GAME BASED LEARNING OF SORTING ALGORITHMS

Software Engineering Lab (CS243)

January – April (2017)

WHITE BOX TESTING DOCUMENT

Instructor

Dr. Samit Bhattacharya

CSE Department, IITG

Group Number: 02

Project Team Members

150101003: Abhishek Kumar

150101045: Patoliya Meetkumar Krushnadas

150101054: Saket Sanjay Agrawal

Index

Whitebox Testing of SwapCubesBubbleSort Module (Update function)

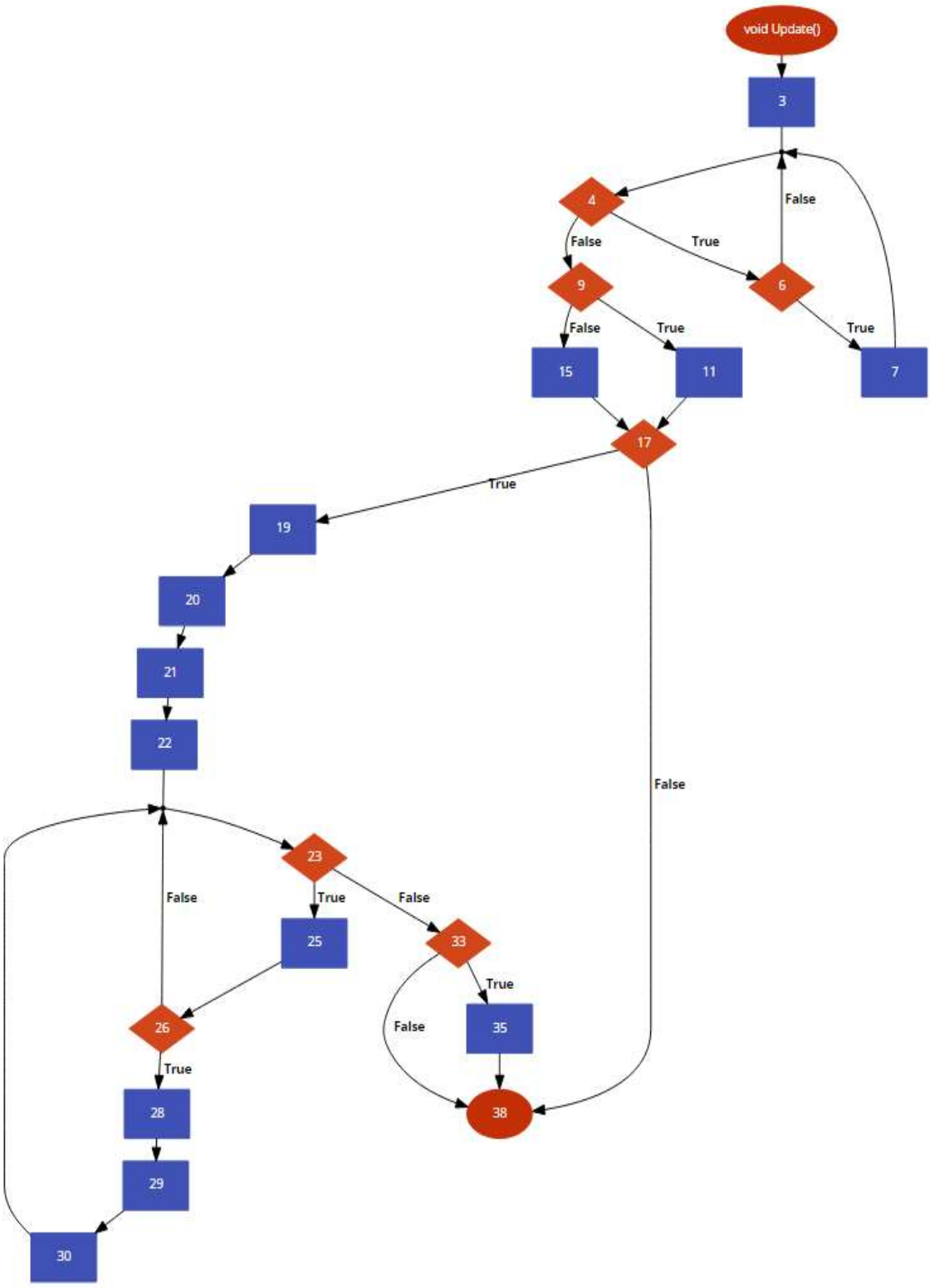
Update function of SwapCubesBubbleSort.cs Module:

Swapping is the main functionality of any sort. **It is most critical part of any sort**. This function will be called and executed in every time frame of Bubble Sort scene. This function will decide whether any of cubes are moving (being swapped) or not. If swapping is not under process, then it keeps updating indices of cubes being touched at each time frame. When two cubes will be touched, then it will trigger the swapping of two cubes touched by user.

