

Napravila: Iva Žitković

## Osvrt na PostScript

### Što je PostScript?

PostScript je programski jezik koji je namijenjen području oblikovanja grafike na stranicama za tisak. Sličan je drugim programskim jezicima kao što su Basic ili Pascal no ima svoj specifični grafički rječnik. Za dobivanje nekog grafičkog oblika slike ili slova u PostScriptu se služimo komandnim jezikom, operatorima i parametrima. Operatori odnosno komande pisane su kurentnim i verzalnim slovima na engleskom jeziku ili kraticama riječi. Bez obzira od koliko se riječi sastoji komanda one se pišu bez razmaka. Operandi odnosno parametri su najčešće brožčane vrijednosti pisane ispred komadi koje određuju njihov način djelovanja. PostScript radi preko standardne veličine točke izražene inčem smještene u x/y koordinatom sustavu čiji se početak nalazi u donjem lijevom dijelu sustava.

### Prvo poglavlje – programiranje grafike

U ovom se poglavlju upoznajemo s osnovnim funkcijama i jednostavnim naredbama za rad u PostScriptu. Kreće se od jednostavnih naredbi kojima dobivamo linije te uz nadogradnju slažemo zanimljive oblike. Prvo nas se upoznaje s osnovnim komandama kao što su moveto (naredba kojom obično započinjemo naše programiranje. Sama komanda neće dati nikakav prikaz već njome samo određujemo početnu poziciju naše točke unosom željenih parametara), lineto (stvara dužinu od početne točke do točke definirane u parametru) i stroke (određujemo debljinu, boju i sivilo te ju time činimo vidljivom na ekranu), kojima stvaramo linije te komanda showpag kojom brišemo prethodni rad i pripremamo se za novi. Slijedeća grupa komandi odnosi se na manipulaciju same linije. Neke od tih komandi su rlineto (

postizemo pomicanje početne točke za neku vertikalnu ili horizontalnu udaljenost), `setdash`, `closepath` (zatvaramo lik tako da računalo samo potraži početnu točku) i `setlinewidth`... Dobro je istaknuti da se linija neće prikazati na ekranu sve dok joj ne odredimo neku dimenziju kao što je širina, boja i/ili sivilo. U ovu grupu komandi svakako moramo smjestiti i naredbe `setlinecap`, `setlinejoin` i `setmiterlimit`. Njima određujemo način povezivanja s drugim linijama. Ako pak želimo ispuniti programirani lik nekom bojom koristiti ćemo komandu `fill`, a promjenu tona sive na liniji ili liku kontrolira komanda `setgray`. Bezierova je krivulja bila tema prošlog predavanja ali je bilo zanimljivo vidjeti kako se ona programira te kako parametri utječu na nju. Njeno crtanje nam omogućava komanda `curveto`. Početnu točku program traži u prethodnoj radnji dok se sljedeće tri upisuju ručno. Lak rad u PostScriptu omogućuju nam naredbe `repeat`, `for`, `scale`, te `rotate`. Njima jednostavno dupliciramo, mijenjamo veličinu i rotiramo nacrtani predmet što štedi vrijeme svakom programeru. Treba imati na umu kako sam PostScript ne pamti prije ubačene vrijednosti već nam tu pomažu komande `gsave` i `grestore`. Veliki plus PostScripta je stack koji služi kao neka mala memorija cijelog našeg rada. Pri radu sa stackom treba uvijek imati na umu koji su nam elementi ispred a koji iza, odnosno koje su varijable napisane prve s desne, a koje prve s lijeve strane. Ja ću ovdje navesti samo primjer zbrajanja (`sub`) u kojem sudjeluju dva podatka s vrha stacka, oni se zbrajaju te se sad njihov zbroj stavlja na vrh stacka. Ostale se matematičke operacije odvijaju na sličan način. Zatim prolazimo kroz 3 različita tipa mreža s kojima se možemo susresti pri radu u PostScriptu. Svaka je detaljno objašnjena te se na danim primjerima vidi jasna razlika među njima. Komande kojima određujemo boju ovise o sustavu boja koje želimo koristiti. Za CMYK sustav koristimo komandu `setcmykcolor`, za RGB sustavu koristimo `setrgbcolor` te za hue, saturation, brightness sustav komandom `sethsbcolor`.

