

Software Testing 2022

S1833951 li fang

January 26 2022

4.1 Identifying gaps and omission in the testing process

The security issue has not been taking in consider during the test, one reason is that the tested part of the system is not the part interacting with data from website directly, so the security feature has been lower priority when structuring this part of the code and test.

Unit test:

And as we know, unit testing only tests individual units or components of an application and not the entire system. Therefore, it may not reveal integration or system-level issues, such as the route does not visit all the point or being too long. Also, unit tests are typically written and executed in a controlled environment, which may not accurately reflect real-world scenarios. In my case, I could only generate limited amount of data for testing, and that may not cover all scenarios for the actual data. For example: I have generated set of paths I choose manually from different points, the test is pass but the test data is certainly too small comparing to actual data, as the route contain above 100 paths.

Unit tests are not designed to measure the performance of an application and may not reveal performance-related issues, the function could be improved on its performance, but the test would not be helpful in judging that.

System test:

System testing can be difficult to reproduce bugs, which can make it challenging to identify and fix issues. Since the test is for two functions together, even the test is failed (such as choosing a point that is not nearest) it can still take me a lot of time wo figured out which function contains bug, or which exact line contains bug.

4.2 Identifying target coverage/performance levels for the different testing procedures

R1:

As planned, the unit test should be able to identify all bug for the single function choose, and any changes to the code that result it perform not as expected should be detected by the unit test. So wrong input, wrong output should all let to unit test failure. As I made unit test separately for each side of the rectangular area, the failed unit test should be capable telling me which part of the function foes wrong too, easier for fixing.

R2:

The system test is expected to failed whenever the function does not perform as expected. It may not tell me directly which part of the code or which function is wrong, but it should be

failed whenever the function does not meet its purposes. I generate the point that it measures nearest point consistently while I am adding point to the list, so the nearest point is changing during the test. This stimulates the actual process of the selection of path.

The unit test for dis is expected to fail whenever not meet the requirement, and it should tell me which part of the code is wrong too.

4.3 Discussing how the testing carried out compares with the target levels

R1:

The unit test does failed when I try to change the variable name to the wrong one, or entering wrong number in the output (0.0 for example), or does not perform as expected (when crossing restricted area and still return false in the function). So, the test is cover all the false condition, if the function is false, then the test is failed all the time. But I can only tell which part of the function is wrong when the initial value of the restricted area is wrong.

For example, if I entering the coordinate of meadows wrongly in the noflyzone class, the unit test will be able to inform me which part of the intersections is wrong, so I may know the initial value of a specific path is wrong then find which point goes wrong.

But If I had changed the variable name or mix it up, the test will fail, but I cannot know from the test result that which part of the function does failed. If I exchange the x coordinate of starting point and end point, test will fail, but I still need to look at the whole function to find the bug.

R2:

The system test does fail when I try to change any variable to a wrong one in the function, and failed when the function is performed wrongly.

One problem would be the manually selected point is less than the point actual path would contain, so if I am using the wrong initial value for the for loop, and the initial value is bigger than the point I generated for test, test does go wrong but always indicate the same line in the test. That will not be helpful to fix the bug in the function, even that was not the expectation for the system test at the beginning.

For system test, it fails every time it does supposed to fail, so it does meet requirement.

For the unit test part, the test does fail whenever the function does wrong, but it can hardly be tell why is it wrong or which part goes wrong. This might because the test does not exam the input value, so it could not check if the function does mix up the variable, it can only check if the decimal place is wrong or the overall performance is not working.

4.4. Discussion of what would be necessary to achieve the target levels.

R1: To achieve the targeted levels, I may have to change the original code, as what I have right now is making it too hard for unit test to indicate which part goes wrong. I should change the core method in the isCrossing function “LinesIntersect”, which is a highly compact method that does everything all together. If I am using this method, I can hardly test anything within the

method. I may separate the method so I can test the logic between them, that will be helpful for me to design a unit test to indicate which part of the code goes wrong.

R2: I am happy with what I have now, still, if I want it to be more reliable, I have to generate more point and arrange this point carefully in order, so they do cover as much of the real scenario as possible. In the test I stimulate the scenario that nearest point is changing consistently, I may have to add more scenario like a duplicate point is included or other failed input, to make the test available for security check and input check.

For Unit test design for dis function alone, I could hardly improve it as test does not see the input value of the function, so it cannot tell if it fails because the variable is wrong or the calculating part is wrong.