Point-in-Polygon Test

Report for the CEGE0096: 1st Assignment

Chenchen Xu
ucescx0
ucescx0.ucl.ac.uk
ccx-github/0096_1st_assignment

Introduction

The assignment is a python programming task, which includes two main methods to judge whether a point is in a polygon.The first one is MBR, Minimum Bounding Rectangle, which means we can make sure the probable position of points by comparing the coordinates of points with the vertexes of MBR.Then more accurate position can be accomplished by RCA.

My software is PyCharm, which needs some environment settings. It is mainly used for programming.The main programming language is python.I think I have completed the MBR part, and not sure for the RCA part. I'm not familiar with something about object-oriented and something professional.This is my really first taste of programming.I will try my best to study it.

For this assignment, It didn't build matplotlib support showing plots.But I have installed it on myself.

Well, first watch the slides that teacher have shown to us and I installed the kind of open sources, which provides the free licence.And then first open it, it gave me some introductions and instructions.I developed it by some small exercises step by step.

Well,I think I haven't developed it completely.I started to used it last week.

Project Description

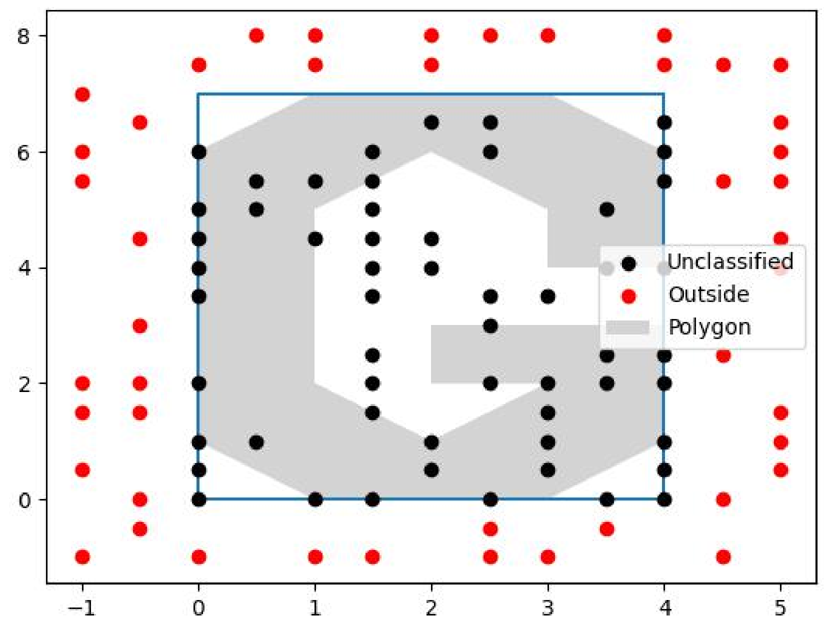
The prerequisites are some environments setting,just as the week3 slides showed.Currently,It is enough for me to develop some programming.I set he Anaconda geospatial env and make sure that git have been chosen.And I also add my github account into the PyCharm so that I can share the repository quickly and commit regularly.

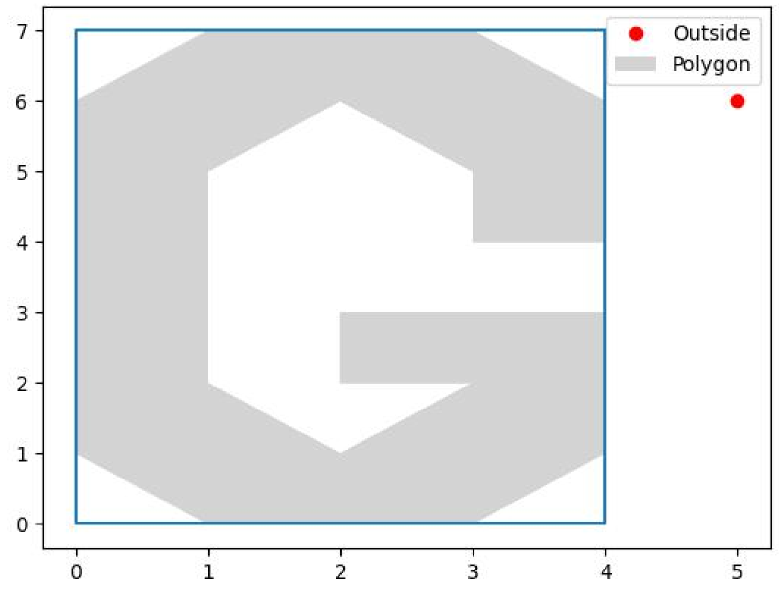
Just download the install package, and run.It will be ok.Yeah,my script is expecting the presence of any file in the same directory. Actually,I don't know Whether I have answered clearly.

