Project Report - Implement and visualize the Election in a Complete Graph

Zefan Liang

School of Electrical Engineering and Computer Science

University of Ottawa

10th April, 2020

CSI 5308 Principles Distributed Compute 2020

Table of content

1 Introduction

This report contains how to implement and visualize the election in a complete graph algorithm in the class.

2 Platform and language

The program is completed by using java language (JDK11) and the IDE of the program that I use is IntelliJ IDEA.

3 Project structure and usage

The project consists of several parts which we can see it in the figure 1.
The library I use in the project is javafx-15.

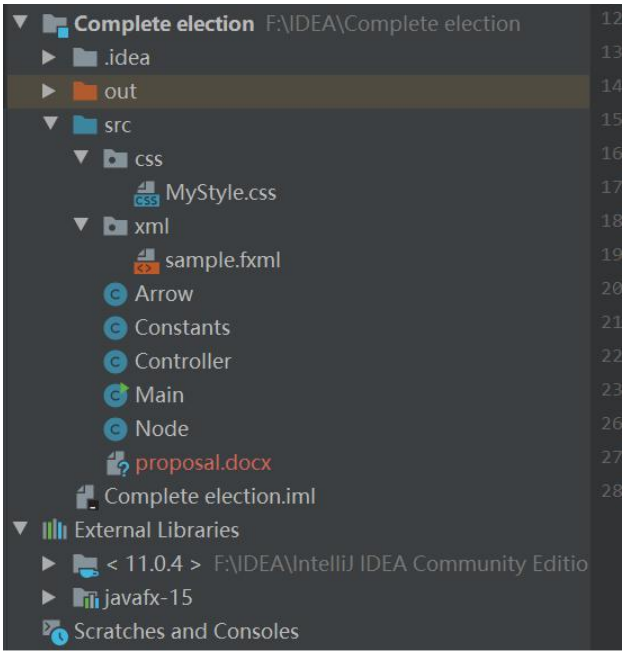

Figure 1 – the project structure

The role of *Main class* is to draw the scene of the program. I define *class Node* and *class Arrow* which represents the node and the line connecting the node.

In the *Controller*, I define how to create the node and create the line connecting the node. You can add them by left click on the scene. Each node will generate a character in lower case (like a, b, c, d...) and a random number (as node ID) between 0 and 10 (like figure 2). Controller also binds the *TextField, TextArea, Console, Submit Button* and *Reset Button* in the scene we have generated.

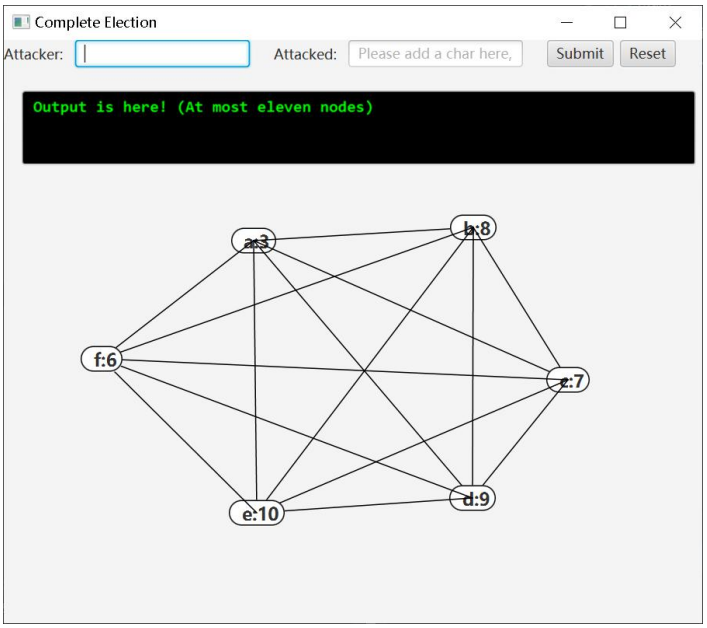

Figure 2 – the scene I generate for example

I create the *sample.fxml* to show the basic page. In figure 2, you should put the character which represents the node behind “Attacker” and “Attacked”. After you click on *submit*. The console will output the result of this attack which is shown in figure 3.

