Indian Institute of Technology Guwahati

CS243 : Software Engeneering Lab

Under the guidance of : Prof. Samit Bhattacharya

Project 1 : An app to detect student activity and alert generation for the instructor

White Box Testing Report

Group 7 :

- Mukul Verma , 150101038
- Piyush Jain , 150101046
- Shubhanshu Verma , 150101073

Date : May 6, 2017

Contents

1. Introduction

1.1 Purpose

This white box testing is intended to check the internal structure (modules) of the system at the most possible lower level.

1.2 References

For references, Software Requirement Specification and Design Document of the project have been referred.

1.3 Module Checked :

The module for detecting flaws in orientation handling by the user has been taken under white box testing , since it is the most complex module in our system.

2. Testing Process

2.1 Process Description

Following components have been covered in this white box testing :

- **Statement Coverage** : It is ensured that each and every statement has been executed at least once.
- **Condition and Branch Coverage** : For every condition, it was checked for it to be both true or false and corresponding branching has been checked.
- **Path Coverage** : All linearly independent paths were found using Control Flow Graph and it was ensured that each of them is executed at least once.

2.2 Functions in the module

Function in the module :

- *way_one_orientation()*
- *flaw_checker()*
- *condition_implementer()*
- *way_zero_orientation()*
- *way_decider()*

2.3 Test Cases for each Function

Following are the functions which have been tested explicitly.
Note : 'A' stands for Arbitrary parameter

- *flaw_checker (pitc, rol, s_time, fla, def, s_count, m_count)*
  Test Cases :
  - (5,45,A,A,A,A,A)
  - (5,75,A,A,A,A,A)
  - (15,45,A,A,A,A,A)
  - (15,75,A,A,A,A,A)
  - Boundary Case : (12,30,A,A,A,A,A)
  - Boundary Case : (12,60,A,A,A,A,A)
- *condition_implementer(s_time, fla, dif, s_count, m_count)*
  Test Cases :
  - (A,0,7,A,A) (A,1,7,A,A) (A,2,7,A,A) , Boundary: (A,0,5,A,A)
  - (A,1,12,A,A) (A,0,12,A,A) (A,2,12,A,A) , Boundary: (A,1,10,A,A)
  - (A,2,20,A,A) (A,0,20,A,A) (A,1,20,A,A) , Boundary: (A,2,15,A,A)
  - (A,0,30,A,A) (A,1,30,A,A) (A,2,30,A,A)
  - (A,0,7,A,A) (A,0,7,A,A) (A,0,7,A,A)
- *way_zero_orientation(acc_x, acc_y, acc_z, pitch, roll)*
  Test Cases :
  - (A,A,A,A,A)

2.4 Testing Inferences

2.4.1 Statement Coverage

- All the statements were executed at least once.
- No runtime error or useless line found.

2.4.2 Condition and Branch Coverage

- There were six conditional statements. Testing was done to check the true and false for all six of them.
- The statement checking the conditions over flaw count and time didn't have an else condition.
- Rest all condition statements did proper branching for both true and false conditions.

2.4.3 Path Coverage

- All independent paths have been identified using Control Flow Graphs.
- Following are the control flow graphs (numbers in the circle represent line number) :
  - *flaw_checker*

