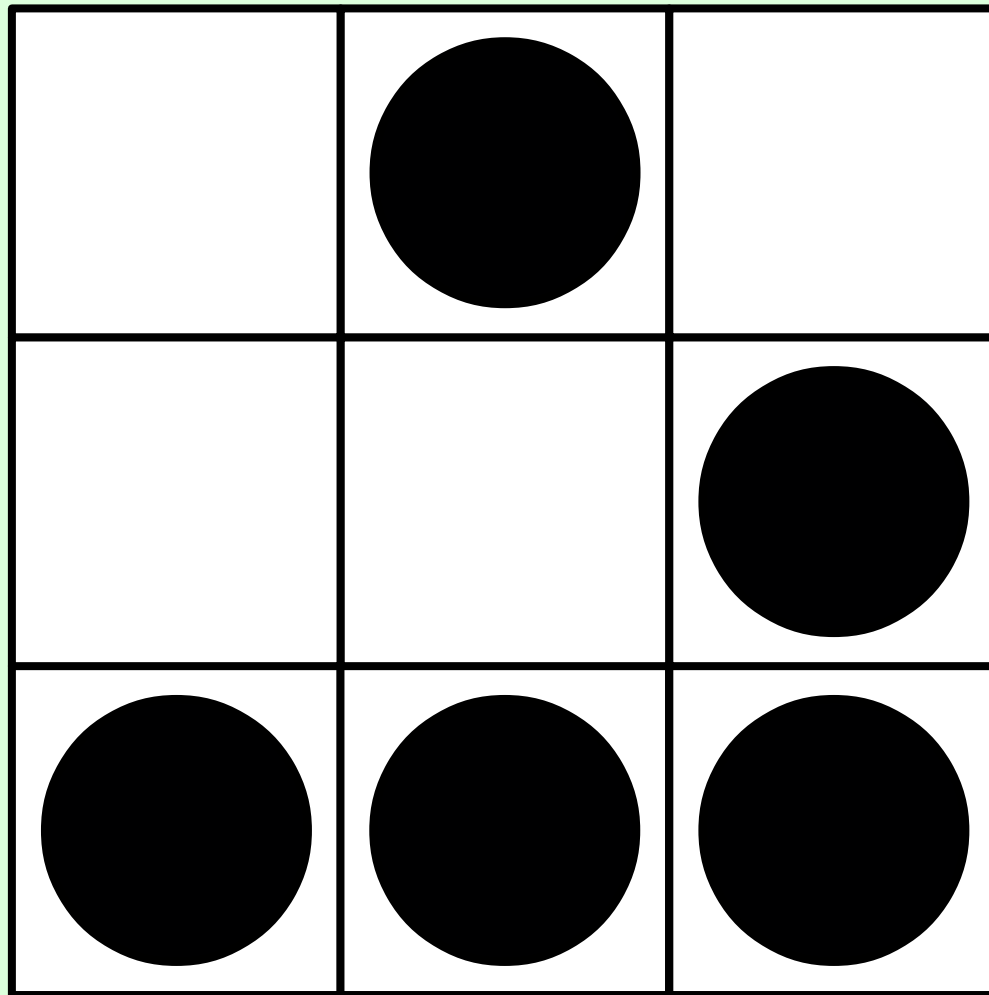


# LibreLogo

Németh László, FSF.hu Alapítvány  
Linux az oktatásban konferencia  
2012. április 28., Budapest  
<http://www.lok.hu>

