

Što je PostScript i što nam nudi?

PostScript je programski jezik za opis stranice, sličan Basic-u, Pascal-u te C-u, i, iako ima sličnu strukturu naredbi, namijenjen je samo području koje se bavi oblikovanjem grafike na stranicama za tisak. Njegov grafički jezik je specifičan jer je nastao razvojem računarske tipografije, fotosloga i računarske reprofotografije te je idealan alat za eksperimentiranje u tipografiji, grafici, slici, rasterima i bojama.

Za početak se postavlja pitanje, kako se koristi PostScript u području programiranja grafike? Za određivanje grafičkog oblika slova ili slike koristi se komandni jezik kojim se određuje tzv. „staza“ (put) te „operator“ i „operandi“ (parametri). Operatori, koji se još nazivaju i komande, pišu se kurentnim i verzalnim slovima (*image*, *show*, *fill*) ili kraticama (*arc*, *def*, *div*), a koristi se engleski jezik odnosno kratice izvedene iz engleskih riječi. Komande su većinom sastavljene od riječi ili kratica kompaktno napisanih kao jedna tipografska riječ (*setlinewidth*, *rlineto*). Operandi, ili parametri, najčešće su numeričke veličine koje se pišu ispred komandi i određuju način djelovanja komande (gdje komanda počinje, završava, koliko traje, kako se prikazuje, kako djeluje). Kao razdjelna oznaka između parametara i komandi se jednakovrijedno koriste razmak između riječi, tabulator i kod za novi redak. Iako se niz komandi može kontinuirano pisati u jednom retku, upotreba tabulatora i „carriage return“ (cr) koda omogućuje preglednije pisanje programa. Također, za olakšavanje snalaženja u programu koristi se znak postotka %. On određuje početak komentara koji programeru služi kao podsjetnik, i on nema utjecaja na PostScript komande. Komentar se zaustavlja prelaskom u novi redak (cr). PostScript ima koordinatni sustav koji je zasnovan u mjernim veličinama „točka“ s X/Y koordinatama, i njegov početak je u donjem lijevom dijelu stranice. Standardna veličina točke određena je preko inča, koji je podijeljen u šest dijelova nazvanih pica, koja je dalje podijeljena u 12 točaka. Stoga, jedan inč ima 72 točke, a jedna točka je 0.353 mm, a budući da koristimo decimalnu podjelu mreža će imati 100 x 100 točaka ili 35.3 x 35.3 mm. Za pisanje programa koristi se tekst editor koji može zapisati tekst u čistoj ASCII formi (bez ikakvih tipografskih zahvata), i da bi se vidio rezultat može se poslati prema PostScript printeru sa programom za download s bilo kojeg operativnog sustava, ili se može vidjeti na sustavima koji podržavaju Display PostScript. Kada započinjemo s radom u PostScriptu, najčešće se kreće s crtanjem linije. Imamo komandu *moveto* koja postavlja početak novog puta grafike u točki koja je određena parametrima koji se pišu ispred same komande, i iz toga možemo povući liniju (komanda *lineto*) u proizvoljnom smjeru ili čak isključivo horizontalno ili vertikalno (*rlineto*). Važno je napomenuti da nemaju sve komande parametre, recimo komande *stroke* (kojom omogućujemo prikazivanje linija) i *showpage* (briše postojeće stanje i postavlja parametre za ispis sljedeće stranice). PostScript daje puno opcija za oblikovanje već same linije, možemo lako odrediti debljinu linije, oblikovati početni i završni oblik linije (tri različite opcije), te za manipulaciju odnosa između više linija (*setlinejoin*). Moguće je linije prikazati kao pune, ali i kao crtkane, točkaste i kombinacija kraćih i dužih crtica. Ako želimo napraviti neki zatvoreni lik ne moramo sami odrediti točku zatvaranja nego komandom *closepath* program sam zatvara taj lik, koji zatim možemo ispuniti nekom bojom. Krug, kružnica, isječci kruga i njihovi dijelovi se rade komandama *arc*,

