CSE403 Project 1
Leo Gao
Candice Miao

Logic Visualization IDE

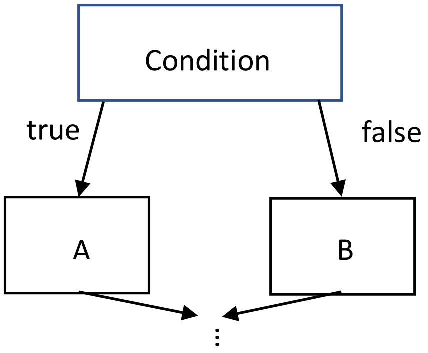
Motivation

Do you remember a time that you have been programming for the whole day and ended up getting a headache from looking at codes? Do you remember a time that you tried so hard to comprehend a piece of code with terrible style? Do you remember the time that you had trouble understanding a code in a new language that you are rusty with? It takes a lot of effort for some programmers such as myself to truly understand a piece of code in our brains. Based on some scientific research, people tend to understand graph better than text. If there exists a tool that can help programmers visualize the code logic, it will be a lot easier for them to understand.

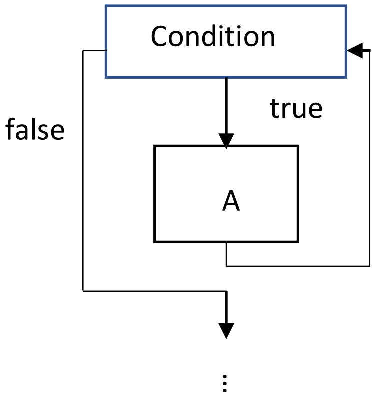
Approach

Our approach is to build this plugin in the IDE to generate a graph for a piece of code. Based on a graphic programming language called “Raptor”, we found a way to represent logic graph from Java code. For example:
Code:

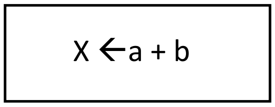
If (Condition) {A} else { B }



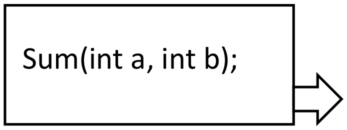
While (Condition) {A}



In the regular assignment code for example: x = a + b we can do:
X can be any types. It can be String, integer, float and array, and the syntax is based on the programming language.



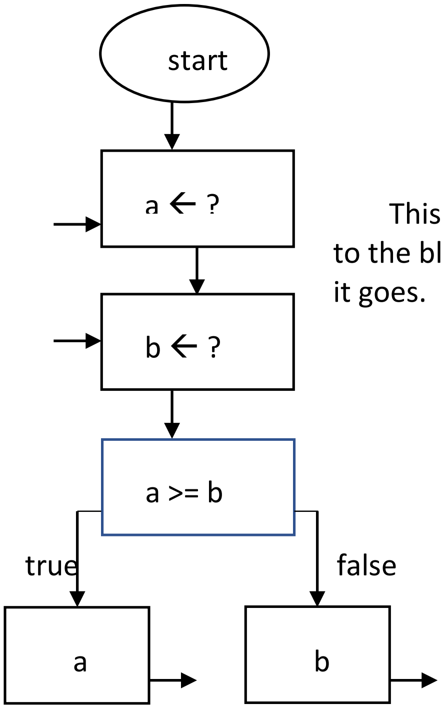
If there is a function call, we can use a special block. For example:
Sum(int a, int b);



Users can click on it and it will jump into another graph that shows the function sum(int a, int b). This way, a recursive call can also be represented in this way. This will also help users to understand recursive call better.

Here is an example of translating max function into graph:

```
int max(int a, int b) {
        If(a >= b) {
                return a;
        } else {
                return b;
        }
}
```



This function takes two parameters a and b. The little arrow next to the block means users can type in these two values and see where it goes.

The little arrow on the right side of the block means returning the value in the block.

On top of this logic graph generator with, we can do it vice versa—generating code with graph. Some programmers are more familiar with the graphic language; they can write their logic graph first and then the software will generate code for you. This graph generator will also help programmers to debug and test the code. In the debugging mode, programmers can keep track the code on the graph tree instead of reading the code line by line.

Possible challenge and risks:

First biggest problem is the different language have different syntax. Even though most of the programming language have if else and while loop logic case, some language has features that others do not have. For example, C and C++ have pointer type and Java python do not have; python allows nested array and Java does not.

One possible way to solve this problem is to have different graph system for different programming language. For example, in Java it is not allowed to have a String and an integer in the array but for python it will be allowed; but the main idea of loops, condition and function calls remains the same.

The second problem is how to represent data structure. For example, in Java, if user created a linked list or a hash map. How would the graph represent them? This is not too difficult to think but it will be a tedious work to create representation of all data structures in a programming language.