Software

Task 1. The MBR Algorithm

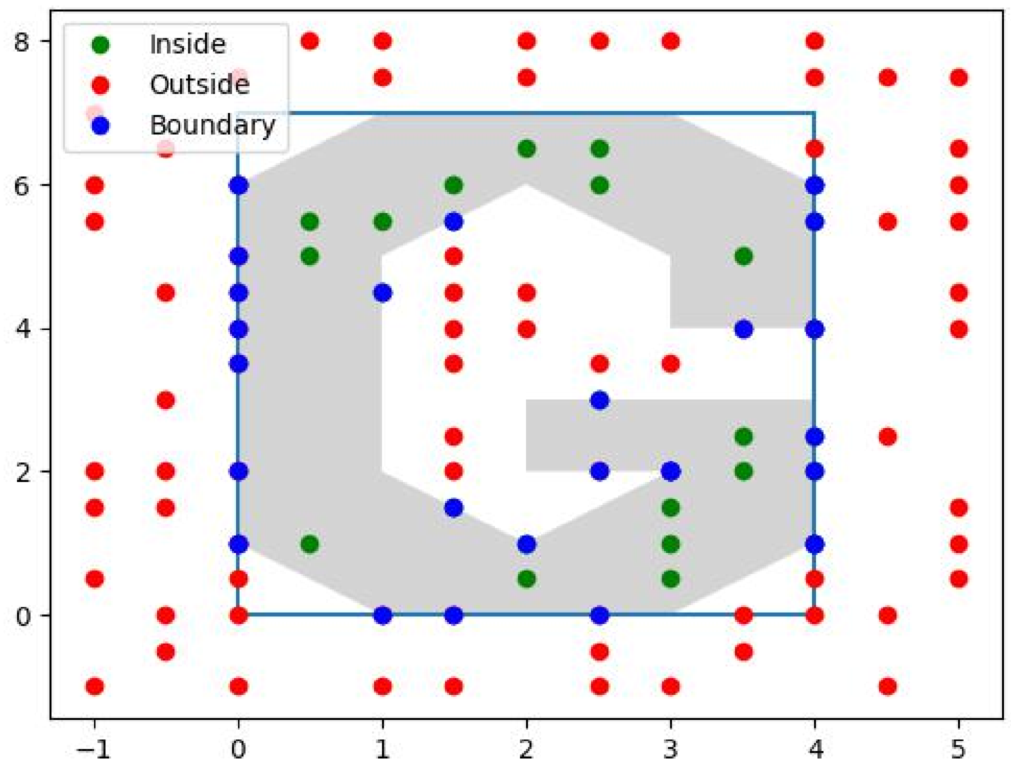
```
for l in range(len(px)):
    res1 = mbr_1(px[l], x_max, x_min)  # get x state of each point
    res2 = mbr_1(py[l], y_max, y_min)  # get y state of each point
    if res1 == 'outside' or res2 == 'outside':  # any of them is outside,then the result is outside
        plotter.add_point(px[l], py[l], 'outside')  # class object call a method from Class Plotter
    else:
        plotter.add_point(px[l], py[l], '')


def mbr_1(p, pn_max, pn_min):  # define an method to judge the position
    if p > pn_max or p < pn_min:  # larger than the maximum and less than the minimum
        return 'outside'
    else:
        return ''

def get_max(lt):  # get maximum of a list
    for k in range(len(lt)-1):
        if lt[k] >= lt[k + 1]:
            lt_max = lt[k]
            lt[k + 1] = lt_max
        else:
            lt_max = lt[k + 1]
    return lt_max

def get_min(lt):  # get minimum  of a list
    for t in range(len(lt)-1):
        if lt[t] <= lt[t + 1]:
            lt[t+1]=lt[t]
            lt_min = lt[t+1]
        else:
            lt_min = lt[t+1]
    return lt_min

def get_extremum(lt):  # get extremum of a list
    lt = sort(lt)  # sort from small to big
    lt_max = lt[len(lt) - 1]  # get the maximum
    lt_min = lt[0]  # get the minimum
    return lt_max, lt_min

with open('E:/0096_1st_assignment/Solution_1st Assignment_ucescx0/polygon.csv', 'r') as f:
    line1 = f.readline()  # read the first line from the polygon csv
    while line1:  # extract every row in the CSV
        line1 = f.readline()
        plgn = line1.split(',')  # split strings with comma
        if line1 != '':  # an empty string will be read at last while looping
            ind1.insert(j, plgn[0])  # get id lists
            plgnx.insert(j, float(plgn[1]))  # get x coordinate lists of vertexes
            plgny.insert(j, float(plgn[2]))  # get y coordinate lists of vertexes
            j = j + 1
plt.fill(plgnx, plgny, 'lightgrey')  # fill the polygon
```

