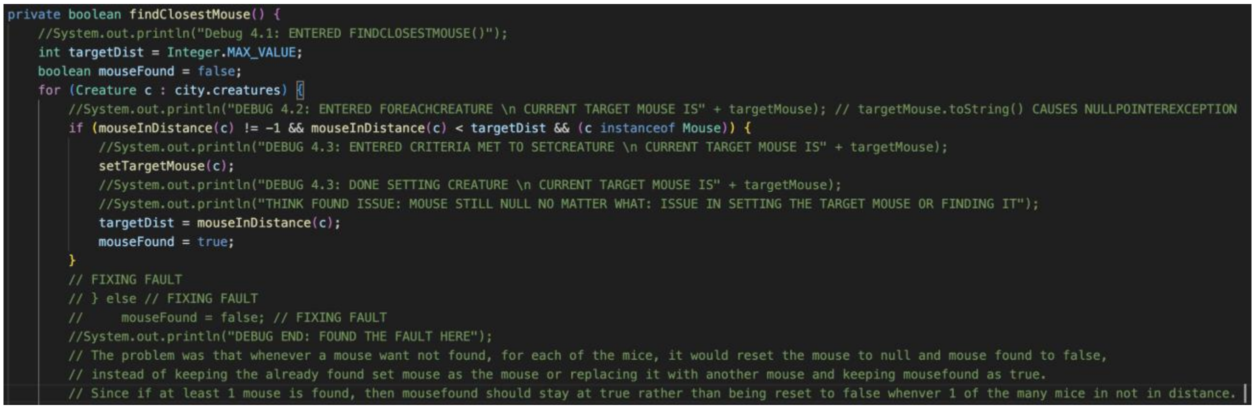
J10_Debugging:

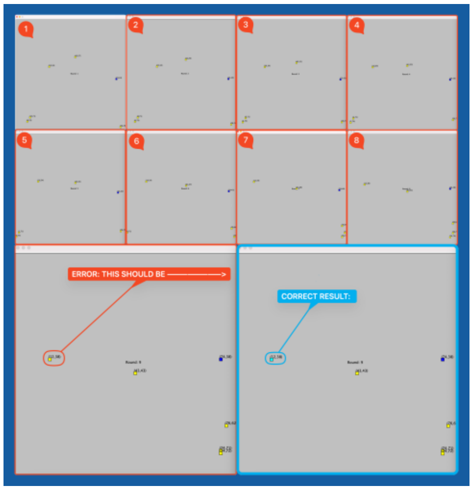
Based on the result from the unit test, the expected value was “76 62 y43 43 y12 38 [c74 72 ydone....”, but the results differed and were “but was:<...76 62 y43 43 y12 38 [y74 72 ydone” starting at the bracket. Since the differing values started at the cat value (“c74” was “y74” instead), the error must have been somewhere in the cat’s code, whether it be its plotting, movement, or initialization object. The difference in letters shows that the cat at that point should now be cyan, meaning having noticed and now chasing a mouse, but it is still yellow and not chasing. So, the issue must be in finding or setting a target mouse.



```java
private boolean findClosestMouse() {
    //System.out.println("Debug 4.1: ENTERED FINDCLOSESTMOUSE()");
    int targetDist = Integer.MAX_VALUE;
    boolean mouseFound = false;
    for (Creature c : city.creatures) {
        //System.out.println("DEBUG 4.2: ENTERED FOREACHCREATURE \n CURRENT TARGET MOUSE IS" + targetMouse); // targetMouse.toString() CAUSES NULLPOINTEREXCEPTION
        if (mouseInDistance(c) != -1 && mouseInDistance(c) < targetDist && (c instanceof Mouse)) {
            //System.out.println("DEBUG 4.3: ENTERED CRITERIA MET TO SETCREATURE \n CURRENT TARGET MOUSE IS" + targetMouse);
            setTargetMouse(c);
            //System.out.println("DEBUG 4.3: DONE SETTING CREATURE \n CURRENT TARGET MOUSE IS" + targetMouse);
            //System.out.println("THINK FOUND ISSUE: MOUSE STILL NULL NO MATTER WHAT: ISSUE IN SETTING THE TARGET MOUSE OR FINDING IT");
            targetDist = mouseInDistance(c);
            mouseFound = true;
        }
        // FIXING FAULT
        // } else // FIXING FAULT
        //     mouseFound = false; // FIXING FAULT
        //System.out.println("DEBUG END: FOUND THE FAULT HERE");
        // The problem was that whenever a mouse want not found, for each of the mice, it would reset the mouse to null and mouse found to false,
        // instead of keeping the already found set mouse as the mouse or replacing it with another mouse and keeping mousefound as true.
        // Since if at least 1 mouse is found, then mousefound should stay at true rather than being reset to false whenver 1 of the many mice in not in distance.
```

The bug was found in the findClosestMouse() method. It was fixed by removing “else { mouseFound = false}” statement. The problem was that whenever a mouse was not found, for each of the mice, it would reset the mouse to null and mouse found to false, instead of keeping the already found set mouse as the mouse or replacing it with another mouse and keeping mouseFound as true. Since if at least 1 mouse is found, then mouseFound should stay at true rather than being reset to false whenever 1 of the many mice is not in distance.

Diagram of where cat should be:


1
2
3
4
5
6
7
8
ERROR: THIS SHOULD BE
CORRECT RESULT: