A module for operations with 2D polygons
Technical Report

Ionescu Iulian Ştefan

December 3, 2016

Student Name:	Ionescu Iulian Ştefan
Group:	CE2.1
Year:	II
Section:	Computers and Information Technology - English

1 Problem Statement

The task of the project is to realise a module for operations with 2D polygons This module will provide the following operations:

- Create a polygon
- Print a polygon
- Get the minimal area rectangle containing inside the polygon
- Get the polygon area
- Get the polygon perimeter
- Get the center of mass
- Test if the polygon is convex or not
- Compare polygons by their area
- Concatenate two polygons

2 Specification of the input-output

2.1 Input

The input will consists in two .txt files containing coordinates ordered by (x,y) pairs, used to create the polygons.

2.2 Output

The output will consists in displaying the coordinates of the first polygon, the coordinates of the obtained minimal area rectangle, the polygon area, the poylgon perimeter , the convexity result,the coordinates of the concatenated polygons and the result of polygons comparison.

2.3 The modules and their description

- main.py
- poligon.py
- PoligonArea.py

i) **Main.py**

In the python file **main.py** there are tested all functions. Using the coordinates from the files, there are created two polygons . On the first polygon are tested the functions for area ,perimeter and convexity computation. The polygons are compared by their area and it's printed the bigger one. In the final , the polygons are concatenated and the resulted polygon is printed

ii) **PolygonArea.py**

In the python file **PolygonArea.py** there are provided the following functions :

- Get the length of a segment given by two points
- Get the area of a triangle
- Get the center of mass
- Get the area of a polygon

iii) **poligon.py**

In the python file **poligon.py** there are provided the following functions:

- Create a polygon
- Print a polygon
- Print the minimal area rectangle that contains inside a polygon
- Get the perimeter area
- Get the crossproduct length
- Determine if a polygon is convex or not, using the cross product function
- Concatenate the polygons and print the resulted polygon

2.4 The functions and their description

The functions from PolygonArea.py :

- **get_length(** $x1, y1, x2, y2$**)**
- **get_triangle_area(**$x0, y0, x1, y1, x2, y2$**)**
- **get_center_of_gravity(***puncte***)**
- **get_polygon_area(***puncte***)**

The functions from poligon.py:

- **create_polygon(***puncte***,***filename***)**
- **print_polygon(***puncte***)**
- **get_rectangle(** *puncte***)**
- **get_perimeter(***puncte***)**
- **CrossProductLength(**Ax, Ay, Bx, By, Cx, Cy**)**

- **IsConvex(**$puncte$**)**
- **concatenate(**$polygon1, polygon2$**)**
- **compare_polygons(**$first, second$**)**

The description of functions:

- **get_length(** $x1, y1, x2, y2$**)**
  It returns the length of a segment given by two points with the coordinates (x1,y1) , (x2,y2), using the formula: $\sqrt{(x1-x2)^2+(y1-y2)^2}$
- **get_triangle_area(**$x0, y0, x1, y1, x2, y2$**)**
  It computes and returns the area of a triangle given by 3 points. The sides of the triangle are obtained using **get_length** function. With the triangle's sides we obtain the semiperimeter so we can apply Heron's formula to compute the triangle area : $\sqrt{p*(p-a)*(p-b)*(p-c)}$
- **get_center_of_gravity(**$puncte$**)**
  It computes and returns the center of mass from a polygon, by computing the arithmetic mean of X and Y coordinates.
- **get_polygon_area(**$puncte$**)**
  It return the polygon area. It is used the following algorithm: find the polygon's center of mass, using the center of mass, split the polygon into triangles, compute the area of each traingle, summ all the traingles' areas.
- **create_polygon(**$puncte$**,**$filename$**)**
  It reads from a .txt file the coordinates of polygon points. Every point (x,y) is seen like a tuple. All tuples are stored in a list of tuples.
- **print_polygon(**$puncte$**)**
  It prints the coordinates of polygon. A pair (x,y) per line
- **get_rectangle(** $puncte$**)**
  It computes and returns the coordinates of the minimal area rectangle that contains inside the polygone. It is used the following algorithm: find the minimum and maximum value of X axis, find the minimum and maximum value of Y axis. The resulted rectangle has the coordinates : (xmin, ymin) , (xmax, ymin) , (xmax,ymax) , (xmin, ymax).
- **get_perimeter(**$puncte$**)**
  It computes and return the perimeter of polygon. We take every two nearby points and compute the lenght of segment given by these two points. The results are stored into a list and then returns the summ of lenghts.
- **cross(**o, a, b**)**
  The 2D cross product of OA and OB vectors, i.e. z-component of their

3D cross product. It returns a positive value, if OAB makes a counter-clockwise turn, negative for clockwise turn, and zero if the points are collinear.

- **IsConvex(***puncte***)**
  It returns 1 if a polygon is convex, 0 otherwise. For each set of three adjacent points A, B, C, find the cross product AB BC. If the sign of all the cross products is the same, the angles are all positive or negative (depending on the order in which we visit them) so the polygon is convex.

- **concatenate(***polygon*1, *polygon*2**)**
  It compute the convex hull of a set of 2D points. The points are provided from the concatenaned coordinates from the both polygons. We sort the points lexicographically (tuples are compared lexicographically). The duplicates are removed to detect the case we have just one unique point. The problem is separated in two subproblems: build the lower hull of the points and build the upper hull of the reversed list. We obtain two lists that are concatented, last point of each list will be omitted because it is repeated at the beginning of the other list. The resulted list gives us the convex hull.

- **compare_polygons(***first*, *second***)**
  It returns which polygon has the biggest area.

3 Results

Input:

date.txt

```
4 5
8 3
10 4
9 5
7 6
```

date2.txt

```
2 3
5 1
5 4
3 6
```

Output:

```
The polygon has the following coordinates
(4.0, 5.0)
(8.0, 3.0)
(10.0, 4.0)
(9.0, 5.0)
(7.0, 6.0)
The rectangle that covers the polygon has these coordinates
A( 4.0 , 3.0 )
B( 10.0 , 3.0 )
C( 10.0 , 6.0 )
D( 4.0 , 6.0 )
The polygon area is 7.5
The perimeter of polygon is 13.520763132540635
It is convex
Second polygon is bigger
The resulted polygon from concatenation has the following coordinates
(2.0, 3.0)
(5.0, 1.0)
(10.0, 4.0)
(9.0, 5.0)
(7.0, 6.0)
(3.0, 6.0)
```

References

[1] Python tutorials http://www.tutorialspoint.com/python/

[2] Python 3.5.2 Documentation https://docs.python.org/3/

[3] Leslie Lamport, *LaTeX: A Document Preparation System*. Addison Wesley, Massachusetts, 2nd Edition, 1994.

[4] LaTeXproject site, http://latex-project.org/