

# Aprenentatge de la Programació al Citilab: Squeak i Scratch

Jordi Delgado

Dept. Llenguatges i Sistemes Informàtics (UPC) & Citilab - Cornellà  
jdelgado@lsi.upc.edu

Joan Güell

Citilab - Cornellà  
jguell@e-citilab.eu

maig de 2008

## Resum

Aquest article vol ser un recull de les primeres experiències al Citilab apropiant la programació dels ordinadors a la gent del carrer. Sense cap requeriment previ, qualsevol pot inscriure's a algun dels cursos de la nostra oferta docent en programació. Això ha estat possible principalment gràcies a l'existència de programari adequat, tot ell realitzat en entorns Smalltalk: Scratch per a nens petits (i no tan petits!), Bot-  
sInc com a iniciació a l'Smalltalk i el mateix Squeak, implementació *open source* d'Smalltalk utilitzada en la creació del programari esmentat, per a programadors experimentats.

## 1 Introducció

L'ensenyament de la programació a la gent jove (nens i adolescents), fora dels itineraris acadèmics estàndard, és una qüestió la viabilitat de la qual no està gens clara. Projectes per ensenyar a programar a tothom han existit des de fa més de 30 anys (nens [13], adolescents [4], estudiants universitaris en general [6]) i tot i així no sembla haver una manera clara i consensuada de fer-ho. Un dels projectes bàsics del Citilab, centre d'innovació tecnològica creat recentment a Cornellà, és la difusió i divulgació de les possibilitats que les tecnologies de la informació posen en mans de la gent del carrer, de tothom. En aquest marc, el Citilab ha endegat una proposta de formació tecnològica que va des dels rudiments de l'utilització d'Internet fins a la creació de xarxes socials amb tecnologia *web 2.0*, passant per l'aprenentatge de la programació d'ordinadors.

Aquest article vol explicar l'experiència del Citilab ensenyant a programar en aquests primers sis mesos d'existència del projecte. A la primera

secció fem una petita i forçadament incompleta reflexió sobre el sentit del nostre projecte i després expliquem els tres blocs en que hem dividit la nostra proposta: Scratch (<http://scratch.mit.edu>) per a nens petits, l'entorn BotsInc (<http://smallwiki.unibe.ch/botsinc>) com a iniciació, partint de zero, a la programació i Squeak (<http://www.squeak.org>), implementació *open source* del llenguatge de programació Smalltalk, per ensenyar programadors experimentats una nova (tot i que existeix des de fa més de 30 anys!) manera de desenvolupar programari.

## 2 Què és el Citilab?

El Citilab (figura 1) és un centre experimental de convergència entre la nova generació d'Internet i la nova generació de projectes de la Societat del Coneixement. Un espai orientat a activar, impulsar i extendre la capacitat creativa i innovadora en tecnologia d'emprenedors, empreses i ciutadans i ciutadanes en la Societat del Coneixement.



Figura 1: El citilab: laboratori d'innovació tecnològica

La hipòtesi que hi ha al darrera del Citilab és que gràcies a la tecnologia i en particular a Internet, dins la societat del coneixement qualsevol persona pot accedir als instruments per crear coneixement en un context d'innovació.

Les idees centrals són:

- Els ciutadans tenen capacitat per ser innovadors.

- La col·laboració és la base de la innovació social
- Les noves tecnologies creen una nova cultura d'innovació diferent de la de l'època industrial.

### 3 Per què volem ensenyar a programar a tothom?

*Educators, generals, dieticians, psychologists, and parents program.  
Armies, students and some societies are programmed.*

Alan Perlis [2]

Les virtuts d'aprendre a programar com a complement de la formació general de qualsevol persona són, ara per ara, qüestió de debat. Hi ha detractors, promotores i, és clar, indiferents. Tots els que tenim certa edat vam haver de sentir moltes raons per aprendre llatí a segon de Batxillerat, curiosament totes elles són aplicables per poder justificar cursos d'introducció a la programació d'ordinadors: crear i/o incentivar la disciplina en el racionament, sistematitzar l'hàbit de treball, fomentar la perseverància i fer-nos una mica més humils (aquesta no es mencionava en relació al llatí, però aquí és pertinent, qui de nosaltres no ha pensat alguna vegada que era el compilador el que no funcionava bé?).