arcn i *arcto*, s time da je bitno napomenuti da luk u tim komandama ima smjer kretanja kazaljke na satu. Također, kako općenito u području vektorske grafike, tako i u PostScript jeziku, važan je pojam Bezierova krivulja, koja je polinom trećeg stupnja. Ona je osnova mnogih zaobljenih linija u grafici, a ovdje se njome manipulira pomoću komandi *curveto* i *arcto*. Nadalje, kod grafičkog dizajna često imamo likove s ponavljanim oblicima, to jest crtež koji se često ponavlja po nekom pravcu, krivulji, kružnici, simetriji, i tako dalje, i za to nam je vrlo korisna komanda *repeat*, koja se najčešće koristi kada precizno treba ponavljati neke oblike pa se taj posao prepušta računalu. Ako želimo zapamtiti put i način iscrtavanja da bi ga kasnije koristili, upotrebljavamo komande *gsave* i *grestore*, kako bismo mogli spremati više grafičkih stanja i restaurirati ih po redosljedu spremanja. Što se tiče transformiranja oblika, imamo opciju skalirati ga, rotirati (rotiramo mrežu i onda se programira rotirani oblik). Preklapanje oblika se može definirati na više načina te se čak može definirati i njihova različita ponašanja nakon komande *fill*. Kako bismo mogli lakše koristiti PostScript, bitno je znati da je on zasnovan na stack procedurama; interpreter dodaje novi objekt iz programa i postavlja ga na vrh stacka pritom gurajući prijašnje vrijednosti za jedno mjesto dublje. Postoje četiri vrste stacka: stack operanda (memorijski lanac podataka i rezultata za skoro sve komande), stack rječnika (uspostavlja parove između imena varijabli i procedura s njihovim sadržajem), stack grafičkih stanja (pamti programirani put i njegov način iscrtavanja) i izvršni stack (služi kao radni stack, i izvršava se samo ono što je na njegovom vrhu). Najbitniji je stack operanda. Najčešće komande za rad sa sadržajem stacka su *exch*, *dup*, *pop*, *clear*, *roll*, *neg*, *index* i *copy*. Procedure su skup rutina s korisnikovim imenom i upotrebljavaju se kao bilo koja komanda. Npr. ilustracija procedure okvir pokazuje se primjerom iscrtavanja pravokutnika sa zadanom samo polaznom i dijagonalnom točkom komandama *dup* i *neg*. Drugo rješenje koristi komandu *index*, kojom se duplicira i prebacuje neki podatak iz sredine stacka na vrh stacka. Zatim, bitno je spomenuti i programiranje mreža. Mogu se isprogramirati različite vrste mreža, npr. da se vide samo vodoravne linije, koliki je razmak među njima i slično. Što se tiče boja u PostScriptu, ima više načina kako definirati boju. Komanda *setcmykcolor* definira boju po CMYK kolor sustavu sa četiri parametra koji se zadaju u intervalu od 0.0 do 1.0, *setrgbcolor* definira boju po RGB sustavu s tri parametra koji se također zadaju u intervalu od 0.0 do 1.0, *sethcbcolor* definira boju po HSB sustavu s parametrom tona (hue), zasićenosti (saturation) i svjetline (brightness).

U drugom poglavlju bavimo se programiranjem tipografije. Slovni znakovi, to jest slova, oblikuju se unutar četverca, pravokutnika. Kada pišemo tekst i definiramo visinu slova, zapravo definiramo visinu četverca, a ne slike slova. Slika slova smještena je unutar četverca većinu vremena, no u nekim slučajevima slovni znakovi djelomično (npr. g, j, y...) ili potpuno (samostalni akcenti) izlaze iz njega. Dno četverca stoji na pismovnoj liniji, a donji lijevi ugao nulta je točka slovnog znaka. Debljinska vrijednost slova uključuje cijelu širinu slike slova sa dodatnom bjelinom do sljedećeg slovnog znaka. Nulta točka pozicioniranja sljedećeg slovnog znaka u tekstu je na točki debljinske vrijednosti prethodnog slova. Latiničko pismo je četverolinijsko, a sastoji se od pismovna linije, krova verzala, krova kurenta i descendera. Visina slike verzalnih slova (A, B, C, D...) je oko 70% četverca, visina kurentnih slova (a, c, e, i, m...) je do 50% četverca, spuštanje descendera u kurentnim slovima (j, g, y...) je od 20% do 30%. Dizajneri dodaju još četiri linije: dvije linije za krov nekih kurentnih slova (t, b, d, f...),

