JAVA BIBLIOTEKA ZA RAD SA 2D RAČUNARSKOM GEOMETRIJOM
2D COMPUTATIONAL GEOMETRY JAVA LIBRARY

Darko Drakulić, Filozofski fakultet Univerziteta u Istočnom Sarajevu

Sadržaj – *U ovom radu su prikazani autorovi rezultati u razvoju Java biblioteke za rad sa dvodimenzionalnim geometrijskim objektima. Osim razvoja klasa za predstavljanje i manipulaciju sa tim objektima implementirani su i neki od osnovnih algoritama računarske geometrije kao i efikasni mehanizmi za vizuelizaciju ovih objekata kao i rezultata primjene algoritama na njima. Takođe, u ovom radu su prikazani i neki od osnove korišćenja ove biblioteke na primjeru Java apleta.*

Abstract – *In this paper it will be presented author's results in development Java library for work with 2D geometric objects. This work covers implementations of classes for work with geometric objects, some basic algorithms of computational geometry and efficient methods for visualization this objects and results of algorithms on those objects. In this paper will be presented some examples of using this library on Java applet.*

Ključne riječi: *Računarska geometrija, Java biblioteke, vizuelizacija geometrijskih objekata*

1. UVOD

Računarska geometrija je mlada grana računarske nauke koja je nastala u ranim sedamdesetim godinama prošlog vijeka kao rezultat pokušaja rješavanja geometrijskih problema pomoću računara. Od samog početka je povezivala različite oblasti nauke i tehnike kao što su teorija algoritama, strukture podataka, kombinatorna i euklidska geometrija, optimizacija, robotika i slično. U današnje vrijeme računarska geometrija ima veliku primjenu u GIS-u, CAD programima, vizuelizaciji i računarskoj grafici.

Algoritmi računarske geometrije kreću se od prostih – ispitivanja kolinearnosti tačaka, njihove međusobne pozicije, pa sve do vrlo složenih – pronalaženja najmanjeg konveksnog omotača skupa tačaka, generisanja Voronojevih dijagrama i sličnih. Da bi ti algoritmi radili efikasno, potrebne su pogodne strukture podataka za reprezentaciju tačaka, segmenata, skupa tačaka i ostalih geometrijskih objekata.

U ovom radu će biti prikazani rezultati u razvoju Java biblioteke za rad sa geometrijskim objektima koja se sastoji od klasa koje reprezentuju osnovne objekte a sadrži i neke od osnovnih algoritama računarske geometrije. Takođe, u biblioteci se mogu pronaći i neke klase koje ne spadaju u oblast računaske geometrije ali su korisne za vizuelizaciju geometrijskih objekata.

Iako računarska geometrija obuhvata i geometriju u prostorima dimenzija većih od dva, biblioteka o kojoj se govori u ovom radu podržava samo dvodimenzionalne objekte.

2. BIBLIOTEKA CGJ

Da bi se u Java kodu koristila biblioteka o kojoj se govori u ovom radu potrebno je u biblioteku fajlova dodati fajl `CGJ.jar` koji se može besplatno preuzeti sa adrese `http://www.darkodrakulic.com/cgj/` i u postojeći kod importovati prostor `com.darkodrakulic.cgj`. Takođe, na pomenutoj adresi se može pronaći i kompletna dokumentacija za sve klase ove biblioteke u `javadoc` formatu.

Klasa koja nije striktno vezana za računarsku geometriju a koja je implementirana u ovoj biblioteci je klasa `CGCanvas` koja je izvedena iz klase `java.awt.Canvas` i njena instanca predstavlja kontejner geometrijskih objekata ove biblioteke sa metodom

```
public void paint(java.awt.Graphics g)
```

koja predstavlja `override` metodu `paint` iz klase `java.awt.Canvas`. Ovo omogućava lak ispis svih objekata koji se nalaze u kontejneru, pozivom metode `repaint()`. Koordinatni sistem ove klase je uobičajeni koordinatni sistem koji se koristi u računarskoj grafici – gornja lijeva tačka ima koordinate (0,0) a pozitivna x-osa ima smjer na desno, a pozitivna y-osa na dole.

Za prikazivanje objekata iz ove biblioteke u nekom apletu potrebno je napraviti objekat klase `CGCanvas` i dodati ga apletu u metodi `init()` na sljedeći način

```
add(canvas, BorderLayout.CENTER);
```

Klasa `CGCanvas` sadrži i metodu

```
public void addObject(CGObject o)
```

koja dodaje objekat u kontejner i omogućava lako manipulisanje sa njima.

