Project phase 5 — PostScript interpreter (graphics) — assigned Thursday 13 October, due Sunday 14 December

This is a continuation of phase 4.

In the tradition of 351 project assignments, this is only a first draft and is subject to change (to add further clarifications).

5.1 Task

Write the remaining Java classes and interfaces to implement the functions of a PostScript interpreter, so that it can execute many graphics programs.

5.2 Task in detail

In this phase, we implement PostScript graphics We implement only a small subset of the graphics capabilities of PostScript. Principally, we implement the PostScript graphics state and the operators used to manipulate it; in particular we implement the operators for querying and updating the current coordinate transformation matrix. We implement most path construction operators for simple paths, but we leave out fonts and many other features.

We do not implement the actual rendering to a raster device—that would require a significant amount of background knowledge in computational geometry and graphics, and a very large amount of Java code. Instead, we produce simplified PostScript code as output, which is then accepted by a viewer such as GhostScript.

5.2.1 Example translations from PostScript to simplified PostScript

Triangle and square This example involves no coordinate transformation operators and has only straight-line control flow.

Given this input:

```
newpath
100 100 moveto
200 0 rlineto
0 200 lineto
closepath
stroke

200 200 moveto
0 50 rlineto
50 0 rlineto
0 -50 rlineto
closepath
stroke
```

the following output is generated:

```
%!PS-Adobe-2.0
[ 1.0 0.0 0.0 1.0 0.0 0.0 ] 6 array defaultmatrix 6 array concatmatrix setmatrix
0.0 0.0 0.0 setrgbcolor 1.0 setlinewidth 0 setlinecap 0 setlinejoin 10.0 setmiterlimit
100.0 100.0 moveto 300.0 100.0 lineto 0.0 200.0 lineto closepath
stroke
[ 1.0 0.0 0.0 1.0 0.0 0.0 ] 6 array defaultmatrix 6 array concatmatrix setmatrix
0.0 0.0 0.0 setrgbcolor 1.0 setlinewidth 0 setlinecap 0 setlinejoin 10.0 setmiterlimit
200.0 200.0 moveto 200.0 250.0 lineto 250.0 250.0 lineto 250.0 200.0 lineto closepath
stroke
```

Simple coordinate transformations This example involves the operators for scaling and rotating the user coordinate system with respect to the device coordinate system. It also uses gsave and grestore, though not in a way that would affect the drawing.

Given this input:

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox: 0 0 595 842

64 dict begin

%% Box example, PLRM2, p.153

/box
{
    newpath
    0 0 moveto
    0 1 lineto
    1 1 lineto
    1 0 lineto
    closepath
} def

gsave
72 72 scale
box fill
2 2 translate
box fill
grestore

end
%showpage
%%EOF
```

the following output is generated:

```
%!PS-Adobe-2.0
[ 72.0 0.0 0.0 72.0 0.0 0.0 ] 6 array defaultmatrix 6 array concatmatrix setmatrix
0.0 0.0 0.0 setrgbcolor
0.0 0.0 moveto 0.0 1.0 lineto 1.0 1.0 lineto 1.0 0.0 lineto closepath
fill
[ 72.0 0.0 0.0 72.0 144.0 144.0 ] 6 array defaultmatrix 6 array concatmatrix setmatrix
0.0 0.0 0.0 setrgbcolor
0.0 0.0 moveto 0.0 1.0 lineto 1.0 1.0 lineto 1.0 0.0 lineto closepath
fill
```

More simple coordinate transformations This example shows compositions of coordinate transformations and also sets RGB colors. Given this input:

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox: 0 0 595 842

64 dict begin

%% Box example, PLRM2, p.153  -- extended

/box
{
    newpath
    0 0 moveto
    0 1 lineto
    1 1 lineto
    1 0 lineto
    closepath
} def

gsave
72 72 scale
box fill
2 2 translate
box fill
grestore

gsave
1.0 0.0 0.0 setrgbcolor
50 50 translate
50 50 scale
box fill
grestore

gsave
0.0 1.0 0.0 setrgbcolor
200 200 translate
30 60 scale
45 rotate
box fill
grestore

gsave
0.0 0.0 1.0 setrgbcolor
300 300 translate
45 rotate
30 60 scale
box fill
grestore

end
showpage
```

```
%%EOF
```

