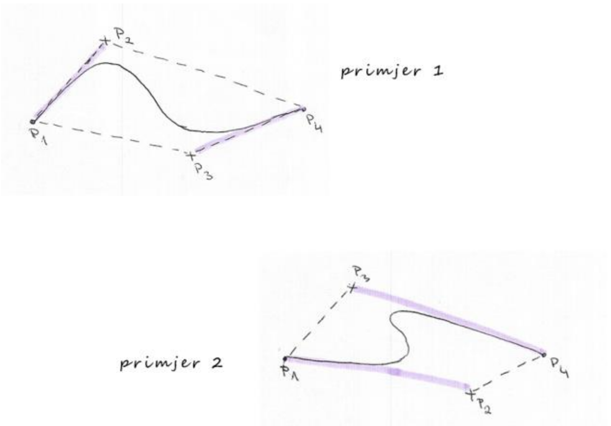
Osvrt na drugo predavanje – Bezier krivulja

U ovome predavanju ćemo se malo bolje upoznati sa Bezierovom krivuljom te izvesti njezinu matematičku formulu kako bi došli do korijena nastanka krivulje u vektorskim programima. Prvo ćemo definirati Bezier krivulju. To je glavna krivulja današnje vektorske grafike koju koristimo za izradu slovnih znakova u bilo kojem programu za izradu fonta. Bezierova krivulja je parametarska krivulja trećeg stupnja koja pripada obitelji krivulja u vektorskoj grafici koja se naziva predvidljive krivulje. Kao što nam sam naziv sugerira, to su krivulje čiji izgled možemo unaprijed predvidjeti i upravo zato se Bezierova krivulja najviše koristi u vektorskoj grafici. Ova krivulja je isto tako važan element bilo kojeg drugog programa vektorske grafike od kojih je najpoznatiji Illustrator. Njezina glavna karakteristika je ta što na temelju postavljanja četiri točke možemo unaprijed predvidjeti rasprostiranje krivulje na površini. Sa samo 4 točke cijela krivulja dobiva svoju punu funkcionalnost.

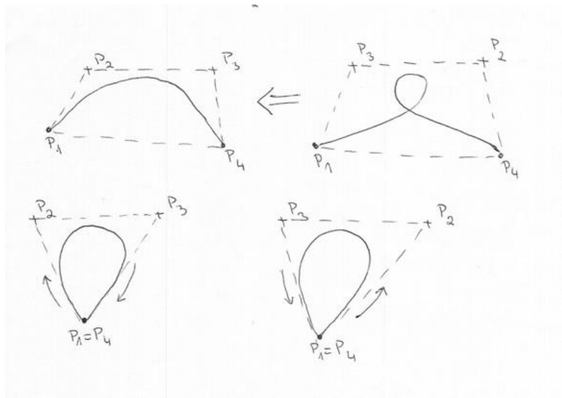
Da bi bolje objasnili pojam krivulje definirane četirima točkama nacrtat ćemo te četiri točke i spojiti dužine između točaka. Dobivamo jedan poligon koji označava prostor unutar kojega moramo nacrtati krivulju. Između točaka P_1 i P_2 te točaka P_3 i P_4 postoji matematička veza. Dužina P_1P_2 je dužina tangente koja nam govori kako krivulja mora ući u točku P_1 , a ista je stvar i sa dužinom P_3P_4 i točkom P_4. Ako imamo istu konstelaciju točaka, ali su točke pre indeksirane krivulja će se potpuno drugačije rasprostrijeti jer su tangente P_1P_2 i P_3P_4 drugačije poredane. U prvom primjeru krivulja izgled kao sinusoida, a u drugom primjeru krivulja izgleda kao točka infleksije(Slika 1).



Iz toga proizlazi zakonitost krivulje koja glasi: ,,Tijelo krivulje uvijek će se rasprostrijeti unutar konveksnog poligona 4 označene točke i to na način da će dužina P_1P_2 činiti tangentu na točku P_1 krivulje, a dužina P_3P_4 činiti tangentu na točku P_4 krivulje s tim da krivuljom ne smijemo izaći iz konveksnog poligona."