N'hi ha que proposen raons una mica més profundes. Nosaltres no ho podríem expressar millor que Abelson i Sussman:

*Underlying our approach to this subject is our conviction that “computer science” is not a science and that its significance has little to do with computers. The computer revolution is a revolution in the way we think and in the way we express what we think. The essence of this change is the emergence of what might best be called procedural epistemology – the study of the structure of knowledge from an imperative point of view, as opposed to the more declarative point of view taken by classical mathematical subjects. Mathematics provides a framework for dealing precisely with notions of “what is.” Computation provides a framework for dealing precisely with notions of “how to.”*

H.Abelson & G.J.Sussman [2]

No només ho proposen, sinó que aquest nou tipus d'epistemologia es posada en pràctica en l'ensenyament, per exemple, de la Mecànica Clàssica [3] al MIT. Sigui com sigui, en aquests temps de creixement de supersticions, fanatismes i irracionalitat, pretendre que la gent pensi més en el “*how to*” pot ajudar a fomentar aquest escepticisme crític que tanta falta fa.

Finalment, sembla clar que el debat continuarà i que nosaltres no el resoldrem aquí, per manca d'espai i perque no és aquest el propòsit d'aquest article. Per això, un cop donades algunes raons, més o menys convincentes, que justifiquen la nostra tasca, anem a detallar com l'hem duta a terme.

## 4 Scratch

Scratch és un entorn d'aprenentatge desenvolupat pel *Lifelong Kindergarten Group* al *MIT Media Lab* en Smalltalk, utilitzant Squeak. És 100% programari lliure, i pretén apropar la programació a tothom a partir de 8 anys. Actualment s'està utilitzant a tot el món, i les experiències amb Scratch es troben a tot arreu, en particular dins el congrés més important del món de *computer science education* vinculat al grup d'educació en informàtica de l'ACM (aquests darrers dos anys, per exemple, podem trobar [12, 14, 11]) fins al punt que aquest mateix any han dedicat una sessió especial a Scratch [15].

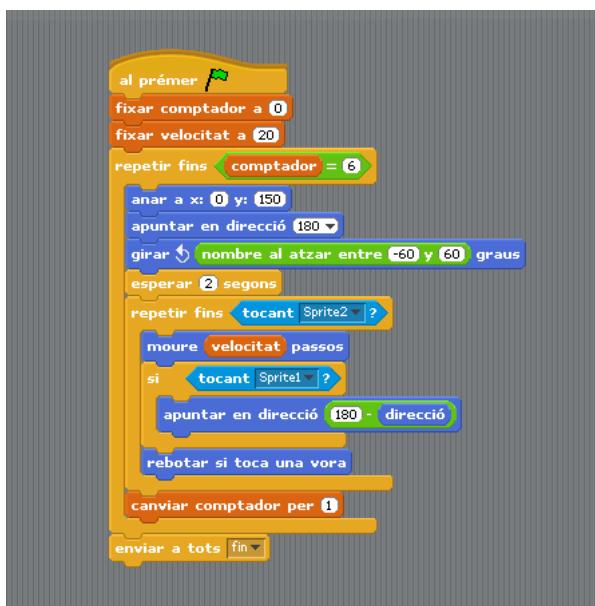


Figura 2: Exemple de programa en Scratch. Es veu clarament l'estrucció del programa.

El llenguatge de programació vinculat a l'entorn Scratch és un llenguatge imperatiu amb les construccions fonamentals (variables, assignació, bucles, condicionals i crida a funcions i accions) ampliades amb un grup molt nombrós d'instruccions per poder treballar en projectes multimèdia. Un projecte Scratch serà típicament un joc o una animació sofisticada amb dibujos, icones, so i altres gràfics en moviment (podeu fer-vos-en una idea visitant els projectes que seguidors d'Scratch deixen a la plana web esmentada més amunt). El que és més original d'Scratch és la metàfora que hi ha al darrera, que és ni més ni menys que la dels jocs de blocs, tipus Lego o l'antic Tente. Cada instrucció, o conjunt d'instruccions encastrades, es relaciona amb d'altres enganxant-se al costat si les succeeix o precedeix o dins si passa a formar part d'una estructura superior, com un bucle. Així anirem

constraint els nostres programes, peça a peça, ben bé com si estiguéssim fent un castell.

