Project One

Christina Torres

Spring 2023

CMSC 330: Advanced Programming Languages

Professor Alin Suciu

UML Class Diagram (from assignment instructions)

A diagram of a computer

Description automatically generated

## Developer’s Guide

|  |  |
| --- | --- |
| 1. Use directory to navigate to the file location. | 1. Compile and run the main file.   javac Main.java && java Main |
| 1. Select your file with the dimensions.   A screenshot of a computer program  Description automatically generated | 1. Watch the result. A screenshot of a computer     Description automatically generated |

## Test Cases

| # | Description |  | P/F |
| --- | --- | --- | --- |
| 1 | **Show All Shapes Correctly** | **Input:**    **scene.txt:**    **Output:** | PASS |
| 2 | **Wrong File Type** | **Input:**    **Output:** | Pass |
| 3 | **Syntactic Incorrect scene File** | **Input:**  **scene-bad-syn.txt:**    **Output:** | Pass |

## What I Learned

I learned many invaluable lessons when creating this program. I am still getting use to a GUI, so this is interesting to me. I’m great with HTML and CSS, but not so much Java GUI things. I am growing to love it though. It is quite fun to see the different attributes and panels. I learned that the window is probably the frame and the sections of it are usually scenes. There is where the different objects live.

I learned quite a bit about how language translators and compilers work. Code starts as source code. It then goes to the Lexical Analyzer which can sometimes be called scanner. It results in a token stream. This token stream is then passed through a syntactic analyzer. A parse tree is created. In some languages there is an intermediate code generated which returns intermediate code. This then goes through a final code generator in some languages. Object code is the result.

I have a greater appreciation of compiling now! Thank you. [GitHub Repo](https://github.com/xtina-lt/cmsc330-compiling-shapes)