

### 3. LABORATORIJSKA VJEŽBA

#### Delaunayeva triangulacija i Voronoijev dijagram

*Nikola Nađ, 0036491265*

U sklopu ove vježbe izrađen je generator Delaunayeve triangulacije i njoj dualnog Voronoijevog dijagrama, na temelju točaka zadanih mišem. Vježba je izrađena u razvojnom okruženju Eclipse i pisana je u programskom jeziku Java. Korištena je biblioteka LWJGL2 (Lightweight Java Game Library), koja među ostalim pruža i OpenGL API. Za pokretanje vježbe najprije je potrebno otvoriti projekt u Eclipse-u. Zatim je potrebno dodati sve JAR datoteke iz „lib“ foldera u projekt te native biblioteke, također locirane u „lib“ folderu.

Delaunayeva triangulacija za dani skup točaka  $S$  je triangulacija  $DT(S)$  takva da niti jedna točka iz  $S$  ne leži unutar opisane kružnice bilo kojeg trokuta iz  $DT(S)$ . Ova triangulacija maksimizira najmanji kut u trokutima – nastoje se izbjeći oštri trokuti.

Voronoijev dijagram je podjela prostora u ćelije na temelju skupa točaka  $S$ . Za svaku točku  $s$  iz  $S$  generira se po jedna ćelija, a ta ćelija sastavljena je od svih točaka kojima je točka  $s$  najbliža točka iz skupa  $S$ . Voronoijev dijagram je dualan graf Delaunayevoj triangulaciji: simetrane stranice trokuta dobivenih triangulacijom predstavljaju stranice Voronoijevih ćelija, dok središta opisanih kružnica predstavljaju vrhove ćelija.

U ovoj implementaciji, najprije se na temelju zadanih točaka računa triangulacija, a zatim se preko dualnosti grafova konstruira Voronoijev dijagram. Triangulacija se konstruira po

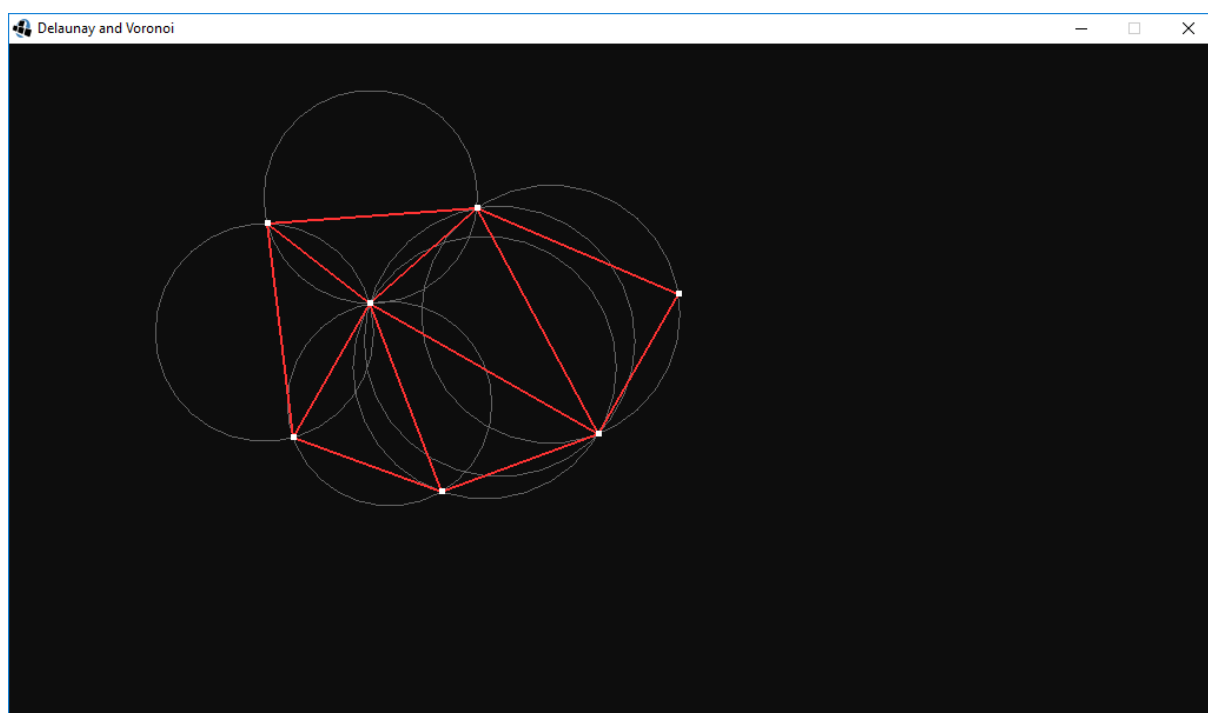
Bowyer-Watsonovom algoritmu, čiji je pseudokod dan u nastavku:

```
function BowyerWatson (pointList)
    triangulation := empty triangle mesh data structure
    add super-triangle to triangulation
    for each point in pointList do
        badTriangles := empty set
        for each triangle in triangulation do
            if point is inside circumcircle of triangle
                add triangle to badTriangles
        polygon := empty set
        for each triangle in badTriangles do
            for each edge in triangle do
                if edge is not shared by any other triangles in
badTriangles
                    add edge to polygon
        for each triangle in badTriangles do
            remove triangle from triangulation
        for each edge in polygon do
            newTri := form a triangle from edge to point
            add newTri to triangulation
    for each triangle in triangulation
        if triangle contains a vertex from original super-triangle
            remove triangle from triangulation
    return triangulation
```

