README for Project Part 2

CSE 464
John Mitchell

Introduction

This project reads a DOT graph file and turns it into a Java Graph object. It then creates a text summary of the graph and also generates a picture of the graph using Graphviz. The goal of this part of the project is to show that my code can take a DOT file, parse it, build the graph, write a summary, and then create a real PNG image of the graph. I built everything using Java and Maven.

How My Program Works

When the program starts, it reads the DOT file by using my GraphParser class. The parser collects all the nodes and edges and stores them inside my Graph class. The Graph class keeps track of node names and the edges between them. It can also remove nodes and edges, print information, and return sets of nodes and edges.

My GraphExporter class creates DOT files and graphic files. It first writes a temporary DOT file and then uses the dot tool from Graphviz to turn that DOT file into a PNG picture. I installed Graphviz so that the dot command works on my laptop.

My Main class is the program that brings everything together. It takes three command line arguments. The first one is the input DOT file. The second one is a text file where the program writes the summary. The third one is the PNG image the program generates. Main reads the graph, prints it, writes the summary, and then tells GraphExporter to make the PNG picture.

How to Run the Program

This is the command I used to run my program:

java -cp target/CSE464_2025_jmmitc15-1.0-SNAPSHOT.jar edu.asu.graph.Main sample.dot summary.txt sample.png

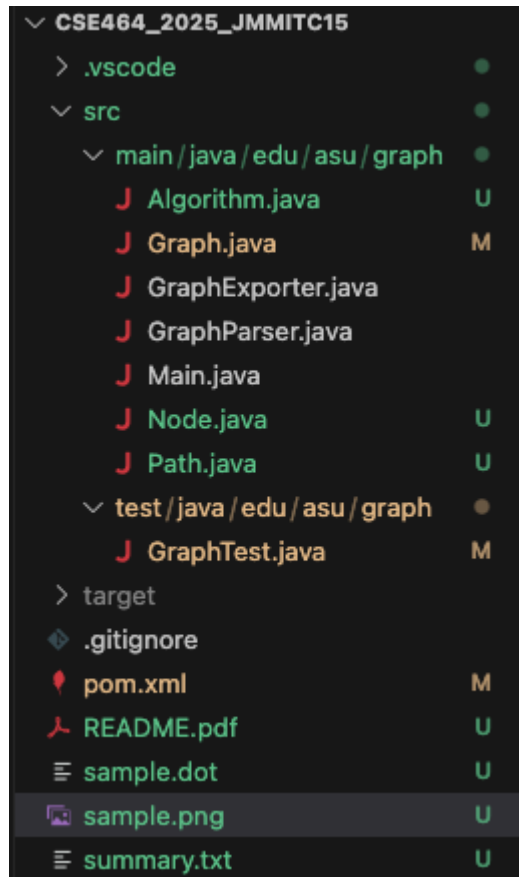This command builds the graph from sample.dot and creates summary.txt and sample.png.

Proof That My Code Works

1.

```
johnnymitchell@MacBook-Air-9 CSE464_2025_jmmitc15 % java -cp target/CSE464_2025_jmmitc15-1.0-SNAPSHOT.jar edu.asu.graph.Main sample.dot summary.txt sample.png

Parsed graph:
Number of nodes: 4
Nodes: [a, b, c, d]
Number of edges: 3
Edges:
a -> b
b -> c
c -> a

Wrote summary to summary.txt
Wrote PNG to sample.png
```
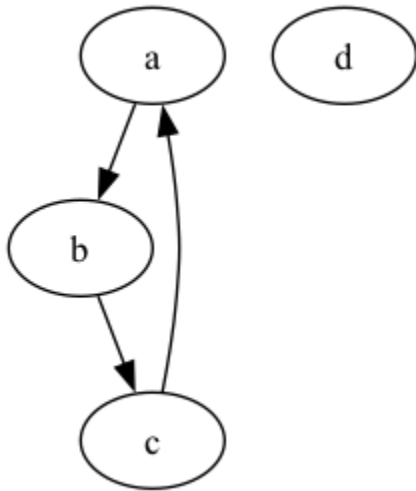
I included a screenshot that shows me running the command above. The output shows the parsed graph information along with the messages that the summary file and PNG file were created.

2.

I included a screenshot of my updated file tree. It shows that summary.txt and sample.png were created in my project folder.

3.

I included a screenshot showing the PNG file opened on my screen. This proves that Graphviz created the picture correctly.

4.


```
≡ summary.txt U ✕
    1    Number of nodes: 4
    2    Nodes: [a, b, c, d]
    3    Number of edges: 3
    4    Edges:
    5    a -> b
    6    b -> c
    7    c -> a
```

I included a screenshot showing the contents of summary.txt. This proves that my program wrote the summary text file correctly.

5.

```
johnnymitchell@MacBook-Air-9 CSE464_2025_jmmitc15 % mvn test
[INFO] Processing build on daemon c60dc1d3
[INFO] Scanning for projects...
[INFO] BuildTimeEventSpy is registered.
[INFO]
[INFO] Using the SmartBuilder implementation with a thread count of 7
[INFO]
[INFO] --------------------< edu.asu:CSE464_2025_jmmitc15 >--------------------
[INFO] Building CSE464_2025_jmmitc15 1.0-SNAPSHOT
[INFO]     from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ CSE464_2025_jmmitc15 ---
[INFO] skip non existing resourceDirectory /Users/johnnymitchell/Downloads/CSE464_2025_jmmitc15/src/main/resources
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ CSE464_2025_jmmitc15 ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ CSE464_2025_jmmitc15 ---
[INFO] skip non existing resourceDirectory /Users/johnnymitchell/Downloads/CSE464_2025_jmmitc15/src/test/resources
[INFO]
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ CSE464_2025_jmmitc15 ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- surefire:3.2.3:test (default-test) @ CSE464_2025_jmmitc15 ---
[INFO] Using auto detected provider org.apache.maven.surefire.junitplatform.JUnitPlatformProvider
[INFO]
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running edu.asu.graph.GraphTest
[INFO] Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.055 s -- in edu.asu.graph.GraphTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 7, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] -------------------------------------------------------
[INFO] Total time:  0.415 s (Wall Clock)
[INFO] Finished at: 2025-11-14T16:53:39-07:00
[INFO] -------------------------------------------------------
```

I included a screenshot showing that all JUnit tests pass with no failures. The message shows "BUILD SUCCESS" and that all seven tests passed.

Conclusion

This part of the project shows that my graph program is fully working from start to finish. It reads DOT files, stores the graph data, writes a clean summary, and creates PNG pictures with Graphviz. All of my tests pass, and all required output files were generated. My project now meets all Part Two requirements.