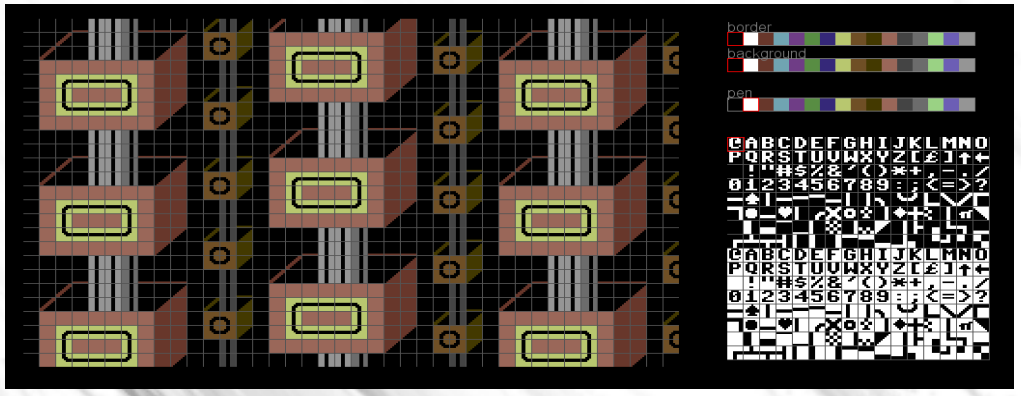
Software Studies for Media Designers

Media Lab,
Markku Reunanen

Personal background

- BASIC
- C
- Pascal
- Assembly language
- Processing
- Shell scripts
- PHP
- Plus some more



Overview of Programming Languages

A bit of history and some ways to categorize existing tools

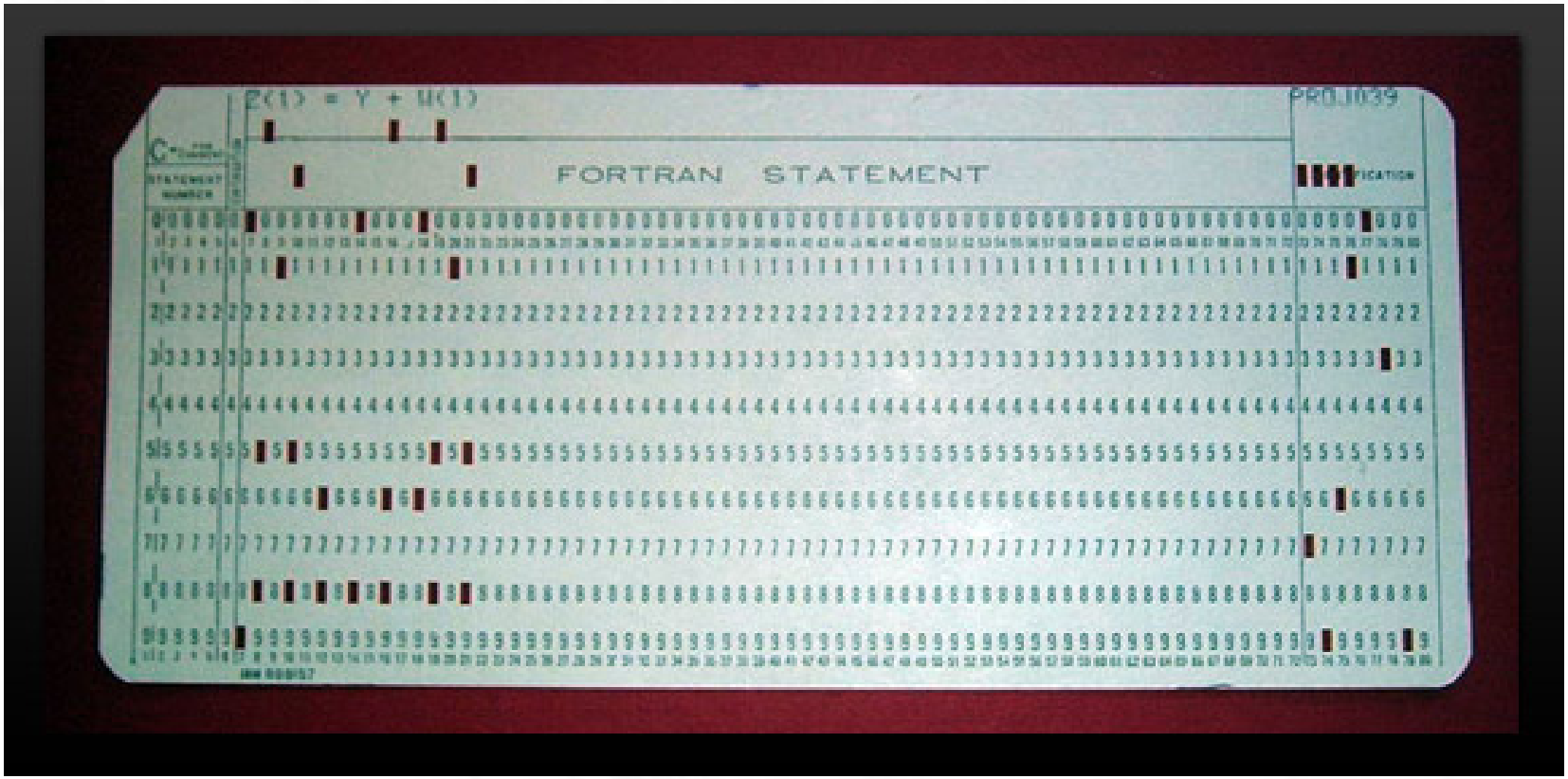
by

Markku Reunanen

History (1)

