

# The Early History of Smalltalk

Papers We Love Boston • May 26th, 2016

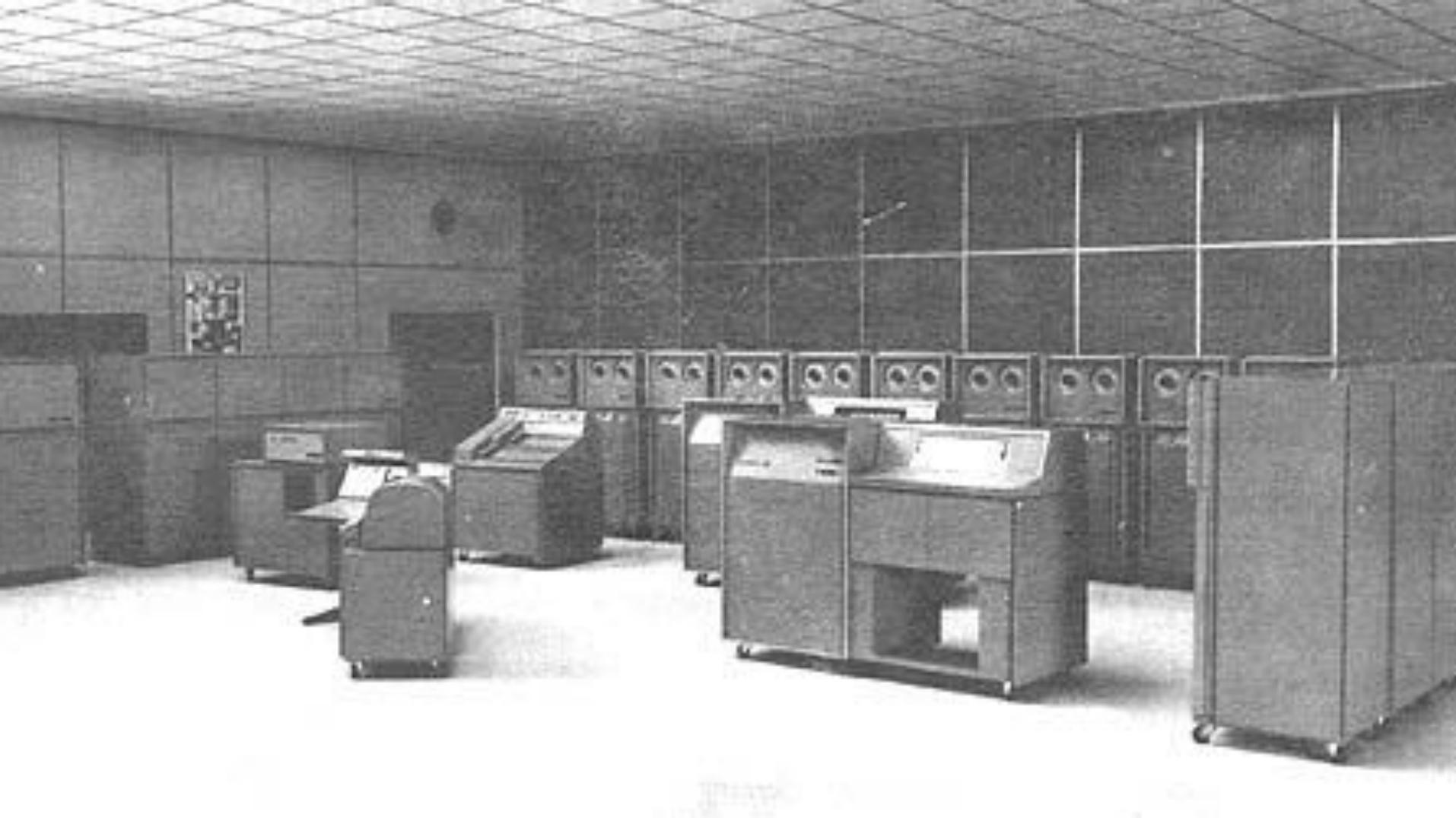
Douglas Creager • [@dcreager](https://twitter.com/dcreager)

A close-up photograph of a vintage clock face. The main dial features large, dark Roman numerals for 12, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, and 11. Between these numerals are smaller, unlabeled tick marks. A small seconds sub-dial is located at the top left of the main dial, showing a scale from 1 to 60. The entire clock face is framed by a decorative gold-colored bezel with a scalloped edge. The background is dark.

# Timeline

# Timeline





# Timeline



1960

1992

2016

**digital**

Digital Equipment Corporation Maynard Massachusetts

**decsystem 10**

NOT AN  
EXIT

# Timeline



1960

1992

2016



# Timeline



1960

1992

2016



# Timeline



1960

1992

2016



# Timeline



1960

1992

2016



# Timeline



1960



1992

2016



# Timeline



1960



1992

2016

PalmPilot

Robotics



# Timeline





# Timeline



1960

1992

2016



# Timeline



1960

1992

2016





# Timeline

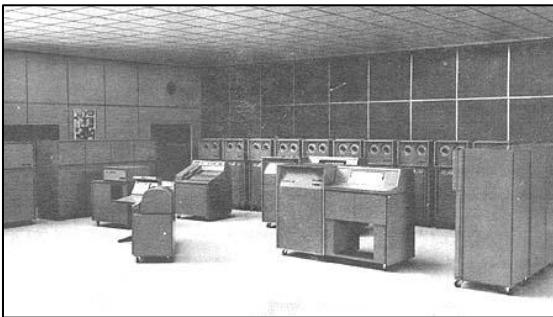


1960

1992

2016

# What a difference 50 years can make



Core memory: 10,000 44-bit words (~53kB)

Secondary storage:

magnetic tape (~12kB/s)

the “Datafile” (capacity ~10MB, avg access time 15 seconds, \$25,000)

Price: \$640,000 to \$1,209,000

Total production: ~55 units



Core memory: 4GB

Secondary storage:

flash storage: up to 256GB

the Internet

Price: \$799 to \$1299

Total production: ~2.6 million units just in 2015 Q4

# Timeline



1960

1992

2016

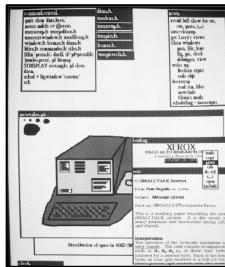
# Timeline



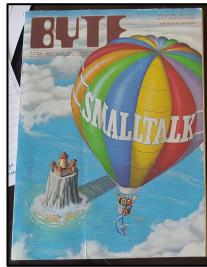
1960

1992

2016



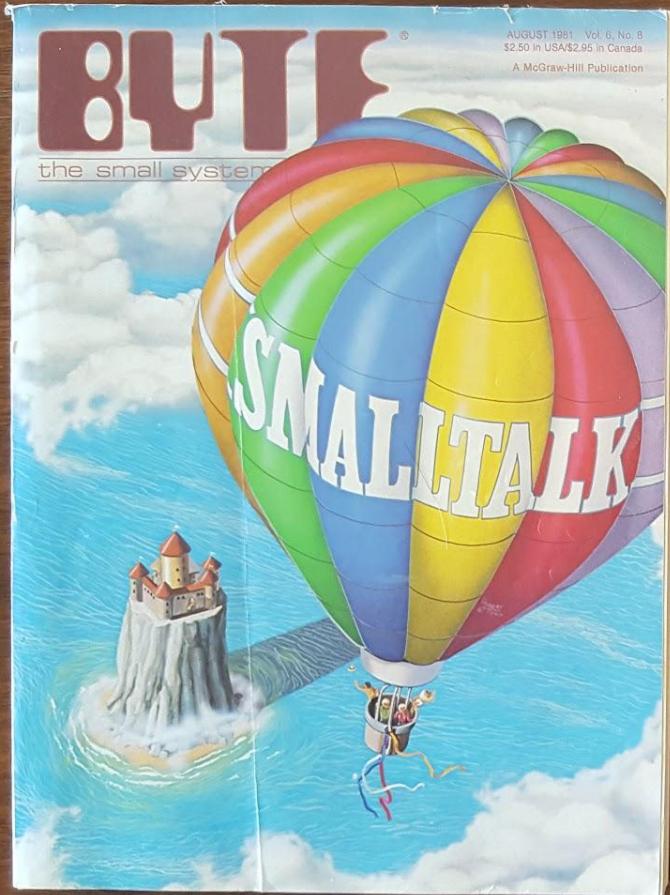
Smalltalk-72



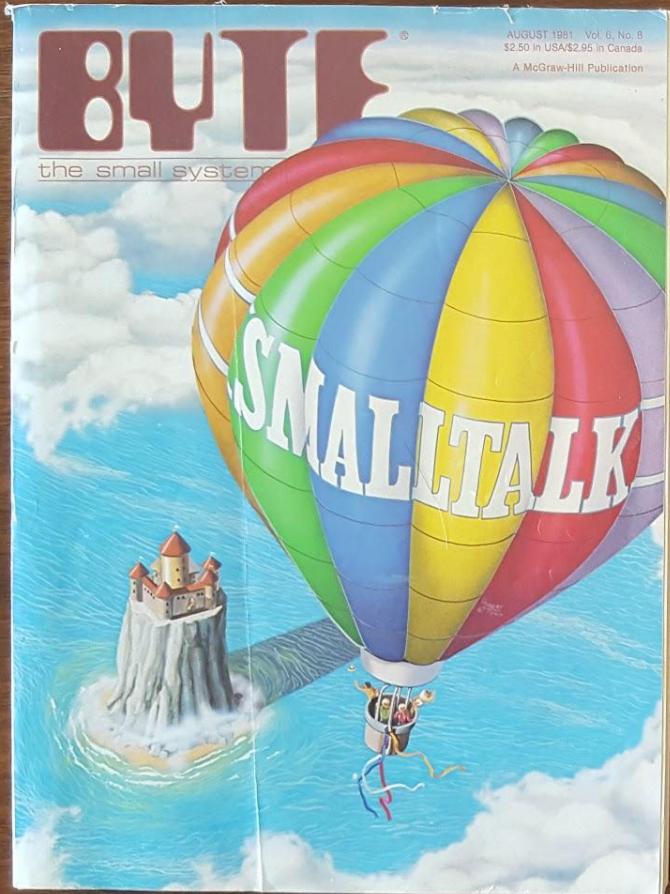
Smalltalk-80



1993: HOPL II



What is Smalltalk?



What is Smalltalk?  
What is OO?

# What is object-oriented programming?

Encapsulation

Access control

Polymorphism

Inheritance

Message passing

Classes / prototypes / objects / instances

Methods

Virtual dispatch / late binding

# Influences



# What did Smalltalk influence?

Basically every modern programming language!

Common Lisp Object System

Objective-C

Go

Java

...and indirectly:

Python

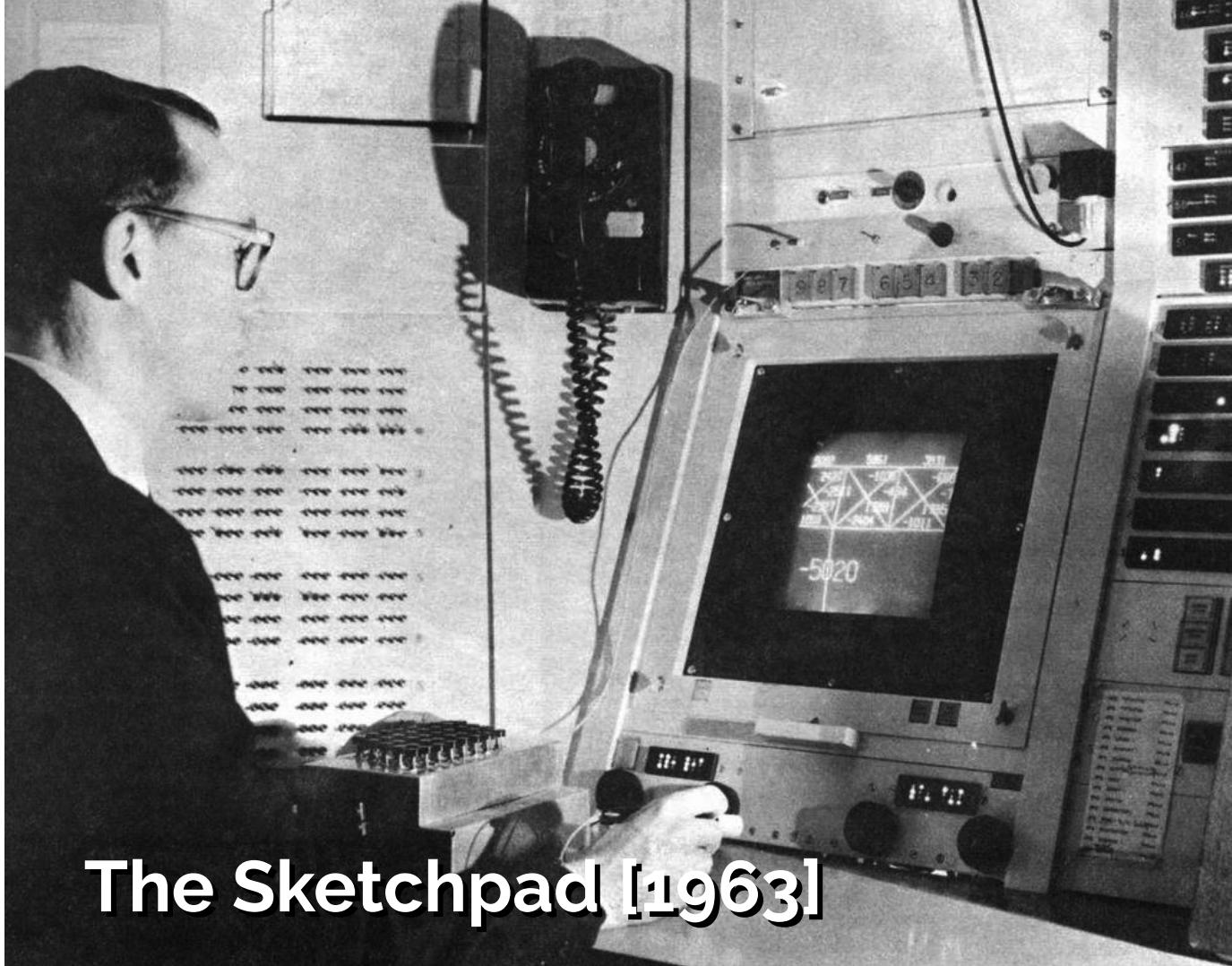
JavaScript

PHP

Ruby

Actor model [Hewitt 1973] → Erlang

# What influenced Smalltalk?



The Sketchpad [1963]

# The Sketchpad [1963]

- “the invention of modern interactive computer graphics”
- “master drawings” and “instance drawings”
- “control and dynamics were supplied by ‘constraints’”
- “clipping and zooming windows”



Simula [1966]

# Simula [1966]

“the documentation read like Norwegian transliterated into English, which in fact is was”

“[we] unrolled the program listing 80 feet down the hall and crawled over it yelling discoveries to each other”

“most of my ideas from then on took their roots from Simula — but not as an attempt to improve it”

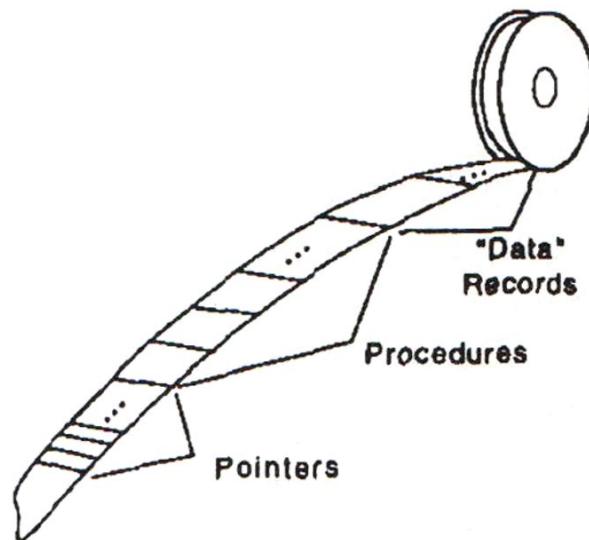
# Burroughs B220 and B5000



# Burroughs B220 and B5000

Tape storage layout:

**FIGURE 11.1** USAF ATG Randolph AFB  
B220 File Format ca. 1961

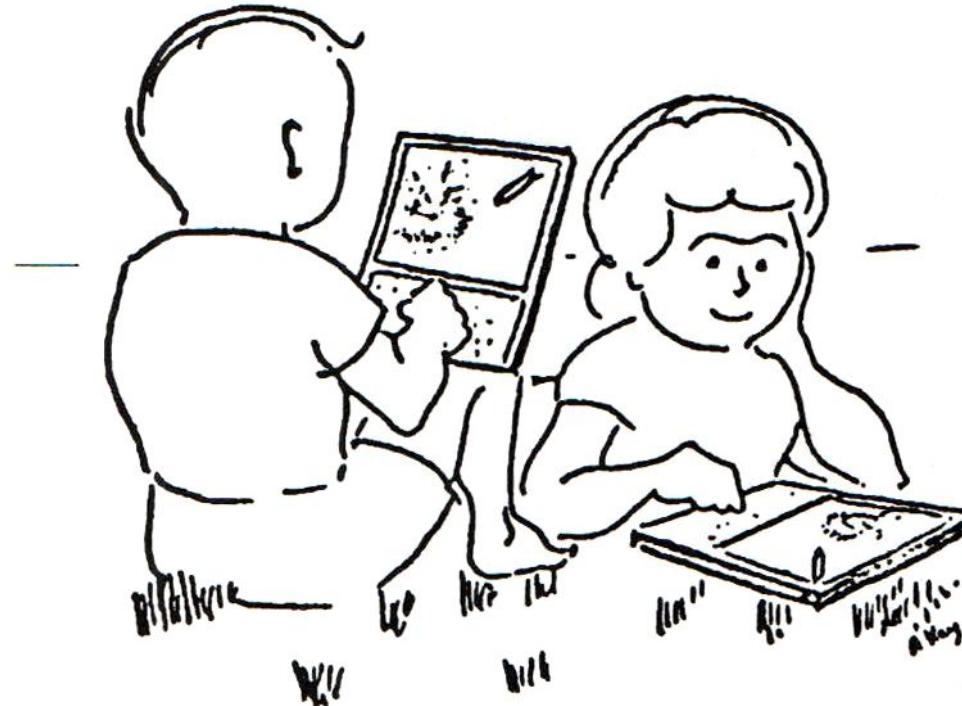


# "Personal" computers



# Kids!

**FIGURE 11.27** Children with Dynabooks from “A Personal Computer for Children of All Ages” [Kay 1972]



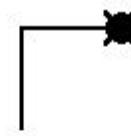
# LOGO



forward 50



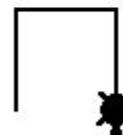
right 90



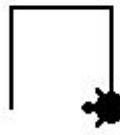
forward 50



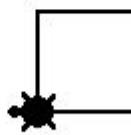
right 90



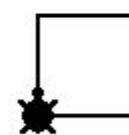
forward 50



right 90



forward 50



right 90

© 2000 Logo Foundation

# Guiding principles





Simplicity and consistency

# LISP special forms

```
(+ 4 5)
```

```
(+ (* 2 4) 8)
```

```
(or (= 4 4) (> 2 (/ 10 0)))
```

# Consistency in the core language

“Why on earth call [LISP] a functional language at all?”

“Why not just...force evaluation on the receiving side when needed?”

→ Haskell and other non-strict languages

“Take the hardest and most profound thing you need to do,  
make it great, and then build every easier thing out of it”

***Everything*** is an object or a message



**Beauty**

# Beautiful programming languages

Parsimony

Generality

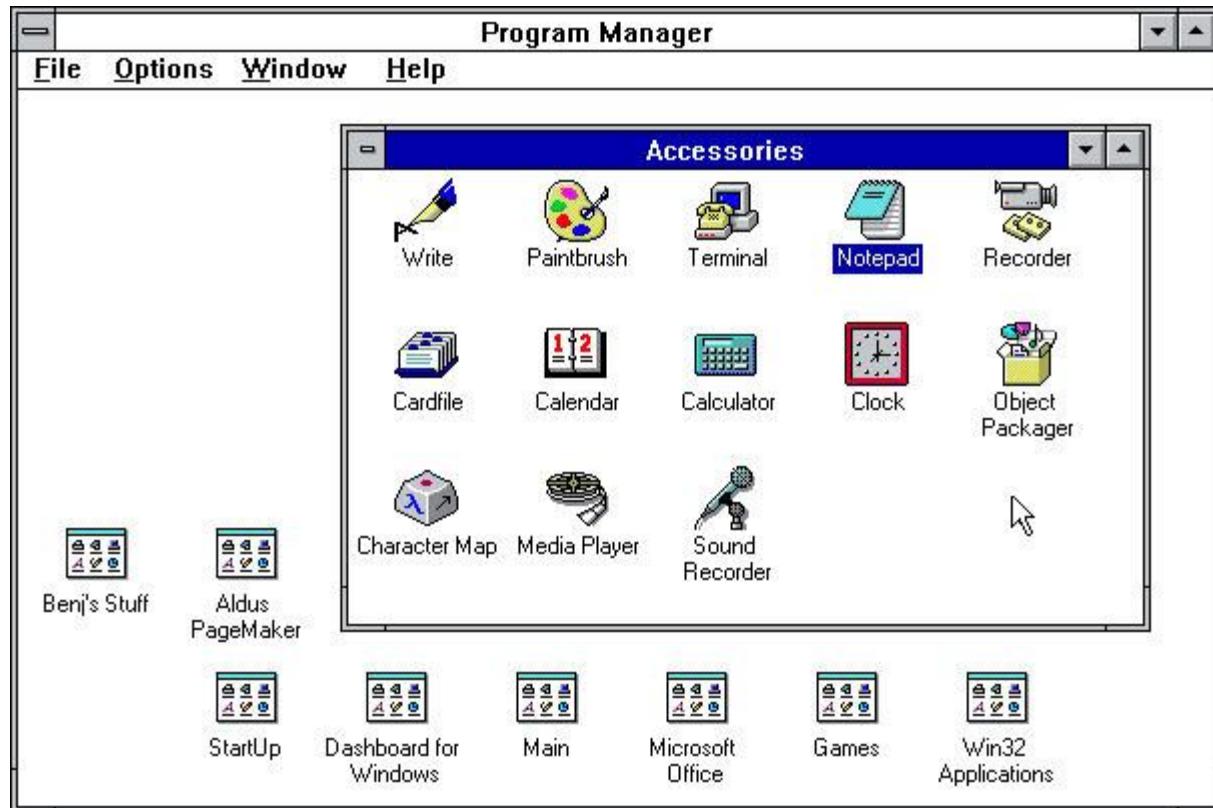
Enlightenment

Finesse

needs “a good match between the degree of interestingness  
and the level of complexity needed to express it”

“Nature is wonderful **both** at elegance and practicality”

# GUIs



# GUIs in the present

Turbo Vision (DOS, character-based, Borland products, mid-1990s)

MFC (Visual C++, 1992)

AWT / Swing (Java)

GNOME / GTK / GObject (Linux, C)

KDE/Qt (Linux etc, C++)

The DOM! (browsers, JavaScript)