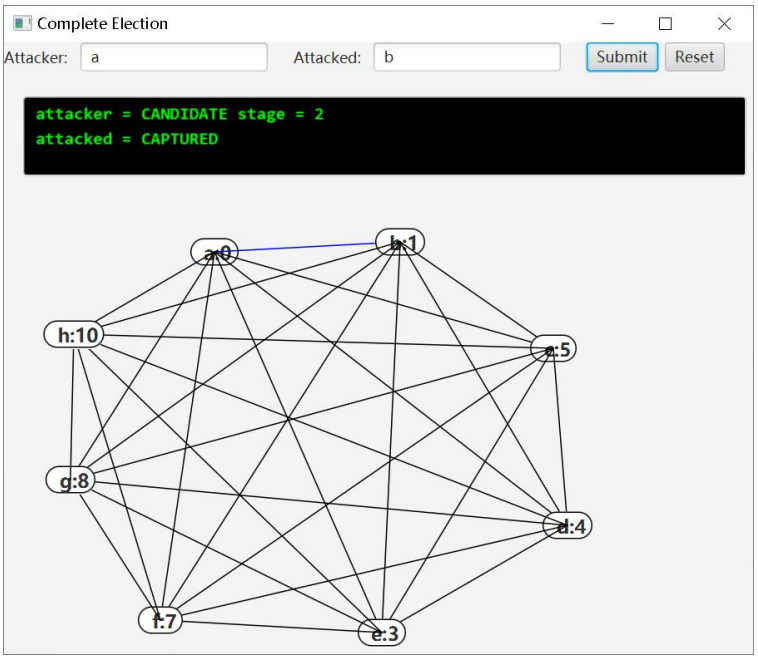

Figure 3 – the scene shows the result which after click submit

The style of the node and *Textfield* is defined in *MyStyle.css*. If you enter the incorrect format in *Textfield*, the bolder of the *Textfield* will become red and there will generate an error sentence in the *Textfield* (like figure 4).

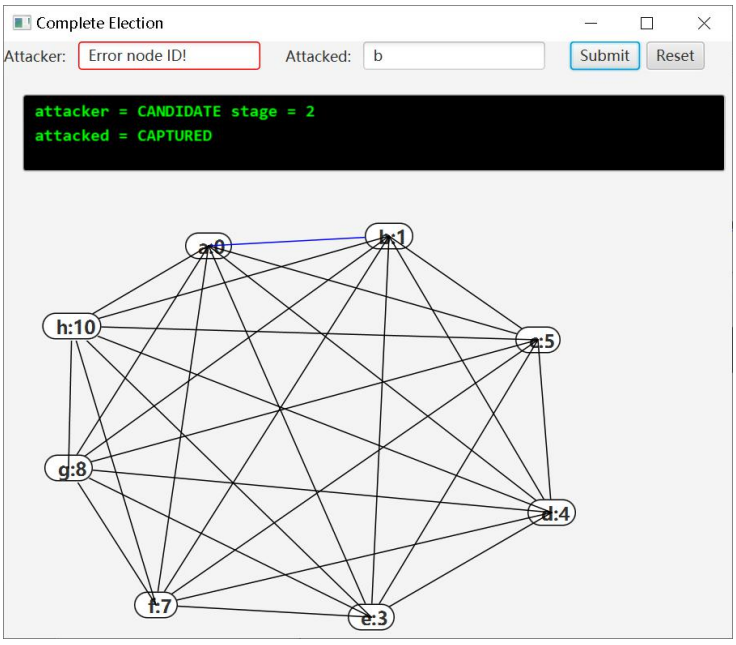

Figure 4 – the scene shows the situation when you enter the incorrect format in Textfield

When the algorithm ends, the winner is shown in the console (like figure 5). And you can click on *reset* to restart.

