BCA 608 Assignment 1

Bugrahan Akbulut

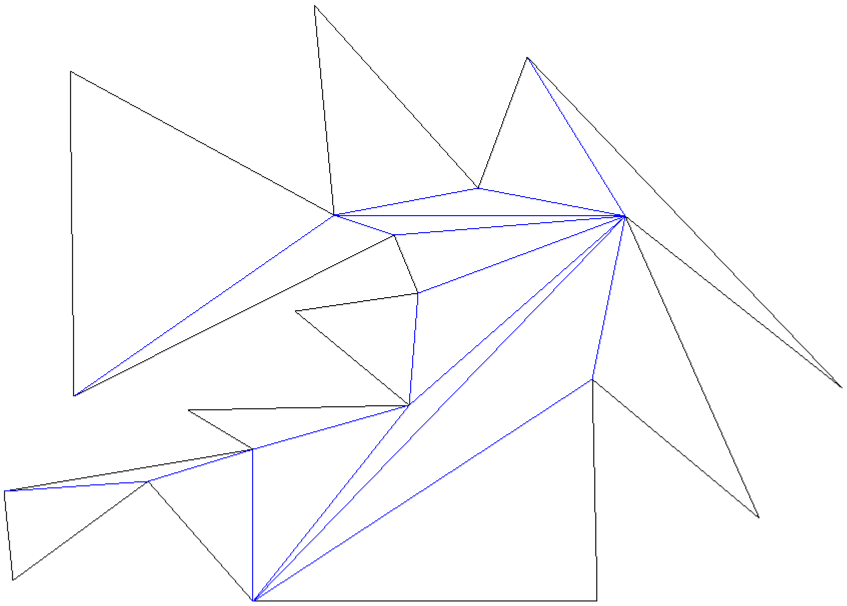
November 27, 2020

1 Introduction and Overview

When I was implementing this assignment, I used OpenGL with C++. Esspecially I used OpenGL Utility Kit(glut) library for drawing window and I/O operations, for mathematical operations I preferred using OpenGL Mathtematics(glm) library.
I used Microsoft Visual Studio for development environment. To keep project clean and solid I put project on **github.** To avoid any kind of cheating I kept repository private until due date.

2 Implementation Details

To implement assignment I created two classes : Polygon and Line.
In Line class there are few fields such as *LineColor*, *StartPosition*, *FinishPosition* etc. Besides it has methods which called *DidIntersect*. This method takes another line and returns true if they are intersected, false otherwise.
The class Polygon keeps vertices, edges and diagonals in *vector* data type. Also it has methods to determine diagonals which provides triuanglate polygon.
I implemented firstly finding corresponding diagonals of polygon. To find the correct diagonals among all the lines between the vertices I applied intersection test firstly. For each "*possible diagonal*" I checked if it has any intersection points with one of the edge of the polygon. After found lines with has not any intersection points with edges, I eliminated which are located outside of polygon. To find triangulate diagonals lastly I created diagonal graph by using same methods for edge-diagonal intersections after build diagonal graph by coloring nodes according to the introduction I created triangulated polygon.



3 Solution Analysis

Beginning of algorithm by traversing each vertex try to create a candidate diagonal for each vertex then it has complexity $O(n^2)$. after that for each diagonal we are testing it has any intersection point it's complexity is $O(n^3)$(In worst case we have $(n-3) \times (n-3)$ candidate diagonals and $(n-1)$ edge than we had to make $(n-3) \times (n-3) \times (n-1)$ intersection test which notated by $O(n^3)$). After intersection test we need to check for every candidate diagonal is in polygon, again for $(n-3) \times (n-3) \times (n-1)$ in polygon test for every candidate diagonals which means complexity is $O(n^3)$. Building Diagonal Graph process has complexity $O(n^2)$ since we need to check every diagonal line has any intersection with other diagonals. In worst case we have $(n-3)$ diagonals, we need to make $(n-3) \times (n-3)/2$ intersection check. Lastly complexity of colorizing created graph is $O(n)$. Than we have complexity :

$$O(n^2) + O(n^3) + O(n^3) + O(n^2) + O(n) = O(n^3)$$

4 Program Usage

You can select vertices by left mouse button you need to use middle mouse button or right mouse button close polygon. If you use right mouse button you can see every step of triangulation process, middle button show results instantly. To run program smoothly and flawless you need to install required drivers by OpenGL.