liniju akcenta (Ž, Š, Č, Ć) i liniju spajanja kurentnih slova nekih rukopisnih fontova. Neke od najbitniji komandi za programiranje tipografije su *findfont* (određuje traženje fonta koji treba biti na raspolaganju programu za slaganje teksta), *scalefont* (definira visinu četverca u točkama), *setfont* (postavlja font aktivnim) i *show* (prikazuje font na PostScript stranici). Bitna komponenta tipografije je čitljivost fonta, koja ovisi o pravilnom odabiru visine slova. „Formula“ koju možemo koristiti je relacija odnosa visine slova i udaljenosti gledanja pisma; visina verzala trebala bi biti tri milimetra ako se čita tekst s udaljenosti 30 centimetara, dakle odnos visine slike verzala i udaljenosti čitanja bi trebao biti 1:100. U tom primjeru kurenti bi trebali biti visine 1.5 milimetra. Što se širine retka tiče, neka se redak sastoji od 50 slovnih znakova. Bitno je naglasiti da se u slovne znakove ubraja i razmak između riječi, koji je zapravo najčešći slovni znak. Slova su u računarskoj grafici određena putanjom ovojnice na različite načine (pravci, dijelovi kružnice); PostScript koristi Bezierovu stazu. Slovni znak najčešće se prikazuje kao popunjen prostor omeđen unutarnjom i vanjskom ovojnicom, za što nam je dovoljna komanda *show*. Slova se također mogu ispisivati i tipa "outline", tj. samo linijama koje leže na Bezierovim putanjama, pri čemu se obavezno mora zadati debljina linije. Vrsta outline-a se definira logičkim sudom true ili false, i to ovisi o vrsti fonta koji se želi pretvoriti u outline - postoje fontovi koji su definirani kao outline programom i koji se u trenutku ispisivanja popune, zatim fontovi definirani nezatvorenim linijama i fontovi definirani bitmapom. Od te tri vrste, danas se upotrebljava samo prva. Ako želimo popuniti zatvoreni put nekim grafičkim rješenjem koristimo komandu *clip*, bez obzira na to je li osnova grafički lik ili ovojnica slova. Komanda *strokepath* omogućuje određivanje ovojnice linija (stazu oko linije koja ima zadanu debljinu). Dalje, ako želimo font transformirati u drugi font koristimo komandu *makefont*, koja to radi po transformacijskom polju (matrici), koje definira linearnu transformaciju koordinatnog para (x, y) u (x', y') . Ako želimo vodoravno spacionirati tekst, to radimo komandom *ashow*. Parametri dx i dy su relativni pomaci po x , odnosno y osi u odnosu na pismovnu liniju za svaki znak iz zadanog stringa. Kada je definiran samo dx parametar, a dy je 0 tada se radi o horizontalnom spacioniranju. Naredba *kshow* naprednija je od svih dosadašnjih komandi za kontrolu ispisa znakova iz stringa; daje mogućnost izvršavanja programske procedure između svakog znaka stringa: 1. ispiše se prvi znak stringa, a tekuća pozicija se pomakne sa širinu tog znaka; 2. na stack se postavlja dekadski ASCII kod prvo prvog znaka, a zatim i drugog znaka; 3. izvršava se procedura {proc}; 4. Ispisuje se drugi znak... i tako dalje. Nadalje, PostScript posjeduje mnoge aritmetičke operatore (*sub*, *add*, *mul*, *div*...), stack operatore (*exch*, *dup*, *pop*...), relacijske operatore (uspoređivanje dva elementa na stacku, rezultat te usporedbe je logički sud true ili false), logičke operatore, operatore uvjetnog izvršavanja procedura, operatore polja, string operatore i tako dalje. Za raspored teksta na stranici koriste se procedure. Procedura L_{TEKST} prikazuje tekst poravnat s lijeve strane, počevši od zadane pozicije, i zapravo se sastoji samo od *moveto* i *show* komandi. Slično tako postoje i komande C_{TEKST} i R_{TEKST}, koje centriraju tekst, odnosno vrše negativni relativni pomak teksta s *rmoveto* po x koordinati za punu širinu stringa. Procedura J_{TEKST} poravnava tekst na lijevu i desnu stranu pismovne linije. Procedura J_{VOKAL} prepoznaje vokale (samoglasnike) u tekstu, i prije nje na stacku mora biti znak (string od jednog elementa) kojeg ona testira da provjeri je li iz skupa vokala i na stacku ostavlja rezultat u obliku suda true ili false. Zadnje što ćemo spomenuti u vezi programiranja tipografije u PostScriptu je prijelom teksta. Procedura L_{PRELOM}, u tekstu, preuzima poziciju

početka prve pismovne linije i u proceduru PRELOM šalje riječ po riječ. Prije poziva procedure LPRELOM u glavnom programu se mora definirati početak prve pismovne linije s *moveto* naredbom i tekst postaviti na vrh stacka.

