

PostScript

Retro Coding Fun

Nicolas Seriot



Vilnius, 22nd May 2025



Nicolas Seriot

<https://seriot.ch>



Software Engineering Manager

Swissquote Bank

Hes·so

arc gestion

EPFL

Software eng.

MAS Eco. Crime

EMBA innovation



Nature lover

Ultra Trail Runner



Nicolas Seriot

Yesterday at 6:31 PM · Vilnius, Vilnius County

10 K 🌞 Vilnius 🇱🇹

Avec Pierrick

Distance
10.57 km

Avg Pace
5:24 /km

Moving Time
57:02

Elevation Gain
160 m

Avg Power
283 W

Calories
685 Cal

Athlete Intelligence

Strong 10K run in Vilnius with consistent pacing, hitting endurance zones and maintaining a solid pace throughout the workout.



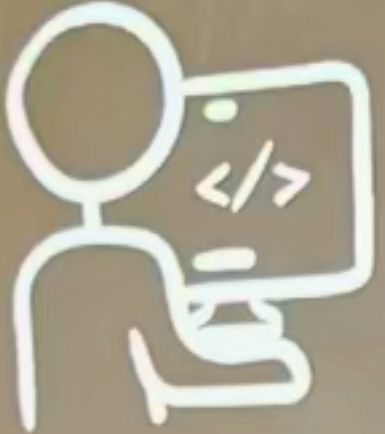
Kvass 🍺

Pork ears 🐷

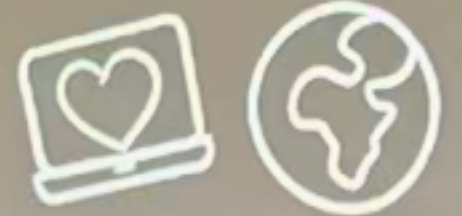
WHEN I STARTED...



COOL STUFF



GOOD TIME



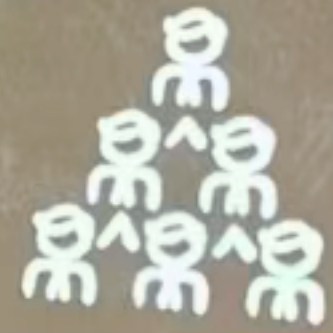
CHANGE THE WORLD



DEMANDING CUSTOMERS



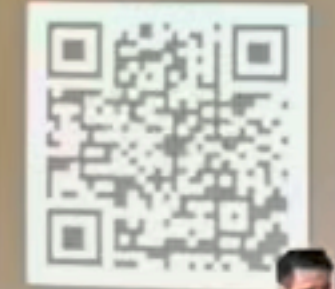
LEGACY



HIERARCHY AND POLITICS

DevDays EUROPE | DEVOPS PRO EUROPE | CyberWise Con

Join at slido.com #1184519



GR8 TECH | GURTAM | LAVINMQ

VNT

WHO WE ARE
Proprietary trading firm deploying quantitative models to support global market efficiency

- Low-latency infrastructure
- Algorithmic execution
- Advanced quantitative models

OUR EDGE

- Quant-Driven Research
- High-Freq Strategies
- In-House Tech Stack
- Cutting-Edge Algorithms

CONTACT US

Deep thinking

Solve puzzles

Explore
technology

Theory

Brain-teasers

Trial & errors

Build stuff

Create new things

Hacking

Art

PSChess — -zsh — 80x24

~/Projects/PSChess — -zsh

```
nseriot@IMAC-GL019 PSChess % cat pschess_compact.ps - | nc 172.20.10.7 9100
```

Deep thinking

Solve puzzles

Explore technology

Theory

Brain-teasers

Trial & errors



Build stuff

Create new things

Hacking

Art



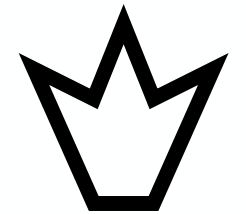
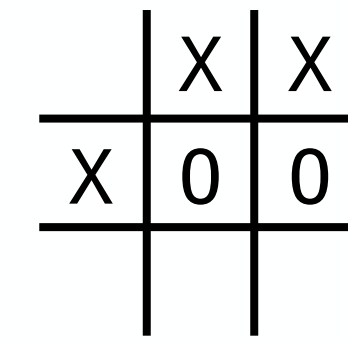
```
12345X@PJL JOBNAME=hplip_rodarnux_3
@PJL SET USERNAME="rodarnux"
@PJL SET JOBNAME="TIPO VEHICULO.pdf"
@PJL ENTER LANGUAGE=POSTSCRIPT
%!PS-Adobe-3.0
%%Invocation: gs -q -dNOPAUSE -dBATCH -dSAFER -dNOMON
=ps2write -dShowAcroForm -sOUTPUTFILE=? -dLanguageLev
false -dNot3CCITT -dNOINTERPOLATE ? ? (ps2write)
%%HiResBoundingBox: 0 0 596.00 842.00
%%Creator: GPL Ghostscript 10021 (ps2write)
%%LanguageLevel: 2
%%CreationDate: D:20240426092825-04'00'
%%For: (rodarnux)
%%Title: (TIPO VEHICULO.pdf)
%%RBI NumCopies: 1
%%Pages: (atend)
%%BoundingBox: (atend)
%%EndComments
%%BeginProlog
10 dict dup h
/DSC
```



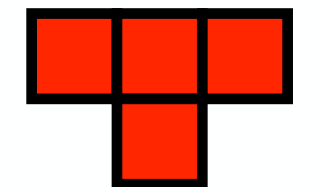
1 Program in PostScript?

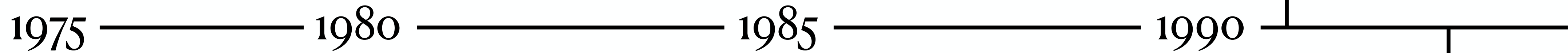
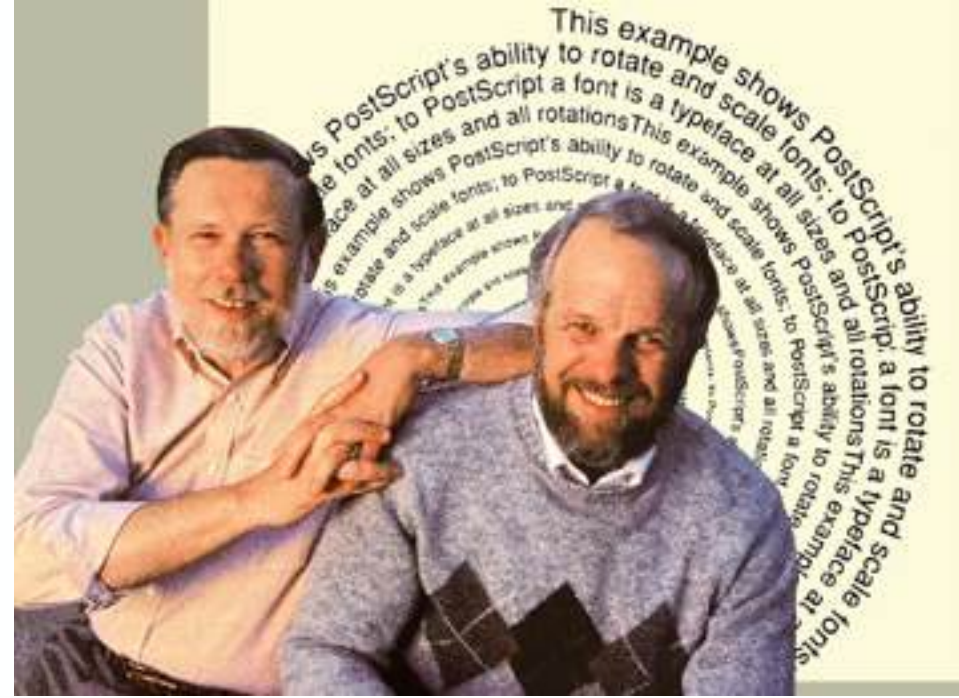


2 Play against a printer?



3 Games for desktop?





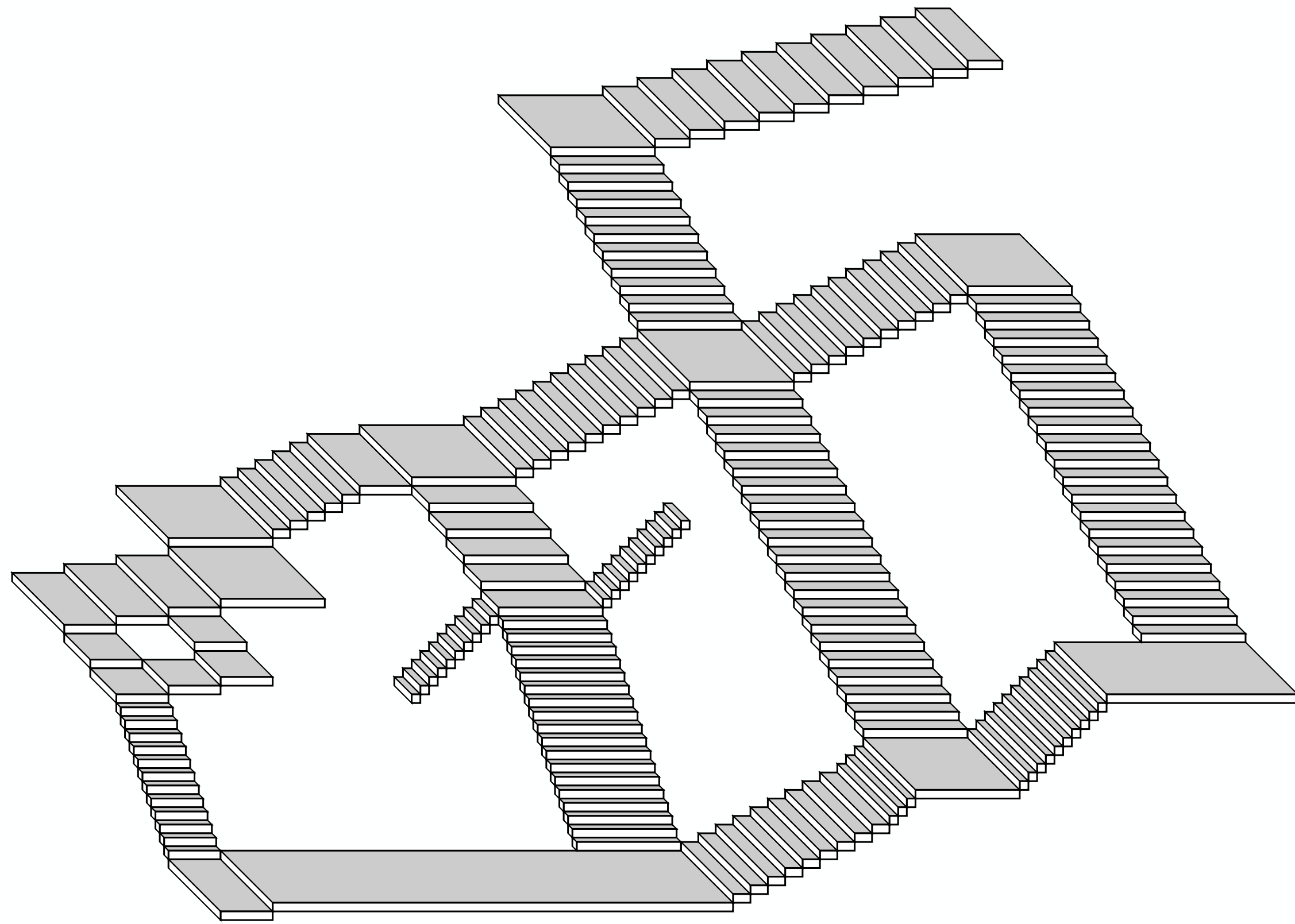
A programming language that generates graphics



Generally software generated, then interpreted on printers



Precursor of PDF



42'000 bytes



842 bytes

stairs.ps

```
<</PageSize[800 600]>>setpagedevice/S 5 def/f{/M exch def/C exch def/h exch def/w exch def gsave M concat C setgray 0 0 w h rectfill 0 setgray 0 0 w h rectstroke grestore}def/b{/H exch def/W exch def gsave W H 2 div 0.8[1 0 -1 1 0 S]f W S 1[1 0 0 1 0 0]f H 2 div S 1[-1 1 0 1 0 0]f grestore}def/U{/H2 exch def{H 2 div neg H 2 div S add translate W H2 b}repeat}def/D{/H exch def{H 2 div H 2 div neg S sub translate W H b}repeat}def/L{/W2 exch def{W2 neg S neg translate W2 H b}repeat}def/R{/W2 exch def{W S translate W2 H b}repeat}def/s{save}def/r{restore}def/W 60 def/H 60 def 100 200 translate 2 30 L 2 30 D s 12 5 D 1 60 D 1 265 R 9 10 R 4 5 R r 2 30 R 1 30 U 1 60 U 1 60 R 1 60 U 5 10 R 1 30 R 1 60 R s 5 20 D s 10 5 L r s 18 5 D r 10 5 R r 10 10 R 1 60 R s 20 10 D 1 60 D 10 5 R 1 110 R r s 10 10 U 1 60 U 10 20 R r 10 10 R 1 60 R 20 10 D
```


Run PostScript

<https://www.ghostscript.com/>



stairs.ps



stairs.pdf

```
ps2pdf stairs.ps
```

```
gs stairs.ps
```

Interactive PostScript

(612,792)

```
% blue rectangle  
0 0 1 setrgbcolor  
100 300 300 100 rectfill  
  
% yellow rectangle  
1 1 0 setrgbcolor  
100 200 300 100 rectfill  
  
% black text  
0 setgray  
110 150 moveto  
/Times-Italic 36 selectfont  
  
(Stand with Ukraine) show
```

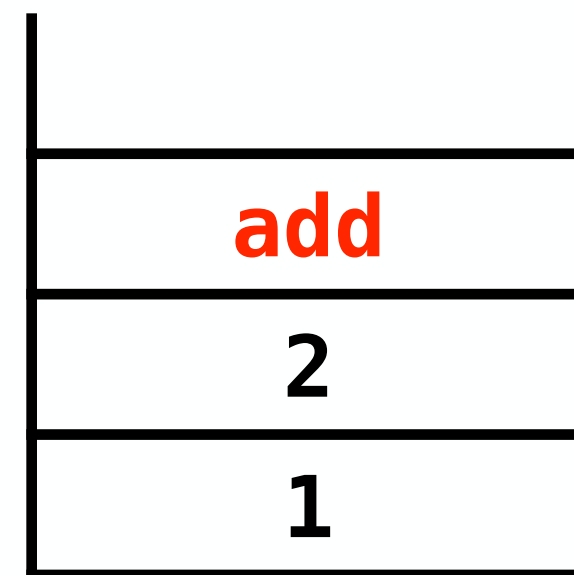
DEMO

(0,0)

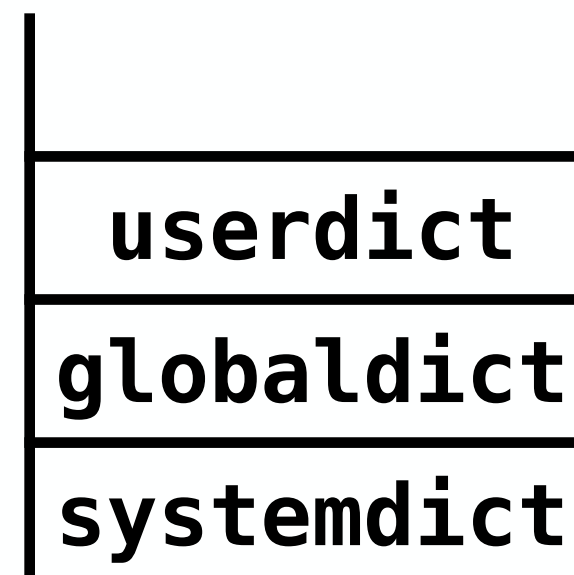


The Language

4 stacks



Operands



Dictionaries



Execution



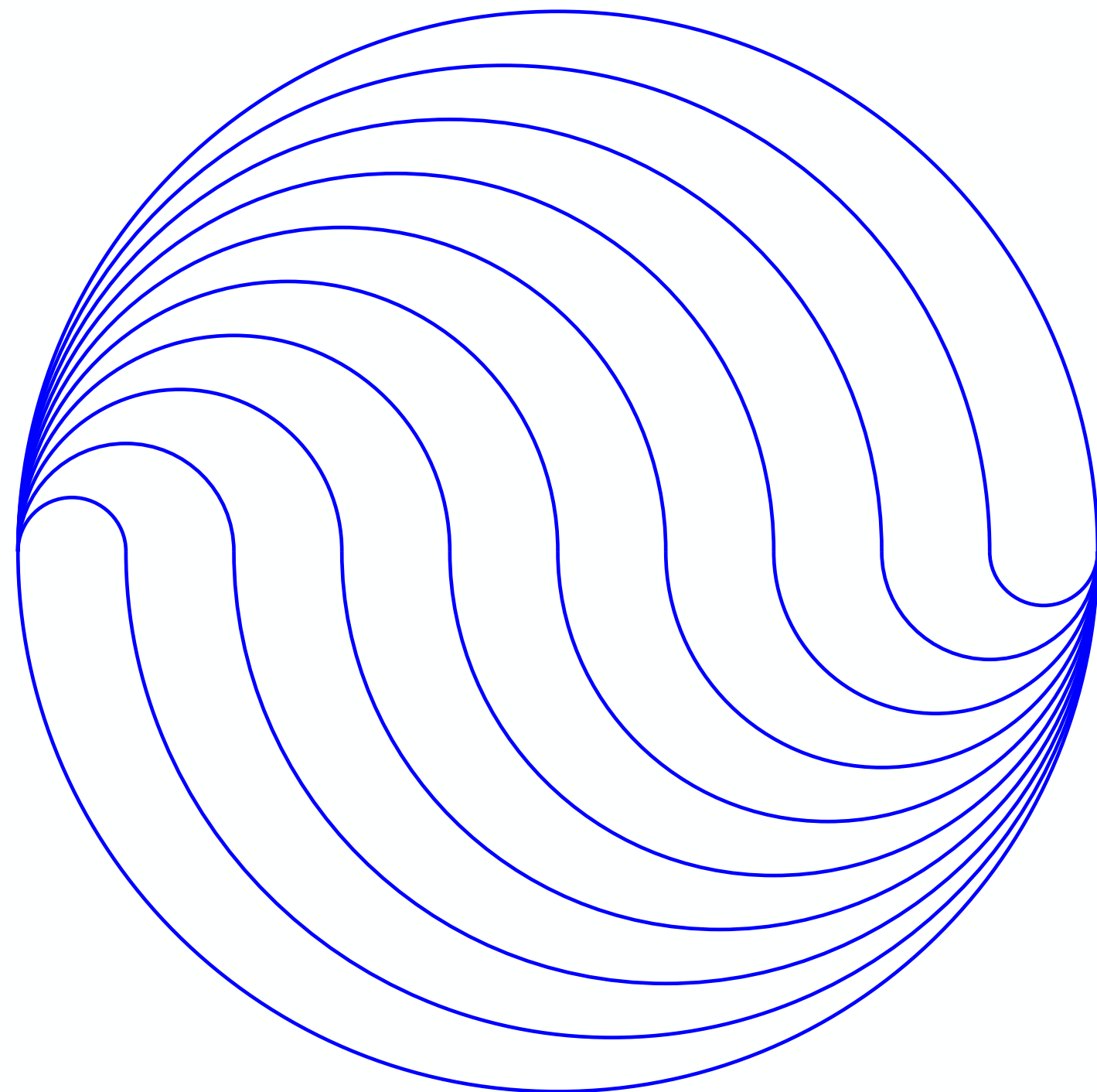
Graphics State

A simple language
with a small,
powerful set of
operators

- PostScript command summary
- Postscript Operators
- Terence Parr PostScript lecture
- seriot.ch Quickies

