

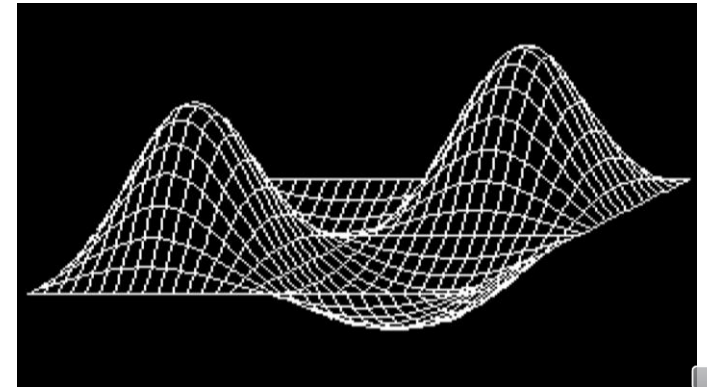
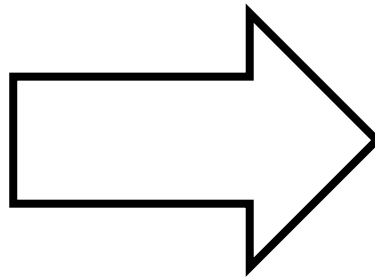
**FOSDEM**

# Scan2Run

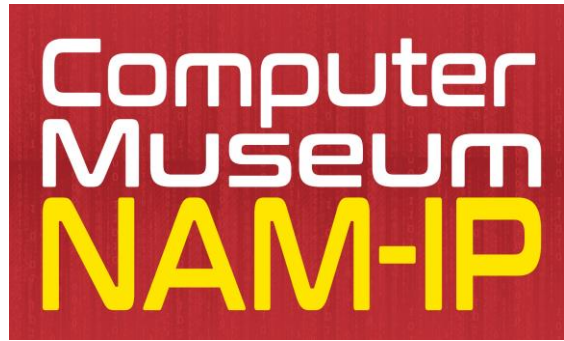
## Reviving old listings in MAME emulator

PONSARD Christophe – FOSDEM 21 (online)

```
1  REM MAASGRAFIKEN - TRANSCRIPTIE VAN ANONIEM APPLE II-PROG.  
2  REM C.W.A. van Dijk - Kampen  
3  REM  
10  MODE 0:GOTO 800  
20  F=SIN(X)*COS(Y)  
30  IF F>Z2 THEN F=Z2  
40  IF F<Z1 THEN F=Z1  
50  RETURN  
100  Y=W0: X=U0:GOSUB 20  
110  W4=S2*(W0-W1): X=S4*(U0-U1)-C1*W4+R1: Y=S3*(F-Z1)-S1*W4+R2:  
    W=1.0: IF P8=0.0 THEN RETURN  
120  W6=Y: U6=X: D=1.0: Y0=-S1*W4+R2-1.0: Y9=S6: IF X>S9 THEN  
    Y9=S5*X+S7  
130  IF Y0<Y9 THEN RETURN  
140  V2=(Y0-Y9)/EA: FOR K=0.0 TO EA: Y8=Y0-K*V2  
150  Y=(R2-Y8)/S1: X=(U6+C1*Y-R1)/S4+U1: Y=Y/S2+W1:GOSUB 20:  
    F=(F-Z1)*S3+Y8: IF F<W6 THEN W=E0  
160  IF F>W6 THEN D=E0  
170  IF W=D=E0 THEN 200  
180  NEXT K  
190  W=1.0  
200  Y=W6: X=U6: RETURN  
210  GOSUB 100: IF W=V9 THEN 330  
220  IF T0=E2 THEN 280  
230  W9=W0: W8=W9-W3  
240  IF S2*(W9-W8)<E2 THEN 320  
250  W0=(W8+W9)/E2:GOSUB 100: IF W<V9 THEN W9=W0  
260  IF W=V9 THEN W8=W0  
270  GOTO 240  
280  U9=U0: U8=U9-U3  
290  U0=(U8+U9)/E2:GOSUB 100: IF W<V9 THEN U9=U0  
300  IF W=V9 THEN U8=U0  
310  IF S4*(U9-U8)>E2 THEN 290  
320  W=V9: V9=1.0-W
```



