

## Što je PostScript i što nam nudi?

Zadatak ovog osvrta bio je pobliže opisati funkcije PostScripta i kako se sve s njime možemo služiti da bismo riješili složene grafičke situacije u budućnosti. Za početak, PostScript je dio kolegija iz računarske tipografije i računarske grafike koji se predaje na Grafičkom fakultetu. A kao studentima Grafičkog fakulteta nama je od iznimne važnosti upoznati se s njime. Nadalje, PostScript je idealni alat i za sve one koji imaju volju eksperimentirati u računarskoj grafici, tipografiji, slici, rasterima i bojama, a na koje sve načine, to smo mogli pročitati u zadanoj knjizi. U predgovoru smo doznali kako PostScript ima svoj specifični grafički rječnik i koriste se samo u području oblikovanja grafike ma stranicama za tisak. Naučili smo neke osnovne pojmove koji će nam služiti pri korištenju PostScripta i učenju istoga, npr. operatore i parametre.

Prve komande, tj. operatore koje smo naučili, bile su *moveto*, *lineto*, *stroke* i *showpage*. Najčešći početak PostScript stranice je *moveto* komanda. Početak te komande zadan je parametrima pisanim ispred komande, a oni označavaju koordinate točke. Same naredbe *moveto* i *lineto* su bezdimenzionalne sve dok se njezinoj liniji ne pridruži debljina. To se postiže komandom *stroke*. Ta komanda iscrtava liniju prema prije postavljenim karakteristikama za debljinu, boju i sivilo, a ako prethodno ništa nije određeno tada će debljina linije biti jedna točka. Komanda *showpage* briše postojeće stanje i postavlja parametre za ispis slijedeće stranice. Nakon tih komandi uslijedile su komande: *rlineto*, *closepath* i *setlinewidth*. Parametri komande *rlineto* određuju za koliko je pomak u vodoravnom ili okomitom smjeru od zadnje točke gdje smo se zaustavili. Komandom *closepath* zatvaramo lik, a željena debljina linije određuje se komandom *setlinewidth*. Nakon što bi napravili linije i oblike prethodno spomenutim naredbama, mogli bismo ih i popuniti ili pobojati. To se radi naredbama *fill* i *setgray*. Također, linije koje smo iscrtali mogu imati različite vrste rubova pa tako s naredbom *setlinecap* rub linije može biti kvadratni i zaobljeni te završavati definicijom linije ili produžen za polovicu debljine linije. Budući da možemo imati više vrsta rubova linija, možemo imati i više vrsta spojeva ili pregiba linija. Tu vidimo kako je PostScript zapravo vrlo intuitivan i lako se koristi u grafici. Pomoću njega možemo manipulirati linijama i oblicima, mijenjati debljinu linije, boju linije i oblika, liniju pretvoriti iz kontinuirane u točkastu ili isprekidanu, rotirati objekte, mijenjati veličinu i još mnoštvo toga. A kada govorimo o zaobljenim linijama, tu je vrlo važno spomenuti Bezierovu krivulju. S njom smo se već susreli na vježbama iz vektorske grafike na predmetu Digitalni multimedij. Bezierova krivulja je osnova mnogih zaobljenih linija, ona je polinom trećeg stupnja, a definirana je s četiri točke. Naredba koja opisuje stazu Bezierove krivulje naziva se *curveto* i ima samo tri parametra, tj. tri koordinate jer je polazna točka već prije određena naredbom *moveto*. Bezierovim krivuljama lako možemo riješiti tipografiju i njezinu problematiku budući da je Bezierova krivulja predvidiva i precizna. Nadalje, grafike rađene pomoću računala prepune su likova s ponavljanim oblicima. Dizajneri najčešće planiraju ponavljanje crteža po nekom pravcu, krivudavom putu, kružnici, simetriji. U PostScriptu imamo naredbu *repeat* za takve slučajeve. Također, pri programiranju PostScriptom važno je napomenuti kako svaka slijedeća naredba poništavaju

prethodnu istoimenu naredbu. Ako želimo zapamtiti put i način iscrtavanja za kasniju upotrebu možemo upotrijebiti komandu *gsave* za spremanje tekućeg grafičkog stanja i komandu *grestore* koja obnavlja zadnje spremljeno grafičko stanje. U prvom poglavlju smo još naučili neke od procedura i kako se duplicira i oduzima te koje se matematičke operacije koriste u PostScriptu, programe mreža i naposljetku sustave boja. Prvo smo naučili kako definirati cmyk boju koja ima četiri parametra i zo se izvodi naredbom *setcmykcolor*. Program CMYK crta tablicu boja koristeći procedure kvadrat i txt da bi se prikazale pojedine boje tog tiskarskog modela boja i pripadni iznos parametara komande. Osim sustava cmyk boja, tu su i rgb i hsb sustavi boja. Komande za njihovo definiranje su *setrgbcolor* i *sethsbcolor*. Kao RGB modela HSB modela konverzija u CMYK događa se kada se PostScript komanda *sethsbcolor* pošalje prema PostScript RIP-u koji ima podešenu konverziju za taj vanjski CMYK uređaj. Slično se događa kada šaljemo kolor komande na crno bijeli PostScript printer pri čemu se vrši konverzija u ekvivalentni parametar *setgray* komande. I time se zaključilo prvo poglavlje.