1) Code

```
void Update()
{
    int count = 0;
    foreach (GameObject cube in cubeList)
    {
        if (cube.layer == LayerMask.NameToLayer("TouchInput"))
        { count++; }
    }
    if (count == 8)
    {
        transferringCubes = false;
    }
    else
    {
        transferringCubes = true;
    }
    if (Input.touchCount == 2 && transferringCubes == false)
    {
        touchOld = new GameObject[touchList.Count];
        touchList.CopyTo(touchOld);
        touchList.Clear();
        index.Clear();
        foreach (Touch touch in Input.touches)
        {
            Ray ray = GetComponent<Camera>().ScreenPointToRay(touch.position);
            if (Physics.Raycast(ray, out hit, Mathf.Infinity, TouchInput))
            {
                GameObject reciepient = hit.transform.gameObject;
                touchList.Add(reciepient);
                index.Add(Convert.ToInt32(reciepient.name));
            }
        }
        if (Input.touches[0].phase == TouchPhase.Ended || Input.touches[1].phase == TouchPhase.Ended)
        {
            StartCoroutine(Swap(10, touchList[0], touchList[1]));
        }
    }
}
```

2) Control Flow Graph



3) Paths

a) 3 4 6 9 11 17 38
b) 3 4 6 9 11 17 19 20 21 22 23 25 26 33 38
c) 3 4 6 9 11 17 19 20 21 22 23 25 26 28 29 30 33 38
d) 3 4 6 9 11 17 19 20 21 22 23 25 26 33 35 38
e) 3 4 6 9 11 17 19 20 21 22 23 25 26 28 29 30 33 35 38
f) 3 4 6 7 9 11 17 38
g) 3 4 6 7 9 11 17 19 20 21 22 23 25 26 33 38
h) 3 4 6 7 9 11 17 19 20 21 22 23 25 26 28 29 30 33 38
i) 3 4 6 7 9 11 17 19 20 21 22 23 25 26 33 35 38
j) 3 4 6 7 9 11 17 19 20 21 22 23 25 26 28 29 30 33 35 38
k) 3 4 6 9 15 17 38
l) 3 4 6 9 15 17 19 20 21 22 23 25 26 33 38
m) 3 4 6 9 15 17 19 20 21 22 23 25 26 28 29 30 33 38
n) 3 4 6 9 15 17 19 20 21 22 23 25 26 33 35 38
o) 3 4 6 9 15 17 19 20 21 22 23 25 26 28 29 30 33 35 38
p) 3 4 6 7 9 15 17 38
q) 3 4 6 7 9 15 17 19 20 21 22 23 25 26 33 38
r) 3 4 6 7 9 15 17 19 20 21 22 23 25 26 28 29 30 33 38
s) 3 4 6 7 9 15 17 19 20 21 22 23 25 26 33 35 38
t) 3 4 6 7 9 15 17 19 20 21 22 23 25 26 28 29 30 33 35 38

Here are some paths in which “line 7” is not executed but “line 11” is executed. But it is not possible for this case to appear as if ”line 7” will not happen “count” will not reach 8 and “line 11” will not be executed. Index “a” to “e” are the paths in which this case is occurring. These paths will not be followed so these paths are redundant.

There are also cases where “line 7” is executed but “line 11” is not executed. These cases are also redundant as if “line 7” gets executed “count” will be set 8 and “line 11” will always be executed. So these paths will never be followed. Index “p” to “t” are such paths.

