

Programmer,  
c'est un peu comme jouer aux LEGO<sup>®</sup> :  
dimensions non-verbales de la programmation

*Simone Martini*

*Alma mater studiorum* • Università di Bologna  
and  
INRIA FoCUS – Sophia / Bologna

Paris, 18 novembre 2021  
UNA Europa : Enjeux sociétaux de l'intelligence artificielle



ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA  
DIPARTIMENTO DI INFORMATICA - SCIENZA E INGEGNERIA

*Inria*  
Informatiques mathématiques

# Une dyscrasie

*"Swipe" et "tap" pour nos portables*



[Photo : ©Minuum]

*Du texte pour nos programmes*

```
def QS(L):  
    if L==[]: return L  
    pivot = L[0]  
    return QS([x for x in L[1:] if x<lt pivot])  
        + [pivot] +  
        QS([x for x in L[1:] if x \ge pivot])
```

# Une dyscrasie

*"Swipe" et "tap" pour nos portables*



*Du texte pour nos programmes*

```
def QS(L):  
    if L==[]: return L  
    pivot = L[0]  
    return QS([x for x in L[1:] if x<lt pivot])  
        + [pivot] +  
        QS([x for x in L[1:] if x \ge pivot])
```

[Photo : ©Minuum]



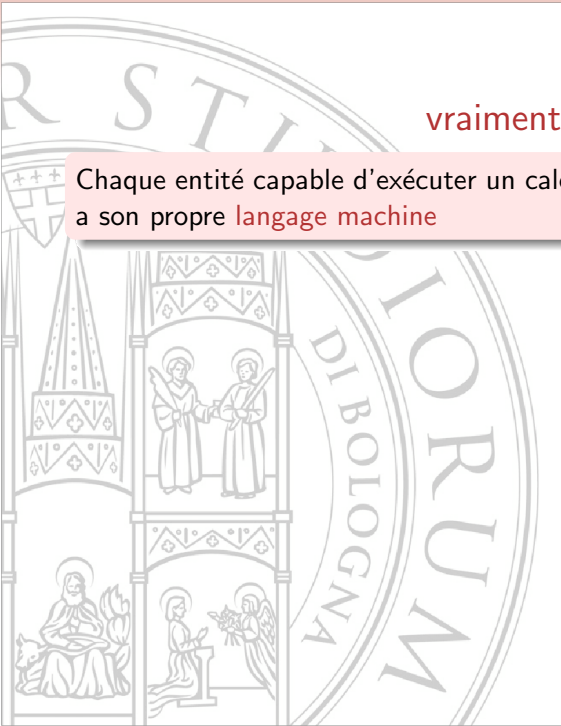
Sed contra :

Utiliser un portable, c'est différent que programmer un ordinateur

Il s'agit de deux activités qui ne sont pas comparables

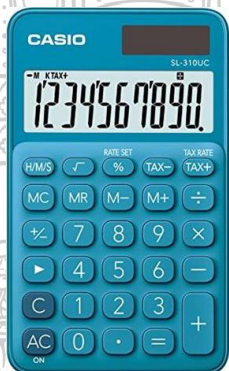
## Sont-elles vraiment incomparables ?

Chaque entité capable d'exécuter un calcul (une *machine abstraite*)  
à son propre langage machine



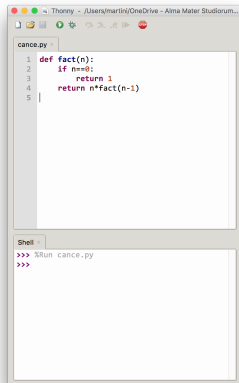
## Sont-elles vraiment incomparables ?

Chaque entité capable d'exécuter un calcul (une *machine abstraite*)  
a son propre langage machine



# Sont-elles vraiment incomparables ?

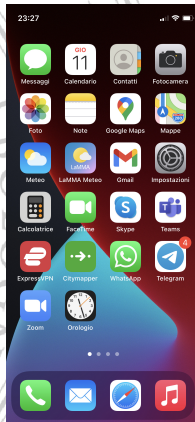
Chaque entité capable d'exécuter un calcul (une *machine abstraite*)  
a son propre langage machine



```
Thony - /Users/martin/OneDrive - Alma Mater Studiorum...
cance.py
1 def fact(n):
2   if n==0:
3     return 1
4   return n*fact(n-1)
5
Shell
>>> %run cance.py
>>>
```

# Sont-elles vraiment incomparables ?

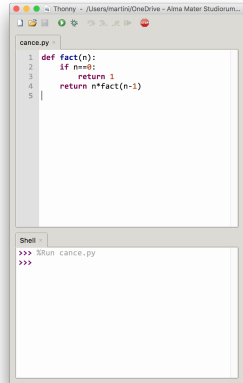
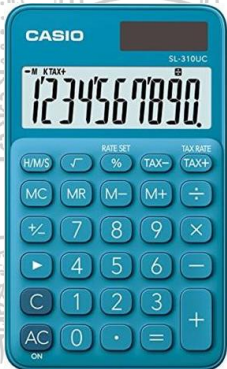
Chaque entité capable d'exécuter un calcul (une *machine abstraite*)  
a son propre langage machine





