NPRG075

Close look at past and today's programs

Tomáš Petříček, 204 (2nd floor)

✉ petricek@d3s.mff.cuni.cz

➡ https://tomasp.net | @tomaspetricek

Lectures: Tuesday 12:20, S6

➡ https://d3s.mff.cuni.cz/teaching/nprg075

Close reading

Two perspectives

Two perspectives on programs

Critical code studies

Interpreting the meaning of code, software or systems in socio-historical context

🔍 Attention to detail
Variable names

🦢 Making broad points
Labyrinths in culture

Complementary science

Use history & philosophy to answer questions science itself neglects

⚖️ Attention to detail
How exactly did it work

⚛️ Making those relevant
New mode of interaction

```cpp
// Your first C++ program
#include <iostream>

int main()
{
  std::cout << "Hello World!\n";
  return 0;
}
```

Close reading

"Close reading is the careful, sustained interpretation of a brief passage of a text"

What can we learn?

Not always educational start (Java, Haskell)

Reference to a long-term hacker culture

Close reading
Programming language design

- Understand socio-historical context
- Design for better social & cultural use?
- Understand lost ideas from the past
- Recover and adapt what may be useful!

Critical code studies
Closer look at code

Hello World in Piet

Why look at esoteric languages?

We must not just observe nature in the raw, but also "twist the lion's tail" to get at hidden insights

May reveal facts about normal languages too!

The meaning of programs

Speaking code

"Like all codes, [source code] is only interpretable within the context of the overall network of relations that make its operations unstable."

```
>+++++++++[<++++++++++>-]<
.>+++++[<+++++++>-]<+.+++
++++..+++.>>+++++++[<++++
+++>-]<++.------------.>
++++++[<+++++++++>-]<+.<
.+++.------.--------.>>>
++++[<++++++++>-]<+.
```
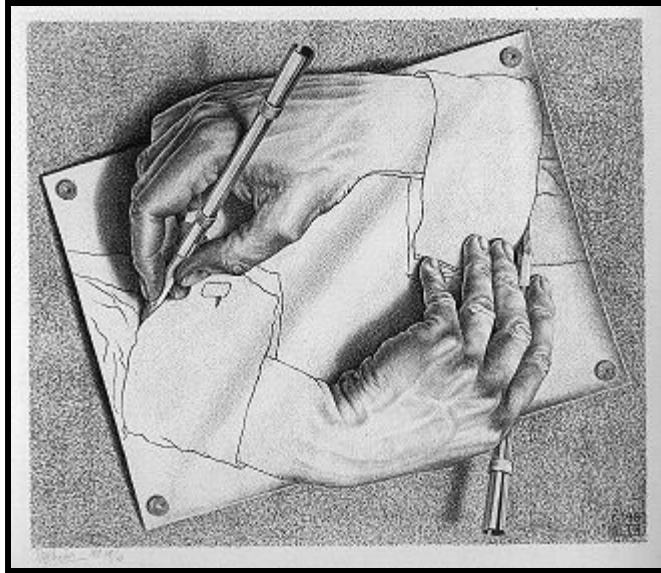
Meaning of code

- ### Meaning for the machine
  Relies on technological context - compilers, specification
- ### Meaning for a human reader
  Relies on socio-cultural context

