

## Osvrt na treće predavanje – PostScript

### Što je PostScript i što nudi?

PostScript je programski jezik namijenjen samo području oblikovanja grafike na stranicama za tisak. Ima sličnu strukturu kao drugi programski jezici (Basic, Pascal, C, itd.). Njegov specifični grafički rječnik nastao je razvojem računarske tipografije, fotosloga, i računarske reprofotografije.

PostScript nam nudi mogućnost oblikovanja grafičkih slova koristeći komandi jezik. Njime se određuju *staze* ili put te *operatori* i *operandi*. Operatori, često nazivani i komande, pisani su kurentnim i verzalnim slovima na engleskom jeziku (na pr. fill, show). Mogu se još pisati i u obliku kratice izvedene iz engleskog jezika, a većina komandi je sastavljena i od riječi i od kratice kompaktно napisane kao jedna tipografska riječ (npr. setlinewidth). Operandi ili parametri su brojčane veličine koje su pisane ispred komandi. Oni određuju način na koji će komande djelovati. Bez njih komanda nema nikakvu svrhu u kodu. Njima možemo odrediti gdje komanda započinje, gdje završava, koliko traje te kako se prikazuje i kako djeluje. Kako bi kod izgleda uredno i pregledno prilikom pisanja parametara i komandi koriste se razmaci među riječima, tabulatori ili kod za novi redak. Iako se niz komandi može kontinuirano pisati u jednom redu korištenjem tabulatora i *carriage return* (cr) koda omogućuje preglednije pisanje i lakše snalaženje u kodu. Za bolje snalaženje u programu koriste se i komentari. Pomoću njih možemo kod učiniti jasnim i onome programeru koji nije nužno pisao kod, a ne samo nama samima koji kod pišemo. Koristeći komentare možemo odvojiti različite segmente koda te opisati napisani kod bez da taj komentar utječe na program. Komentar se označava %, a aktiviranjem (cr) prelaska u novi red završavamo isti komentar.

Da bi se bolje snalazili u položaju grafike unutar programa kao pomoć nam služi koordinatni sustav koji se prikazuje u plavoj boji. Program te mreže objašnjen je tek sa rutinama ponavljanja i petlje. Koordinatni sustav PostScripta zasnovan je na *točkama* sa X i Y koordinatama. Početak koordinatnog sustava je u donjem lijevom kutu stranice. Standardna veličina točke određena je preko inča. Inč je podijeljen u 6 dijelova tj. pica (pajk - engleski cicero), a pica u 12 točaka. Tako jedan inč ima 72 točke, odnosno jedna točka je 0.353 mm. Stoga ako imamo mrežu 100 x 100 točaka ona će u milimetrima iznositi 35,3 x 35,3 mm.

Pošto se svi primjeri pišu u tekst editoru kada želimo vidjeti rezultat našeg pisanog PostScript programa to možemo tako da pisani program pošaljemo PostScript printeru koji ima program za download ili preko sustava koji podržava Display PostScript. Jedan od tih sustava je Ghostscript koji je PostScript Level 2 interpreter za Windows, Mac, Unix, Amiga i Atati platforme. Unutar tih sustava rezultati komandi i parametara prikazani su crnom bojom. Rasteri koji dočaravaju sivilo također su prikazani crnom bojom. Dok su komentar i neke brojčane veličine dodane crvenom bojom.

## 1.poglavlje: Programiranje grafike

Najjednostavniji predmet za crtanje bila bi linija. Kod definiranja linije u PostScriptu potrebno je definirati njezin početak u koordinatnom sustavu. Početak linije označavamo komandom **moveto** ispred koje upišemo parametre. Zatim komandom **lineto** određujemo kolika će biti duljina linije. Nakon određivanja polazne točke, pomicanje do sljedeće točke za neku udaljenost horizontalno ili vertikalno postiže se komandom **rlineto**. Višestruko korištenje komande otežava precizno računanje položaja početne točke. Trebali bi sada izračunati koliki je pomak gore ili dolje, lijevo ili desno da dođemo do početne točke. Zatvaranje lika demonstriramo komandom **closepath**. Komanda nema parametre već potraži polaznu točku u nizu prije zadnjeg pozicioniranja startne točke.