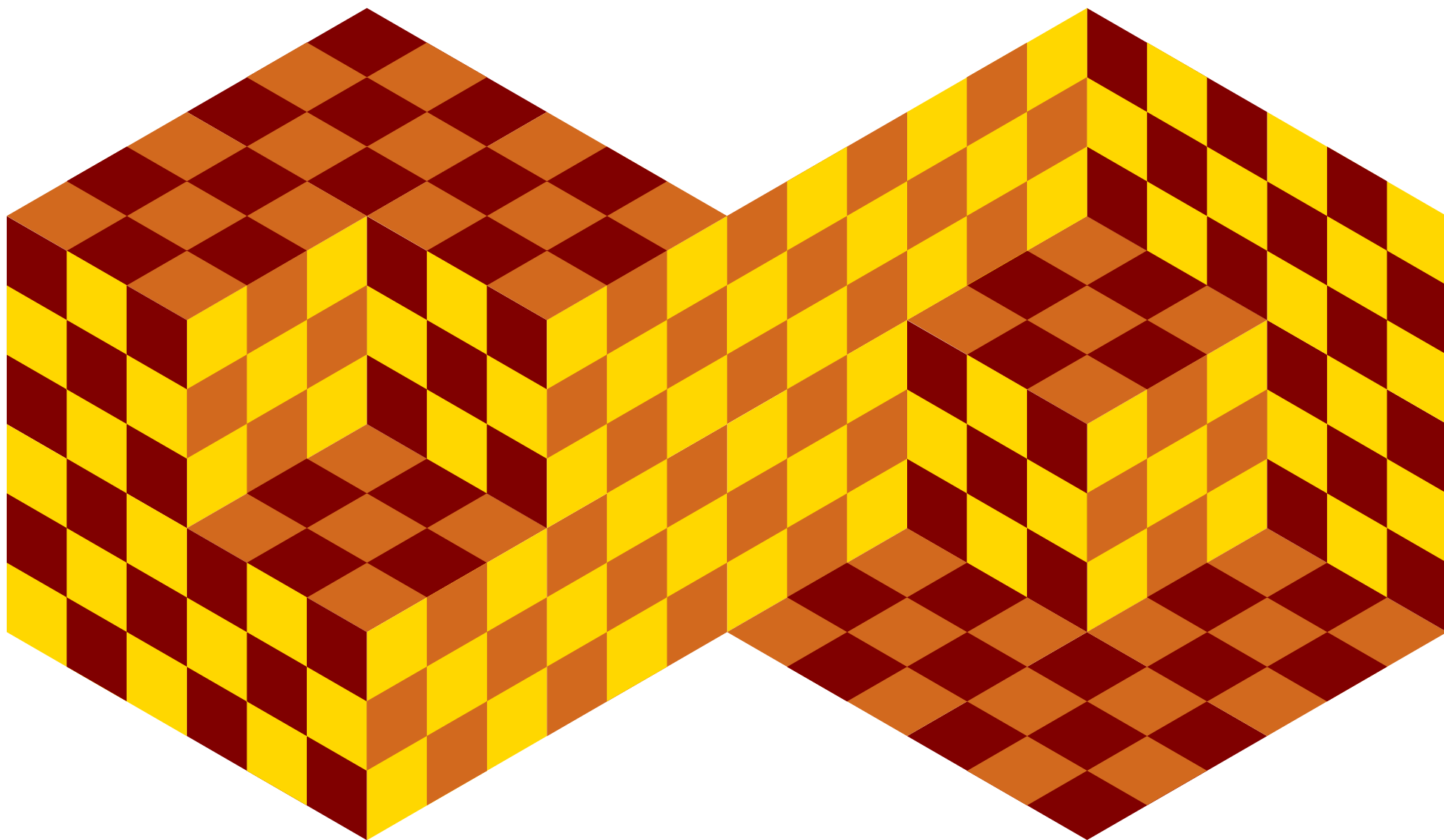
# Szabad szoftver



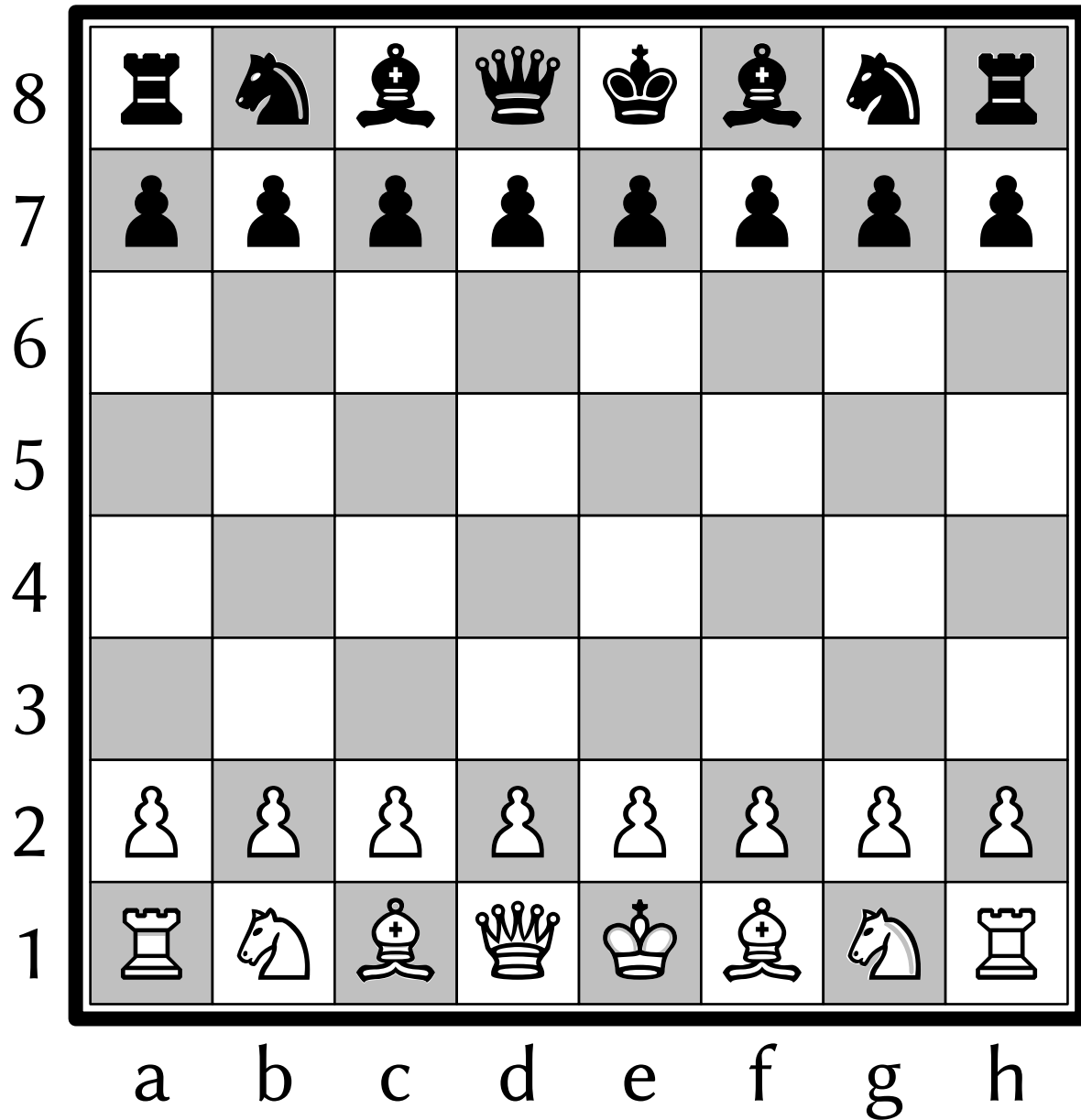
# Tanuláshoz



# Alkotáshoz

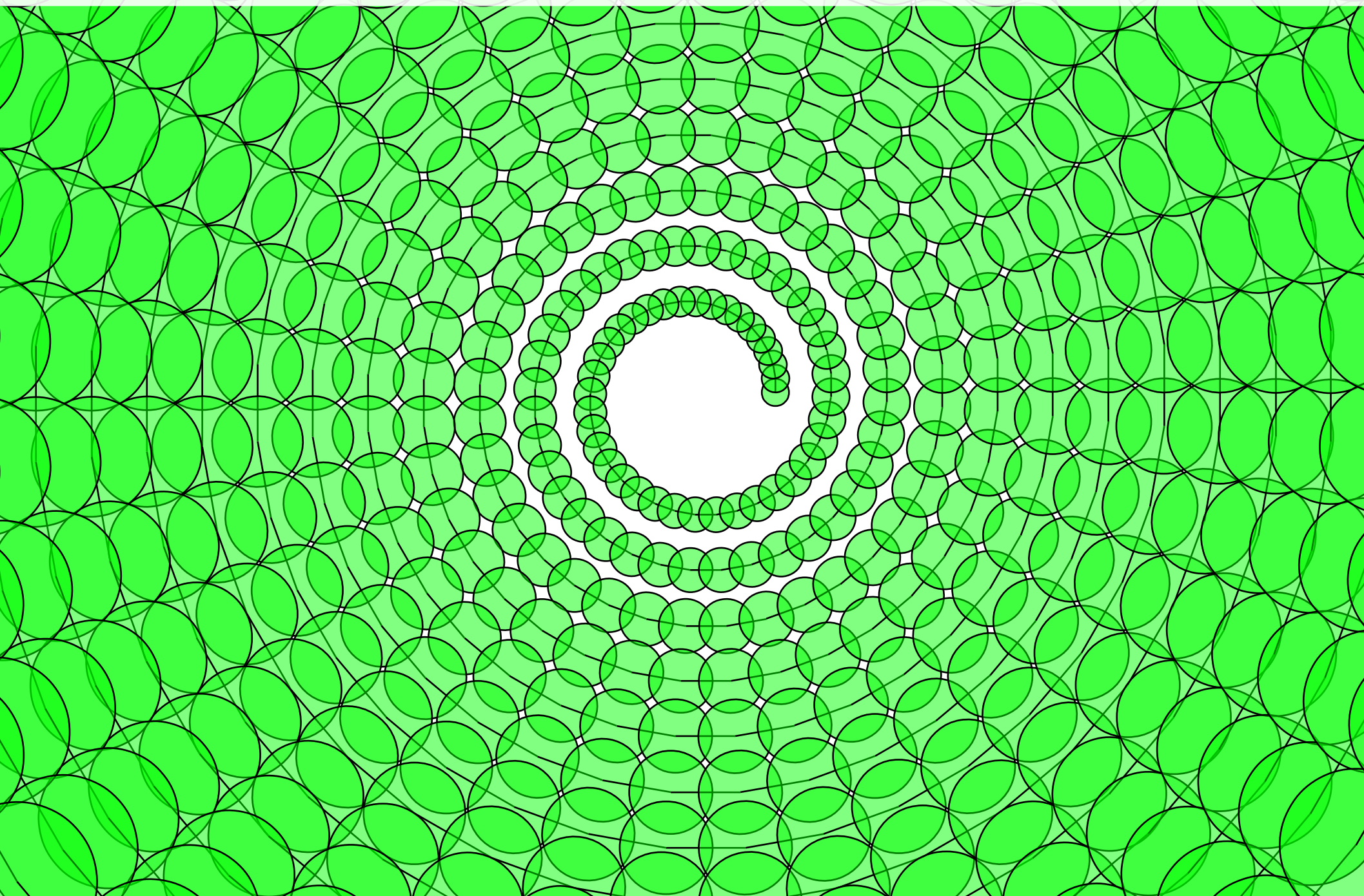


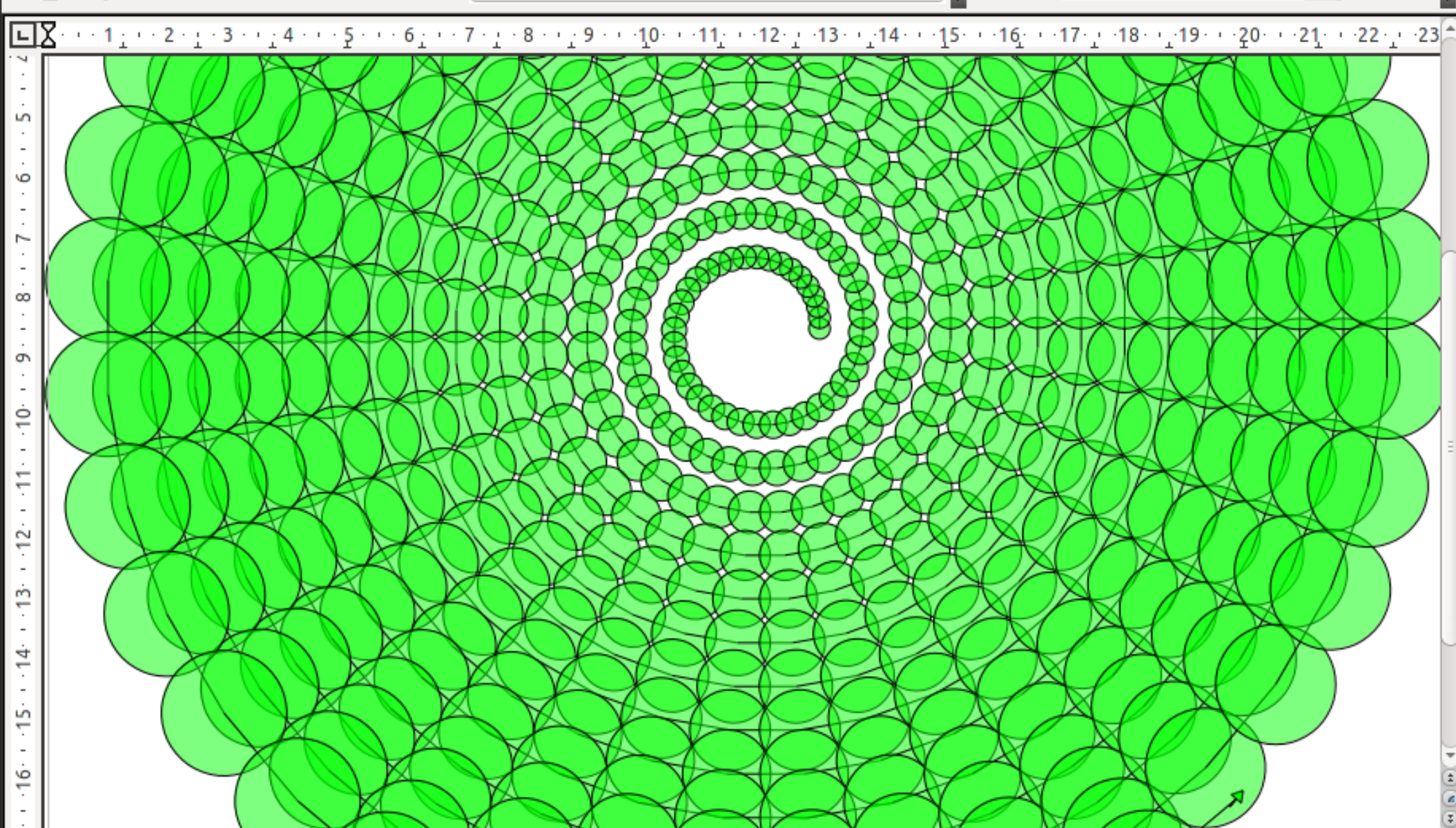
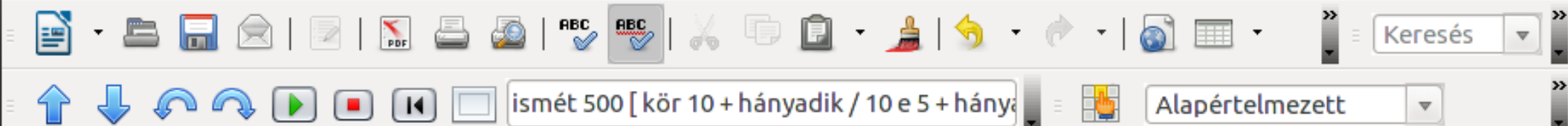
# Kiadványszerkesztéshez





tv 1 ismét 500 [ kör 10 + hányadik/10 e 5 + hányadik/10 b 10 ]







# Teknőc-vektorgrafika a LibreOffice-ban

- modern Logo programozási környezet
  - nyomdai mértékegységek, csoportosítható alakzatok, zoom és teknőc-nyomkövetés, Python alap és integráció, honosítható parancsok (jelenleg magyar és angol)
- minőségi megjelenítés, adattárolás és nyomtatás
  - interaktív vektorgrafika, élsimítás, áttetszőség, szín- és alfa átmenet, OpenDocument formátum, PDF és SVG export, Graphite betűtechnológia
