CS 2110 Homework 1
Intro to Circuitsim

Maddie Brickell, Daniel Becker, and Jim Harris

Spring 2019

Contents

1 Setup

The software you will be using for this assignment and all future circuit based assignments is called CircuitSim.

CircuitSim.v1.jar has been included in this assignment and the proper version will have a permanent location on canvas at Files/Tools/CircuitSim
Please only use the jar file we provide for circuitsim. This has the correct version. CircuitSim downloaded elsewhere may not work with our grader as intended. You have been warned.

Note that you must have Java 8u152, Java 9, or Java 10 installed for CircuitSim to work properly. If you have an older version of Java (even Java 8u151), you will encounter several annoying JavaFX bugs. You have been warned.

2 Assignment

CircuitSim is an interactive circuit simulation package. We will be using this program for the next couple of homework assignments. Before you start please ensure that you have at least Java 8u152 (JDK) installed on your computer. This may be a problem if you installed Java during CS 1331 last semester and never updated. Make sure you update to the latest version of Java 8! CircuitSim will only run on machines with at least Java 8 installed. Your next homework will involve programming in Java so make sure you have the JDK installed as well.

CircuitSim is a powerful simulation tool designed for educational use. This gives it the advantage of being a little more forgiving than some of the more commercial simulators. However, it still requires some time and effort to be able to use the program efficiently. With this in mind, we present you with the following assignment:

2.1 Part 1 — Read Resources

Read through the following resources

- CircuitSim Wires Documentation `https://ra4king.github.io/CircuitSim/docs/wires/`
- Tutorial 1: My First Circuit `https://ra4king.github.io/CircuitSim/tutorial/tut-1-beginner`

2.2 Part 2 — Complete Tutorial 2

Complete Tutorial 2 `https://ra4king.github.io/CircuitSim/tutorial/tut-2-xor`

Instead of saving your file as **xor.sim**, save your file as **part1.sim**. As well, make sure you label your two inputs **a** and **b**, and your output as **c**.

2.3 Part 3 — Complete Tutorial 3

Complete Tutorial 3 `https://ra4king.github.io/CircuitSim/tutorial/tut-3-tunnels-splitters`

Name the subcircuit **umbrella**, the input **in**, and the output **out**. Save your file as **part2.sim**.

3 Checker

To run the checker, run

```
$ java -jar hw01-tester.jar
```

at a command prompt in the same directory as `part1.sim` and `part2.sim`. Gradescope uses this same checker to evaluate your work.

4 Deliverables

Submit the following files:

- `part1.sim`
- `part2.sim`

to Gradescope under the assignment "Homework 1."

5 Rules and Regulations

5.1 General Rules

1. Starting with the assembly homeworks, any code you write must be meaningfully commented. You should comment your code in terms of the algorithm you are implementing; we all know what each line of code does.
2. Although you may ask TAs for clarification, you are ultimately responsible for what you submit. This means that (in the case of demos) you should come prepared to explain to the TA how any piece of code you submitted works, even if you copied it from the book or read about it on the internet.
3. Please read the assignment in its entirety before asking questions.
4. Please start assignments early, and ask for help early. Do not email us the night the assignment is due with questions.
5. If you find any problems with the assignment it would be greatly appreciated if you reported them to the author (which can be found at the top of the assignment). Announcements will be posted if the assignment changes.

5.2 Submission Conventions

1. All files you submit for assignments in this course should have your name at the top of the file as a comment for any source code file, and somewhere in the file, near the top, for other files unless otherwise noted.
2. When preparing your submission you may either submit the files individually to Canvas/Gradescope or you may submit an archive (zip or tar.gz only please) of the files. You can create an archive by right clicking on files and selecting the appropriate compress option on your system. Both ways (uploading raw files or an archive) are exactly equivalent, so choose whichever is most convenient for you.
3. Do not submit compiled files, that is .class files for Java code and .o files for C code. Only submit the files we ask for in the assignment.
4. Do not submit links to files. The autograder does not understand it, and we will not manually grade assignments submitted this way as it is easy to change the files after the submission period ends.

5.3 Submission Guidelines

1. You are responsible for turning in assignments on time. This includes allowing for unforeseen circumstances. If you have an emergency let us know **IN ADVANCE** of the due time supplying documentation (i.e. note from the dean, doctor's note, etc). Extensions will only be granted to those who contact us in advance of the deadline and no extensions will be made after the due date.

