LibreLogo

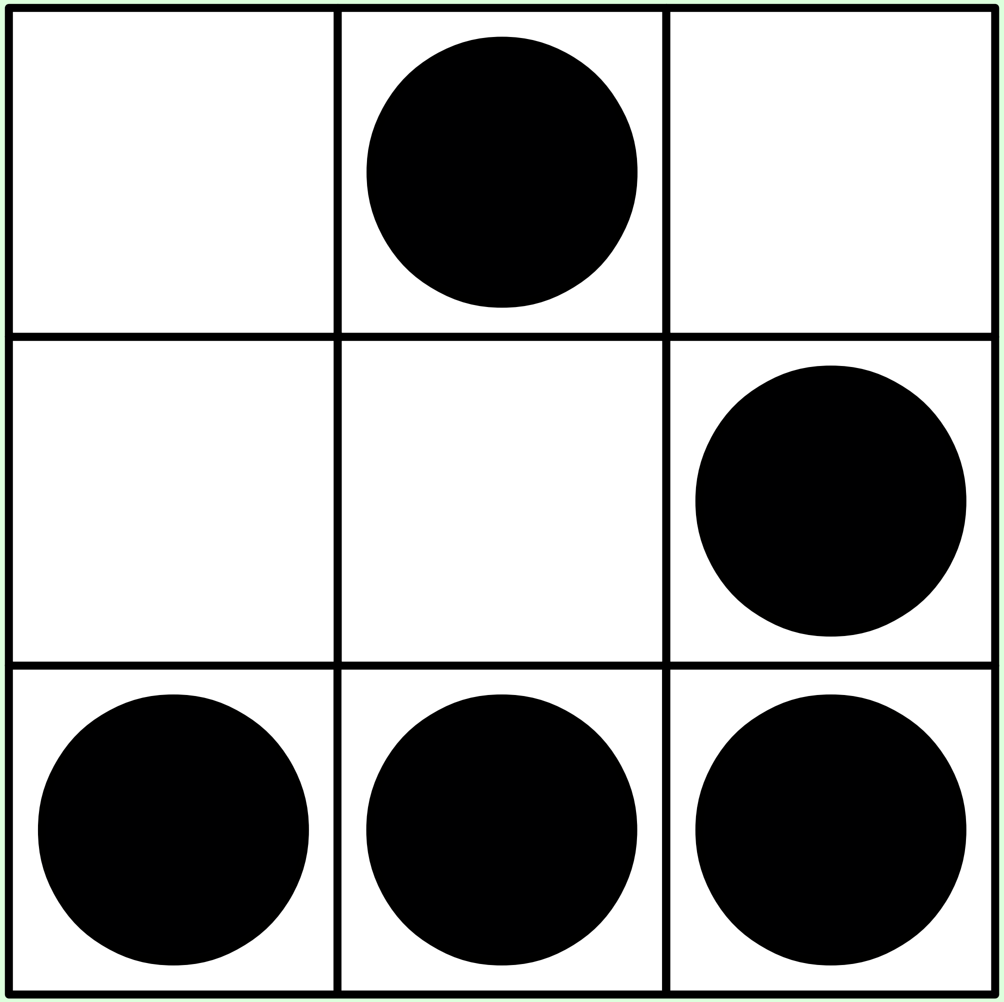
László Németh, FSF.hu Foundation

Linux in Education Conference, Hungary

