

## VAŽNOST IKT U METODICI NASTAVE

Tema predavanja je case study vezan za informacijsko-komunikacijske tehnologije koji je konkretno vezan za kompleksnost gradiva i zadataka zadanih kao oblik prakse. Prvo polje grafičke obrade koje je profesor Pap naveo su fontovi. Zanimljivo je to što izradom možemo iskusiti ono što su iskusili i tvorci fonotva na novoj razini jer se izradba/editiranje istih prilagodio suvremenom načinu života te je najčešći program obrade Fontforge dok postoje i besplatne varijante. Opća definicija fonta smatra se nakupinom dogovorenih znakova. Naglašava važnost poznavanja engleskog jezika jer se sustavi za obradu temelje na ASCII (Američki standardni kod za razmjenu informacija). Kodna mjesta olakšala su nam izradu slova zbog zamjenjene upotrebe kodiranja unaprijet te na ovaj način sami određujemo izgled slova u standardnom tkzv. digitalni četvercem. Nekada se on mogao nositi u rukama zbog toga što je bio lijevan u olovu. Kako bismo vidjeli izgled, složili i koristili proizvedena slova kao vremenski isplativa tehnika možemo napraviti u navedenom programu otići u Metric i provjeriti kako se slovni znakovi ponašaju jedan pored drugoga. Iznimke ili parovi podrezivanja slova su između kojih nastanu preveliki razmaci te u opciji Metrics možemo podrezati za određeni broj jedinica. Važna napomena je da proizvod slovni znakova ne mjerimo u cm, m..itd, već u jedinicama. Pri dizajniranu slova koje ne postoje u ASCII kodnom sustavu, prvo mijenjamo kodiranje (najčešće se za hrvatski koristi Latinica2), a zatim kronološki radimo outline slova. Na primjer pri izradi Č i Ć baza je slovo C pa se njime inicijalno bavimo, a na njegov outline dodajemo kvačice. Nakon priče o fontu prelazimo na simulator napravljen na Grafičkom fakultetu. Otvaranjem softvera određujemo mu parametre te on otvara sliku .tif formata u Photoshopu. Što smo zapravo napravili? Parametar gustoće odredio je koliko blizu će iscrtati linije, koliko će biti nakošene, koja slova iscrtava itd. Cijeli program napisan je C++ jezikom. Tema kojom se bavimo nakon je još jedan simulator kojim možemo mijenjati rezoluciju proizvoda ovisno o potrebi i želji. Profesor navodi niz primjera, za ekrane dovoljno nam je 70-100dpi, dok za svjetlosne osvjetljivače rezolucija raste čak iznad 3000dpi. Također možemo utjecati na izgled i okvir grafike te je studentu dopušteno i zadano da ih sam mijenja te istražuje okvire rada u praktičnom smislu. Točke slovni znakova (ili bilokakve vektorske grafike) spojene su krajnje točke Bezierove krivulje(najčešće 3 točke); krajnje točke povezane su i označene križićem/plusom,a tangenta točka centar je koji utječe na krivulju. Pomicanje točaka ovisi o poziciji koju ima u Bezierovoj krivulji te radi boljeg baratanja možemo točkama mijenjati funkcije u kutne ako

želimo da funkcioniraju samostalno. Dizajne krivulja radimo predviđanjem Bezierove krivulje i cijelo područje parametričnih krivulja bavi se predviđanjem mogućih ishoda. Njen tvorac Piere Beziere radio ju je pri dizajnu „haube“ Renaulta kako bi površina dizajna bila glatka, gdje vidimo još jednu sjajnu uporabu vektorske grafike. GSU je software koji simulira kako bi izgledao ispis printera te se znanje još primjenjuje izmjenom dijelova (npr. auto kojemu reguliramo izbočenost haube) koda poznavstvom koordinatnog sustava i spomenute Bezierove krivulje. Otvaramo PostScript file u tekstovnom editoru pa tražimo koji dio koda se odnosi na traženi dio automobila/nacrtanog predmeta. Move to naredba koristi se za crtanje prve, poznatije kao tekuće točke te se za svaki idući element koristi koordinata manje. Adobe je tekućim točkama pojednostavio i olakšao korisnicima uporabu vektorskih programa. Na taj način dobili smo mogućnost shvatiti na vidljivim primjerima navedenu funkciju. Nakon toga tema se razvija u smjeru SVG-a, jedan od XMLovih jezika, konkretno korišten za dvodimenzionalnu vektorsku grafiku. Vektorska grafika zanemaruje rezoluciju te kod nje možemo beskonačno puno puta nešto uvećati. Učenje o Bezieru i osnovama vektorske grafike i SVG-a koristi se kao uvod u učenje osnovnih animacija. Profesor naglašava važnost postupnog učenja, od Bezierove krivulje pa postepeno do programskih jezika te upotrebu simulacijskih software. Zatim pomoću softvera koji generira kružnice različitih boja te „lopatice“ prikazuje mijenjanje istih. Lopatice zamjenjujemo s kodiranom funkcijom lopaticex koja ih crta na drugačiji način, dok kružnicama mijenjamo promjer, položaj te na poslijetku boju. Pri korištenju boje koristimo komandu Adobeovog PostScript jezika „setHSBcolor“, HSB je tkzv. „humani naziv“ za boje, a akronim je za HUE(određuje samu boju)- SATURATION(koliko je boja „puna“, 0% S daje nam sivu nijansu korištene boje)- BRIGHTNESS(0% daje crnu, a 100% ukoliko je 50%, bijelu). Hue je definiran kao kružnica (od 0 do 360) u kodu je označen pomoću 1 i 0 te željenu boju dobivamo znanjem gdje na kružnici se nalazi boja. Kontrola rastriranja još je jedan dio grafike koji je gotovo neizvršiv bez printanja, pa postoji simulacija kojom brže i učinkovitije možemo naučiti parametre te domene. Smisao rastriranja je uporaba jedne boje kako bismo dobili veći broj nijanse na ekranima, ali pogotovo u tisku. Što su točkice rastera manje to je rezolucija veća, bolja, realističnija... Pri kodiranju, bitno je primijeniti znanje iz matematike te vremenom prilagođavamo već poznate formule (npr. kvadratna jednadžba) našim potrebama. Možemo odrediti kako se točka/sinusoida ponaša na određenoj domeni ukoliko dobro baratamo formulama. Sinusni raster (sinusoida od 0 do 1) korištena je na hrvatskim kunama. Digitalna boja u svom pigmentnom, tiskovnom obliku printa se CYMK bojama, a grafičkim tehnologizima transformacija iz RGB (ekran) i CYMK (printer) mora biti poznata zbog nemogućnosti

pripravka nekih boja. HTML također dopušta samo RGB sustav. Crteži vektorske grafike (.svg) mogu se prikazati u bilo kojem browseru stavljanjem u HTML kod te ne moramo programirati ilustracije već ih možemo crtati u npr. Adobe Illustratoru.