

## **Važnost IKT (informacijsko-komunikacijskih tehnologija) u metodici nastave na Grafičkom fakultetu - Osvrt na predavanje**

Počnemo s fontovima. Definicija fonta je tekst kojim se oblikuju priče u nekim tiskovinama, novinama, knjigama, i slično. Font je osnovni resurs koji na Grafičkom fakultetu izrađujemo potpuno samostalno. Čitkost i čitanje se nije promijenilo od prošlosti do danas, promijenila se samo izrada teksta i slovnih znakova. Fontographer (čija je inačica Fontlab) je program s kojim se izrađuje potpuno novi font ili uređuje postojeći. Ako otvorimo neki postojeći font, otvara se prozor u kojem vidimo set znakova. Font je u Fontographeru uređena nakupina kodnih pozicija i na svakoj kodnoj poziciji imamo sliku (na engleskom glyph). Ako otvorenu tablicu fonta stavimo u decimalni oblik vidimo klasičnu ASCII (American Standard Code for Information Interchange) tablicu. Verzal A je u njoj na kodnoj poziciji 65 po ASCII standardu. ASCII standard dolazi iz doba matičnih printera koji su kodirali kako će se iglice u njima podesiti za određeni znak, što je bilo unaprijed kodirano. Sada više nije tako, danas imamo kodno mjesto i mi samo biramo što će na njemu biti prikazano. Kada otvorimo koordinatni sustav samo jednog kodnog mjesta, na primjer za verzal A, vidimo neke pravce. To su pravci koji ograđuju polupovršine koje u presjeku stvaraju digitalni četverac. Digitalni četverac se u prošlosti mogao nositi u rukama izliven u olovu, no danas to nije tako. Danas je digitalni četverac koordinatni sustav (to jest zatvoreni prostor, poligon) koji je zatvoren jednadžbama pravca. Ti pravci su beskonačni, što je bitno jer možemo npr. kvačice za slova Č, Ć, Š, Ž staviti koliko god visoko želimo, što se prije nije moglo.

Također, bitno je uzeti u obzir kako će dva znaka izgledati jedan kraj drugoga na pisnoj liniji, što vidimo u programima za obradu i slaganje teksta (Word, InDesign, Photoshop). Da ne bi morali otvarati te programe, dizajnirati svaki znak, generirati font, instalirati ga u Windows ili Mac mapu, koristimo prozor za metriku fonta, koji služi kao simulacija. Dobar primjer su verzali slova A i V, i tu provjeravamo kako bi izgledalo da se ta dva znaka susretnu jedan kraj drugoga na istoj pisnoj liniji. U font metrici se može napisati neka riječ, i vidjeti ponašanja svakog znaka unutar fonta. Što se tiče slova A i V, ako idu jedno za drugim, razmak se mora smanjiti jer inače dolazi do problema s čitljivosti fonta. Desna linija četverca se može pomicati i utjecati na razmak znakova u fontu, ali ona utječe na cijeli font. Zato radimo iznimke; a njih zovemo parovi podrezivanja ili na engleskom kerning pairs. Pomoću kerning linije možemo namjestiti da to vrijedi samo za određene znakove.

Dakle, Fontographer (ili Fontlab) je jedan od glavnih softvera s kojim ne samo da simuliramo izradu jednog slovnog znaka, zareza, razmaka između riječi, i tako dalje, nego i radimo kako će se kerning parovi raditi i kada će se taj font upotrijebiti u nekom softveru za slaganje teksta.

Bitno je naglasiti da u fontu ne postoje fizičke jedinice, nego samo relativne, te da tek softveru za slova skaliraju fontove na fizičku dimenziju.

True type fontovi u sebi imaju mogućnost da u sebi mogu napraviti beskonačno mnogo parova podrezivanja, koji se automatski uključe dok mi ispisujemo neki tekst u softverima za slaganje i obradu teksta.

Što se tiče logike rađanja znakova, ili geneze izrade fonta - radimo prvo slovo O, onda Q, ili prvo D pa Đ i slično.

Koristimo i softver napisan u c++ koji otvara prozor u koji stavljamo parametre, kao što su širina, visina, znakovi itd. i generira tif file koji otvaramo u Photoshopu. To možemo raditi i u Adobe Illustratoru, ali će nam trebati puno više vremena. Ultraedit je PostScript program, u njemu su PostScript naredbe koje čita c++ program i stvara parametre koje dinamički mijenjamo kroz prozor i samim time mijenjamo i file.

Bezierove krivulje su parametarske krivulje trećeg stupnja iz skupine base plane krivulja. Spadaju i u predvidljive krivulje (na engleskom predictable curves), što znači da se sa položajem kontrolnih točaka koje su u domeni rada krivulje odmah radi predikcija za čovjeka gdje bi tijelo te krivulje stvarno trebalo ići. I zato je Bezier bolji od svih ostalih jednadžba krivulje koje su trebale ići u vektorsku grafiku. Pierre Bezier je tu krivulju iz base plane porodice krivulja počeo primjenjivati za dizajn haube u Renaultovoj tvornici automobila.