Circles

Implicit graphic context

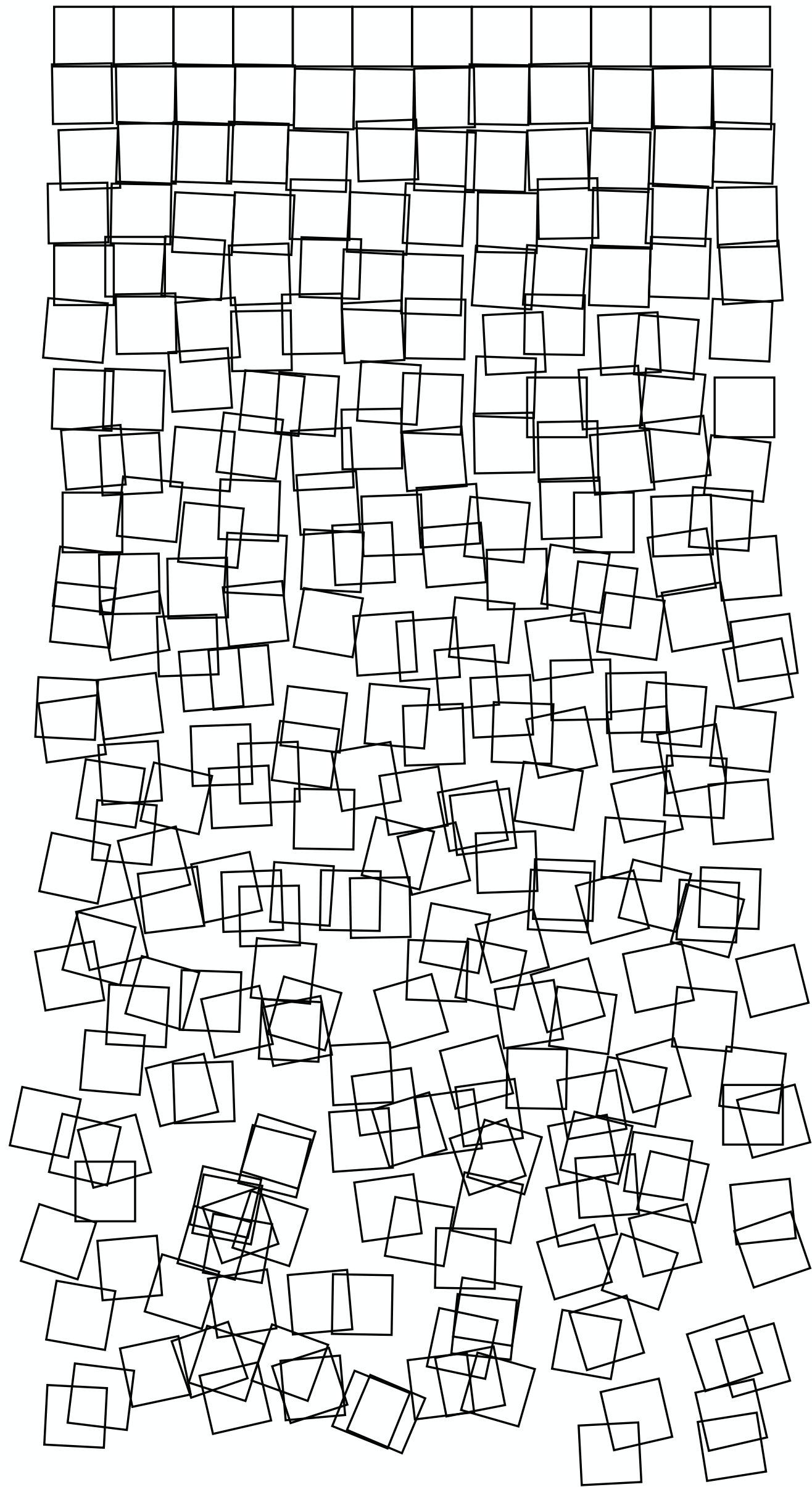


```
0 0 1 setrgbcolor

/N 10 def
/STEP 15 def

1 1 N {
  STEP mul /r exch def
  r 0 r 0 180 arc stroke
  N 2 mul STEP mul r sub 0 r 180 0 arc stroke
} for
```

control
structure



Schotter

Georg Nees, 1965

```
/COLS 12 def  
/ROWS 24 def  
/SIZE 28 def
```

Random values

```
/jig { rand r mod 2 mul r sub 1 add } def
```

```
1 1 COLS {  
  /c exch def  
  1 1 ROWS {  
    /r exch def
```

Geometric transformations

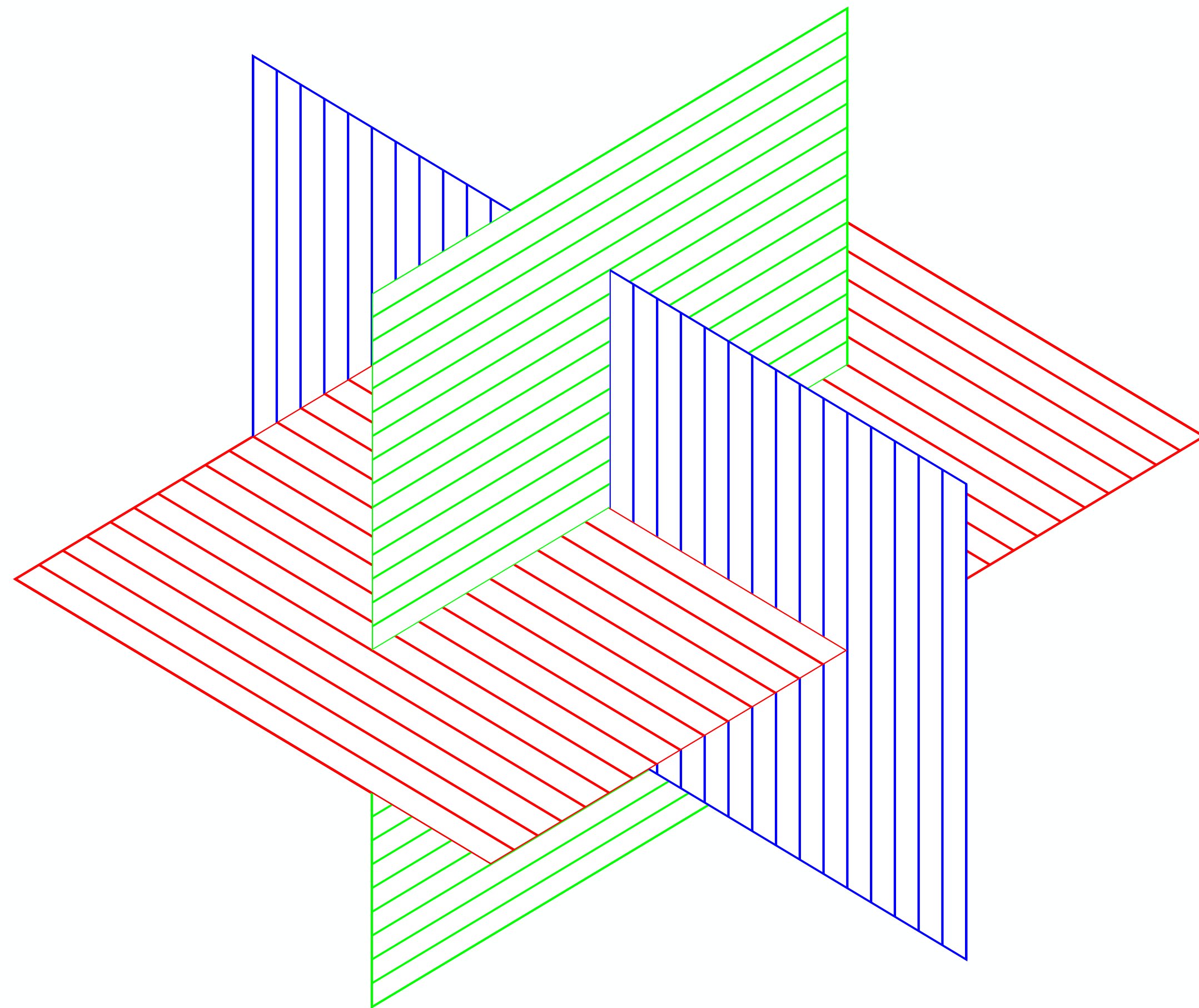
```
gsave  
c SIZE mul r SIZE mul translate  
jig rotate  
jig jig SIZE SIZE rectstroke  
grestore
```

```
} for  
} for
```

```
/S 24 def/j{rand r mod 2 mul r sub 1 add}def 1 1 12{1 1 24{  
/r exch def gsave dup S mul r S mul neg S 2 add S mul add  
translate j rotate j j S S rectstroke grestore}for}for
```

Minified version

Intersecting Planes



%!PS

200 200 translate

/W 300 def
 /H 200 def
 /W2 W 2 div def
 /H2 H 2 div def

```

/P {
  1 setgray
  0 0 W H rectfill

  setrgbcolor
  0 0 W H rectstroke

  0 10 W {
    dup
    0 moveto
    H lineto
  } for

  stroke
} def
  
```

```

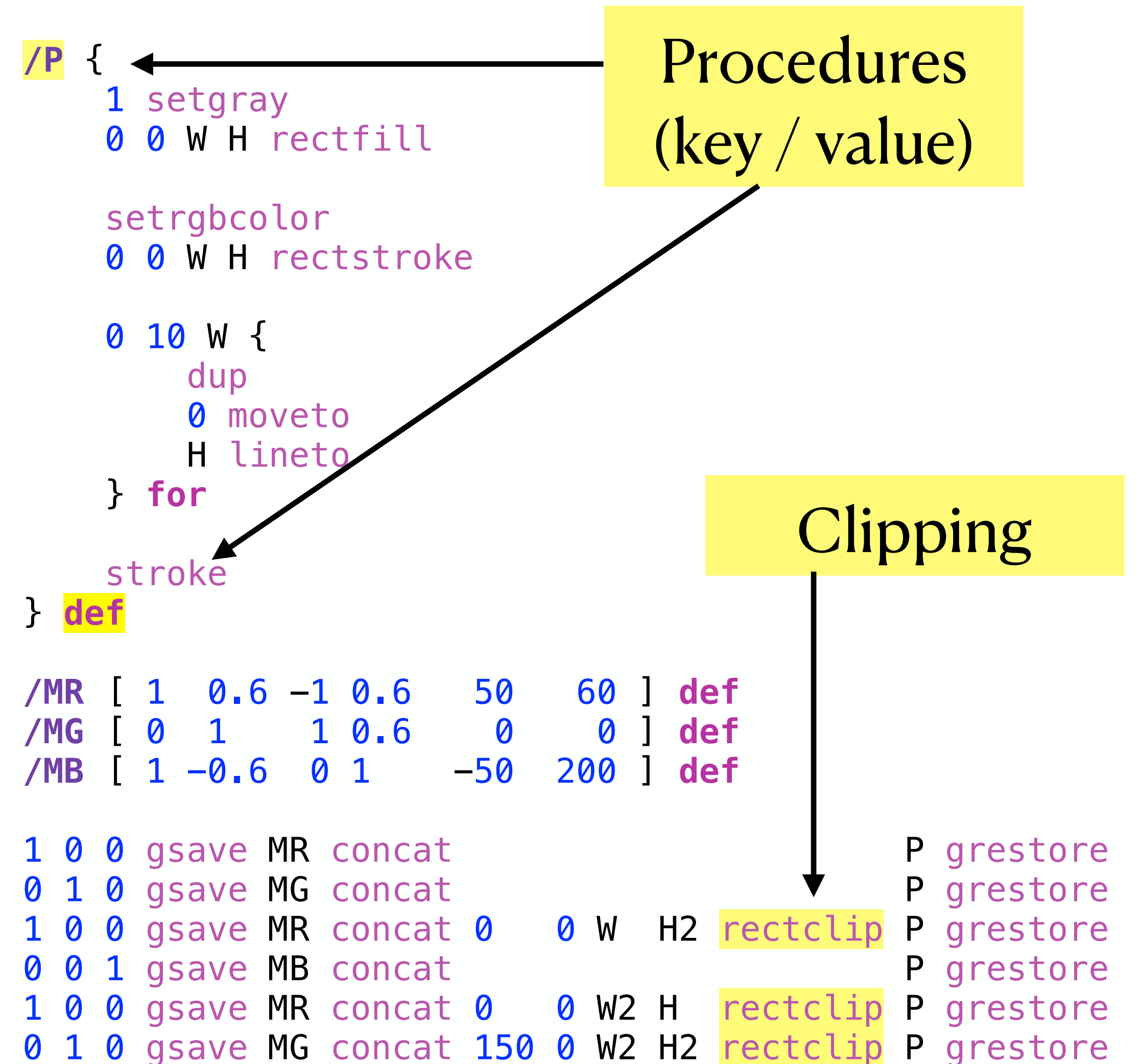
/MR [ 1 0.6 -1 0.6 50 60 ] def
/MG [ 0 1 1 0.6 0 0 ] def
/MB [ 1 -0.6 0 1 -50 200 ] def
  
```

```

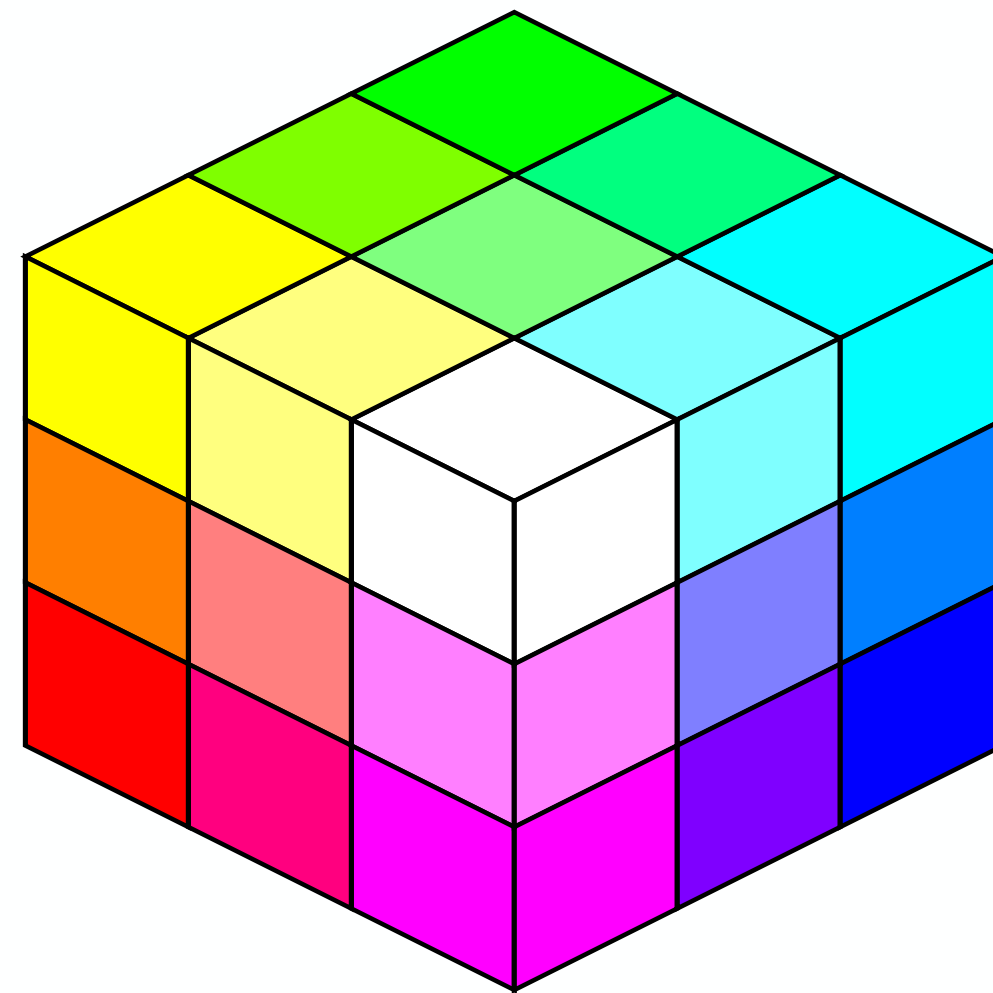
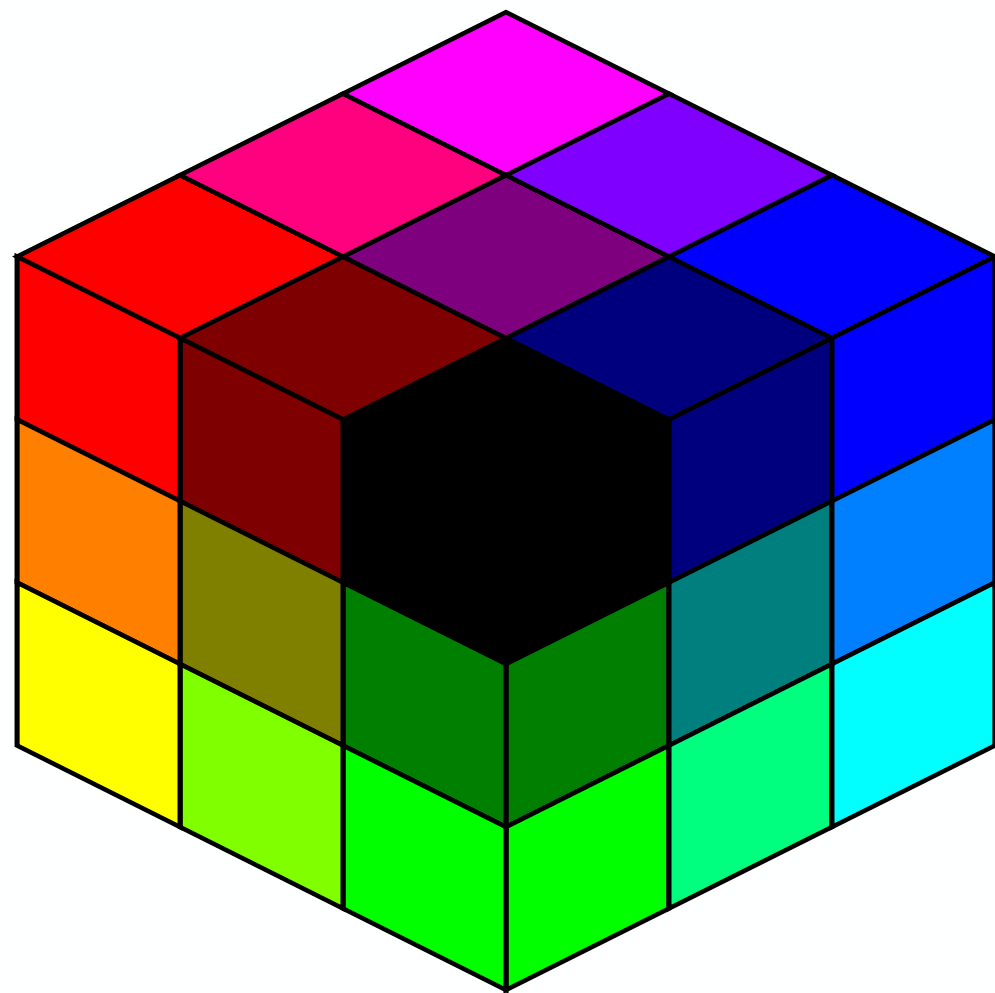
1 0 0 gsave MR concat
0 1 0 gsave MG concat
1 0 0 gsave MR concat 0 0 W H2 rectclip P grestore
0 0 1 gsave MB concat P grestore
1 0 0 gsave MR concat 0 0 W2 H rectclip P grestore
0 1 0 gsave MG concat 150 0 W2 H2 rectclip P grestore
  
```

Procedures
(key / value)

Clipping



Amstrad CPC Color Palette



Geometric transformations

```

/S 32 def
/N 2 def % steps 0-1-2

/_r_g_b_m_DrawSide {
  gsave
  concat
  /B exch def
  /G exch def
  /R exch def
  0 1 N { /x exch def
    gsave
    0 1 N { /y exch def
      R G B setrgbcolor
      0 0 S S rectfill
      0 setgray
      0 0 S S rectstroke
      0 S translate
    } for
    grestore
    S 0 translate
  } for
  grestore
} def

```

```

/_a_b_c_DrawCube {
  /c exch def
  /b exch def
  /a exch def

  gsave
  c 1 eq { [-1 0 0 1 0 0] concat } if % flip left-right
  { b } { c } { a } [ 1 0.5 -1 0.5 0 0 ] _r_g_b_m_DrawSide % top
  { c } { b } { a } [ 1 0.5 0 -1 0 0 ] _r_g_b_m_DrawSide % right
  { a } { b } { c } [-1 0.5 0 -1 0 0 ] _r_g_b_m_DrawSide % left
  grestore
} def