Multiple levels of meaning

```
Až jde každý vlastní cestou!
Až jde každý vlastní cestou!
Až jde každý vlastní cestou!
Až jde každý vlastní cestou!
Až jde každý vlastní cestou!
Až jde každý vlastní cestou!
Až jde každý vlastní cestou!
Až jde každý vlastní cestou!
Až jde každý vlastní cestou!
Až jde každý vlastní cestou!
Až jde každý vlastní cestou!
Až jde každý vlastní cestou!
Až jde každý vlastní cestou!
Až jde každý vlastní cestou!
Až jde každý vlastní cestou!
xx xxxAž jdexkaždýxvlastníxxx
xx xxxxxx xvlastníxxx

Až jde každý vlastní cestou!! xcestouixxxxx
Až jde každý vlastní cestou!!
Až jde každý vlastní cestou!!
Až jde každý vlastní cestou!!
Až jde každý vlastní cestou!!
Až jde každý vlastní cestou!!
```

```c
                                                        char
                                                  _3141592654[3141
                     ],__3141[3141];_314159[31415],_3141[31415];main(){register char*
                       _3_141,*_3_1415, *_3__1415; register int _314,_31415,__31415,*_31,
                        _3_14159,__3_1415;*_3141592654=__31415=2,_3141592654[0][_3141592654
                         -1]=1[__3141]=5;__3_1415=1;do{_3_14159=_314=0,__31415++;for( _31415
                        =0;_31415<(3,14-4)*__31415;_31415++)_31415[_3141]=_314159[_31415]= -
                      1;_3141[*_314159=_3_14159]=_314;_3_141=_3141592654+__3_1415;_3_1415=
                __3_1415      +__3141;for                          (_31415 = 3141-
                      __3_1415  ;                                  _31415;_31415--
                      ,_3_141 ++,                                  _3_1415++){_314
                      +=_314<<2 ;                                  _314<<=1;_314+=
                      *_3_1415;_31                                 =_314159+_314;
                      if(!(*_31+1)                                 )* _31 =_314 /
                      __31415,_314                                 [_3141]=_314 %
                      __31415 ;* (                                 _3__1415=_3_141
                      )+= *_3_1415                                 = *_31;while(*
                      _3__1415 >=                                  31415/3141 ) *
                      _3__1415+= -                                 10,(*--_3__1415
                      )++;_314=_314                                [_3141]; if ( !
                      _3_14159 && *                                _3_1415)_3_14159
                      =1,__3_1415 =                                3141-_31415;}if(
                      _314+(__31415                                >>1)>=__31415 )
                      while ( ++ *                                 _3_141==3141/314
                      )*_3_141-==0                                 ;}while(_3_14159
                      ) ; { char *                                 __3_14= "3.1415";
                      write((3,1),                                 (--*__3_14,__3_14
                      ),(_3_14159                                  ++,++_3_14159))+
                      3.1415926; }                                 for ( _31415 = 1;
                      _31415<3141-                                 1;_31415++)write(
                      31415% 314-(                                 3,14),_3141592654[
                      _31415   ] +                                 "0123456789","314"
                     [ 3]+1)-_314;                                 puts((*_3141592654=0
                   ,_3141592654))                                 ;_314= *"3.141592";}
```

Mutual influences

Social shapes technical

Programming reflects our thinking about the world e.g. division of labour

Technical shapes social

Abstractions define how we think about software e.g. information hiding

Foo, bar, baz, …

(Lennon, 2018)

Cultural pointer

Akin to programming language pointers

Marks work as belonging to a particular culture

Abstract

   Approximately 212 RFCs so far, starting with RFC 269, contain the
   terms `foo', `bar', or `foobar' as metasyntactic variables without
   any proper explanation or definition. This document rectifies that
   deficiency.

Table of Contents

1. Introduction

   Approximately 212 RFCs, or about 7% of RFCs issued so far, starting
   with [RFC269], contain the terms `foo', `bar', or `foobar' used as a
   metasyntactic variable without any proper explanation or definition.
   This may seem trivial, but a number of newcomers, especially if
   English is not their native language, have had problems in
   understanding the origin of those terms. This document rectifies
   that deficiency.

Foo, bar, baz...
As cultural pointers

- Metasyntactic variable / meaning placeholder
- Variable names and comments are for humans
- Neither x nor `AbstractSingletonProxyFactoryBean`
- ARPANET and Request For Comments (RFCs)

Close look at UNIX 6

Process switching function

- Released in 1975 for PDP-11
- What can we learn about it?
- tinyurl.com/nprg075-unix

Close reading UNIX code

- Variable names: i, n, p, rp
- "set up his segmentation registers"
- "You are not expected to understand this."

You are not expected to understand this

The real problem is that we didn't understand what was going on either. The savu/retu mechanism (...) was fundamentally broken (...).

