CEGE0096: Point-in-Polygon Test
1st Assignment (50%)

Module Coordinator: Dr. Aldo Lipani

1 Introduction

In this assignment you will apply your programming knowledge to the practical problem of determining whether a point lies inside or outside a polygon, which is a fundamental operation of a Geographic Information System (GIS). The point-in-polygon (PiP) problem is illustrated in Fig. 1. In this figure, the grey area represents a polygon. The red points lie outside the polygon, the green points lie inside, and the blue points like on the boundary. Visually, this is easy to see, however, it is not so straightforward to determine computationally.

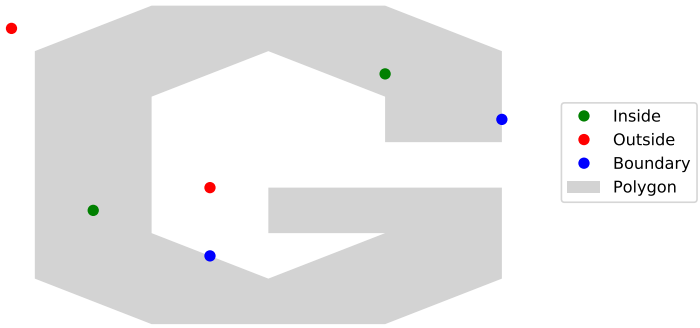

Figure 1: Example of PiP problem.

The procedure for PiP that you will use involves two steps:

1. Test if the point is inside the minimum bounding rectangle (MBR) of the polygon;
2. If it is, use PiP algorithm to test whether the point is inside the polygon.

These steps are introduced in turn in the following sections.

2 The Minimum Bounding Rectangle Algorithm

PiP is a computationally intensive operation. Therefore, it is common to first get the MBR of a polygon and test whether the point lies inside this rectangle. For the purposes of this assignment, the MBR can be found by simply taking the minimum and maximum of both coordinates of the the polygon. If a given point lies outside this rectangle, then it is definitely outside the polygon and there is no need to proceed to the full PiP algorithm. This is shown graphically in Fig. 2.

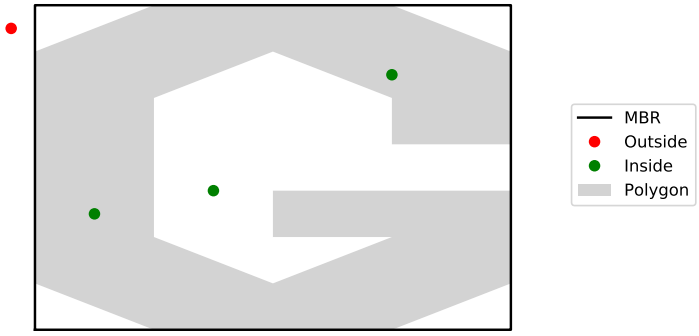

Figure 2: Example of MBR.

In Fig. 2, we can see that the MBR correctly categorises the top-left point as outside the polygon, but incorrectly categorises some of the other points. Therefore, it is necessary to use a more sophisticated algorithm to determine whether the other points lie inside or on the boundary of the polygon.

3 The Point-in-Polygon Algorithm

There are two commonly used PiP algorithms; the ray casting algorithm (RCA) and the winding number algorithm (WNA). In this assignment you are asked to implement the RCA.

The Ray Casting Algorithm. The RCA involves drawing a straight line (in any direction) from the test point, and counting how many times it crosses the boundary of the polygon. If the line crosses the boundary an odd number of times then the point lies inside the polygon. If the line crosses the boundary an even number of times then the point lies outside the polygon. This is depicted graphically in Fig. 3.

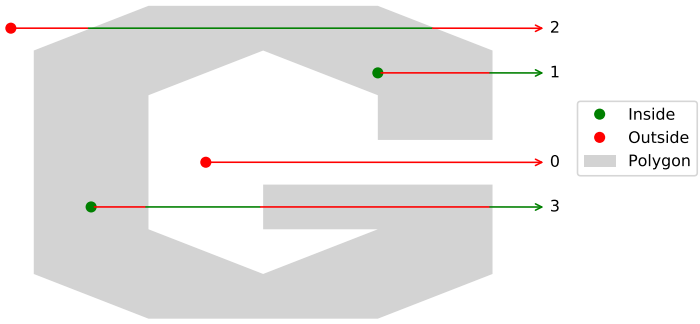

Figure 3: Example of RCA.

Special Cases. A point may lie on the boundary of the polygon in which case it is neither inside nor outside. The algorithm should return "boundary".

There is a situation where the ray casting algorithm may produce inconsistent results. If the ray passes through a vertex (point) of the polygon, this will count as crossing the boundary twice. However, this is fine as long as the ray remains inside the polygon, if the ray goes outside the polygon it produces a miss categorisation.