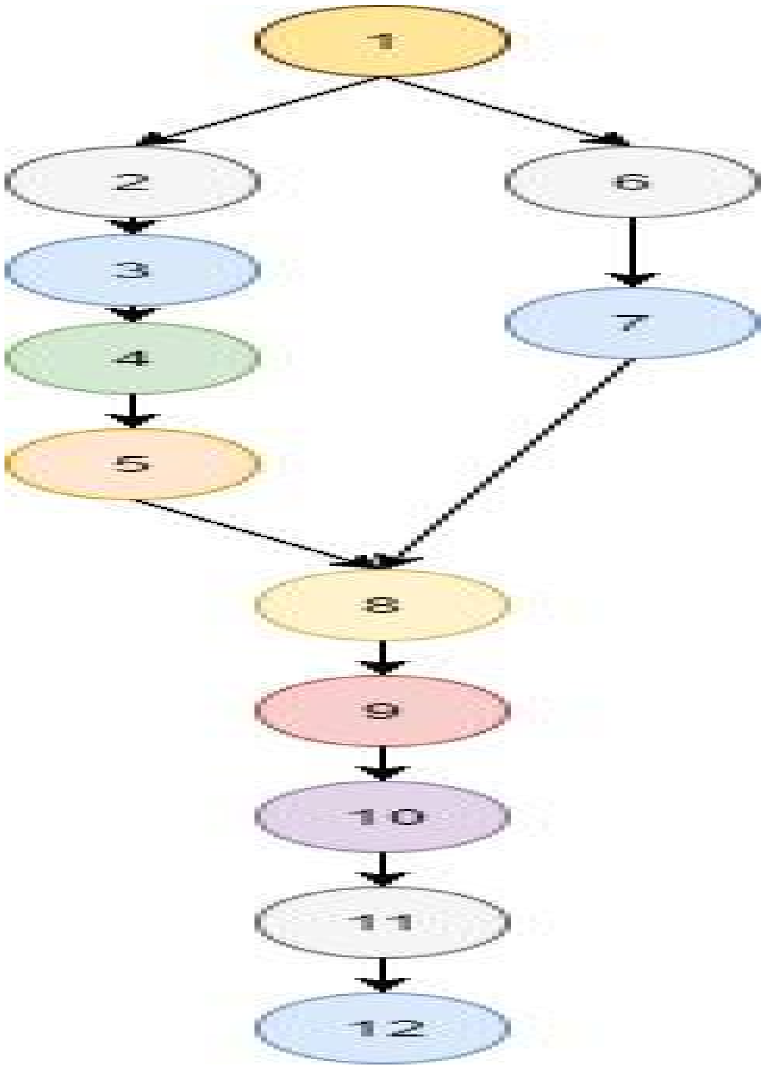

Figure 1: CFG for flaw_checker

- *condition_implementer*

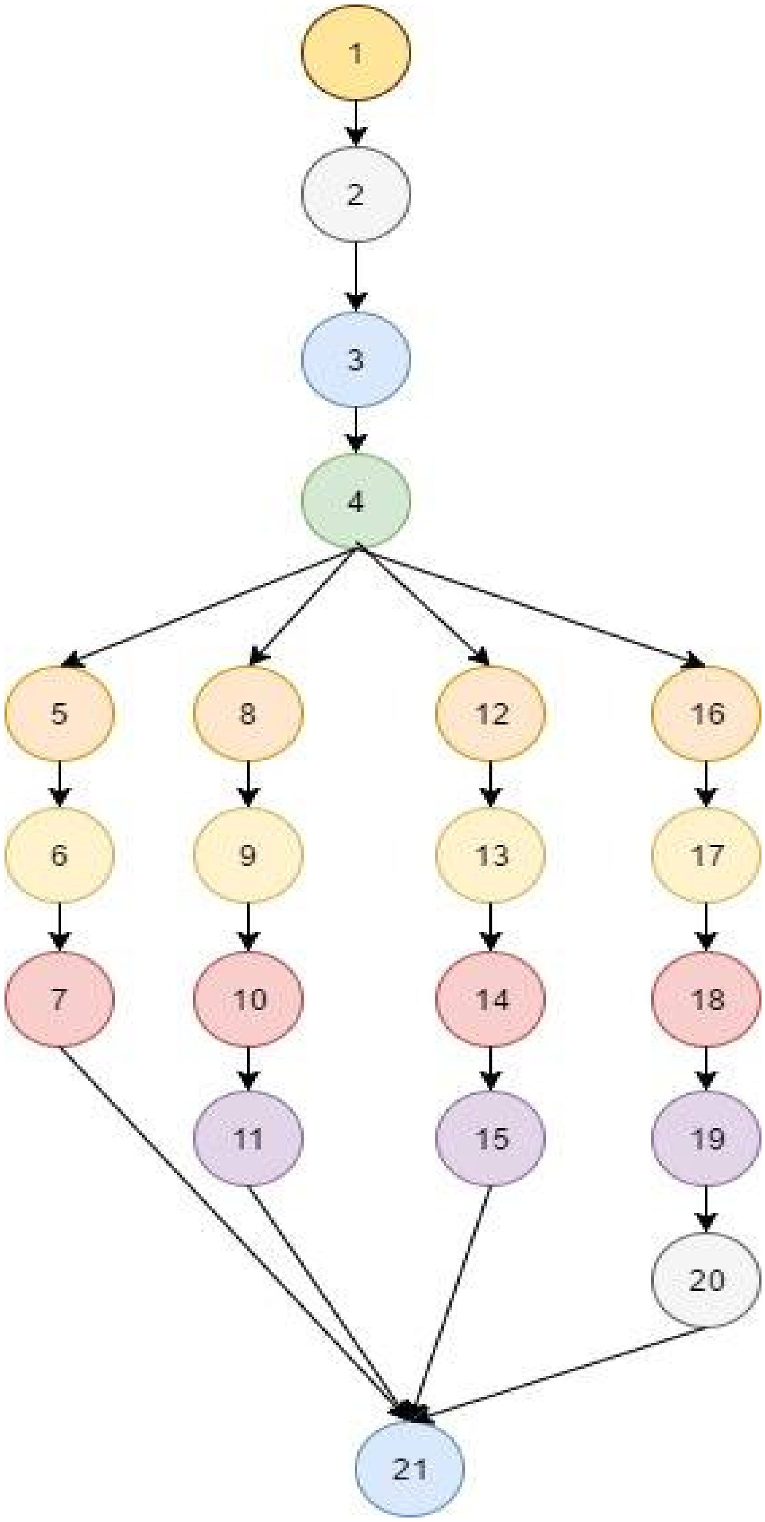

Figure 2: CFG for condition_implementer

- *way_decider*

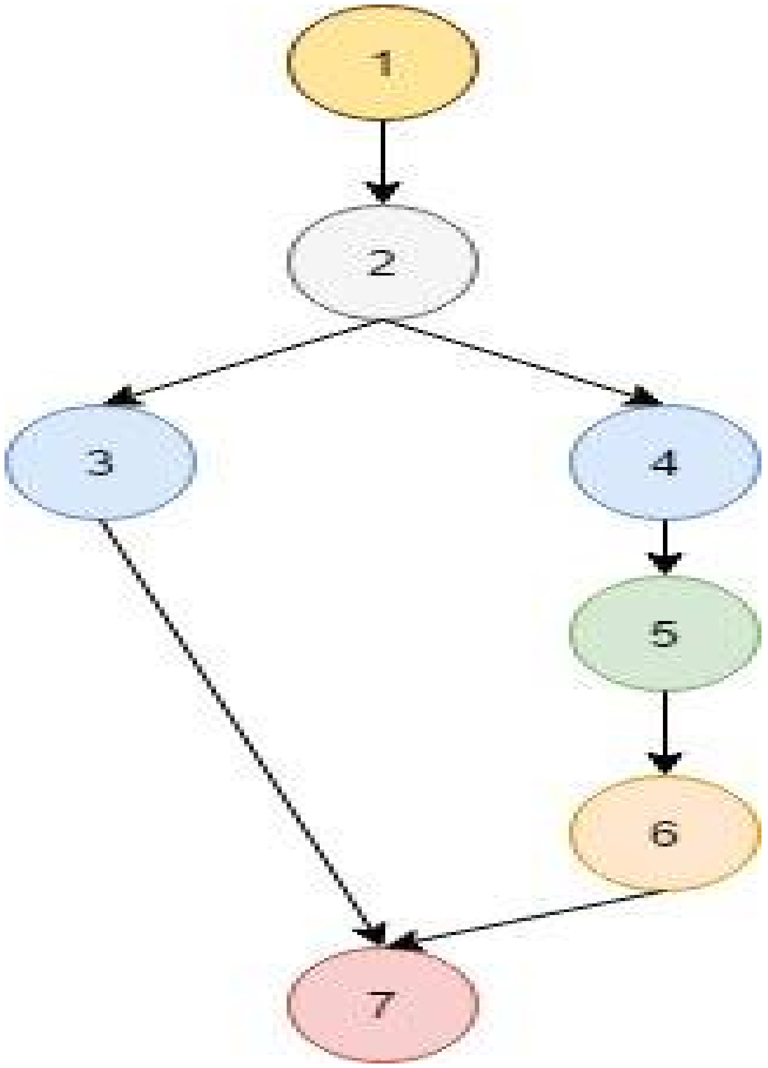

Figure 3: CFG for way_decider

- As it can be seen, there are 2 , 4 and 2 independent paths respectively in these three functions. Each of these paths have been tested.
- Other than these, rest of the functions have a straigt control sequence, constituting single independent path.
- All of these paths are being followed correctly. No error reported here.

2.5 Conclusion

- There is a need to correct branch logic of module *condition_implementer* because missing of an else statement was reported there.
- The module passed the rest all tests.