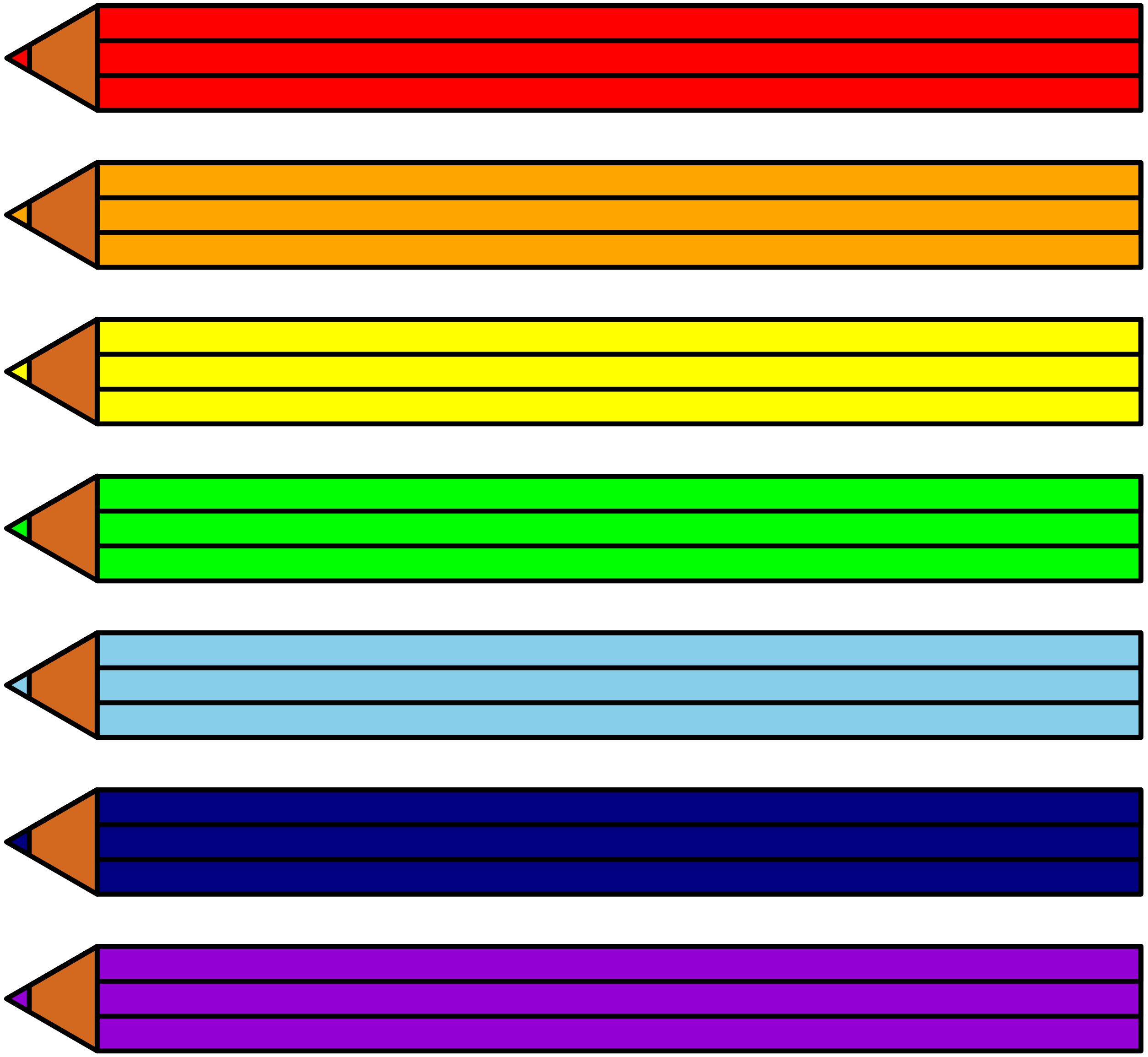
28 April 2012, Budapest

http://numbertext.org/logo

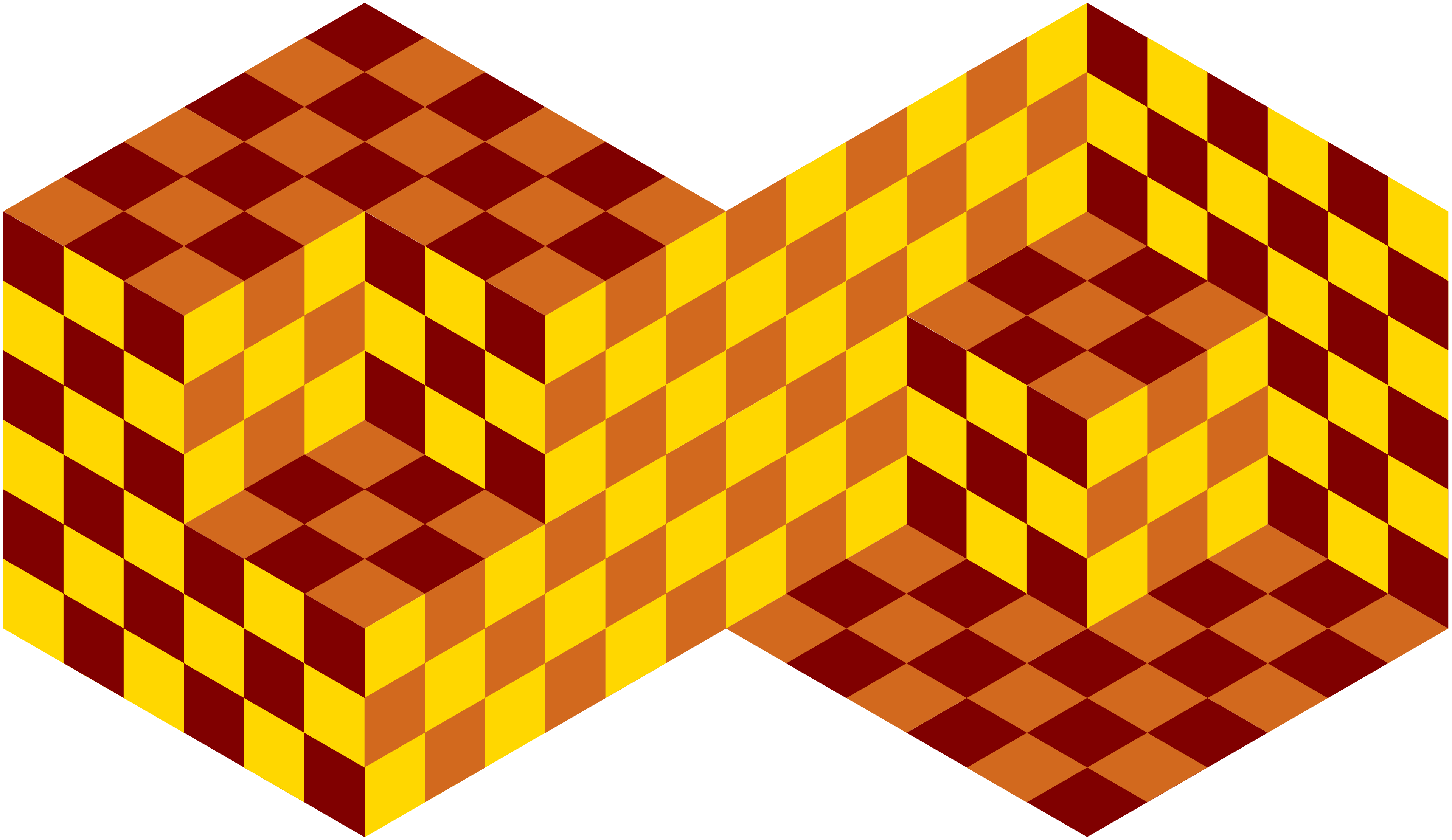
Free software for...



