

CPSC 231 – Fall 2017

Assignment 1: Analytic Geometry

Due: 9 October 2017, 9:00AM

In this assignment, you will use elements of the Python programming language including, variables, expressions, statements, functions, and conditional expressions. While the assignment does not require you to use more than this, you may use additional features of Python as you like.

Finding Intersections with Analytic Geometry

Analytical geometry, geometry embedded in a coordinate system, is fundamental to computer graphics and physics simulation (both key elements of video games). You will write a program that computes and displays the intersection of a line segment with a circle. This is a simplification of a common operation in video games, i.e., casting a ray that represents a line of sight (the line segment) to determine if an object (the circle, or sphere in three dimensions) blocks the line of sight.

Figure 1(a) shows one way to describe a line. Two points with coordinates (x_1, y_1) and (x_2, y_2) define the direction and position of the line. Given these two points, the parameter α defines the position of points along the line:

$$x = (1 - \alpha)x_1 + \alpha x_2 \quad (1)$$

$$y = (1 - \alpha)y_1 + \alpha y_2. \quad (2)$$

Note that $\alpha = 0$ corresponds to (x_1, y_1) , $\alpha = 1$ corresponds to (x_2, y_2) , $0 < \alpha < 1$, corresponds to the points between (x_1, y_1) and (x_2, y_2) , and other values of alpha correspond to points on either end of the line segment.

Figure 1(b) shows the three possible situations that can arise when finding the intersection of a line with circle: no intersection, one point of intersection when the line is tangent to the circle, and two points when the line cuts through the circle. Let (x_c, y_c) be the centre of a circle with radius r . We can find if and where the intersections occur by solving the quadratic

$$a\alpha^2 + b\alpha + c = 0$$

where

$$a = (x_2 - x_1)^2 + (y_2 - y_1)^2 \quad (3)$$

$$b = 2[(x_1 - x_c)(x_2 - x_1) + (y_1 - y_c)(y_2 - y_1)] \quad (4)$$

$$c = (x_1 - x_c)^2 + (y_1 - y_c)^2 - r^2. \quad (5)$$

Solutions are given by the (what should be familiar) quadratic formula:

$$\alpha = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

When $b^2 - 4ac < 0$, α is complex and the line does not intersect the circle. When $b^2 - 4ac = 0$, there is exactly one solution corresponding to the point where the line touches the circle tangentially. When $b^2 - 4ac > 0$, there are two real solutions corresponding to the intersection points when the line cuts through the circle. Note that if $\alpha < 0$ or $\alpha > 1$, the line intersects the circle, but not between (x_1, y_1) and (x_2, y_2) .

Instructions

Write a program to do the following.