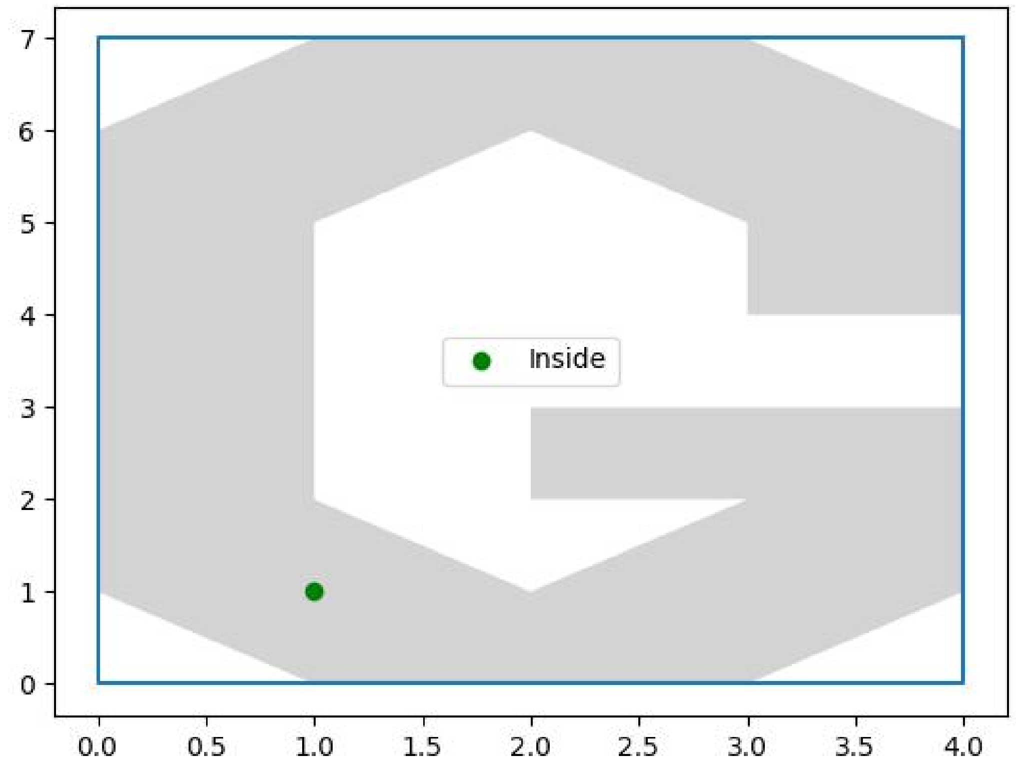




Task 2. The RCA Algorithm

```
for l in range(len(px)):  # judge each point
    res1 = mbr_1(px[l], x_max, x_min)  # use the MBR to save time and space
    res2 = mbr_1(py[l], y_max, y_min)
    if res1 == 'outside' or res2 == 'outside':
        plotter.add_point(px[l], py[l], 'outside')
    else:
        count=0
        for q in range(len(plgnx) - 1):
            # judge whether the two ends of the line segment are on both sides of the ray.
            # if not,they must not intersect.
            if (plgny[q] < py[l] <= plgny[q + 1]) or (plgny[q] >= py[l] > plgny[q + 1]):
                # point is on line or not,just the relationship of slope
                line2 = plgnx[q + 1] - (plgny[q + 1] - py[l]) * (plgnx[q + 1] - plgnx[q]) / (
                        plgny[q + 1] - plgny[q])
                if line2 < px[l]:  # point is not on line
                    count += 1  # the ray and the line intersect,point add.
        if count > 0 and count % 2 != 0:  # judge whether outside or (inside and boundary)
            plotter.add_point(px[l], py[l], 'inside')  # green points include inside and boundary

        else:
            plotter.add_point(px[l], py[l], 'outside')  # red point
```