L'entorn de treball que tenim al Citilab consisteix en ordinadors portàtils amb l'Scratch instal·lat en cada un d'ells, per a que cada nen faci un seguiment individual de la classe (figura 3, on podeu veure en Joan fent una de les seves lliçons)



Figura 3: Nens i nenes fent els seus primers programes amb Scratch.

El curs ofert consisteix en començar a familiaritzar el nen amb l'entorn de treball per acabar amb un projecte senzill fet pel nen. Els primers projectes que els presentem tenen a veure amb l'utilització del programa (guardar i recuperar projectes, afegir, treure i modificar *sprites*, etc.) i la introducció a la geometria bidimensional de l'espai on farem viure els nostres ninotets i dibuixos. Un cop tenen més o menys clars aquests conceptes comencem a introduir les diverses nocions bàsiques: repeticions, condicionals, variables (pot sorprendre la introducció de les variables al final, però Scratch d'alguna manera ho promou). Fins ara s'han fet un parell de cursos amb nens al voltant de 7-9 anys, que actualment (abril 2008) estan dividits en funció del rendiment i la capacitat dels estudiants. Hi ha un grup que està tornant a veure els conceptes fonamentals, explicats d'una altra manera, i un altre grup està treballant amb projectes més avançats (un senzill "pong" o una variant fàcil del Sokoban).

El curs és assimilat bé pels nens que hem tingut fins ara a l'aula. Les dificultats principals que ens hem trobat tenen a veure poc amb la programació i molt amb qüestions de disciplina i (manca de) concentració per part

d'alguns dels nens. Cal mantenir la motivació molt alta durant tota la sessió i no fer sessions massa llargues. En altre cas, de seguida comencen a jugar amb l'entorn sense fer massa cas del que diem els professors. Això, però, pot veure's positivament, ja que fet i fet treballen/juguen amb Scratch, que és el que volem. És massa aviat, però, per dir res concloent sobre la capacitat de programar que els nens han aconseguit gràcies als nostres cursos. Suposem que dins de tres mesos, el dia de la presentació a les jornades, tindrem més experiència per comentar resultats una mica més fiables.

## 5 BotsInc

Un dels entorns o llenguatges més innovadors per ensenyar a programar nens i adolescents ha estat Logo [13, 1], on s'aprenien els conceptes fonamentals de la programació fent dibujos geomètrics donant ordres al que s'anomenava una “tortuga”. L'entorn BotsInc [7] programat en Smalltalk/Squeak pel professor Stéphane Ducasse (actualment a l'Inria francès) serveix per iniciar a la programació amb Smalltalk, en un entorn molt similar, tot i que més complet, al Logo original. Originalment fet per a la seva dona, professora d'informàtica d'institut, BotsInc és un entorn que permet aprendre a programar en un llenguatge essencialment imperatiu, subconjunt d'Smalltalk, però que ja deixa entreveure alguns conceptes d'orientació a objectes inherents a Smalltalk: Tot és un objecte, res no es fa si no és per pas de missatges, creació d'objectes a partir de classes, utilització de *browsers* de codi per afegir mètodes a la classe principal de l'entorn, anomenada Bot, etc.



Figura 4: La classe dels que volen iniciar-se a l'Smalltalk, amb BotsInc.

Les classes al Citilab han començat fa tres setmanes (principis d'abril) i

tenim uns onze estudiants, tots ells entre 16 i 19 anys (figura 4, on teniu el Jordi impartint una classe). De moment el seguiment és molt satisfactòri, tot i que en aquest cas és evident que és massa aviat per dir res de res. Altre cop, haurem d'esperar a la presentació dins les jornades per poder proporcionar informació sobre com ha anat el curs.

La traducció al català del llibre del professor Ducasse [7] és a punt d'acabar-se i s'espera la seva publicació per abans d'aquest estiu [8]. De moment, els estudiants estan fent servir un esborrany que el Citilab distribueix sense cost. Aquesta traducció ha estat necessària, ja que en aquest país el nivell d'anglès és insuficient entre la població general com per pretendre fer servir un llibre de text que no estigui escrit en català o castellà.