3. KLASE OSNOVNIH GEOMETRIJSKIH OBJEKATA

Osnovna klasa ove bibliteke je klasa `CGObject` koju nasljeđuju sve ostale klase koje predstavljaju geometrijske objekte. Klasa ima samo jednu apstraktnu metodu, metodu `abstract void draw(java.awt.Graphics2D g2D)` koja iscrtava objekat na `Graphics2D` uređaj. Ova metoda

omogućava da se svaki objekat može na lak način iscrtati na grafičkom uređaju.

Klasa za rad sa tačakama je klasa `CGPoint.` Atributi ove klase su dva cijela broja koja predstavljaju poziciju tačke na objektu `CGCanvas`. Ovo ne spriječava kreiranje tačaka sa realnim koordinatama –koordinate tačke nisu koordinate u nekom određenom koordinatnom sistemu već koordinate tačke na `GCCanvas` objektu. Za prikaz tačaka u koordinatnom sistemu dugačijem od koordinatnog sistema `GCCanvas` objekta potrebno je definisati preslikavanje koje slika tačke sa potrebnog koordinatnog sisitema na sistem `GCCanvas` objekta. U narednoj verziji ove Java biblioteke ovaj problem će biti rješen na drugačiji način, preko koordinata tačaka u zadanom koordinatnom sistemu.

Ostale klase osnovnih geometrijskih objekata su:
`CGPointSet` – klasa za rad sa skupovima tačaka
`CGSegment` – klasa za rad sa dužima
`CGSegmentSet` – klasa za rad sa skupovima duži
`CGPolygon` – klasa za rad sa prostim poligonima

Detaljan opis ovih klasa se može pronaći u detaljnoj dokumentaciji na pomenutoj web lokaciji.

4. METODE ČLANICE IMPLEMENTIRANIH KLASA

Algoritmi računarske geometrije se kreću od vrlo prostih pa sve do vrlo zahtjevnih, a u biblioteci o kojoj se govori u ovom radu su do sada implementirani neki od osnovnih algoritama. Najprostiji algoritmi će biti detaljnije opisani dok će za one složenije biti dat samo opis za njihovo korišnjenje, sa njihovom kompleksnošću. Većina algoritama je implementirano na osnovu rezultata i opisa iz [1].

4.1. CGPoint

Najprostiji algoritmi su algoritmi nad klasom tačaka. Klasa `CGPoint` osim metode za iscrtavanje sadrži i metode aksesore, metode za manipulisanje sa koordinatama, oznakom i bojom tačke kao i metode za pomjeranje tačke.

Da bi na `CGCanvas` objekat dodali tačku na lokaciji (100, 100) crne boje sa labelom „A“ potrebno je izvršiti sljedeće linije koda:

```
CGPoint  point = new CGPoint();
point.setLocation(100, 100);
point.setColor(Color.black);
point.setLabel("A");
canvas.addObject(point);
```

Za iscrtavanje ove tačke potrebno je pozvati metodu
`canvas.repaint();`

Najprostiji algoritmi za rad sa tačkama su algoritmi za određivanje međusobnog položaja tri tačke. Neka je potrebno odrediti da li se tačka C nalazi sa lijeve ili desne strane (usmjerene) duži AB ili se možda nalazi na njoj. Iako se ovaj problem čini lagan, u implementaciji se može naići na određene probleme pri određivanju položaja te tačke. Ovaj problem se najefikasnije rješava primjenom vektorskog proizvoda. Posmatrajmo vektore AB i AC. Njihov vektorski proizvod predstavlja površinu paralelograma konstruisanog nad njima. Upravo ova površina omogućava da se odredi položaj tačke C u odnosu na duž AB: ako je površina trougla ABC pozitivna onda su tačke A-B-C u smjeru obrnutom od kazaljke na satu a to implicira da je tačka C sa lijeve strane duži AB; ukoliko je negativna onda se tačka C nalazi sa desne strane a ako je površina jednaka nuli ona su tačke A, B i C kolinearne. Metode koje ispituju ove osobine tačke se nalaze u klasi `CGPoint` i imaju sljedeće deklaracije:

```
boolean left(CGPoint a, CGPoint b)
boolean leftOn(CGPoint a, CGPoint b)
boolean right(CGPoint a, CGPoint b)
boolean rightOn(CGPoint a, CGPoint b)
boolean collinear(CGPoint a, CGPoint b)
boolean between(CGPoint a, CGPoint b)
```

Svi prikazani algoritmi imaju linearno vrijeme izvršavanja.

4.2. CGUtility

Metoda koja izračunava površinu trorugla ABC je jedna (za sada i jedina) statična metoda koja se nalazi u klasi `CGUtility` koja je rezervisana za sve pomoćne funkcije one biblioteke.