- LibreOffice Writer kiegészítő eszköztár
  - teknőc előre, hátra, balra, jobbra, programindítás és -leállítás, kezdőpozíció, képernyőtörlés, gyors parancssor



# Oktatás és LibreLogo

- Nemzeti Alaptanterv: algoritmusok (teknőcgrafika, ciklusok, feltételvizsgálat, eljárások, rekurzió, tömbök)
- Comenius és Imagine Logo alap utasításkészlet támogatása (köz- és felsőoktatás, versenyek)
- irodai programcsomag (LibreOffice) oktatása (képkezelés, oldalbeállítás, PDF-export stb.)
- egyéb gyakorlati programozói ismeretek: Python (lista, sor, szótár, halmaz adatszerkezetek, stb.)
- új motivációk: művészet, kiadványszerkesztés, a LibreLogo nyílt forráskódjának (ezer sor Python/PyUNO-ban) megismerése

# Comenius/Imagine Logo és LibreLogo

Logo	összehasonlítás	LibreLogo
jobbra 90 (röv. j 90)	órapozíció is (alsósoknak) ►	jobbra 90° = j 90 = jobbra 3ó = j 3h
előre 1 (röv. e 1)	nyomdai pont, mm, cm, col ► ◄ képpont	előre 1pt = előre 1 = e 2,54cm/72 = e 1"/72
tölt (egybefüggő terület kitöltése, pozicionálni kell)	vektorgrafika ► ◄ rasztergrafika	tölt: aktuális alakzat lezárása és kitöltése zár: csak lezárása
tanuld/eljárás/elj	„to” pontosabb fordítása ►	„ez” (a régiek mellett)
"szó [karakterlánc]	karakterlánc megadása nyelvi standard ► ◄ formális (LISP prog. nyelv)	„szöveg” (helyesírás, Writer), "szó, "szó", 'szöveg' (Python)
[] lista: pl. utasításlista	Python Logo teknőspáncélban ► ◄ funkcionális prog. nyelv	[ ] blokk (szóköz vagy új sor kell) és lista []

