



In short

- Escrito is a PostScript interpreter written in lua.
- The goal is to bring native EPS and pstricks support to LuaT_EX.
- The project has started in April 2010.

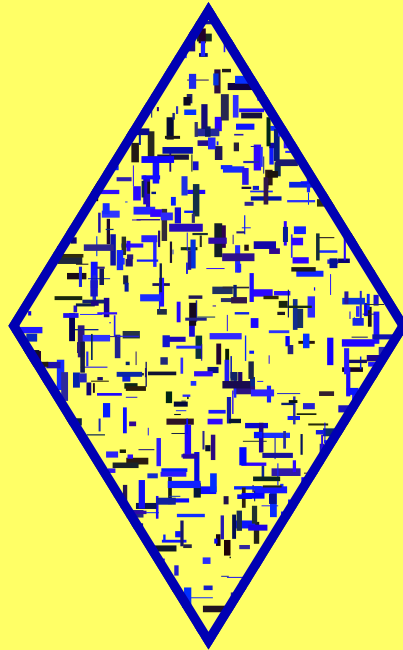
Example: tiger.eps from GhostScript



Example source

```
\startluacode
escrito = require('escrito')
function do_showpage (v,page)
  local minx, miny, maxx, maxy = escrito.boundingBox(escrito, page)
  tex.sprint (string.format(
    "\\leavevmode" ..
    "\\vbox to %fbp{\\vss\\hbox to %fbp{\\kern %fbp\\pdfliteral{%s}\\hss}\\kern -%fbp}",
    (maxy-miny), -- total height
    (maxx-minx), -- total width
    -minx, -- horizontal offset
    v,
    miny )) -- vertical offset
end
escrito.new({showpage = do_showpage})
escrito.interpret('gsave .5 .5 scale (tiger.eps) run grestore')
\stopluacode
```

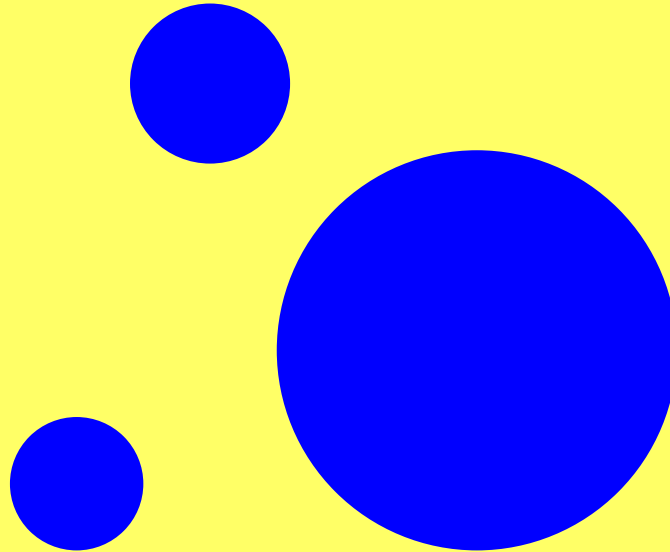
Example: à la Mondriaan



Example source

```
\startluacode  
escrito.interpret([[  
gsave .5 .5 scale  
(PSlib.eps) run  
05012010 0 0 0.7 2 Mondrian  
showpage]])  
\stopluacode
```