```

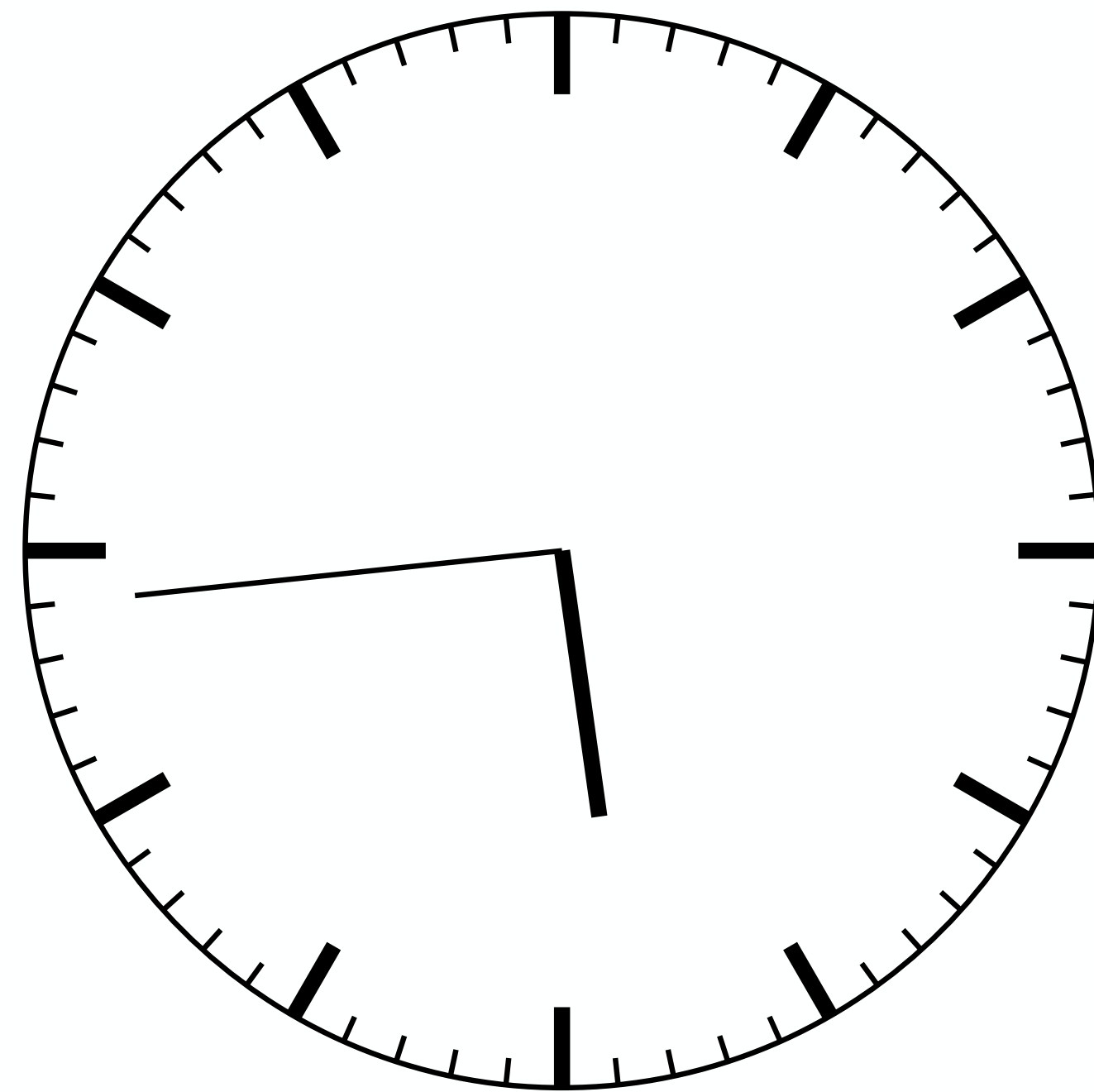
First-class functions,
passed as arguments

```

gsave
{ x N div } { y N div } 0 _a_b_c_DrawCube
240 0 translate
{ N x sub N div } { N y sub N div } 1 _a_b_c_DrawCube
grestore

```

Clock



17:44

```
/datetime (%Calendar%) currentdevparams def
```

```
/H datetime /Hour get def  
/M datetime /Minute get def
```

```
/x 150 def  
/y 170 def  
/r 100 def
```

```
% Clock circle  
newpath  
x y r 0 360 arc  
stroke
```

```
x y translate
```

```
% Side marks  
gsave  
0 1 59 {  
  r 0 moveto  
  5 mod 0 eq {  
    3 setlinewidth  
    r 0.85 mul 0 lineto  
  } {  
    1 setlinewidth  
    r 0.95 mul 0 lineto  
  } ifelse  
  stroke  
  6 rotate  
} for  
grestore
```

ifelse

Environment data in
various dictionaries

```
% Minutes hand
```

```
gsave  
360 60 div M mul  
neg rotate
```

```
0 0 moveto  
0 r 0.8 mul lineto  
1 setlinewidth  
stroke  
grestore
```

```
% Hours hand
```

```
gsave  
360 12 div H mul  
360 12 div 60 div M mul add  
neg rotate
```

```
0 0 moveto  
0 r 0.5 mul lineto  
3 setlinewidth  
stroke  
grestore
```

A Turing-Complete Language

```
/Courier findfont 12 scalefont setfont 32 740 moveto
```

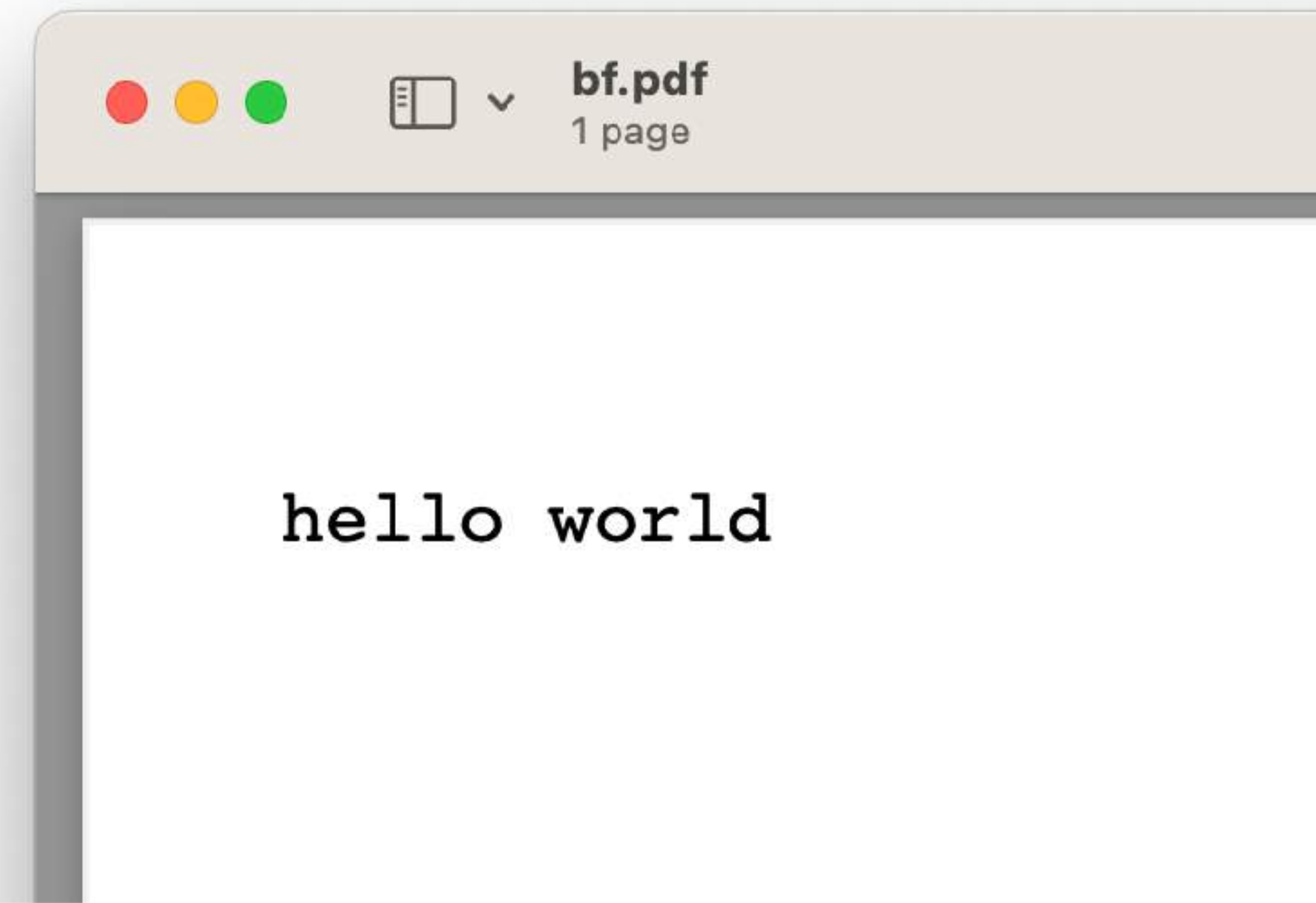
```
/P (+[-[<<+[--->]-[<<<]]>>>-]>-.-- -.>..>.<<<<-.<+.>>>>.>.<<.<-.>.) def
```

```
/d{def}def/p 0 d/_{/p p 1 add d}d/C{P p 1 getinterval}d/M 30000 string d/m 0 d  
/V{M m get}d{p P length ge{exit}if C(+)eq{M m V 1 add 255 and put}if C(-)eq{M m  
V 1 sub 255 and put}if C(>)eq{/m m 1 add d}if C(<)eq{/m m 1 sub d}if C([)eq{p V  
0 eq{mark{C([)eq{0}if C(])eq{pop}if counttomark 0 eq{pop exit}if _}loop}if}if C  
(])eq{/x exch d V 0 ne{/p x d p}if}if C(.)eq{/s 1 string d s 0 V put s show}if  
C(,)eq{/f(%lineedit)(r)file d f read pop M m 3 -1 roll put}if _}loop
```

PostScript can interpret
a TC language, so it's
TC as well

Comman	Description
>	Move the pointer to the right
<	Move the pointer to the left
+	Increment the memory cell at the pointer
-	Decrement the memory cell at the pointer
.	Output the character signified by the cell at the pointer
,	Input a character and store it in the cell at the pointer
[Jump past the matching] if the cell at the pointer is 0
]	Jump back to the matching [if the cell at the pointer is

bf.ps



<https://esolangs.org/wiki/Brainfuck>

PostScript Capabilities

Interpreters can
interact with (part of)
their environment

stdin
→
←
stdout,
stderr



↕ File System

- ✗ syscalls
- ✗ network access
- ✗ threads
- ✗ Unicode support
- ✗ strong typing



1985

- **A programming language**
- **Generates graphics**
- Program might never stop
- Number of pages unknown

a program



1993

- **A document format**
- **Describes final appearance**
- Predictable behavior
- Can reference individual pages
- Can contain hyperlinks, forms...
JavaScript!

a doc format

Agenda



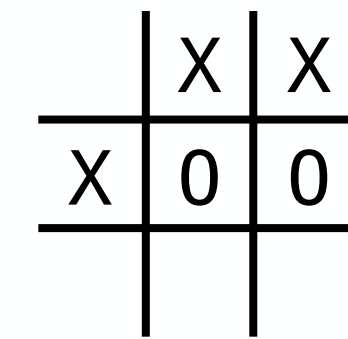
RICOH M C240FW

< 300 €

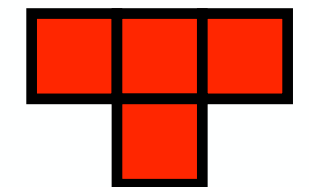
① Program in PostScript?



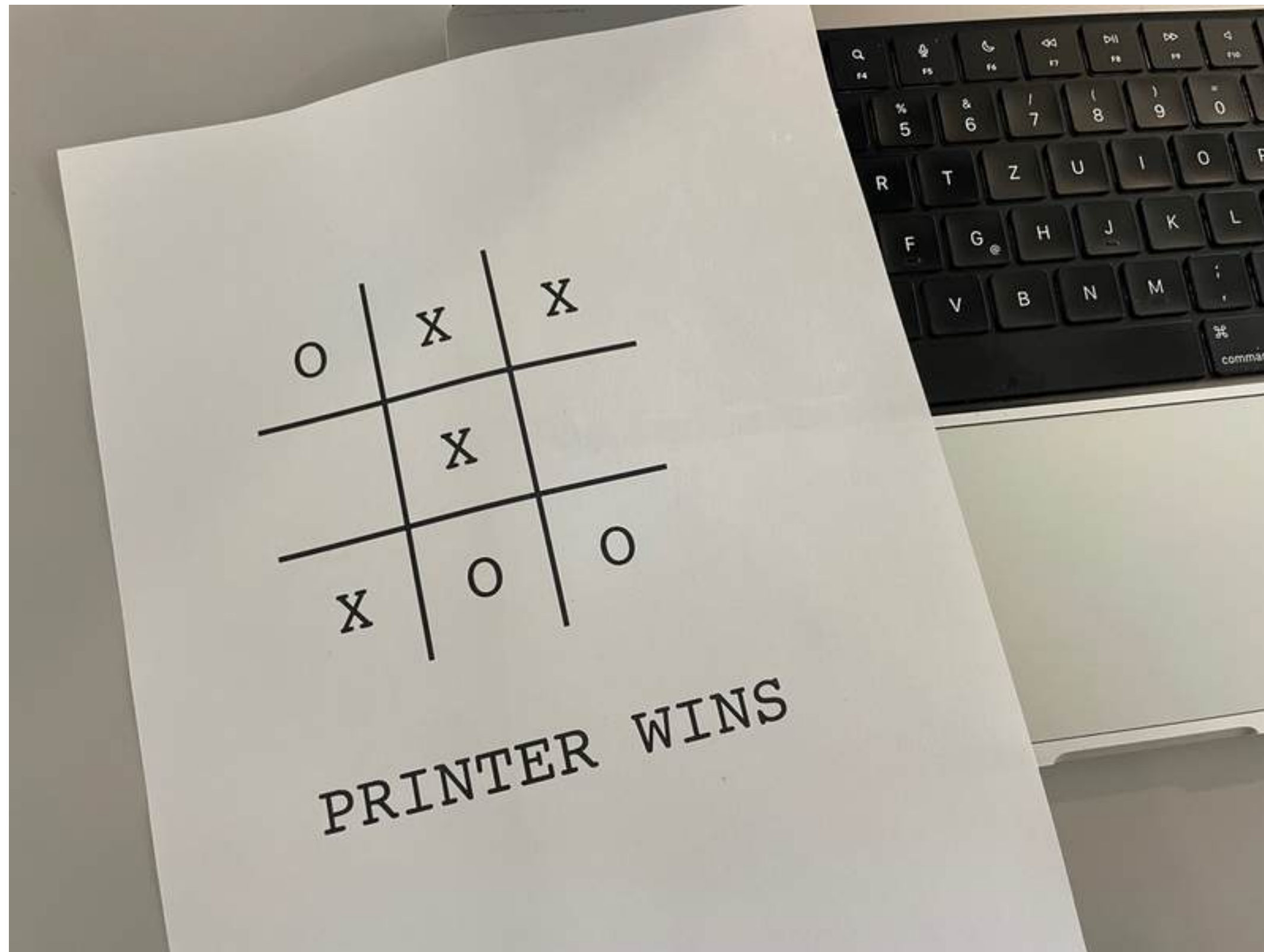
② Play against a printer?



③ Games for desktop?



Tic-Tac-Toe



One-way Communication

```
cat stairs.ps | nc 172.16.158.21 9100
```

Executive Mode

```
nc 172.20.10.4 9100  
%!PS  
executive
```



Bi-directional Communication

```
cat x.ps - | nc 192.168.2.10 9100
```

Dash “-“ to read stdin

Read user Input

```
(%lineedit) (r) file ( ) readline pop
```

Data Structure

```
/b (.....) def
/numberOfFreePos {
  0 % reduce
  b { 46 eq { 1 add } if } forall
} def
```

	X	X
X	0	0

Algorithm

LOOP:

IF someone_wins:

EXIT

ELIF can_win:

win

ELIF can_block_human:

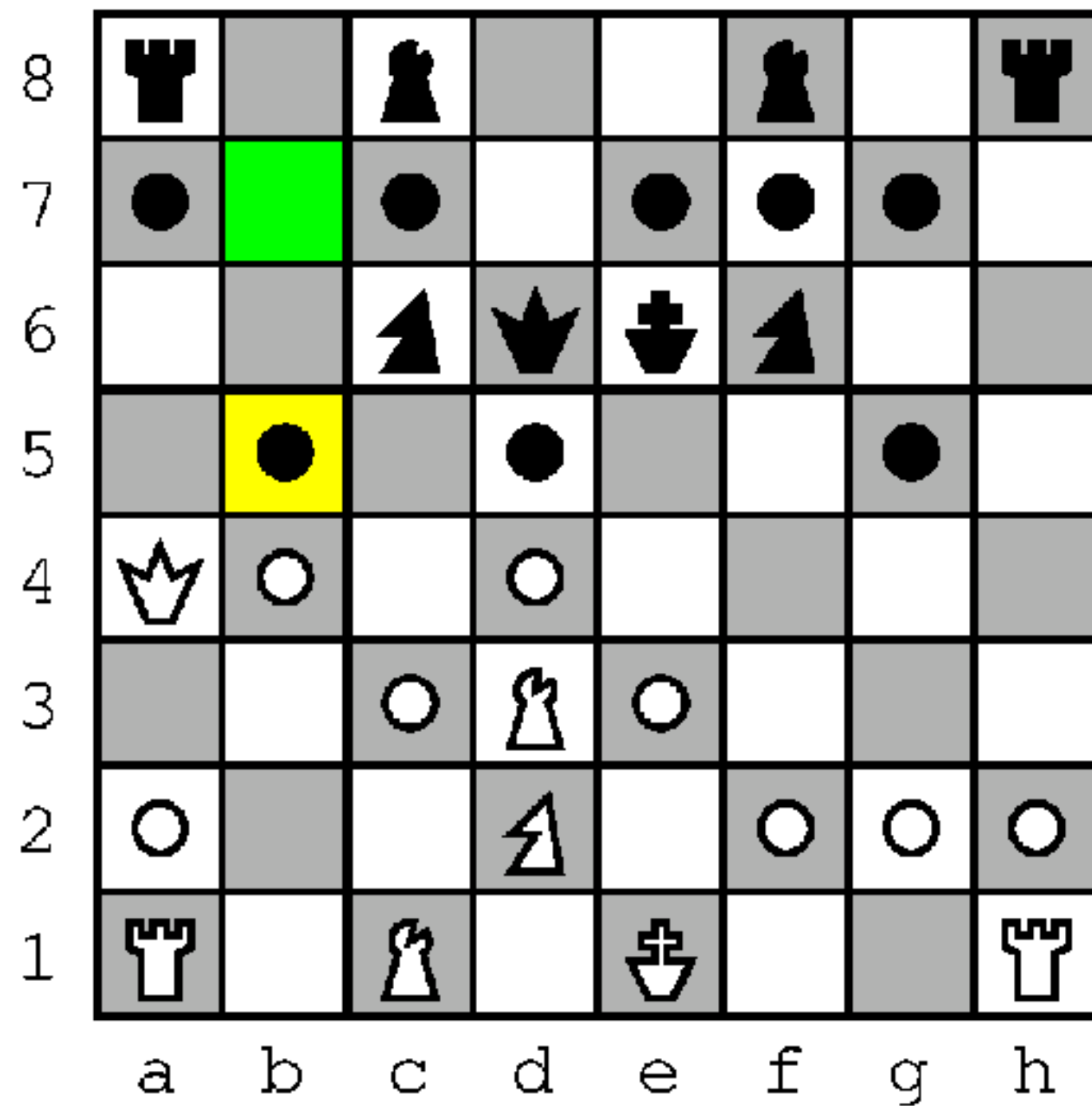
block_human

ELSE:

play_randomly

get_human_input

PSChess



black b7 b5

-375

-

white turn



r...r...

pppkb.Q.

..bqp...

...p..Np

...B.Pn.

..P...PB

PP.NP..P

..KR...R

black h8 e8

320

P.....

npp.....

