Voronoi Diagram Solver with OpenGL

Cagin Agirdemir
Assignment Report in scope of
Computional Geometry Course
Hacettepe University, Ankara
caginagirdemir@hacettepe.edu.tr

Abstract

This program aims to solving Voronoi diagram which comes from user. Areas can be arrange in terms of square size from user input. As a method used to brute force which means every voronoi area through user select point calculated according to other voronoi areas these selected by user.

Keywords

Voronoi Diagrams

I. Introduction

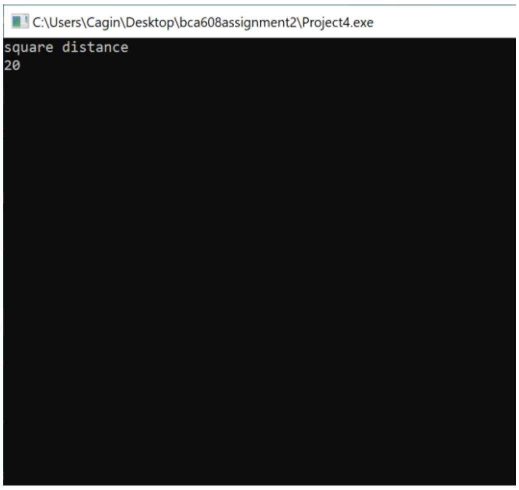
This paper related to Voronoi Diagrams and this paper explains what is Voronoi Diagrams and how to solve and simulate. Additionally, this paper explain which used to calculations when calculate voronoi diagrams linked to user selected points.

II. Voronoi Diagrams

A Voronoi diagram is the computational geometry concept that represents partition of the given space onto regions, with bounds determined by distances to a specified family of objects. Polygon Voronoi extension provides implementation of the Voronoi diagram data structure in the 2D space. The internal representation consists of the three arrays, that respectively contain: Voronoi cells (represent the area around the input sites bounded by the Voronoi edges), Voronoi vertices (points where three or more Voronoi edges intersect), Voronoi edges (one dimensional curves containing points equidistant from the two closest input sites). Each of the primitives (cell, vertex, edge) contains pointers to the other linked primitives, so that it's always possible to efficiently traverse the Voronoi graph. The picture below shows the Voronoi vertices in red, Voronoi edges in black, input sites that correspond to the Voronoi cells in blue. It is considered, that each input segment consists of the three sites: segment itself and its endpoints. As the result, two additional Voronoi edges are constructed per each input segment. This is made to simplify the representation of the Voronoi diagram and Voronoi edges in particular. [1]

III. Application

First of all, when the program start, first task is asked to enter square distance as unit to user.



C:\Users\Cagin\Desktop\bca608assignment2\Project4.exe

```
square distance
20
```

Figure 1. Square distance prompt

Afterwards this prompt, draw working space consisting of squares as a second window which include squares according to entered square distances from user.

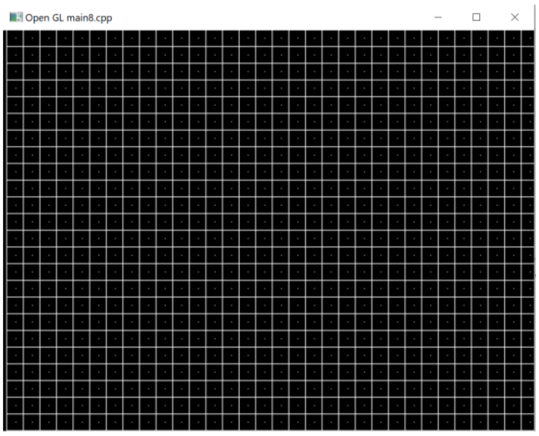


Figure 2. Voronoi Diagram Space

All square would placed according to window of size (640,480) which defined by me druing the programing. This place process define how much square draw for first line and how much square draw along the column.

Also drawn a midpoint with all squares and these points determine squqare belong to which voronoi area by calculating distance between square mid point and voronoi vertex from user.

When user click for first voronoi area, all squares would pointed same color because of there is no other comparable voronoi vertex in terms of distance. Color again determine for every voronoi area by the way of random number function.

