

Grafički Programski Jezici

Sedmi Osvrt Jere Rinčić

PIKSEL GRAFIKA U POSTSCRIPTU

Slika se sastoji od kvadratića tj. piksela i svaki piksel zauzima točnu poziciju i obojenje na ekranu. Svi pikseli kad su poredani u stupce i retke, a također da su i određenog obojenja, čine dvodimenzionalnu matricu tj. sliku koja je definirana širinom i visinom. Parametri koji su nam potrebni za ovakav proces su veličina i boja piksela, broj stupaca i redaka, i isto tako parametar koji određuje koliko će podataka jedan piksel uzeti da bi definirao boju. Njega češće nazivamo **color depth**, i ovisno koliko mu bitova dajemo toliko će biti raznovrsniji raspon nijansi boje.

Uzmimo na primjer 2-bitni color depth. Sa tom dubinom možemo dobiti 4 nijanse sivog tona, a broj kombinacija znamo putem formule 2^n , gdje je n broj bitova.

U samome kodu ovako vrijedi zapis informacija: *10 1 8 [1 001 00]*. Prvi broj označava broj piksela u stupcima, dok drugi u redovima. Treći označava bitove podataka, i nakon njega uglata zagrada definira transformacijsku matricu u kojoj se definira veličina svakog piksela. Jedinice definiraju jediničnu veličinu piksela, što uključuje visinu i širinu.

Također možemo imati i kod ograđen vitičastim zagradama (*npr. {<00FFEE99>}*) koji sadrži proceduru niza slovnih znakova. Razmak i interpunkcijski znakovi tvore crni piksel, a bilo koji string na kraju tvori boju. U primjeru gore je boja definirana u heksadecimalnom zapisu koji se najčešće koristi.

Piksel ne mora biti nužno kvadrat, već deformacijama možemo dobiti paralelograme te rombove. Ako dodamo komandu *div* u matricu ona će deformirati sljedeće piksele.

Nakon svega ovoga završavamo proceduru sa komandom *image*, ali moramo još spomenuti operator imena *colorimage*, prije kojeg inače mora stajati komanda *false*. Između njih pišemo broj koji govori o kojem

sustavu boja govorimo (npr. 3 je RGB, a 4 CMYK), a operator *false* označava na koji način će se obraditi vrijednosti boje za pojedini kanal.

Crvena boja bi se zapisala ovako:

```
4 1 8 [ 1 50 div 0 0 1 50 div 0 0 ]
```

```
{<FF0000>} false 3 colorimage
```

Za CMYK i RGB sustav možemo koristiti dva pristupa pri pisanju koda. Već smo isprobali metodu putem *false* operatora, a druga metoda je putem *true* operatora. U ovoj metodi svaki kanal će imati svoju zasebnu proceduru, pa ćemo morati imati 4 ili 3 zagrade a ne samo jednu kao kod *false* metode. U photoshopu možemo pretvoriti sliku u heksadecimalni tekst (EPS format) koji onda odmah možemo ubaciti u naš postscript kod između vitičastih zagrada, i tako dobiti sliku u toj proceduri. Naravno moramo paziti u kojem je sustavu boja slika. Ako je slika isključivo crno bijela onda proces nije toliko kompliciran. Sliku spremimo u EPS formatu, koji kad otvorimo u notepad-u izgleda kao i postscript kod. Kod u interesu započinje nakon komande *beginimage*. Širina i visina slike u photoshopu mora biti upisana naknadno pod parametre koji definiraju stupce i retke.

Visinu piksela je važno staviti negativnom ako je naša slika iz photoshopa bila u standardnoj orijentaciji, jer će inače slika u postscriptu biti naopaka zbog razlike u konstrukciji slike među dva programa.

Postupak je malo kompleksniji u slučaju da je slika višebojna (npr. CMYK ili RGB). Prvo ćemo trebati sliku pretvoriti u njen negativ, i onda spremiti u EPS format sa multiple file opcijom i ASCII encoding-om. Tako ćemo dobiti više file-ova koje onda kopiramo u određene zagrade u naš kod.