Iako je linija definirana ona neće biti vidljiva dok ne odredimo njezinu debljinu koristeći komandi **stroke**. Ako debljina linije nije definirana parametrom linija je onda debela jednu točku. Željena debljina linije određuje se komandom **setlinewidth**. Početni i završni oblik linije postaje uočljiv kada imamo deblje linije. Zato ih možemo oblikovati komandom **setlinecap**. Komanda ima jedan parametar koji može imati 3 vrijednosti: 0 - kvadratni rub koji završava s definicijom linije; 1 - zaobljeni rub radijusa jednakog polovici debljine linije; i 2 - kvadratni završetak ali produžen za polovicu debljine linije.

Nakon određivanja debljine linije ona se prikazuje ispunjena, ali moguće ju je pretvoriti u crtkanu, točkastu ili kombinaciju kraćih i dužih crtica. Izmjenično iscrtavanje crnih i bijelih crtica postiže se komandom **setdash**. Koristeći komandu **setlinecap** mogu se dobiti ostale varijacije linija. Osim linija možemo definirati i kružne oblike koristeći komande **arc**, **arcn** i **arcto** te krivulje koristeći komandu **curveto** s kojim smo se upoznali u prethodnim predavanjima.

Likove u koordinatnom sustavu možemo i rotirati, povećavati te duplicirati koristeći komande **rotate**, **scale** i **repeat**.

Ukoliko želimo ispuniti lik ili obojati ga koristimo komandu **fill** dok za mijenjanje nijanse sive boje koristimo komandu **setgray**. Ona ima jedan parametar koji služi da određivanje svjetline. Broj 0 označava najtamniju nijansu sive odnosno crnu boju. Svako povećanje od 0.1 posvijetliti će nijansu za 10% dok ne dođemo do broja 1 koji daje najsvjetliju nijansu sive – bijelu. Osim sivih tonova likove možemo obojati koristeći komande za različite sustave boja. Komanda **setcmycolor** definira boju sa četiri parametra: cijan, magenta, žuta i crna. Komanda **setrgbcolor** definira boju sa tri parametra: crvena, zelena i plava, komanda **sethbscolor** definira boju sa tri parametra koja se odnose na ton, zasićenost i svjetlinu boje. Programskim mijenjanjem parametara color komandi možemo dizajnirati različite prijelazne efekte među bojama. Koristeći for petlju sa promjenom nekog parametra lika npr. debljine linije ili radijusa kružnice mijenja se i neki odabrani parametar boje npr. hue odnosno ton.

## 2.poglavlje: Programiranje tipografije

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.

Svi slovni znakovi definirani su unutar četverca. To je pravokutnik unutar kojeg se postavlja slika slovni znakova i njegova geometrija. Dno četverca leži na pismovnoj liniji, a donji lijevi kut četverca je nulta točka slovnog znaka. Debljina slovnog znaka uključuje cijelu širinu slova skupa sa bjelinom dok idućeg slovnog znaka. Latinično pismo je četverolinijsko: pismovna linija, krov verzala, krov kurenta i descender. Osim ovih linija dizajneri su dodali još četiri linije: dvije za krov nekih kurentnih sola kao što su t, b, f ; liniju akcenta (Č, Ć, Š, Ž) i liniju spajanja kurentnih i slova rukopisnih fontova. Visina slike verzalnih slova je oko 70% četverca, a visina kurentnih je od 20% do 30%.

Komanda **findfont** određuje traženje fonta koji treba biti na raspolaganju programu slaganja teksta, **scalefont** definira visinu četverca u točkama, **setfont** postavlja font aktivnim i **show** prikazuje fonta na postscript stranici.

Čitljivost teksta ovisi o pravilnom odabiru visine slova. Savjetuje se da se visina slova određuje ovisno o udaljenosti čitanja. Ako tekst čitamo sa udaljenosti od 20 metara visina verzala trebala bi biti 20centimetara, a visina kurenta 10 centimetara. Odnos slike verzala i udaljenosti čitatelja je 1:100. Slično pravilo vrijedi i kod širine retka.

Slovni znakovi najčešće se prikazuju kao popunjeni prostor omeđen unutarnjom i vanjskom ovojnicom, ali ako želimo slova ispisati kao outline naredbom **charpath** definiramo ovojnici slova zadanog stringa koje će se prikazati tek upotrebom naredbe **stroke**. Charparh se može kombinirati sa linijama bilo kojeg geometrijskog lika kao naprimjer kvadrat, krug i slobodna Bezierova linija. Treba voditi računa da se linija proširuje okomito na njenu putanju jednako prema centru i prema van. Kada želimo popuniti zatvoreni put nekim grafičkim rješenjem koristimo naredbu **clip**. Ona ne stvara iza sebe mogućnost programiranja nove staze, već ako se to želi postići iza naredbi se piše naredba **newpath**. Komandom **strokepath** određujemo izgled staze oko koje linija ima zadanu debljinu.

