

Osvrt na knjigu "PostScript"

Što je PostScript?

PostScript je programski jezik te ima sličnu strukturu komandi kao npr. Basic. Namijenjen je isključivo području oblikovanja grafike na stranicama za tisak. PostScript također ima jedinstven grafički rječnik koji je nastao razvojem računarske tipografije, fotosloga te računarske reprofotografije. PostScript jezik brzo i jednostavno prikazuje grafiku, a onda sekundarno prikazuje i njegovu apstraktnu strukturu.

I poglavlje – Programiranje grafike

Kako bi se odredio grafički oblik slova ili slike koristi se komandni jezik kojim se određuje "staza" ili put te "operator" i "operandi" (parametri). Operatori, koji se još nazivaju "komande" pisane su kurentima i verzalima na engleskom jeziku ili kraticama koje su izvedene iz engleske riječi. Najveći dio komandi sastavljen je od nekoliko engleskih riječi ili kratica, te se riječi i kratice pišu kao jedna tipografska riječ (npr. setlinewidth, rlineto). Parametri (operandi) su uglavnom numeričke veličine pisane ispred komandi, oni određuju način djelovanje komande (npr. gdje komanda počinje, završava, koliko traje, kako se prikazuje, itd.) U pisanju se jednako koriste razmak između riječi, tabulator ili kod za novi redak, kao razdjelna oznaka između parametara i komandi. Niz komandi moguće je kontinuirano pisati u istom retku. Zbog upotrebe tabulatora i "carriage return" koda moguće je preglednije pisanje programa. Znak "%" određuje početak komentara koji služi programeru kao podsjetnik, ili pak za olakšavanje snalaženja u programu. Komentar prestaje aktiviranjem (cr) prelaza u novi red. Koordinatni sustav PostScripta zasnovan je u mjernim veličinama "točka" sa X/Y koordinatama, početak je u donjem lijevom dijelu stranice. Standardna veličina točke određena je preko inča. Uobičajeni početak PostScript stranice je "moveto" komanda koja postavlja početak novog puta grafike u točki određenoj parametrima pisanim ispred iste komande. Svaka linija koja se iscrtava sa "lineto" ili "rlineto" mora imati definiran početak sa "moveto" naredbom. Dužine se povlače komandom "lineto", a komanda "stroke" ih čini vidljivima. Moguće je bojanje linija i zatvorenih površina, postoje kombinacije različitih načina prekrivanja površina. a. Redoslijed pisanja komandi u ovim primjerima Na redoslijed prekrivanja površina utječe redoslijed pisanja komandi. Linija se uglavnom prikazuje kao puna, ali može ju se pretvoriti u crtkanu, točkastu ili kombinaciju kratkih i dugih crtica. Bezierova krivulja tvori osnovu mnogih zaobljenih linija (npr. ovojnica slovnih znakova). Bezierova krivulja je polinom trećeg stupnja te je definirana s četiri točke (prva - početna točka, zadnja točka krivulje i dvije tangentne točke koje određuju smjer napredovanja linije. Prva Bezierova točka se ne zadaje već je to ona točka koja je zadnja ostala u upotrebi. PostScript tipografija je riješena Bezierovim krivuljama. Grafike koje su rađene pomoću računala prepune su likova s ponavljanim oblicima. Dizajneri uglavnom planiraju ponavljanje crteža po nekom putu. U PostScriptu komande "repeat" i "for" imaju u slična pravila kao i neki drugi programski jezici. Ako želimo