# Context – NAM-IP Computer Museum



[www.nam-ip.be](http://www.nam-ip.be)

- Located in Namur/Belgium - 30' from Brussels  
(worth a visit next FOSDEM when back at ULB)
- Missions:
  - Preservation: safeguarding digital heritage, focus on local pioneers
  - Acquisition of artefacts, enriching collections
  - Exhibition: for all, specific animation, permanent/temporary
  - Research: about machines, software, communities
- “Container design”, an historical parallel



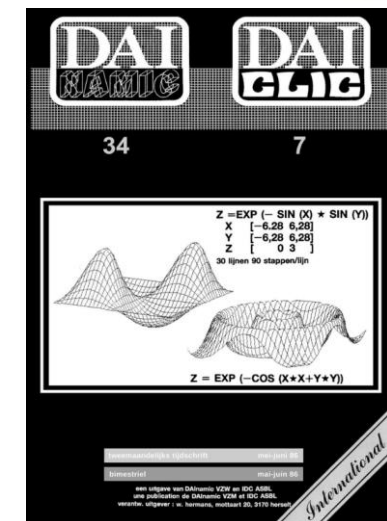
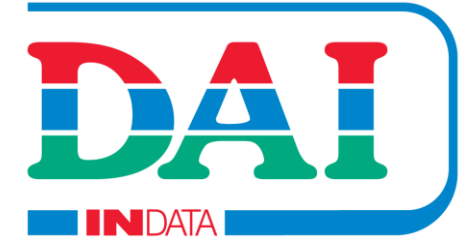
# Digital Preservation Constraints

- Machine in preservation mode:
  - Hard to use, not always multiple machines available, e.g. inDATA DAI rare Belgian computer
  - ➔ benefit of using an emulator, e.g.



- Variety of (native) supports:
  - old floppies (8, 5 ¼ inch ...), magazines with listing,...
  - ➔ focus here on listings

- Bottom line: SCAN listing to RUN in emulator



# SCAN STEP – Listing Specificities

- listing “fonts” (e.g. dot matrix printer)
  - usually low res, not very good quality scan
  - aging issues
- listing not text !
  - statistical models based on corpus not useful
  - even degrading e.g. variable mixing letter/number  
→ numbers tend to become letters 1/I, 0/O
  - Initial numbers can be identified as a column
  - could train a “listing” corpus (e.g. keywords) possibly with some parsing rules
- reuse:
  - same printing style across pages of long listing
  - in multiple numbers of the same magazine,...

F O S D E M

```
10      MODE 0:GOTO 800
20      F=SIN(X)*COS(Y)
30      IF F>Z2 THEN F=Z2
40      IF F<Z1 THEN F=Z1
50      RETURN
```



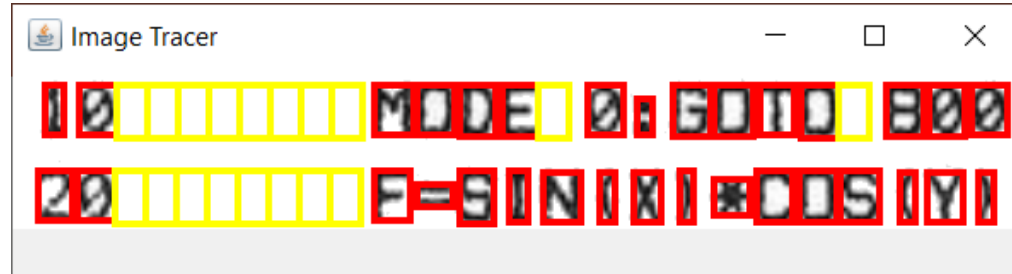
# Scanning Listings – Selected Approach

- Open Source solution
- Avoid fully automated approach
  - User in loop but try to involve it only when necessary  
unknown character, ambiguities, possible artefact...
  - Expected decreasing interactions “learning” phase → “confirmation” phase
- Globally KISS approach – Keep It Simple Stupid (for now)
  - Avoid big black box not easy to control
    - Tested but not very successful with Tesseract OCR
  - Simple understandable code base
    - Character tracer → recognition
    - Take control over a “learning” loop
    - JavaOCR (already used in a previous project)
  - Potential for improving later, e.g. feeding into neural network etc.

