LCL Programming Language Developed by Luis F. Ruelas and Eduardo Nieves

Introduction:
LCL is a simple programming language that allows a user to simulate a logical circuit with a few lines of code which are parsed into a Java intermediate code. After the successful compilation and execution of LCL intermediate code the user will be presented with their circuit in a GUI that allows them to see the result of their circuit simulation as well as edit the current circuit in real time to simulate other logical circuits by following the LCL code syntax.

Syntax:
The language allows for two types of declarations: input declarations and circuit declarations. The input declarations must precede the circuit ones and both must start with a single capital letter. The inputs are constructed as follows: capital letter naming the input, equal sign('=') and either a '1' or a '0'. For circuit declarations the user can use one of six logical gate operations: 'and','or','xor','nand','nor', and 'xnor' all of which have a limit of two inputs.. A circuit declaration is constructed as follows: capital letter naming the circuit, equal sign('='),input name, one of the six logical operations('and','or', etc) and another input name. A circuit may have various operations and they can be separated by a parenthesis for clearer understanding however please note the construction of the circuit is not altered by the use of parentheses. A circuit declaration may use the name of another circuit's name as one of the input names in one of its logical operators, if that circuit has been declared before it. Construction of circuit is done from left to right.

Example LCL Code:
```
A=1
B=0
C=1
D=AandBorC
E=CorD
F=(AxorE)nor(BandC)
```

LCL Requirements:
a)Latest Java JDK installed(Java 9 or higher).
b)Python 2.7 installed.
c)Files found in LCL repository.

LCL Compilation and Execution Instructions:
1)Download zip file of LCL repository.
2)Unzip LCL repository file in an adequate location.

3)Open Terminal and cd into the directory containing the LCL files.
4)To create your LCL code, open and edit the text file: ‘lclCode.txt’
5)Save your changes to the ‘lclCode.txt’ file.
6)Now in the Terminal write: python LCLAPP.py and press ‘Enter’
7)After receiving successful confirmation message the your Java intermidiate code is ready to be compiled and executed.
8)For Java compilation, in the Terminal write: ‘javac -d runnable -sourcepath LCLCompilable LCLCompilable/app/LCLApplication.java’ and press ‘Enter’
9)To run your Java compiled code, in the Terminal write: ‘java -classpath runnable app.LCLApplication’ and press ‘Enter’.
10) You will be presented with a GUI showcasing your logical circuit simulation.

GUI Instructions:

1)If you want to edit the inputs values or edit the logical circuit created, press the Edit Inputs button in right side of the navigation bar (See figure 1) .
2) You will be presented with a GUI showcasing your input values and the logical circuit code (See Figure 2).
3) Make the desired changes following the LCL code syntax.
4) Once finished press the Run button, a message will appear indicating either a successful Circuit Simulation or an error (See figure 3 & 4).
5) If there was no errors, you should see your new circuit shown in the GUI.

