graalvm/substratevm
- Run “mx build”
- Run “mx native-image -H:+PrintAnalysisCallTree HelloWorld” where HelloWorld is the program being run on
- Verify that the new JSON file is created in the graalvm/substratevm/reports folder
- Copy JSON file contents to a JSON formatter to verify contents are correct

Intellij IDE Plugin:

- JSON Reader:
    - Verify that a file was successfully imported.
    - Assert that the imported file is a JSON object.
    - Assert that the imported JSON file contains readable data.
    - Verify each field is the correct type.
- Plugin Functionality:
    - Verify that the right click menu contains a “Display Call Trace Visualization” and “Display Source Code”.
    - Verify that selected “Display Call Trace Visualization” opens a tool window in the IDE.
    - Verify that the tool window contains the text from the reports JSON file.
    - Verify that plugin can verify if selected text is a method call.
    - Verify error window opens on all cases that are not method calls.

System Test:

GraalVM Static Analysis Report to JSON:

- Navigate to graalvm/substratevm
- Run “mx build”
- Run “mx native-image -H:+PrintAnalysisCallTree HelloWorld” where HelloWorld is the program being run on
- Verify that the new JSON file is created in the graalvm/substratevm/reports folder
- Copy JSON file contents to a JSON formatter to verify contents are correct

JSON to Intellij Plugin:

- Change the “filepath” variable to the file path of the GraalVM Static Analysis Report JSON file.
- Set run configuration to “:runIDE”.
- Verify that “Display Call Trace Visualization” only works on functions
- Verify that text displayed in the tool window contains the data retrieved from the call trace report JSON.