Task 3. The Categorisation of Special Cases

I have completed the cases is point in boundary and point coincide with the vertexes.

```
for q in range(len(plgnx) - 1):
    if px[l] == plgnx[q] and py[l] == plgny[q]:# points coincide with vertexes
        plotter.add_point(px[l], py[l], 'boundary')  # blue point

    if (plgny[q] < py[l] <= plgny[q + 1]) or (plgny[q] >= py[l] > plgny[q + 1]):
        line2 = plgnx[q + 1] - (plgny[q + 1] - py[l]) * (plgnx[q + 1] - plgnx[q]) / (
                plgny[q + 1] - plgny[q])  # as above
        if line2 == px[l] and plgny[q + 1] != plgny[q]:
            plotter.add_point(px[l], py[l], 'boundary') # blue point

    elif (plgnx[q] < px[l] <= plgnx[q + 1]) or (plgnx[q] >= px[l] > plgnx[q + 1]):
        # judge the same y coordinate and point on line
        if plgny[q] - py[l] == 0 and plgny[q + 1] - py[l] == 0:
            plotter.add_point(px[l], py[l], 'boundary')  # blue point
```

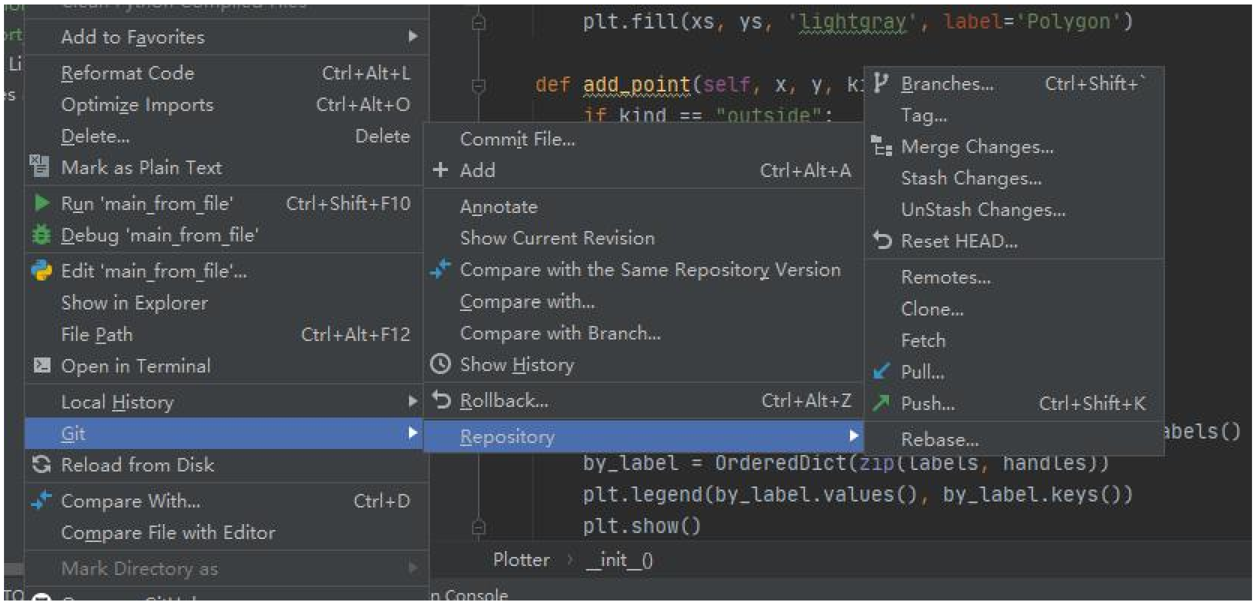
Task 5. Object-Oriented Programming

Sorry,I think I haven't make some object-oriented programming.Maybe plotter is a class object.I just use it.