-

white turn

DEMO

*"PostScript is deep magic that I admire from a distance.
This is some mad scientist stuff. Great work."
-tambourine_man on Hacker News*

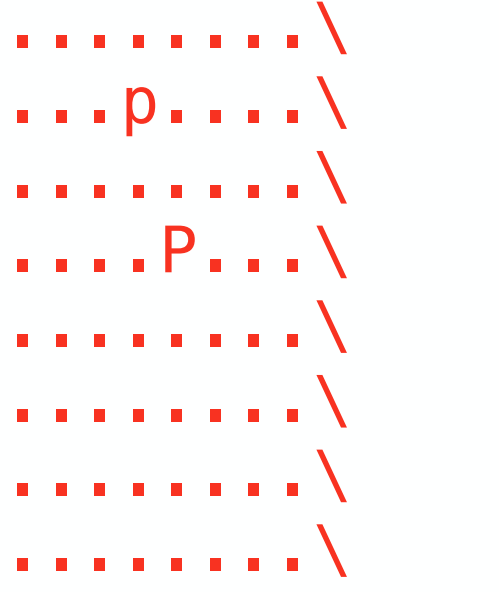
Imports and Tests

test_logic.ps

```
/TestEnPassant {
  (TestEnPassant) CS_PUT
```

save

```
/board (\
```



Data Structure

```
) _s_Dup_s_ def
```

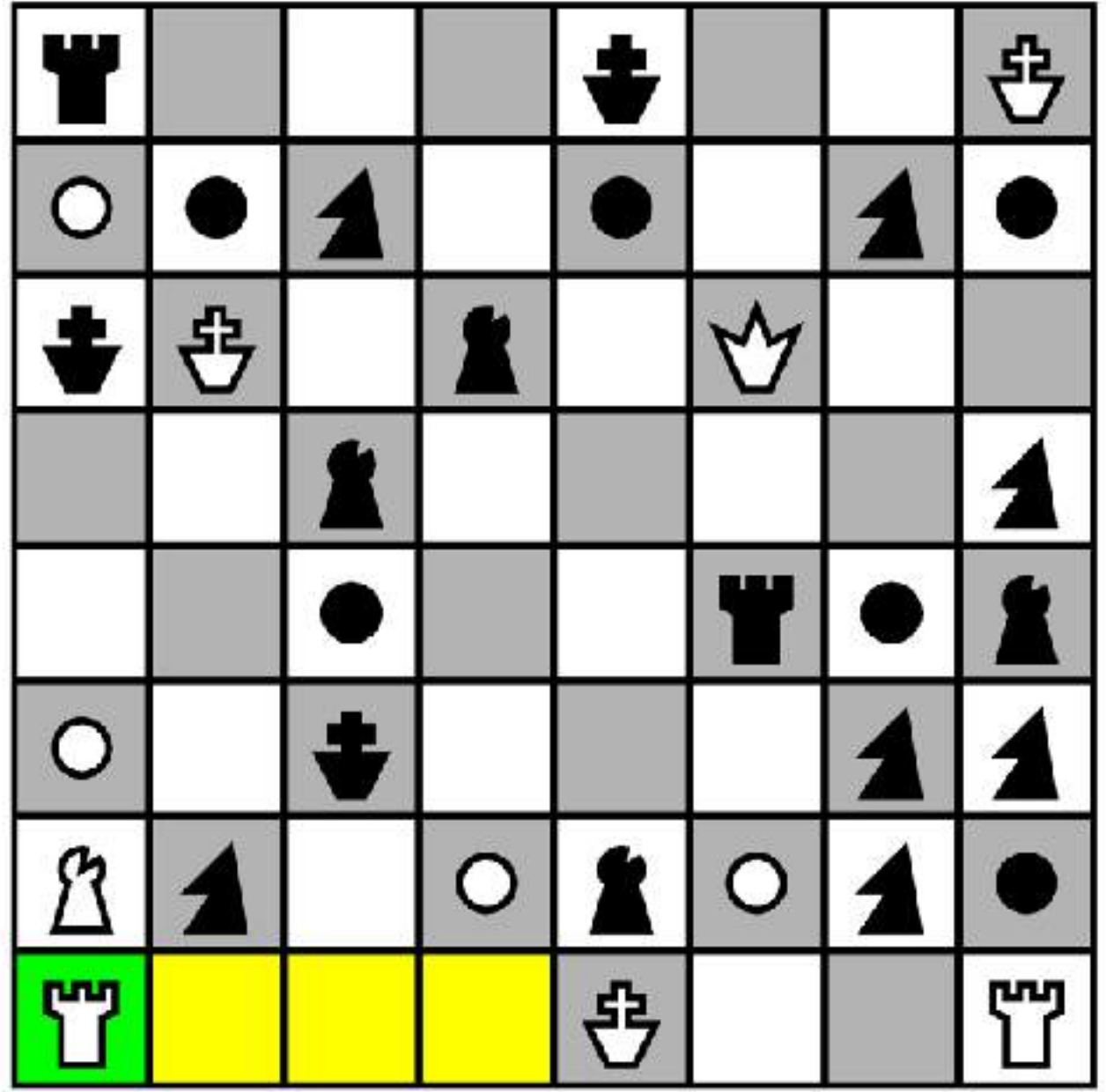
*% black pawn moves 2 steps forward
% white takes it "en-passant"*

```
GameState (CURRENT_PLAYER) (black) put
```

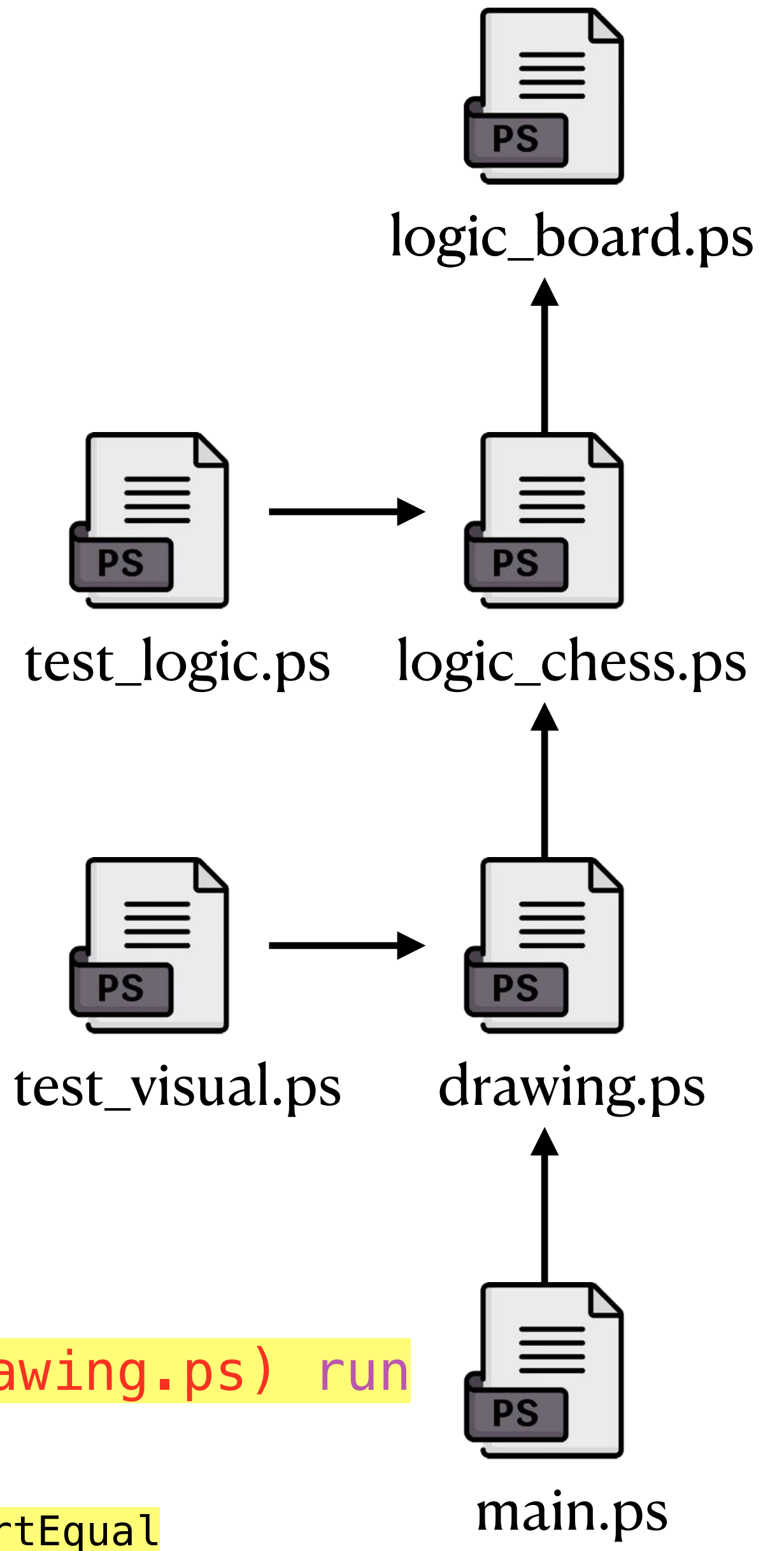
```
(TEST TestEnPassant.1) { GameState (EN_PASSANT_BLACK_CANDIDATE) get } _s_proc_AssertNull
```

```
board (d7) (d5) false _board_fc_tc_dry_Move_canMove_squaresStatus_msg_
/_canMove exch def
/_highlights exch def
/_msg exch def
```

```
(TEST TestEnPassant.2) { GameState (EN_PASSANT_BLACK_CANDIDATE) get } { (d5) } _s_p1_p2_AssertEqual
```

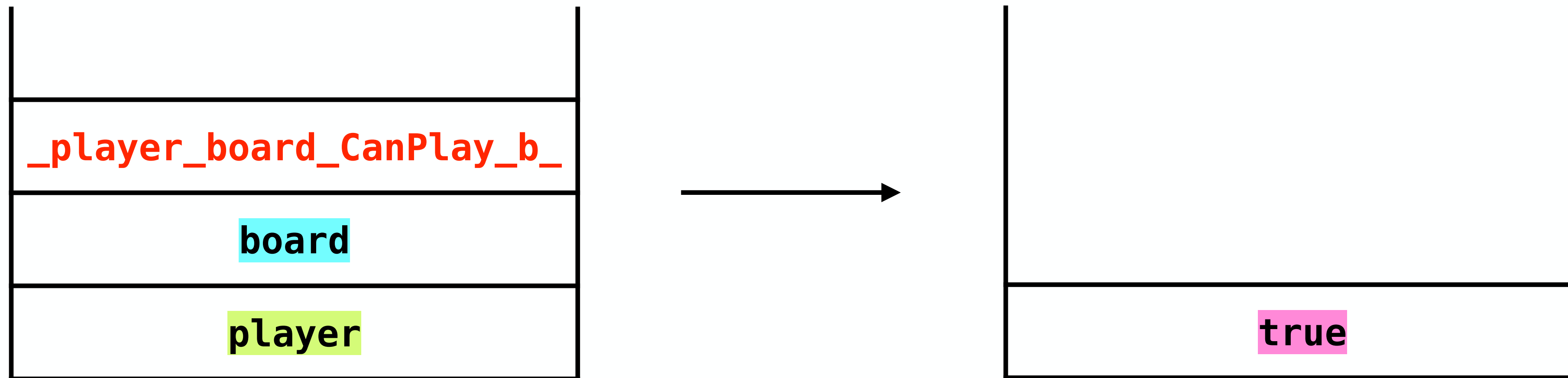


test_visual.ps



(drawing.ps) run

Naming Conventions



```
/_player_board_CanPlay_b_ { } def
```

```
player board _player_board_CanPlay_b_ { } if
```

```
def canPlay(player, board) -> bool
```

UI Design

```
/m { moveto } def  
/l { lineto } def
```

```
/p { newpath SQUARE_SIZE 2 div SQUARE_SIZE 2 div 4 0 360 arc closepath } def
```

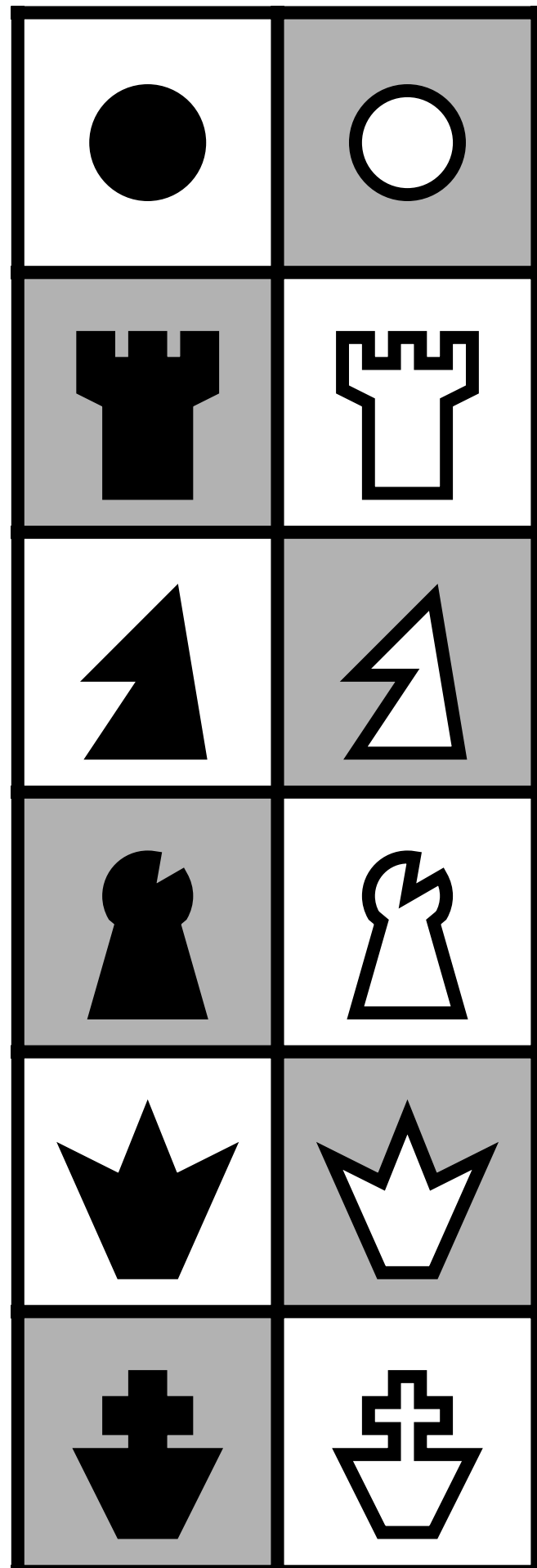
```
/r { newpath 5 15 m 7 15 l 7 13 l 9 13 l 9 15 l 11 15 l 11 13 l 13 13 l  
13 15 l 15 15 l 15 11 l 13 10 l 13 3 l 7 3 l 7 10 l 5 11 l 5 15 l  
closepath } def
```

```
/n { newpath 12 15 m 14 3 l 6 3 l 10 9 l 6 9 l closepath } def
```

```
/b { newpath 14 3 m 12 10 l 10 12 3 -30 30 arc  
10 12 l 10 12 3 80 210 arc 8 10 l 6 3 l closepath } def
```

```
/q { newpath 4 12 m 8 10 l 10 15 l 12 10 l 16 12 l 12 3 l 8 3 l closepath } def
```

```
/k { newpath 8 3 m 5 9 l 9 9 l 9 11 l 7 11 l 7 13 l 9 13 l 9 15 l 11 15 l  
11 13 l 13 13 l 13 11 l 11 11 l 11 9 l 15 9 l 12 3 l closepath } def
```



Evaluation Function

/WhitePieceValueDict

<<

```
(P) 100
(N) 320
(B) 330
(R) 500
(Q) 900
(K) 20000
(.) 0
```

>> def

```
(N) [ -50 -40 -30 -30 -30 -30 -40 -50
      -40 -20  0  0  0  0 -20 -40
      -30  0 10 15 15 10  0 -30
      -30  5 15 20 20 15  5 -30
      -30  0 15 20 20 15  0 -30
      -30  5 10 15 15 10  5 -30
      -40 -20  0  5  5  0 -20 -40
      -50 -40 -30 -30 -30 -30 -40 -50 ]
```

```
(B) [ -20 -10 -10 -10 -10 -10 -10 -20
      -10  0  0  0  0  0  0 -10
      -10  0  5 10 10  5  0 -10
      -10  5  5 10 10  5  5 -10
      -10  0 10 10 10 10  0 -10
      -10 10 10 10 10 10 10 -10
      -10  5  0  0  0  0  5 -10
      -20 -10 -10 -10 -10 -10 -10 -20 ]
```

% ...

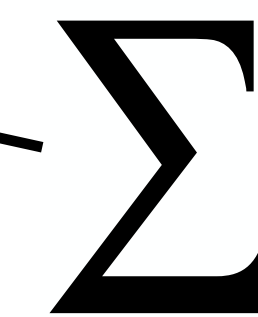
(Tomasz Michniewski)

8								
7								
6								
5								
4								
3								
2								
1								
	a	b	c	d	e	f	g	h

black h8 e8

320

-
white turn

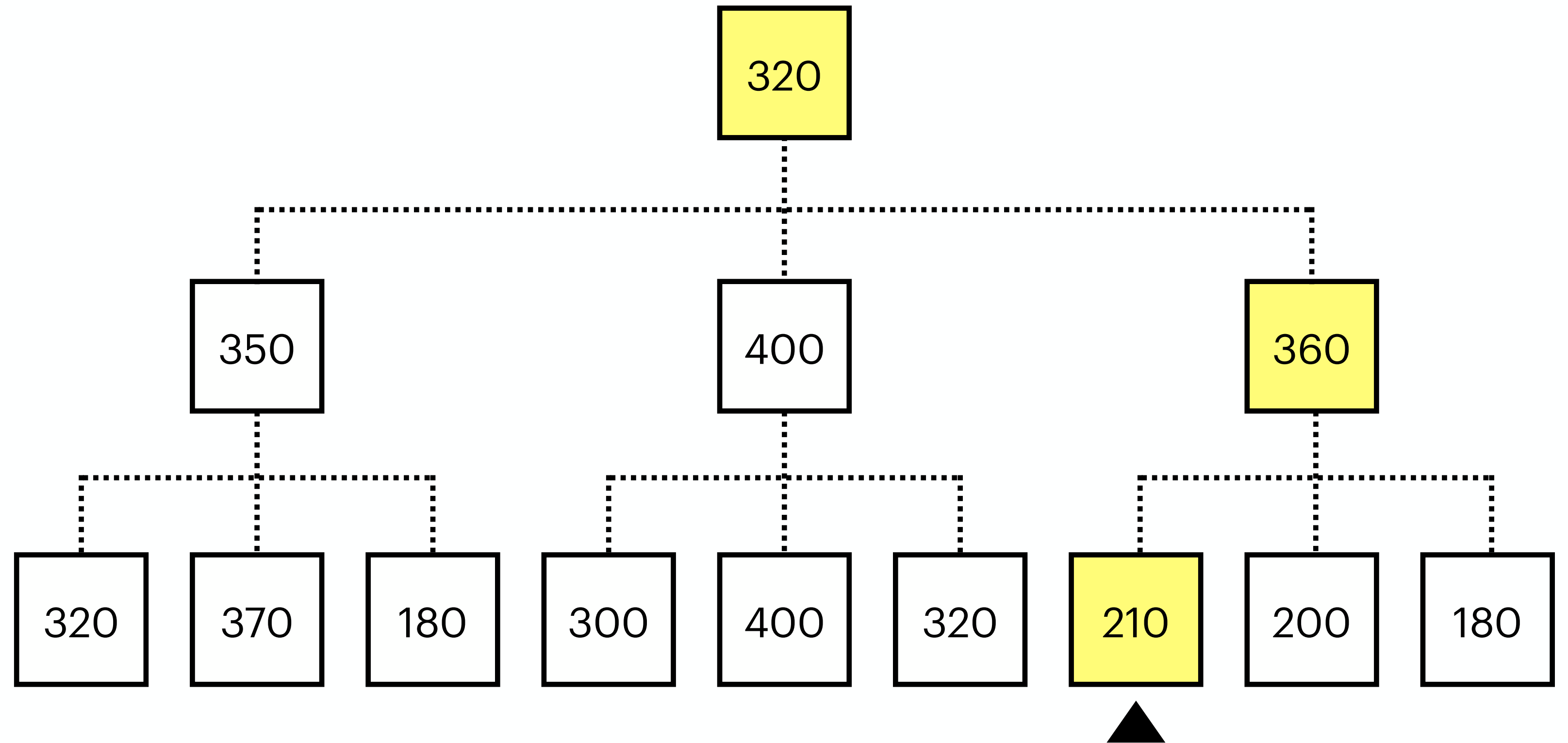


Minimax

Black played

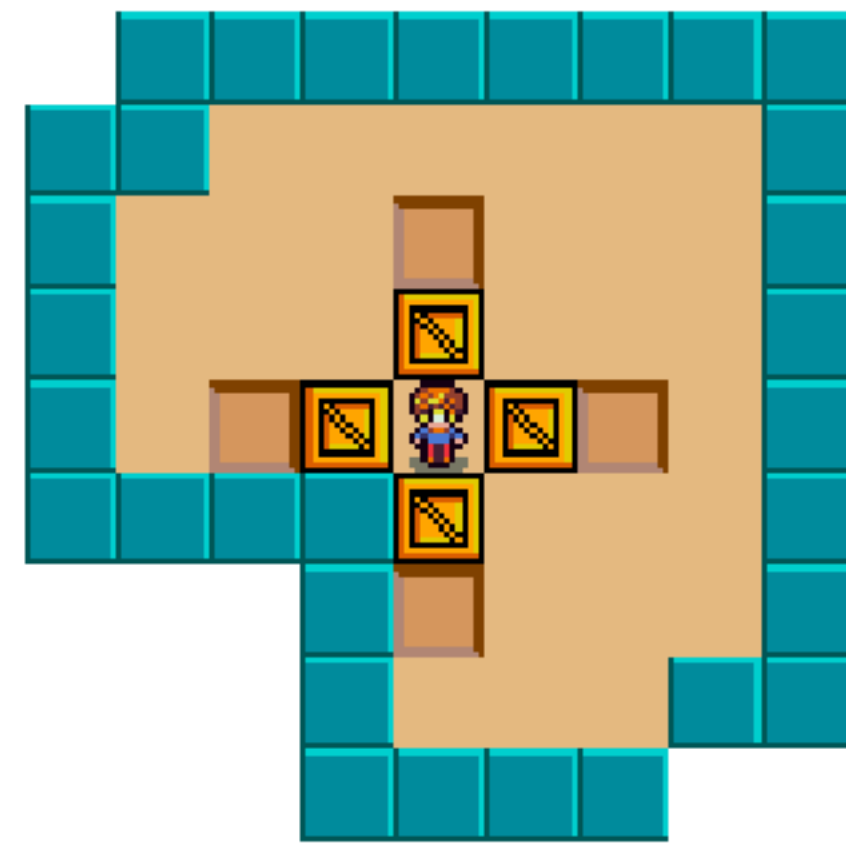
White may play

Black may play



Minimize opponent's advantage

Agenda

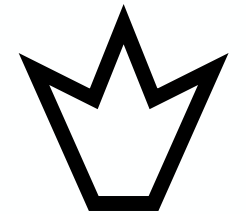
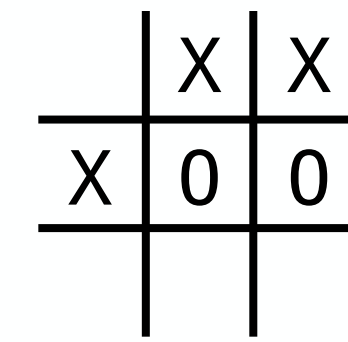


Level:0 Moves:0 Push:0

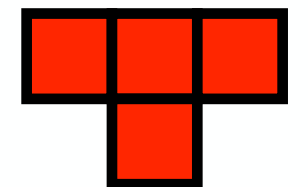
① Program in PostScript?