[It] worked on the PDP-11 because its compiler always used the same context-save mechanism (...). [Eventually we] redid the coroutine control-passing primitives altogether, and this code section,
and the comment, passed into history.
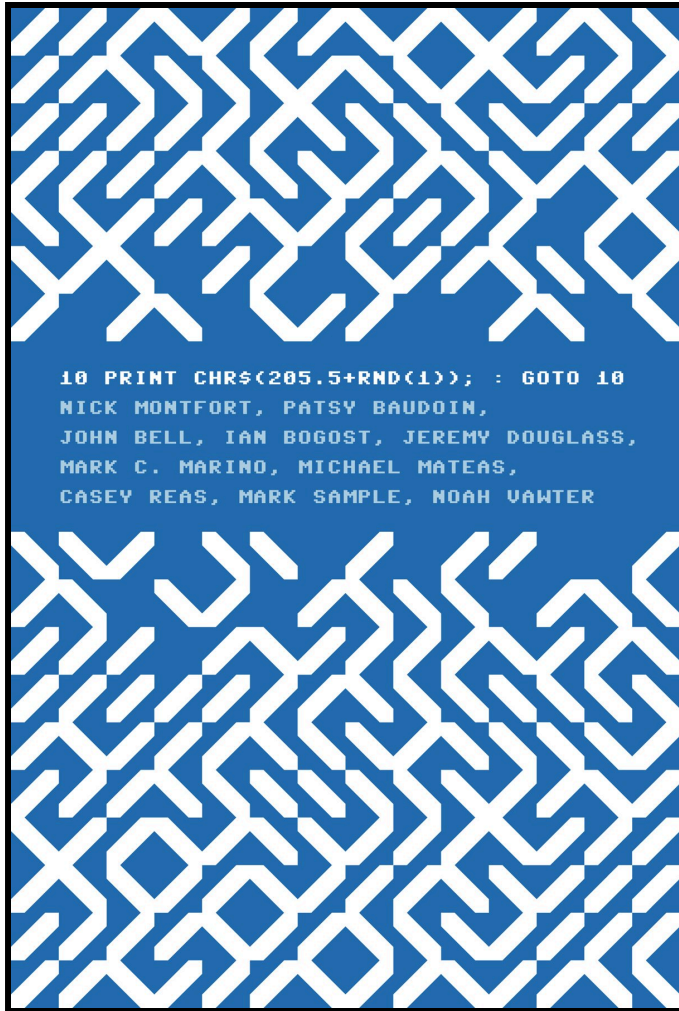
10 PRINT

Cultural context of a BASIC one-liner

The birth of microcomputers and tinkerer culture

Randomness and variations of the pattern

Recreating the one-liner in other systems


```
10 PRINT CHR$(205.5+RND(1)); : GOTO 10
NICK MONTFORT, PATSY BAUDOIN,
JOHN BELL, IAN BOGOST, JEREMY DOUGLASS,
MARK C. MARINO, MICHAEL MATEAS,
CASEY REAS, MARK SAMPLE, NOAH VAWTER
```

Critical code studies
Ideas for programming

🏛 What socio-technical context design uses?

👥 Design for hackers or non-programmers?

📖 Analyse what exists, show what could exist

🎭 "Performative science fiction" demos

Thimbl: Performative science fiction

Federated social network (~2011)

- Artwork, not to compete with Twitter
- Built with a different social context
- Can it work without investments?

How is it supposed to work?

- Built with as little code as possible
- Using SSH and Finger protocol (1970s)
- Low-tech version of ActivityPub (Mastodon)

Programming system demos

Future programming

- Imagining alternative ways
- Often through (limited) demos
- End-user, visual, domain-specific
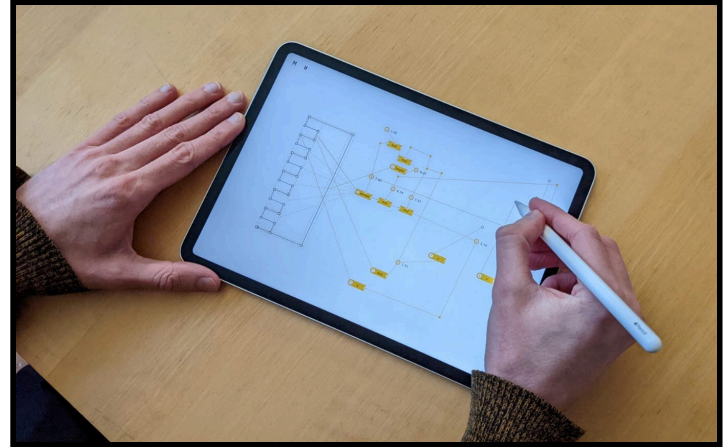
Places to look at