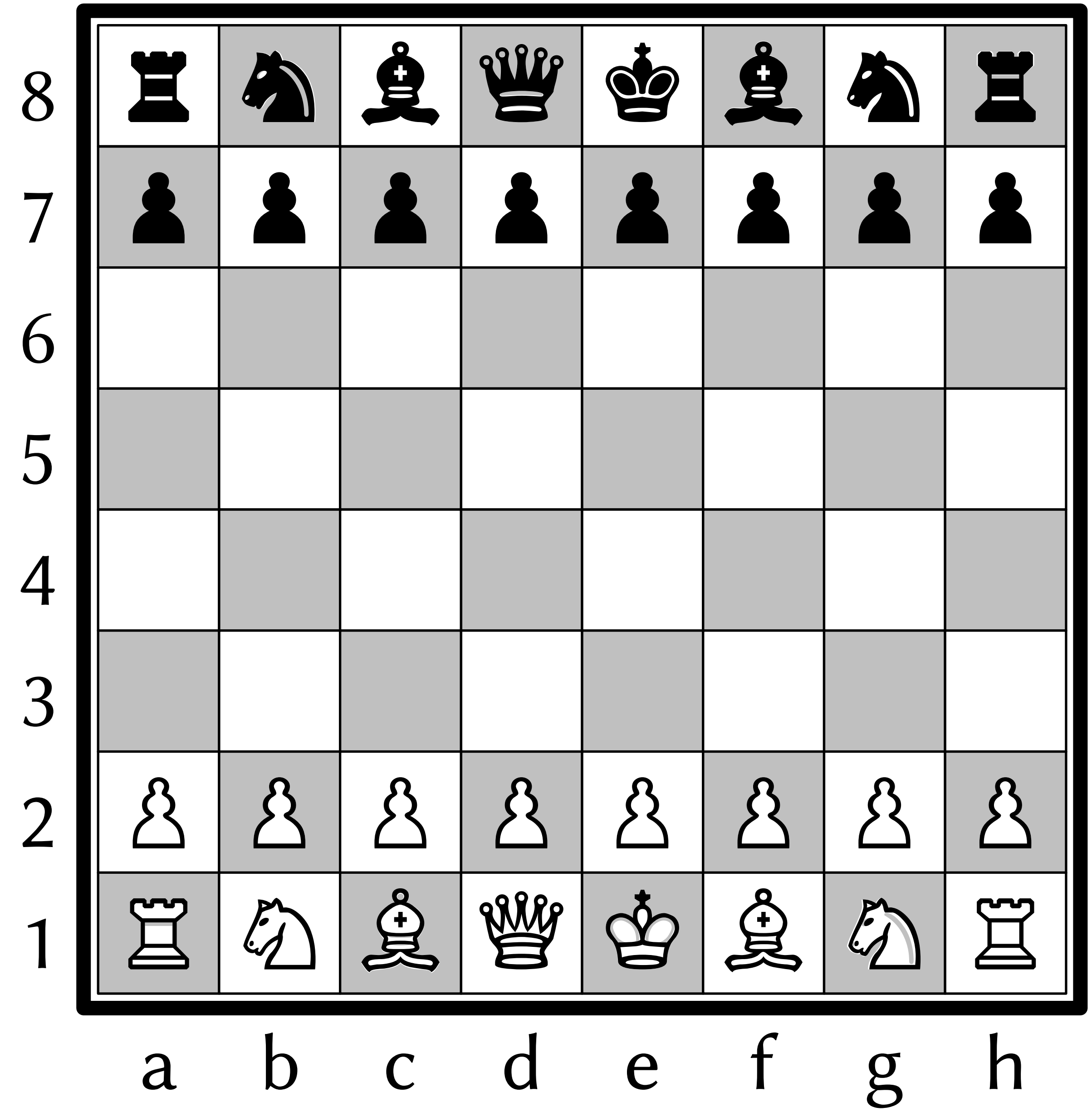
Learning



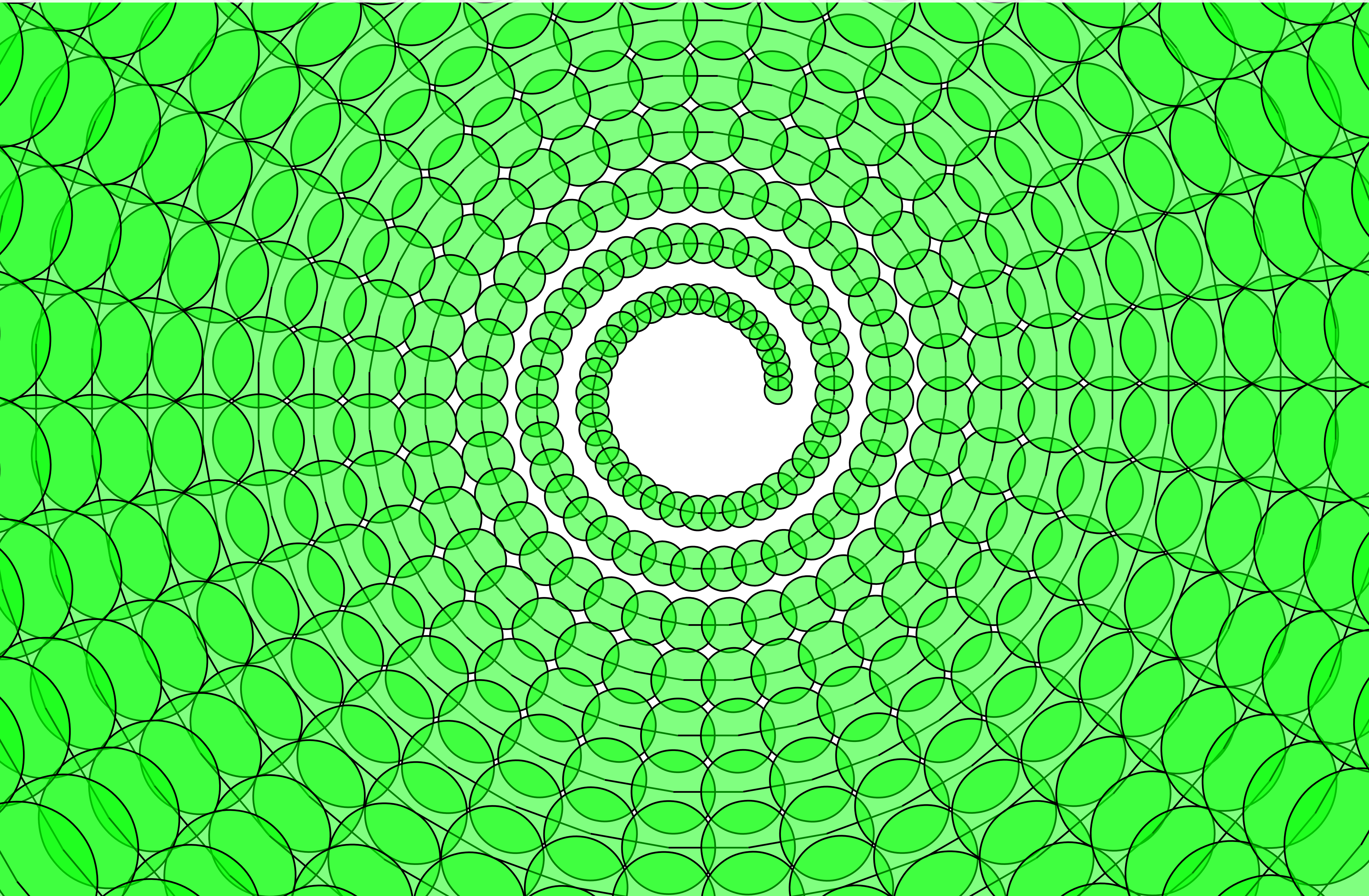
Art

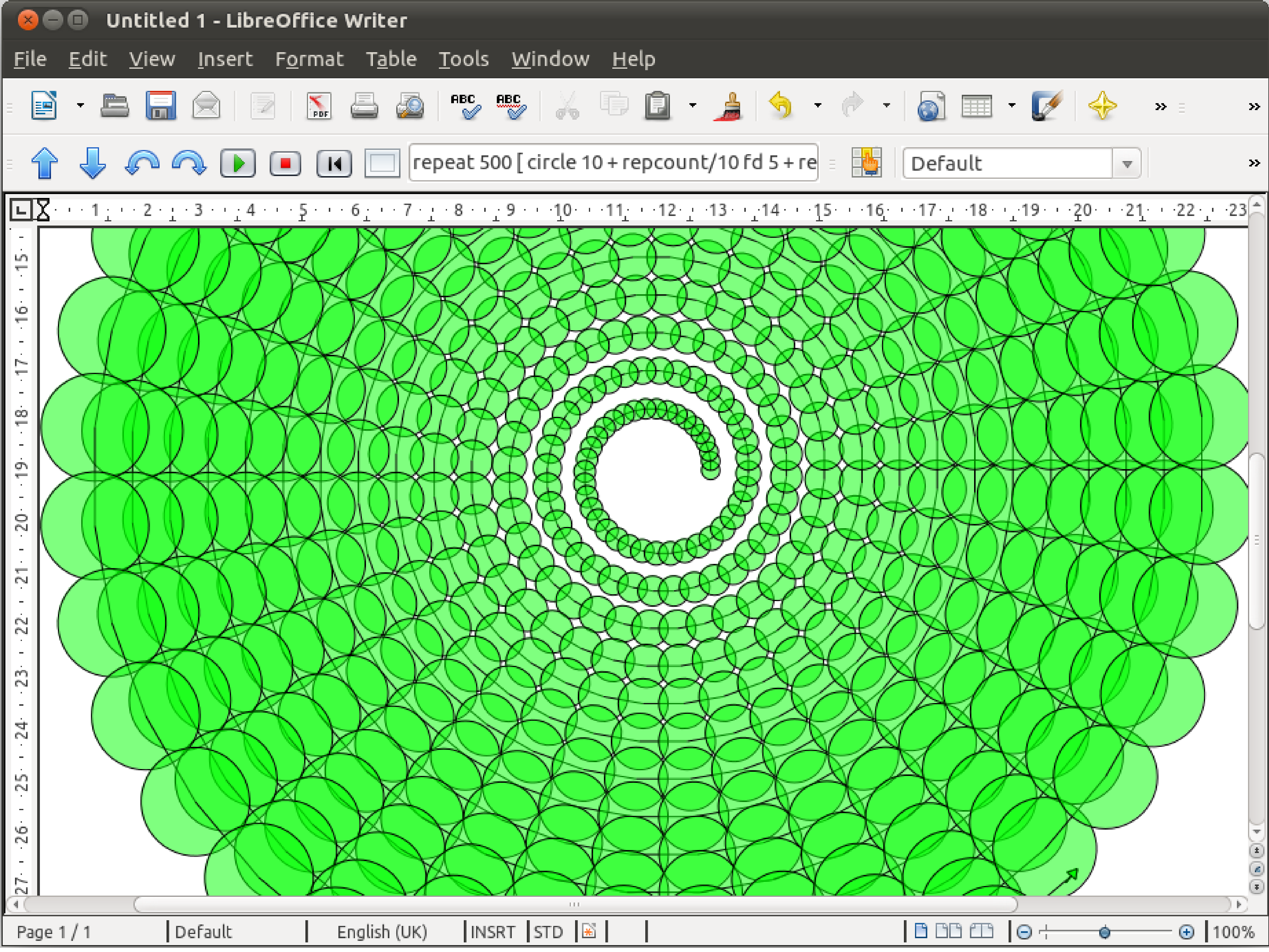


Desktop publishing



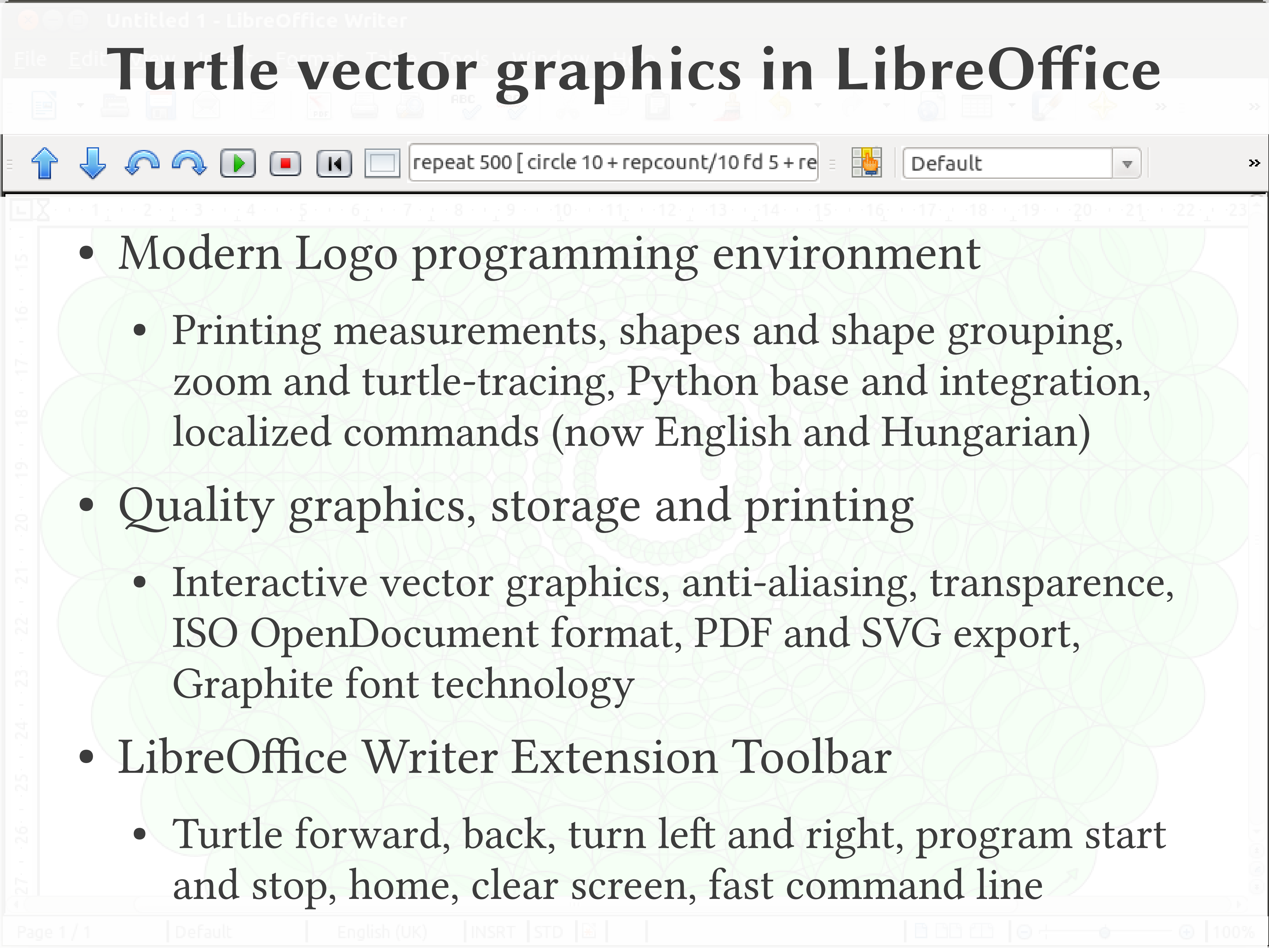
```
repeat 500 [ circle 10 + repcount/10 fd 5 + repcount/10 lt 10 ]
```




Untitled 1 - LibreOffice Writer
File Edit View Insert Format Table Tools Window Help
Default
Page 1 / 1
Default
English (UK)
INSRT
STD
100%

```
repeat 500 [ circle 10 + repcount/10 fd 5 + re
```

Turtle vector graphics in LibreOffice


Untitled 1 - LibreOffice Writer
File Edit View Insert Format Table Tools Window Help
Default
Page 1 / 1
Default
English (UK)
INSRT
STD
100%

```
repeat 500 [ circle 10 + repcount/10 fd 5 + re
```

- Modern Logo programming environment
  - Printing measurements, shapes and shape grouping, zoom and turtle-tracing, Python base and integration, localized commands (now English and Hungarian)
- Quality graphics, storage and printing
  - Interactive vector graphics, anti-aliasing, transparence, ISO OpenDocument format, PDF and SVG export, Graphite font technology
- LibreOffice Writer Extension Toolbar
  - Turtle forward, back, turn left and right, program start and stop, home, clear screen, fast command line

Education and LibreLogo

- Turtle graphics, algorithms in elementary schools
- Using office suites (LibreOffice): handling pictures, page settings, PDF-export etc.
- Practical programming knowledge: Python list, tuple, dict, set data structures etc.
- New motivations: art, desktop publishing, open source code of LibreLogo (thousand lines in Python/PyUNO)