4) Test Cases

For the function calls we will be using stubs to provide the output. For the paths “a” to “t” the following are the test cases.

a) No test case possible.
b) No test case possible.
c) No test case possible.
d) No test case possible.
e) No test case possible.

f) cube.layer = 0
LayerMask.NameToLayer(“TouchInput”) = 1
Input.touchCount=1

g) cube.layer = 0
LayerMask.NameToLayer(“TouchInput”) = 1
Input.touchCount=2
Transferring.Cubes = false
Input.touches[0].phase = 0
Input.touches[1].phase = 0
TouchPhase.Ended=1
GameObject recipient = 0
hit.transform.gameObject = 1

h) cube.layer = 0
LayerMask.NameToLayer(“TouchInput”) = 1
Input.touchCount=2
Transferring.Cubes = false
Physics.Raycast(ray, out hit, Mathf.Infinity, TouchInput) = 0
Input.touches[0].phase = 0
Input.touches[1].phase = 0
TouchPhase.Ended=1
GameObject recipient = 0
hit.transform.gameObject = 1

i) cube.layer = 0
LayerMask.NameToLayer(“TouchInput”) = 1
Input.touchCount=2
Transferring.Cubes = false
Input.touches[0].phase = 1
Input.touches[1].phase = 1
TouchPhase.Ended=1
GameObject recipient = 0
hit.transform.gameObject = 1

j) cube.layer = 0
LayerMask.NameToLayer(“TouchInput”) = 1
Input.touchCount=2
Transferring.Cubes = false
Physics.Raycast(ray, out hit, Mathf.Infinity, TouchInput) = 0
Input.touches[0].phase = 1
Input.touches[1].phase = 1
TouchPhase.Ended=1
GameObject recipient = 0
hit.transform.gameObject = 1

k) cube.layer = 1
LayerMask.NameToLayer(“TouchInput”) = 1
Input.touchCount=1
GameObject recipient = 0
hit.transform.gameObject = 1

l) cube.layer = 1
LayerMask.NameToLayer(“TouchInput”) = 1
Input.touchCount=2
Transferring.Cubes = false
Input.touches[0].phase = 0
Input.touches[1].phase = 0
TouchPhase.Ended=1
GameObject recipient = 0
hit.transform.gameObject = 1

m) cube.layer = 1
LayerMask.NameToLayer(“TouchInput”) = 1
Input.touchCount=2
Transferring.Cubes = false
Physics.Raycast(ray, out hit, Mathf.Infinity, TouchInput) = 0
Input.touches[0].phase = 0
Input.touches[1].phase = 0
TouchPhase.Ended=1
GameObject recipient = 0
hit.transform.gameObject = 1

n) cube.layer = 1
LayerMask.NameToLayer(“TouchInput”) = 1
Input.touchCount=2
Transferring.Cubes = false
Input.touches[0].phase = 1
Input.touches[1].phase = 1
TouchPhase.Ended=1
GameObject recipient = 0
hit.transform.gameObject = 1

o) cube.layer = 1
LayerMask.NameToLayer(“TouchInput”) = 1
Input.touchCount=2
Transferring.Cubes = false
Physics.Raycast(ray, out hit, Mathf.Infinity, TouchInput) = 0
Input.touches[0].phase = 1
Input.touches[1].phase = 1

TouchPhase.Ended=1
GameObject recipient = 0
hit.transform.gameObject = 1

p) No test case possible.
q) No test case possible.
r) No test case possible.
s) No test case possible.
t) No test case possible.

5) Output

a) Not possible
b) Not possible
c) Not possible
d) Not possible
e) Not possible

f) count = 8
transferringCubes = false
GameObject recipient = 0
hit.transform.gameObject = 1

g) count = 8
transferringCubes = false
GameObject recipient = 0
hit.transform.gameObject = 1

h) count = 8
transferringCubes = false
GameObject recipient = 1
hit.transform.gameObject = 1

i) count = 8
transferringCubes = false
GameObject recipient = 0
hit.transform.gameObject = 1

j) count = 8
transferringCubes = false
GameObject recipient = 1
hit.transform.gameObject = 1

k) count = 0
transferringCubes = true
GameObject recipient = 0
hit.transform.gameObject = 1

l) count = 0
transferringCubes = true
GameObject recipient = 0
hit.transform.gameObject = 1

m) count = 0
transferringCubes = true
GameObject recipient = 1
hit.transform.gameObject = 1

n) count = 8
transferringCubes = true
GameObject recipient = 0
hit.transform.gameObject = 1

o) count = 8
transferringCubes = true
GameObject recipient = 1
hit.transform.gameObject = 1

p) Not possible
q) Not possible
r) Not possible
s) Not possible
t) Not possible