## Drugo poglavlje – programiranje tipografije

Ovo je poglavlje u potpunosti posvećeno tipografiji odnosno fontu. Kreiranjem fonta smo se bavili na prvim predavanjima tako da mi ovo poglavlje nije bilo u potpunosti strano. Super mi je bilo povezivati praktična znanja iz FontForgea i ovisnost tih radnji o parametrima odnosno komandama zadanim u programiranju u PostScriptu. Ponavljanje pojmova poput četverca te proširenje

znanja o nekim normama pri kreiranju fonta (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%) dobro dođu za našu struku. I u ovom dijelu upoznajemo mnoge korisne komande kao što su findfont, scalefont, setfont, show... Novi pojam s kojim sam se susrela je string, on predstavlja setove znakova koje mi prepoznamo kao riječi, rečenice i drugo. Stringovi su važni u PostScriptu jer olakšavaju programu upisivanje raznih slovničkih znakova. Ovu povezanost Bezierove staze i stringa opisuje komanda charpath. Važnost stringova možda je najbolje vidljiva kod procedura za poravnavanje teksta; LATEX (ulijevo), RTEX (udesno), CTX (centrirano) i JTEX (obostrano). Ovdje program na temelju dužina izraza i razmaka među skupovima definira optimalno rješenje za estetski dobar izgled cijelog teksta za zadano poravnanje. Uz to stringovi igraju veliku ulogu u procedurama CRODIELI i LPRELOM. Oni omogućuju da se riječi na kraju retka podijele prema hrvatskim pravopisnim standardima i to na način da traže samoglasnik najbliži zadnjoj liniji retka, iza njega stavlja crticu a ostali skup znakova prebacuje u novi redak. Mogućnost manipulacije izgleda slovničkih znakova jednaka je mogućnostima manipulacija linija koje sam prethodno navela.

## Treće poglavlje – programiranje piksel grafike

U ovome posljednjem poglavlju knjige naglasak ima piksel, mala točkica određene obojenosti i sivog tona koja s drugim pikselima tvori sliku. Vrijednost sivoće piksela dana je na 255 stepenica sivog tona dok ljudsko oko razlikuje oko 50 nijansi sive skale pa razdiobu od 256 doživljavamo kao kontinuirani ton. U ovom poglavlju pratimo transformaciju piksela prilikom skaliranja, rotacije i drugih promjena. Dakle piksel mogu odrediti kao element slike koji je obojen jednom bojom, više piksela različitih boja čini sliku, a samim time više piksela dati će kvalitetniju sliku. Gustoću i veličinu piksela mi kao grafičari moramo znati rasporediti i primijeniti ovisno o tome gdje će se slika nalaziti. Na primjer ako imamo sliku na nekom jambo plakatu nećemo se morati zamarati tiskanjem u visokoj kvaliteti jer će se plakat gledati s udaljenosti pa će i ljudskom oku biti teže odrediti granicu pojedinih piksela. Ako pripremamo sliku za neki časopis ili internetski prikaz moramo voditi računa o gusto složenim sitnim pikselima. Isto se događa pri skaliranju slika. Ako odlučimo početnu sliku smanjiti, a

kasnije tu istu povećati izgubit ćemo na kvaliteti jer će program sam popuniti praznine na temelju ponavljanja podataka/piksela s početka stringa, a ne na temelju početne slike- neće vratiti piksele izgubljene smanjivanjem već će rupe nadomjestiti pikselima sličnima onima u blizini praznine.

