OOP Assignment 10

Due: 11:59 p.m., 15 June 2016

Problem. In the zipped file 2016-0608.zip, in the project "fun," the unit test TEST (ConvexPolygon,shear) in tests.h generates a file called shear.svg can be viewed with a browser. Make use of the existing code in 2016-0608.zip, write additional code to generate a file called RegularPolygons.svg that contains the first 25 regular polygons arranged in 5 rows with five polygons in each row (see page 2, see also the file 25.svg). Note that in the function "string toSVG(vector<ConvexPolygon> & cps)" in the file ConvexPolygon.h, the size of the image has been set to 500 pixels by 500 pixels, and the view box has been set to 0, 0, 10, 10. Given that, you are free to decide the size of the regular polygons so long as they are clearly visible and do not overlap with each other when displayed in a browser.

Here are the additional functions you need. First, write a function to generate a regular polygon of n sides:

```
ConvexPolygon createRegularPolygon(int n);
```

The regular polygon generated will have its center (or centroid) located at (0,0) and the distance from the center to a vertex is 1. After generating a polygon, scale it to an appropriate dimension by multiplying it with a matrix of your choice. This can be done by the function already implemented in 2016-0608.zip:

```
ConvexPolygon apply (Matrix const & m, ConvexPolygon const & cp);
```

Lastly, translate the polygon to its final location by adding a displacement vector to it. For example, by adding (3,3) to every vertex, the polygon is moved from having a center of (0,0) to having a center of (3,3). Do this with a function with the following interface:

```
ConvexPolygon translate(Vector const & v, ConvexPolygon const & p);
```

In addition to the two functions above, add the following unit tests:

- A unit test for the function createRegularPolygon to make sure that a regular triangle is created by the by the call createRegularPolygon(3). Check that the polygon has three sides, the sides have an equal length of $\sqrt{3}$, the perimeter is $3\sqrt{3}$, and the area is $3\sqrt{3}/4$.
- A unit test for the function translate. Design your own test.
- A unit test to make sure that after scaling and translating, a regular polygon of your choice is placed at the right location and has the right size.
- A unit test for generating the file RegularPolygon.svg.

Create a project called HW10 and put the three tests in a file called utHW10.h. These tests should pass and the file RegularPolygons.svg should be generated.