# Sont-elles vraiment incomparables ?

Chaque entité capable d'exécuter un calcul (une *machine abstraite*)  
a son propre langage machine



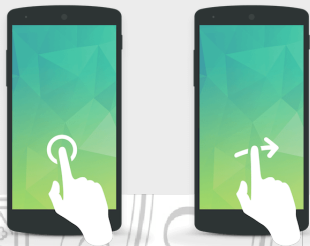
# Langages de programmation

Quand un langage machine est “suffisamment puissant”  
(= Turing-complet)  
on l’appelle **langage de programmation**

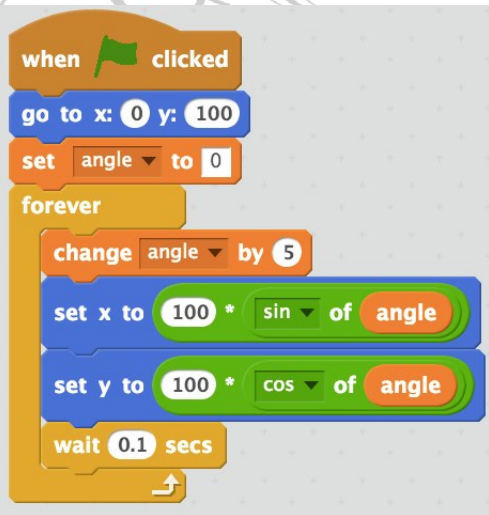
Mais le cadre conceptuel est le même :  
une machine abstraite avec son langage

## Langages visuels, de gestes

Est-il possible un vrai langage de programmation *visuel, de gestes*?

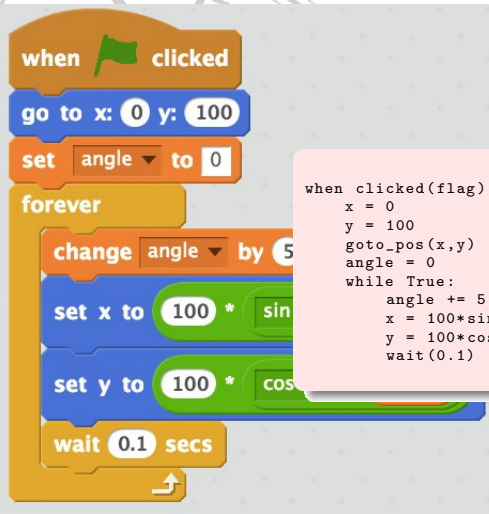


# Scratch



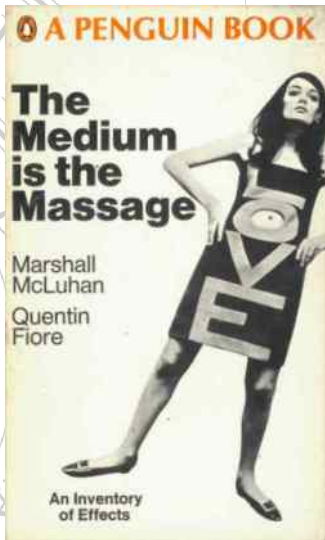
A project of the Lifelong Kindergarten Group at the MIT Media Lab

# Scratch



```
when clicked(flag):  
  x = 0  
  y = 100  
  goto_pos(x,y)  
  angle = 0  
  while True:  
    angle += 5  
    x = 100*sin(angle)  
    y = 100*cos(angle)  
    wait(0.1)
```

Le medium est le message



# La métaphore linguistique

## **When Technology Became Language: The Origins of the Linguistic Conception of Computer Programming, 1950-1960**

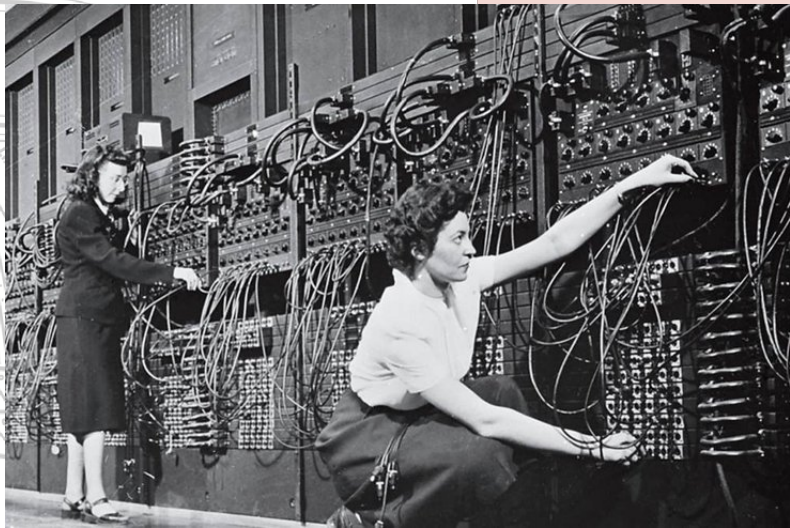
David Nofre, Mark Priestley, Gerard Alberts