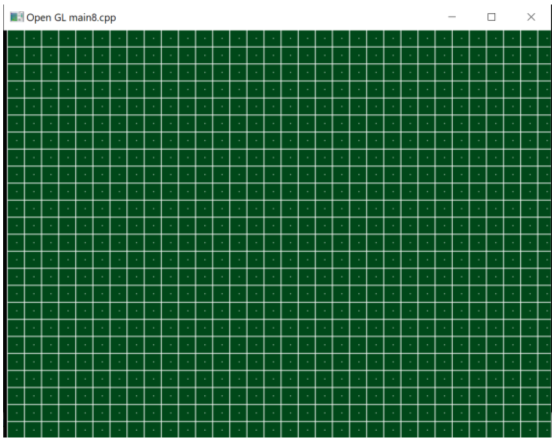


Figure 3. First Voronoi Diagram Area

When user click for determine new voronoi vertex also new voronoi area, program compare distance between every square mid ponint and every voronoi vertex. In this way, the square is painted in the color of that voronoi area whichever is close to the voronoi vertex.

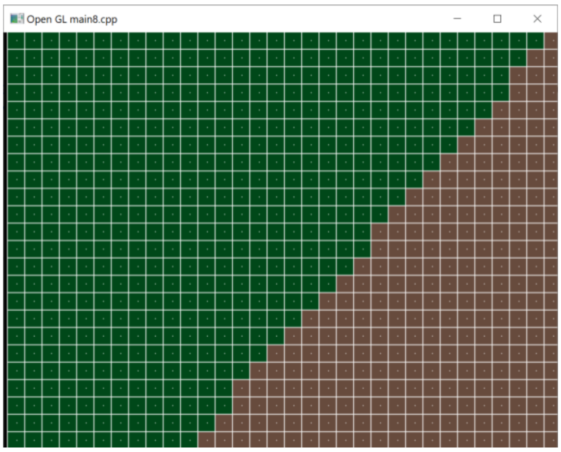


Figure 4. Selected two Voronoi Area

If selected than one voronoi vertex, calculate distances again in loop all time. If the user wants to delete vertex, could use left click menu which include just last vertex as options.

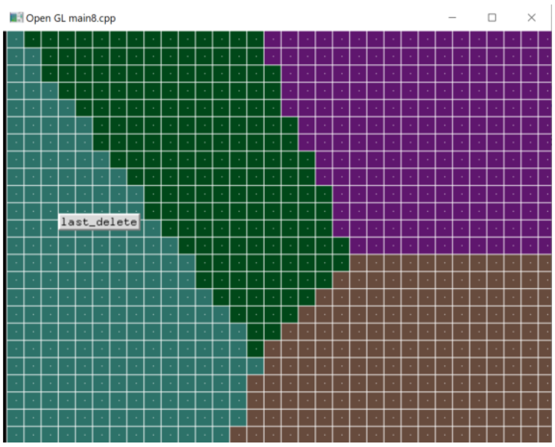


Figure 5. Voronoi area delete funcion

User can delete all voronoi areas though right click menu option. It is necessary to repeat this process until all voronoi fields are deleted. In Figure 5, pruple area is last selected voronoi area. When delete a area, program shown last situation of squares engagements.

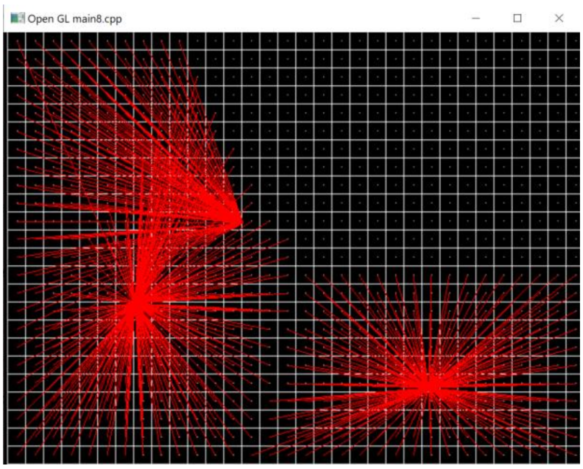


Figure 6. After delete function, square engagements

After delete a voronoi area, this area mid point be idle. When user click to anywhere, program calculate distances for every square midpoints linked voronoi vertexs. After this calculation, all square painted whichever is close to the voronoi vertes again.

REFERENCES

[1] Kang J. (2008) Voronoi Diagram. In: Shekhar S., Xiong H. (eds) Encyclopedia of GIS. Springer, Boston, MA.