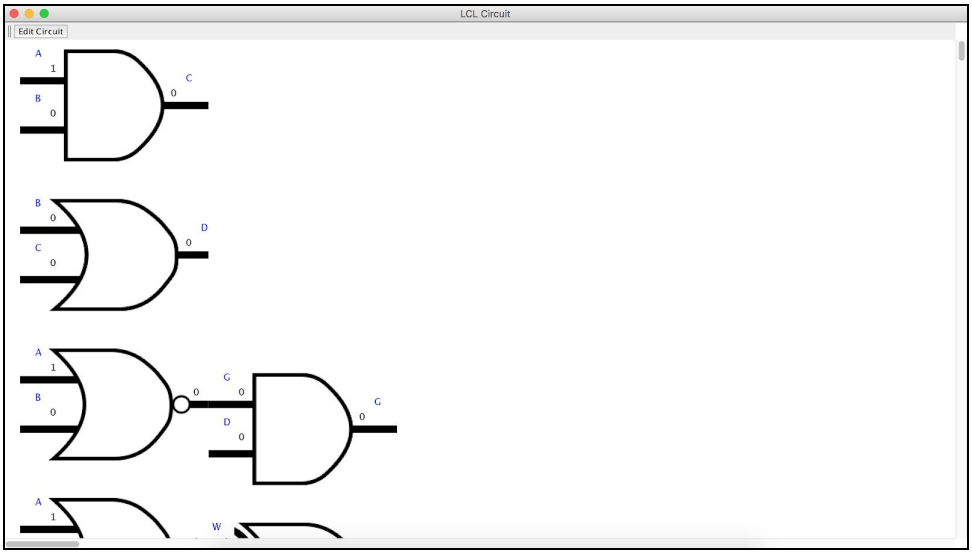

Figure 1: LCL Circuit GUI

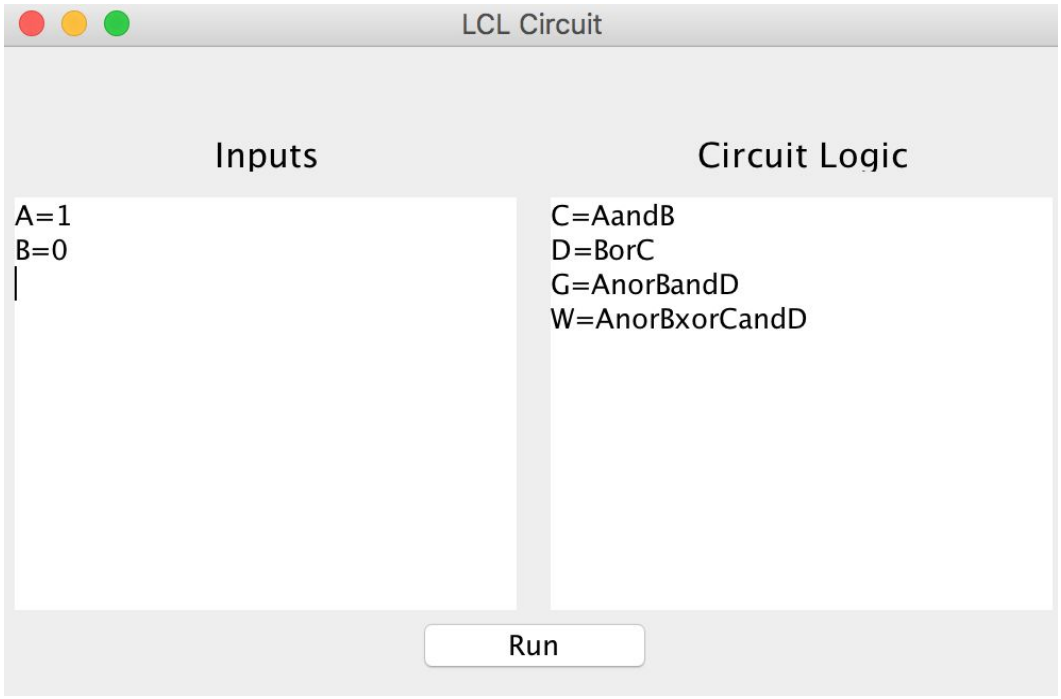

Figure 2: Edit Circuit GUI.

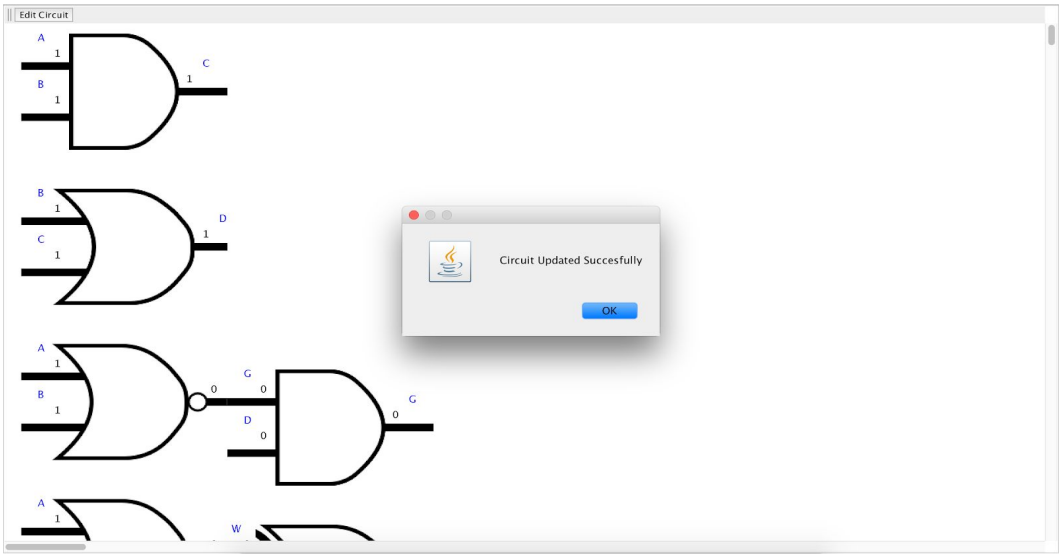

Figure 3: Success Message.

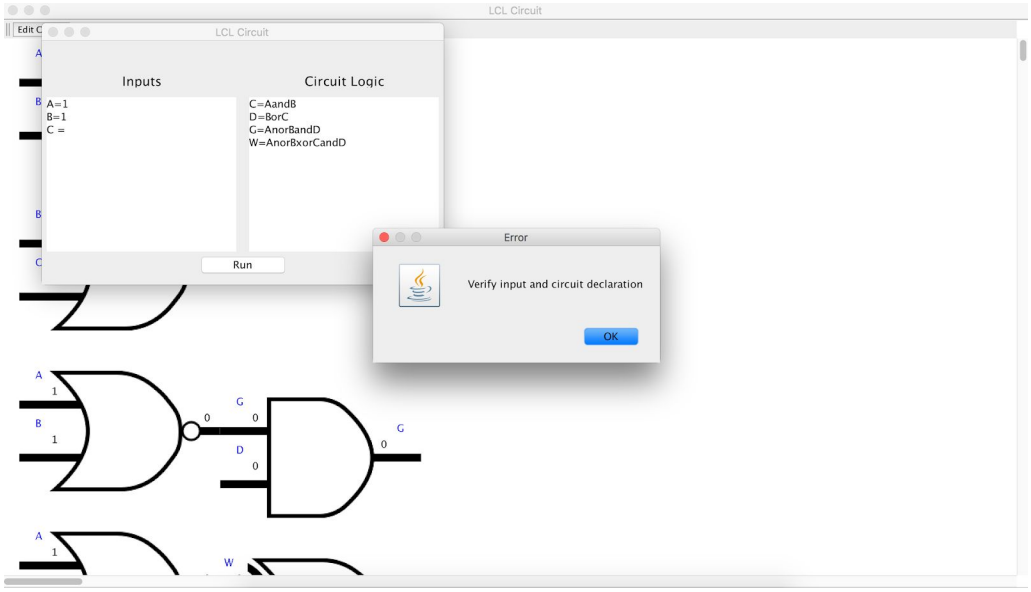

Figure 4: Error Message.