## 6 Squeak / Smalltalk

*One could actually argue -as I sometimes do- that the success of commercial personal computing and operating systems has actually led to a considerable retrogression in many, many respects.  
You could think of it as putting a low-pass filter on some of the good ideas from the '60s and '70s, as computing spread out much, much faster than educating unsophisticated people can happen.*

Alan Kay [9]

Squeak [5] és una implementació del llenguatge de programació Smalltalk-80, creat al Xerox PARC durant els anys setanta [10]. Aquesta implementació es començà a tirar endavant a principis dels anys 90 a Apple, de la mà d'Alan Kay, Dan Ingalls, Ted Kaehler i altres, que són els mateixos que van inventar Smalltalk a Xerox. Squeak ara per ara és considerat programari lliure, tot i que la seva llicència és actualment motiu de discussió degut a les moltes col·laboracions que l'han fet el que és ara (la versió estàndard actual és la 3.9).

Squeak és la implementació d'Smalltalk-80 que hem escollit per ensenyar Smalltalk, l'autèntic eix vertebrador del nostre projecte. Hem triat Squeak ja que és programari lliure i perquè és la implementació d'Smalltalk en la que s'han desenvolupat projectes com Sophie, Croquet o Seaside, demostrant sobradament la seva capacitat per crear programari professional a l'alçada del que es pot fer amb qualsevol altre llenguatge i/o entorn avui dia. A més, amb Squeak s'han desenvolupat tant Scratch com BotsInc, per tant era la culminació natural en aquesta seqüència de tres cursos.

Per quina raó Smalltalk? Smalltalk és un llenguatge orientat a objectes *pur*, tipat dinàmic i una sintaxi molt senzilla, amb un nombre molt petit de regles fàcils d'aprendre. Tot a Smalltalk succeeix per pas de missatges entre objectes i *tot* és un objecte (fins i tot les classes). Pràcticament totes les implementacions Smalltalk vénen amb un entorn integrat de desenvolupament, amb exploradors de codi (classes i mètodes), inspectors d'objectes, depurador, gestor de versions, eines de *refactoring*, etc. (figura 5) A més,

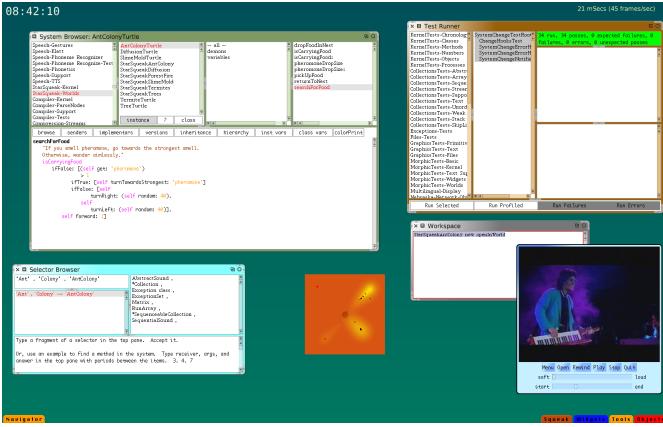


Figura 5: L'entorn Squeak, ideal per desenvolupar en Smalltalk professionalment, des d'aplicacions *web* (*seaside.st*) a entorns col·laboratius en 3D (*opencroquet.org*).

Smalltalk és un llenguatge on el codi és un ciutadà de primera classe, és a dir, podem passar-lo com a argument, retornar-lo, assignar-lo a variables, avaluar-lo quan vulguem. En aquest sentit, és una gran millora sobre els seus successors, encara que molts d'ells no amaguen el seu deute amb Smalltalk, per exemple Objective-C o Ruby.

