Yuqing Liu
NetId#yl7164
Assignment1 Report
CSCI-GA 2270-001
Graduate Computer Graphics

Assignment1 Report

Environment:

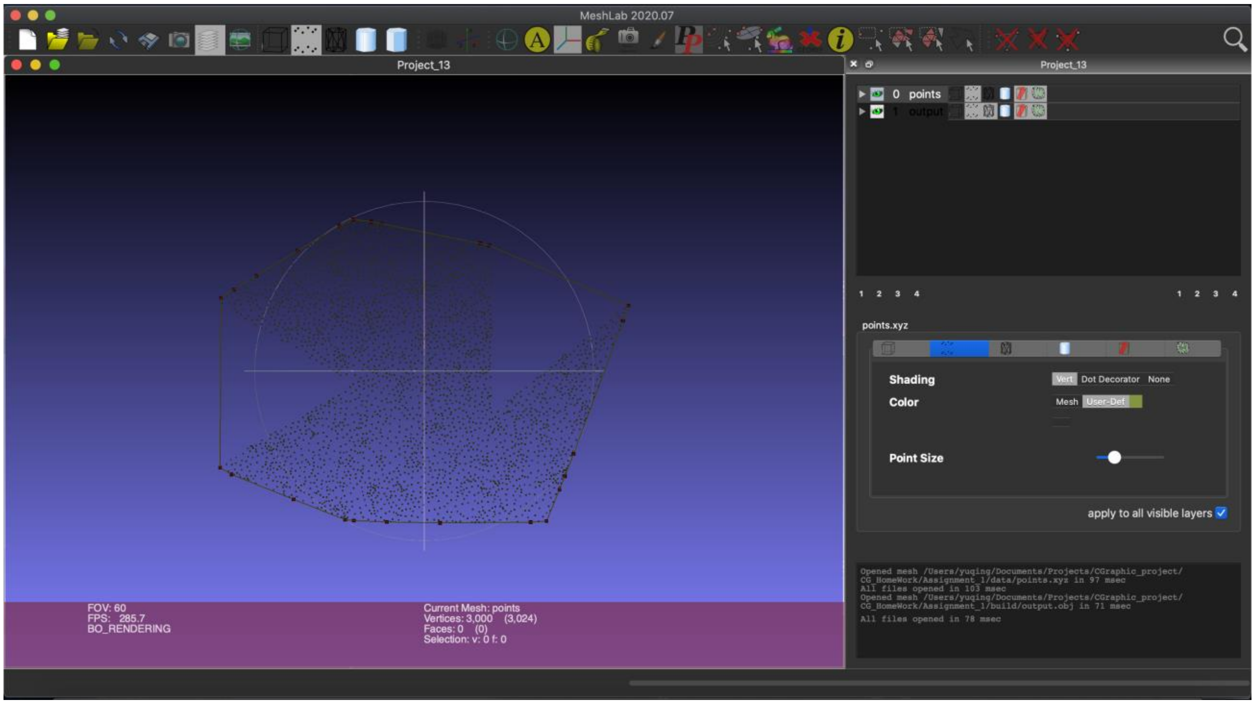
Operating System: macOS Catalina; Version: 10.15.6
Compiler: cmake version 3.18.3 ; Apple clang version 12.0.0

Convex Hull

Implementation:

1. Function det takes two points as input and returns the square of the distance between two points.
2. Struct Compare is using in sorting all the input points in counter-clock order from point p0. (If 2 points and p0 are inline, the closer to p0 one comes first.)
3. Function salientAngle is used to determine if the rob is doing a left turn of not. It returns true if it a left turn or the point is colinear to the line before, returns false if it a right turn.
4. Function convex_hull returns the result of hull. It first find the most down and left point of all the point and use that as p0 to do a sort. Then it tests if there are more than 3 points. Finally if there is more than 3 points, it calls salientAngle to determine if the point is belongs to hull or not.
5. Function load_xyz takes filename as input and read all the points from the file.
6. Function save_obj saves the result point in hull as an output obj file.

Result:



```
FOV: 60
FPS:   285.7
BO_RENDERING
```

```
Current Mesh: points
Vertices: 3,000    (3,024)
Faces: 0    (0)
Selection: v: 0 f: 0
```

```
Opened mesh /Users/yuqing/Documents/Projects/CGraphic_project/
CG_HomeWork/Assignment_1/data/points.xyz in 97 msec
All files opened in 103 msec
Opened mesh /Users/yuqing/Documents/Projects/CGraphic_project/
CG_HomeWork/Assignment_1/build/output.obj in 71 msec
All files opened in 78 msec
```

The all points are showed in small green dots. The points belong to result Hull is showed in red larger dots. And the hull is in lines.