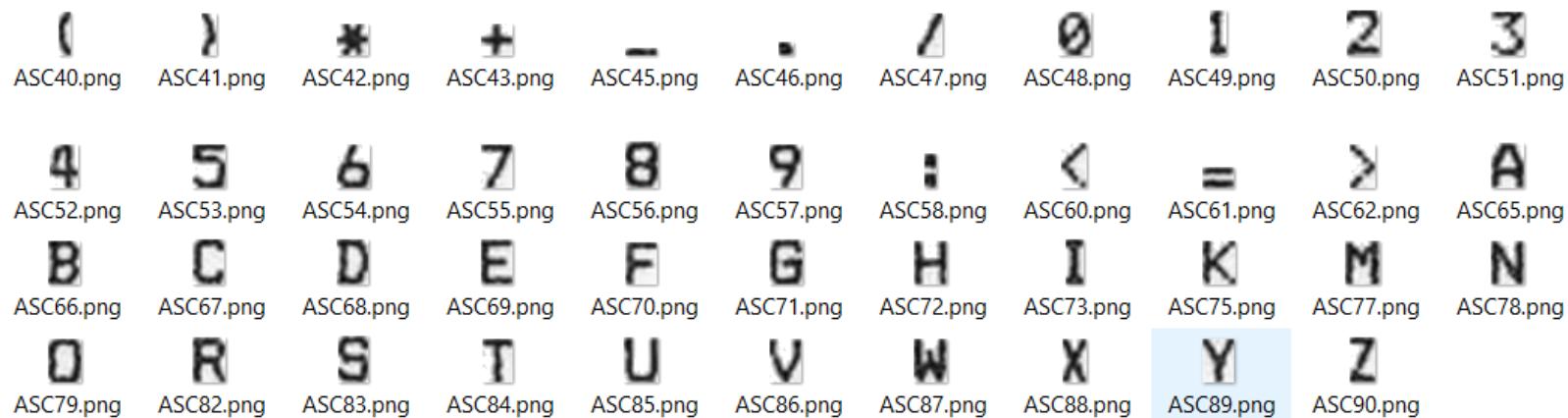


# Logical Steps in original JavaOCR

- Character and sequencing: document → rows → characters



- Character is compared with training set and best match is selected (Mean Square Error used)



- NOT efficient: need to manually collect characters BEFORE starting scan  
→ MORE DYNAMIC APPROACH