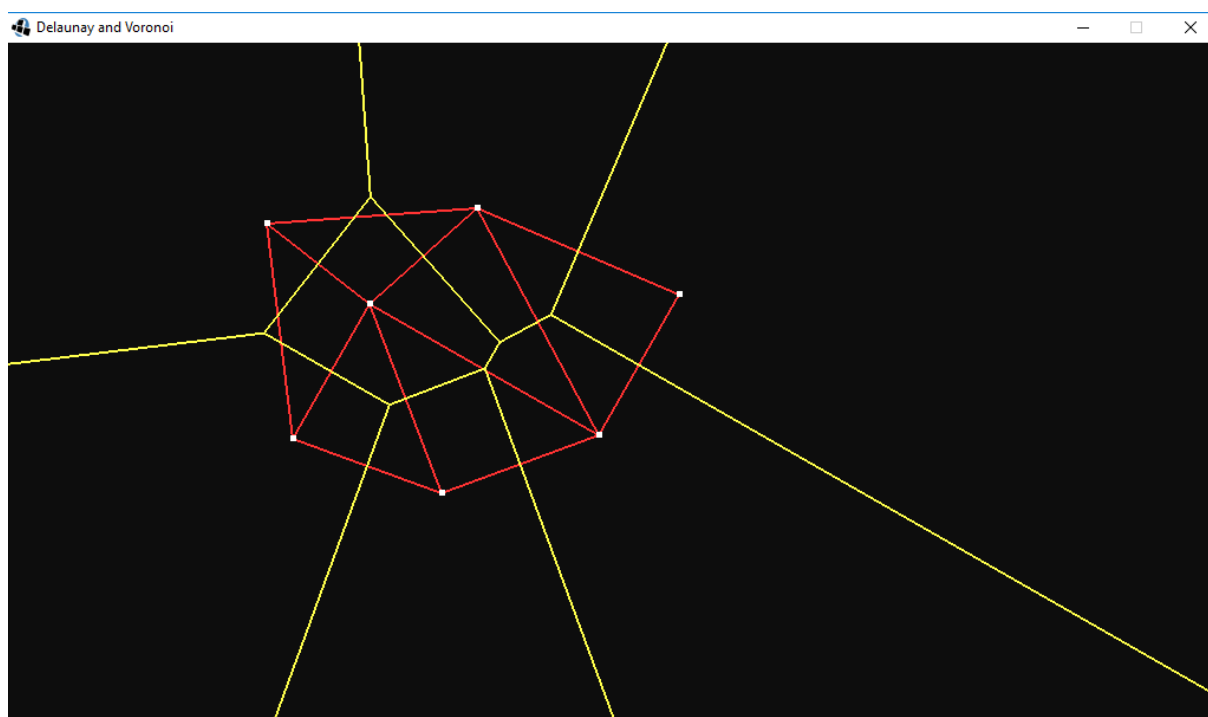
Funkcionalnosti programa opisane su u sljedećoj tablici:

Tipka	Funkcionalnost
lijeva tipka miša	dodavanje novih točaka u triangulaciju
D	prikaz ili skrivanje Delaunayeve triangulacije
V	prikaz ili skrivanje Voronoijevog dijagrama
C	prikaz ili skrivanje opisanih kružnica trokuta

Prikaz programa u izvođenju, nakon dodavanja 7 točaka:



*Slika 1 Delaunayeva triangulacija (crveno) i opisane kružnice trokuta (sivo)*



*Slika 2 Delaunayeva triangulacija (crveno) i Voronoijev dijagram (žuto)*