Drugo poglavlje bavi se programiranjem tipografije, tj. slovniim znakovima. Tipografija je proces kreiranja tekstovne forme, pri čemu se poštuju dobro proučene stare grafičke norme. Zadatak je tipografije da pomoću pisma na najbolji mogući način prenese informaciju. Osnovni je element tipografije slovni znak: verzal, kurent, kurziv, brojke, posebni znak, interpunkcija i razmak između riječi. Slovni znakovi oblikuju se unutra četverca. To smo imali priliku vidjeti na prvoj vježbi u vektorskoj grafici kada smo izrađivali vlastiti font. Naime, četverac je bio pravokutnik unutar koje smo postavili sliku slovnog znaka i njegovu geometriju. Njega čine četiri pravca. Također, mogli smo primijetiti kako si neki slovni znakovi u potpunosti smješteni unutar četverca, a neki poput slova g j Ć i sličnih slovnih znakova su djelomično izlazili iz četverca. Dijakritički znakovi ponekad u potpunosti bili izvan četverca. Dno četverca leži na pismovnoj liniji, donji lijevi ugao četverca je nulta točka slovnog znaka. Debljinska vrijednost slova uglavnom uključuje cijelu širinu slike slova sa dodatnom bjelinom do slijedećeg slovnog znaka. Nulta točka pozicioniranja slijedećeg slova u tekstu je na točki debljinske vrijednosti prethodnog slova. Mi smo na vježbama, naravno, dizajnirali font latiničkog pisma. Latiničko pismo ima četiri linije, a to su: pismovna linija, krov verzala, krov kurenta i descender. Također, naučili smo neke osnovne visine i dimenzije slovnih znakova koje su detaljnije pojašnjene u ovoj knjizi. Za početak, dobro je spomenuti neke od komandi kojima se koristimo u izradi fonta. To su: *findfont*, *scalefont*, *setfont* i *show*. Različiti fontovi međusobno se razlikuju po debljinskim vrijednostima i po tome što nemaju istovrsni set znakova pa se zato moralo definirati više setova znakova koji su se željeli prikazati. Nadalje, da bi program na PostScript uređaju potrebno je poslati fontove sa programom kao što je Downloader. A prije slanja, važno je provjeriti funkcionalnost fonta. Ima li nekakvih nepostojanost u ispisivanju, različitih razmaka između slova ili možda debljini slova. Također, vrlo je važna čitljivost, a ona ovisi o pravilnom odabiru visine slova. Relacija visine slova i udaljenosti gledanja pisma je vrlo važna.

Slova su u računarskoj grafici određena putanjom ovojnice na različite načine: pravci, dijelovi kružnice. Naravno, PostScript tu koristi Bezierovu krivulju pa se tako slovni znak najčešće prikazuje kao popunjeni prostor omeđen unutarnjom i vanjskom ovojnicom i za to nam je dovoljna komanda *show*. Slova možemo ispisati i samo outlineom, tj. linijama koje leže na Bezierovim putanjama, naredbom *charpath*. Ovojnice koje napravimo tom naredbom tek postaju vidljive upotrebom naredbe *stroke*. Postoje fontovi koji su definirani kao outline koji se popune u momentu ispisivanja, nadalje čiji su znakovi definirani nezatvorenim linijama ili fontovi definirani bitmapom. Danas se upotrebljava samo outline vrsta fonta za koju je rezultat naredbe *charpath* identičan i za *true* i *false* logički sud. Nakon toga smo vidjeli primjere u kojima se koristila upravo ta vrsta fonta. Također, saznali smo kako popuniti zatvoreni put komandom *clip* bez obzira je li osnova grafički lik ili ovojnica slova. Tu naredbu koristili smo i na vježbama kada smo radili masku za slova, a sličan primjer prikazan je i u knjizi. U ovom poglavlju još se spominje mnoštvo komandi kojima se možemo koristiti za neka grafička rješenja i uređivanje fonta. Pomoću njih možemo raditi maske slova, možemo mijenjati font, debljinu i visinu fonta, možemo mijenjati razmak između slovni znakova i kontrolirani ispis znakova iz stringa. Nadalje, PostScript posjeduje aritmetičke operatore, stack operatore, relacijske operatore, logičke operatore, operatore uvjetnog izvršavanja procedura, operatore polja, string operatore i mnoge druge čije smo primjere i funkcije mogli vidjeti u knjizi. Većinu toga smo znali iz iskustva s raznim programima, a najviše Wordom. Ovdje smo samo naučili imena tih operatora i način programiranja istih. Nadalje, vidjeli smo proceduru prijeloma teksta procedurom LPRELOM. Ta procedura preuzima poziciju početka prve pismovne linije i šalje riječ po riječ u proceduru PRELOM. U proceduri LPRELOM rješavaju se dva moguća slučaja, a to su: nađen razmak između riječi i nije nađen razmak između riječi. Te situacije rješava procedura PRELOM ovisno stane li riječ u redak ili ne. U knjizi smo vidjeli kako to sve zapravo izgleda u kodu. Nakon toga, prelazimo na poglavlje broj tri.

