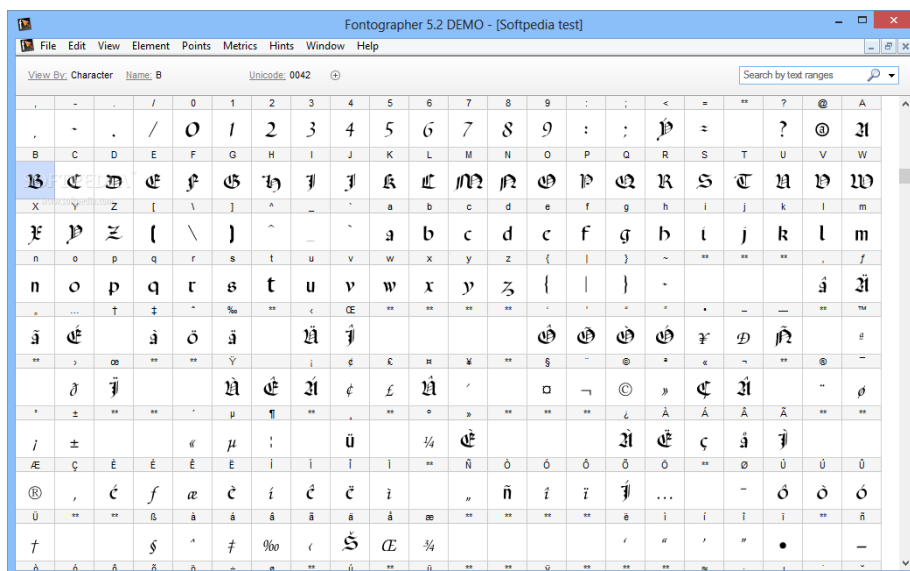


# Važnost IKT u metodici nastave na grafičkom fakultetu

Font je elektronički niz podataka u kojem se nalazi skup slovnih znakova, i slovnih simbola (tipografskih) poput slovnih ornamenata. Termin font isprva se upotrebljavao za skup metalnih slova jedne veličine i jednog tipa slova (jednog tipografskog karaktera), od 1990. fontovi su uglavnom digitalni i upotrebljavaju se u računalstvu. Programi u kojima radimo novi font nazivaju se fontographer i fontlab. U tom programu svaki student može napraviti sam svoj font koji može otvoriti u tri programa: Word, Adobe Illustrator i Adobe Photoshop. Trenutno mi koristimo fontographer. Postoje različiti pogledi na font tablicu. Jedan od njih je oblik u kojem vidimo klasičnu ASCII kodnu tablicu, koja je po američkom standard koji je određivao kako će se iglice u matičnom printeru podesiti za prikaz određenog znaka. U koordinatnom sustavu slova vidimo određene pravce koji stvaraju digitalin četverac. Ti pravci su beskonačni te omogućuju da sami dizajniramo slova. Slova možemo podešavati kako će biti udaljena, kako bi čitljivost teksta bila što bolja. Standardni problem su što se slova nekad poklapaju, a znamo da to baš nije poželjno. Zbog toga radimo iznimke, to su tzv. parovi podrezivanja. Slova pomičemo tako da kliknemo na slovo I na liniju na kojoj piše slovo K. U donjoj tablici možemo vidjeti naredbe koje smo napravili za određena slova, tj. za koliko smo koje slovo pomakli u lijevo ili desno, gore ili dole. Kod dizajniranja fonta potrebno je jako puno znanja o slovima da bi si olakšali rad. Tako ćemo prvo dizajnirati slovo O da bi iz njega napravili slovo Q, slovo D da bi iz njega napravili slovo Đ. Ako radimo rukopisni font potrebno je ispisati što više slova svojim rukopisom, te izabrati ono koje nam se čini tipično za naš rukopis, skeniramo ga, stavimo ga u kodnu tablicu i aktiviramo vektorizaciju slike te program automatski pretvori cijelu sliku po rubovima u Bezierove krivulje.