Technology and Culture, Volume 55, Number 1, January 2014, pp. 40-75 (Article)

Published by The Johns Hopkins University Press

DOI: [10.1353/tech.2014.0031](https://doi.org/10.1353/tech.2014.0031)





*La programmation de l'ENIAC : câbles (1945-46)*



# Langages de programmation

- outil technique
- objet d'étude
- meta-langages :  
algorithmes publiés en Algol dans les *Communications of ACM*, années 50s-60s

ALGORITHM 64  
QUICKSORT

C. A. R. HOARE

Elliott Brothers Ltd., Borehamwood, Hertfordshire, Eng.

```
procedure quicksort (A,M,N); value M,N;  
          array A; integer M,N;
```

**comment** Quicksort is a very fast and convenient method of sorting an array in the random-access store of a computer. The entire contents of the store may be sorted, since no extra space is required. The average number of comparisons made is  $2(M-N) \ln(N-M)$ , and the average number of exchanges is one sixth this amount. Suitable refinements of this method will be desirable for its implementation on any actual computer;

```
begin      integer I,J;  
          if M < N then begin partition (A,M,N,I,J);  
              quicksort (A,M,J);  
              quicksort (A, I, N)  
          end  
end      quicksort
```

# Utiliser Scratch

The screenshot displays the Scratch programming interface. On the left, the 'Blocks' palette is visible, categorized into Motion, Looks, Sound, Events, Control, Sensing, Operators, Variables, and My Blocks. The central workspace contains a script for a cat sprite:

```
when green flag clicked
  forever loop
    move 4 steps
    if x position > 200 then
      set x to -180
```

On the right, the stage shows a cat sprite flying over a landscape with a volcano and colorful plants. Below the stage, the 'Sprite' panel shows 'Cat1 Flying' with a size of 100 and a direction of 90. The 'Stage' panel shows 'Backdrops' with a count of 2.

Des briques, et non des phrases dans un langage. . .

# Utiliser Scratch 1 : concret

## Un outil concret

- on joue avec les éléments, comme au LEGO
- pas de règles de grammaire :  
les briques s'encastrent, ou ne s'encastrent pas
- les briques s'encastrent seulement en des façons sensées :  
pas d'erreurs syntaxiques

## Utiliser Scratch 2 : en direct

### Un outil "live"

- le système réponds en continu
- chaque action sur les briques a une conséquence immédiate
- pas de cycle *code-compile-run*  
(coder-compiler-exécuter) ( ?? )

## Utiliser Scratch 3 : bricolable

### Un outil “tinkerable”

- “Il permet d’expérimenter avec des commandes et des extraits de code comme on pourrait bricoler des composants mécaniques ou électroniques”
- “Les formes des briques suggèrent ce qui est possible”
- “L’expérimentation et l’expérience enseignent ce qui fonctionne”

## Utiliser Scratch :

- concret (visuel)
- live
- bricolable

## Utiliser Scratch :

- concret (visuel)
- live
- bricolable

*Tous éléments présents aussi dans le langage "tap and swipe" de nos portables*



## Utiliser Scratch :

- concret (visuel)
- live
- bricolable

*On peut tracer ces éléments dans l'histoire de langages de programmation. . .*

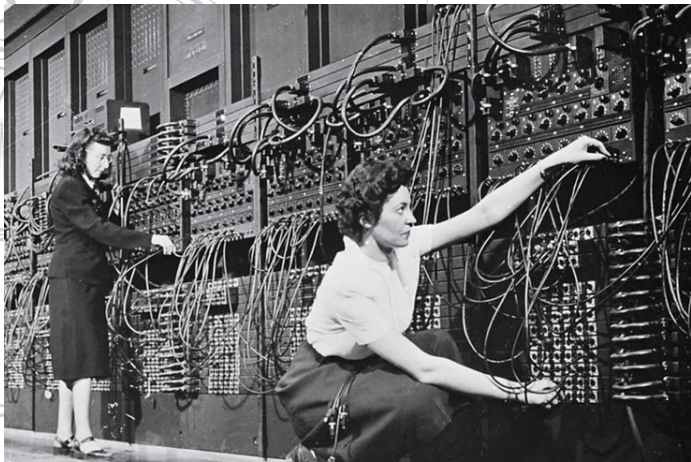
## Utiliser Scratch :

