

## Vježba 2: Podjela ravnine na poligone

Cilj je ove vježbe podjela ravnine na poligone koristeći Delaunay triangulaciju i OpenCV biblioteku.

Kako bi mogli odrediti Delaunay triangulaciju potrebno je prvo stvoriti prozor koji će nam djelovati kao radno područje. Stvaramo prozor veličine 440x440px te unutar njega izdvajamo centrirano područje mape koje je dimenzija 400x400px. Sada je potrebno stvoriti točke pomoću kojih ćemo ravninu podijeliti na poligone. Prve 4 točke postaviti ćemo kao rubne točke (po jedna na svaki vrh pravokutnika mape), a još 16 točaka nasumično generiramo unutar zadanog pravokutnika mape.

Koristeći klasu *Subdiv2D* iz OpenCV biblioteke, određujemo pravokutnik mape kao radno područje te mu pridružujemo prethodno generirane točke nakon čega se obavlja Delaunay triangulacija. Određujemo broj polurubova koji su generirani Delaunay triangulacijom te određene rubove crtamo u prethodno zadanom prozoru.

Sada želimo omogućiti odabir pojedinog trokuta u ravnini te označiti taj trokut i sve one trokute koji s njime dijele stranicu. Stvaramo funkciju *onClick* te ju pozivamo pomoću *MouseCallback* metode. U funkciji *onClick* predajemo korisnički zadane parametre prozora i varijablu *subdiv* u kojoj je obavljena triangulacija. Kako bi odredili u kojem trokutu se nalazi pokazivač miša, u trenutku kada se nakon klika otpusti lijeva tipka miša, provjeravamo poziciju miša (koordinate relativne s obzirom na ishodište zadanog prozora). Trokute dobivene Delaunay triangulacijom možemo dobiti koristeći funkciju *subdiv.getTriangleList* koja nam vraća vektor vektora triju točaka koje čine pojedini trokut. Kako imamo sve točke svakog od trokuta možemo koristeći metodu *pointPolygonTest* odrediti nalazi li se pokazivač unutar poligona (odnosno trokuta). Kada pronađemo traženi trokut, bilježimo koji je to trokut po redu, bojamo ga crvenom bojom te pronalazimo njegove susjede. Kao što vrlo lako možemo zaključiti, trokut susjed našem odabranom trokutu je onaj trokut koji s njime dijeli jednu stranicu. Pretražujemo ostale trokute, pronalazimo potrebne te ih spremamo i bojamo plavom bojom.

Ovime smo uspješno ostvarili zadatke:

1. Generiranje 20 nasumično izabranih točaka unutar kvadrata dimenzije 400px;
2. Generiranje Delaunay triangulacije skupa točaka iz 1;
3. Prikaz triangulacije iz koraka 2 u kvadratnom prozoru dimenzija 440px;
4. Omogućavanje izbora nekog od trokuta triangulacije klikom lijeve tipke miša;
5. Prikaz izabranog trokuta crvenom bojom, a njemu susjednih trokutova plavom bojom