# Pepita rácsháló

- kitöltött komplex alakzat (egy vonallal megrajzolva, l. alsó kép)

ez pepita méret  $x$   $y$  ;  $2x \times 2y$  négyzet  
ismét  $x$  [

    e méret  $y*2$  j 90 e méret j 90

    e méret  $y*2$  b 90 e méret b 90

] e méret  $y*2$  b 90

ismét  $y$  [

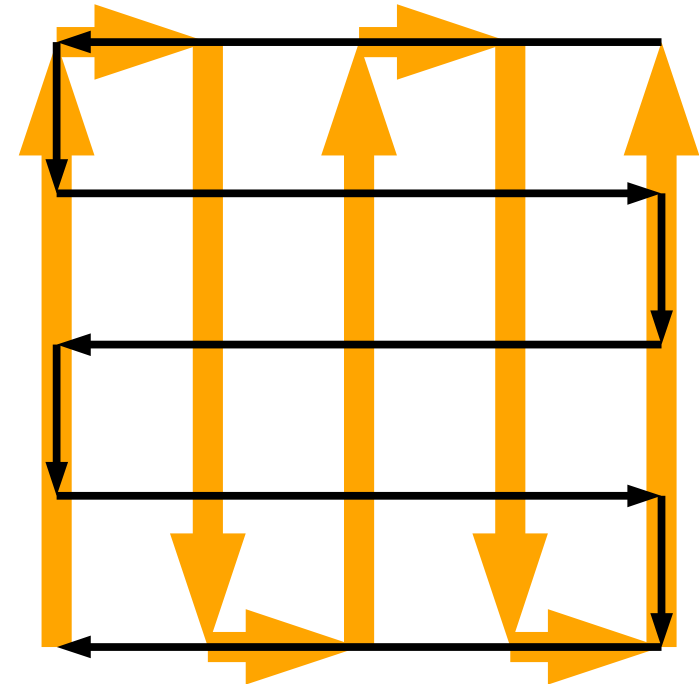
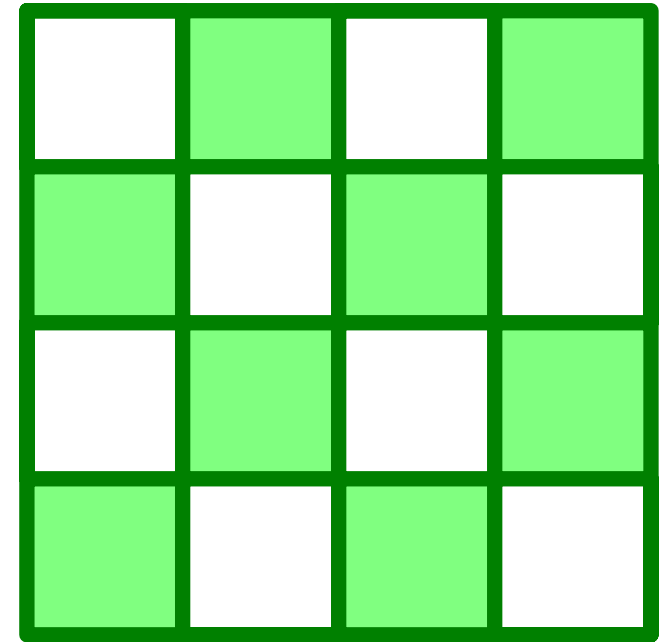
    e méret  $x*2$  b 90 e méret b 90

    e méret  $x*2$  j 90 e méret j 90

] e méret  $x*2$  tölt

vége

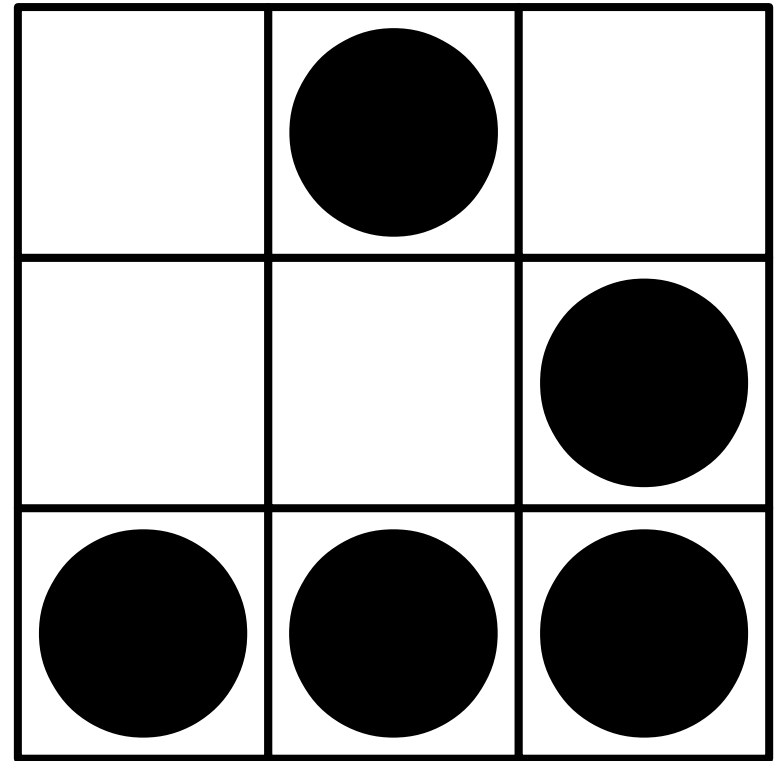
pepita 1cm 2 2



# Szabad szoftveres közösségi logó

```
ez rács minta
  fut i minta-ban [
    tf e 10 tl tlsx „fehér”
    négyzet 10
    tlsx „fekete”
    ha i = „x” [ kör 8 ]
  ]
  tf j 90 e 10 b 90
  h 10 * db minta
vége

j 90
rács „ x ”
rács „  x”
rács „xxx”
```