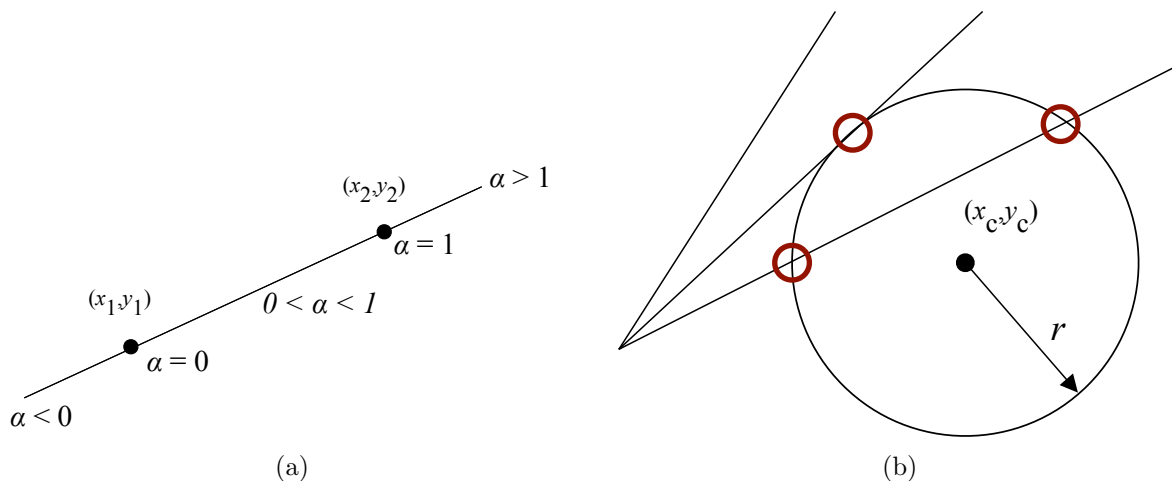


Figure 1: Lines and circles: (a) two points, (x_1, y_1) and (x_2, y_2) , define a line and the parameter α defines a position along the line, and (b) the ways in which a line intersects (or does not intersect) a circle.

1. Read in values for x_c , y_c , r , x_1 , y_1 , x_2 , and y_2 . Your program should be able to read the values from the example input files when redirected to *stdin*.
2. Use the *turtle* graphics module to display the line segment and the circle.
3. Compute the positions of the intersections between the line segment and circle, and display them if they exist.

To get full marks, you must make effective use of conditional statements, comments, and functions. There are examples of suitable displays in Figure 2 for the sample input files provided with the assignment.

Here are some suggestions to help you.

1. You may, for this assignment, assume that we will test your program with properly formed four-line input files containing
 - x_c, y_c
 - r
 - x_1, y_1
 - x_2, y_2

Thus, the following Python statement can read the first line of the file.

```
xc,yc = eval(input())
```

This is a risky way to handle input because your Python program will crash if the input is not properly formed to match what is on the left-hand side of the assignment. However, you do not have the tools to do a better job of input yet, so this simple approach will be sufficient for this assignment. You may also assume that all input we test your program with will fit into a 800-by-600 *pixel* display.

2. So, when you run your program, the command will look something like this.

```
$ python myprog.py < sample1.dat
```

The `<` character directs *python* to get its *stdin* from the file *sample1.dat*.

3. Remember to make good use of functions in your code, and good documentation is always a requirement for any programs you write.

Hand In

1. Electronic copy of your program use D2L, as per your TAs instructions. The TA will run your program to test that it functions correctly.