- Bret Victor: worrydream.com
- LIVE workshop: liveprog.org
- Ink & Switch: inkandswitch.com

Demo
Crosscut: Drawing Dynamic Models

Complementary science
Learning from the past

Inventing Temperature
Measurement and Scientific Progress
HASOK CHANG
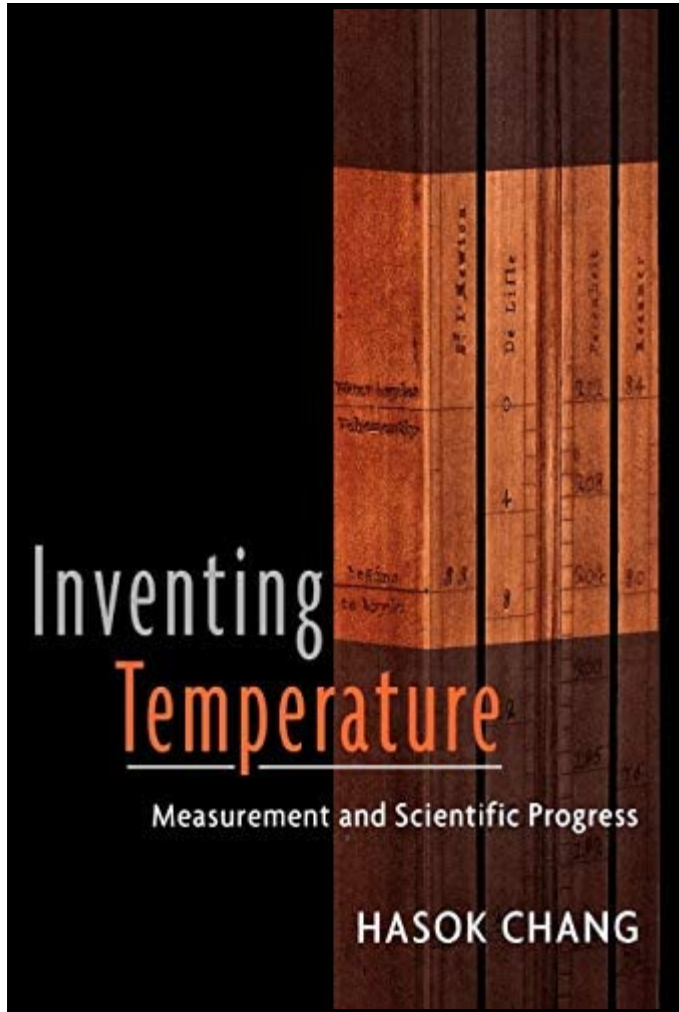
Complementary science

Contribute to scientific knowledge through historical and philosophical investigations

Effectiveness of science leads to dogmatism

Narrow focus can result in loss of knowledges

Heat reflection (1791)
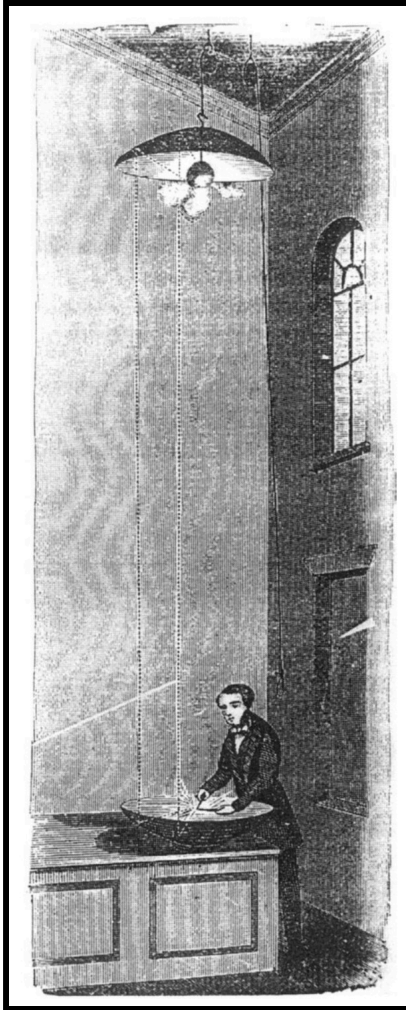
Heat produced by "caloric", cold maybe by another "positive" substance.