# Ceruzák

- minden ceruza külön kép (alakzatcsoport)

ez háromszög méret szín

ismét 3 [ e méret b 120 ] tlsx szín tölt  
vége



ez négyszög méret f

ismét 2 [ e méret\*10 j 90 e méret\*f j 90 ]  
vége



ez ceruza méret szín

négyszög méret 1 tlsx szín tölt  
négyszög méret 2/3 négyszög méret 1/3  
zár j 150 háromszög méret „világosbarna”  
e méret\*0,75 háromszög méret/4 szín  
h méret\*0,75 b 150  
vége



tv 2 j 90

fut szín [„piros”, „narancs”, „sárga”, „világoszöld”, ~  
„világoskék”, „sötétkék”, „ibolya”]-ban [  
kép [ ceruza 30 szín ]  
tf j 90 e 45 b 90 tl

]

# Vasarely nyomán

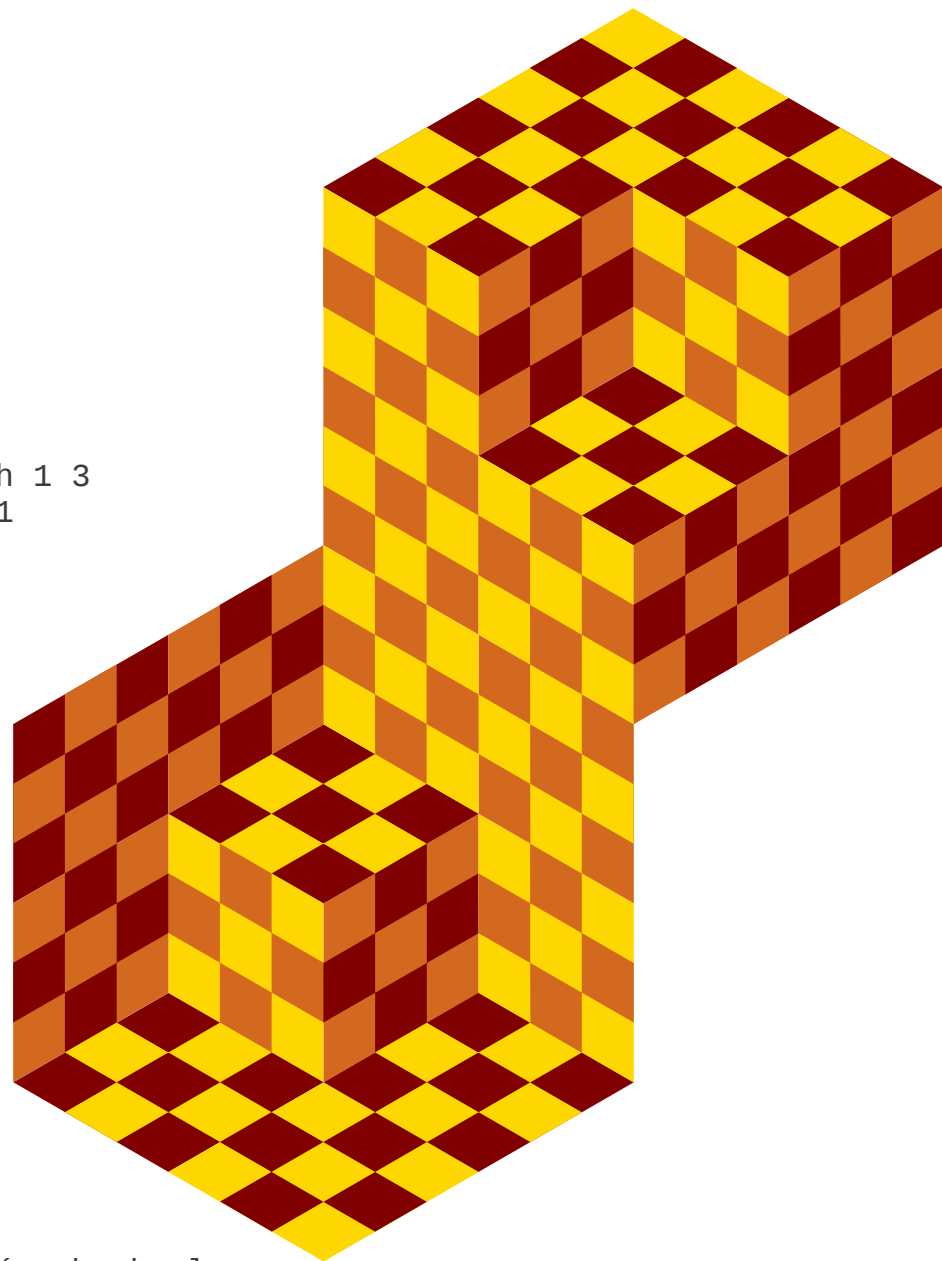
```
ez csempe :méret :n :m
    ismét 2 [ j 60 e :méret*:n j 120 e :méret*:m ]
vége
```

```
ez oldal :h :háttér :előtér
    töltőszín :háttér csempe :h*3 1 1 tölt
    töltőszín :előtér tf j 60 e :h b 60 tl csempe :h 1 3
    tf h :h*2 b 120 e :h j 120 e :h tl csempe :h 3 1
    tölt tf e :h tl
vége
```

```
ez félkép :h :k :l :m
    oldal :h :k :l b 120
    oldal :h :l :m b 120
    oldal :h :m :k tf b 120 h :h*3 j 60 tl
    oldal :h :m :l tf e :h*3 tl
    oldal :h :l :m b 120
    oldal :h :m :k j 60 tf e :h*3 tl b 60
    oldal :h :k :m tf e :h*3 tl
    oldal :h :m :k b 120
    oldal :h :k :l tf j 60 e :h*3 b 60 tl
    oldal :h :l :k tf e :h*3 tl
    oldal :h :k :l b 120
    oldal :h :l :m
vége
```

```
ez teljes :h :k :l :m
    tf j 90 e :h*5+5 b 90 tl félkép :h :k :l :m
    tf h :h*6 b 60 e :h*3 b 60 e :h*3 b 120 tl félkép :h :k :l :m
vége
```

```
törölkép haza elrejt tollszín „láthatatlan”
kép [ teljes 50 „aransárga” „világosbarna” „sötétbarna” ]
```