4.3. CGSegment

Klasa CGSegment sadrži standarne metode (aksesori, dodavanje i uklanjanje tačaka, manipulacija bojom i oznakom tjemena) ali i jednu klasičnu metodu računarske geometrije, metodu koja ispituje da li se dva segmenta sijeku:

```
boolean intersect(CGSegment l)
```

Iako se čini da je presjek segmenata lako pronaći rješavanjem sistema jednačina koje predstavljaju jednačine pravih na kojima leže ovi segmenti, to nije tako. Osim naknadne provjere da li presječna tačka leži na segmentima, postoje i dvije opasne zamke – ukoliko su segmenti paralelni ili ukoliko je jedan segment paralelan y-osi. U oba slučaja se javlja dijeljenje s nulom koje može dovesti do problema u radu programa. Osim toga, izračunavanje tačke presjeka ovom metodom može dovesti do grešaka u izračunavanjima pri zaokruživanju realnih brojeva. Ovaj problem se može efikasno rješiti korišćenjem metoda za ispitivanje položaja tačke u odnosu na segment, za svaki par segmenata ispitamo da li se nalazi s raznih strana drugog segmenta, korišćenjem metoda iz klase `CGPoint`.

4.4. CGPolygon

Klasa `CGPolygon` sadrži metode aksesore, metode za dodavanje i uklanjanje tjemena poligona (pri dodavanju se provjerava da li poligon ostaje prost – da ne postoje dvije stranice koje se sijeku), za manipulaciju tjemenima, zatvaranje poligona (dodavanje stranice između prvog i posljednjeg tjemena) kao i dva algoritma računarske geometrije – izračunavanje površine poligona i metodu za triangulaciju poligona:

```
double computeArea()
CGSegmentSet triangulate()
```

Površina poligona se izračunava u linearnom vremenu preko diskretne verzije Grinove teoreme [1, str. 21]. Trinagulacija se izvršava za vrijeme $O(n^2)$ korišćenjem algoritma opisanog u [1, str. 38]

4.5. CGPointSet

Klasa `CGPointSet` sadrži metode aksesore, metode za dodavanje i uklanjanje tačaka, metode za manipulisanje bojama i oznakama tačaka kao i jedan od poznatijih algoritama računarske geometrije – algoritam za pronalaženje najmanjeg konveksnog omotača datih tačaka. Metoda ima sljedeći izgled:

```
CGPolygon convexHull()
```

Kao što se iz deklaracije vidi, rezultat ove metode je objekat klase `CGPolygon` a za implementaciju se koristi Grahamov algoritam [1, str. 72] kompleksnosti O(*n*lg*n*).

5. PRAVCI DALJEG RAZVOJA

U opisu implementiranih klasa je bilo riječi o nekim njihovim nedostacima. Jedan od najvećih nedostataka je korišćenje koordinatnog sistema sa cjelobrojnim, pozitivnim koordinatama što bi u sljedećoj implementaciji trebalo biti prevaziđeno. Pored uvođenja proizvoljnog koordinatnog sistema planira se podrška za rad u nekim drugim koordinatnim sistemima osim Dekartovog, prije svega podrška za rad sa baricentričnim koodrinatama. Osim toga, planira se implementacija još nekih algoritama računarske geometrije (monotona triangulacija poligona i ostali algoritmi problema umjetničke galerije, uvođenje Voronojevih dijagrama, algoritmi pronalaska najkraćih puteva itd.).

Osim problema računarske geometrije, planira se i poboljšanje klase `CGCanvas` i metoda `draw` svake od klasa objekata takvo da one ne koriste metode za iscrtavanje primitiva klase `Graphics2D`. Umjesto toga planira se razvoj sopstvenih metoda za iscrtavanje objekata.

Nakon toga bi se otvarili preduslovi za uvođenje klasa za rad sa krivima koje se mogu zadavati parametarski i algebarski.

Kao kruna razvoja ove biblioteke planira se razvoj funkcionalnog softvera za geometrijske konstrukcije u ravni sa akcentom na geometriju trougla korišćenjem klasa iz biblioteke CGJ.

LITERATURA

[1] J. O'Rourke, *Computational Geometry in C*, Cambridge University Press 1997.

[2] J. O'Rourke, *Art Gallery Theorems and Algorithms,* Oxford University Press 1987.

[3] M. de Berg, O. Cheong, M. van Kreveld, M. Overmars, *Computational Geometry Algorithms and Application, Third edition,* Springer 2008.

[4] J. R. Sack, J. Urrutia, *Handbook of Computational Geometry,* Elsevier 2000.