Heat is reflected by mirror!
Cold is absence of heat?
But also reflected!

Modern physicists never talk about reflection of cold!

Complementary programming?
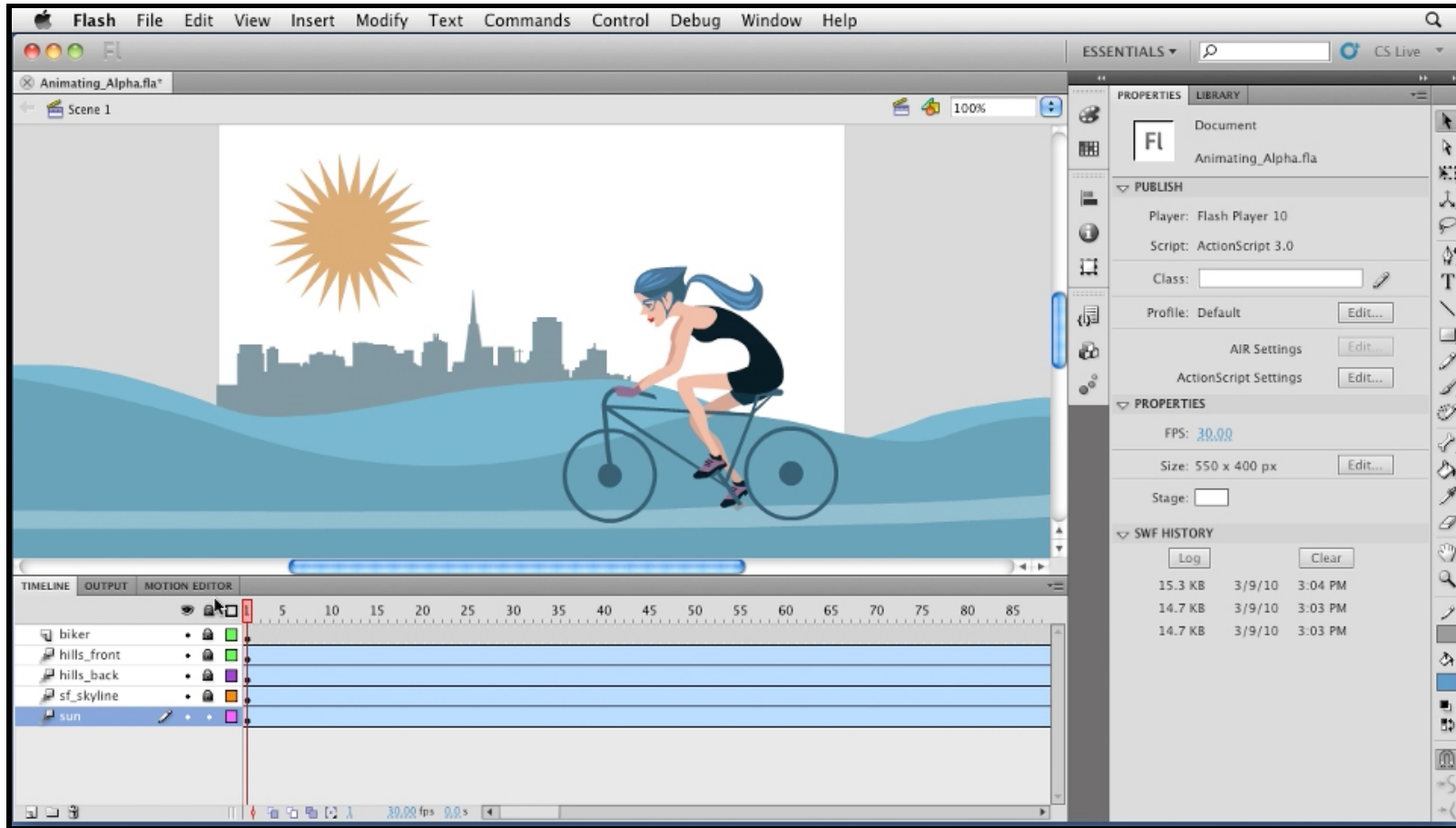
Dot-Com Design

(Ankerson, 2018)

Amateur can easily cobble something together

Hackability and familiarity of graphical editors

Gives designers full control

Complementary science
Why use it for programming

🚫 Feel all programming is the same?