the following output is generated:

```
%!PS-Adobe-2.0
[ 72.0 0.0 0.0 72.0 0.0 0.0 ] 6 array defaultmatrix 6 array concatmatrix setmatrix
0.0 0.0 0.0 setrgbcolor
0.0 0.0 moveto
0.0 1.0 lineto
1.0 1.0 lineto
1.0 0.0 lineto
closepath
fill
[ 72.0 0.0 0.0 72.0 144.0 144.0 ] 6 array defaultmatrix 6 array concatmatrix setmatrix
0.0 0.0 0.0 setrgbcolor
0.0 0.0 moveto
0.0 1.0 lineto
1.0 1.0 lineto
1.0 0.0 lineto
closepath
fill
[ 50.0 0.0 0.0 50.0 50.0 50.0 ] 6 array defaultmatrix 6 array concatmatrix setmatrix
1.0 0.0 0.0 setrgbcolor
0.0 0.0 moveto
0.0 1.0 lineto
1.0 1.0 lineto
1.0 0.0 lineto
closepath
fill
[ 21.213203435596427 42.426406871192846 -21.213203435596423
42.42640687119285 200.0 200.0 ]
6 array defaultmatrix 6 array concatmatrix setmatrix
0.0 1.0 0.0 setrgbcolor
8.881784197001252E-16 4.440892098500626E-16 moveto
8.881784197001252E-16 1.0000000000000004 lineto
1.0000000000000009 1.0000000000000004 lineto
1.0000000000000009 4.440892098500626E-16 lineto
closepath
fill
[ 21.213203435596427 21.213203435596423 -42.426406871192846
42.42640687119285 300.0 300.0 ]
6 array defaultmatrix 6 array concatmatrix setmatrix
0.0 0.0 1.0 setrgbcolor
0.0 3.8290358538183177E-16 moveto
0.0 0.9999999999999999 lineto
1.0 0.9999999999999999 lineto
1.0 3.8290358538183177E-16 lineto
closepath
fill
showpage
```

Control flow This example shows how control-flow structures such as the for loop must be expanded ("unrolled"). Given this input:

```
/inch {72 mul} def
newpath
2 inch 8 inch moveto
0.7 0.7 0.7 setrgbcolor
300 300 translate
0 30 360 {%for
  pop
  0 -50 moveto 200 0 lineto 0 50 lineto
  30 rotate
} for
stroke
showpage
```

the following output is generated:

```
%!PS-Adobe-2.0
[ 0.8660254037844393 0.499999999999999 -0.499999999999999
0.8660254037844393 300.0 300.0 ] 6 array defaultmatrix 6 array
concatmatrix setmatrix 0.7 0.7 0.7 setrgbcolor 1.0 setlinewidth 0
setlinecap 0 setlinejoin 10.0 setmiterlimit -25.0 -43.30127018922195
moveto 173.2050807568878 -99.9999999999998 lineto 24.999999999999943
43.30127018922195 lineto 5.6843418860808015E-14 -50.00000000000003
moveto 200.0 2.5579538487363607E-13 lineto -5.6843418860808015E-14
50.00000000000003 lineto 25.000000000000057 -43.301270189221924
moveto 173.20508075688758 100.00000000000011 lineto
-25.000000000000057 43.301270189221924 lineto 43.301270189221896
-24.99999999999997 moveto 99.99999999999983 173.20508075688784 lineto
-43.30127018922201 24.999999999999943 lineto 50.0
5.6843418860808015E-14 moveto -2.2737367544323206E-13
199.99999999999997 lineto -50.0 -5.6843418860808015E-14 lineto
43.301270189221896 25.0 moveto -100.00000000000011 173.20508075688767
lineto -43.301270189221896 -25.00000000000003 lineto
24.999999999999943 43.30127018922195 moveto -173.2050807568878
99.99999999999991 lineto -25.0 -43.30127018922195 lineto
-5.6843418860808015E-14 50.00000000000003 moveto -200.0
-1.1368683772161603E-13 lineto 5.6843418860808015E-14
-50.00000000000003 lineto -25.0 43.30127018922195 moveto
-173.2050807568877 -100.00000000000009 lineto 25.0 -43.30127018922195
lineto -43.30127018922195 25.0 moveto -99.99999999999994
-173.20508075688775 lineto 43.301270189221896 -25.00000000000003
lineto -50.0 0.0 moveto 0.0 -200.00000000000003 lineto 50.0 0.0
lineto -43.30127018922195 -25.0 moveto 100.00000000000006
-173.20508075688772 lineto 43.30127018922195 24.99999999999997 lineto
-25.0 -43.301270189221924 moveto 173.2050807568877 -99.99999999999997
lineto 25.0 43.301270189221924 lineto
stroke
showpage
```