Zadnje poglavlje se bavi temom programiranja piksel grafike. Za primjer ćemo uzeti sliku sastavljenu od sivih kvadratića nazvanih pikseli (picture element). Za svaki pojedinačni piksel karakteristično je da na cijeloj svojoj površini ima jednoličnu sivoću, čija je vrijednost zadana jednim brojem u jednom bajtu, što znači da je raspon od bijelog do crnog razdijeljen nekontinuirano na 265 stepenica sivog tona. U digitalnoj reprofotografiji koristi se termin "siva skala" kako bi se razlikovao od kontinuiranog tonskog prelaza sivoće kakva se nalazi u tradicionalnoj fotografskoj tehnici. Broj razina sive skale prva je karakteristika digitalne slike. PostScript ima mogućnost interpretirati sliku sa 2, 4, 16, 256 i 4096 stepenica sivih tonova. Ljudsko oko razlikuje oko 50 nijansi sive skale, i zato mi podjelu od 256 stepenica sivog tona doživljavamo kao jedan kontinuirani ton. Reprodukcijska fotografija danas je najčešća s 8 bitnim razinama sivog. Kao što je spomenuto ranije, ljudsko oko dobro razlikuje 50 stepenica sive skale za što je bilo dovoljno 6 bita ($2^6 = 64$). Standard od 8 bita proširen je i na boju RGB i CMYK. Za PostScript se slika određuje nizom brojaka koje 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 otisnutog piksela i položaju slike na stranici. Interpretacija slike tiskom, metodom piksela i raster uzimaju u obzir nekoliko faktora: reprodukcija detalja na slici, zauzetost memorije i brzina procesora računala, ograničenja tiskarskog procesa digitalnog i analognog. Slika, češće nazivana original, u računalo se najčešće unosi skeniranjem, pri čemu se određuje veličina piksela, tj. elementarna kvadratična površina od koje će se sastojati digitalni zapis slike. Dvodimenzionalni raspored piksela ima dva načina slaganja; prvi odgovara točnom iskorištenju podataka u stringu slike, a drugi se odnosi na nejednak broj piksela u slici s brojem podataka u stringu slike. Direktno definiranje širine i visine slike može se postići upotrebom komande *scale* prije komande *image* i prikladnom transformacijskom matricom, koja kad ju podesimo formira dimenziju slike kao jedinični kvadrat koji se skalira. Uvriježeno je da je piksel kvadratičnog oblika jer mnogi programi imaju samo takvu mogućnost manipulacije njim, no pikseli su definirani kao paralelogrami pa svaki pojedini piksel može poprimiti deformirane oblike. U standardnim programima za piksel grafiku, npr. u Photoshopu, piksel na ekranu je uvijek kvadratične dimenzije. Veličina piksela zadaje se implicitno preko pojma rezolucije, odnosno brojem piksela po nekoj dužinskoj mjernoj jedinici (po inču ili centimetru). Na primjer rezolucija 400 ppi (pixl per inch) određuje kvadratični piksel čija je stranica velika 0.18 tipografskih točaka (1 inch = 72 tp). Ako se širina i visina cijele slike u takvim programima ne izrazi kao višekratnik od 0.18 tp tada se ulazi u za njih nedozvoljeno stanje koje se može ispraviti jedino zaokruživanjem na najbliže višekratnike od 0.18 tp. To tržišni softveri kao Photoshop rade pri upisivanju željene dimenzije slike automatski pretvarajući nedozvoljeno upisanu dimenziju u najbliži višekratnik dimenzije piksela. Veličina piksela određuje se ispisom, tj. prikazom slike. Dobri rezultati se dobiju ako je ispis slike oko 300 piksela po inču.