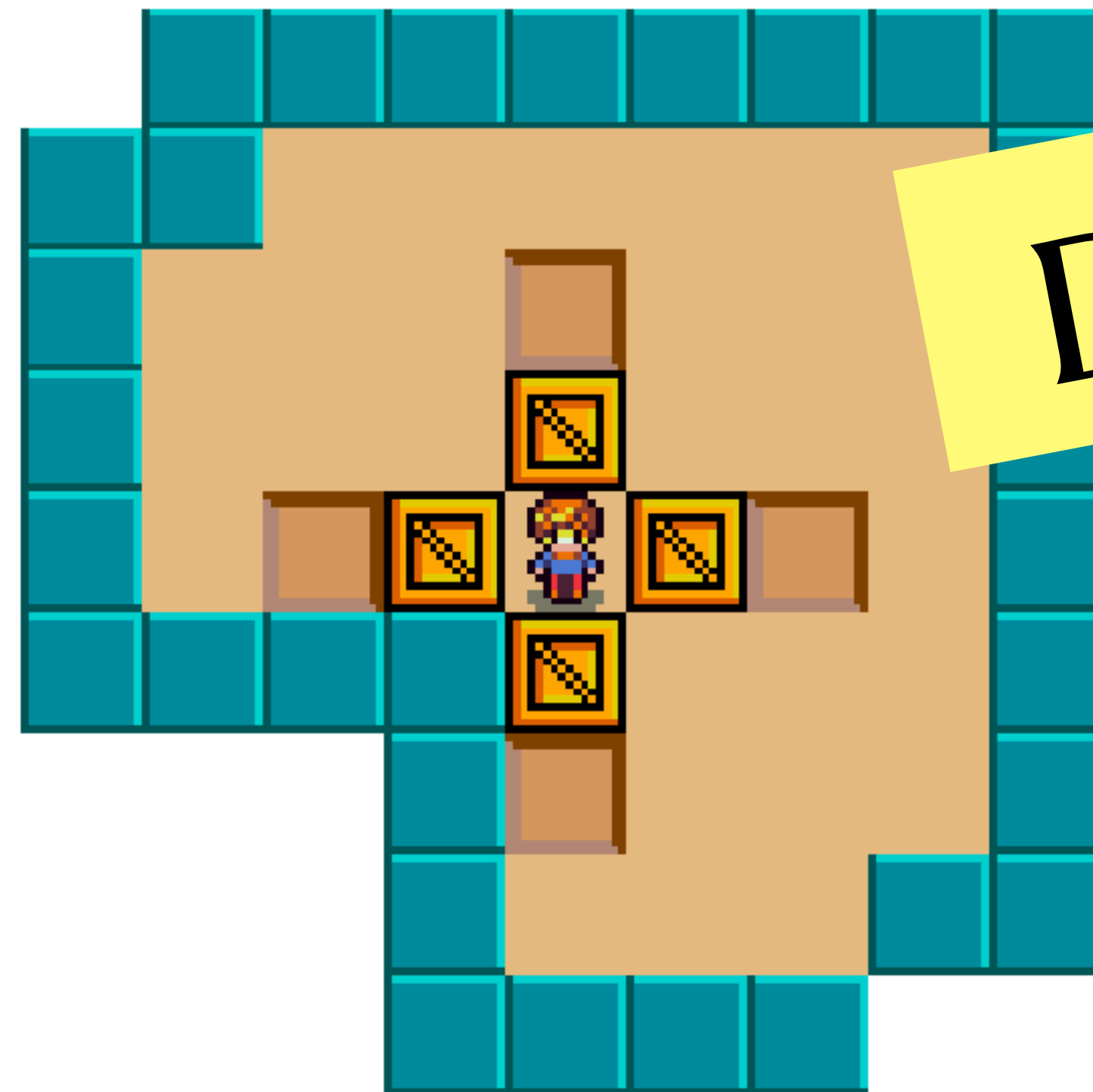
② Play against a printer?



③ Games for desktop?



PSSokoban



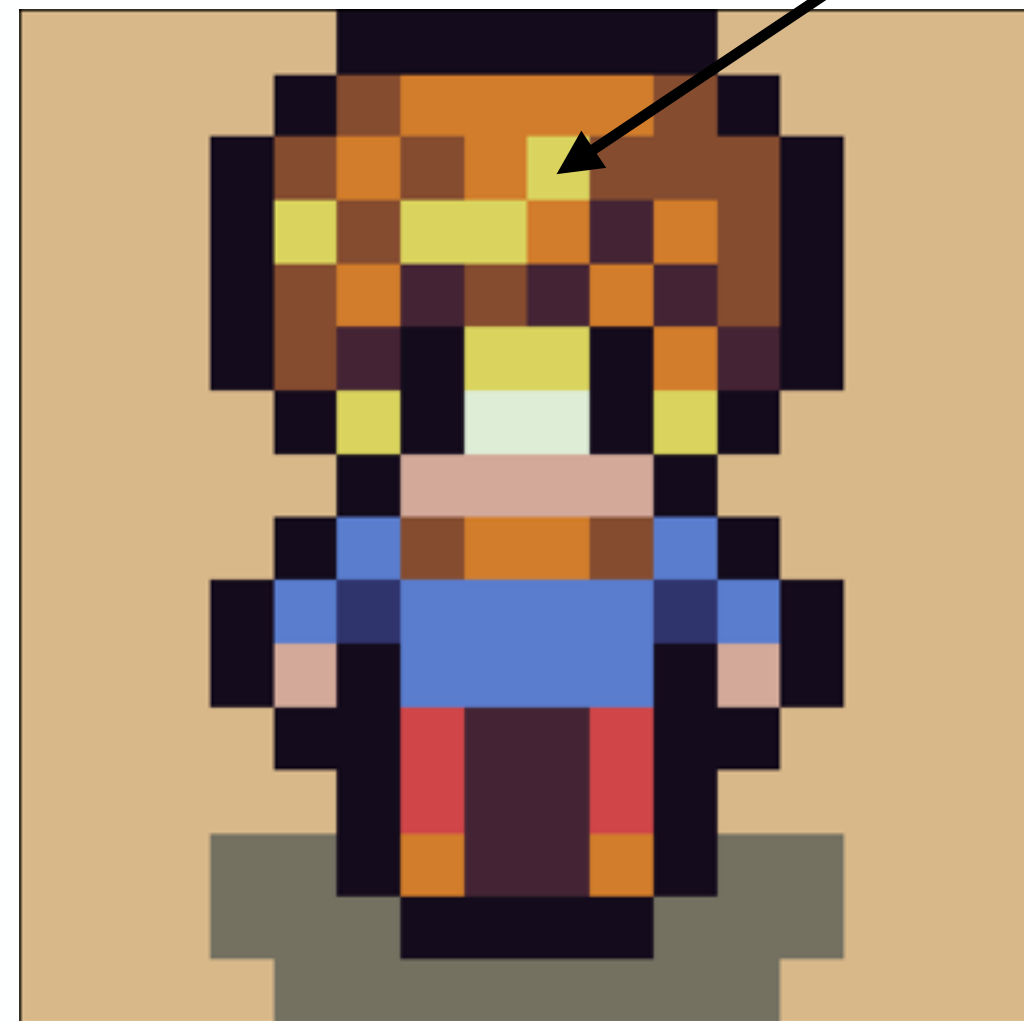
DEMO

Level:0

Moves:0

Push:0

Images



<https://opengameart.org/>

DD5

```
/DeviceRGB setcolorspace

/imageData {
    (
        DB8 DB8 DB8 DB8 DB8 101 101 101 101 101 101 DB8 DB8 DB8 DB8 DB8
        DB8 DB8 DB8 DB8 101 843 D72 D72 D72 D72 843 101 DB8 DB8 DB8 DB8
        DB8 DB8 DB8 101 843 D72 843 D72 DD5 843 843 843 101 DB8 DB8 DB8
        DB8 DB8 DB8 101 DD5 843 DD5 DD5 D72 423 D72 843 101 DB8 DB8 DB8
        DB8 DB8 DB8 101 843 D72 423 843 423 D72 423 843 101 DB8 DB8 DB8
        DB8 DB8 DB8 101 843 423 101 DD5 DD5 101 D72 423 101 DB8 DB8 DB8
        DB8 DB8 DB8 DB8 101 DD5 101 DED DED 101 DD5 101 DB8 DB8 DB8 DB8
        DB8 DB8 DB8 DB8 DB8 101 DA9 DA9 DA9 DA9 101 DB8 DB8 DB8 DB8 DB8
        DB8 DB8 DB8 DB8 101 57C 843 D72 D72 843 57C 101 DB8 DB8 DB8 DB8
        DB8 DB8 DB8 101 57C 336 57C 57C 57C 57C 336 57C 101 DB8 DB8 DB8
        DB8 DB8 DB8 101 DA9 101 57C 57C 57C 57C 101 DA9 101 DB8 DB8 DB8
        DB8 DB8 DB8 DB8 101 101 D44 423 423 D44 101 101 DB8 DB8 DB8 DB8
        DB8 DB8 DB8 DB8 DB8 101 D44 423 423 D44 101 DB8 DB8 DB8 DB8 DB8
        DB8 DB8 DB8 776 776 101 D72 423 423 D72 101 776 776 DB8 DB8 DB8
        DB8 DB8 DB8 776 776 776 101 101 101 101 776 776 776 DB8 DB8 DB8
        DB8 DB8 DB8 DB8 776 776 776 776 776 776 776 DB8 DB8 DB8 DB8
    )
} def
<<

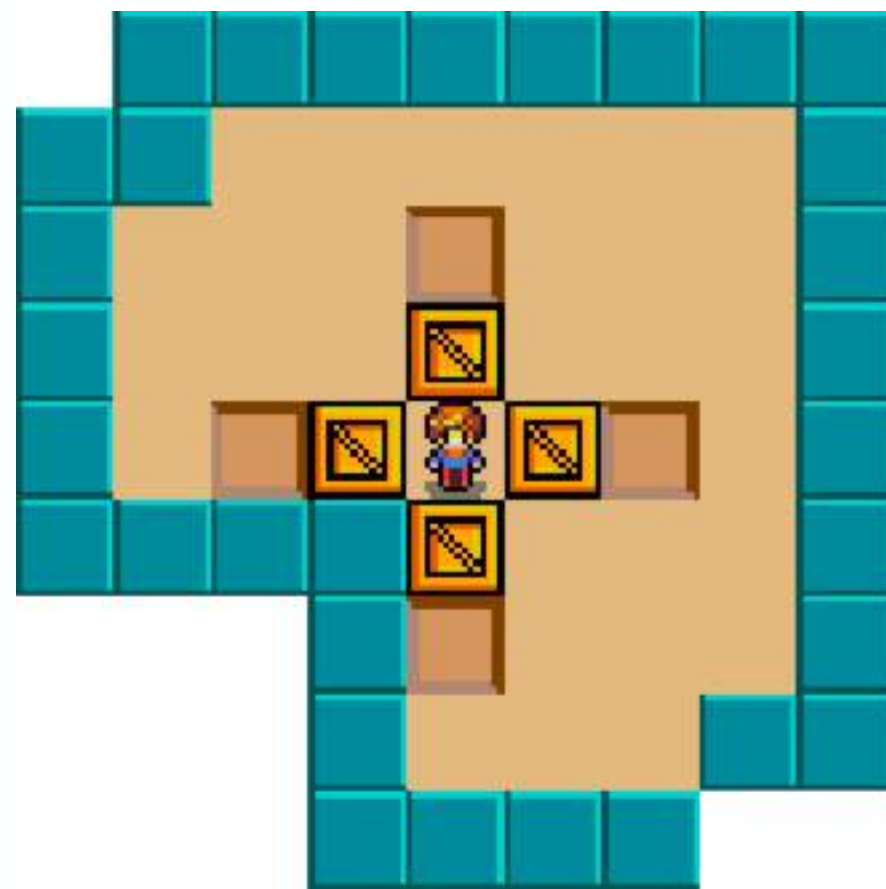
/ImageType      1
/Width          16
/Height         16
/BitsPerComponent 4 % 0-F
/ColorSpace     /DeviceRGB
/ImageMatrix    [1 0 0 -1 0 0]
/DataSource     imageData /ASCIIHexDecode filter
>>
image
```

inverted y coord.

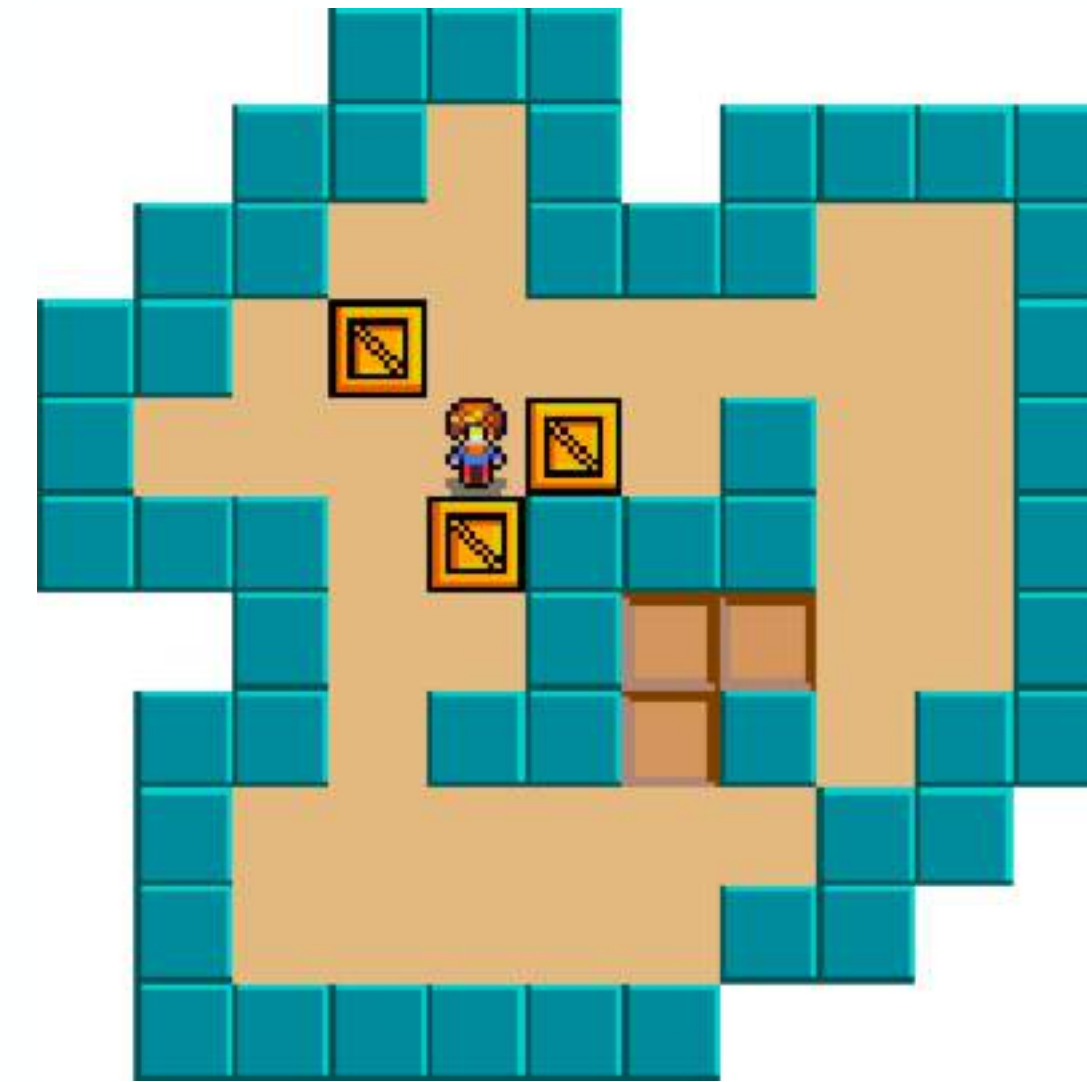
[1 0 0 -1 0 0]

Levels

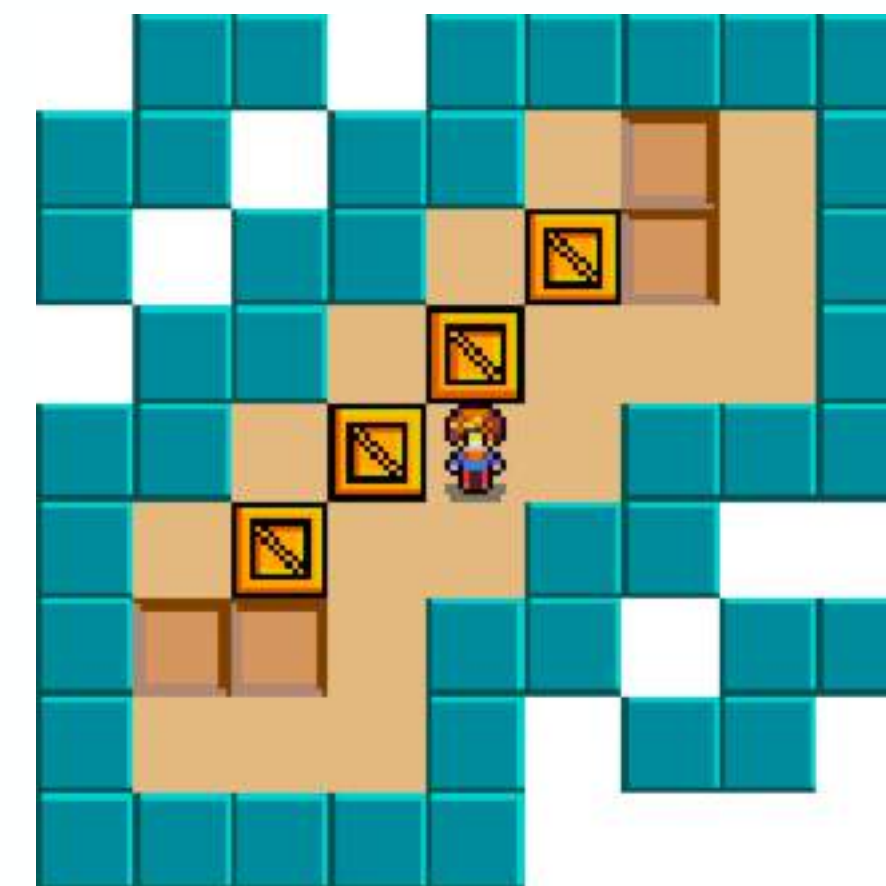
```
% Game symbols:
% (@) Man
% (+) ManOnGoal
% ($) Box
% (*) BoxOnGoal
% (#) Wall
% (.) Goal
% ( ) or (-) Floor
```



```
/l0 (
#####
##-----#
#----.---#
#---$---#
#-.$@$.-#
####$---#
#.----#
#---##
#####) def
```

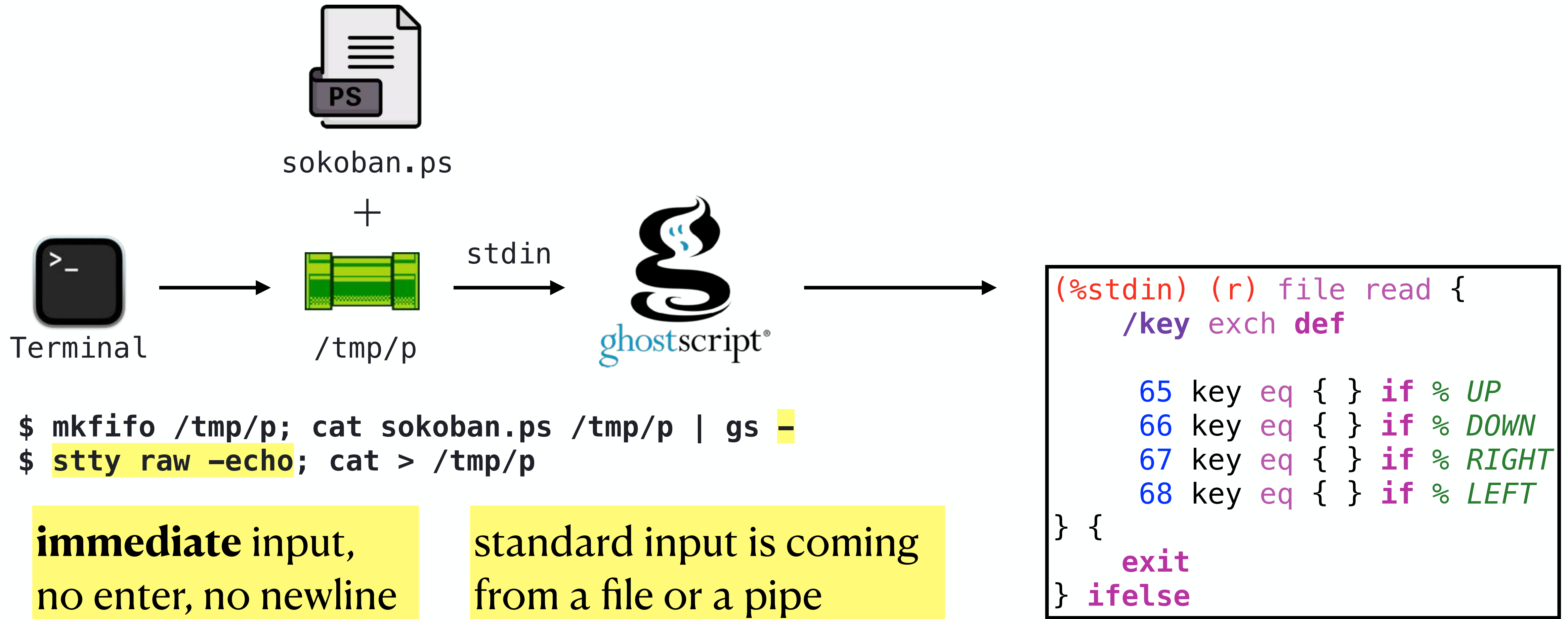


```
/l1 (
###
##-# ####
##--###--#
##-$-----#
#---@$-#--#
###-$###--#
#--#. .--#
##-##.#-##
#-----##
#-----##
#####) def
```