- concret (visuel)
- live
- bricolable

*On peut tracer ces éléments dans l'histoire de langages de programmation...*

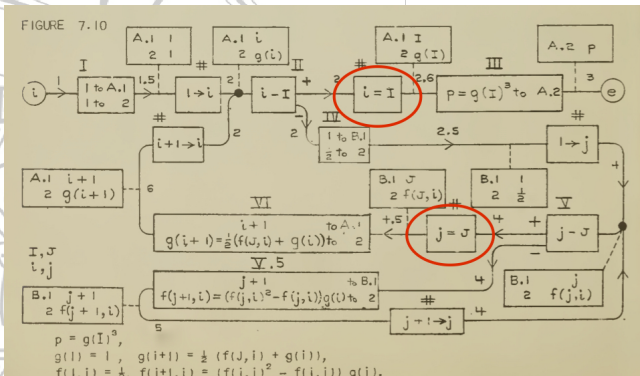
*Mais la chose importante est leur présence ensemble...*

## Concret - Visuel : ENIAC



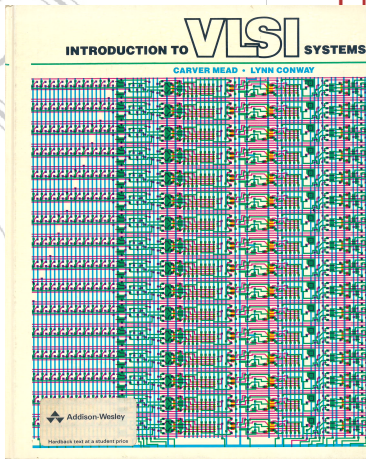
ENIAC : 1945-46

# Concret - Visuel : Organigrammes



Flow diagrams, Goldstine and von Neumann : 1947

Concret - Visuel :  
Projet VLSI



*Intégration à très grande échelle selon Mead et Conway, 1977ff ;  
livre ©1980*

# Concret - Visuel : Projet VLSI

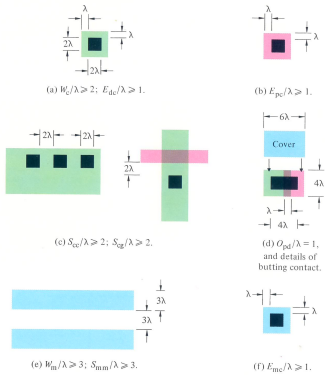
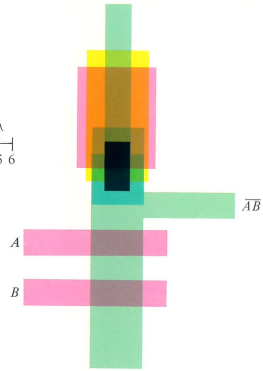


PLATE 3 nMOS design rules (continued)



(a) NAND gate layout geometry.

Mead and Conway, 1977ff; livre ©1980

# Mead and Conway's VLSI

Comme Scratch pour la programmation :

- projet de circuits intégrés possible pour gens sans aucune formation en électronique/semi-conducteurs
- des règles graphiques de composition
- qui assurent la correction du projet

## Concret - Visuel : langages des années 80s

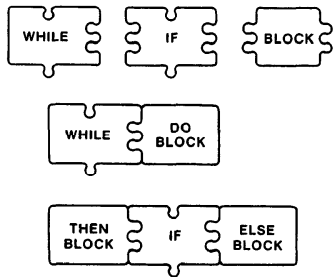


Figure 1: Possible Realizations of Some Imperative Procedural Constructs Using Tiles.

*BLOX, 1987; E. P. Glinert*





Live :  
Maloney and Smith, 1995

**Directness and Liveness in the Morphic  
User Interface Construction Environment**

*John H. Maloney*  
Apple Computer, Inc.  
1 Infinite Loop, M/S 301-3E  
Cupertino, CA 95014 USA  
+1-408-974-7293  
J.Maloney@eWorld.com

*Randall B. Smith*  
Sun Microsystems Laboratories  
2550 Garcia Avenue, MTV 29-116  
Mountain View, CA 94043 USA  
+1-415-336-2620  
Randall.Smith@Sun.com



Live :  
Maloney and Smith, 1995

**Directness and Liveness in the Morphic  
User Interface Construction Environment**

*John H. Maloney*  
Apple Computer, Inc.  
1 Infinite Loop, M/S 301-3E  
Cupertino, CA 95014 USA  
+1-408-974-7293  
J.Maloney@eWorld.com

*Randall B. Smith*  
Sun Microsystems Laboratories  
2550 Garcia Avenue, MTV 29-116  
Mountain View, CA 94043 USA  
+1-415-336-2620  
Randall.Smith@Sun.com

Liveness signifie que l'interface est toujours active et réactive :

- les objets répondent aux actions des utilisateurs
- les animations sont exécutées
- la mise en page se produit
- les affichages sont mis à jour en continu