Ako želimo tekst vodoravno proširiti na zadani puni format povećavajući razak između znakova koristimo komandu **ashow**. Naredba **kshow** je naprednija naredbe **ashow** i svih ostalih naredbi koje se tiču kontrole ispisa znakova iz stringa. Ona daje mogućnost izvršavanja programske procedure između svakog znaka stringa.

## 3.poglavlje: Programiranje piksel grafike

Piksel grafika temelji se na pikselu. Piksel je skraćenica za engleski pojam picture element. Za svaki pojedinačni piksel karakteristično je da na cijeloj svojoj površini ima jednoličnu sivoću. Vrijednost te sivoće data je brojem u jednom bajtu a to znači da je raspon

od bijelog do crnog razdijeljen na 265 stepenica sivog tona. Broj razina sive skale je prva karakteristika digitalne slike. PostScript ima mogućnost interpretirati sliku sa 2, 4, 16, 256 i 4096 stepenica sivih tonova. Kada formiramo sliku od piksela ona je određena nizom brojki koje pravilno grade ukupnu sliku od gornjeg lijevog kuta, horizontalno desno tako tvoreći retke od vrha do dna slike. Slika, često nazivana original, obično se unosi u računalo skeniranjem. Pri tome se određuje veličina piksela to jest, elementarna površina od kojih će se sastojati digitalni zapis slike. Slika u piksel grafici definira se nizom podataka koji određuju zacrnjenje pojedinog piksela u nizu. Svaki piksel zapisan je preko 8 bitova. Raspored slaganja piksela unutar dvodimezionalne slike ima dva načina slaganja. Prvi način bio bi potpuno iskorištavanje podataka u stringu slike, a drugi način je nejednak broj piksela u slici s brojem podataka u stringu slike.

Naredba **image** koristi 5 parametara pomoću kojih formira sliku. Ti parametri moraju biti postavljeni na stacku. Prvi parametar je broj piksela u retku čime je određen broj stupaca  $S$ , zatim broj piksela u stupcu (broj redaka)  $R$ , onda broj bitova po pikselu  $G$  čime je određen broj stepenica sive skale. Četvrti parametar je transformacijska matrica slike i na kraju string vrijednosti sivih razina slike.

Prilikom određivanja veličine slike koristimo naredbu **scale**. Transformacijska matrica  $[1\ 0\ 0\ 1\ t_x\ t_y]$  daje dimenziju piksela kao kvadrata građenog od jedne točke. Time je širina slike od  $S$  piksela jednaka  $S$  točaka, a visine slike od  $R$  piksela je  $R$  točaka. Vidi se da je dimenzija slike preko matricom  $[A\ 0\ 0\ D\ t_x\ t_y]$  zadana tako da kvocijent  $S/A$  definira širinu slike, a kvocijent  $R/D$  visinu slike. Direktno definiranje širine i visine slike može se postići upotrebom naredbe **scale** prije **image** naredbe i prikladnom transformacijskom matricom. Tako podešena transformacijska matrica formira dimenziju slike kao jedinični kvadrat koji se skalira (povećava ili smanjuje) na dimenziju  $W$  i  $H$  točaka.

Veličina piksela određuje se ispisom slike. Dobri rezultati će se dobiti ako je ispis slike oko 300 piksela po inču. Budući da je kvaliteta otiska određena veličinom piksela, prilikom skeniranja fotografije, rezolucija skeniranja podređena je dvjema brojkama: povećanje (smanjenje) originala i broja piksela po dužinskom inču (najčešće 300). U obzir može biti uzet i treći faktor - udaljenost gledanja otiska.

Pomoću komande **colorimage** pikseli se mogu ispisivati kao sivi (Gray), RGB ili CMYK. Prva tri parametra unutar naredbe su istog značenja kao i kod **image** naredbe.  $S$  parametrom  $N$  definira se broj komponenata boje po pikselu. Za sivi zapis  $N$  je jednako 1, za RGB  $N$  je 3, a za CMYK  $N$  je jednako 4. RGB i CMYK komponente mogu se zapisivati na jednostruki i višestruki način. U jednostrukom zapisu se sve komponente zapisuju u jednom stringu naizmjenično piksel po piksel, a u višestrukome se komponente zapisuju u odvojenim stringovima kao u primjeru na ovoj stranici.