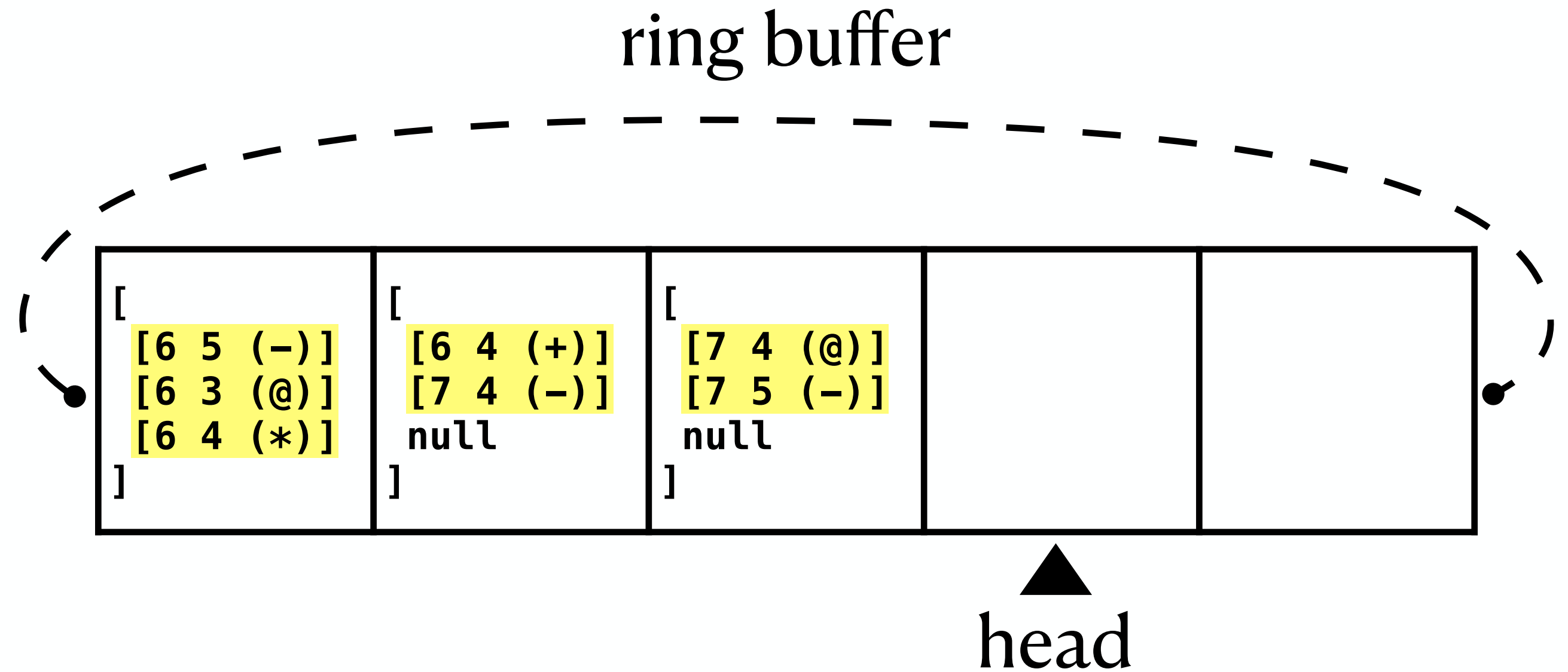


```
/l2 (
## #####
## ##-.-#
# ##-$.-#
##-$---#
##-$@-###
#-$--##
#.-## ##
#---# ##
##### #) def
```

Keystrokes



Undo Manager



```
/umSize 1000 def
/umRingBuffer umSize array def
/umHead 0 def
/umDo {
  umRingBuffer umHead 3 -1 roll put
  /umHead umHead 1 add umSize mod def
} def
/umUndo {
  /umHead umHead 0 eq { umSize 1 sub } { umHead 1 sub } ifelse def
  umRingBuffer umHead get % result
  umRingBuffer umHead null put
} def
```

store contents of cells
about to change

(see OOP attempt)

PSTris

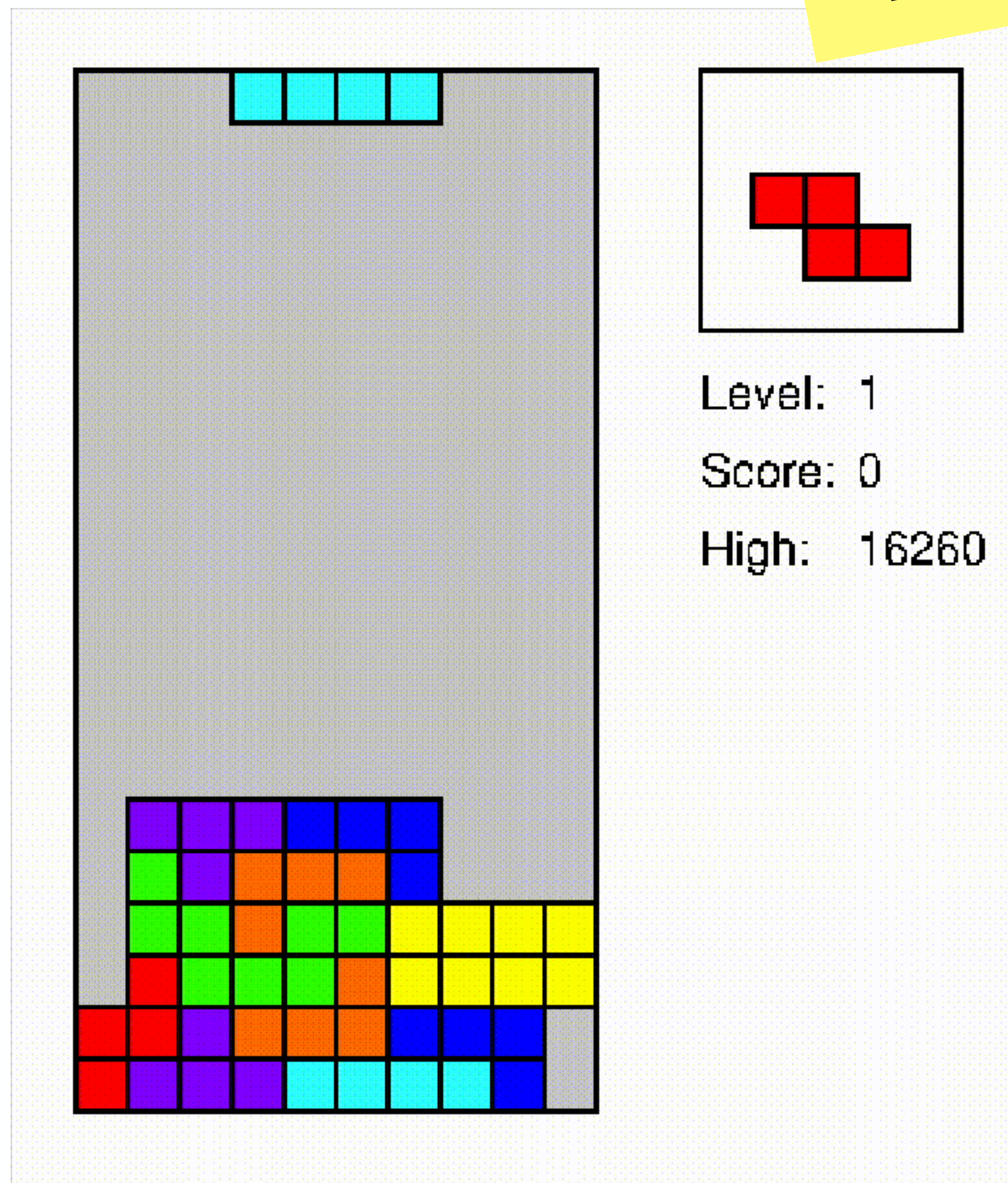
DEMO

“The sheer existence of this creation is both equally impressing and depressing.”

–MBussard45 on Reddit

< 600 lines

< 70 operators

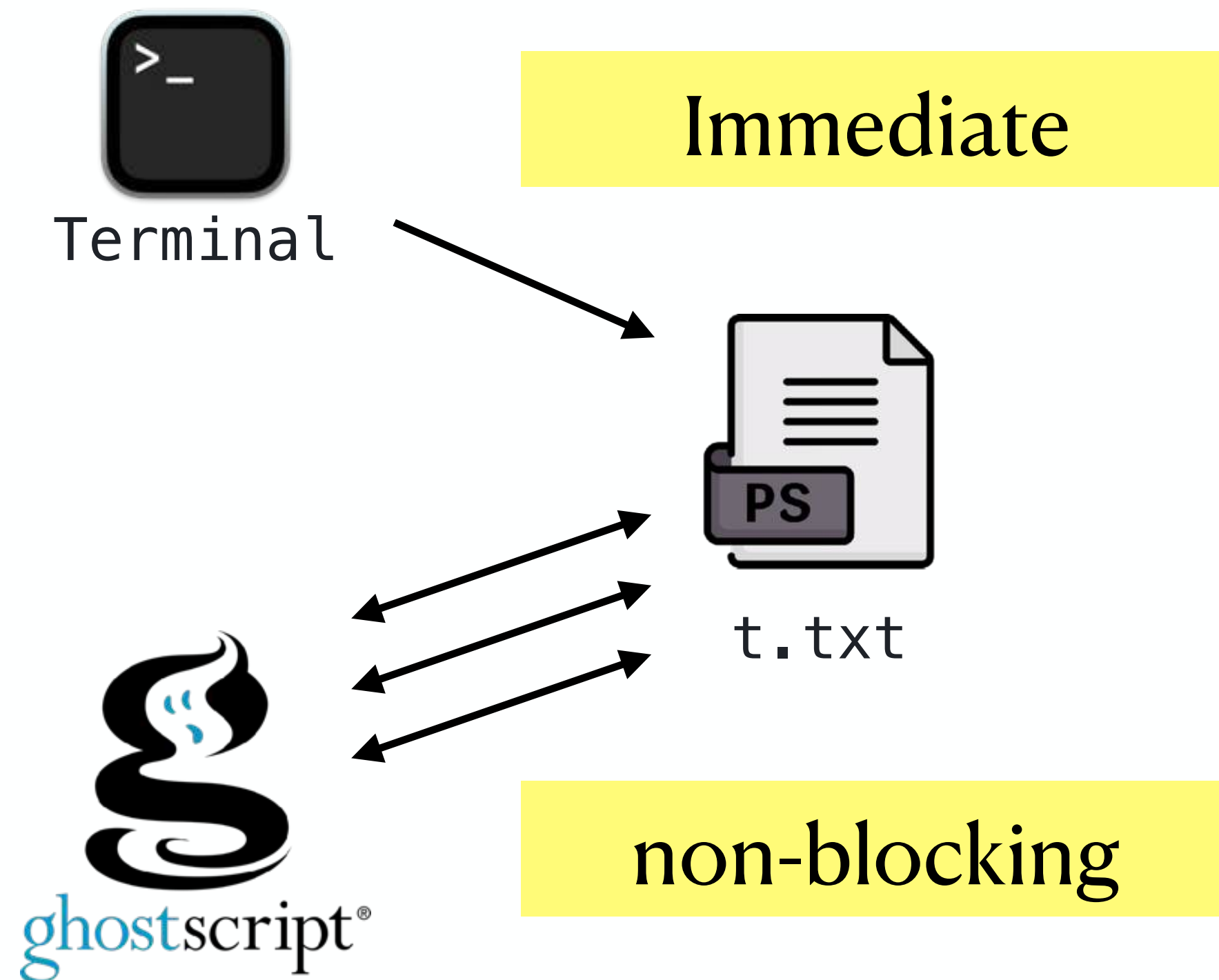


“using raw postscript in this perverse way to get interactivity is truly inspired. My hat is off to the author.”

–hamburglar on Hacker News

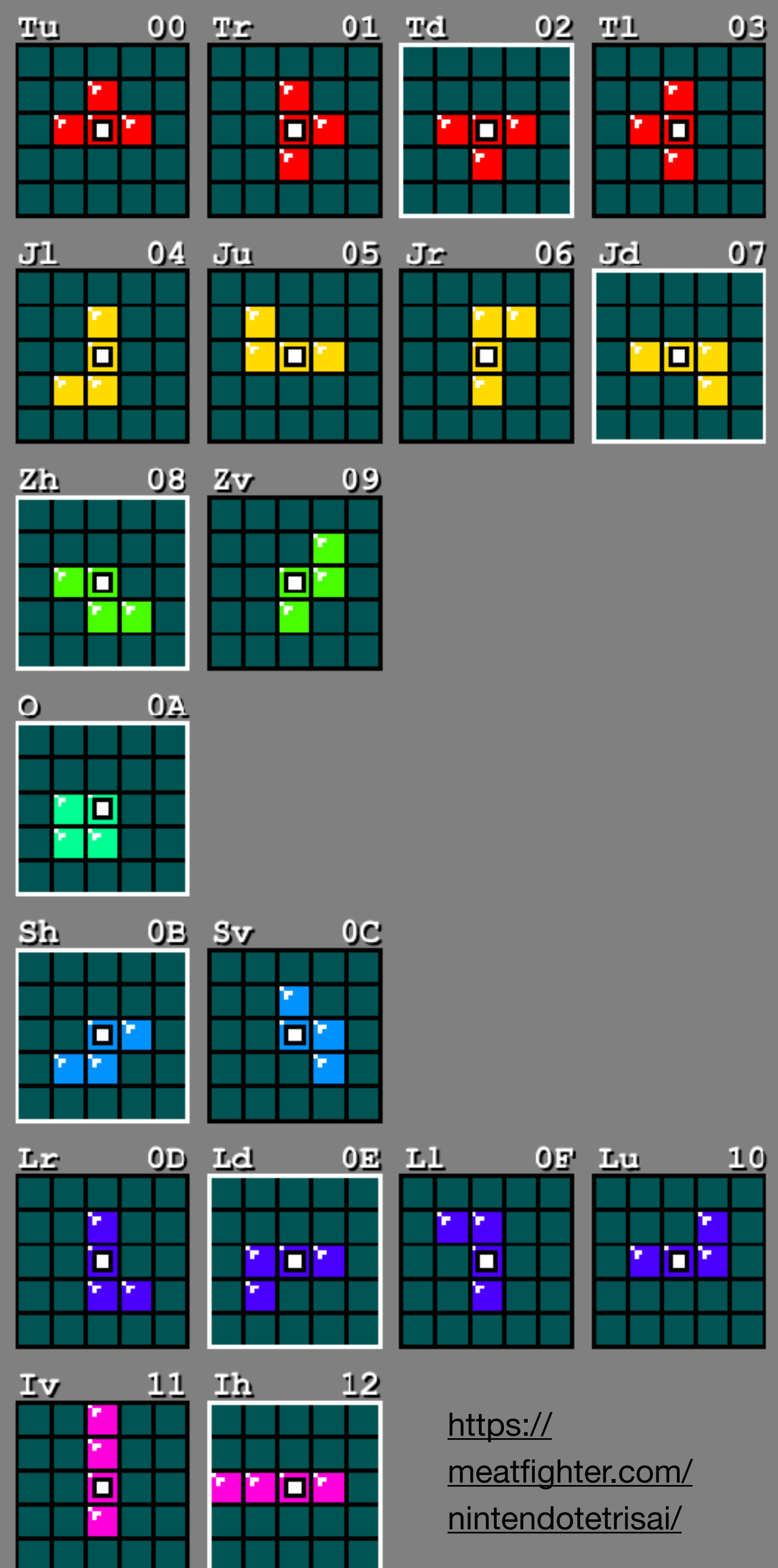
User Input

```
$ stty raw -echo; cat >> t.txt
```



```
$ gs -sNOSAfer tetris.ps
```

```
/ReadInput_s_ {  
  (t.txt) (r) file 8 string readstring pop  
  dup length 0 ne { (t.txt) (w) file closefile } if  
} def  
  
{ % read continuously  
  ReadInput_s_ {  
    /key exch def  
    gameIsOver {  
      key 110 eq { } if % new game  
    } {  
      key 67 eq { } if % right  
      key 68 eq { } if % left  
      key 66 eq { } if % down  
      key 65 eq { } if % rotate  
      key 32 eq { } if % drop  
    } ifelse  
  } forall  
} loop
```



<https://meatfighter.com/nintendotetrisai/>

```

/Tu [0 0 0 0 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0] def
/Tr [0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0] def
/Td [0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 0 0 0 0 0] def
/Tl [0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0] def
/Jl [0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0] def
/Ju [0 0 0 0 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0] def
/Jr [0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0] def
/Jd [0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 0 0 0 0] def
/Zh [0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0] def
/Zv [0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0 0 0 0] def
/O [0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 0 0 0 0 0] def
/Sh [0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1 0 0 0 0 0] def
/Sv [0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0] def
/Lr [0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0] def
/Ld [0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0 0 0 0] def
/Ll [0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0] def
/Lu [0 0 0 0 0 0 0 0 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0] def
/Iv [0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0] def
/Ih [0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0] def

```

```

/Tetriminos [
  [Td Tl Tu Tr]
  [Jd Jl Ju Jr]
  [Zh Zv]
  [O]
  [Sh Sv]
  [Ld Ll Lu Lr]
  [Ih Iv]
] bind def