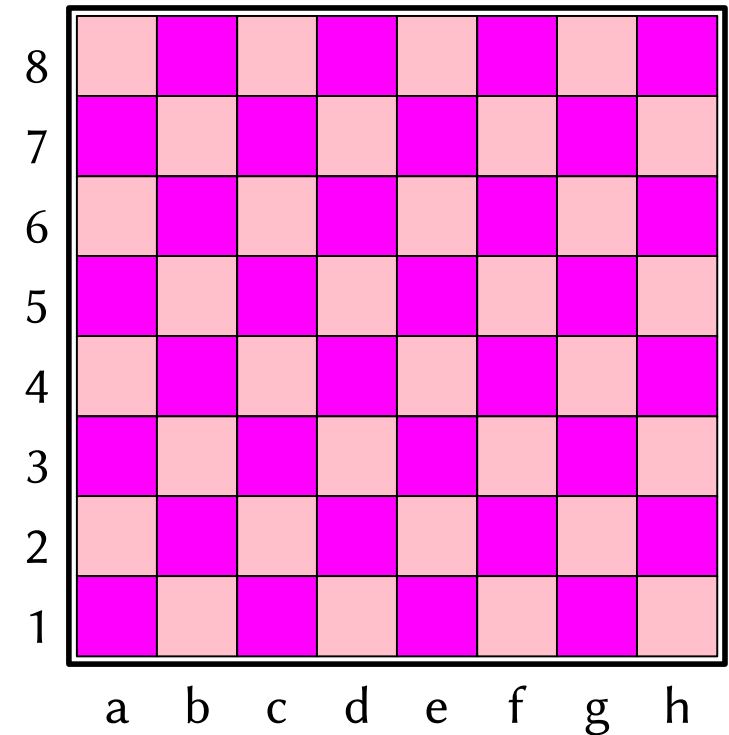


# Sakktábla

- a mezők külön alakzatok (bábok kézi illesztéséhez)

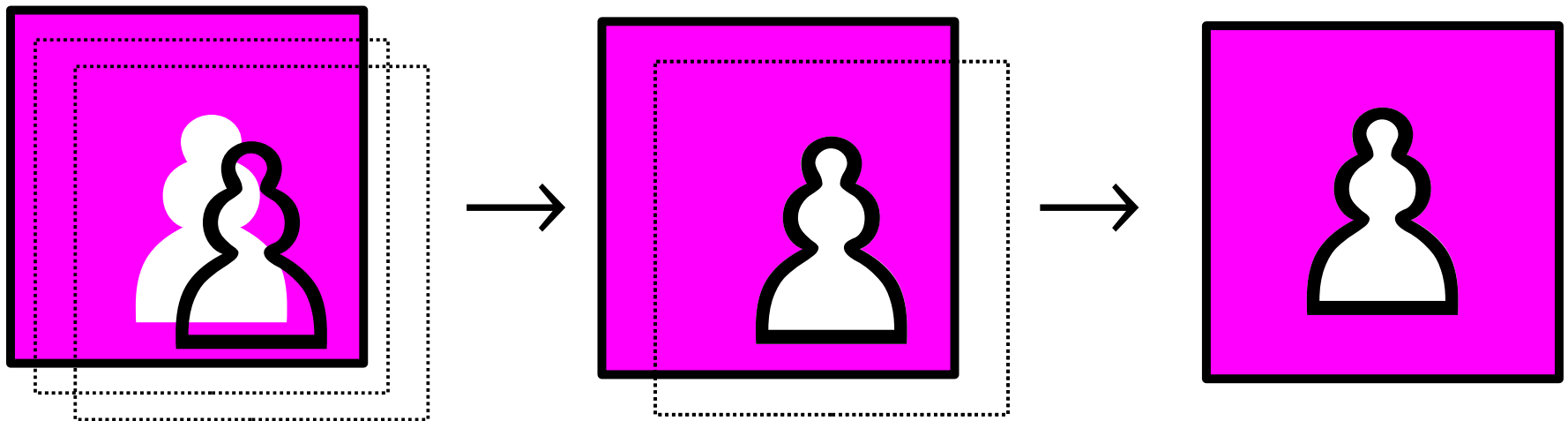
```
ez sakktábla :méret :szín1 :szín2
  betűcsalád „Linux Biolinum G”
  :hely = hely
  :irány = irány
  :színek = [:szín1, :szín2]
  :szín = 1
  tf tv 0 tollszín „fekete”
  b 90 e :méret j 90
  ismét 8 [
    betűméret :méret/1,5
    címke 9 - hányadik
    j 90 e :méret b 90
    ismét 8 [
      töltőszín :színek[:szín] tl
      téglalap [:méret, :méret] tf
      :szín = 1 - :szín
      j 90 előre :méret b 90
    ]
    balra 90 előre :méret * 9 jobbra 90 hátra :méret
    :szín = 1 - :szín
  ]
  fut :i „abcdefgh”-ban [ címke :i j 90 előre :méret b 90 ]
  e :méret*0,5-2 b 90 e :méret/2-2
  tsz „fekete” tv 1 tl
  ismét 4 [ e :méret*8+4 j 90 ] zár tf
  hely :hely
  irány :irány
vége

sakktábla 12 „rózsaszín” „bíbor”
```