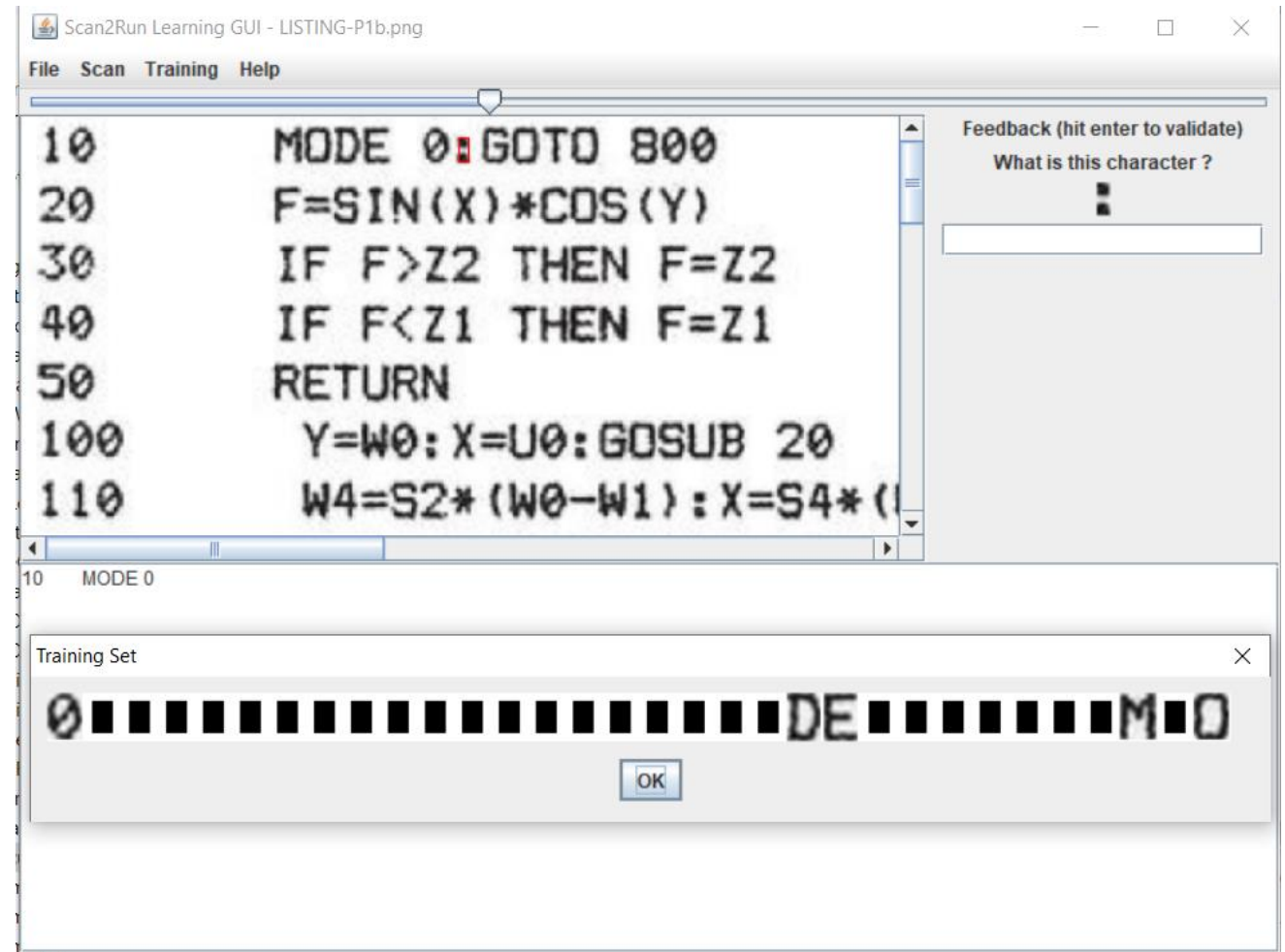


# Scan2Run UI: build training set dynamically (if does not exist or reuse/improve previous)

- Learning phase  
(few know character)  
→ high match required  
(to avoid matching 8-S, 0/O etc)

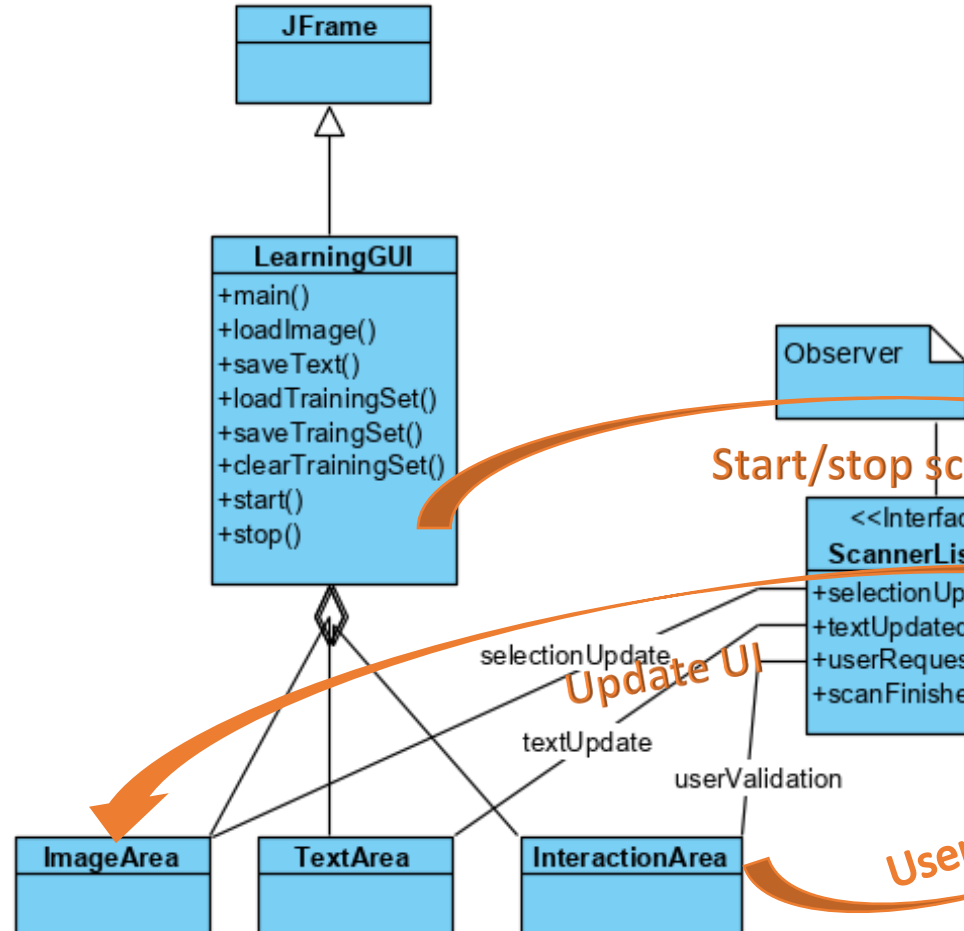
- Then lower MSE threshold

```
public double computeAskThreshold() {  
    int s=trainingImages.keySet().size();  
    if (s>20) return askThreshold;  
    return askThreshold-0.025*s;  
}  
  
• Also ask if second match  
  is close to first match !  
  
boolean ambiguity=false;  
if (bestCount>1) diff=bestMSEs[1]-bestMSEs[0];  
if (diff<0.3) ambiguity=true;
```