Slika nap prikazuje jedan font u fontographeru.

PSConvert napisan u C++. Ima jednostavnu korisničku masku koja prikazuje tipične grafičke parametre, a to su broj točaka po inchu (rezolucija), širina i visina. Također još su tri parametra: prvo slovo, drugo slovo i gustoća. Kad startamo softver, on stvara sliku koju otvaramo u Photoshopu jer softver generira tif zapis. Softver stvara određeni koordinatni sustav kako bi pozicionirali određene slike, a također i maske jer ovaj program dizajnira linije koje se kao lepeza šire iz jedne točke od 0 do 180 stupnjeva, a vidljivo je samo kroz masku slova koje smo upisali kao parametre u PSConvert. PSConvert je jako brz i zato i jest dobar, dok ni za istu stvar Illustratoru trebalo od 3 do 4 sata. PSTipograf ima mali simulator koji ima puno veće moćnosti od prethodnog softvera. Tako ipa prozor za upis teksta koji želimo staviti u krug, također možemo definirati i rezoluciju te visinu i širinu. Bazierove krivulje na primjeru slova O u Fontograferu - točke koje vidimo naznačavaju spojne točke Bezierovih krivulja i kad pritisnemo na točku vidimo dva plusa (oni označavaju pomoćne ili tangentne točke krivulja). Krivulja se zapravo sastoji od 4 točke, imamo prvu točku, onda dvije natezne ili tangentne točke i sljedeću točku, i onda između prve i zadnje točke dobivamo krivulju. To je matematički način rada. U svim softverima se radi o spoju jer je jedan plus preko spojne točke funkcijski u vezi s drugim plusom i onda dobivamo jednadžbu kroz tri točke. Ako spojnu točku prebacimo u drugi način rada onda je to nezavisno, nema više jednadžbe između dva plusa. Ako tu točku prebacimo u treći način spajanja onda to postaje tangenta na krivulju. Zato je bitno da znamo razim točaka spajanja s Bazierovim krivuljama, a ne čiste matematičke jednadžbe. Prvi put se Bazierova krivulja počela koristiti za dizajn haube u tvornici automobila, a danas se koristi u svim programima gdje je potrebna vektorska staza. Ghostscript odnosno simulacija ispisa. Možemo mu poslati bilo koji postscript i program će nam ga otvoriti. Možemo mijenjati način ispisa, odnosno hoće li on biti crni ili bijeli ili u boji. Na zadanom primjeru iz videa, mijenja se zaobljenost haube. Curve to je naredba za Bazierovu krivulju. Tu vidimo 6 brojeva, a ne 8 jer je naredba curve mikro-kodom definirana tako da prvu točku uzima kao momentalno tekuću radnu točku postscripta koju moramo stvoriti prije naredbe curve to. Zato prije naredbe curve to ide naredba move to. Bazierovu krivulju možemo koristiti i u drugim programskim jezicima i tehnologijama npr. u vektorskoj grafici za web. Jedna od najstarijih današnjih tehnologija je SVG jezik koji poznaju svi browseri. On u sebi ima slične naredbe kao postscript, tako da kad se on nauči koristiti nije problem naučiti koristiti bilo koji drugi program koji radi vektorsku grafiku. SVG je iz porodice XML jezika. Vektorska grafika nije vezana za rezoluciju. Vezana je samo za moment ispisa kad nešto prikazujemo. Ako promijenimo rezoluciju krivulja će biti glatka. Na primjeru vidimo krivuljastu stazu i trokut koji putuje po stazi. Staza može biti nevidljiva i vidljiva. U programu vidimo dvije kratice, ako želimo da je vidljiva "M" označava move to, naredba "C" označava curve to. Move to ima 2 brojeke, a naredba curve to ima 6 brojki. Također imamo i naredbu style tu koja nam pokazuje kakve će boje biti staza, hoće li biti ispunjena ili ne. Postoji i naredba za trokut koji govori kojom će brzinom ići. Na ekranu cijelo vrijeme vidimo istu boju, samo je na nekim djelovima tamnije, a negdje rijeđe, ovisno o tome gdje je rasterski element bio udaljen jedan od drugoga ili na istoj udaljenosti, ali veći. To se zove amplitudno