# Live : Maloney and Smith, 1995

## Directness and Liveness in the Morphic User Interface Construction Environment

*John H. Maloney*  
Apple Computer, Inc.  
1 Infinite Loop, M/S 301-3E  
Cupertino, CA 95014 USA  
+1-408-974-7293  
J.Maloney@eWorld.com

*Randall B. Smith*  
Sun Microsystems Laboratories  
2550 Garcia Avenue, MTV 29-116  
Mountain View, CA 94043 USA  
+1-415-336-2620  
Randall.Smith@Sun.com

Liveness signifie que l'interface est toujours active et réactive :

- les *objets* répondent aux actions des utilisateurs



## Une longue histoire

Programmer c'est interagir avec l'exécuteur, la machine

- Logo, 1969 :

W. Feurzeig and S. Papert. Programming languages as a conceptual framework for teaching mathematics. Final report on the first fifteen months of the Logo Project. TR 1889. BBN, Cambridge, MA.

- Smalltalk, 1972

Alan Key, XEROX PARC

L'interaction est exprimée avec des métaphores puissantes :

la tortue

les objets

## Une longue histoire

Programmer c'est interagir avec l'exécuteur, la machine

- Logo, 1969 :

W. Feurzeig and S. Papert. Programming languages as a conceptual framework for teaching mathematics. Final report on the first fifteen months of the Logo Project. TR 1889. BBN, Cambridge, MA.

- Smalltalk, 1972

Alan Key, XEROX PARC

L'interaction est exprimée avec des métaphores puissantes :

la tortue

les objets

## La tortue



## Bricolable

*Il faut faire une petite pause,  
pour expliquer pourquoi c'est important*

## Bricolable : le constructionnisme

*Per conoscere il mondo bisogna costruirlo*

*Cesare Pavese, Il mestiere di vivere (français : Le métier de vivre). 1952*

*Pour connaître le monde, il faut le construire*

*(il faut le construire dans une histoire – ou **comme** une histoire)*

*In other words, we make not just to have, but to know.*

*But the having can happen without most of the knowing taking place.*

*Alan Kay, The early history of Smalltalk. 1993*

*En d'autres termes, nous ne faisons (construisons) pas seulement pour avoir, mais pour connaître.*

*Mais l'avoir peut se produire sans que le connaître ait lieu.*



## Bricolable : le constructionnisme

*Per conoscere il mondo bisogna costruirlo*

*Cesare Pavese, Il mestiere di vivere (français : Le métier de vivre). 1952*

*Pour connaître le monde, il faut le construire*

(il faut le construire dans une histoire – ou **comme** une histoire)

*In other words, we make not just to have, but to know.*

*But the having can happen without most of the knowing taking place.*

*Alan Kay, The early history of Smalltalk. 1993*

*En d'autres termes, nous ne faisons (construisons) pas seulement pour avoir, mais pour connaître.*

*Mais l'avoir peut se produire sans que le connaître ait lieu.*

## Le problème :

Si "l'avoir" peut se produire sans que "le savoir" ait lieu, est-ce qu'il y a des conditions dans lesquelles "le faire" produit "le connaitre" et pas seulement "l'avoir" ?

Faire pour connaître :

Seymour Papert :

il faut construire des objets et des relations **significatives**

## Le problème :

Si "l'avoir" peut se produire sans que "le savoir" ait lieu, est-ce qu'il y a des conditions dans lesquelles "le faire" produit "le connaître" et pas seulement "l'avoir" ?

**Faire pour connaître :**

Seymour Papert :

il faut construire des objets et des relations **significatives**

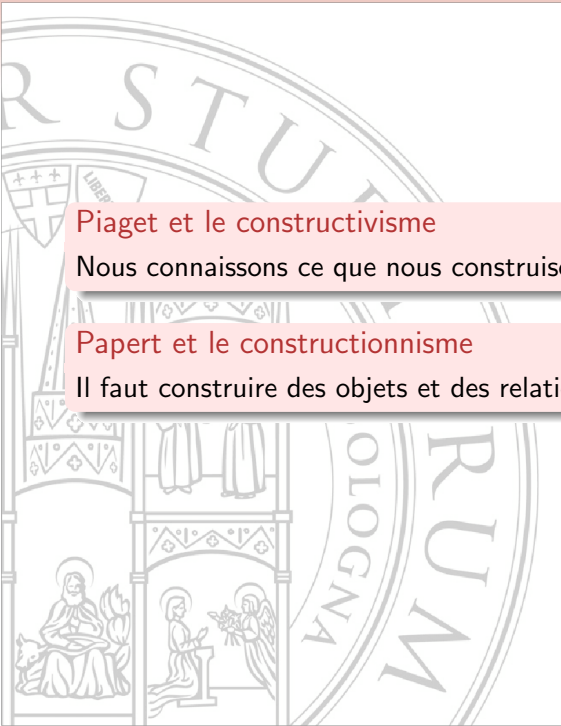


## Piaget et le constructivisme

Nous connaissons ce que nous construisons

## Papert et le constructionnisme

Il faut construire des objets et des relations significatives



## Piaget et le constructivisme

Nous connaissons ce que nous construisons

## Papert et le constructionnisme

Il faut construire des objets et des relations **significatives**

## Papert et le constructionnisme

constructivisme + signification

On construit des versions concrètes de concepts abstraits  
et on entre en relation avec ces objets concrets



L'utilisation d'un langage de programmation est l'un des moyens les plus efficaces et économiques pour les enfants d'obtenir de tels modèles de manière autonome.

