

Sweep line algoritam za presjek kružnica

Projekat iz Uvoda u kompjutersku geometriju

Problem: Zadano je n kružnica u ravni. Potrebno je u $O(n \log n)$ vremenu pronaći da li između njih postoji presjek.

Pomoćne funkcije i strukture:

- Kružnica

Struktura kruznica ima dva atributa: centar tipa Tacka, te radius kruga r . Sadrži metodu za iscrtavanje kružnice.

- daliSeSijekuKružnice(Kružnica*, Kružnica*)

Funkcija provjera za dvije proslijedene kružnice da li postoji presjek između njih. Radi na način da ako je udaljenost između centara dva kruga veća od zbira njihovih radiusa, tada se kružnice ne sijeku. Ukoliko je ta udaljenost manja ili jednaka zbiru dva radiusa r_1 i r_2 , tada se kružnice sijeku.

- KružniceKriterij

Ova struktura predstavlja kriterij pomoću kojega se porede polukružnice u stablu.

Bool operator ove strukture prima dva para: `pair<Tacka, pair<bool, Kružnica*>>` prva, `pair<Tacka, pair<bool, Kružnica*>>` druga. Prvi element para predstavlja tačku sa sljedećim koordinatama. X koordinatu ove tačke dobijemo kada od x koordinate centra oduzmemo radius kružnice. Y koordinata ove tačke je y koordinata centra kružnice. Drugi element para jeste par koji se sastoji od bool varijable i pokazivača na strukturu Kružnica. Bool varijabla nam govori da li se radi o gornjoj ili donjoj polukružnici. Gornja je označena sa 0, donja sa 1. Pokazivač na Kružnicu govori nam kojoj to kružnici data polukružnica odgovara (polukružnica označena sa bool varijablom kao što je već navedeno).

Dvije polukružnice se porede na sljedeći način:

Prvo se poredi da li dvije polukružnice pripadaju istoj kružnici. Ukoliko je to slučaj, u stablu prvo treba da ide gornja pa donja polukružnica. Ukoliko pripadaju različitim kružnicama vrši se sljedeća provjera.

Ukoliko je `prva.first.x > druga.first.x` tj. početak (ulazni event) prve polukružnice se nalazi iza početka (ulaznog eventa) druge polukružnice, tada je potrebno naći presjek vertikalne prave koja prolazi kroz `prva.first` i druge kružnice tj. `druga.second.second`. Oznacimo tu tacku sa t .

Nakon što je nađen presjek, provjerava se orijentacija (druga.first, t, first.first). Ukoliko je orijentacija pozitivna vraća se true (prva polukružnica je ispod (ne u matematičkom smislu) druge na sweep linu). U suprotnom vraća se false.

Ukoliko je $\text{prva.first.x} < \text{druga.first.x}$ tj. Početak prve polukružnice se nalazi iza početka druge, tada u gornjem postupku samo zamijenimo mjesta prvoj i drugoj polukružnici.

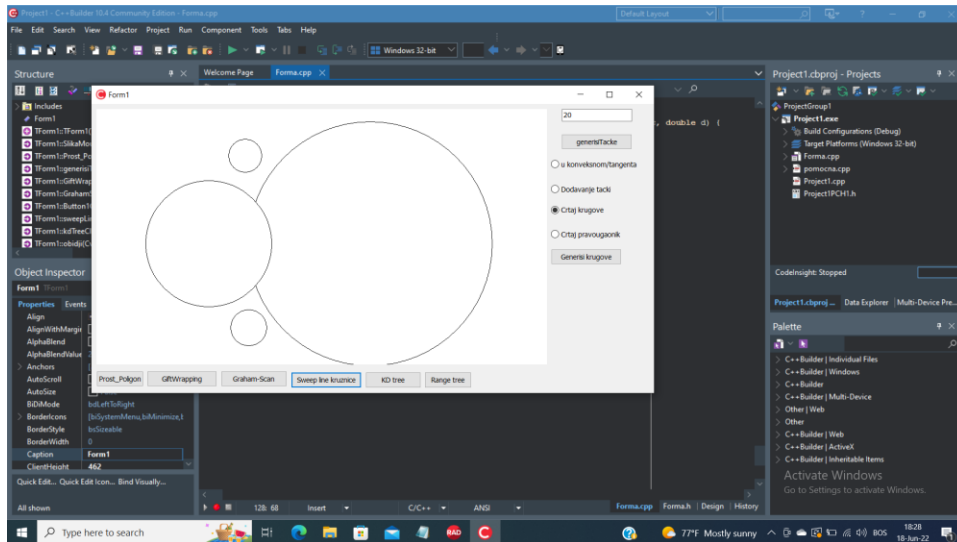
Prilikom traženja presjeka prave i kružnice, potrebno je voditi računa o tome sa kojom se polukružnicom vrši provjera, da li sa donjom ili gornjom polukružnicom. U zavisnosti od toga, uzima se određena tačka presjeka (jedna od dvije nađene).