🍃 Programming has brief but rich history

💵 Not discarded for experimental failures

>_ Ideas are (relatively) easy to recreate!

Demo
Annoying pop-ups of the 1990s

Learning from the 1990s web

View-source, copy and edit culture

Hosting on Geocities & creative community

Limited user protection (hacks are for fun)
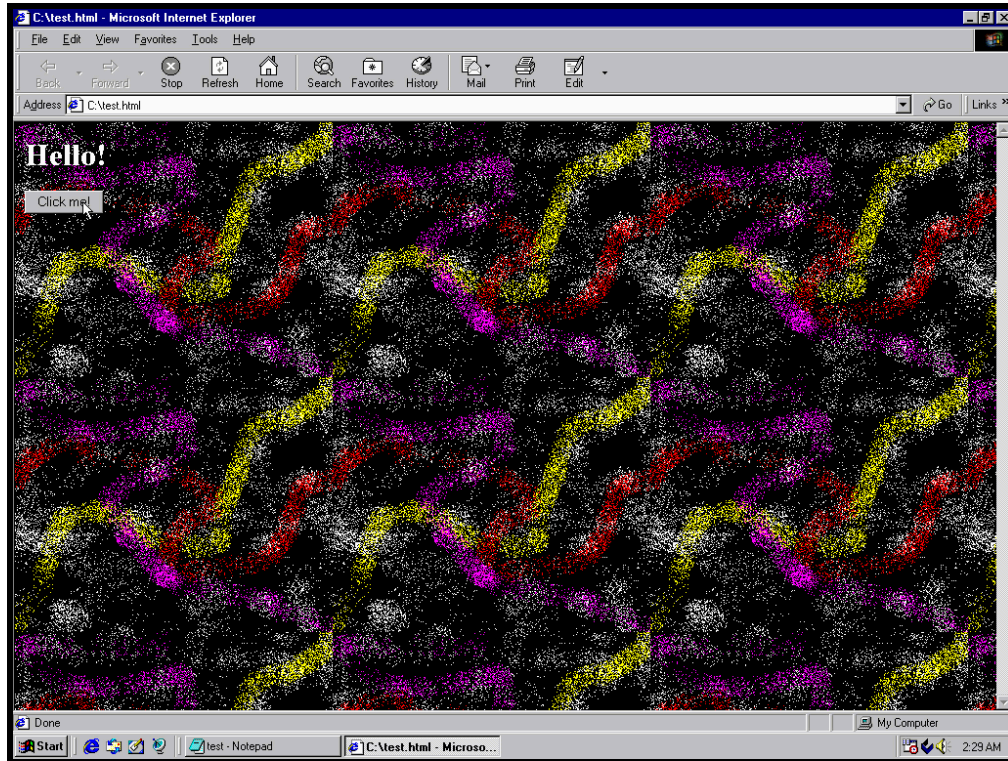
Two eras of the web

🔒 **2010s web**

Compiled code
Minified with dependencies

Custom elements
Pop-ups using `<div>`

Opaque structure
WebAssembly & Canvas

🔓 **1990s web**

View source
Readable source code

Copy & paste
Self-contained scripts

Pop-up windows
Unchecked window.open

Learning from the 90s web

WebStrates project

- Shareable dynamic media
- Document and code in DOM
- Synchronized across clients
- In-page editor & dev tools
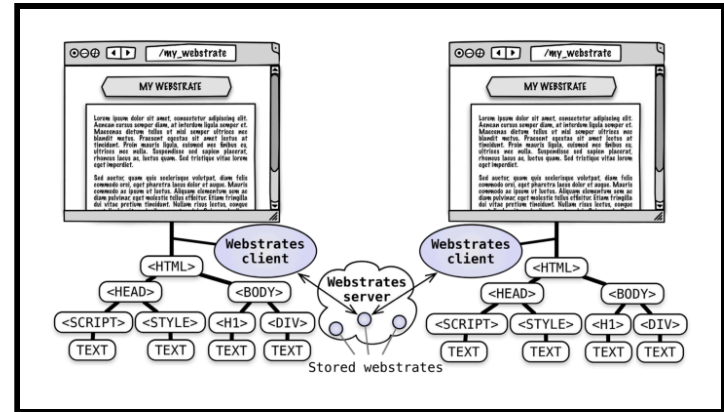
Further ideas