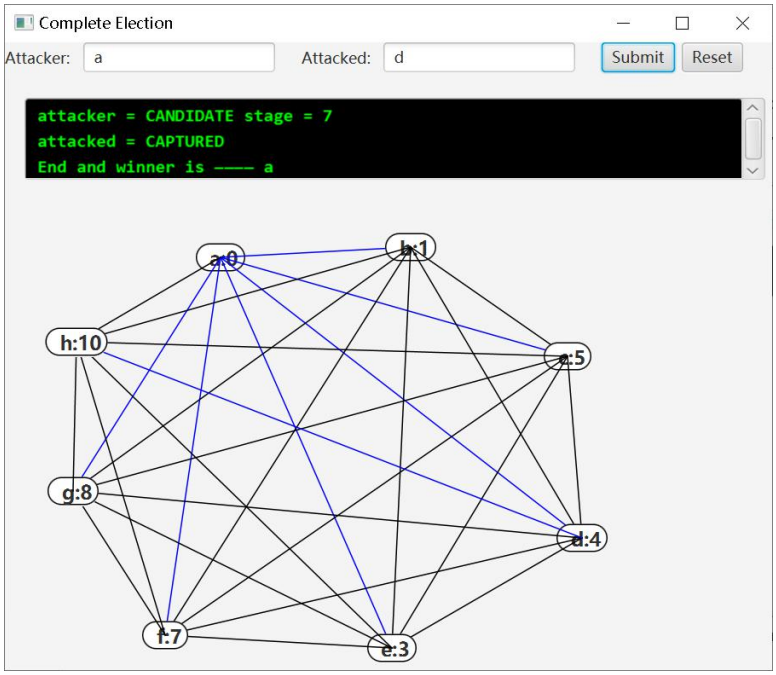

Figure 5 – the scene shows the situation when the algorithm ends

4 Algorithm display

When the attacker attacks a candidate, if the attacked node has a smaller stage (or equal but with larger Id), the attacked node accept and be captured. The attacker increases territory and its stage (+1) (like figure 6a). And if the attacker is smaller (or equal but with big Id), reject (like figure 6b). If the attacker is still a candidate, it becomes passive(like figure 6b).

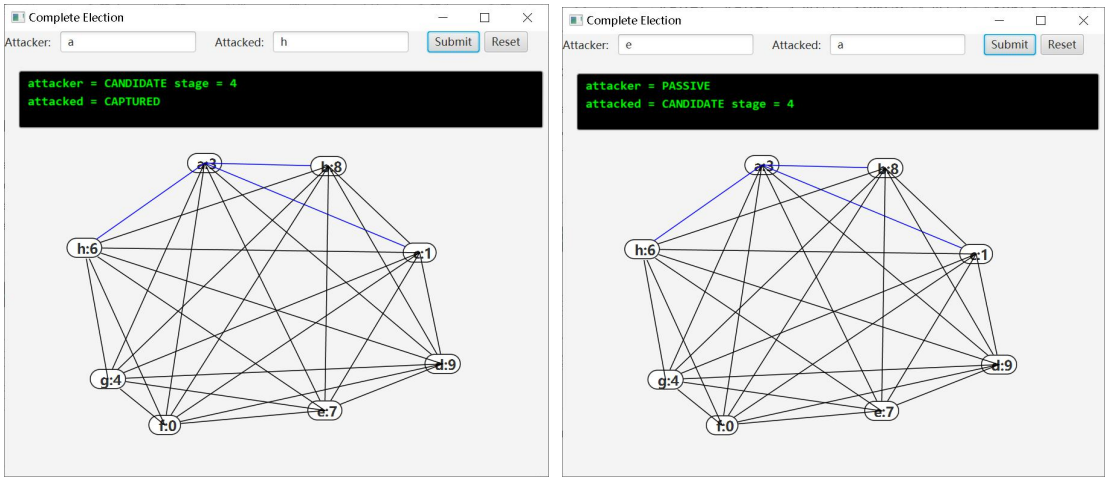

Figure 6 (a, b) – shows the situation when the attacker attacks a candidate (in figure 6a, attacker a attacks candidate h, in figure 6b, attacker e attacks candidate a)

When the attacker attacks a passive node, the attacker increases territory and stage (+1) (like figure 7a and figure 7b).

