Pygame Assignment 1

For this assignment, you will use Pygame to learn a bit about 2D programming.

Your first task is to read Chapter 2 of the Pygame book and to digest what you see there. Try the examples. The code may be downloaded and run. You will need to comprehend the information in this chapter to be able to complete the first Pygame assignment.

In this assignment, you will draw a triangle onto the screen and allow it to rotate when the left and right arrow keys on the keyboard are pressed.

Once you have played around with the examples in chapter 2 please take a look at the files in your assignment. The files contain skeleton code that you will fill out according to the descriptions in the files. Please note the files:

rotate_triangle_pygame.py	The main program containing Pygame code
triangleutils.py	Routines for geometric manipulation of the triangle
test_triangleutils.py	A unit test file to test geometric manipulation routines

Task Description

Your job is to modify **rotate_triangle_pygame.py** and **triangleutils.py**. Please look through these files and become familiar with the contents of them.

Your only task for **rotate_triangle_pygame.py** is to add a Pygame routine that draws a polygon within the 'Game Loop' of the file. The polygon in this case will be a triangle. This should be a single line of code. Other than that, all of the code for responding to keyboard events is done for you.

Most of your work lays in modifying **triangleutils.py**. Here are your tasks for this file:

1. Modify the function **degrees_to_radians** to meet the specifications in the comments. This function should accept an angle in degrees and return the corresponding radians for the number of degrees entered. Many of the trigonometric routines in the Python math library use radians as input (e.g. sin cos).
2. Modify the **create_triangle** function. This routine will create a polygon (triangle in this case) according to the specification in the comments. The points of the triangle will be a list of tuples, each tuple being a point of the triangle in Cartesian form (x, y).
3. Modify the **rotate_a_point** function. This function takes a point that is a tuple containing x and y coordinates of a point to be rotated, the x and y coordinates of the center of rotation and the number of degrees to rotate. You will have to come up with a 2D rotation formula for this task. The guide to the formula is found here: http://www.sunshine2k.de/articles/RotationDerivation.pdf
4. Modify the **rotate_polygon** function. The function takes a polygon, the number of degrees to rotate and a center of rotation. The function uses the **rotate_a_point** routine to create and return a **new** list of tuples. Each tuple in the new list should contain a new Cartesian coordinate corresponding to the result of rotating the point in the corresponding position of the original list.

Unit testing is a method that many software engineers use to test that their software meets specification. Unit tests are code that one writes to test other code that one has written.

test_triangleutils.py is a file containing unit tests. If you run this file then all of the tests should fail. As you correctly implement the routines in **triangleutils.py**, you will find that the unit tests should start to pass. If you are feeling up to it, you may modify this file on your own to add additional checks of your code.

Good Luck!!! Have Fun!!!!