```

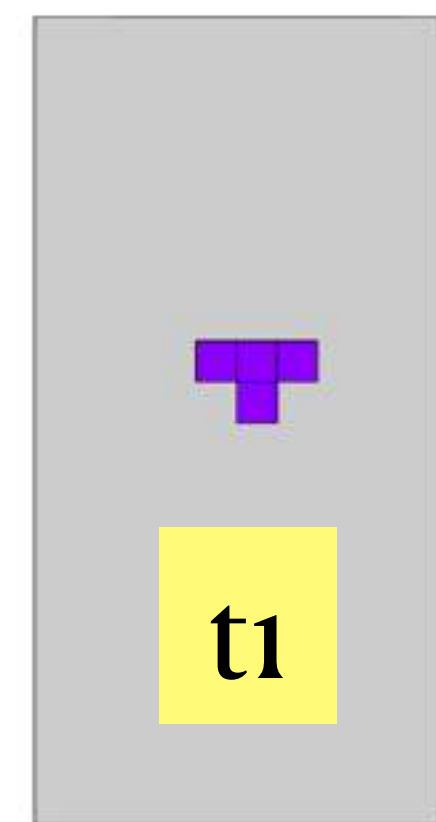
Memory Storage

```

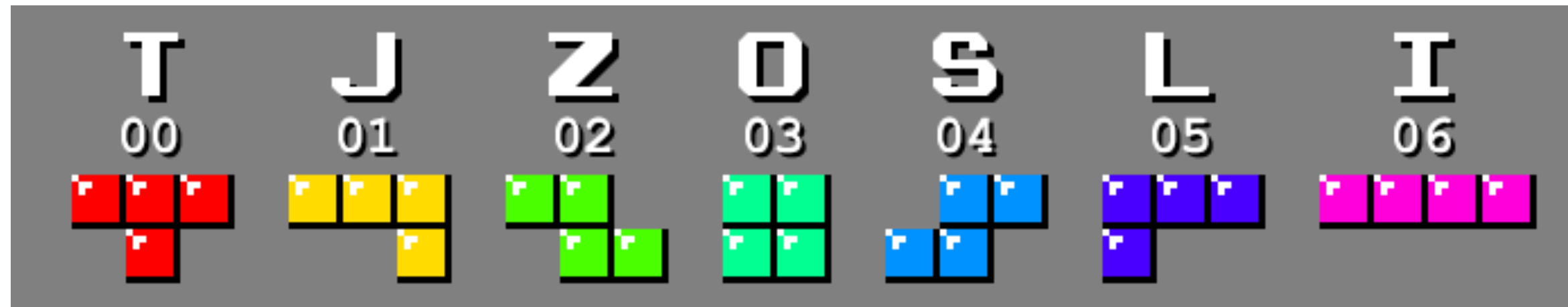
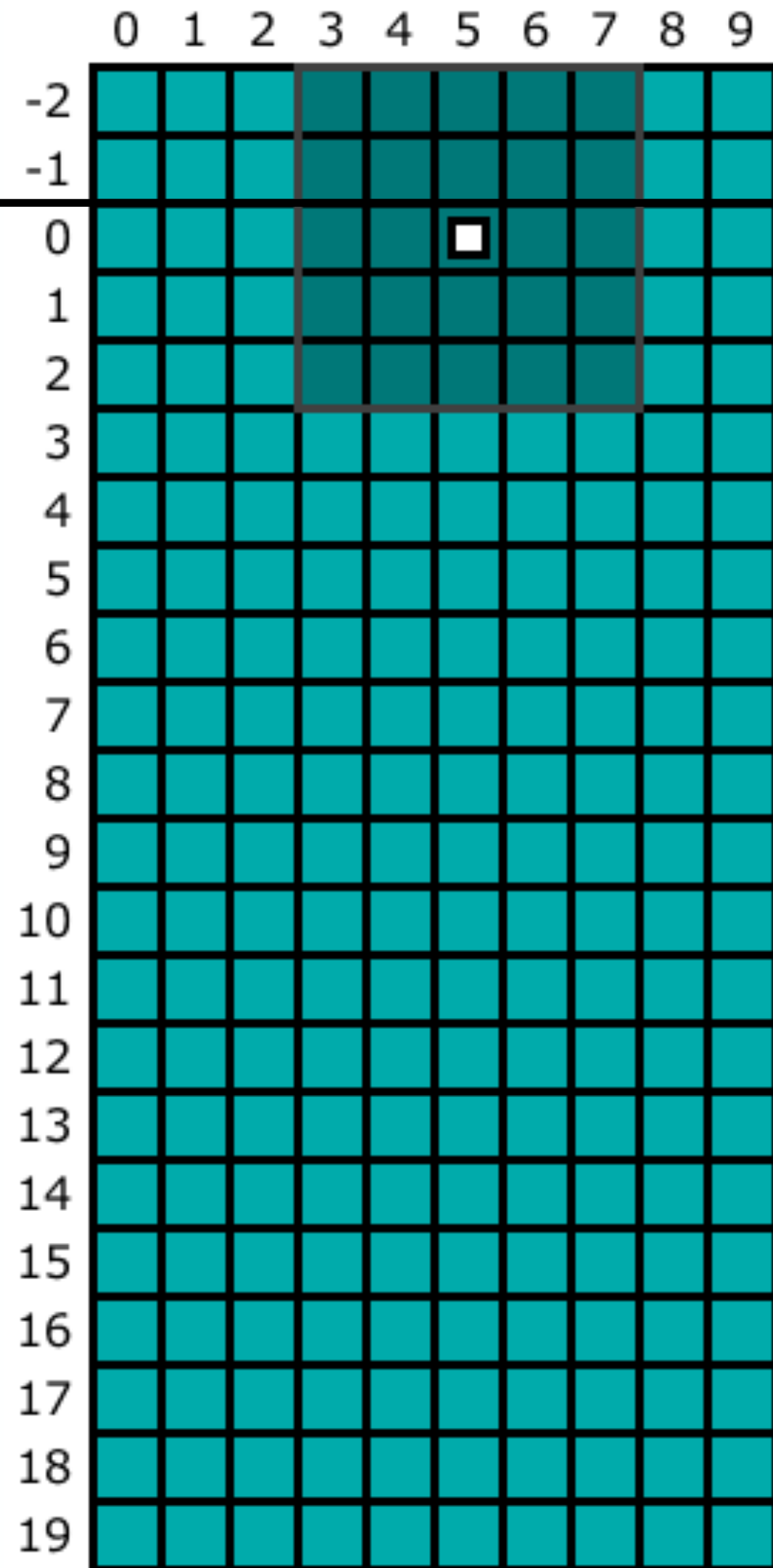
/t1c 0 def % code
/t1r 0 def % rotation

/t2c 2 def % code
/t2r 0 def % rotation

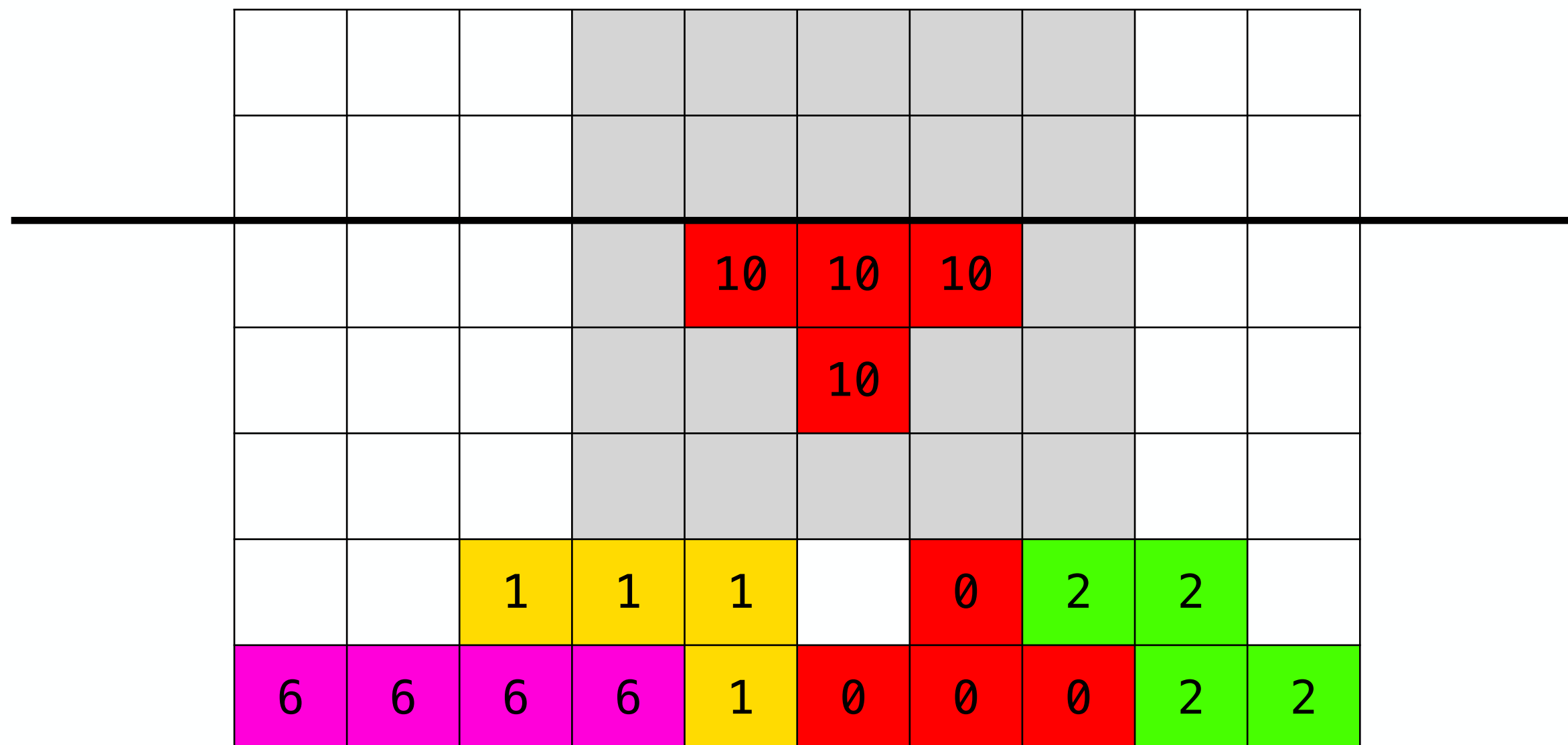
```



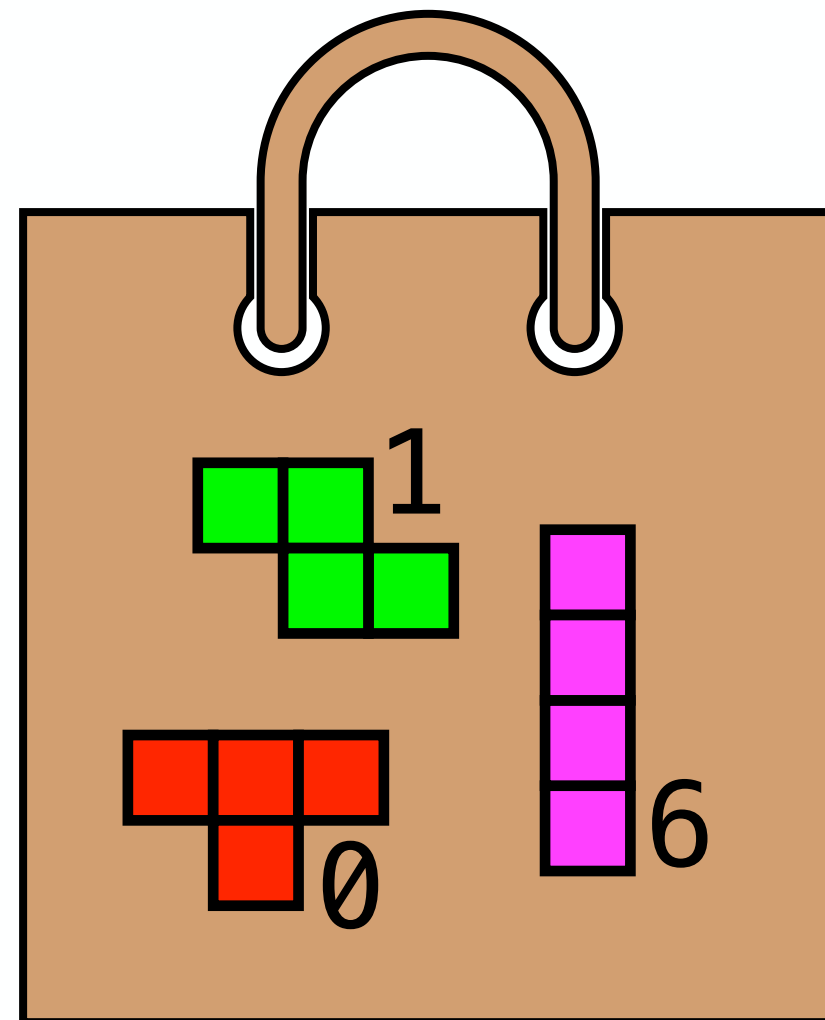
Level: 1
Score: 0
High: 25580



+10 for active tetrimino



Random Bag



T 00	J 01	Z 02	O 03	S 04	L 05	I 06
	✓		✓	✓	✓	

```
/BAG_SIZE 7 def
/randomBag BAG_SIZE array def
```

conciseness

```
/NewTetrimino_i_ {
    % pick random index with null value, set value to true
    {
        /i rand BAG_SIZE mod def
        randomBag i get null eq { randomBag i true put exit } if
    } loop
    i % leave this index on stack

    % if array is filled with true, create a new empty array
    /isFull true def
    randomBag { null eq { /isFull false def exit } if } forall
    isFull { /randomBag BAG_SIZE array def } if
} def
```

Levels, Speed and Scoring

```
hasFullRows {  
  % ...  
  
  % count lines cleared  
  /linesCleared linesCleared n add def  
  
  % next level every 10 lines  
  /level linesCleared 10 idiv 1 add def  
  
  % increase speed as level increase  
  /delay 300 level 1 sub 30 mul sub def  
  
  % Nintendo scoring system  
  /score [40 100 300 1200] n 1 sub get level mul score add def  
}
```

concise

easy

LEVEL	0	1	2	3	4	5	6	7	8	9
Single	40	80	120	160	200	240	280	320	360	400
Double	100	200	300	400	500	600	700	800	900	1000
Triple	300	600	900	1200	1500	1800	2100	2400	2700	3000
TETRIS	1200	2400	3600	4800	6000	7200	8400	9600	10800	12000

* This table applies to both game A and B.

* The score increases in the same way at LEVEL 10 and higher.

High Scores

Save arbitrary files, on printers as well, persisted across sessions.

“You’re not supposed to do that”

```
/ListFiles {  
  (*) { == } 256 string filenameforall  
} def
```

```
/_s_fn_WriteFile {  
  /f exch (w) file def  
  f exch writestring  
  f flushfile  
  f closefile  
} def
```

```
/_fn_ReadFile_s_ {  
  (r) file  
  256 string  
  readstring  
  pop  
} def
```

```
PS>ListFiles
```

```
PS>(DevDays) (x.txt) _s_fn_WriteFile
```

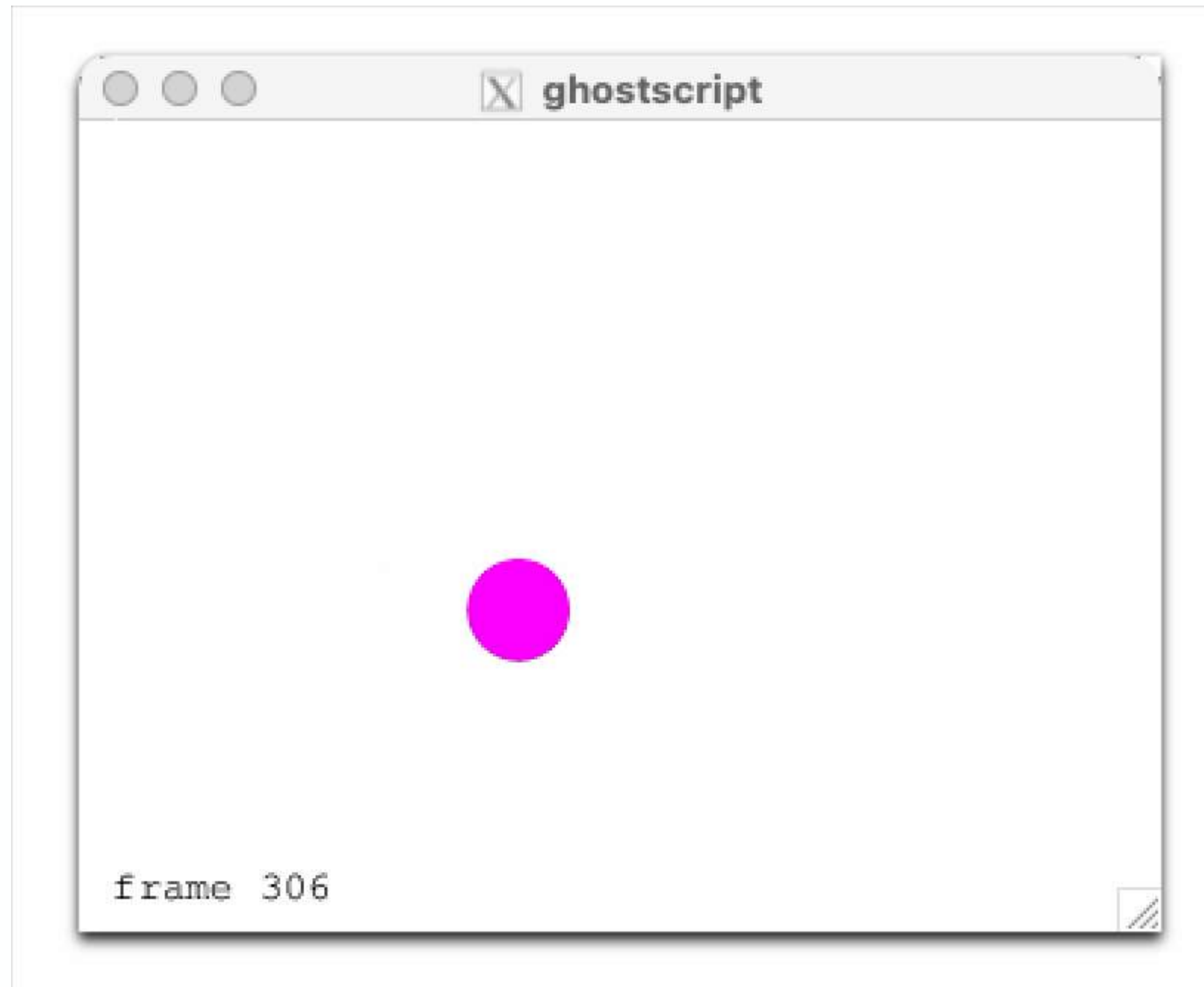
```
PS>ListFiles  
(x.txt)
```

```
PS>(x.txt) _fn_ReadFile_s_ ==  
(DevDays)
```

High scores, full games and state secrets alike.



Animations



```
/W 320 def /H 240 def /R 15
/x 100 def /y 50 def def /x_ 2 def /y_ 1 def
<< /PageSize [W H] >> setpagedevice

{
  % Clear
  1 setgray 0 0 320 240 rectfill

  % Draw
  x R sub 0 lt x R add W gt or {
    /x_ x_ -1 mul def
  } if
  y R sub 0 lt y R add H gt or {
    /y_ y_ -1 mul def
  } if
  /x x x_ add def /y y y_ add def
  newpath x y R 0 360 arc 1 0 1 setrgbcolor fill
  10 10 moveto 0 setgray
  flushpage

  % Sleep
  1000000 {} repeat
} loop
```