# GUIs in the 1960s and 1970s

Bitmapped displays

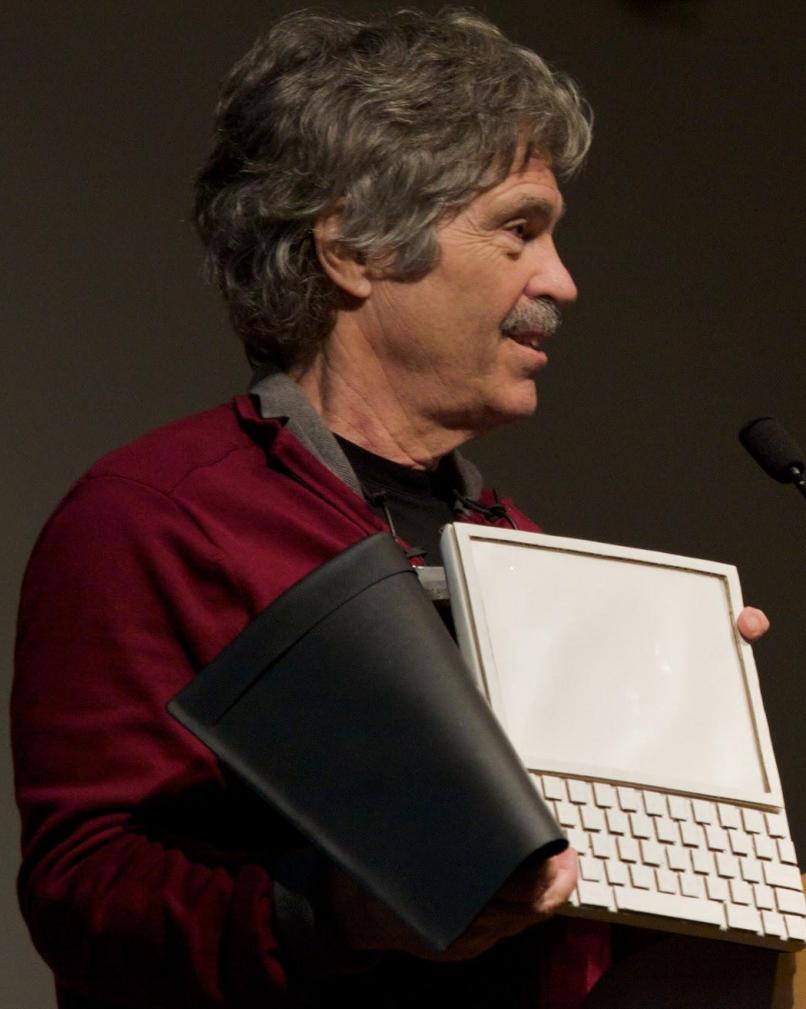
The mouse

Multiple overlapping windows

“Multimedia” documents

Clipped and zoomable windows (infinite scroll?)

# The Dynabook!!



# The Dynabook!

Size of a notepad

Weighs less than 2 pounds

Stylus / touch *and* a built-in physical keyboard

Wireless connectivity (!)

Eternal battery life (!!)

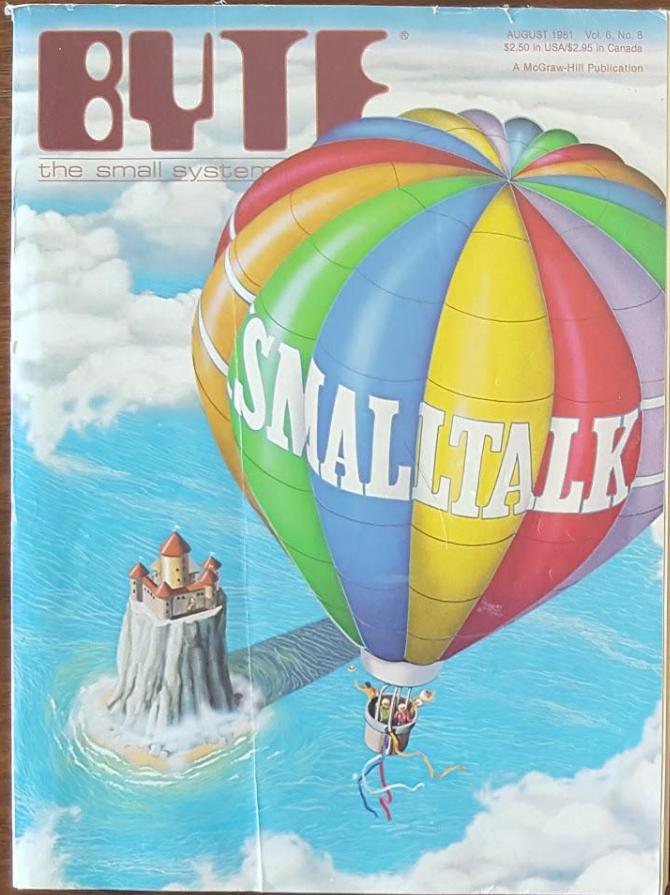
Seriously it's amazing:

20 years before first commercial laptops

30 years before Internet became widespread

40 years before iPhone, 3G, Wifi

50 years before iPad



# What is Smalltalk?

# What is object-oriented programming?

Encapsulation

Access control

Polymorphism

Inheritance

Message passing

Classes / prototypes / objects / instances

Methods

Virtual dispatch / late binding

# What is object-oriented programming?

Encapsulation

Access control

Polymorphism

Inheritance

**Message passing** ←

Classes / prototypes / objects / instances

Methods

Virtual dispatch / late binding

# Message passing



# Message passing

Late binding

*Everything* is an object

Ask objects to perform high-level goals

“wondered why anyone would want to divide [the computer] up into weaker things called data structures and procedures”

“Why not divide it up into little computers...thousands of them, each simulating a useful structure?”

# Late-bound syntax

```
temporary variable  
↓  
instance variables  
↓  
to Pair b | h t      "b is temp. h, t are internal instance vars"  
  (ISNEW » (:h. :t)  "cons-if no explicit return is given, SELF is returned"  
  □hd    » (<- » (^:h)^h)    "replaca and car"  
  □tl    » (<- » (:t)^t)    "replacd and cdr"  
  □isPair » (^true)  
  □print » ('(print. SELF mprint)  
  □mprint » (h print. t isNil » ('print t isPair » (t mprint) »  
           '•print. t print.' print)  
  □length » (t isPair » (^1+t length 1))
```

**to** like LOGO, except makes a class from its message

**ISNEW** is true if a new instance has been created

**true** any object not false acts as true

**true** » m n will evaluate m and escape from surrounding ()

**false** » m n will evaluate n

: evals the next part of message and binds result to the variable in its message

☒ eyeball looks to see if its message is a literal token in the message stream

^ send-back returns its value to sender

. “statement separator” value is following message



# Final thoughts

# Distributed systems

# **Object-oriented → Distributed systems**

“Thousands of computers” connected by a “very fast network”

Tempting to reuse OO principles when designing distributed systems

But the network is unreliable!!!

OO has no concept of a “dropped message”



Roving bands of agents

# Roving bands of agents

Why does everyone always suggest this?

Objects will somehow discover the interface of the system when it arrives

Security threats?

What really happens?

A wide-angle photograph of a vast body of water during sunset or sunrise. The sky is a gradient of warm orange and yellow at the top, transitioning to a cooler blue and purple towards the horizon. The water's surface is calm with subtle ripples and reflections of the sky. In the bottom left corner, a single dark-colored duck stands on a dark, narrow strip of land or a concrete ledge that extends into the water. The overall atmosphere is serene and minimalist.

Popularity

# **Popularity**

“the most widely used system that claims to be object-oriented”

# Popularity (as of 1992)

Smalltalk-80	1980-1981
Lisp (Flavors)	1982
C++	1983 (g++ 1987, Borland C++ 1990, MS C++ 1992)
Objective-C	1986-1988
Lisp (CLOS)	1988
Turbo Pascal	1989 (5.5)
Python	1991 (0.9.0), 1994 (1.0.0)
Perl	1993 (5)
Java	1993 (1.0)
Ruby	1995 (0.95), 1996 (1.0)
OCaml	1996

A photograph of a young child standing in the middle of a dark, textured tunnel. The child is silhouetted, facing towards the bright light at the far end of the tunnel. The walls of the tunnel are rough and dark, creating a dramatic contrast with the bright light.

What's next?