

Osvrt na knjigu PostScript

ŠTO JE POSTSCRIPT ?

Postscript je programski poput „*Basica*“, „*Pascala*“ i „*C*“, ujedno je idealan alat za sve one koji imaju volju eksperimentirati u računarskoj tipografiji, grafici, slici, rasterima, i bojama. Za **PostScript** ne treba imati nikakvo posebno predznanje, već samu sklonost prema grafičkoj umjetnosti i tipografiji, on kao takav ima svoj specifičan grafički rječnik koji je nastao razvojem računalske tipografije, fotografskog sloga te računalske repro-fotografije.

Koordinatni sustav **PostScripta** zasnovan je u mjernim veličinama točka sa x/y koordinatama, čiji je početak u donjem lijevom dijelu stranice. Standardnu veličinu točke odredili smo preko inča, on je podijeljen u 6 dijelova nazvanih pica (cicero), a pica u 12 točaka. On tako ima 62 točke od kojih svaka iznosi 0.353 mm. Decimalna podjela korištena na tome primjeru ima mrežu 100 x 100 točaka ili 35.3 x 35.3. Program i grafike u prvim primjerima su rezultat programskih komandi prikazanih u crnoj boji. Rasteri koji dočaravaju sivilo, a rezultat su programskih komandi također su prikazani crnom bojom. Zbog lakšeg snalaženja u položaju grafike, preko grafike je dodan koordinatni sustav u plavoj boji, a komentar i nužne numeričke veličine dočrtane su crvenom bojom kako bi se pojasnile.

PROGRAMIRANJE GRAFIKE

PostScript program ima specifičan rječnik, komandni rječnik kojim se određuju staza/put, operatori/komande pisani/ne na engleskome jeziku te operandi/parametri. Nekada se radi o zasebnim riječima poput „*image*“ i „*fill*“, o kraticama („*def*“, „*div*“), a najčešće se koriste kompaktne sastavnice od više riječi – „*rline*“, „*moveto*“, „*lineto*“, „*closepath*“, „*setlinewidth*“ i slično. Parametri određuju način djelovanja komande, poput njenog početka, trajanja, prikaza i djelovanja.

Svi programski jezici imaju slične karakteristike, odnosno sličnu strukturu i unos komandi, no **Postscript** je namijenjen isključivo za područje oblikovanja grafike u tisku. On kao i svi ostali jezici ima niz komandi koje se koriste za određivanje grafičkog oblika slova ili slika, takozvani komandni jezik, kojim se određuje staza, put, operator i operandi (parametri). Operateri, odnosno komande, pisane su kurentnim i verzalnim slovima na engleskome jeziku - npr. „*image*“, „*show*“, „*fill*“; ili pak kraticama – „*arc*“, „*def*“, „*div*“, izvedenih od engleskih riječi. Parametri, odnosno operandi, numeričke su veličine najčešće pisane ispred komandi, određuju način na koji je reagirati komanda – npr. gdje neka komanda počinje, a gdje završava, koliko ona traje, kako se prikazuje te kako djeluje. U pisanju se jednako koriste razmak između riječi, tabulator odnosno kod za novi redak, tj. razdjeljna oznaka između parametra i komandi. Niz komandi može se kontinuirano pisati u istom retku, preglednije pisanje programa omogućiti će nam upotreba tabulatora i koda „*carriage return*“ („*cr*“), dok će nam znak postotka % služiti kao podsjetnik koji će nam pomoći pri

određivanju početka komentara, ili će nam olakšati snalaženje u programu. Taj komentar nema utjecaja na **PostScript** komande te prestaje aktiviranjem („*cr*“) prelaza u novi redak.