Algoritam za kružnice sweep line:

Ukoliko je dato n kružnica, potrebno je prvo napraviti evente. Ti eventi se čuvaju u priority queue. Za n kružnica bit će $2n$ evenata. Dakle u red ubacujemo za jednu kružnicu dvije tačke. Tačku "početka kružnice" tj. Kada sweep line tek naiđe na kružnicu, te "krajnju tačku kružnice" tj. Kada sweep line "izađe" iz kružnice. Prioriteti ovih tačaka je x koordinata tačke (od manjih ka većim).

Prilikom prolaska kroz evente, potrebno ih je dodavati u stablo T ili izbacivati iz njega. Dodavanje se vrši prilikom nailaska na "početnu" tačku polukružnice, a izbacivanje nailaskom na "krajnju". U stablu T se nalaze samo one polukružnice koje sweep line trenutno siječe. Međutim, potrebno je napraviti kriterij na osnovu kojeg će biti sortirane u stablu.

Prva ideja mi je bila sljedeća: Stablo T je stablo parova $\text{pair} < \text{Tacka}, \text{Kružnica} >$ gdje Tacka ukoliko govorimo o gornjoj polukružnici ima koordinate ($\text{kružnica} \rightarrow \text{centar.x}, \text{kružnica} \rightarrow \text{centar.y} - \text{kružnica} \rightarrow r$), a ako je donja polukružnica ($\text{kružnica} \rightarrow \text{centar.x}, \text{kružnica} \rightarrow \text{centar.y} + \text{kružnica} \rightarrow r$). Kriterij ubacivanja u stablo mi je bio da se u stablo elementi dodavaju po y koordinati ove tačke. Međutim, ovaj kriterij nije dobar iz razloga što ne garantuje da će one koje se sijeku biti susjedne u nekome trenutku. Primjer toga dat je na slici ispod:



Druga ideja: Ispravan kriterij za ubacivanje u stablo T je kriterij naveden iznad tj. KružniceKriterij, koji u trenutku kada sweep line naiđe na novu polukružnicu, poredi da li je ta polukružnica iznad ili ispod drugih polukružnica na sweep linu. Ovaj kriterij garantuje da one koje se sijeku, u nekom trenutku moraju biti susjedni u stablu.

Dakle kada ubacujemo u stablo, vršimo provjeru da li se dodana polukružnica siječe sa prethodnom ili idućom u stablu. Takođe, prilikom izbacivanja se vrši provjera da li eventualni novi susjedi se sijeku. Ako je to slučaj ispisuje se poruka da postoji presjek i algoritam staje sa radom. Nakon što se algoritam završi, ukoliko nije detektovan presjek, ispisuje se poruka da nema presjeka među kružnicama.

Podjelu na polukružnice je bilo potrebno izvršiti iz razloga toga što na sweep linu prvo nam može doći gornja polukružnica kružnice A, pa npr. Gornja polukružnica B, pa donja polukružnica B, pa donja od A i sl. Odnosno, pri traženju presjeka u gore navedenom kriteriju, koju bi tačku presjeka prave i kružnice uzeli prilikom određivanja orijentacije (+ ili minus korijen, orijentacija ovisi od ovih vrijednosti). Npr. Ako bi uzeli + korijen, tada bismo imali informaciju da se naša kružnica nalazi iznad ili ispod donje polukružnice na sweep linu, ali ne znamo da li je iznad ili ispod gornje polukružnice i sl.