postići da višestruko i precizno oblikujemo lik u pravilnom pomaku onda je računalo idealno da samo izračuna te korake. Programiranje grafičkog puta pomoću komandi kao što su "moveto", "lineto", "translate" i "rotate" definira jedno grafičko stanje u kojemu je zapamćena debljina linija, svjetlina, vrsta spajanja i sve ostale komande koje se brinu za način iscrtavanja puta. U PostScriptu se na više načina definira kako će se preklapati dva objekta i njihovo ponašanje nakon naredbe "fill". U PostScriptu postoje četiri vrste stacka: stack operanada, stack riječnika, stack grafičkih stanja i izvršni stack. Stack operanada je memorijski lanac podataka i rezultata za skoro sve PostScript komande. Stack riječnika uspostavlja parove između imena varijabli i procedura sa njihovim sadržajem. Stack grafičkih stanja pamti programirani put i njegov način iscrtavanja. PostScript izvršava samo ono što je na vrhu izvršnog stacka koji mu služi kao radni stack. PostScript je utemeljen na stack procedurama i to uglavnom za stack operanada. Interpreter dodaje novi objekt iz programa te ga postavlja na vrh stacka te gura prethodne vrijednosti u stacku za jedno mjesto dublje. Kada PostScript interpreter pokrene izvršenje komande uzima se podatak sa vrha stacka te se uklanja. Tada program čita parametre s lijeva na desno, a komanda uzima podatke s desna na lijevo. Procedure su skup rutina sa korisničkim imenom koje se upotrebljavaju kao i bilo koja PostScript komanda. Matematičke operacije imaju slične parametre kao i u drugi programski jezici. Simetrični kontinuitet Bazierove krivulje postiže se napredovanjem krivulje po tangenti suprotnog smjera, iste udaljenosti od posljednje Bazierove točke. Komanda "setcmykcolor" definira boju sa četiri parametra: cijan(c), magenta(m), žuta(y), crna(k). Komanda "setrgbcolor" definira boju sa tri parametra: crvena(r), zelena(g), plava(b). Komanda "sethsbcolor" definira boju sa parametrom tona, zasićenosti i svjetline. Da bi se postavio HSB model boja na programskoj traci isprogramirana su četiri kolorna kruga. Svaki od njih ima drugu svjetlinu, istu zasićenost te se unutar kruga mijenja ton boje.

II poglavlje – Programiranje tipografije

Slovni znakovi oblikuju se unutar "četverca". To pravokutnik unutar kojeg se postavlja slika slovnog znaka te njegova geometrija. Dok se piše tekst, odnosno kada se definira visina slova zapravo se definira visina četverca, a ne visina slike slova. Slika slova je smještena unutar četverca, a kod nekih slovnih znakova dijelovi djelomice ili potpuno izlaze iz četverca. Dno četverca leži na pismovnoj liniji, donji lijevi ugao je nulta točka slovnog znaka. Debljinska vrijednost slova uključuje cijelu širinu slike slova sa dodatnom bjelinom do slijedećeg slovnog znaka. Latiničko pismo je četverolinijsko: pismovna linija, krov verzala, krov kurenta i descender. Visina slike verzalnih slova je oko 70% četverca, visina kurentnih slova je do 50% četverca, spuštanje descendera u kurentnim slovima je od 20% do 30%. Dodane se još četiri linije: dvije linije za krov nekih kurentnih slova, linija akcenta i linija spajanja kurentnih slova nekih rukopisnih fontova. Fontovi se razlikuju po debljinskim vrijednostima i po tome što nemaju istovrsni set znakova. Zato se moralo definirati više setova znakova koji su se trebali prikazati. Čitljivost ovisi o pravilnom odabiru visine slova. Slova u računarskoj grafici su određena putanjom ovojnice na razne načine. PostScript koristi Bezierovu stazu. Slovni znak uglavnom se prikazuje kao popunjen prostor omeđen unutarnjom i vanjskom ovojnicom. Slova se mogu ispisivati samo linijama koje se nalaze na

