# Drugi projekt pri predmetu Matematično modeliranje

# Bézierovi zlepki

## Glavna ideja:

Konstruirali bi radi gladek zlepek sestavljen iz Bézierovih krivulj, ki gre

skozi dane točke v ravnini. Za izbrani zlepek izračunamo še dolžino

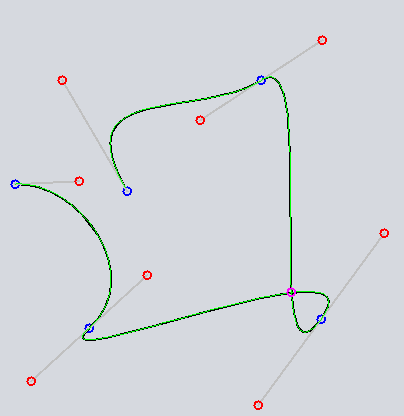
zlepka in morebitna samopresečišča.

## Naloga:

V okviru naloge je potrebno konstruirati zlepek sestavljen iz Bézierovih krivulj

iste stopnje, ki je načeloma majhna (med 2 in 5 – v našem primeru 3). Naloge so:

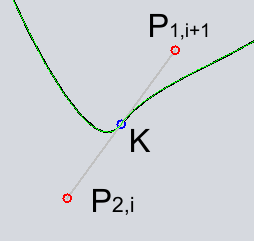
1. Izpeljati, katerim pogojem morajo zadoščati kontrolni poligoni krivulj v zlepku, da bo zlepek zvezno odvedljiv.
2. Napisati pomožne funkcije za računanje točk in izris Bézierove krivulje s podanim poligonom. Izračun naj bo izveden z De Casteljauovem algoritmom.
3. Za dane točke v ravnini poiskati zvezno odvedljiv zlepek sestavljen iz Bézierovih krivulj iste stopnje.



Slika : Bezierov zlepek

Izpeljava:

* za lepljenje zlepkov s pomočjo 1. odvoda izračunamo mesto 1. kontrolne točke krivulje, ki jo zlepku priključujemo
* ko izpeljemo 1. kontrolno točko naslednje krivulje, opazimo, da je to le preslikava 2. kontrolne točke iz trenutne krivulje čez točko K (zadnjo točko trenutne in prvo naslednje krivulje)
* za določanje 2. kontrolne točke, ki je v našem programu še prosta, se uporablja izpeljava iz 2. odvoda
* iz zgornjih postopkov dobimo zvezno odvedljiv zlepek

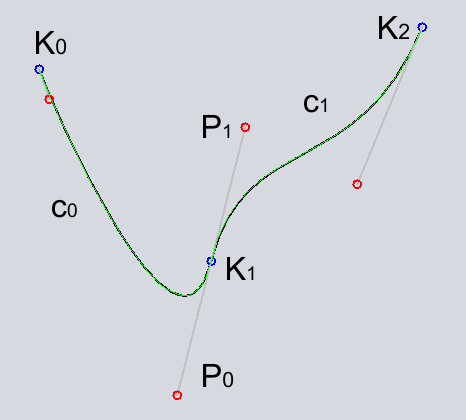


Slika : Položaj kontrolnih točk

## Postopek:

Uporabili smo programski jezik Java, ki ima soliden vmesnik za izrisovanje in interaktivno dodajanje točk, daljic, krivulj…

* Grafični vmesnik programa uporablja Java Swing – s pomočjo vgrajenih funkcij izrisujemo kontrolne točke in daljice, krivulje in točke skozi katere poteka krivulja
* Dodatno uporabljamo De Casteljauov algoritem za izris krivulj (zelene črte), ter za izračun dolžine krivulje in samopresečišč
* V programu s kliki v ravnini določamo skozi katere točke (Kx) naj poteka Bezierjev zlepek
* Program določi usmerjenosti kontrolnih točk (Px) in potem izriše krivuljo med točkama (cx), glede na kontrolni poligon, določen s točkami
* Pri vstavljanju nadaljnjih točk Kx, program določi lego pripadajoče kontrolne točke po zgodaj dokazani enačbi, kar pomeni da morata biti daljici, ki se dotikata iste podane točke, na isti premici
* Program omogoča interaktivno prestavljanje kontrolnih točk, in s tem tudi spreminjanje oblike samega zlepka
* Pri vsaki postavitvi nove točke, ali pri spreminjanju kontrolnega poligona, program izračuna in izpiše skupno dolžino zlepka in število samopresečišč



Slika : Primer krivulje

Legenda:

* Kx - podana točka
* cx – krivulja, ki povezuje točki Kx in Kx+1
* Px – izračunana kontrolna točka, ki določa obliko krivulje

Ugotovitve:

1. Pri sosednjih kontrolnih poligonih (2 podani točki in pripadajoči kontrolni točki) morata biti stranici poligonov, ki imata skupno podani točko, na isti premici (Slika 3: daljici P0K1 in K1P1 morata biti na isti premici)
2. Funkcija v Java Swing pri izrisu krivulj vrača primerljive krivulje kot De Casteljauov algoritem (na slikah: črna krivulja je izrisana z vgrajenimi funkcijami, zelena pa z De Casteljauovem algoritmom)
3. De Casteljauov algoritem se uporablja za izračun dolžine zlepka in za šteje samopresešišč, zadnje program tudi označi z točko

## Razporeditev dela:

* Juš Debelak: risanje podatkov, dodajanje in spreminjanje krivulj, izpeljava zveznosti zlepkov, poročilo
* Jure Jesenšek: uporabniški vmesnik, pomoč pri programiranju, poročilo
* Egidij Egej Vencelj: De Casteljauvov algoritem, računanje dolžine, iskanje samopresečišč
* Tilen Fišer: predstavitev

Viri:

* Java dokumentacija
* <http://en.wikipedia.org/wiki/B%C3%A9zier_curve>
* <http://en.wikipedia.org/wiki/De_Casteljau%27s_algorithm>
* <https://www.particleincell.com/2012/bezier-splines/>