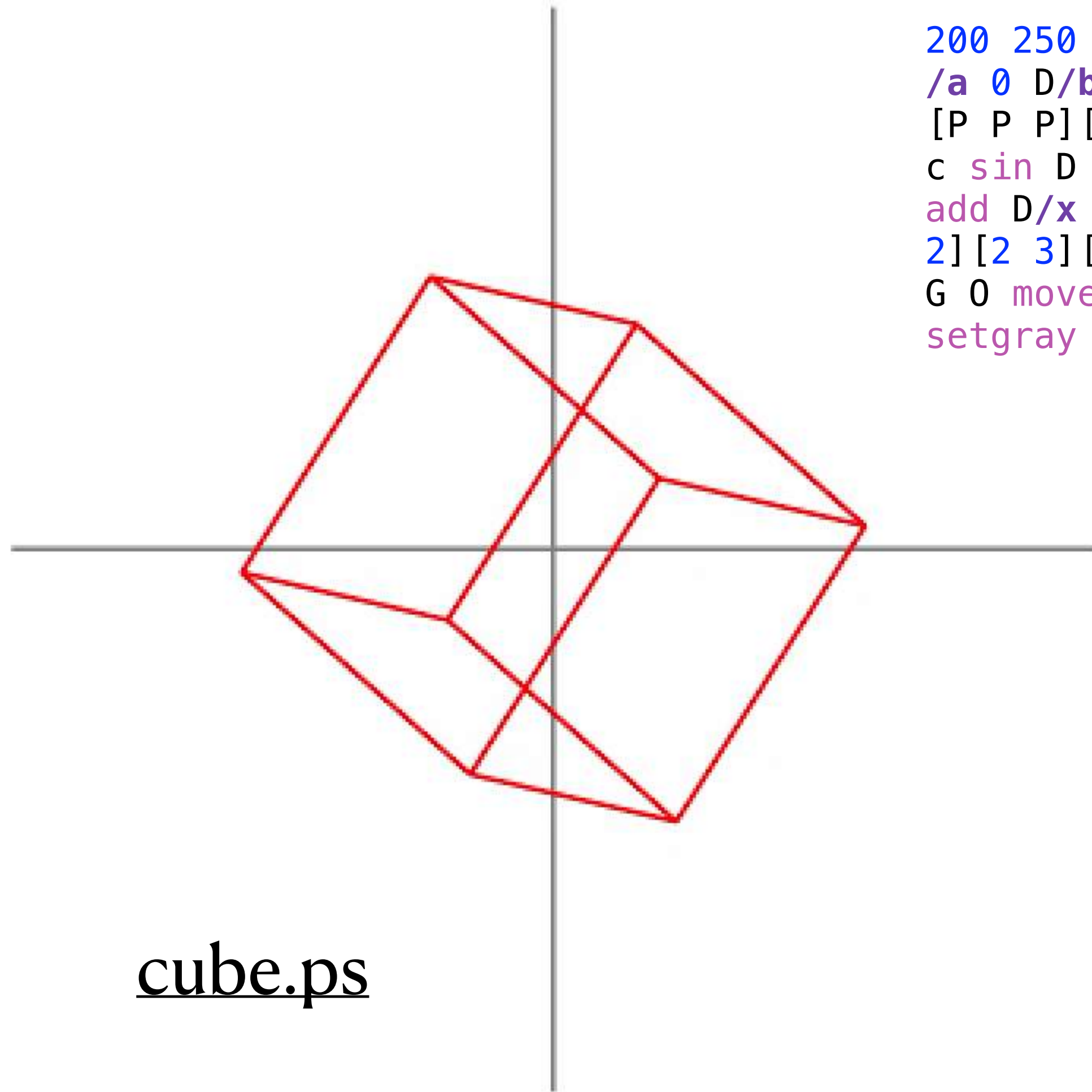
Clear

Draw

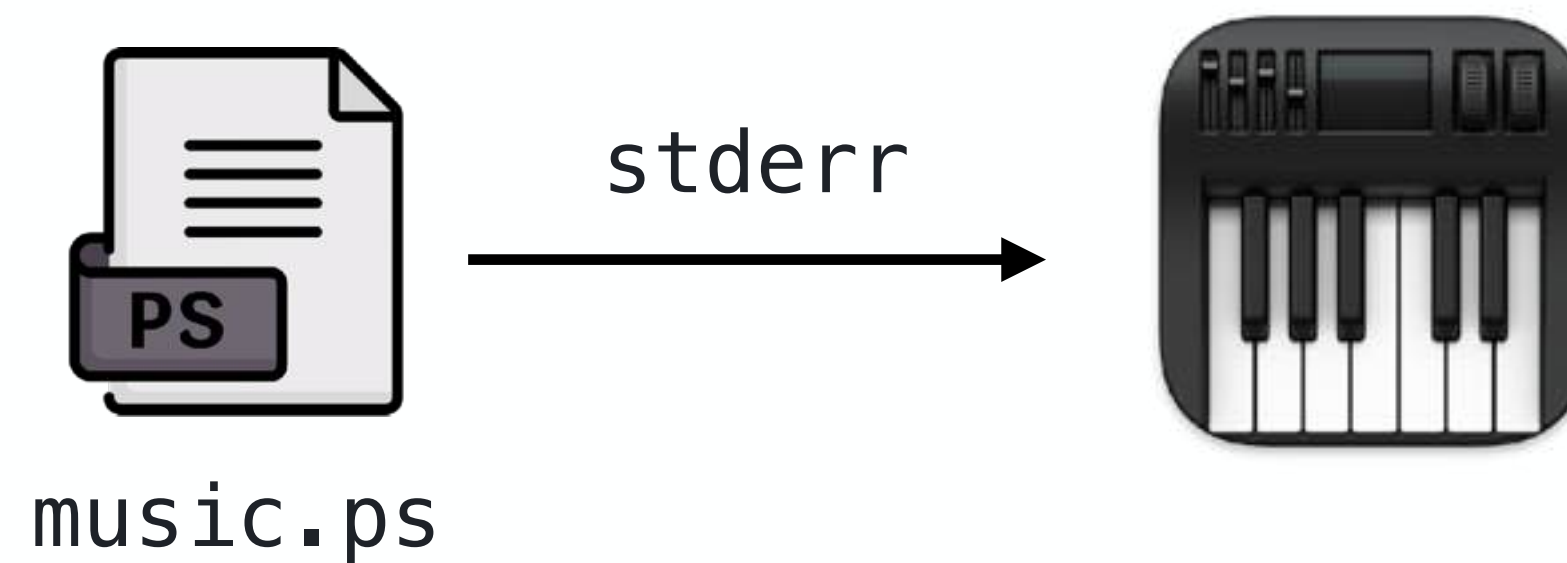
Sleep

3D and Music

```
200 250 translate/D{def}def/M{mul}D/Z{2000000}{repeat}D/P 50 D/N{P neg}D/G{get}D
/a 0 D/b 0 D/c 0 D/o{0 G}D/O{1 G}D{/v[[N N N][P N N][P P N][N P N][N N P][P N P]
[P P P][N P P]]D/w 8 array D/f a cos D/d b cos D/e c cos D/s a sin D/t b sin D/u
c sin D 0 1 7{/i exch D/p v i G D/y p 0 f M p 2 G s M sub D/z p 0 s M p 2 G f M
add D/x p o d M z t M add D w i[x e M y u M sub x u M y e M add]put}for[[0 1][1
2][2 3][3 0][4 5][5 6][6 7][7 4][0 4][1 5][2 6][3 7]]{/E exch D w E o G o w E o
G 0 moveto w E 0 G o w E 0 G 0 lineto}forall 1 0 0 setrgbcolor stroke flushpage Z 1
setgray -100 dup 200 dup rectfill/a a 0.1 add D/b b 0.3 sub D/c c 0.5 sub D}loop
```



Send MIDI messages to
MIDI synthesizer





1

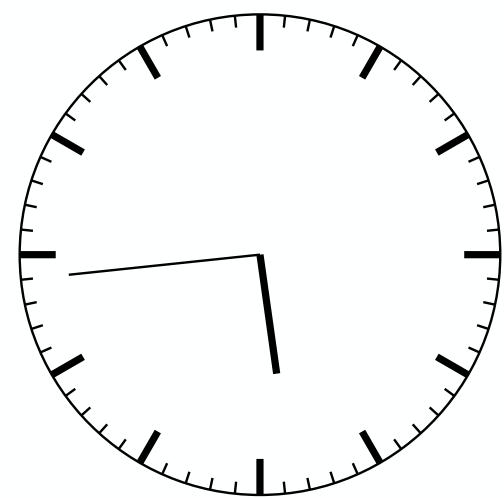
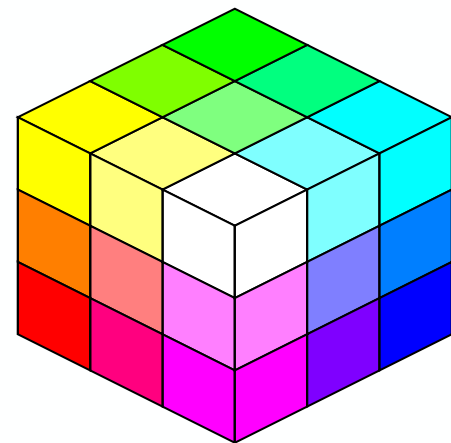
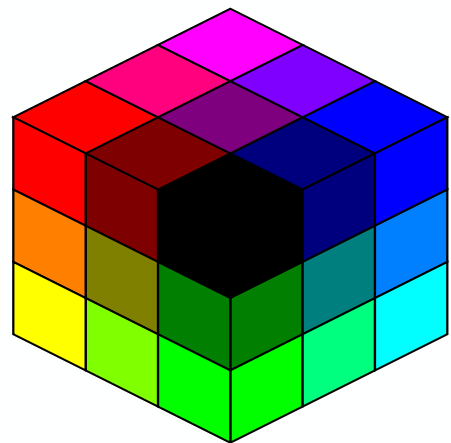
Program in PostScript?

add
2
1

Operands

userdict
globaldict
systemdict

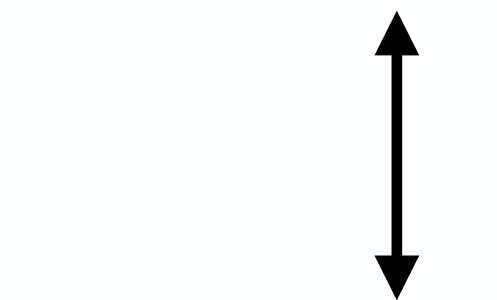
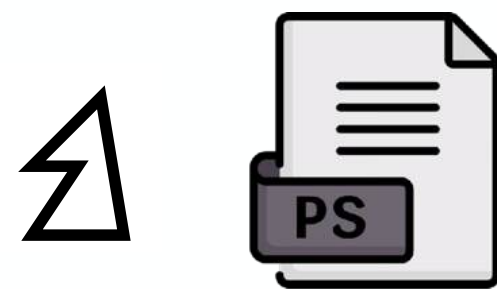
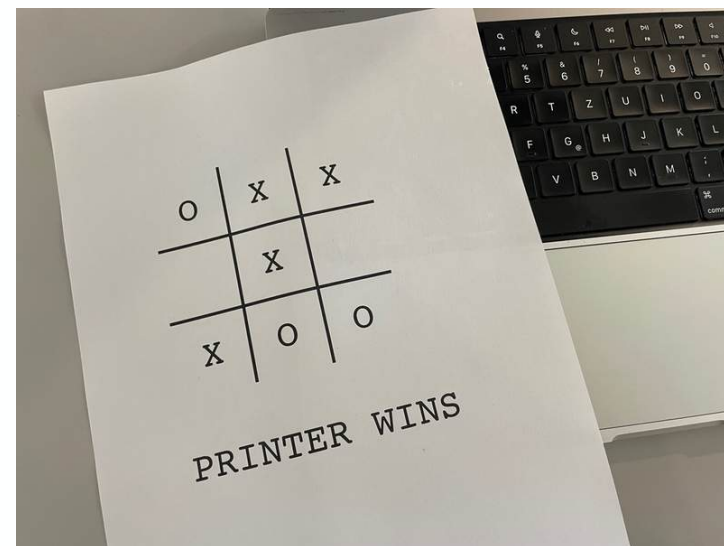
Dictionaries



17:44

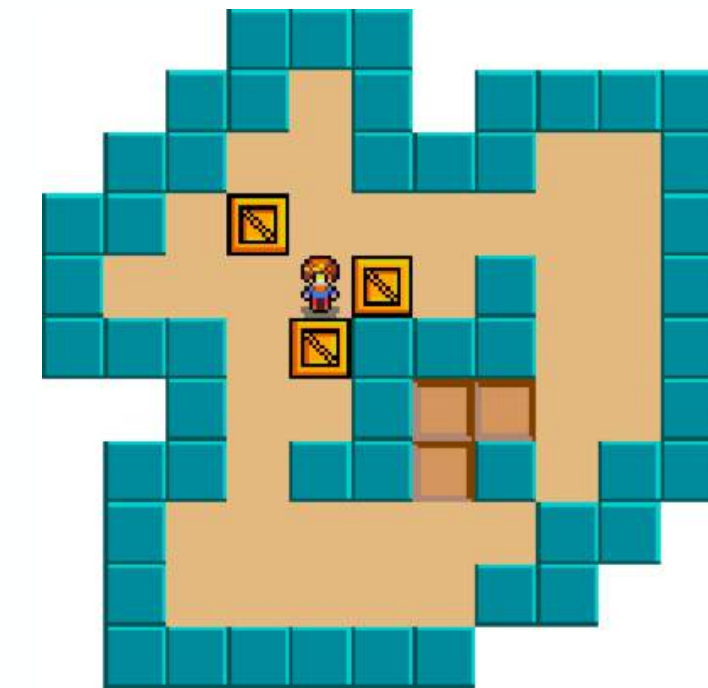
2

Play against a printer?



3

Games for desktop?



```

/l1 (
  ###
  ##-# ####
  ##-###-#
  ##-$-----#
  #---@$-##-#
  ###-$###-#
  #--#. .-#
  ##-##.#-##
  #-----#
  #-----#
  #####) def

```

				10	10	10			
					10				
		1	1	1		0	2	2	
6	6	6	6	1	0	0	0	2	2

Take Aways

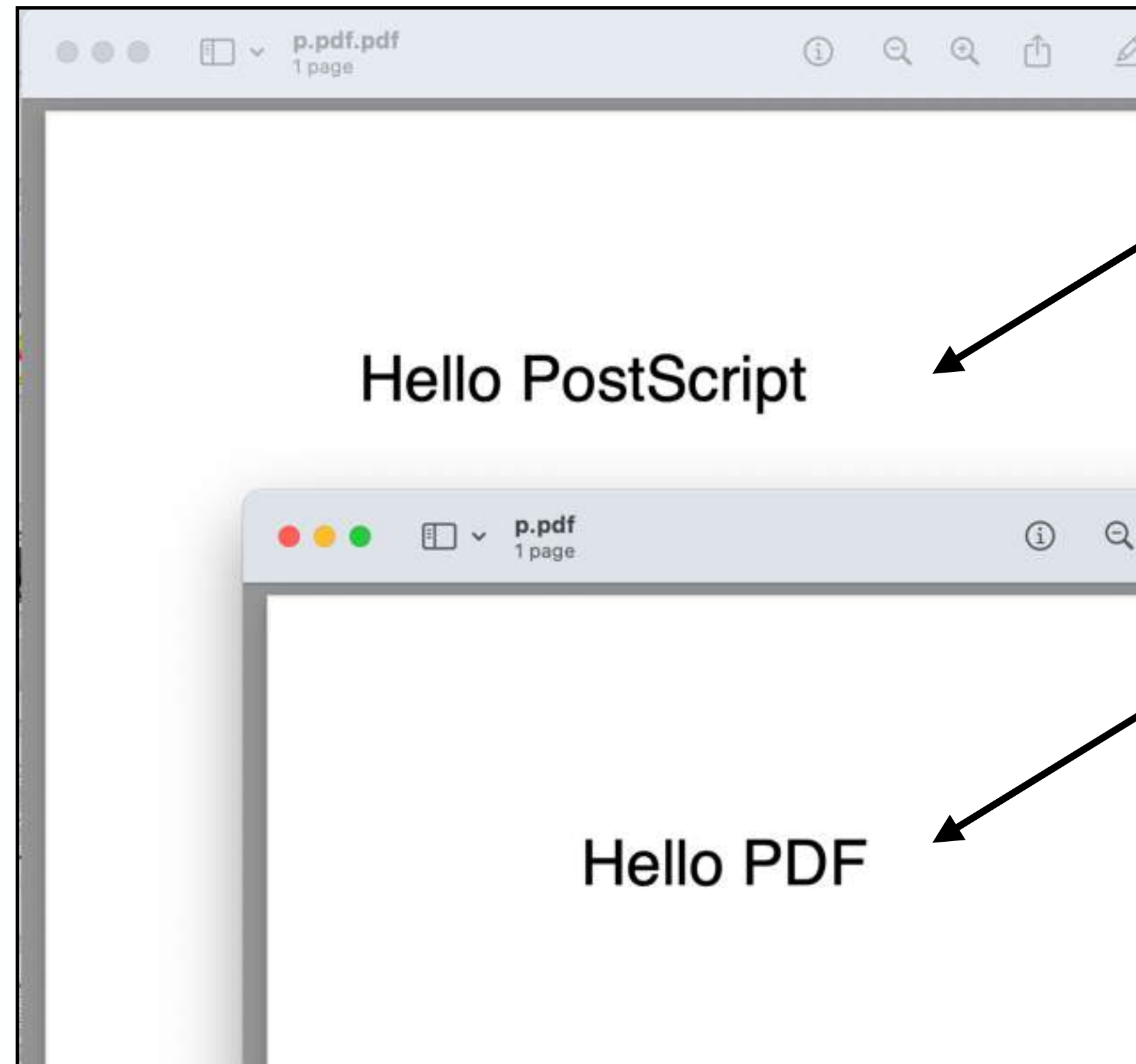
- You won't overlook **printers** anymore
- Supposedly dumb tools may be **cool stuff**
- Constraints foster **creativity**
- **Building new things** with unexpected tools is great fun
- **Hacking** is not just about security, it's also about **fun and art**



Nicolas Seriot

Slides and code on seriot.ch

More Cool Stuff



```

%!PS
%PDF-1.4

```

```

/Helvetica findfont 20 scalefont setfont
100 700 moveto (Hello PostScript) show showpage stop

1 0 obj << /Type /Catalog /Pages 2 0 R >> endobj
2 0 obj << /Type /Pages /Kids [3 0 R] /Count 1 >> endobj
3 0 obj <<
  /Type /Page
  /Parent 2 0 R
  /MediaBox [0 0 612 792]
  /Resources << /Font << /F1 5 0 R >> >>
  /Contents 4 0 R
>> endobj
4 0 obj << /Length 41 >> stream
BT /F1 20 Tf 100 700 Td (Hello PDF) Tj ET
endstream endobj
5 0 obj <<
  /Type /Font
  /Subtype /Type1
  /BaseFont /Helvetica
>> endobj

```

```

xref
0 6
0000000000 65535 f
0000000110 00000 n
0000000159 00000 n
0000000216 00000 n
0000000352 00000 n
0000000443 00000 n
trailer << /Size 6 /Root 1 0 R >>
startxref
523

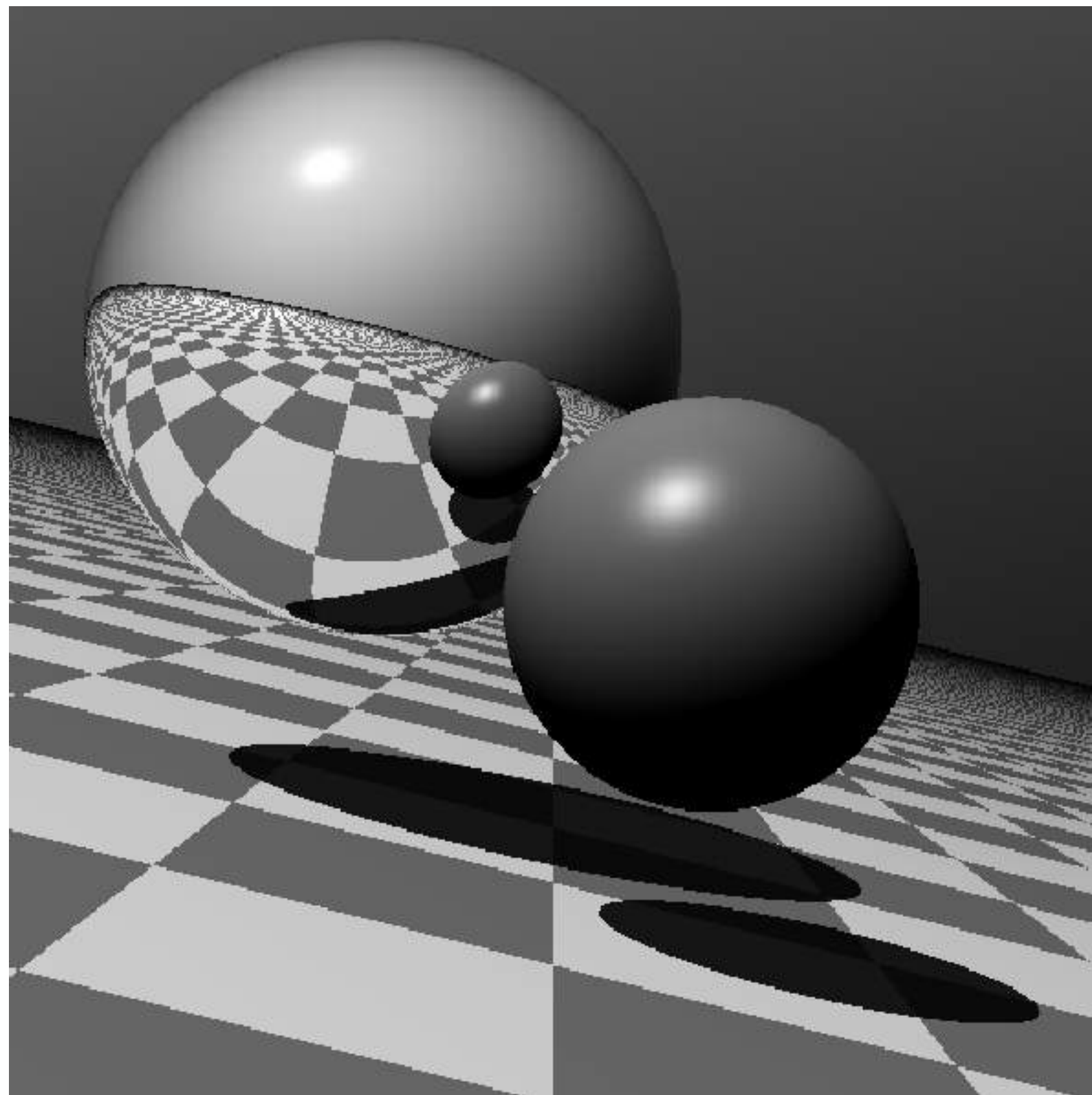
```

A polyglot file



Tiny Ray Tracer

Obfuscated PostScript Contest 1993



from 760 down to 727 bytes

```
/p/floor/S/add/A/copy/n/exch/i/index/J/otherwise  
/H{count 2 idiv exch repeat}def  
/q/gt/h/exp/t/and/C/neg/T/dup/Y/pop/d/mul/w/d
```

Define operator aliases

```
/c(j1idj2id42rd)/G(140N7)/Q(31C85d4)/B(V0R0VR100P) #(#0#) #(#0577 17)  
300 T translate  
/I(3STinTinTinY)/L(993dC99Cc96raN)/k(X&E9!&1!  
/O(Y43d9rE3IaN96r63rvx2dcaN)/z(&93r6IQ02Z4o3A  
/W 270 def  
/L(1i2A00053r45hNvQXz&vUX&U0vQXzFJ!FJ!J)/D(cjS5o32rS4oS3o)/v(6A)/b(7o)  
/F(&vGYx4oGbxSd0nq&3IGbxSGY4Ixwca3AlvvUkbQkdbGYx4ofwnw!&vLx2w13wSb8Z4wS!J!)  
/X(4I3Ax52r8Ia3A3Ax65rTdCS4iw5o5IxnwTTd32rCST0q&eCST0q&D1!&EYE0!J!&EY0!J0q)  
/V 3 def  
/x(jd5o32rd4odSS)/a(1CD)/E(YYY)/o(1r)/f(nY9wn7wpSps1t1S)
```

Encode procedures into strings

```
{  
  [  
    n {  
      ( )T 0 4 3 r put  
      T(/)q {  
        T(9)q {  
          cvn  
        } {  
          s  
        } J  
      } {  
        ($)q {  
          [  
          ]  
        } J  
      } J  
      cvx  
    } forall  
  ]  
  cvx def  
} H
```

Extract procedures out of encoded strings

1 2 (add) % literal
cvx % executable
exec % 3

```
K {  
  K {  
    L setgray moveto B fill  
  } for Y  
} for
```

Main call