- U trećem poglavlju proučavali smo programiranje piksel grafike. Poglavlje je započelo slikom sastavljenom od sivih piksela. Piksel je najmanji element slike i za njega je karakteristično da na cijeloj svojoj površini ima jednoličnu sivoću. Vrijednost sivoće piksela dana je jednim brojem u jednom bajtu. Slika se određuje nizom brojaka koje pravilno grade ukupnu površinu slike od gornjeg lijevog ugla, horizontalno desno tvoreći retke od vrha slike do dna, završavajući s donjim desnim uglom kao zadnjim pikselom. Prikaz slike na ekranu ili pisaču traži podatke o broju piksela u retku, broju stupaca, veličini piksela i položaju slike na stranici. Za interpretaciju slike važni su parametri: reprodukcija detalja na slici, zauzetost memorije računala, brzina procesora računala, ograničenja tiskarskog procesa. Slika u piksel grafici definira se nizom podataka koji određuju zacrnjenje pojedinog piksela u nizu. Tri su parametra: broj stupaca, broj redaka i broj bitova za sivu skalu. Dva su načina slaganja dvodimenzionalnog rasporeda piksela: potpuno iskorištenje podataka u stringu slike i nejednak broj piksela u slici s brojem podataka u stringu slike.

Zatim je navedena naredba *image* koja koristi pet parametara:

- broj piksela u retku čime je određen broj stupaca
- broj piksela u stupcu
- broj bitova po pikselu čime je određen broj stepenica sive skale
- transformacijska matrica slike
- string vrijednosti sivih razina slike

Gore spomenuta transformacijska matrica daje dimenziju piksela i nju možemo podesiti naredbom *scale* koja ju smanjuje ili povećava ovisno što želimo postići. Piksel možemo definirati i preko inverzne transformacije budući da su oni definirani kao paralelogrami pa svaki pojedini piksel može poprimiti zakrenute deformirane oblike. To je vrlo korisno znati ako radimo u photoshopu jer nam omogućuje razne transformacije kojima mijenjamo perspektivu željenih predmeta i oblika kao da su u trodimenzionalnom prostoru. To je u knjizi prikazano na primjeru slike GRGA koja je transformirana nekoliko puta. Nadalje, broj razina sive skale prva je karakteristika digitalne slike. PostScript ima mogućnost interpretirati sliku sa 2, 4, 16, 256 i 4096 stepenica sivih tonova. Naše oko prepoznaje pedeset nijansi sive boje što bi značilo da ako slika ima više stepenica sivih tonova mi to vidimo kao kontinuirani prijelaz nijansi. Također, što je veći broj stepenica sive boje to slika zauzima više memorije. Osim toga, u poglavlju se spominje i rezolucija. Rezolucija je bitna jer se preko nje zadaje veličina piksela, a izražava se u pikselima po inču. Tržišni softveri poput Photoshopa automatski zadaju dimenziju sliku prema tome. Također, ako želimo napisati nedozvoljenu dimenziju, softveri će ju pretvoriti u najbliži višekratnik dimenzije piksela. Rezolucija je iznimno važna u današnje doba. Svi teže što boljoj rezoluciji, a pogotovo pri ispisu, prikazu slike. Dobri rezultati ispisa slike su ako je rezolucija oko 300 ppi ili više. Budući da je kvaliteta otiska određena veličinom piksela, prilikom skeniranja fotografije, rezolucija skeniranja podređena je povećanju (smanjenju) originala i broju piksela po inču. Također, nećemo printati isto plakate koje gledamo iz velike udaljenosti ili male slike koje gledamo s udaljenosti od pola metra. Veličina piksela će biti puno veća na plakatu jer naše oko stvara sliku s jako velike udaljenosti pa nije potrebno imati male piksele na tako velikim dimenzijama. Te informacije su nam vrlo korisne u tisku, dizajnu i sličnim s kojima bismo se mogli susresti u budućnosti. Naposljetku, PostScript je zapravo svuda oko nas, susreli se s njime direktno ili ni ne znajući bio oko nas. On nam nudi mnoštvo mogućnosti kao što smo mogli vidjeti u knjizi, ali i na vježbama na našem fakultetu, a na nama je da odlučimo gdje i kako ćemo ga koristiti u budućnosti.