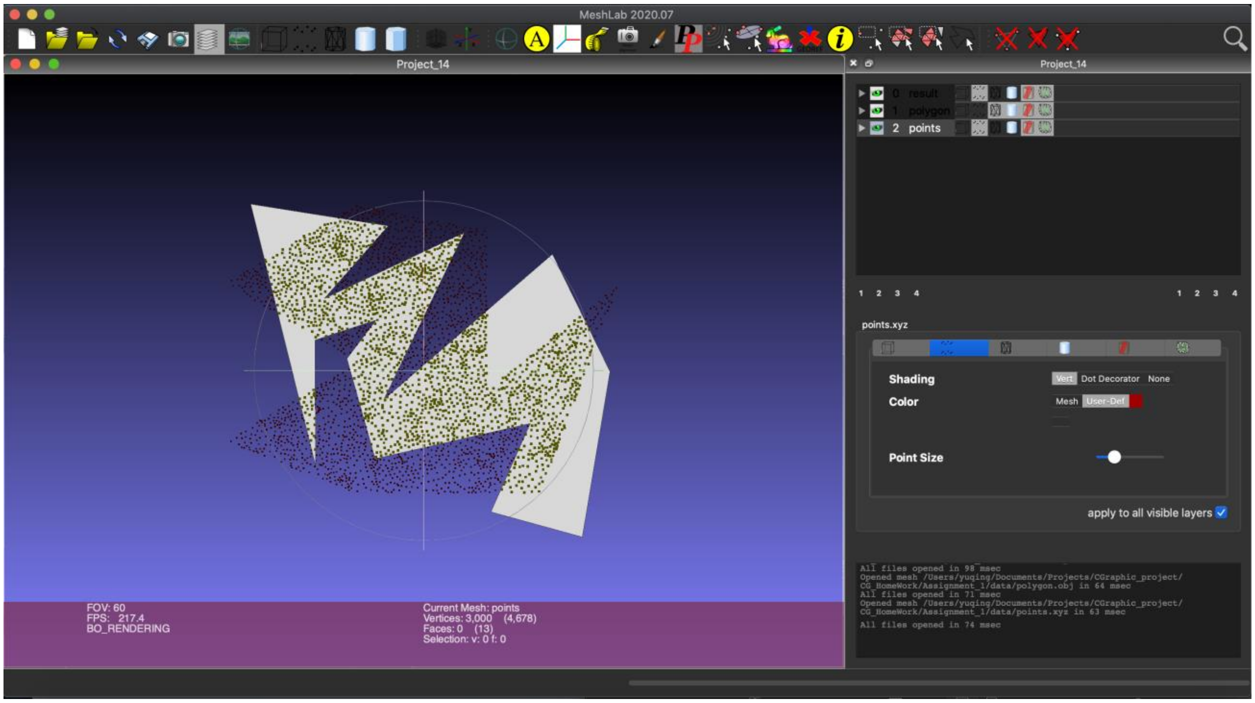
Point In Polygon

Implementation:

1. Function salientAngle takes 3 points as input, and determent if point c is on the line ab's left (return 1), right (return -1), colinear (return 0).
2. Function online takes 3 points as input, and determent if point c is on the segment ab.
3. Function intersect_segment takes 4 points, return true iff [a,b] intersects [c,d]. It checks if point c and point d are on the different side of segment ab, and if point a and point b are on the different side of segment cd. If they are all not at same side, there is intersection. There are special cases with two segment are colinear. If there are colinear exist, the function checks if they have overlap or not. It there is overlap, there is intersection, otherwise not.
4. Function is_inside checks if point query is inside polygon or not. It first finds the max coordinate and time 1000 on it to find a point is definitely onside the polygon. The it compares the segment of query and outside point with each edge of the polygon to see if there is a intersect or not, and count the intersect number. If there is even number of intersection, the point is outside, otherwise it's inside.

5. Function load_xyz takes filename as input and read all the points from the file.
6. Function load_obj takes filename as input and get all points of the polygon.
7. Function save_xyz save the result of all the points inside the polygon into a xyz file.

Result:



Shading: Vert | Dot Decorator | None

Color: Mesh | User-Def

Point Size

apply to all visible layers ☑

```
FOV: 60
FPS:   217.4
BO_RENDERING
```

```
Current Mesh: points
Vertices: 3,000   (4,678)
Faces: 0   (13)
Selection: v: 0 f: 0
```

```
All files opened in 98 msec
Opened mesh /Users/yuqing/Documents/Projects/CGraphic_project/
CG_HomeWork/Assignment_1/data/polygon.obj in 64 msec
All files opened in 71 msec
Opened mesh /Users/yuqing/Documents/Projects/CGraphic_project/
CG_HomeWork/Assignment_1/data/points.xyz in 63 msec
All files opened in 74 msec
```

The all points are showed in small red dots. The points inside the polygon is showed in yellow larger dots. And the polygon is in white.