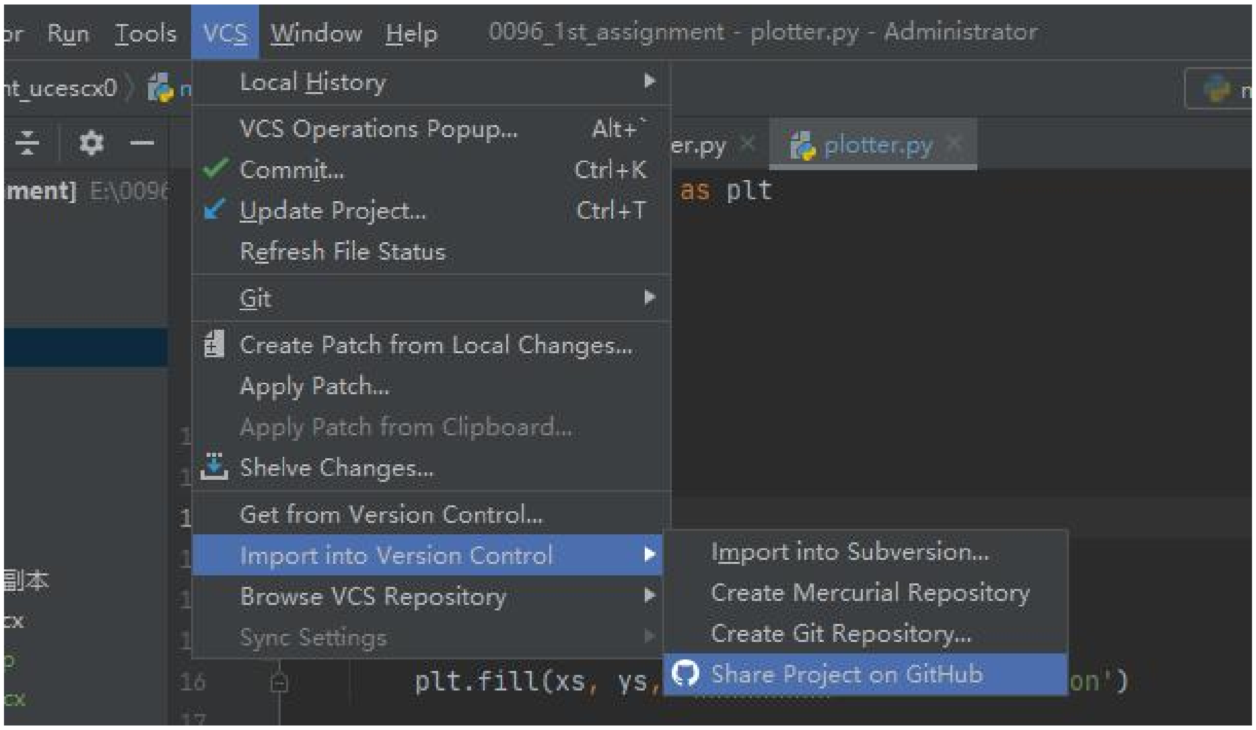
```
 if res1 == 'outside' or res2 == 'outside':  # any of them is outside,then the result is outside
      plotter.add_point(px[l], py[l], 'outside')  # class object call a method from Class Plotter
   else:
      plotter.add_point(px[l], py[l], '')
plotter.add_polygon(plgnx, plgny)
```

Task 6. On the Use of Git and GitHub

I add the file into the git and share the repository to the github and commit regularly.