- How to support reuse by copying?
- Openness and addressability of DOM

Good old programming systems
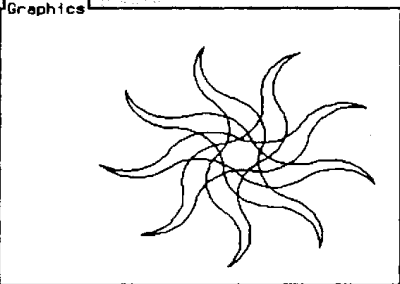
Boxer's naive realism
You see all there is

Smalltalk's self-sustainability
Built in itself

Good old programming systems

Pygmalion's programming
By demonstration
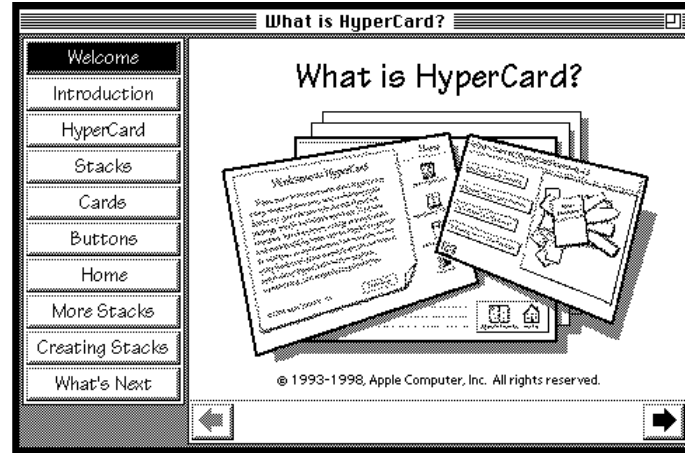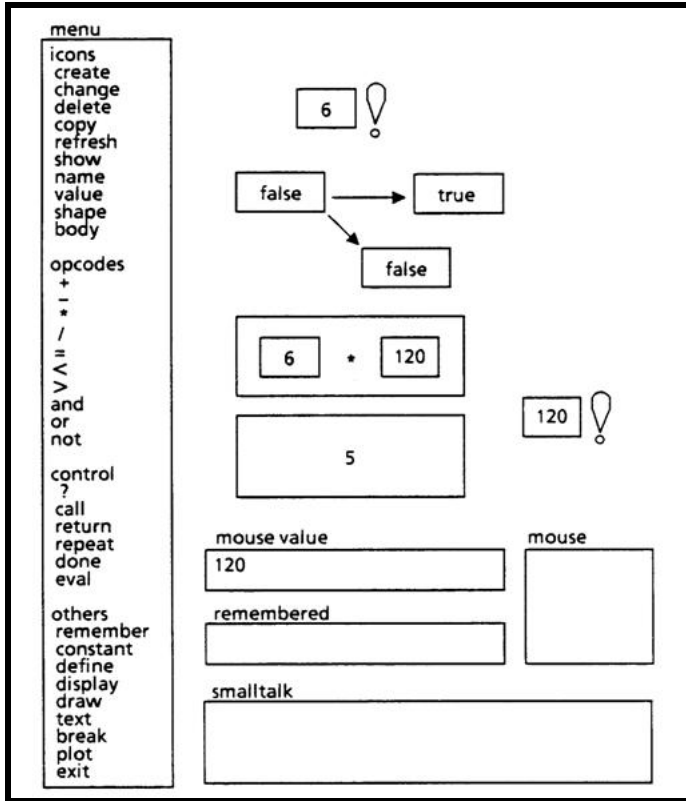
Hypercard's usability
From user to programmer