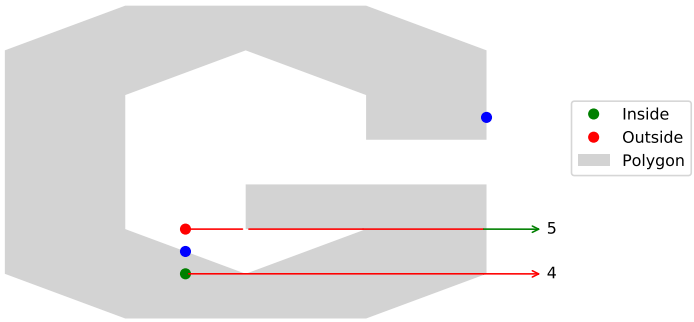

Figure 4: Example of Special Cases for RCA.

The bottom ray passes through a first vertex and two crosses are counted (= 2). However, the ray continues inside the polygon, until it crosses again a second vertex and another two crosses are counted (= 4). However, this time the ray goes outside the polygon making the point to be incorrectly categorised as outside (4 is even). Something similar happens also to the top ray. However, after having found the solution for the bottom ray, you will soon realise that this solution does not work for the top ray because another observation is required.

4 Instructions

To complete this assignment, you will build on Geometry, Polygon, Point and Line classes that you created in the practicals. Your task is to create two Python programs. The first program called 'main_from_file.py' should in order:

1. Read a list of x, y coordinates from a comma-separated values (CSV) file and create a polygon object from them. The points are provided in clock-wise order;
2. Read a list of x, y coordinates from a file and create a list of points objects for testing;
3. Categorise these points and write into a file whether they are: "inside", "outside", or "boundary";
4. Write the result of each point in a CSV file;
5. Plot the points and polygon in a plot window.

The second program called 'main_from_user.py' should:

1. Read a list of x, y coordinates from a comma-separated values (CSV) file and create a polygon object from them. The points are provided in clock-wise order;
2. Read a point from a user for testing;
3. Categorise this point and print out on screen whether it is: "inside", "outside", or "boundary";
4. Plot the point and polygon in a plot window.

You will get additional marks if you make the code objected-oriented. For example, there may be other uses for the RCA other than PiP. Are you able to implement methods that can be recycled for use in different scenarios? You should try and implement this kind of thinking throughout your code as much as possible.

There is a wealth of information on how to implement the PiP algorithm online. You are free to adapt code from online sources to work with your data and classes. If you do, any code you use should be referenced in the comments of your code with an URL and author and date (if available). Do not simply copy and paste code verbatim.

5 Material

You will be supplied with data, a project and a report template.

The data consists of (1) a CSV file containing the coordinates of a polygon, (2) a CSV file containing several test points, and (3) a CSV file containing a sample output. You can use these to test whether you PiP algorithm works. However, your work will be assessed using an unseen polygon and set of points. These polygon and set of points will be in the same format. You can use the sample output to check if your output CSV file is in the correct format.

The project template consists of a project skeleton with some indications of where to write the code. This includes also a "Plotter" class, which implements part of the visualisation required by one of the tasks. You must use this template. The template and the project repository will be available to you after following the instructions you find at the following weblink: https://classroom.github.com/a/jXWpIDCV. This weblink points to a page where you will be requested to link your GitHub account to your UCL email address. You should find your email address in the page, if you do not find it please contact the module coordinator.

The report template contains indications of what to write. These indications are highlighted in orange. Please delete them when you are done. You must use this template and its format should not be changed. If you wish, you can add an image in the cover page. This template should not be used only to get a full mark on the task related to the report, but also to make sure you get a full mark on the other tasks by presenting what you have done in a clear and satisfactory manner. The report has no word limit, but be reasonable.

6 Submission

This assignment should be submitted as follows:

1. using the provided GitHub repository in GitHub Classroom;
2. a zip file containing the project solution to the Assessment tab of the module Moodle page;
3. a pdf file of the project report to the Assessment tab of the module Moodle page;

Failing to carefully follow these instructions may result in penalties.

Note that an element of collaborative work is allowed in this project but please do not submit scripts that are identical to one another. In this case your work will not be accepted.

7 Marking Scheme

The mark scheme is distributed in tasks as follows (total of 100):

n.	Task Description	Marks
1	The successful implementation of MBR.	15
2	The successful implementation of RCA.	15
3	The successful categorisation of special cases.	10
4	Write a report about the project.	10
5	Make your code Object-oriented.	5
6	Make regular commits on the GitHub repository.	5
7	Comment your code clearly so that it can be understood by others.	5
8	Apply PEP8 style.	5
9	Plotting of points, polygons, and ray with appropriate axis labels and annotations.	5
10	Incorporate some simple error handling functionality.	5
11	Creativity marks are available under certain conditions for adding features that have not been specified.*	20

*The creativity marks will be considered only if you have scored more than zero on all the other tasks.

Any of the following is not allowed, their violation will result in penalties:

- The use of any Python module;
- Screenshots of your code pasted in your report. Please copy and paste your code as text;
- Copying code from other sources without referencing;
- Plagiarising is severely punished. Please read the assessment tab in Moodle for more details about it;
- Copying your classmate report or code is also considered plagiarism, and in this case all students involved are punished equally.
- Copying the work of previous students of this module is also considered plagiarism, referencing the code in this case does not help.