Ben Williams
Alex Craig
January 30th, 2022
COSC-10

PointQuadTree

Alex and I really had no issues when writing this code. Once we understood each of the variables, writing the recursive functions was pretty simple. It wasn't until writing DotTreeGUI that we were able to test the tree.

DotTreeGUI

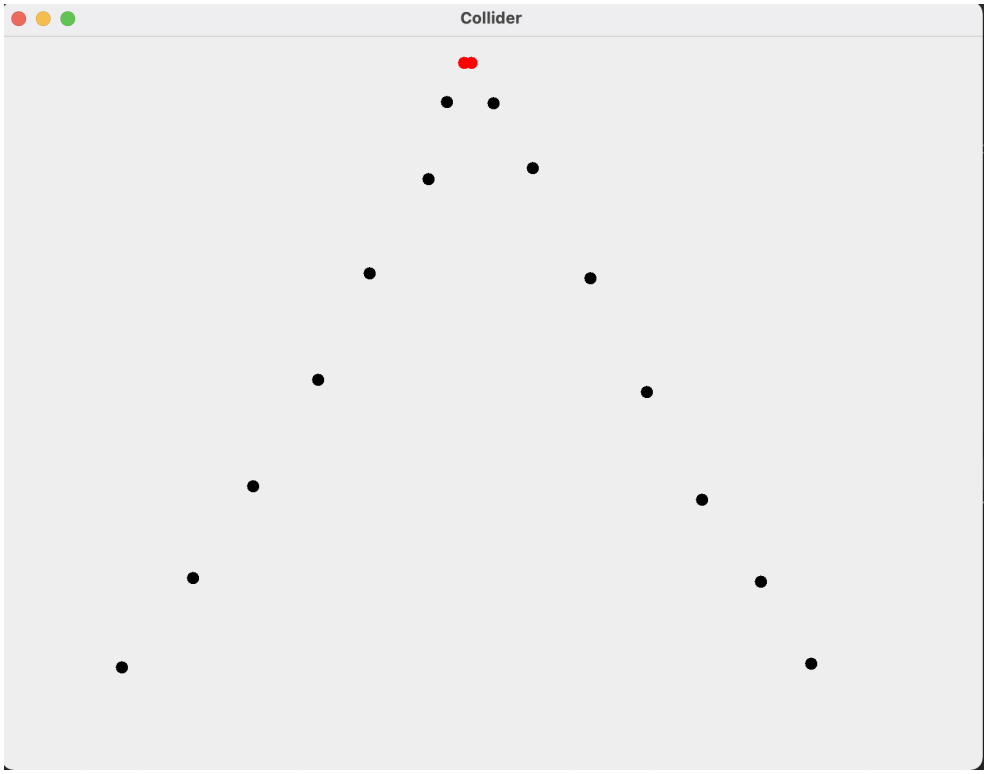
The most challenging part of DotTreeGUI was figuring out the drawTree() method. We had issues with getting the dots and the lines to draw at exactly where the mouse was clicked, but eventually we figured it out. The recursion was not the issue here, but the location of the drawing was. Next was to do the actual tests given to us, and to write our own.

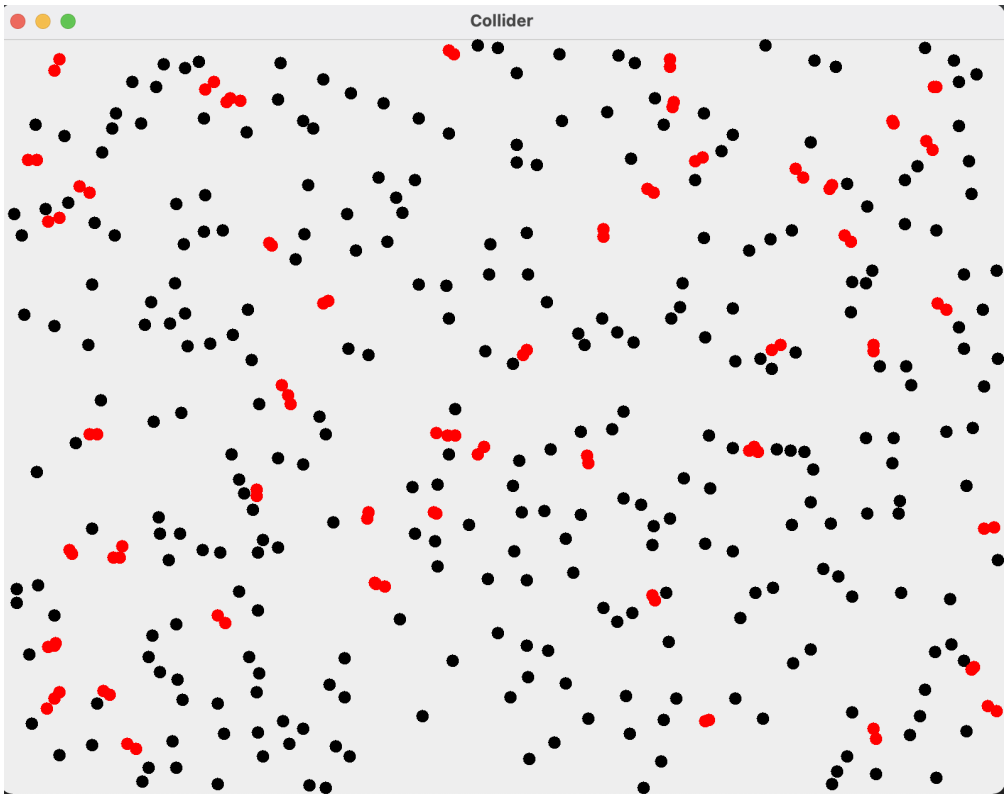
Test0 passed the first try, so we did not learn too much with that. Same with test1. But we were having issues with our 'q' mode not working properly, so we wrote test2 to check with some of the radii of the tests differing. Test 2 also passed, and it ended up that we had mistakenly just written *dotRadius* instead of *mouseRadius* where the findIncircle() was looking.

CollisionGUI

Originally, the main issue we had was a blob colliding with itself and potentially adding itself to *collided* more than once. We fixed this with an additional boolean and an if statement within our for loops. We also had an issue with setting *collided* to null and getting NullPointerExceptions, but this was fixed by re-defining *collided* as an empty ArrayList rather than as null. Our 'd' destroy method was not working originally either, as it was also getting NullPointerExceptions, but the same fix as mentioned above also worked here.

We mainly just tested by eye, slowing down the time if necessary to watch individual blobs. These are two examples of tests that we did. The first is pretty clear, making sure no unwanted points collide, while the second lets us see it work at scale:


Collider


Collider