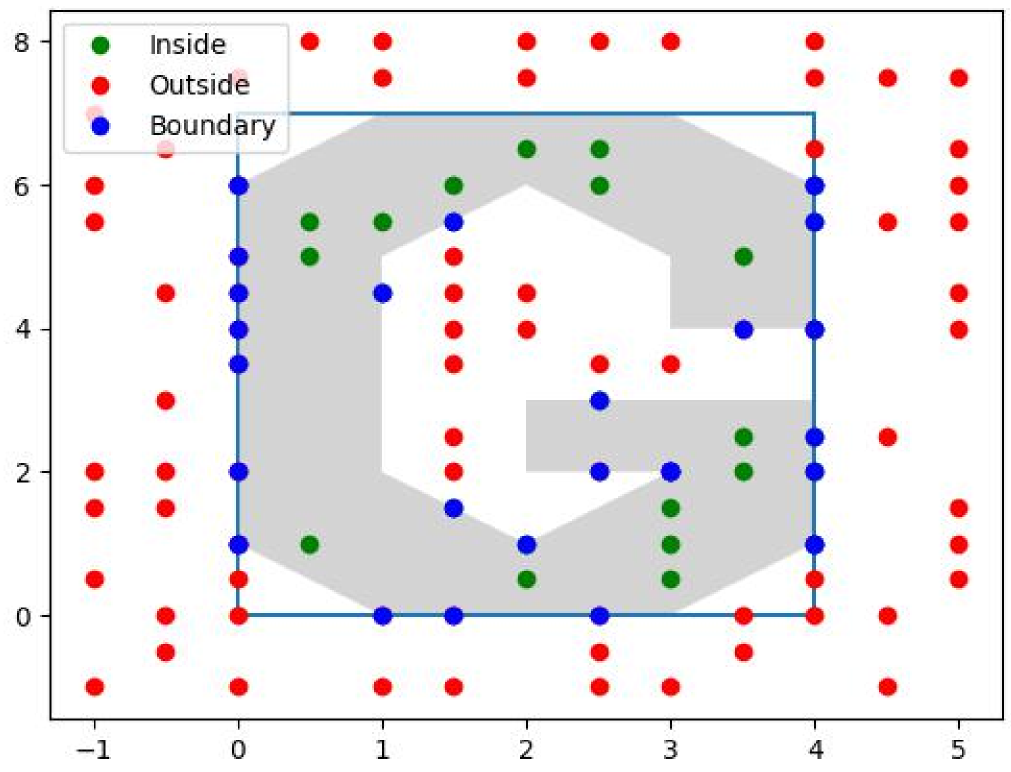
```
        plt.fill(xs, ys, 'lightgray', label='Polygon')

    def add_point(self, x, y, k
        if kind == "outside":
                                                          abels()
        by_label = OrderedDict(zip(labels, handles))
        plt.legend(by_label.values(), by_label.keys())
        plt.show()
```


Run
Tools
VCS
Window
Help
0096_1st_assignment - plotter.py - Administrator
Local History
VCS Operations Popup... Alt+`
Commit... Ctrl+K
Update Project... Ctrl+T
Refresh File Status
Git
Create Patch from Local Changes...
Apply Patch...
Apply Patch from Clipboard...
Shelve Changes...
Get from Version Control...
Import into Version Control
Browse VCS Repository
Sync Settings
Import into Subversion...
Create Mercurial Repository
Create Git Repository...
Share Project on GitHub
plotter.py

Task 9. Plotting

I didn't implement any additional functionalities.

```
if res1 == 'outside' or res2 == 'outside':  # any of them is outside,then the result is outside
        plotter.add_point(px[l], py[l], 'outside')  # class object call a method from Class Plotter
    else:
        plotter.add_point(px[l], py[l], '')
plotter.add_polygon(plgnx, plgny)

Plotter.show()
```



Task 10. Error Handling

When I want to get the minimum of y coordinate of vertexes in the polygon,the minimum is always one,but in real,it should be zero.I really can’t understand which error I have made.You know, the code is right when splitting from the whole project.Later,I use the numpy,I don’t know whether it can be used.Is it a python module?Then I got the right answer.