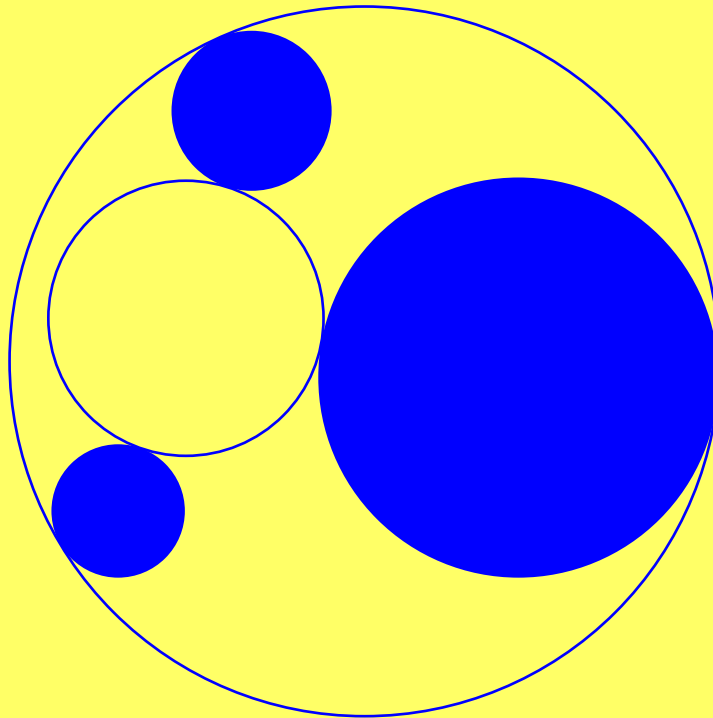
Example: Three circles



Example source

```
\startluacode
escrito.interpret([[
gsave .5 .5 scale
(PSlib.eps) run
/x1 300 def /y1 100 def /r1 150 def
/x2 100 def /y2 300 def /r2 60 def
blue
newpath 0 0 50 0 360 arc fill
newpath x1 y1 r1 0 360 arc fill
newpath x2 y2 r2 0 360 arc fill
grestore cypage
]])
\stopluacode
```


Example: Apollonius 2



Example source

```
\startluacode
escrito.interpret([[
gsave .5 .5 scale
x1 y1 r1 neg x2 y2 r2 neg 0 0 50 neg Apollonius2
/rsnd1 exch def /ysnd1 exch def /xsd1 exch def
/rsnd2 exch def /ysnd2 exch def /xsd2 exch def
blue
newpath xsnd1 ysnd1 rsnd1 0 360 arc stroke
newpath xsnd2 ysnd2 rsnd2 0 360 arc stroke
grestore showpage
]])
\stopluacode
```

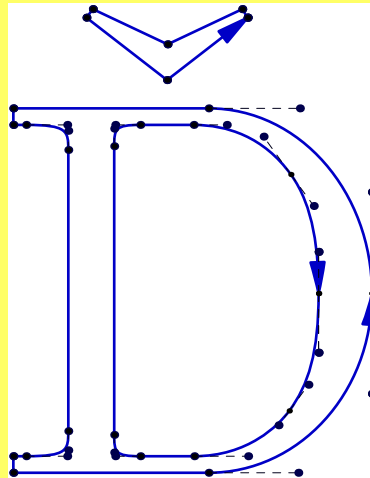
Example: gradient



Example source

```
\startluacode
escrito.interpret([[
  .25 .5 0 setrgbcolor
  0 1 100 {
    /i exch def
    currentrgbcolor pop i 100 div sqrt setrgbcolor
    newpath 0 0 100 i sub 0 360 arc fill } for
  showpage
]])
\stopluacode
```

Example: a MetaPost picture, adjusted



Example source

```
\startluacode
escrito.interpret([[
1 1 1 setrgbcolor
5 -2 moveto 144 -2 lineto 144 176 lineto 5 176 lineto closepath fill
0 dict begin
/setrgb { setrgbcolor } bind def
/togray { /b exch def /g exch def /r exch def
          r 0.30 mul g 0.59 mul b 0.11 mul add add } def
/setgray { 0 0 3 -1 roll setrgb } def
/setrgbcolor { togray 0 0 3 -1 roll setrgb } def
(mpman-56.eps) run
end
]])
\stopluacode
```

Example: A broken example



Current status

- Partial PostScript level 1 implementation (no level 2 features yet)
- Alpha 'quality'.
- Type 1 font loading not done yet (current version reads t1disasm output).
- Unoptimized lua code.

Missing level 1 operators

reversepath

strokepath

charpath

pathbbox

pathforall

image

imagemask

ashow

widthshow

awidthshow

stringwidth

cachestatus

setcachedevice

setcharwidth

setcachelimit

Partial level 1 operators

setflat
settransfer
clip
eoclip
findfont
show
ashow

Current problems

...of which there are many:

- The bounding box calculation does not account for text, miterlimits, or linewidth.
- The flatness setting is not written to the PDF output.
- The transfer function is not actually applied.
- Simple clipping works sort of OK in the PDF output, but complex clippaths produces rubbish.
- PDF fonts really need an extension to LuaT_EX, or lots of extra code.
- show & ashow do not update the current point.

Finally

Escrito has a website, mailing lists, but tracker, etc.

See <http://escrito.luatex.org/> for details.

Escrito will be released under the simplified BSD license.