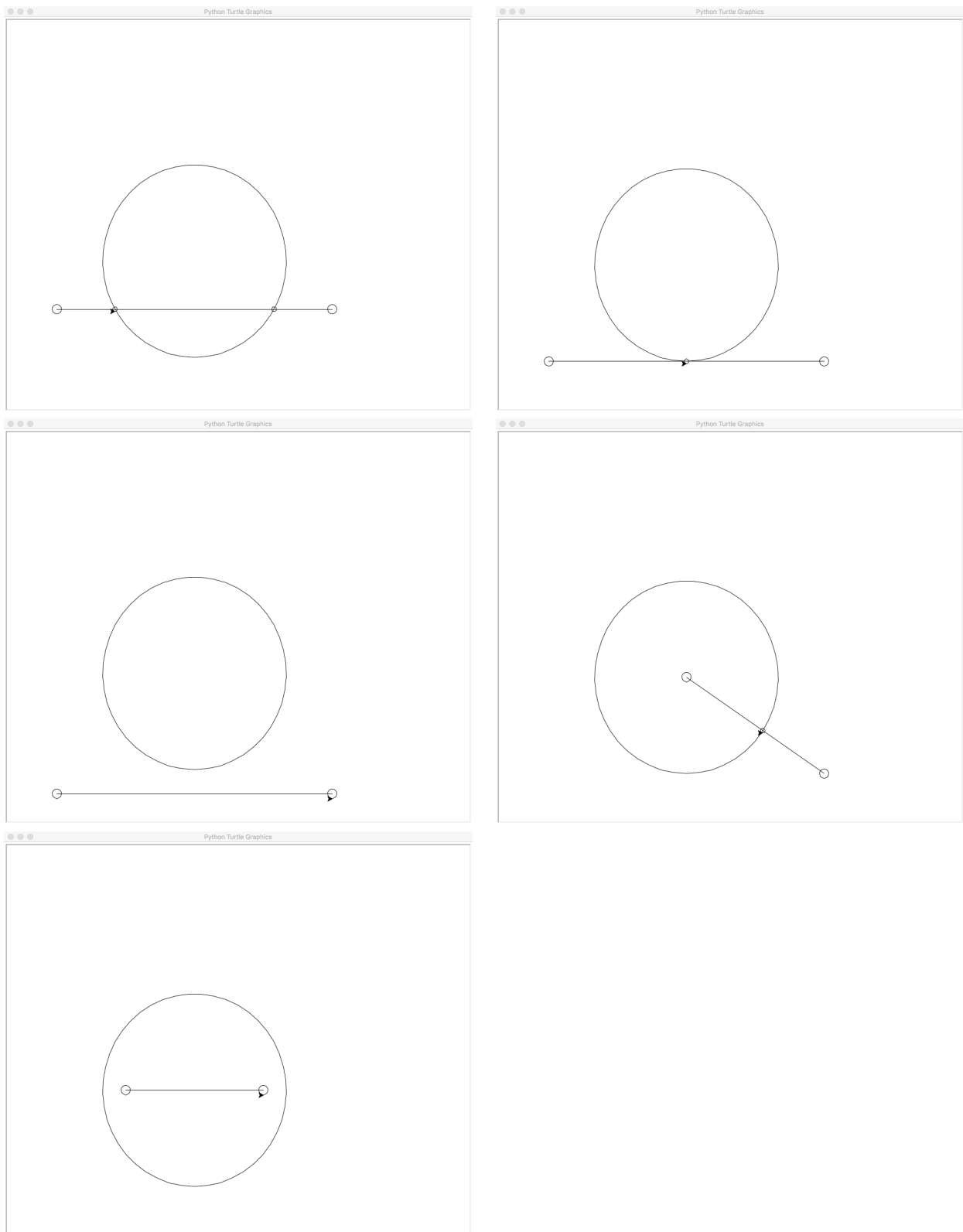


Figure 2: What your display should look like for the sample input files, *sample1.dat* through *sample5.dat*

runs: completes w/o run-time errors (1)	yes	no			
functionality: produces correct output (4)	excellent	good	fair	poor	
readability: code is clear and easy to read (2)	excellent	good	fair	poor	
modularity: use of functions (3)	excellent	good	fair	poor	
Total:					/10

Table 1: Grading rubric.

Marking

Table 1 shows the grading rubric for the assignment.

To get a grade above 3/10, your program must run without run-time errors.

Late work

After the deadline and up to 24hrs late: -5pts. After 24hrs and up to 48hrs late: -10pts. Over 48hrs late: -20pts, i.e., no assignment will be accepted beyond 48hrs after the deadline.

Collaboration

Although it can be helpful to you to discuss your program with other people, and that is a reasonable thing to do and a good way to learn, the work you hand in must ultimately be *your own work*. This is essential for you to benefit from the learning experience, and for the instructors and TAs to grade you fairly. Handing in work that is not your original work, but is represented as such, is *plagiarism* and academic misconduct. Penalties for academic misconduct are outlined in the university calendar.

Here are some tips to avoid plagiarism in your programming assignments.

1. Cite all sources of code that you hand in that are not your original work. You can put the citation into comments in your program. For example, if you find and use code found on a web site, include a comment that says, for example,

```
# the following code is from http://stackoverflow.com.
```

Use the complete URL so that the marker can check the source.
2. Citing sources will avoid accusations of plagiarism and penalties for academic misconduct. However, you may still get a low grade if your work is not the product of your own efforts.
3. Discuss and share ideas with other programmers as much as you like, but make sure that when you write your code that it is your own. A good rule of thumb is to wait 20 minutes after talking with somebody before writing your code. If you find you are exchanging code by electronic means, writing code while sitting and discussing with a fellow student, typing what you see on another person's console, then you can be sure that your code is not substantially your own, and your sources must then be cited to avoid plagiarism.
4. We will be looking for plagiarism in your code, possibly using automated software designed for the task. For example, see *Measures of Software Similarity* (MOSS - <https://theory.stanford.edu/~aiken/moss/>).

Remember, if you are having trouble with an assignment, it is always better to go to your TA and/or instructor to get help, than it is to plagiarize.