# Bábok

- Unicode karakterek (♔♚♛♜♖♗♘♙♞♟♡♢)
- „láthatatlan” négyzetben a pontos kézi illesztéshez
- a fehér Unicode bábok átlátszóak, ezért fehér színű sötét bábokkal készítünk háttérrel
- csoportosítva a fehér bábokat a háttérükkel





# Bábok kirajzolása lejegyzés alapján

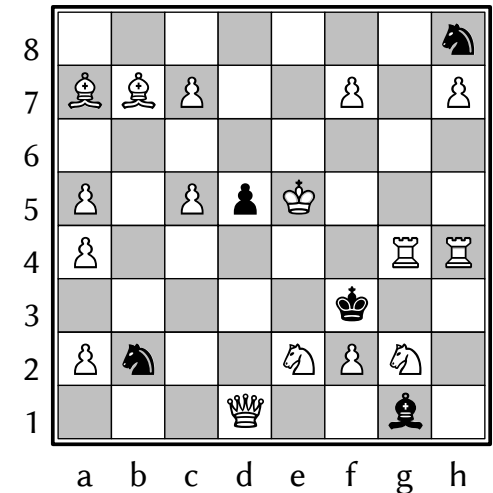
```
ez báb :hossz :betűszín :báb
    téglalap [:hossz, :hossz]
    betűszín :betűszín
    j 90 szöveg :báb b 90
vége
```

```
ez bábok :méret :fehér :fekete
    betűméret :méret
    betűcsalád „FreeSerif”
    tf tollszín „láthatatlan” töltőszín „láthatatlan”
    :adatok = {„K”: „♔♚”, „V”: „♜♞”, „B”: „♖♗”, „F”: „♘♙”, „H”: „♠♡”, „ ”: „♠♡”}
    fut :i „87654321”-ben [
        fut :j „abcdefgh”-ban [
            :benne = :fehér.find(:j+:i) ; egy kis Python
            ha :benne > -1 [
                :adat = :adatok[(„ ” + :fehér)[:benne]]
                :b = :adat[0]
                :háttér = :adat[1]
            ]
            :benne = :fekete.find(:j+:i)
            ha :benne > -1 [
                :adat = :adatok[(„ ” + :fekete)[:benne]]
                :b = :adat[1]
                :háttér = :adat[0]
            ]
        ]
        ha :benne > -1 [
            kép [ báb :méret „fehér” :háttér báb :méret „fekete” :b ]
        ]
    ]
    j 90 e :méret b 90
]
balra 90 előre :méret * 8 jobbra 90 hátra :méret
]
```

vége

; matt egy lépésben (Perkins, 1950)

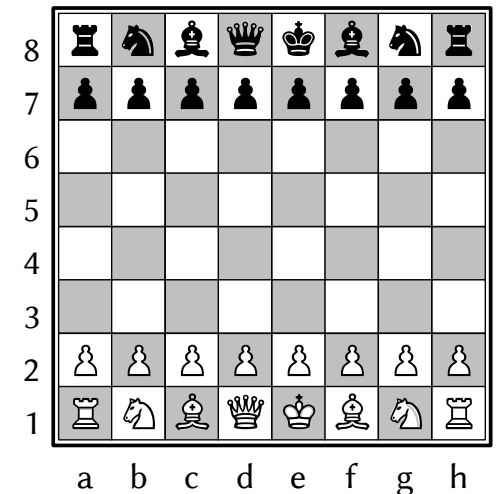
bábok 20 „a2 a4 a5 c5 c7 f2 f7 h7 He2 Hg2 Fa7 Fb7 Bg4 Bh4 Vd1 Ke5” „d5 Hb2 Hh8 Fg1 Kf3”



# Fő eljárás

- ha nem adjuk meg a bábok lejegyzését, a program kirajzolja az alapfelállást, de nem csoportosítja a bábokat a táblával (egyből áthelyezhetők egérrel)

```
törölkép haza elrejt
oldal = oldalméret
:méret = int(be „Kérem a sakkfigurák méretét (pontban):”)
tf hely [oldal[0]/2, :méret] tl irány 0
:fehér = be „Fehér bábok (pl. Ka7 a2), vagy semmi (kézi felállítás):”
ha :fehér == „” [
    kép [ sakktábla :méret „világosszürke” „fehér” ]
    bábok :méret „Ba1 Hb1 Fc1 Vd1 Ke1 Ff1 Hg1 Bh1 a2 b2 c2 d2 e2 f2 g2 h2” „Ba8
Hb8 Fc8 Vd8 Ke8 Ff8 Hg8 Bh8 a7 b7 c7 d7 e7 f7 g7 h7”
    ki „A sakkfigurák levételéhez húzzuk ki a táblát a bábok alól.”
] [
    :fekete = be „Kérem a fekete bábokat:”
    kép [
        kép [ sakktábla :méret „világosszürke” „fehér” ]
        bábok :méret :fehér :fekete
    ]
]
```





<http://www.numbertext.org/logo>