Bezierovim putanjama. Pri tome se mora zadati debljina linije. Pomoću logičkih sudova "true" ili "false" definira se vrsta outline koja će se dobiti što ovisi o vrsti fonta koje želimo pretvoriti u outline. Postoje fontovi koji su definirani kao outline sa PostScript programom koji se popune u momentu ispisivanja te fontovi čiji su znakovi definirani nezatvorenim linijama ili fontovi definirani bitmapom. Danas se upotrebljava samo prva vrsta fonta (outline font) za koju je rezultat naredbe "charpath" jednak i za "false" i "true" logički sud. Višeslojno iscrtavanje ovojnice s raznim debljinama, zacrnjenjem, bojama i redoslijedom prikaza, omogućuje veoma kompleksna rješenja. "Charparh" se može kombinirati sa linijama bilo kojeg geometrijskog lika (npr. kvadrat, krug, slobodna Bezierova linija). Linija se mora proširivati okomito na svoju putanju jednako prema centru i prema van. Popunjavanje zatvorenog puta nekim grafičkim rješenjem postiže se komandom "clip" i to bez obzira je li osnova grafički lik ili ovojnica slova. Prema toj naredbi odnosimo se kao prema maski, prozoru ili izrezu za neki dizajn. Kada se iza programiranih zatvorenih staza upotrijebi naredba "clip" sve ono što se nadalje programira biti će vidljivo samo kroz stvorenu masku. Komanda "strokepath" omogućuje određivanje ovojnice linija, odnosno stazu oko linije koja ima zadanu debljinu. Staze nastale komandom "strokepath" sastoje se od poligona. U PostScriptu tekst se može proširiti na zadani puni format povećavajući razmak između slova. PostScript sadrži aritmetičke operatore (sub, add, mul, div, itd.), stack operatore (exch, dup, pop, itd.), relacijske operatore, logičke operatore, operatore uvjetnog izvršavanja procedura, operatore polja, string operatore i još mnoge druge. Pomoću relacijskih operatora se uspoređuju dva elementa na stacku. Rezultat te usporedbe je logički sud koji je istina ili laž te se pojavljuje se na stacku sa riječima "true" ili "false". Te logičke sudove najčešće koriste operatori uvjetnog grananja "if" i "ifelse", te logički operatori "not", "and", "or" i "xor". Stvaranjem biblioteke procedura olakšavamo programiranje složenih PostScript programa. Plavom bojom označene su procedure koje će se kasnije samo pozvati po imenu. Da bi se takva procedura mogla pravilno upotrijebiti kao gotova naredba treba korisniku dati definiciju argumenata na stacku prije upotrebe procedure i rezultata na stacku nakon upotrebe procedure. Sa procedurom "CISCVEKTOR" mogu se izbaciti svi jednaki elementi iz zadanog vektora. Sadržaj vektora može biti raznovrstan. Prije poziva procedure na stacku mora biti ulazni vektor koji se želi pročitati, a iza izvršenja procedure na stacku ostane pročišćeni vektor. Vektori sa dva člana pročišćavaju se jednom usporedbom. Vektori s jednim članom ili bez članova (nul vektor) prolaze kroz cijelu proceduru netaknuti i opet se na kraju pojavljuju na vrhu stacka. Tekst ulazi u PostScript program preko ulazne tekstualne datoteke koja se prethodno napunila sa nekim tekst editorom, pisanjem rukom, programskim zapisivanjem ili pak generiranjem podataka. Često je potrebno izbaciti neki član vektora po zadanom indeksu odnosno poziciji na kojoj se član nalazi unutar vektora. Prvi član vektora je na poziciji 0. Tu nastupa procedura "IZBACICLAN" kojoj su argumenti na stacku ulazni vektor i indeks, a njen rezultat na stacku je reducirani vektor za izbačeni član po zadanom indeksu.

III poglavlje – Programiranje piksel grafike

Slika je sastavljena od sivih kvadratića koji se zovu piksel (picture element). Za svaki pojedinačni piksel karakteristično je da na cijeloj svojoj površini ima jednoličnu sivoću. Vrijednost sivoće piksela zadana je jednim brojem u jednom bajtu što znači da je raspon od bijelog do crnog podijeljen nekontinuirano na 265 stepenica sivog tona. Kod digitalne reprofotografiji koristi se termin "siva skala" kako bi se razlikovala od kontinuiranog tonskog prelaza sivoće kakvu imamo u tradicionalnoj fotografskoj tehnici. Slika je određena nizom brojaka koje pravilno grade ukupnu površinu slike od gornjeg lijevog ugla, horizontalno desno tvoreći retke od vrha slike do dna te završavajući s donjim desnim uglom kao zadnjim pikselom. Prikaz slike na ekranu ili pisaću traži podatke o broju piksela po retku, broju stupaca, veličini otisnutog piksela i položaju slike na stranici. 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. Slika, uglavnom nazivana original, unosi se u računalo skeniranjem. Tada se određuje veličina piksela, odnosno elementarna kvadratična površina od kojih će se sastojati digitalni zapis slike. Na površini svakog elementarnog djela slike integralno se čita zacrnjenje i taj broj se pridruži pikselu. O detaljima ili strukturi originala unutar površine piksela, nakon skeniranja, ne postoji nikakva informacija. Razlikovanje tih detalja moguće je ponovnim skeniranjem, smanjivanjem elementarne površine čitanja, što znači da se ista slika interpretira s mnogo više podataka. Slika u piksel grafici definira se nizom podataka koji određuju zacrnjenje pojedinog piksela u nizu. Veličina piksela određuje se ispisom, prikazom slike.