# Scan2Run (Improved) Design over JavaOCR

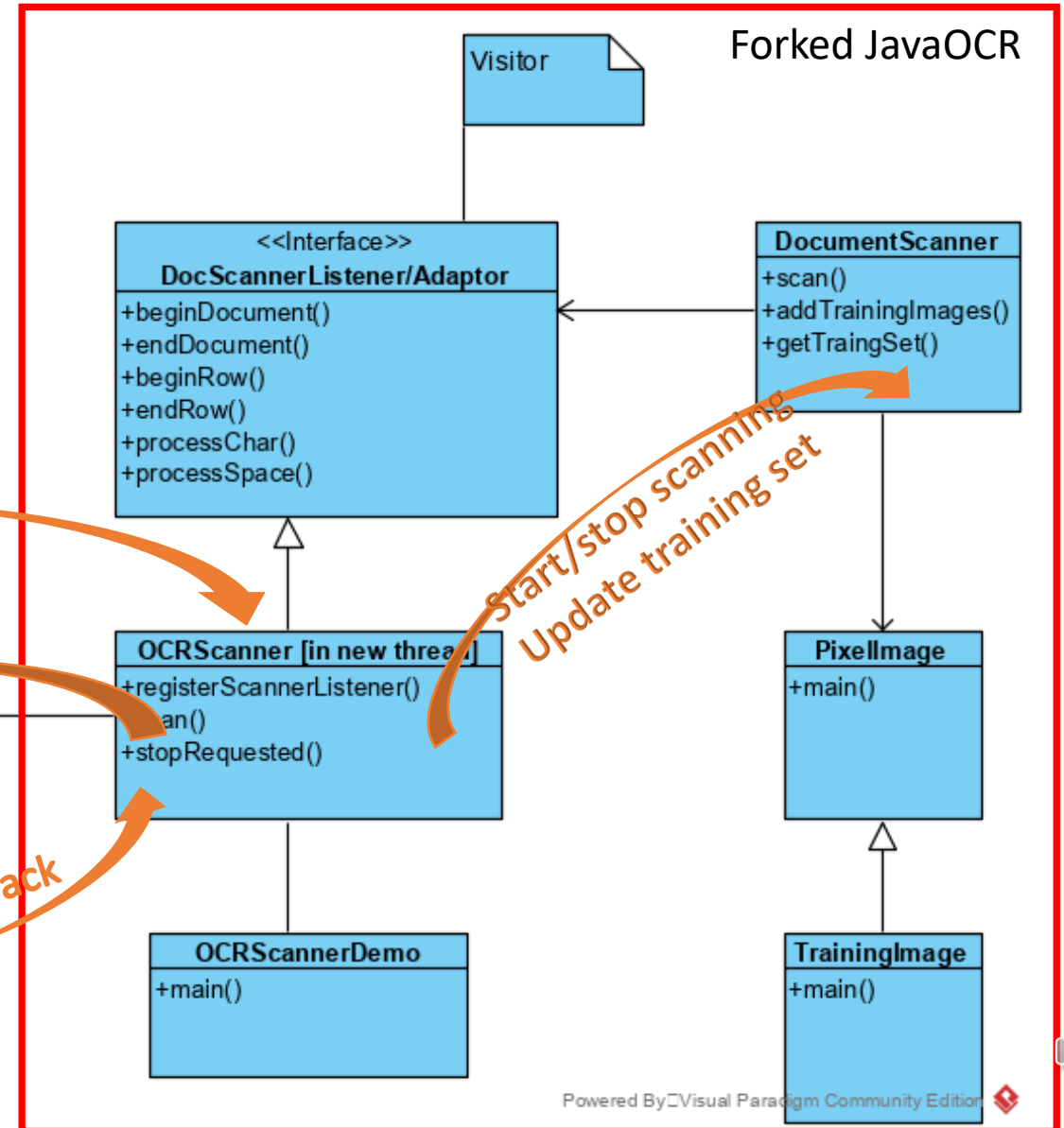
Interactive scanning application



Start/stop scanning

Update UI

User Feedback



Forked JavaOCR





# Demo Scan

```
File Scan Training Help

1  MODE 0
2  REM M.DIERCKX
3  REM IMP INT
10  REM het maken van een doolhof
20  MODE 2:COLORG 0 8 8 0
80  FOR I=0 TO XMAX STEP 10:DRAW I,0 I,YMAX 21:NEXT
90  FOR I=0 TO YMAX STEP 10:DRAW 0,I XMAX,I 21:NEXT
130  EIND=RND(30)+60
135  X1=0:Y1=0
140  FOR I=1 TO EIND
150  IF INT(RND(2))=0 THEN FLAG=1:X=RND(XMAX):DRAW X1,Y1 X,Y1 0:
160  IF FLAG=0 THEN Y=RND(YMAX):DRAW X1,Y1 X1,Y 0:Y1=Y
165  FLAG=0
170  NEXT I
175  DRAW X1,Y1 X1,YMAX 0
176  DRAW X1,YMAX XMAX,YMAX 0
```

Feedback (hit enter to validate)  
Not started yet

# Performance: % of requests, % of errors ?

- Depends on scan quality !
  - Use good scan quality ( $\geq 300$  DPI)
- Some internal preprocessing but better to:
  - Remove non listing part, artefacts
  - Enhance contrast
- Typical requests:
  - Medium quality 10% requests on 1 page listing with no learning set
  - High quality, more pages: can go down to 3% → easier than retyping
- Error rate:
  - Not assessed exactly so far: partial visual inspection
  - Can be detected when injecting (syntax error) or running (not expected behaviour)
  - Very good listing → no error but usually some systematic (ambiguity) → correct or adapt parameter
- Know limitations with workaround or needing enhancement
  - Character tracer can merge some character → possible to encode more than 1 char at request
  - Small punctuation characters have high MSE probably due to small size, need to “normalize” ?
  - Row problem → need to adjust internal parameter (not available at UI) or to edit scan (more heavy)



# RUN STEP - Scripting MAME to Inject Code

- Use MAME in console mode with LUA scripting

```
mame -console -autoboot_script <import_functions>.lua
```

- LUA scripting functionalities
  - usual functions, e.g. for **file IO to guest system**
  - list configuration
  - pause/unpause
  - I/O: access memory, **post key events**, write on screen, K7,...
  - various listeners: **frame**, frameDone, Sound, Periodic
  - ...
- Simplest design: send lines
  - assume some buffering if emulator limits input
  - In two lines:

```
f=io.open(path)
for line in f:lines() do emu.keypost(line.."\\n") end
```