Arcs This example shows how arcs are handled. Given this input:

```
newpath
100 100 60 0 90 arc
100 100 30 90 0 arcn
closepath
stroke
showpage
```

the following output is generated:

```
%!PS-Adobe-2.0
[ 1.0 0.0 0.0 1.0 0.0 0.0 ] 6 array defaultmatrix 6 array
concatmatrix setmatrix 0.0 0.0 0.0 setrgbcolor 1.0 setlinewidth 0
setlinecap 0 setlinejoin 10.0 setmiterlimit
160.0 100.0 moveto 159.91777208527444 103.14015737457663 lineto
159.6713137220964 106.2717077960592 lineto 159.26130043570828
109.38606790241386 lineto 158.68885604402834 112.47470144906556 lineto
157.9555495773441 115.52914270615125 lineto 157.06339097770922
118.54101966249684 lineto 156.0148255898321 121.50207697271802 lineto
154.81272745855605 124.40419858454801 lineto 153.4603914513021
127.2394299843728 lineto 151.96152422706632 130.0 lineto
150.32023407672546 132.6783421009016 lineto 148.54101966249684
135.2671151375484 lineto 146.62875768741824 137.75922346299024 lineto
144.58868952864367 140.14783638153148 lineto 142.42640687119285
142.42640687119285 lineto 140.14783638153148 144.58868952864367 lineto
137.75922346299024 146.62875768741824 lineto 135.2671151375484
148.54101966249686 lineto 132.67834210090163 150.32023407672543 lineto
130.0 151.96152422706632 lineto 127.2394299843728 153.46039145130206
lineto 124.40419858454803 154.81272745855605 lineto 121.50207697271802
156.0148255898321 lineto 118.54101966249685 157.06339097770922 lineto
115.52914270615125 157.9555495773441 lineto 112.47470144906555
158.68885604402834 lineto 109.38606790241386 159.26130043570828 lineto
106.2717077960592 159.6713137220964 lineto 103.14015737457663
159.91777208527444 lineto 100.0 160.0 lineto 100.0 130.0 lineto
101.57007868728832 129.9588860426372 lineto 103.13585389802961
129.8356568610482 lineto 104.69303395120693 129.63065021785414 lineto
106.23735072453277 129.34442802201417 lineto 107.76457135307562
128.97777478867204 lineto 109.27050983124842 128.5316954888546 lineto
110.75103848635901 128.00741279491606 lineto 112.20209929227401
127.40636372927803 lineto 113.6197149921864 126.73019572565103 lineto
115.0 125.98076211353316 lineto 116.33917105045082 125.16011703836271
lineto 117.6335575687742 124.27050983124843 lineto 118.87961173149512
123.31437884370912 lineto 120.07391819076574 122.29434476432183 lineto
121.21320343559643 121.21320343559643 lineto 122.29434476432183
120.07391819076574 lineto 123.31437884370912 118.87961173149512 lineto
124.27050983124842 117.6335575687742 lineto 125.16011703836273
116.3391710504508 lineto 125.98076211353316 115.0 lineto
126.73019572565104 113.6197149921864 lineto 127.40636372927803
112.202099292274 lineto 128.00741279491606 110.75103848635901 lineto
128.5316954888546 109.27050983124842 lineto 128.97777478867204
107.76457135307562 lineto 129.34442802201417 106.23735072453277 lineto
```

```
129.63065021785414 104.69303395120693 lineto 129.8356568610482
103.13585389802961 lineto 129.95888604263723 101.57007868728832 lineto
130.0 100.0 lineto closepath
stroke
showpage
```

Additional examples are provided as online files.

