Text and Graphics

CSE 413, Autumn 2005
Programming Languages

http://www.cs.washington.edu/education/courses/413/05au/

Postscript is a Page Description Language!

- In the previous lectures I talked mostly about how Postscript implements the standard constructs of a programming language
  - variables, control flow, procedures
- The main purpose of Postscript is to put marks on a page, using those constructs
  - text
  - graphics
  - images

Paths

- A drawing starts with a path on the current page
- path is a set of straight lines and curves that define:
  - a region to be filled (fill)
  - a trajectory that is to be drawn (stroke)
**basic path construction operators**

- **newpath**
  - initialize current path to be empty

- **closepath**
  - Connect subpath back to its starting point

- **moveto, rmoveto**
  - set current point to (x,y)
  - set current point to (curX+dx, curY+dy)

- **lineto, rlineto**
  - append straight line to (x,y)
  - append straight line to (curX+dx, curY+dy)

**Curve path operators**

- **arc, arcn**
  - append clockwise, counterclockwise arcs
  - x y r angle1, angle2 arc

- **arct, arcto**
  - append tangent arcs
  - x1 y1 x2 y2 r arct

- **curveto, rcurveto**
  - append Bezier curve
  - x1 y1 x2 y2 x3 y3 curveto

**saving the graphics state**

- Sometimes we need to save the current graphics state (including the path) so that we can reuse it
  - the stroke and fill operators clear the current path
  - blocks of code may change path, gray value, line width, user coordinate system, etc

- **gsave**
  - save a copy of the current state on the graphics state stack

- **grestore**
  - restore to the state at the time of the last save

**Text**

- Postscript treats text as just another way to define graphics paths
  - The content of the text is maintained in a string object
  - The visible representation of the text is determined by the font
  - Fonts are stored as a set of curves for each letter
    - the representation is the *glyph* for this character in this font

*Font: Tannarin BT*  
*Font design is fun!*

*Font: Murray Hill*  
*Font design is fun!*

Using a font

- Find the information describing the font
  - the info is in a font dictionary
  - use the findfont operator
- Scale the font to the size needed
  - original font is 1 unit high (usually 1 point)
  - use the scalefont operator to scale
- Set the scaled font as the current font
  - use the setfont operator

Show a text string

```
/Palatino-Italic findfont
15 scalefont
setfont
72 720 moveto
(Font design is fun!) show
```

```
/NewCenturySchlbk-Roman 15 selectfont
72 700 moveto
(Font design is fun!) show
```

```
/StandardSymL 15 selectfont
72 680 moveto
(Font design is fun!) show
```

Fun with fonts

- Postscript provides much more power for dealing with fonts
  - fonts are paths - they can be filled, stroked, clipped to, etc
  - there are several glyph painting operators that provide a variety of width modification effects
  - numerous font type definitions to support different ways of identifying the characters and defining the glyphs
“The name is Pond ... LilyPond”

```
\begin{verbatim}
\begin{verbatim}
\end{verbatim}
\end{verbatim}
```

- LilyPond is an "automated engraving system." It formats music beautifully and automatically, and has a friendly syntax for its input files.
  - input is done in the form of a textual music language
  - content (the music) and the layout are strictly separated
  - users can extend the program by using the built-in Scheme interpreter.
  - PostScript output is generated via the TeX typesetting system.

Graphviz

- graphviz is a set of graph drawing tools
  - dot - makes hierarchical layouts of directed graphs
  - neato - makes "spring" model layouts of undirected graphs
- Graphs are described in DOT language
  - abstract grammar defining DOT
    ```
    graph:[ strict ] ( [ graph | digraph ] ) [ ID ] '{' stmt_list '}'
    stmt_list:[ stmt ';' ] [ stmt_list ]
    ```
    etc
- Output in Postscript and other languages

http://lilypond.org/web/index.html

http://www.graphviz.org/