# Injecting into MAME - DEMO

```
MOD=8
basic_s="";
basic_l=0;

--
function readAll()
    local f=io.open("C:\\DEV\\NAM-IP-WEB\\DAI\\CPC_BOMB.BAS")
    basic_s = f:read("*all")
    basic_l = string.len(basic_s)
end

function basic_load()
    readAll();
end

i=0;
j=0;
function basic_post()
    if ((i%MOD==0) and (j<=basic_l)) then
        emu.keypost(c);
        io.write(c);
        j=j+1;
    end
    i=i+1;
end

emu.pause();
readAll();
emu.unpause();
emu.register_frame(basic_post)
```

Will write to MAME every MOD frame  
Can be tuned to “match” emulator  
ability to capture character

Fix developed for some problems with DAI  
(but did not work actually, see other workaround)



The screenshot shows a Windows desktop with two windows open. The top window is Notepad++, editing a file named 'LIST-MAZE.bas'. The code in the file is a BASIC program for a maze game, with comments in Dutch. The bottom window is a MAME console, showing the execution of the 'mame64 -console' command. The console output displays a maze made of slashes, followed by version information for MAME (0.222) and Lua (5.3). The user has then executed a Lua script that reads the 'LIST-MAZE.BAS' file and prints its contents line by line, along with the average speed (213.21% (476 seconds)).

**Notepad++ - LIST-MAZE.bas**

```
1 1 MODE 0
2 2 REM M.DIERCKX
3 3 REM IMP INT
4 10 REM het maken van een doolhof
5 20 MODE 2:COLORG 0 8 8 0
6 80 FOR I=0 TO XMAX STEP 10:DRAW I,0 I,YMAX 21:NEXT
7 90 FOR I=0 TO YMAX STEP 10:DRAW 0,I XMAX,I 21:NEXT
8 130 EIND=RND(30)+60
9 135 X1=0:Y1=0
10 140 FOR I=1 TO EIND
11 150 IF INT(RND(2))=0 THEN FLAG=1:X=RND(XMAX):DRAW X1,Y1 X,Y1 0:X1=X
12 160 IF FLAG=0 THEN Y=RND(YMAX):DRAW X1,Y1 X1,Y 0:Y1=Y
```

**MAME Console**

```
C:\Users\cp\Documents\PERSONAL\IP\EMU\MAME>mame64 -console
// // // // // // //
// // // // // // //
// // // // // // //
// // // // // // //
// // // // // // //

mame 0.222
Copyright (C) Nicola Salmoria and the MAME team

Lua 5.3
Copyright (C) Lua.org, PUC-Rio

[MAME]>
[MAME]> f=io.open("C:\\Users\\cp\\git\\scan2run\\listingTests\\LIST-MAZE.BAS")
[MAME]> for line in f:lines() do emu.keypost("\n\n\n\n"..line.."\\n\n\n") end
[MAME]> Average speed: 213.21% (476 seconds)

C:\Users\cp\Documents\PERSONAL\IP\EMU\MAME>
```

Alain-LabNaf.d...