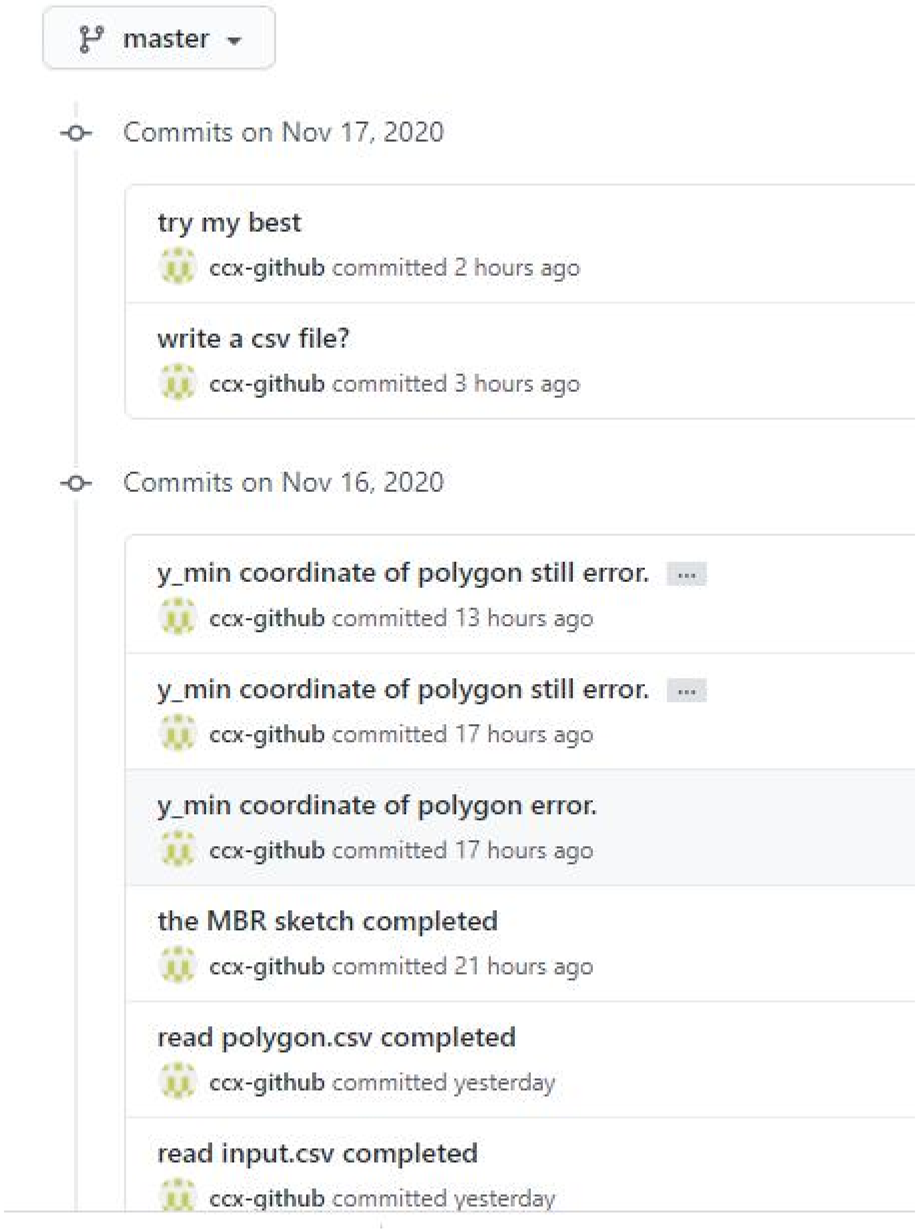
```
def get_max(lt):  # get maximum of a list
    for k in range(len(lt)-1):
        if lt[k] >= lt[k + 1]:
            lt_max = lt[k]
            lt[k + 1] = lt_max
        else:
            lt_max = lt[k + 1]
    return lt_max

def get_min(lt):  # get minimum  of a list
    for t in range(len(lt)-1):
        if lt[t] <= lt[t + 1]:
            lt[t+1]=lt[t]
            lt_min = lt[t+1]
        else:
            lt_min = lt[t+1]

    return lt_min
for l in range(len(px)):
    res1 = mbr_1(px[l], x_max, x_min)  # get x state of each point
    res2 = mbr_1(py[l], y_max, y_min)  # get y state of each point
    if res1 == 'outside' or res2 == 'outside':  # any of them is outside,then the result is outside
        plotter.add_point(px[l], py[l], 'outside')  # class object call a method from Class Plotter
    else:
        plotter.add_point(px[l], py[l], '')
```

Task 11. Additional Features

Sorry,I don't know what is a additional feature.I think I don't have any of them.I must take more efforts to this module.

Git Log

There is nothing about my git log.About the git,there are always some errors.I reinstalled so many times. I think the git log is about commit condition.Here is my part of commit screenshot.


master

Commits on Nov 17, 2020

try my best
ccx-github committed 2 hours ago

write a csv file?
ccx-github committed 3 hours ago

Commits on Nov 16, 2020

y_min coordinate of polygon still error. ...
ccx-github committed 13 hours ago

y_min coordinate of polygon still error. ...
ccx-github committed 17 hours ago

y_min coordinate of polygon error.
ccx-github committed 17 hours ago

the MBR sketch completed
ccx-github committed 21 hours ago

read polygon.csv completed
ccx-github committed yesterday

read input.csv completed
ccx-github committed yesterday