Tangente ili pomoćne linije se u programima vektorske grafike standardno označavaju s plus simbolom. Da bi se to shvatilo, mora se jako dobro znati kako Bezier, koji se sastoji od samo četiri točke (prva točka, natezna ili tangentna, sljedeća točka), funkcionira da se između tih točaka dobije krivulja. Početna, natezna, završna, i druga točka je matematički izraz i koristi se u matematici, a u svim softverima se razgovara o spoju, o spojnim točkama. Plus preko spojne točke je s drugom funkcijski spojen, što je bitno za naglastiti. U Fontographeru imamo više načina spajanja krivulja - različiti mode-ovi za spajanje krivulja, recimo koordinatni (u kojem je točka nezavisna) i tangentni način (ili mode).

Ghostscript ili GSview softver je softver koji simulira ispis, te je simulacija PostScripta. Ako bilo koji PostScript kod u eps ili ps opciji pošaljemo u softver, on će ga prikazati. Možemo namjestiti kako bi izgledao ispis ovisno o mnogo parametara, recimo ovisno o vrsti printera (je li printer u boji, crno bijelo, i ostalo).

Kada radimo krivulju, koristimo naredbu *curve to* u Ultraeditu, a *move to* naredba stvara prvu točku *curve to* naredbe.

Što se tiče vektorske grafike za web, koristimo svg jezik. SVG je kratica za Scalable Vector Graphics, i to je jezik za vektorsku grafiku na webu. Današnji svg jezik poznaju svi browseri. Može se reći da je svg dijete Adobe-a, a ima jako sličan jezik kao PostScript jezik. Svg jezik je iz porodice grafičkih xml jezika. Svg se može otvoriti u firefoxu, operi ili bilo kojem drugom browseru.

Svojstvo vektorske grafike je da nije vezana za rezoluciju, nego samo za moment ispisa kada nešto prikazujemo. U pixel based grafici, koju koriste softveri kao što je Photoshop, rezolucija je zadana i ne možemo ju skalirati nego samo resample-ati, i ako to napravimo slika se zamuti i gube se informacije.

Ta tehnika je dobra za rađanje animacija, pogotovo animacija fizikalnih pokusa jer se mogu precizno podesiti svi parametri, na primjer animacija tijela koje klizi niz kosinu. Lako možemo precizno podesiti trajanje animacije, ubrzavanje tijela i slično.

Koristimo puno tih malih softvera s maskama da se ne bi izgubili u PostScript kodu koji ih generira. Koristimo ih da bi se upoznali s kodom koji je zapravo jednostavan.

U radu s bojama bitno je da moramo znati u kojem kolornom sustavu radimo. Boje rade u određenim domenama, na primjer hue je spektar napravljen u kružnici i stavljen u parametar od nula do jedan (jedan je 360, nula je nula).

Rastriranje je proces koji se radi da bi se s jednom bojom moglo napraviti n mnogo nijansi te boje. Rasterski element utječe na to da boju vidimo negdje tamnije, negdje rjeđe ovisno o tome gdje se nalazi rasterski element. Bitno je i naglasti jesu li rasterski elementi bili udaljeniji jedan od drugog, ili su bili na istoj udaljenosti ali su pojedinačni element veći. Amplitudarno modalno rastriranje je rastriranje kada je ista frekvencija udaljenosti, ali se simulacija radi s povećanim ili smanjenim rasterskim elementom. Što smo udaljeniji od slike, to bolje radi ta iluzija.

Kad se radi o digitalnoj boji (ne samo u fizičkom nego i pigmentnom obliku), sve to se treba izgenerirati iz računala, to jest iz ekrana na kojem je boja simulirana. Različito je kada radimo virtualno, i kako će to zaista izgledati u fizičkom svijetu kada se isprinta negdje. Na ekranu radimo u RGB sustavu, koji ne postoji u tiskovnoj tehnologiji. Bitno je znati koji je opseg obojenja u RGB sustavu, to jest što može prikazati ekran, a koji je u CMY sustavu.

U HTML (Hypertext Markup Language) jeziku se može raditi s bojama, ali samo u RGB sustavu.

U PostScript jeziku se boje određuju naredbama. Recimo, ako želimo postići žutu boju u CYMK kolor sustavu, pišemo naredbu *setcmykcolor*, i moramo odrediti četiri parametra. Namještamo da je cyan 0, magenta također 0, žuta je 1 i crna je 0.

Radimo i u JavaScript, u kojem je bitno koji kolorni sustav koristimo. Na primjer kod promjene pozadine Java mora znati hoće li pristupiti po HTML standardu s imenima (yellow), ili će se koristiti recimo heksadecimalni zapis.

Word poznaje samo RGB kolor sustav. Kada dolazi do ispisa u Wordu, ako ubacimo tif file u CMYK sustavu, događa se konverzija iz CMYKa u RGB sustav i slika koja izlazi više nije ista onoj koju smo htjeli dobiti u našem tif file-u.

Razlika između HTML-a i PDF-a je da oboje prikazuju tekst, slike (iako PDF može RGB, HSB i CMYK sustav, a HTML samo RGB), ali PDF poznaje pojam stranice (na engleskom page), a HTML ne. U PDF-u imamo naredbe koje reguliraju lijevu, desnu, gornju, donju, čak i središnju marginu (ako je duplerica). Postoji xml jezik za kontrolu PDF-a.

Destiler iz PostScripta stvara PDF, i mi kontroliramo na koji način. S xml-om imamo mnogo mogućnosti pri oblikovanju PDF-a, možemo u margine staviti razne komentare te stvarati moderne dinamične PDF-ove koji stvaraju animacije.