

## Zadatak 1

Doradite Vaš program za crtanje linija i krugova (zadaca broj 16) tako da dodate podršku za crtanje popunjenih **konveksnih** poligona (test za konveksnost pojašnjen je u nastavku). Minimalni broj točaka poligona je 3. Točke poligona definiraju se klikanjem lijevog gumba miša. Na prvi klik miša definira se prva točka. Potom, kako korisnik pomiče miš, automatski se iscrtava linija između zapamćene točke i kazaljke miša. Na drugi klik miša, definira se druga točka. Potom, kako korisnik pomiče miš, automatski se iscrtava poligon definiran sa zapamćene dvije točke te vrhom kazaljke miša kao trećom točkom. Na treći klik miša, definira se treća točka. Potom, kako korisnik pomiče miš, automatski se iscrtava poligon definiran sa zapamćene tri točke te vrhom kazaljke miša kao četvrtom točkom. Potom, ... (shvaćate ideju). Definiranje poligona je gotovo kada korisnik klikne mišem dovoljno blizu posljednjoj definiranoj točki (dovoljno blizu ovdje znači da je Euklidska udaljenost kliknute pozicije i posljednje dodane točke manja ili jednaka 3). Međutim, na ovaj se način ne može prekinuti definiranje poligona ako broj dodanih točaka nije barem 3: u tom slučaju takav klik naprosto ignorirajte. Također, pri pokušaju dodavanja četvrte, pete, ... točke, prije no što se točka prihvati i doda kao novi vrh poligona potrebno je napraviti provjeru hoće li takav poligon biti konveksan (test je opisan u nastavku); tek ako test pokaže da će poligon i dalje biti konveksan, dodat ćemo točku kao novi vrh; inače pokušaj dodavanja ignoriramo i korisniku u prikladnom dijalogu dojavljujemo da se točka ne prihvaća kao vrh poligona, jer isti ne bi bio konveksan. Nakon ove dojave, korisnik kazaljku miša može pomaknuti na neki novi položaj pa točku ponovno pokušati dodati kao novi vrh poligona.

Ako korisnik napravi klik desnog gumba miša, odustaje se od definiranja trenutnog poligona (kao da još ništa nije započeto).

Konveksnost poligona morate ispitati na sljedeći način (test radimo samo kada je  $k=4$  ili veći). Neka je za poligon već definiran ( $k-1$ ) vrh te neka položaj kazaljke miša predstavlja kandidata za  $k$ -ti vrh za koji razmatramo hoćemo li ga prihvatiti ili ne smijemo dopustiti da on stvarno postane  $k$ -ti vrh. Dajmo tim vrhovima oznake  $V_1, V_2, \dots, V_k$  (svaki vrh je točka u 2D određena svojim  $(x,y)$  koordinatama). Pretpostavite sada da se zapravo radi o točkama u 3D prostoru te da sve točke imaju z-koordinatu postavljenu na 0. Razmotrimo sada parove vektora  $R_{i,j} = V_j - V_i$  gdje je  $j$  najprije  $i+1$  a potom  $i+2$ . Dakle, razmatrat ćemo sljedeće parove vektora:  $(R_{1,2}, R_{1,3}), (R_{2,3}, R_{2,4}), (R_{3,4}, R_{3,5}), \dots, (R_{k-1,k}, R_{k-1,k+1})$ . Pazite kod izračuna: poligon je zatvoren pa u izrazu gdje biste dobili primjerice indeks  $k+1$ , zapravo se ponovno radi o indeksu 1 (skicirajte si primjer konveksnog poligona pa pogledajte koje vektore računamo). Za svaki par vektora  $(R_{i,i+1}, R_{i,i+2})$  odredite njihov **vektorski produkt** (pri tome uvijek množite prvi vektor para drugim vektorom para – poredak je bitan). Kako su vektori  $R_{i,i+1}$  i  $R_{i,i+2}$  oba 3D, njihov vektorski produkt također će biti vektor u 3D. Zapamtite predznak z-komponente tako izračunatog vektorskog produkta (predznaci x- i y-komponenta nam nisu bitni). Ako su predznaci z-komponenta svih  $k$  izračunatih vektorskih produkata jednaki, poligon je konveksan pa ćemo dopustiti da vrh miša doista bude i prihvaćen kao  $k$ -ti vrh. U suprotnom, poligon nije konveksan i ne dopuštamo da se vrh miša prihvati kao novi vrh poligona. Ako bi z-koordinata bila 0, nemojte nulu tretirati niti kao pozitivnu, niti kao negativnu već je naprosto preskočite u provjerama.