**Mais :**

La modalité d'interaction avec le support informatique est aussi (et probablement **plus**) importante que son contenu

Le cycle "essai et correction" :

Obtenir un retour des objets computationnels.

## Smalltalk, au début

### Alan Kay :

- 1966-1969 : étudiant de master, University of Utah
- été 1967 : il apprend les idées de Papert des Marvin Minsky
- hiver 1968 : il se rencontre avec Papert et son group

*Cette rencontre m'a finalement frappé avec ce que le destin du **personal computing** allait vraiment être :[...] un **support dynamique** personnel [qui] devait s'étendre dans le monde de l'enfance.*

*un **personal computer** : Il ne devait pas être plus gros qu'un cahier et avec une interface conviviale, mais avec la puissance d'un vrai langage de programmation.*



## Smalltalk, au début

### Alan Kay :

- 1966-1969 : étudiant de master, University of Utah
- été 1967 : il apprend les idées de Papert des Marvin Minsky
- hiver 1968 : il se rencontre avec Papert et son group

*Cette rencontre m'a finalement frappé avec ce que le destin du **personal computing** allait vraiment être :[...] un **support dynamique** personnel [qui] devait s'étendre dans le monde de l'enfance.*

*un **personal computer** : Il ne devait pas être plus gros qu'un cahier et avec une interface conviviale, mais avec la puissance d'un vrai langage de programmation.*

## Smalltalk, au début

### Alan Kay :

- 1966-1969 : étudiant de master, University of Utah
- été 1967 : il apprend les idées de Papert des Marvin Minsky
- hiver 1968 : il se rencontre avec Papert et son group

*Cette rencontre m'a finalement frappé avec ce que le destin du **personal computing** allait vraiment être :[...] un **support dynamique** personnel [qui] devait s'étendre dans le monde de l'enfance.*

*un **personal computer** : Il ne devait pas être plus gros qu'un cahier et avec une interface conviviale, mais avec la puissance d'un vrai langage de programmation.*

## Smalltalk

*Il ne suffit pas d'apprendre à lire et à écrire. Il y a aussi une littérature. À un moment donné, l'organisation des idées commence à dominer les simples capacités linguistiques.*

*Et ça aide beaucoup d'avoir des idées fortes à son actif pour mieux acquérir des idées plus fortes [Papert 70s]. Nous avons donc décidé d'enseigner le design.*

*And it helps greatly to have some powerful ideas under one's belt to better acquire more powerful ideas [Papert 70s]. So, we decided we should teach design.*

## Smalltalk

*Des objets et classes des langages de programmation  
à un écosystème d'objets en interaction*

# Smalltalk for Alto, fin 1970s

**System Browser**

Collections-Sequence	Interval	accessing	collect:
Collections-Text	LinkedList	do:	do:andBetweenDo:
Collections-Array	MappedCollection	adding	promoteFirstSuchT
Collections-Stream	OrderedCollection	removing	reverse
Collections-Support	SortedCollection	enumerating	reverseDo:
Graphics-Primitives	-----	private	select:
Graphics-Display	-----	-----	-----
Graphics-Media	-----	-----	-----
Graphics-Paths	instance	class	-----

**collect: aBlock**

*"Evaluate aBlock with each of my elements as the argument. C  
resulting values into a collection that is like me. Answer with  
collection. Override superclass in order to use add:, not at:put:*

```
] newCollection |  
newCollection + self species new.  
self do: [:each | newCollection add: (aBlock value: each)].  
newCollection
```

**User Interrupt**

```
Paragraph>>characterBlockAtPoint:  
Paragraph>>mouseSelect:to:  
CodeController(ParagraphEditor)>>processRedButton  
CodeController(ParagraphEditor)>>processMouseButtons  
CodeController(ParagraphEditor)>>controlActivity  
CodeController(Controller)>>controlLoop
```

**controlActivity**

```
self scrollBarContainsCursor  
ifTrue:  
[self scroll]  
ifFalse:  
[self processKeybo  
self processMouse
```

**File List**

```
[File] (Robson)SP*  
[File] (Robson)SP>ScreenForm.txt  
[File] (Robson)SP>ScreenForm.txt  
[File] (Robson)SP>ScreenFormChanges.st  
[File] (Robson)SP>WordGraphics.form  
-----  
Rectangle fromUser origin  
ScreenForm setFullPageWidth.  
ScreenForm  
printRectangle:  
(0045 extent: 674@790)  
onFullNamed: 't: ampaScreen press'  
(Form readFrom: 'FiledSkate.form') edit
```