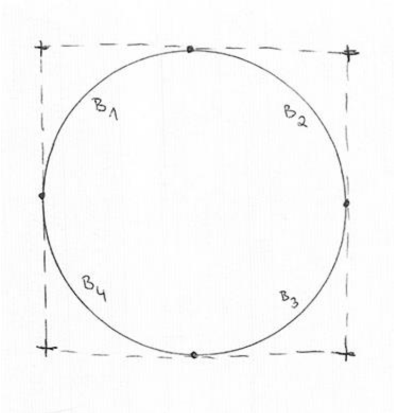
Slika 1

Prema tome možemo zaključiti da ako imamo istu konstelaciju točaka mijenjanjem indeksa točaka njena se i izgled krivulje. Zamjenom točaka P_2 i P_3 dobijemo petlju. Kada je P_1=P_4 dobijemo oblik kapljice, ako zamijenimo točku P_2 i P_3 dobijemo istu krivulju samo je njezin tok u suprotnom smjeru(Slika 2). Indeksacija točaka jako bitna jer utječe na sam tijek, tok i izgled krivulje. Pošto se indeksacija u programima ne koristi unaprijed moramo znati kako će nam krivulja izgledati.

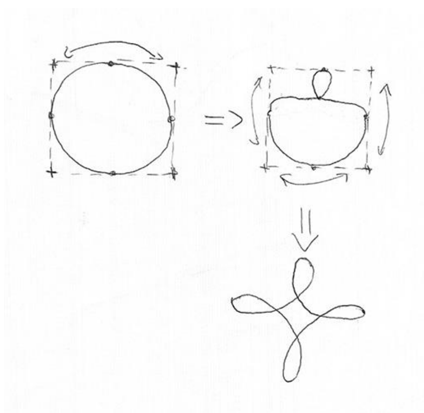


Slika 2

Želimo li dobiti ravninu pomoću Bezierove krivlje točke P_3 i P_4 moraju biti bilo gdje na ravnini između točaka P_1 i P_4. Inicijalno, ako želimo dobiti ravninu, u svim programima se to radi na način da P_2 stavimo na isto mjesto gdje i P_1, a P_3 gdje i P_4. Kružnicu dobivamo tako da napravimo 4 Bezierove krivulje s time da su natezne točke postavljene tako da zatvaraju kvadrat (Slika 4). Zamjenom mjesta nateznih točaka krivlje mijenjaju oblik te gubimo kružnicu (Slika 5).



Slika 4



Slika 5

MATEMATIČKI IZVOD BEZIER KRIVULJE

Nacrtamo li krivulju u koordinatnom sistemu svaka točka ima svoju x i y koordinatu. Iz tih koordinata dobivamo matematički izvod Bezierove krivulje. Za prikaz Bezierove krivulje trebamo 8 brojeva zato što svaku točku označavamo sa 2 broja (x i y koordinata).

Bitno svojstvo kod matematičkog izvoda Bezierove krivulje je to što je ona parametarska krivulja jer se takva krivulja lakše programira. Parametre stavimo u jednu for petlju i s tim parametrom rađamo x i y koordinate svake od točaka da bi se na kraju krivulja vidjela. Pri učenju parametarska krivulja se prvo piše u jednoj dimenziji pa se kasnije može proširiti na dvije dimenzije ili čak na tri dimenzije ako želimo raditi trodimenzionalne Bezierove plohe.

Postupak izvođenja matematičke formule počinjemo definiranjem krivulje u jednoj dimenziji. Tada krivulju označavamo sa c, a parametar se označava sa t. Zapisujemo je u matričnoj formi jer je tako kraća i lakše se pamti. Prvo pišemo jedan vektor zatim matrično množenje sa B (Bezierova matrica) i na kraju imamo četiri točke definirane u jednoj dimenziji (Slika 6).

$$c(t) = \underbrace{\begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix}}_{1\times4} \cdot B \cdot \underbrace{\begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix}}_{4\times1}$$

$$B = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Slika 6

U prvom dijelu formule imamo 1 redak puta 4 stupca, a na kraju formule imamo 1 stupac puta 4 retka što znači da će Bezierova matrica biti 4 retka * 4 stupca (Slika 6). Cijela tajna Bezierove krivulje se nalazi u brojevima matrice. Svojstvo Bezierove matrice je da je suma brojeva u 1.retku ista kao suma brojeva u 2. i 3. retku, a to je nula. Dok je suma brojeva u 4. retku jednaka broju 1. Isto tako je suma brojeva prvog, drugog i trećeg stupca jednaka broju jedan, a suma brojeva četvrtog stupca je jednaka nuli.