- Digital computers appeared in the late 1940s and early 1950s
- At first very crude machines
  - Hardwired functionality
  - Hand-coded machine language
  - “Typing” by flicking switches or writing software on punch cards
  - Symbolic machine language easier for humans: assembly language

History (2)



History (3)

- Some assembly language (directly mapped to machine instructions)

```
ld    a,100
ld    b,[var]
cmp a,b
jr    z,equal
ld    [var],b
ret
```

History (4)

- First high-level languages appeared in the 1950s
  - FORTRAN (1954), still used in scientific computing. Later on evolves to BASIC (1964).
  - LISP (1958), “artificial intelligence” language, still used for scripting
  - COBOL (1959), used to be very popular in business applications
- Terminals, punch cards, teletype

History (5)

- FORTRAN:

```
      INTEGER A,B,C
      READ(5,501) A,B,C
  501 FORMAT(3I5)
      IF(A.EQ.0 .OR. B.EQ.0) STOP 1
      S = (A + B + C) / 2.0
      WRITE(6,601) A,B,C
```

History (6)

- BASIC was the operating system for most 8-bit home computers in the 1980s

```
10 REM MY FIRST PROGRAM
20 PRINT”ENTER YOUR NAME:”
30 INPUT A$
40 IF A$=”MARKKU” THEN PRINT”YEAH!”
50 END
```

History (7)

- 1960s and 1970s: procedural programming languages
  - Pascal (1970), still in use as Borland Delphi (1995)
  - C (1972), still in wide use, forefather of C++, Java etc.
- The advent of object-oriented languages
  - Smalltalk (1972) by Alan Kay
  - C++ (1980), still in wide use
  - Java (1995), both server and client side

Terminal time

- Digital VT100 terminal

History (8)

- An example in C

```
#include <stdio.h>
main()
{
   int n;
   printf(“Even numbers up to 100:\n”);
   for(n=2;n<=100;n+=2)
      printf(“%d\n”,n);
}
```

History (9)

- Pascal is conceptually very similar:

```
program Numbas;
var
    n:integer;
begin
    writeln('All numbers up to 100');
    for n:=1 to 100 do
        writeln(a);
end.
```

History (10)

- 1990s: web-oriented languages
- Server-side script languages
  - Perl (1987)
  - Python (1991), now used for many purposes
  - PHP (1995)
- Client-side languages
  - JavaScript (1995)
  - ActionScript (1998)

History (11)

- Other notable appearances
  - Logo (1968), for kids, developed by Wally Feurzeig & Seymour Papert. Turtle graphics.
  - Forth (1970), stack-based language
  - Max (mid-1980s), Pure Data (mid-1990s), visual programming
  - Visual Basic (1991), for easy GUI programming
  - C# (2001), Microsoft Java-like

History (12)

- Logo turtle graphics example

```
TO PROGGIS
FD 100
LT 90
FD 100
RT 90
END
```

Future?

- General trend from low-level to high-level
- Object-oriented features, parallelism
- Standard libraries, components
- Web as application platform, platform independency
- Toys such as JavaScript became viable tools
- HTML5 Canvas, WebGL, WebAssembly ...

Processing (1)

- Website: http://www.processing.org/
- Casey Reas & Ben Fry from MIT, 2001
- Free, open source
- Available for Linux/Win/Mac
- Based on Java, Java components can be used
- Closest relatives: C, C++, C#, JavaScript

Processing (2)

- Making interactive and graphics programming easy to approach
- Not a toy – large-scale software can be written
- Basic functionality can be extended through a variety of libraries
- Programs are called “sketches”
- End result can be exported as a standalone Java application for desktop or Android

Processing? (1)

- This is *not* a Processing course. What you are actually learning are:
  - Fundamental concepts and terminology of programming
  - A programming mindset
  - These principles can be applied to many other languages and development tools as well, not just Processing

Processing? (2)

- Not just one monolithic “Processing”
  - Processing.js and P5.js: JavaScript frameworks
  - Processing iCompiler: iPhone version
  - Mobile Processing: J2ME for low-end mobiles
  - Processing for Android: Part of the normal distribution package already. Easy development for smartphones and tablets.

Processing? (3)

- Because of its legacy, learning Processing gets you started with other languages, too:
  - C/C++
  - Objective C, C#
  - Java
  - JavaScript/ECMAScript
  - Arduino (Wiring): Hacking electronics
  - PHP/Perl/Pascal/Delphi not far either
- You're being empowered! :)