### Des objets et classes de langages de programmation à un écosystème d'objets en interaction

Smalltalk n'est PAS seulement sa syntaxe.  
Je suis désolé d'avoir inventé il y a longtemps le  
terme "objet", car il amène de nombreuses personnes  
à se concentrer sur une idée qui n'est pas  
importante.  
La grande idée est "échanger des messages" [...]  
Les Japonais ont un petit mot -- ma -- pour "ce qui  
est entre les deux"

*A. Kay, message to the Squeak-dev mailing list. Sat Oct 10 1998*

## Smalltalk

Des objets et classes de langages de programmation  
à un écosystème d'objets en interaction

Smalltalk n'est PAS seulement sa syntaxe.  
Je suis désolé d'avoir inventé il y a longtemps le  
terme "objet", car il amène de nombreuses personnes  
à se concentrer sur une idée qui n'est pas  
importante.

La grande idée est "échanger des messages" [...]  
Les Japonais ont un petit mot -- ma -- pour "ce qui  
est entre les deux"

*A. Kay, message to the Squeak-dev mailing list. Sat Oct 10 1998*

## Smalltalk

Et cela se reflète dans Smalltalk lui-même :

quand ST a été connu hors de notre labo, c'était à peu près considéré comme "quelque chose à apprendre", comme s'il s'agissait de Pascal ou d'Algol.

alors que c'est quelque chose qu'il faut bricoler :

dans notre labo, nous avons constamment changé Smalltalk, en le traitant toujours comme un travail en progrès



## Smalltalk

Et cela se reflète dans Smalltalk lui-même :

quand ST a été connu hors de notre labo, c'était à peu près considéré comme "quelque chose à apprendre", comme s'il s'agissait de Pascal ou d'Algol.

alors que c'est quelque chose qu'il faut bricoler :

dans notre labo, nous avons constamment changé Smalltalk, en le traitant toujours comme un travail en progrès

## Smalltalk

Et cela se reflète dans Smalltalk lui-même :

quand ST a été connu hors de notre labo, c'était à peu près considéré comme "quelque chose à apprendre", comme s'il s'agissait de Pascal ou d'Algol.

alors que c'est quelque chose qu'il faut bricoler :

dans notre labo, nous avons constamment changé Smalltalk, en le traitant toujours comme un travail en progrès

## Conclusions

La programmation dans les langages visuels, et l'interaction avec nos moyens de calcul, exploite de moins en moins la métaphore linguistique

Ce point de vue a des racines anciennes et bien établies, même dans le contexte linguistique classique



RICHARD SENNETT

CE QUE SAIT  
**LA MAIN**

LA CULTURE  
DE L'ARTISANAT

ALBIN MICHEL

RICHARD SENNETT

# CE QUE SAIT **LA MAIN**

LA CULTURE  
DE L'ARTISANAT

ALBIN MICHEL

*Ce que nous pouvons dire avec des mots peut être plus limité de ce que nous pouvons faire avec des choses. [...]*

*Voici une, peut-être la limite humaine fondamentale : le langage n'est pas un "outil-miroir" adéquat pour les mouvements physiques du corps humain.*

*[R. Sennett, The Craftsman. 2009]*

## Une observation ancienne

Janvier 1751.

**ENCYCLOPÉDIE,**  
OU  
DICTIONNAIRE RAISONNÉ  
DES SCIENCES,  
DES ARTS ET DES MÉTIERS,  
RECUEILLI  
DES MEILLEURS AUTEURS  
ET PARTICULIÈREMENT  
DES DICTIONNAIRES ANGLAIS  
DE CHAMBERS, D'HARRIS, DE DYCHE, &c.  
PAR UNE SOCIÉTÉ DE GENS DE LETTRES.

Mis en ordre & publié par M. DIDEROT, & quant à la PARTIE MATHÉMATIQUE,  
par M. D'ALEMBERT, de l'Académie Royale des Sciences de Paris  
& de l'Académie Royale de Berlin.

*Tantum series juncturae polles,  
Tantum de medio sumptis accedit honoris! HORAT.*

DIX VOLUMES IN-FOLIO,  
DONT DEUX DE PLANCHES EN TAILLE-DOUCE,

---

PROPOSÉS PAR SOUSCRIPTION.



A PARIS, Chez

- BRIASSON, rue Saint Jacques, à la Science.
- DAVID l'ainé, rue Saint Jacques, à la Plume d'Or.
- LE BRETON, Imprimeur ordinaire du Roy, rue de la Harpe.
- DURAND, rue Saint Jacques, à Saint Landry, & au Griffon.

M. DCC. LI.

AVEC APPROBATION ET PRIVILEGE DU ROY.

