

To test the PointQuadTree and DrawingGUI classes, I designed the 'test2' method to test the functionality and correctness, specifically its ability to efficiently locate points in close proximity. The method begins by initializing a quadtree with a central point (400, 300), which acts as the root node. Three additional points are then inserted into the tree. Following the setup, the method carries out three distinct tests using the testFind method. Each testFind call inspects a specific point and its surroundings by a given radius to determine how many points should be found within that vicinity. The results from testFind are compared to the expected outcomes to determine the accuracy of the point location mechanism of the PointQuadtree. The cumulative sum of discrepancies (bad) across the tests is calculated, and if there are no discrepancies, it signifies that the tests have passed. I used this to test the class, and I received no errors when running the method.

To test the CollisionGUI class's collision detection mechanism, I embarked on a two-pronged testing approach suggested by the instructions for ps-2. Initially, I opted for ad hoc testing by casually inserting points using the mouse in the GUI interface. This helped in garnering a holistic view of the system under random conditions and allowed me to observe unexpected behavior or anomalies. I also used the 'r' key many times to fill the screen with blobs, specifically looking at when Blobs would collide or not, and whether they were acting in accordance. After this casual testing, I meticulously designed two specific test cases in the Main function: one in which two bouncers should run into each other, and another whether the bouncers were guaranteed not to run into each other. Here is the code below:

```
public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            SwingUtilities.invokeLater(new Runnable() {
                public void run() {
                    CollisionGUI gui = new CollisionGUI();

                    // test 1: Blobs that should collide
                    Blob blob1 = new Bouncer(100, 100, width, height);
                    Blob blob2 = new Bouncer(105, 105, width, height);
                    gui.blobs.add(blob1);
                    gui.blobs.add(blob2);

                    gui.findColliders();
                    if(gui.colliders.contains(blob1) &&
gui.colliders.contains(blob2)) {
                        System.out.println("Test 1 passed: blobs correctly
detected as colliding");
                    }

                    gui.blobs.clear();
                    if(gui.colliders != null) {
                        gui.colliders.clear();
                    }
                }
            });
        }
    });
}
```

