Lab 5: Intersection Tests

So far, you have been detecting whether a mouse click falls within a geometric shape by testing if a position is within the bounds of the object (e.g. if a click happens in a box by checking the four walls of the box). In this lab, you will be implementing two additional ways to do intersection tests: the odd-even test and the pickbuffer test.

Odd-even test: this test is used to test intersection between a point and a shape made up of a series of line segments (e.g., star, hexagon, etc.). The general idea is to count the number of times a line (emanating from the mouse position) intersects with the set of line segments. If the number of intersection is odd, then the mouse is inside the shape. Otherwise, it’s outside.

Pickbuffer test: this is a very general test used for intersection detection with any arbitrary shapes or objects, including sprites. The idea is to make use of a secondary frame buffer (that’s invisible to the user) where each shape is rendered using a unique color. To determine if the mouse is over a shape, you would retrieve the color of the pixel where the mouse is (in the pick buffer), and compare that color to the unique color of the shape. If the two matches, then an intersection occurs.

**Basic requirements:**

1. Start with the odd_even_test. First get the support code to run.
2. Notice that in the support code, there’s a function called isect() that’s not filled in. Your job is to fill in that function such that it returns true if the mouse position (mouseX, mouseY are the two special Processing variables that keep track of the mouse position at any time) is within the bounds of the randomly generated shape. Conversely, the function should return false if the point is outside.
3. Next, work on the pickbuffer_test. Get the support code to run.
4. There are 2 functions that you will need to fill in. The first is the drawPickBuffer() function that renders the backbuffer. The second is the MyCircle.isect() function that should return true if (mouseX, mouseY) is within the circle (of course, you are required to make use of the pickbuffer in this function), and false otherwise.
   a. Note that you might need to pass extra parameters to the isect() function. If you do so, don’t forget to change the mouseMove() function where isect() is called.

**Grading:**

We will ask you to to demo in class on Thursday, October 19, 2017.