modreliranje gdje imamo uvijek istu frekvenciju, ali se simulacija sivoće radi s povećanjem ili smanjenjem rasterskih elemenata. Podloga je pod regulacijom jedne određene formule koja poziva puno naredbi. Nema piksela, rasterski elementi koji se fizički tiskaju sa određenom gustoćom nama simuliraju sivoću. Da bi doživjeli formule koje su jako dosadne i suhoparne, koristimo jako puno matematičkih programa. Ako znamo matematičku formulu, onda to treba znati napisati u nekom jeziku određenom sintaksom sa naredbama, add, sub, div, idiv. Prvo se treba vizualizirati što uopće želimo dobiti. Postoji eksperimentacijski notebook u matematici koji ima puno formula već proređenih da bi se vidio mehanizam. U primjeru se koristi plot 3D i countorplot naredbe iz matematike. To sve treba izregenerirati da bi se vidjela boja na ekranu. Na ekranu vidimo RBG sustav boja, a on ne postoji u ispisnom sustavu tj. on nema plavu, crvenu i zelenu bočicu boja. To je napravljeno u ispisu CMYK boja. Treba se znati koji su opsezi obojani u RGB sustavu, a koji u CMYK sustavu. Da bi se programski koristile boje treba ih dobro poznavati. Kada treba ući u reprodukciju boje javlja se mali problem. Npr. ako napravimo odličnu pripremu, radimo s jako kvalitetnim bojama, finalni proizvod neće biti dobar, ili ako imamo potkrivene ili transparentne boje gdje onda ako smo otisnuli boju na boju, ako je transparentna, u oku ćemo dobiti treću boju, ako su pokrivne onda dobijemo onu prvu boju i u tom slučaju priprema nije ista. U web okolini se može raditi s bojama, ali smo u RGB sustavu i to samo u nekim određenim standardima. Da bi boju u postscript koristili možemo se igrati s različitim color sustavima. U primjeru imamo kvadrat koji je u boji u hsb sustavu. Mijenjamo ga u CMYK sustav i onda imamo četiti parametra. Paralelno s tim ulazimo u Javascript. Mijenjamo boju pozadine, te odlučujemo hoće li program pristupiti HTML sustavu sa imenima boja ili možda kodom iz RGB-a. Može se upotrijebiti nekoliko tehnologija da bi prikazali RGB i CMYK color sustav. Word poznaje samo RGB sustav, a u njega ubacujemo CMYK tiff i ispišemo iz worda na color printer dolazi do konverzije CMYK tiffa a u RGB na kraju dobijemo i nakraju dobijemo iz RGB-a konverziju na printer i onda to nije više ona slika koju smo u Photoshopu napravili. Razvoj grafičkih jezika baziranih na XML-u, razlika između HTML-a i PDF-a može biti prikazana i u CMYK sustavu, dok HTML, samo u RGB sustavu, ali najvažnije je to što PDF poznaje pogla stranice (page), dok HTML ne, i zbog toga im treba drugačije pristupati. U PDF-u imamo naredbe koje reguliraju, lijevo i desnu stranu pa čak i središnju ako postoji margina. Postoji XML jezik za kontrolu PDF-a. To je jezik gdje imamo visinu i širinu stranice, margine. To se sve radi i u najmodernijim browserima koji imaju određeni plug in, ali to se radi s PDF generatorima, a ne samo da zna s PDF-om raditi s tzv. destilatrom gdje se ima jako puno parametara, a destiler služi za postscripta stvara PDF i to možemo kontrolirati na koji način tj. destiliramo na svoj način. Ali s druge strane to sve možemo kontrolirati s različitim naredbama. Desna je stranica uvijek neparna, a lijeva parna.

