

## Osvrt na predavanje

### Važnost IKT u metodici nastave na grafičkom fakultetu

FONTOVI (pisma ili tekst),eng. GIF:

Font je nakupina kodnih pozicija, a na svakoj kodnoj poziciji se nalazi neka slika. Program za izradu fontova je Fontografer.

Po ASKI standardu svaka slika ili slovo je označeno sa određenim brojem na tom kodnom mjestu. Prije se to nije moglo raditi jer se sve to izljevalo s olovom, ali danas je to moguće.

Svako kodno mjesto ima svoj koordinatni sustav, kada odaberemo jedno slovo ili kodno mjesto otvori se koordinatni sustav u kojem su naznačena 5 pravca. Svi ti pravci zajedno ostvaruju digitalni četverac i njime možemo ostaviti znakove iznad slova kao naprimjer za slovo č.

Dizajnirani font se može provjeriti u Photoshopu ili Illustratoru, ali da lakše provjerimo ispravnost veličine fonta koristimo prozor za metriku fonta i s njime možemo vidjeti kako oni izgledaju jedan pored drugoga. Kada imamo veće praznine između slova onda se desna linija uz prvo slovo pomiče tako da se poklapa s drugim slovom uz njega. Ponekad radimo i neke iznimke, a to su parovi podrezivanja. U fontu ne postoji niti jedna od mjernih jedinica kao što su milimetri, centimetri ili inči nego relativna jedinica jer tekst softver skalira na digitalnu verziju.

Postoje programi da iščitavaju parove podrezivanja, a ne da se mora selektirati približavanje dva slovna znaka. Ako želimo dizajnirati neki novi slovni znak ili slovo, npr. slovo č, selektiramo slovo c i kopiramo ga i postavimo gdje želimo postaviti to slovo i onda dodamo gore kvačicu. Postoji puno vrsta parametara, jedan od parametara s kojima se simulira font je parametar rezolucije. Student sam mijenjarezoluciju dali želi nižu ili višu rezoluciju. Ovisno kako tko želi on može prilagoditi rezoluciju preko programa, dali to bila boja, spirala ili veličina, tj. visina. Svaka varijabla ima svoj kod i svaka utječe na nešto. U interface-u imamo sve varijable koje program koristi da bi se uočilo na kojem se parametru radi.

### BAZIEROVA KRIVULJA

Standard za sve vektorske grafike u svim alatima je da se tzv. pomoćne ili tangentne točke označuju s plusom (+). I još je potrebno znati da se bazierova točka sastoji od 4 točke. matematički rečeno prva točka je natezna, a druga točka je završna. U svim softverima je to povezano pa se ne naziva tako nego se tu radi o spoju jer se jedna točka sa plusom preko središnje točke povezuje na drugu točku, te je to jednadžba kroz tri točke. Naprimjer ako točku prebacimo u drugi režim rada onda je ta točka nezavisna i više ne djeluje kao jednadžba triju točaka ili ako prebacimo u sljedeći režim rada ili treći onda je ta točka tangenta na tu krivulju. U tangentnom načinu nema pomicanja lijevo i desno i radi se predviđanje ponašanja tih točaka.

Ako se radi na nekim dizajnima krivulje moramo koristiti baziera u toj predvidljivosti jer su bazierove krivulje parametarske krivulje trećeg stupnja iz skupine predvidljivih krivulja. Postoje točno krivulje gdje se sa položajem kontrolnih točaka koje su u domeni rada te krivulje, te se za njih odmah radi predikcija čovjeka gdje bi te krivulje stvarno trebale ići i to je bazier. Zbog toga je bazier "pobjednik" svih tih jednadžbi koje su htjele ući u vektorku grafiku. Kvadratna jednadžba ili jednadžba drugog reda naprimjer može raditi krivulje prema gore ili dolje i gore se uvijek paziti na ono što se dobije pod korijenom, mora se paziti dali ćemo dobiti binarno kompleksni dio ili realni. Bazierova krivulja ima svoju naredbu u pisaćem kodu. Bazier se pojavio kad je Pier Bazier krivulju upotrebio u tvornici automobila u njihovom sustavu.

Postscript jezik je jedini jezik koji razumije ispisne tehnologije. Ispisne tehnologije ne poznaju AI format ili doc format, one poznaju samo postscript jezik i zato za njih trebamo imati postscript driver-e.

Go – script simulira bilo koji postscript kod, može biti u bilo kojoj boji. Mijenja se po koordinatnom sustavu i u kodu. Kod bazierove krivulje ima 4 točke, prva točka se uzima kao tekuća točka radnog postscripta i zato imamo naredbu prije te krivulje da stvori točku koja postaje prva točka te naredbe. Dajući nove naredbe tim točkama možemo redizajnirati sve stvari koje imamo, samim mijenjanjem naredbi se to učini. Studentima se zadaju primjeri preko crteža da preko naredba shvati kako se sve te točke i krivulje pomiču i da shvate naravno bazierovu krivulju koja se koristi u svim softverima vektorske grafike kao glavni alat.

U vektorksoj grafici bazierovu krivulju koristimo za WEB. Danas jedan od najstandardnijih jezika ili tehnologija je SVG (skalabilna vektorska grafika) jezik, on u sebi ima slične naredbe kao postscript jezik tako da ako znamo postscript znati ćemo i SVG jezik. Taj jezik se može otvoriti u FireFoxu, Operi...