## Une observation ancienne

Janvier 1751.  
**ENCYCLOPÉDIE,**  
OU  
DICTIONNAIRE RAISONNÉ  
DES SCIENCES,  
DES ARTS ET DES MÉTIERS,  
RECUEILLI  
DES MEILLEURS AUTEURS.

ET PARTICULIÈREMENT  
DES DICTIONNAIRES ANGLAIS  
DE CHAMBERS, D'HARRIS, DE  
PAR UNE SOCIÉTÉ DE GENS DE LETTRES

Mis en ordre & publié par M. DIDEROT; & quant à la PARTIE  
par M. D'ALEMBERT, de l'Académie Royale des Sciences  
& de l'Académie Royale de Berlin.

*Tantum series juncturae polles,  
Tantum de medio sumptis accedit honoris!* HORAT.

DIX VOLUMES INFOLIO  
DONT DEUX DE PLANCHES EN TAILLE-DOUBLE

PROPOSÉS PAR SOUSCRIPTION



A PARIS, Chez  
BRIASSON, rue Saint Jacques, à la Science.  
DAVID l'Écuyer, rue Saint Jacques, à la Plume d'Or.  
LE BRETON, Imprimeur ordinaire du Roy, rue de la Harpe.  
DURAND, rue Saint Jacques, à Saint Landry, & au Griffon.

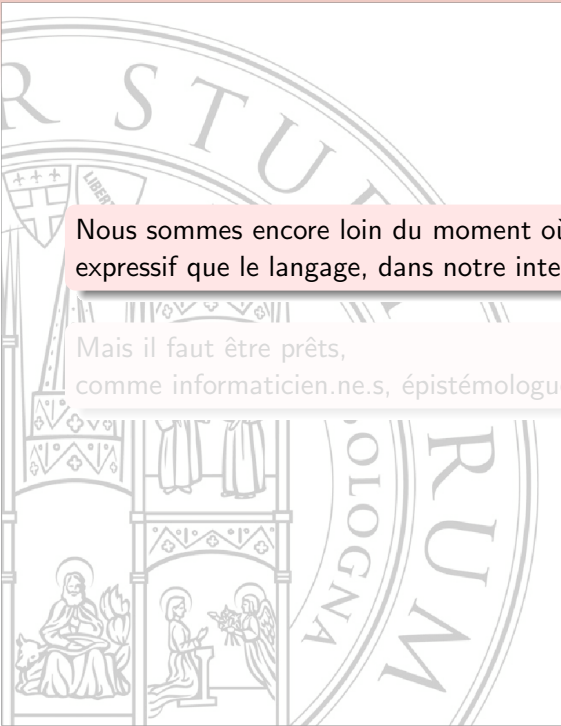
M. DCC. LI.

AVEC APPROBATION ET PRIVILEGE DU ROY.

*On s'est adressé aux plus habiles de Paris et du royaume. On s'est donné la peine d'aller dans leurs ateliers [...]*

*À peine, entre mille, en trouve-t-on une douzaine en état de s'exprimer avec quelque clarté sur les instruments qu'ils emploient et sur les ouvrages qu'ils fabriquent.*

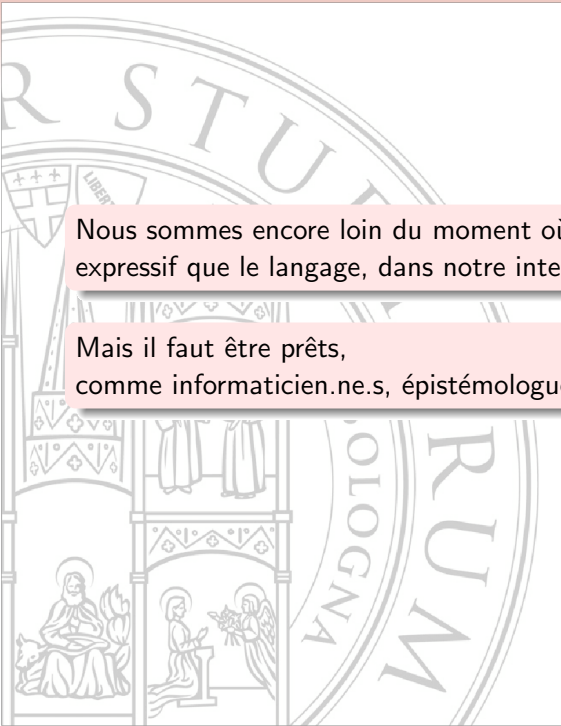
[D. Diderot, Prospectus à l'Encyclopédie, 141 ; 1751.]



Nous sommes encore loin du moment où le geste sera plus expressif que le langage, dans notre interaction avec les ordinateurs

Mais il faut être prêts,  
comme informaticien.ne.s, épistémologues, gens de culture, ...





Nous sommes encore loin du moment où le geste sera plus expressif que le langage, dans notre interaction avec les ordinateurs

Mais il faut être prêts,  
comme informaticien.ne.s, épistémologues, gens de culture, ...