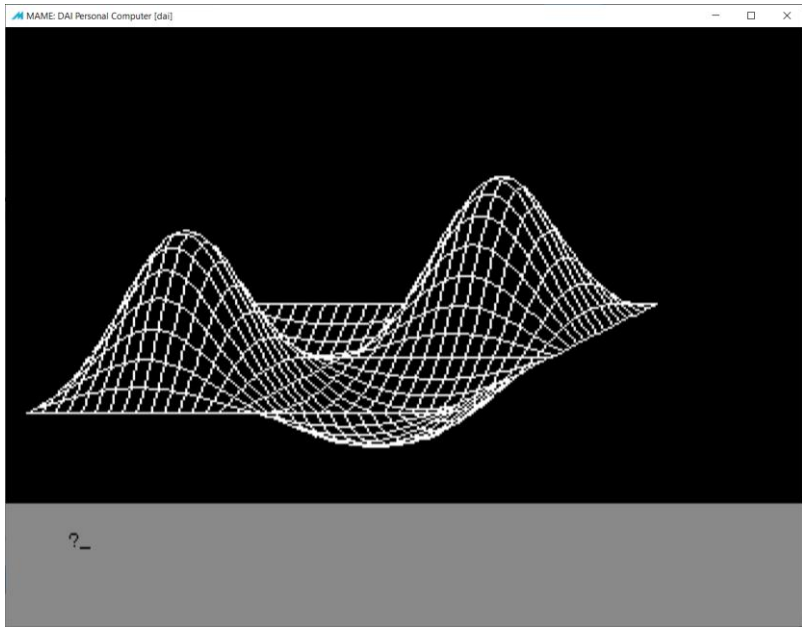
POINT OP.txt

achats.txt

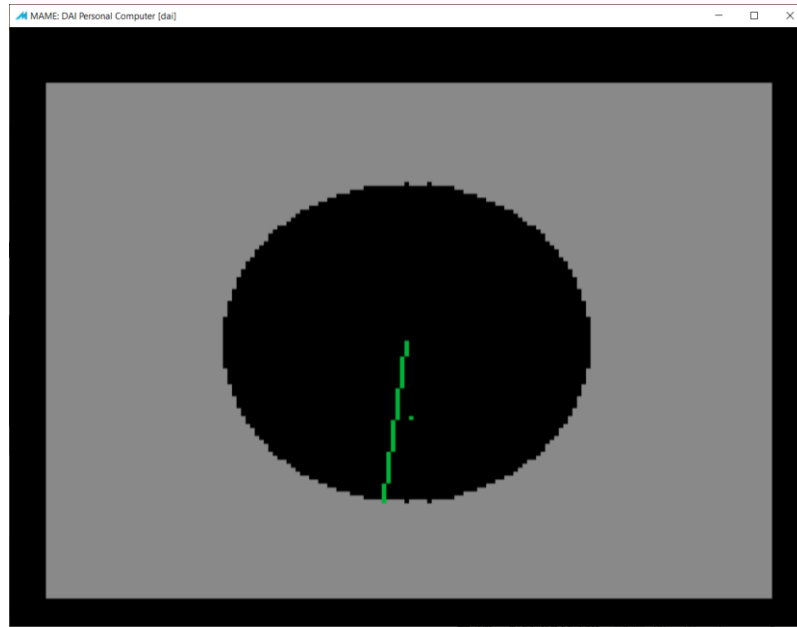
IDEES-TFE-FSES...

FreeMind

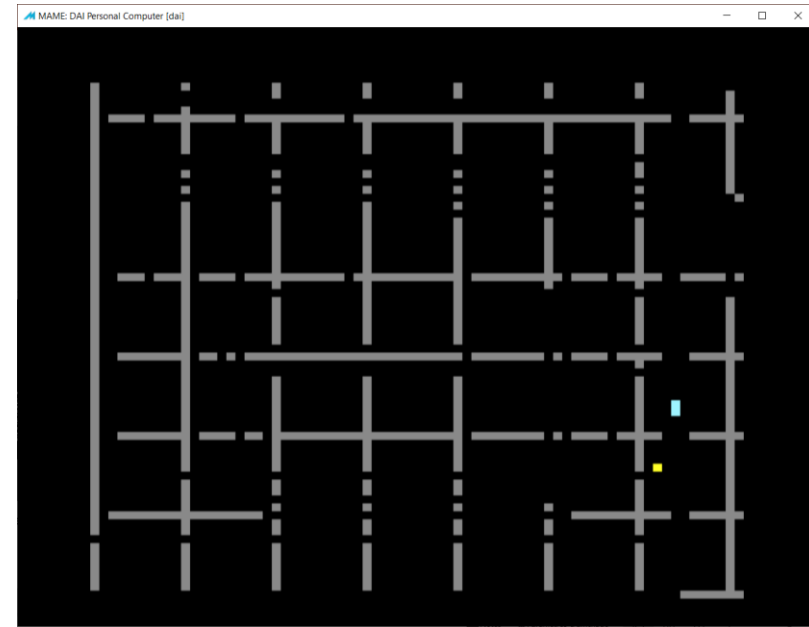
# Some recovered programs on DAI (from DAINAMIC magazine)



Math 3D plotter



Radar simulator (with sound)



MAZE Game





# Conclusion / Next Steps

- Try it from here – beware this is still work in progress

<https://github.com/NAMIP-Computer-Museum/scan2run>

- Quite “BASIC” approach but gets the work done for such BASIC programs ;-)
- Architecture quite easy to master
  - ➔ possible to extend with more powerful learning capabilities !
- Feedback / ideas / contributions welcome !