Svi primjeri za **PostScript** jezik pišu se u tekst editoru, koji ima mogućnost zapisivanja teksta u čistoj **ASCII** formi. Kako bi se rezultat pisanog **PostScripta** programa uvidio, može ga se poslati u PostScript printer koji ima program za preuzimanje operativnog sustava, ili se može vidjeti na sustavima koji podržavaju **Display Postscript**. Najčešći početak **Postscript** stranice je „*move to*“ komanda. Ta komanda predstavlja početak novog puta grafike u točki koja je određena parametrima pisanim ispred komande „*moveto*“. Sama komanda nije prikazala neka točku na ekranu već je samo odredila početak nekog puta.

Prvi primjer oblikovat ćemo tako da nacrtamo nekoliko povezanih pravaca. Dužine se povlače od točke do točke koje su određene sa obje koordinate: horizontalna i vertikalna. To omogućuje iscrtavanje istim alatom kose, vodoravne i okomite linije. Od točke koja je posljednja određena, povlači se dužina do točke koja je definirana komandom „*lineto*“. Premda komandama određujemo put spojivši četiri točke, nije sve postalo vidljivo. Sama linija je bezdimenzionalna dok joj se ne pridruži debljina. Komanda „*stroke*“ omogućuje prikazivanje linija. Sama komanda nema parametara. Ako prethodno ništa nije određeno tada će se iscrtati crna linija debljine jedne točke. Grafička stranica šalje se na ispisni uređaj bilo to ekran, pisač, fotoosvjetljivač, ovisno o tome kako je ispis zamišljen da se realizira. Parametar koji određuje intenzivnost sive boje – 1 označuje suprotno od prirodnog tiska, tj. potpuno svjetlo, odnosno bijelo. Vrijednost parametra 0.9 određuje 90% svijetli ton a vrijednost nula je isijavanje svjetline to jest lik će se prikazati u crnom tonu. **Postscript** tipografija je riješena

Bezierovim krivuljama. Ilustriramo rukopisno slovo Ž iz fonta Shelly iz FS podataka. Podaci za krivulje su prepisani iz originalnog slova pomoću **Fontographera**. Grafike rađene pomoću računala prepune su likova s ponovljenim oblicima. Dizajneri najčešće planiraju ponavljanje crteža po nekom pravcu, krivudavom putu, kružnici, simetriji. Komande „*repeat for*“ imaju u **Postscriptu** slična pravila kao i u drugim programskim jezicima.

Postoje četiri vrste „*stacka*“ – „*stack*“ parametara, rječnika, grafičkih stanja i izvršni „*stack*“. Matematičkim operacijama su potrebni određeni parametri. Zbrajanje uzima dva podatka s vrha *stacka*, zbraja ih, a rezultat stavlja na vrh „*stacka*“. Slično se ponašaju komande za oduzimanje, dijeljenje i množenje. Boja se može odrediti komandom „*setcmykcolor*“ koja se koristi CMYK sustavom i za koju su potrebna 4 parametra, pomoću „*setrgbcolor*“ u kojem se slučaju određuje pomoću tri parametra u RGB sustavu, ili komandom „*sethsbcolor*“ koja definira boju s obzirom na ton (**hue**), zasićenost (**saturation**) i svjetlinu (**brightness**).

PROGRAMIRANJE TIPOGRAFIJE

Tipografija je proces kreiranja teksta i njegove forme, pri čemu se poštuju dobro proučene stare grafičke norme izrade. Zadatak tipografije je prijenos informacije pomoću pisma na najbolji i najjednostavniji način. Osnovni element tipografije je slovni znak: verzal, kurent, kurziv, brojke, posebni znak, interpunkcija i razmak između riječi. Slovni znakovi oblikovati

će se unutar četverca, tj. unutar pravokutnika u kojemu će se predstavljati slika slovnog znaka i njegova geometrija. Tijekom pisanja teksta, kada definiramo visinu slova, mi definiramo visinu četverca, a ne visinu slike slova. Slika slova, za veličinu slova, smještena je unutar četverca (A, B, C, a, b, c,...), a neki slovni znakovi djelomično ili u cijelosti izlaze iz četverca. Dno četverca leži na pismovnoj liniji, donji lijevi ugao četverca je nulta točka slovnog znaka.