Kako bi krivulju prikazati u dvije dimenzije, da se može nacrtati, napisati ćemo formulu za x(t), a zatim za y(t). Te dvije formule će biti koordinate točaka krivulje(Slika 7).

$$x(t) = (-t^3 + 3t^2 - 3t + 1) \cdot P_1^x + (3t^3 - 6t^2 + 3t) \cdot P_2^x + (-3t^3 + 3t^2) \cdot P_3^x + t^3 \cdot P_4^x$$

$$y(t) = (-t^3 + 3t^2 + 3t + 1) \cdot P_1^y + (3t^3 - 6t^2 + 3t)\, P_2^y + (-3t^3 + 3t^2) \cdot P_3^y + t^3 \cdot P_4^y$$

Slika 7

Za primjer jednadžbe stavili smo da je t parametar 0 u x(t). Uvrštavanjem u jednadžbu dobijemo rješenje P_1^x .Kada uvrstimo nulu u y(t) dobijemo rješenje P1y. tako dobijemo koordinate prve točke. Ako za t uzmemo 1 kao rješenje za x(t) ćemo dobiti p4x, a za y(t) i p4y. To su koordinate zadnje točke. Iz toga zaključujemo da se sve točke koje čine krivulju crtaju sa parametrima t koji se nalaze u intervalu od 0 do 1. Zato uz formulu moramo dodati uvjet da je domena rada Bezierove krivulje $t \in [0,1]$.

Niti jedan ekran i printer ne crta krivulju u jednom potezu kao što je mi crtamo rukom već oni crtaju krivulju pomoću mnogo gusti točkica. Toliko gustih da je međuprostor okom nevidljiv. Zato se pojavljuje pitanje s kojom gustoćom, to jest rezolucijom, moramo prevariti ljudsko oko da ono ne vidi taj međuprostor. Taj podatak ovisi i o rezoluciji ekrana ili nekog ispisnog uređaja. Na primjer, printer rezolucije 600dpi točkice najbliže može prikazati na 600tom dijelu inča.

Kako bi matematički definirali broj točkica koje moramo ispisati uvodimo vrijednost Δt. Δt je u funkcionalnoj vezi rezolucije tehnologije na kojoj krivulju želimo pokazat. Ako je Δt = 0.1 onda je t=0, t_1=0.1, t_2=0.2, …, t_{10}= 1.0 što znači da krivulja ima 11 točaka kada je Δt=0.1. Ukoliko je Δt=0.01 tada će krivulja imati 101 točka, a kada je Δt=0.001 tada imamo 1001 točku. Iz tih podataka možemo vidjeti da broj točaka na krivulji, ako imamo zadani t, definiramo formulom: 1/Δt+1

SPOJNE BEZIER TOČKE

Postoje 3 vrste spojnih Bezier točaka:

1. **KUTNI SPOJ**

Kutni spoj se u softveru označava se kvadratićem □ . Njime spajamo ulazni spoj B1 i izlazni spoj B2. Ulazni i izlazni spoj znamo ako definiramo orijentaciju krivulje. Točke kojima mijenjamo oblik krivulje zovemo Bezier Control Point, skraćeno BCP. Ako mičemo mišem jednu BCP lijevo-desno ili naprijed-natrag to neće utjecati na drugu BCP. Tako dolazimo do zaključka da nezavisnosti jedne BCP nije u nikakvoj funkcijskoj vezi sa drugom BCP, tj. Da redizajn izlaznoj spoja neće utjecati na redizajn ulaznog spoja.

2. KRIVULJNI SPOJ

U krivuljnom spoji pomicanjem ulazne BCP za neki kut alfa za toliki kut ali u suprotnom smjeru se pomiče i izlazna BCP. Izlazna BCP je u funkciji pravca točaka BCPul. i same spojne točke. Svaki pomak održava pravac koji je na početku definiran.

3. TANGENTNI SPOJ

U softverima se najčešće označava sa trokutićem Δ. Riješava problem kako napraviti idealni zavoj iz smjera gore prema dolje u smjer lijevo prema desno. Kako bi dobili idealan zavoj stavljamo trokute na pravac svakog od tih smjerova, na tangenti te krivulje stvaraju se + koji omogućavaju stvaranje idealnog zavoja te sprječavaju da napravimo neki krivi potez ili infleksiju. Micanjem + možemo manipuliari izgled tog idealan zavoj. Ova tehnika najčešće se koristi kada se definiraju serifi slovnih znakova.