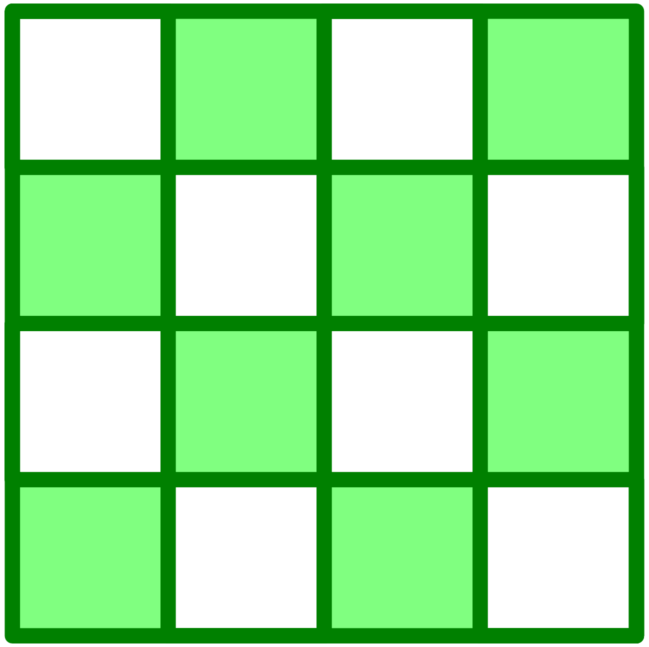
Logo and LibreLogo

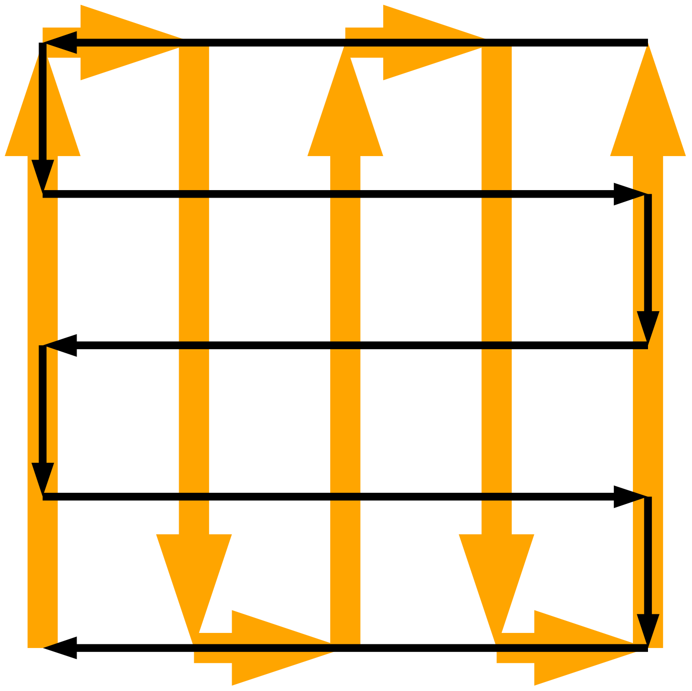
Logo	Differences	LibreLogo
turnright 90 = rt 90	**optional clock positions ▶** (suitable for the lower grades)	turnright 90° = rt 90 = turnright 3h
forward 1 = fd 90	**DTP point, inch, cm, mm ▶** **◀ pixel**	forward 1pt = fd 1 = fd 1in/72 = fd 2.54cm/72
fill (flood-fill, need position)	**vector graphics ▶** **◀ raster graphics**	fill (close and fill actual shape)
"word [string]	text notation **writing standard ▶** **◀ formal (LISP)**	“string” (orthography, Writer), 'string' (Python), "word, "word"
lists [] (eg. 1-line instruction list)	Python in Logo turtle shell ▶ ◀ functional programming language	blocks [ ] (need space or line break) and lists [], eg. repeat 5 [ellipse [5, 10]]

Checkerboard

- Filled complex shape (with a single line, see illustration)

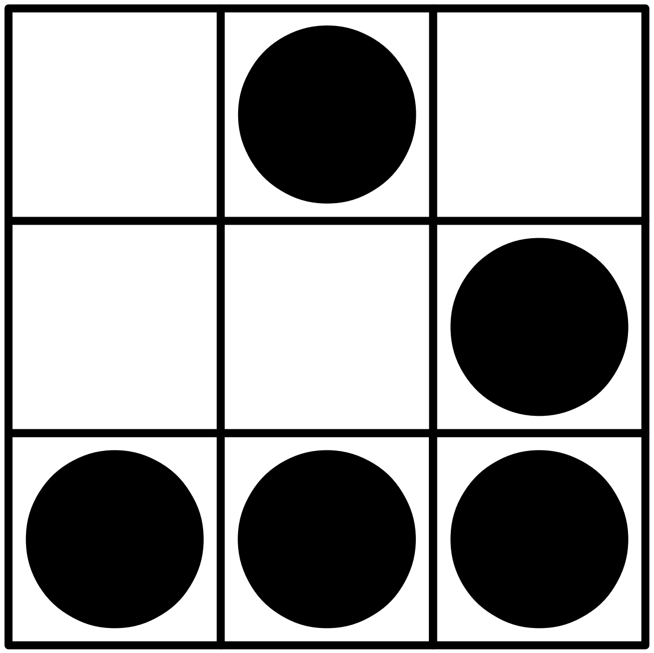
```
to checkerboard size x y ; 2x × 2y sq.
repeat x [
   fd size*y*2 rt 90 fd size rt 90
   fd size*y*2 lt 90 fd size lt 90
] fd size*y*2 lt 90
repeat y [
   fd size*x*2 lt 90 fd size lt 90
   fd size*x*2 rt 90 fd size rt 90
] fd size*x*2 fill
end

checkerboard 1cm 2 2
```





