

PostScript I/O

mobytrick

11 gennaio 2017

Il linguaggio PostScript possiede alcune primitive che permettono la manipolazione dei file esterni. La gestione non è delle più semplici, ma con un po' di buona volontà e tante prove si riesce ad ottenere qualcosa.

I file scritti in PostScript sono in genere autarchici, in quanto contengono al loro interno tutti i dati necessari. Prendiamo un caso abbastanza semplice, qual è il tracciamento di un rettangolo. Sono necessari 4 dati: le coordinate del vertice in basso a sinistra, la base e l'altezza. I dati numerici vanno inseriti all'interno del programma, rendendolo quindi non parametrico. Mi sono quindi posto il problema di ovviare a questa rigidità scrivendo un file PostScript che legga i dati da un file di testo. La parametricità è ottenuta in quanto basta modificare preventivamente il file di configurazione.

Il sorgente del file sotto riportato legge da un file (che nello specifico si chiama `alfa`) 4 dati numerici, scritti separatamente su altrettante righe. Sono nell'ordine: ascissa ed ordinata del vertice in basso a sinistra, base ed altezza del rettangolo. C'è un unico controllo: devono essere stati letti 4 dati. Sono possibili delle miglorie, tipo gestione della mancanza del file e segnalazione se la base oppure l'altezza sono nulle.

La mia conoscenza del PostScript non è superficiale, ma nemmeno approfondita. Per la lettura mi sono valso di un `for` eseguito 4 volte. Però non sono stato capace di utilizzare l'indice (che pure si trova sullo `stack`) per distribuire l'input nelle variabili appropriate. Ho fatto ricorso ad una variabile di comodo. Il conoscitore del PostScript sa come intervenire in modo da snellire il programma.

```
%!PS-Adobe-3.0
/count_num 0 def
/Infile (alfa) (r) file def
```

```

/Linea 15 string def
0 1 3 {
  Infile Linea readline
  { % lettura
    count_num 0 eq
    { /Asc Linea cvi def }
    if
    count_num 1 eq
    { /Ord Linea cvi def }
    if
    count_num 2 eq
    { /Base Linea cvi def }
    if
    count_num 3 eq
    { /Alt Linea cvi def }
    if
    /count_num count_num 1 add def
  }
  { % eof
    Infile closefile
  }
  ifelse
} for
count_num 4 eq
{
  gsave
  newpath
  Asc Ord moveto
  Asc Base add Ord lineto
  Asc Base add Ord Alt add lineto
  Asc Ord Alt add lineto
  closepath stroke
  grestore
  showpage
}
{ (input insufficiente\n) print flush }
ifelse
%%EOF

```

Da qui in poi si assume che si operi in ambiente Linux.

Il file di cui sopra, per come è stato costruito, deve essere mandato in esecuzione. Non può essere usato **evince** che è un visualizzatore. Si ricorre perciò a **ghostscript** che si lancia in esecuzione col seguente comando:

```
gs -q
```

l'opzione **-q** indica la modalità **quiet** ovvero vengono soppressi alcuni messaggi iniziali in modo che la chiarezza dell'output ne guadagni. **ghostscript** apre una finestra accessoria ed emette il proprio prompt (**GS>**) rimanendo in attesa dei comandi. Per lanciare in esecuzione un file PostScript e supposto che questo si chiami **rettpar.ps**, il comando è:

```
(rettpar.ps) run
```

In assenza di errori, il rettangolo viene tracciato nella finestra accessoria. A meno di non voler immettere ulteriori comandi per uscire da **ghostscript** si usa **quit**. La fine della sessione comporta l'eliminazione della finestra accessoria.