Both original and simplified PostScript code is provided for example1, example2, example3, example4, example5, gtiger, gxfig2, and peano.

Original code only is provided for: adobe-magazine-1, adobe-magazine-2, adobe-magazine-3, appolonian, bifur, colorcir-notext, cross, dragon2, dragon_too, escher2, escher-flat, escher, fern, FourColTouchingCircles, FractalArrows, fractal-fern, fractal, garrow_demo, gbox, gbutterfly, gcolumbia, gdoretree, gdot1, ggolfer, gingerbread, gnozzle, goptical_illusion, gparrot, gspiro, hilbert, ifs_coral, ifs_dragon, ifs_fern, ifs_heart2, ifs, ifs_spiral, ifs_tri, kochknife, koch_snowflake1, mandelbrot_color, mandelbrot, Mandelbrot_Set, penrose, ras_cyl, ras_molauto_nice, ras_molauto, ras_rmsd, ras_std, rule110, sierp1, sierp2, sierp3a, sierp3b, sierpent1, sierpinski-1, sierpinski2, sierpinski, sierpinski_trial, snowflakes, snowflak, tree2, TreePhylum, wells-1-6-7-8, and zeta-Lg-family-measured-exponential.

5.2.2 PostScript operators that must be implemented (in addition to those implemented in Phase 3)

We add a third batch, consisting of 46 operators, to the language of Phase 4:

Graphics State Operators - Device Independent

- gsave
- grestore
- grestoreall
- initgraphics
- setlinewidth
- currentlinewidth
- setlinecap
- currentlinecap
- setlinejoin
- currentlinejoin
- setmiterlimit
- currentmiterlimit
- setrgbcolor
- sethsbcolor
- currentrgbcolor

Coordinate System and Matrix Operators

- matrix
- initmatrix
- identmatrix
- defaultmatrix
- currentmatrix
- setmatrix
- translate
- scale
- rotate
- concat
- concatmatrix
- transform
- itransform

Path Construction Operators

- newpath
- currentpoint
- moveto
- rmoveto
- lineto
- rlineto
- arc
- arcn
- arct
- arcto
- curveto
- rcurveto
- closepath
- clippath

Painting Operators

- stroke
- fill
- eofill

Device Setup and Output Operators

- showpage

5.2.3 Clarifications of PLRM2

- none

5.2.4 Deviations from PLRM2

- Arcs: A circular arc is to be divided into smaller pieces and then approximated by one of the PostScript primitive segments. You may divide an arc into line segments as opposed to Bézier curve segments (which PLRM2 indicates). Always divide an arc into segments corresponding to subarcs of 3 degrees.

5.2.5 Features in PLRM2 that need not be implemented (for graphics)

- dash patterns for line segments
- operators that query the current path, such as insideness testing (PLRM2 section 4.5.3)
- path clipping
- "user paths" (PLRM2 section 4.6)
- forms (PLRM2 section 4.7)
- color spaces other than RGB (PLRM2 section 4.8)
- patterns (PLRM2 section 4.9)
- images (PLRM2 section 4.10)
- device setup (PLRM2 section 4.11)
- fonts, font management, and font character rendering (PLRM2 chapter 5)
- rendering (PLRM2 chapter 6)
- Display PostScript (PLRM2 chapter 7)
- any operators not listed in 5.2.2 above

How to turn in

Turn in your code by running

~barrick/handin your-file

on a regular UNM CS machine.

You should use whatever filename is appropriate in place of your-file. You can put multiple files on the command line, or even directories. Directories will have their entire contents handed in, so please be sure to clean out any cruft.