Hacker logo

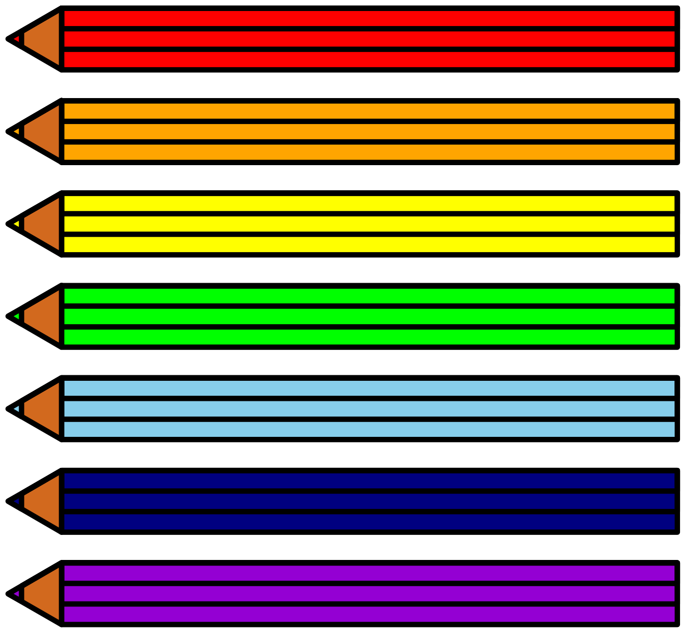
```
to line pattern
   for i in pattern [
      pu fd 10 pd fillcolor “white”
      rectangle [10, 10]
      fillcolor “black”
      if i = “x” [ circle 8 ]
   ]
   pu rt 90 fd 10 lt 90
   back 10 * count pattern
end

rt 90
line “ x ”
line “  x”
line “xxx”
```



Pencils

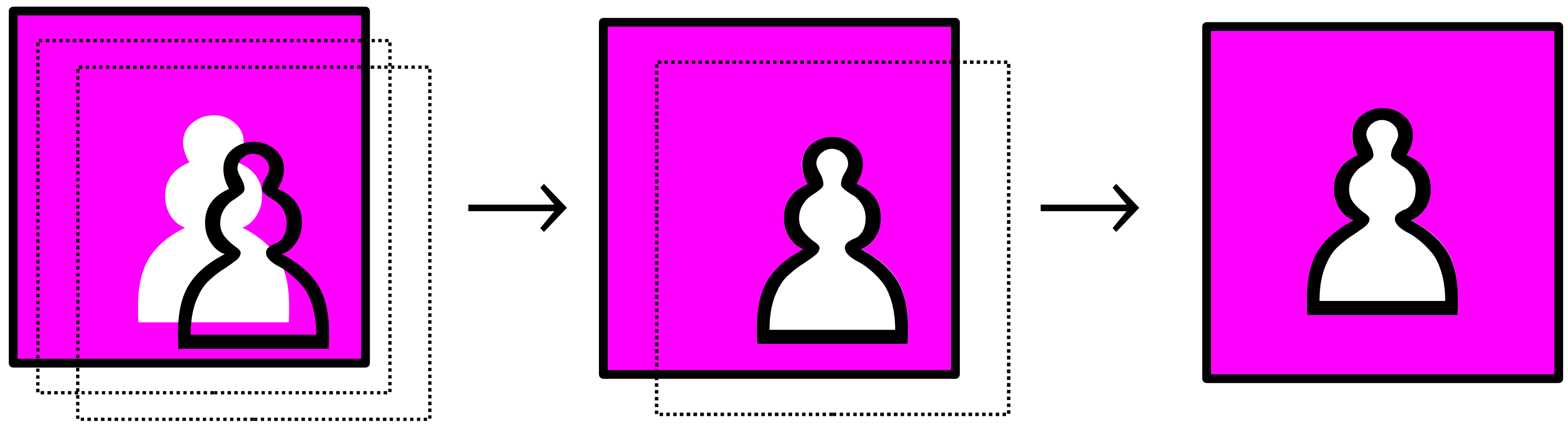
- Pencils are different pictures (grouped shapes)

```
to triangle size color
    repeat 3 [ fd size lt 120 ] fc color fill
end

to box size f
    repeat 2 [ fd size*10 rt 90 fd size*f rt 90 ]
end

to pencil size color
    box size 1 fc color fill
    box size 2/3 box size 1/3
    close rt 150 triangle size “chocolate”
    fd size*0.75 triangle size/4 color
    back size*0.75 lt 150
end

pensize 2 rt 90
for color in [“red”, “orange”, “yellow”, “lime”, ~
    “skyblue”, “navy”, “violet”] [
        pic [ pencil 30 color ]
        pu rt 90 fd 45 lt 90 pd
]
```



Pieces

- Unicode characters (♔♚♕♛♖♜♗♝♘♞♙♟)
- In “invisible” squares for manual positioning
- White pieces: transparent white Unicode pieces combined with dark pieces in white
- Grouping pieces with their background



http://www.numbertext.org/logo