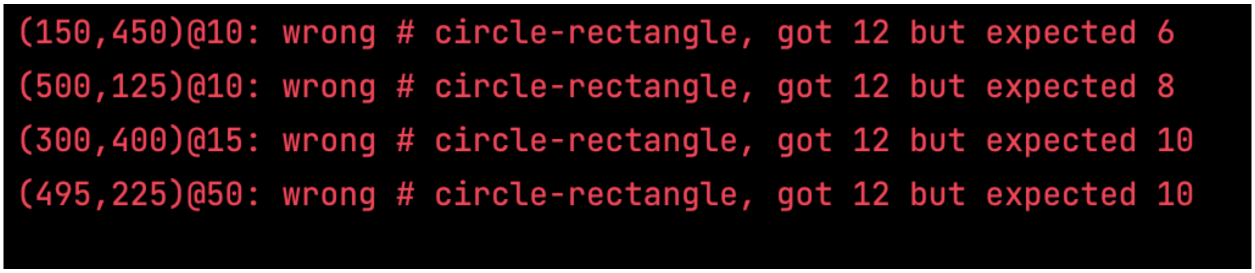
Helen Cui
4/18/24
COSC PS-2
AQL

The first testing that I did was for the individual PointQuadtree methods, I inserted a couple points in main and I tested to see the size of the PointQuadtree as I added more points to it. By checking to see if the number of points were the expected number of points in size, I was able to see if my functions were moving correctly against each other even if I'm not visualizing the tree itself.

The second testing that I did was for DotTreeGUI, by far this was the most rigorous testing I did. I first ran both test0 and test1 before designing my own tests, when both of those tests failed I designed test2 and test3 in order to narrow down what my issues were. My test2 was designed first and essentially copied test1, as I had the same error for every line:



```
(150,450)@10: wrong # circle-rectangle, got 12 but expected 6
(500,125)@10: wrong # circle-rectangle, got 12 but expected 8
(300,400)@15: wrong # circle-rectangle, got 12 but expected 10
(495,225)@50: wrong # circle-rectangle, got 12 but expected 10
```

I attempted to isolate the error, so I also put print statements into my Geometry class to check which specific points were getting checked and if my checks worked. I also added print statements into my findincircle helper in order to understand if my findinCircle return list was working correctly, it was not. I figured out from there that the order of my if statements were incorrect and that I needed to save/return a list in my findinCircle. Additionally, I used the mouse and select 'q' mode in order to make it larger to see if I could 'find' two points at once. That also showed me that my findinCircle wasn't returning a cumulative list properly. From there, my test0 still didn't work so I went through my test2 and slightly adjusted the individual points to see what the issues were, from there I came to the realization that I had accidentally altered my original test0 case.

In order to test my collisions, I first tested just two points and saw how long it took for both of those points to touch. Then, I tested putting points one after the other in the same spot, given their immediate close proximity to each other they should have immediately turned red to show that they were within the same radius. After, I crowded the screen in order to see if naturally when the moving points interacted they also showed that they collided. While still in this case I pressed 'd' to also see if the deletion function worked properly.