2. You are also responsible for ensuring that what you turned in is what you meant to turn in. After submitting you should be sure to download your submission into a brand new folder and test if it works. No excuses if you submit the wrong files, what you turn in is what we grade. In addition, your assignment must be turned in via Canvas/Gradescope. Under no circumstances whatsoever we will accept any email submission of an assignment. Note: if you were granted an extension you will still turn in the assignment over Canvas/Gradescope.

3. There is a 6-hour grace period added to all assignments. You may submit your assignment without penalty up until 11:55PM, or with 25% penalty up until 5:55AM. So what you should take from this is not to start assignments on the last day and plan to submit right at 11:54AM. You alone are responsible for submitting your homework before the grace period begins or ends; neither Canvas/Gradescope, nor your flaky internet are to blame if you are unable to submit because you banked on your computer working up until 11:54PM. The penalty for submitting during the grace period (25%) or after (no credit) is non-negotiable.

5.4 Syllabus Excerpt on Academic Misconduct

Academic misconduct is taken very seriously in this class. Quizzes, timed labs and the final examination are individual work.

Homework assignments are collaborative, In addition many if not all homework assignments will be evaluated via demo or code review. During this evaluation, you will be expected to be able to explain every aspect of your submission. Homework assignments will also be examined using computer programs to find evidence of unauthorized collaboration.

What is unauthorized collaboration? Each individual programming assignment should be coded by you. You may work with others, but each student should be turning in their own version of the assignment. Submissions that are essentially identical will receive a zero and will be sent to the Dean of Students' Office of Academic Integrity. Submissions that are copies that have been superficially modified to conceal that they are copies are also considered unauthorized collaboration.

You are expressly forbidden to supply a copy of your homework to another student via electronic means. This includes simply e-mailing it to them so they can look at it. If you supply an electronic copy of your homework to another student and they are charged with copying, you will also be charged. This includes storing your code on any site which would allow other parties to obtain your code such as but not limited to public repositories (Github), pastebin, etc. If you would like to use version control, use github.gatech.edu

5.5 Is collaboration allowed?

Collaboration is allowed on a high level, meaning that you may discuss design points and concepts relevant to the homework with your peers, share algorithms and pseudo-code, as well as help each other debug code. What you shouldn't be doing, however, is pair programming where you collaborate with each other on a single instance of the code. Furthermore, sending an electronic copy of your homework to another student for them to look at and figure out what is wrong with their code is not an acceptable way to help them, because it is frequently the case that the recipient will simply modify the code and submit it as their own. Consider instead using a screen-sharing collaboration app, such as `http://webex.gatech.edu/`, to help someone with debugging if you're not in the same room.

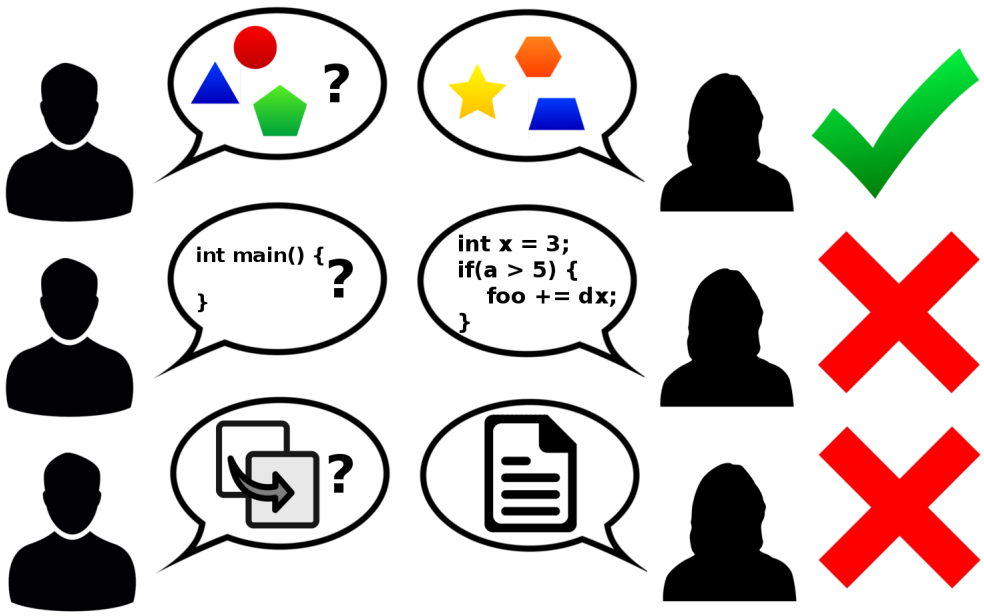

Figure 1: Collaboration rules, explained colorfully