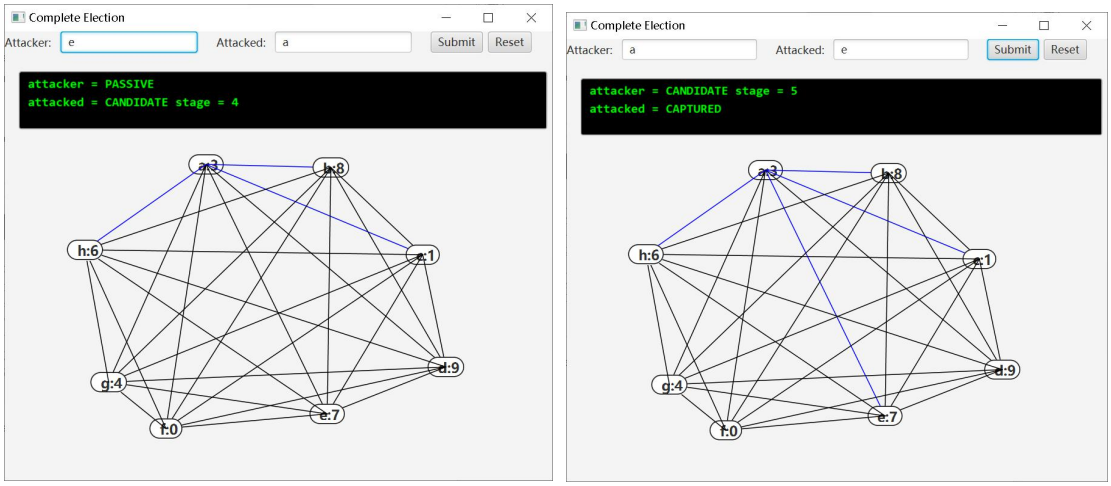

Figure 7 (a,b) – shows the situation when the attacker attacks a passive node (as shown, e is passive)

When the attacker attacks a captured node, the captured node will ask its owner. If captured node's owner is smaller (or equal with larger Id), accept and the captured node's owner becomes passive (as figure 8a and figure 8b).

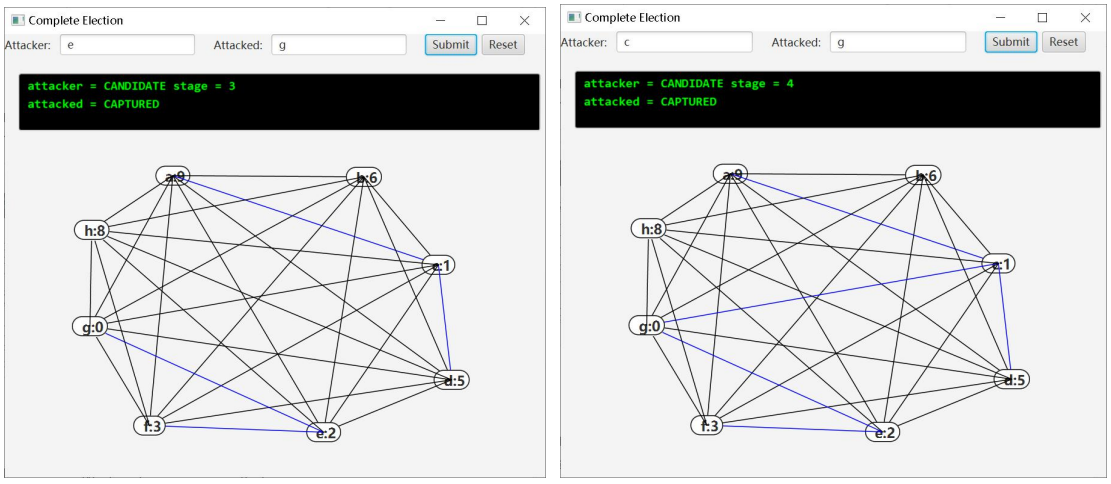

Figure 8 (a,b) – shows the situation when the attacker attacks a captured node (as shown, when c attacks g, e becomes passive)

When the attacker attacks a captured node, the captured node will ask its owner. If captured node's owner is bigger (or equal with smaller Id), reject and the attacker becomes passive (as figure 9a and figure 9b).

