**Debugger for the source**

**Team Debugger:**

Do Andre Khoi Nguyen A0161346J

Jia Zhixin:

Nguyen Thi Viet Ha A0161296B

Programming report

1.      Specification: Create demo debugger for the source by using generator and changing the interpreter to ES6. The idea of using generator is adding the yield.

In the old interpreter, every time student want to debug, they have to print out the additional comments in order to understand what is the flow of the program, and understand what is the error, and what is the value for specific valuable

2.      Design: Using Ace editor

-

-

3.      The new feature of the source:

a.      Adding the breakpoint in specific line: The program will run until it reaches the breakpoint. In the margin line, the user can add the breakpoint here.

b.      Adding the look-up table: The user will type the variable name, and the interpreter will return its result.

4. The source code:

(the separate code will be included here)

Interpreter.js: has been modified from JS to ES6, and adding some missing stuffs for running correctly.

Debugger: debugger.js

stream.js:

object.js

misc.js:

Ace.html: demo page

5.      Implementation:

Changing from the old interpreter using ES6, which supports the generators. What we have done is changing some syntax errors and missing some definition of the source to the ES6 parser.

The program will run line by line until it reaches the first breakpoint. The debugger will create the function run which parse the input\_text to the function parse\_and\_evaluate of interpreter.

A generator is a special type of function that works as a factory for iterators. A function becomes a generator if it contains one or more yield expressions and if it uses the function\* syntax.

The Idea of generator functions: The yield is added in generator function which is evaluate\_sequence, evaluate\_while\_statement, evaluate\_for\_statement.

(needed to add more here)

The idea of the look-up variable: Generally, the environment is the list of frames. Each time we define a new variable, it will be added to the look-up variable.

6.      Testing:

This testing program and the result will be included in the test file.

Function:

Recursion:

Stream:

OOP:

Breakpoint:

7.      Conclusion:

8.      Appendix:

Generator: [https://developer.mozilla.org/enUS/docs/Web/JavaScript/Reference/Statements/function\*](https://developer.mozilla.org/enUS/docs/Web/JavaScript/Reference/Statements/function*)

Ace editor:

<https://ace.c9.io/>