U SVG jeziku možemo napraviti animaciju i dok god ju povećavamo na vektorskoj grafici preko WEB-a rezolucija će ostati ista i to je jedno svojstvo vektorske grafike. Ona je vezana samo za moment ispisa kada nešto prikazujemo, ako promijenimo rezoluciju onda će krivulja biti glatka što je moguće biti viša u toj izlaznoj rezoluciji. Dok naprimjer pixel u klasičnoj grafici, čiju strukturu stvaraju konstruktori slike kao što je Photoshop, rezolucija je zadane i nemože se skalirati, može se samo rezemplirati, ali s time se obično slika zamuti ili se izgube informacije.

U kodu stvaramo krivulju ili stazu iz primjera. Želimo da je vidljiva pa to i stavimo, ali ako želimo da je nevidljiva mi moramo prvo po vidljivome dizajnirati stazu jer želimo po nekoj stazi postaviti stvar, naprimjer u ovom slučaju trokut koji će se kretati po stazi u obliku animacije. Preko ovakvih animacija se mogu raditi pokusi da se vidi kako tijelo pada niz kosinu ili provjere koliko šta traje i koje je ubrzavanje toga tijela preko animacije. Ako želimo naučiti napraviti nekakvu animaciju prvo postepeno krenemo raditi dio po dio, od jedne slabije i lakše tehnologije

možemo preći na neku jaču i noviju tehnologiju. Danas je važno znati skriptni jezik, ali u vektorskoj grafici to nije važno jer se uvijek uči iz nova. Kad se savlada bazierova krivulja dalje će se slične stvari upotrebljavati. Prvi dio koda se povezuje sa interspaceom, a drugi dio ide korak po korak i tako polako radimo sa različitim parametrima poscripta. Glavni dio koda se dešava u jednoj određenoj petlji koja ima različite naredbe (broj kor, lopatica, kut rotate...). Za rad s bojama moramo znati kolonarni sustav, a da bi ga naučili moramo položiti nekoliko kolegija na fakultetu, ali najpraktičniji dio preko kojeg zapravo mi nešto napravimo i naučimo je preko nečeg konkretnog. Sethsbcolor sustav smo koristili za promjenu boja, moramo znati i da boje rade u određenim domenama. Naprimjer postoji parametar koji je napravljen u kružnici od 0 do 360 stupnjeva, to je klasičan spektar negdje od 400 nm ili 700 nm što mi vidimo je stavljeno u tu kružnicu za korištenje u digitalnim tehnologijama u prametu od 0 do 1. To je naprimjer ako stavimo 0,5 onda je to točno na 180 stupnjeva. Možemo broj postaviti ako da ide rendom postavljanje. Ako želimo regulirati boju z svaku kružnicu posebno onda moramo napraviti petlju u kojoj ćemo to mijenjati ili stavimo samo random number i on će raditi umjesto vas. Hsb sustav se koristi za umjetno koloriranje kad naprimjer želimo umjetno kolorirati stare crno bijele filmove onda najčešće to radimo s tim kanalom jer ako mi kažemo da je nešto pod točno 180 stupnjeva onda je to točno ta valna duljina. U photoshopu ako želimo točno određeno boju onda je to trokolorna mješavina koja je nije jako bliska čovjeku osim na nekim djelovima, a zato je puno bliža u cmy, ali onda opet ta mješavina sadržava naprimjer cijanižuta daje zelenu i ovdje se samo mora znatikoji je to dio kruga ta zelena boja naprimjer.

Rastriranje je postupak kojim čovjek s jednom bojom može napraviti n nijansi. Znači kad radimo to rastriranje mi na ekranu vidimo cijelo vrijeme jednu boju, ali nam je negdje tamniji ili negdje rijeđe ovisno gdje se nalazi taj rasterski element, dali je on bio na istoj udaljenosti jedan od drugo ili je veći. To se zove amplitudno modelirano rastriranje gdje je stalno ista frekvencija udaljenosti, ali se simulacija sivoće radi sa povećanim ili smanjenim rasterskim elementom. Svaka podloga je u kodu pod regulacijom jedne formule i ako mi to promijenimo promijenit ćemo i izgled te pozadine. Ta se pozadina simulira sa određenom gustoćom i to je simulacija sive. Sve te točke u pozadine, što se točke više smanjuju i ta frekvencija to mo ju bolje prevarili. Rasterski elementi koji se fizički tiskaju u tisku sa određenom gustoćom nama simuliraju sivoću. Rastriranje se uči sa simulacijom te da bi student doživio sve te formule korisiti se program matematika te ako znamo formulu u matematičkoj formi onda se samo treba znati kako se to u nekom određenom jeziku treb napisati. Kako domena napada u toj formuli točku to ovisi o sivoći, te da bi ona simulirala zacrnjenje. Ako želimo to postići onda moramo podmetati različite formule i onda vidimo završno šta smo dobili. Rgb color sustav ne postoji u tisku, nema tehnologije za ispis koja radi u tom sustavu. U digitalnom djelu za boje postoje standardi, da se u WEb okolini i html radi sve u rgb sustavu. Boje se u Java sustavu mogu zapisati kao riječi ili kao njihov zadani kod. Upotrebljava se više tehnologija da bi se to prikazalo u određenom sustavu. Naprimjer Word poznaje samo rgb sustav i tu se trebaju raditi te promjene i ograničenja.