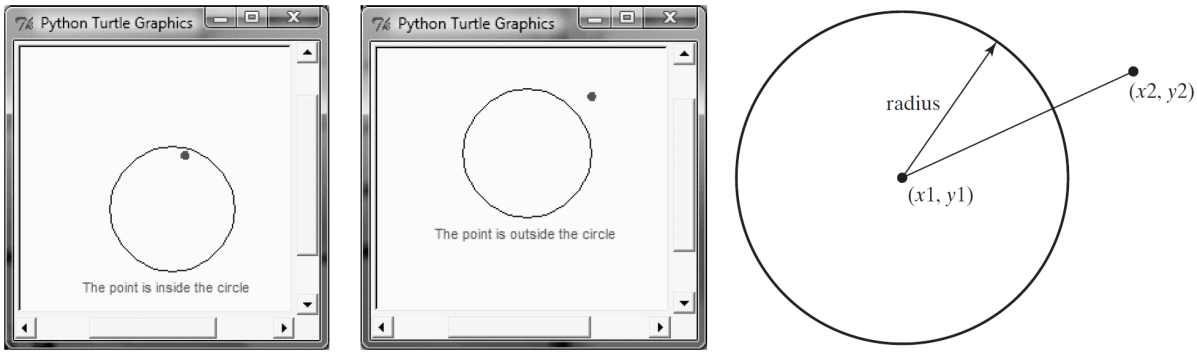
Detecting the Location of an Object

Detecting whether an object is inside another object is a common task in game programming.

In game programming, often you need to determine whether an object is inside another object. This section gives an example of testing whether a point is inside a circle. The program prompts the user to enter the center of a circle, the radius, and a point. The program then displays the circle and the point along with a message indicating whether the point is inside or outside the circle, as shown in Figure a–b.



(a) (b) (c)

FIGURE The program displays a circle, a point, and a message indicating whether the point is inside or outside the circle.

HINT: A point is in the circle if its distance to the center of the circle is less than or equal to the radius of the circle, as shown in Figure c. The formula for computing the distance is $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.

The program given in the next page that determines if a given point is located in a circle centered at x_1 *and* y_1 of radius, *r*.

This program is similar to the Lab 3 example excersize problem 21 where the origin is the center of the radius, in other words $x_1 = 0$ *and* $y_1 = 0$.

COME103 / CENG111 Computer Programming I

Lab - 3 Before Lab 3 Example Program

3 October 2021

```
1  import turtle
2
3  x1, y1 = float (input("Enter the center of a circle x, y: "))
4  radius = float (input("Enter the radius of the circle: "))
5  x2, y2 = float (input("Enter a point x, y: "))
6
7  # Draw the circle
8  turtle.penup()        # Pull the pen up
9  turtle.goto(x1, y1 - radius)
10 turtle.pendown()      # Pull the pen down
11 turtle.circle(radius)
12 # Draw the point
13 turtle.penup()        # Pull the pen up
14 turtle.goto(x2, y2)
15 turtle.pendown()      # Pull the pen down
16 turtle.begin_fill()   # Begin to fill color in a shape
17 turtle.color("red")
18 turtle.circle(3)
19 turtle.end_fill()     # Fill the shape
20
21 # Display the status
22 turtle.penup()        # Pull the pen up
23 turtle.goto(x1 - 70, y1 - radius - 20)
24 turtle.pendown()
25
26 d = ((x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1)) ** 0.5
27 if d <= radius:
28     turtle.write("The point is inside the circle")
29 else:
30     turtle.write("The point is outside the circle")
31
32 turtle.hideturtle()
33
34 turtle.done()
```

import turtle

enter input

draw a circle

draw a point

compute distance

in the circle

not in the circle

pause