Vaš program mora podržati dodavanje popunjenih poligona (zasebnom bojom prikazuje se obrub a zasebnom površina), snimanje i učitavanje ovih likova kao i export u PNG (sve ovo zajedno s prethodno već postojećim linijama, kružnicama i obojanim krugovima - sve mora raditi). Također, Vaš program mora podržati uređivanje (edit, na dvoklik u listi dodanih objekata). Pri tome uređivač ne treba podržavati dodavanje ili brisanje vrhova već samo uređivanje njihovih x i y koordinata; no, uređivač ne smije dopustiti da se prekrši ograničenje da dva uzastopna vrha budu na manjoj ili jednakoj udaljenosti od 3 ili pak da modificirani poligon više ne bude konveksan.

Za crtanje poligona i obruba koristite ugrađene metoda razreda `Graphics` koji smo i inače koristili za crtanje linija i kružnica/krugova.

Sintaksa u datoteci \*.jvd za popunjene poligone mora biti oblika:

```
FPOLY 3 10 20 30 40 50 60 255 0 0 0 255 255
```

pri čemu prvi broj predstavlja broj vrhova poligona (u primjeru: 3), potom slijedi toliko x y parova koordinata vrhova (u primjeru, vrhovi su na položajima (10,20), (30,40) i (50,60)). Na kraju slijede RGB komponente boje obruba (tri broja, u primjeru 255 0 0), te RGB komponente boje ispune (tri broja, u primjeru 0 255 255).

Popis stavki za provjeriti (pomoć da vidite jeste li implementirali relevantne stvari):

1. u korisničko sučelje dodan je gumb za definiranje popunjenih konveksnih poligona
2. vrhovi poligona dodaju se na lijevi klik miša
3. pomicanjem kazaljke miša nakon prvog klika na dalje, na slici se dinamički docrtava kako će izgledati definirani lik
4. broj vrhova konveksnog poligona može biti proizvoljan (ali veći ili jednak 3)
5. dva uzastopna vrha ne mogu biti na manjoj ili jednakoj udaljenosti od 3
  - pazite da provjerite i udaljenost posljednjeg vrha od prvog vrha!
6. ne dopušta se definiranje lika koji sadrži manje od 3 vrha (odnosno, takav lik ne može biti dodan u model dokumenta)
7. ne dopušta se definiranje poligona koji nije konveksan
8. preko liste dodanih objekata moguće je uređivati i popunjene poligone
  1. ne dopušta se prihvaćanje uređenog poligona koji sadrži dva uzastopna vrha na udaljenosti manjoj ili jednakoj 3
  2. ne dopušta se prihvaćanje uređenog poligona koji više nije konveksan
9. slika koja sadrži linije, kružnice, popunjene krugove i popunjene poligone uredno se može:
  1. snimati u \*.jvd datoteku
  2. učitati iz \*.jvd datoteke
  3. exportati u PNG datoteku

## Zadatak 2

Ovaj zadatak nastavlja se na prethodni. Napravite web-aplikaciju kojoj se pristupa na:

<http://localhost:8080/jvddrawer/index.html>

Aplikacija se sastoji od stranice `index.html` koja prikazuje formular u koji je moguće zalijepiti (engl. *paste*) sadržaj neke `*.jvd` datoteke te jednog servleta mapiranog na `/crtaj`. Sadržaj formulara se šalje servletu metodom POST. Ako je poslani sadržaj ispravan `*.jvd`, servlet dinamički generira i pozivatelju vraća PNG-sliku (identičnu onoj koja bi nastala exportom u domaćoj zadaći). Ako je sadržaj neispravan, servlet treba vratiti HTML-sadržaj naslova “Pogreška” koji prikazuje poruku “Neispravan format primljenih podataka.”

Web-aplikaciju možete demonstrirati u *Jetty*-ju.

Popis stavki za provjeriti (pomoć da vidite jeste li implementirali relevantne stvari):

1. Stranica `index.html` sadrži jedan formular u koji je moguće iskopirati sadržaj `*.jvd` datoteke
2. Formular se šalje metodom POST servletu
3. Servlet pozivatelju vraća PNG sliku nastalu temeljem crtanja primljenih podataka, ako nema pogreške
4. Servlet pozivatelju prikazuje HTML dokument s porukom pogreške, ako je došlo do pogreške

---

Nakon što ste završili projekt/projekte, obavezno ga/ih zapakirajte u jednu ZIP-arhivu, uploadajte na Ferka pod 1. laboratorijsku vježbu i upload zaključajte. Ako do predviđenog vremena niste sve riješili, projekt trebate uploadati prije no što izađete iz prostorije. Ako ste do predviđenog vremena riješili sve zadano, po isteku vremena dežurni će Vas asistent udaljiti od računala; nakon što pred asistentom demonstrirate napravljeno i odgovorite na postavljena pitanja, napraviti ćete arhivu i uploadati je na Ferka.