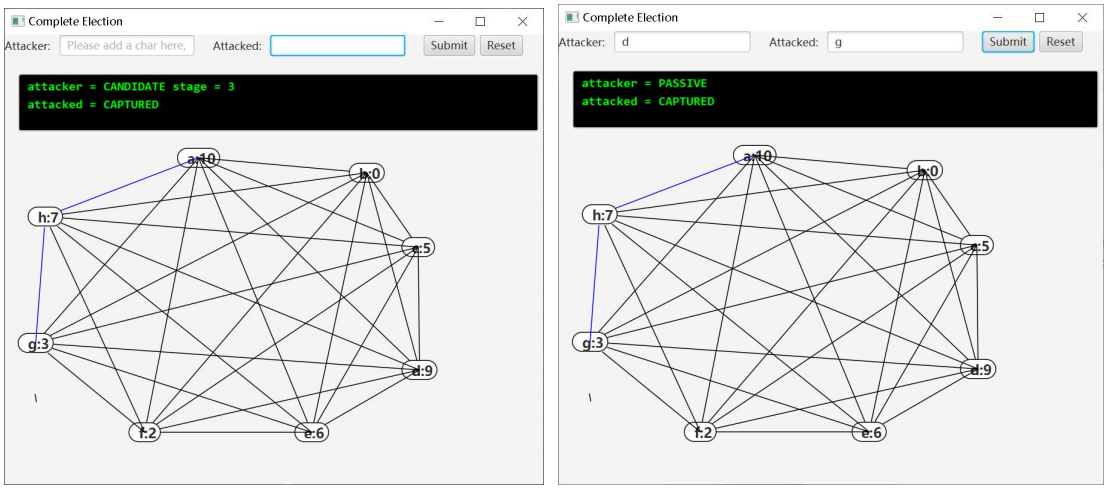

Figure 9 (a, b) – shows the situation when the attacker attacks a captured node (as shown, when d attacks g, d becomes passive)

5 The submit file structure and how to run the program

Figure 10 – shows the structure of submit file

The jar package is exported as "Complete_election.jar".

It runs when jdk version is 11 or higher.

Double click on run.bat, the program will run (as figure 11).

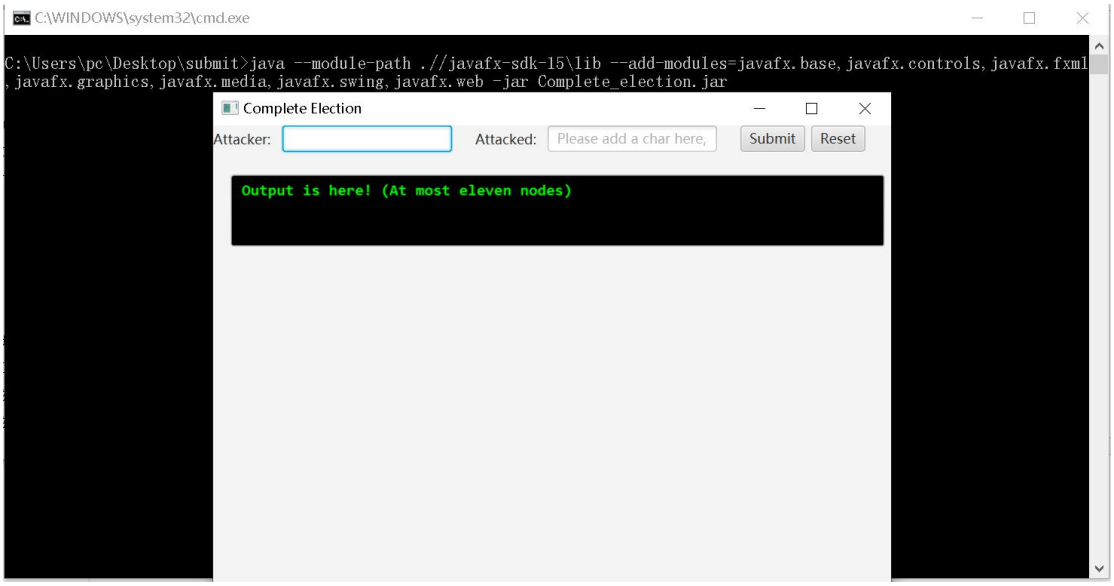

Figure 11 – the program is running when double click on run.bat

6 References

[1] CSI5308[W] Principles Distributed Compute - complete18. Retrieved from https://uottawa.brightspace.com/d2l/le/content/135166/viewContent/2509744/View