Conclusions

Close look at programs

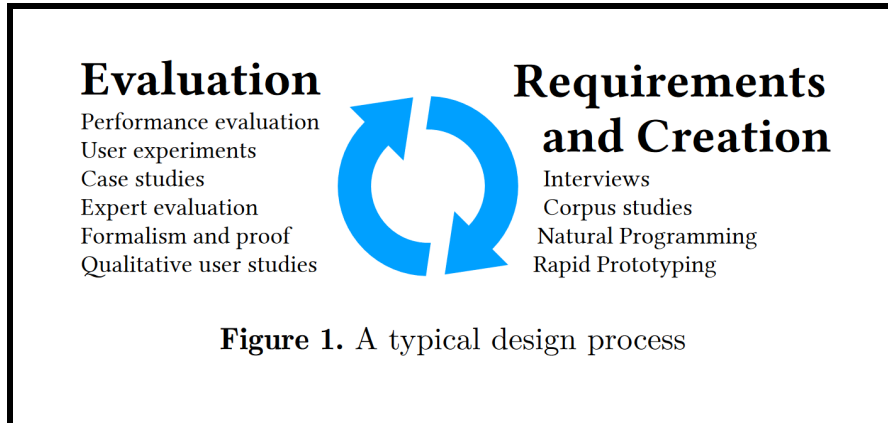Figure 1. A typical design process

Close look

As evaluation
Reveals more than one may immediately see

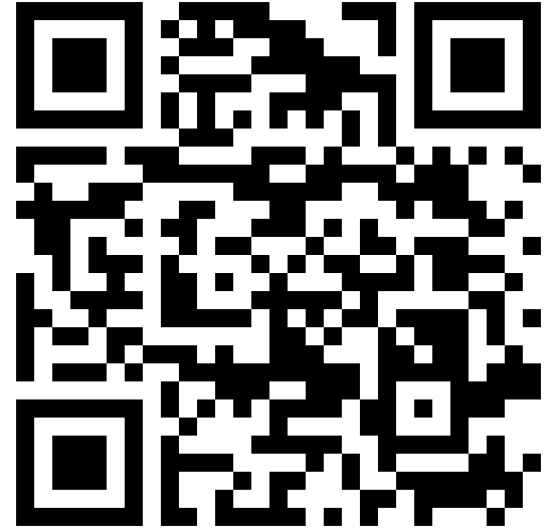## As design tool
Think about programming from new perspectives

Reading

Are spreadsheets programming?

- Spreadsheets are Code: An Overview of Software Engineering Approaches Applied to Spreadsheets
- tinyurl.com/nprg075-excel

Why should you read this?

- Interesting case of end-user programming
- How to use programming ideas in new domains

Conclusions

Close reading and complementary science

- Close look at fine coding details
- Reveals broader cultural points
- Close look at past programming systems
- Reveals ideas we may have forgotten

Tomáš Petříček, 204 (2nd floor)

✉ petricek@d3s.mff.cuni.cz

➡ https://tomasp.net | @tomaspetricek

➡ https://d3s.mff.cuni.cz/teaching/nprg075

References (1/2)

Critical code studies

- Marino, M. C. (2020). Critical Code Studies, MIT Press
- Cox, G. (2012). Speaking Code, MIT Press
- Lennon, B. (2021). Foo, Bar, Baz...: The Metasyntactic Variable and the Programming Language Hierarchy, Philosophy & Technology
- Montfort, N. et al. (2013). 10 PRINT CHR$(205.5+RND(1)); : GOTO 10, MIT Press

Interesting past systems

- diSessa, A. A., Abelson, H. (1986). Boxer: A Reconstructible Computational Medium., CACM
- Smith, D. C. (1975). PYGMALION: A creative programming environment, MIT PhD

References (2/2)

Complementary science & programming

- Chang, H. (2008). Inventing temperature, Oxford
- Ankerson, M. S. (2018). Dot-Com Design: The Rise of a Usable, Social, Commercial Web, NYU
- Petricek, T. (2021). Pop-up from Hell On the growing opacity of web programs, Online

History of UNIX

- Ritchie, D. (2002). Odd Comments and Strange Doings in Unix
- Bosch, T. ed. (2022). You Are Not Expected to Understand This: How 26 Lines of Code Changed the World, Princeton

Programming demos

- Kaliski, S. et al. (2022). Crosscut: Drawing Dynamic Models, Ink & Switch