ME/IE/CS 558 – Spring 2016

Assignment 1

Due at midnight on February 8, 2016

For all assignments: *Unless specifically indicated, you are free to use any publicly available sources: papers, books, programs, online material, etc. – as long as you clearly indicate and attribute the origin and the source of the information.*

Background

As described on Wolfram's pages a well known theorem states the following. Given a rectangular billiard table with only corner pockets and sides of integer lengths m and n (with m and n relatively prime), a ball sent at a 45 degrees angle from a corner will be pocketed in another corner after $m + n - 2$. Needless to say, this theorem makes a number of unrealistic assumptions: that the ball travels on perfecly straght line, that it starts at exacty 45 degrees, that the friction plays no role, that the ball continues travel until it is pocketed, and so on.

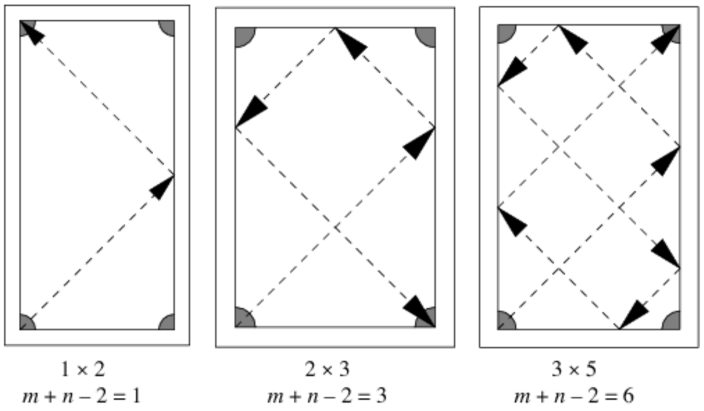

Figure 1: illustration of the theorem about motion of the billiard ball sent from a lower left corner at 45 degrees angle.

The purpose of this assignment is to simulate this idealized process, confirming the theorem. (Clearly, your program will allow you to simulate the path of the ball in variety of other situations: different starting angle, different shape of the pool, and so on. Hopefully, you will have some fun with it.)

Assignment

You are required to design and implement a program according to the following specifications:

Input: m, n, p_x, p_y, q_x, q_y, where
m is the width (horizontal x dimension) of the pool,
n is the length (vertical y dimension of the pool),
p_x, p_y and q_x, q_y are real valued coordinates of two points p and q respectively, located anywhere within the pool and specifying the two end points of a line segment pq. The coordinates of the lower left corner is $(0, 0)$ and the upper right corner is (m, n).

Output (1) The path travelled by the ball starting in the lower left corner descrbed as a sequence of points' coordinates: $(0, 0), (x_1, y_1), \ldots (x_n, y_n)$, where each points corresponds to the location where the ball collided with the wall of the pool.
(2) An integer corresponding to the number of times the path *crossed* the line segment pq. To cross each other,

the line segments must intersect at a point other than their end points.
(3) Display the path and the line segment graphically.

Other requirements Please avoid using trigonometric functions, square roots, and division, as much as possible. Make sure that you program handles all special cases and use appropriate tolerances, for all valid inputs.

Deliverables

Analysis – 50 points

1. Give careful description of logical and numerical queries required to solve the problem. Example of relevant queries may include: checking for intersection of line segments, computing intersection of segments, computing reflection, an so on. Describe how these queries are implemented. If you are using queries written by somebody else or from an existing package, provide reference and describe how the query is implemented.
2. For numerical queries explain the choice of numerical tolerances. For all queries, make sure that you cover all special cases.
3. Give a *high-level* description of your algorithm to to perform all required computations, and estimate running time of your algorithm.

Program – 50 points vspace-12pt

1. Implement a *Python 2.7* program that correspond to your analysis above. *Note: if you would like to use different programming language, you must get pre-approval.* The overall structure of your program should be explained and documented; your code should contain appropriate comments.
2. Test your program on a variety of inputs and special cases. Can your program fail due to numerical errors or missed special cases? Can you predict for what inputs this may happen, and what will happen to your program? (Programs can fail in several ways; some examples: it can produce a wrong answer, it may produce unpredictable or contradictory results, or it may simply 'die'.)

Extra credit – up to 10 points Extend the program: animate the path, apply the algorithm to more general shapes, account for loss of energy during the collision or friction, etc. Use you imagination.

Submission

Please use the course website to submit a single zip named FirstName_LastName_HW1.zip The zip archive should contain: (1) the analysis portion of the assignment, (2) the documented python source file, and (3) a PDF readme file specfying the instructions for running the code. It should also include at least 1 sample run with input and output, and specify any specific dependencies or requirements of your code.