Latinično pismo je četverolinijsko: pismovna linija, krov verzala, kuranata 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..) je od 20 do 30%. Točke veličine za karakteristična slova i neke fontove se prethodno ilustriralo programom. Dizajneri su dodali još četiri linije, od kojih su dvije linije bila za krov nekih kurentnih slova (t, b, d, f, ...) liniju akcenata i liniju spajanja kurentnih slova nekih rukopisnih fontova.

Slova u računarskoj grafici, još od fotografskoga sloga treće generacije, određena su putanjom ovojnice na različite načine: pravci, dijelovi kružnice. **PostScript** koristi **Bézierovu** stazu. Slovni znak najčešće se prikazuje kao popunjen prostor omeđen unutarnjom i vanjskom ovojnicom. Za to nam je dovoljna komanda „show”. Slova se mogu ispisivati tipa obrubom, odnosno linijama koje leže na **Bezierovim** putanjama . Pri tome mora se znati debljina linije. Naredbom „*charpath*” stvaraju se ovojnice slova zadanog „*stringa*“ koje će se prikazati tek uporabom naredbe „*stroke*“. Tekst može ući u **PostScript** program preko ulazne tekstualne datoteke („*file*“) koja se prethodno napunila sa nekim tekst editorom, pisanjem rukom, programskim zapisivanjem ili generiranjem podataka.

PROGRAMIRANJE PIKSEL GRAFIKE

Ovaj dio poglavlja prikazati će nam sastav slike od sivih kvadratića, tj. piksela. Za svaki pojedinačni piksel karakteristično je da na cijeloj svojoj površini ima jednoliku sivu boju. Vrijednost sive boje, odnosno sivoće piksela daje jednim brojem u jednom bajtu a to znači da je raspon od bijelog do crnog razdijeljen nekontinuirano na 265 stepenica sivog tona. U digitalnoj repro-fotografiji koristi se termin „*gray scale*” odnosno siva skala kako bi se razlikovala od kontinuiranog tonskog prijelaza sivoće kakvu imamo u tradicionalnoj fotografskoj tehnici. Ljudsko oko razlikuje oko 50 nijansi sive skale pa razdiobu od 256 doživljavamo kao kontinuirani ton.

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.

Interpretacija slike tiskom, metodom piksela i raster, uzima u obzir nekoliko faktora: reprodukcija detalja na slici. Zauzetost memorije računala, brzina procesora računala, ograničenja tiskarskog procesa digitalnog i analognog, Slika češće nazivana original, obično se unosi u računalo skeniranjem. Pri tome se određuje veličina piksela, tj. elementarna kvadratna površina od kojih će se sastojati digitalni zapis slike. O detaljima ili strukturi originala unutar površine piksela, nakon skeniranja neće postojati nikakva informacija te će

razlikovanje tih detalja biti moguće jedino ponovnim skeniranjem i smanjivanjem elementarne površine čitanja, u prijevodu će se ta ista slika interpretirati s mnogo više podataka.

PostScript ima mogućnost interpretirati sliku sa 2, 4, 16, 256 i 4096 stepenica sivih tonova. Reprodukcijska današnje fotografije je najčešća sa 8-bitnim razinama sivog, budući da ljudsko oko razlikuje oko 50 stepenica sive boje, za što bi bilo dovoljno 6 bitova ($2^6=64$). 8-bitnim standard proširen je i na RGB i CMYK boju. Píksel je opisan kao prostor slike koji je na cijeloj svojoj površini istoga tona. Skeniramo li, prostor píksela integrirati će sve tonove na tome području te će ih usmjeriti u samo jednu vrijednost. U slučaju da je píksel grub, nakon skeniranja ili resempliranja zauvijek će se izgubiti detalji koji su manji od píksela. Veličinu píksela odrediti ćemo ispisom i prikazom slike, a dobre rezultate dobiti ćemo, ako je ispis slike oko 300 ppi.