El curs que impartim al Citilab és un curs per a gent amb coneixements previs de programació (no cal que siguin d'Smalltalk). El curs té uns 8 assistents i portem tres setmanes a l'hora d'escriure això. Ens trobem, per tant, en la mateixa situació que amb el curs de BotsInc, que és massa aviat com per dir res. Fins ara hem enfocat el curs d'una manera pràctica, ja que un dels problemes d'iniciar-se en Smalltalk és que l'aprenentatge del llenguatge i de l'entorn de desenvolupament han d'anar en paral·lel al principi. Això fa que l'aprenentatge del llenguatge sigui una mica més lent del que seria desitjable i és per això que insistim des de ben aviat en fer programes dins l'entorn, no sobre el paper.

## 7 Conclusions

*Every reader should ask himself periodically “Toward what end, toward what end?”, but do not ask it too often lest you pass up the fun of programming for the constipation of bittersweet philosophy.*

Alan Perlis [2]

En aquest article hem provat d'argumentar les raons per les que en tirar-nos a la piscina, estavem convençuts que hi havia aigua. Si ens hem fet mal o no, és massa d'hora per dir-ho. Creiem que, essencialment, aquest és un projecte a mig-llarg termini que necessita perseverància per part dels

professors i el Citilab, i fidelitat per part de la gent que vol formar part d'aquest experiment. Per cert, la qüestió de la fidelitat dels estudiants, quan l'assistència és completament voluntària i no reben cap títol al final, és tot un altre tema que mereixeria una discussió a part.

Resumint, sobre la taula tenim un parell de qüestions que s'han revisat superficialment en aquest article, però que creiem importants:

1. Cal ensenyar a programar a tothom? És un objectiu desitjable? Què aporta exactament la programació d'ordinadors a la formació intel·lectual i cultural d'una persona? Nosaltres creiem que la programació d'ordinadors aporta quelcom de positiu i diferent a la formació d'una persona: Hàbits i coneixements que tenen un cert valor pràctic en el dia a dia de, com a mínim, qualsevol persona que visqui en un entorn urbà del primer món.
2. Si la pregunta anterior té una resposta afirmativa, com s'ha de fer això? Ja sabem que les diverses maneres de programar (orientada a objectes, funcional, lògica) impliquen visions del món força diferents, la qual cosa fa que els debats per respondre aquesta pregunta sovint semblin debats de caire religiós més que debats assenyats i seriosos. Nosaltres hem pres partit per Smalltalk, però clarament no és la única solució possible.

Les decisions ja han estat preses i el projecte ja està en marxa. Com ja hem dit, és massa d'hora com per concloure res, i suposem que dins les jornades, amb tres mesos més per treballar, podrem aportar algun resultat que il·lumi una mica la viabilitat de les nostres intencions.

## 8 Agraïments

Voldríem agrair al Citilab i al seu personal la confiança que ha permés posar en marxa el projecte aquí descrit.

## 9 Llicència

Aquest article es distribueix sota una llicència Creative Commons Reconeixement-Sense obres derivades 2.5 Espanya. Veieu <http://creativecommons.org/licenses/by-nd/2.5/es/deed.ca> per més informació.

## Referències

- [1] Harold Abelson and Andrea diSessa. *Turtle Geometry — The Computer as a Medium for Exploring Mathematics*. MIT Press, 1986.

- [2] Harold Abelson and Gerald Jay Sussman with Julie Sussman. *Structure and Interpretation of Computer Programs*. MIT Press, 2nd edition, 1996.
- [3] Harold Abelson and Jack Wisdom with Meinhard Mayer. *Structure and Interpretation of Classical Mechanics*. MIT Press, 2001.
- [4] R. M. Aiken. Experiences and observations on teaching computer programming and simulation concepts to high school students. In *SIGCSE '72: Proceedings of the second SIGCSE technical symposium on Education in computer science*, pages 67–71, New York, NY, USA, 1972. ACM.
- [5] Oscar Nierstrasz Andrew P. Black, Stéphane Ducasse and Damien Pallet. *Squeak By Example*. Square Bracket Associates, 2nd edition, 2008.
- [6] Leila de Campo. Introducing the computer at a small liberal arts college. In *SIGCSE '70: Proceedings of the first SIGCSE technical symposium on Education in computer science*, pages 113–117, New York, NY, USA, 1970. ACM.
- [7] Stéphane Ducasse. *Squeak – Learn Programming with Robots*. Apress, 2005.
- [8] Stéphane Ducasse. *Squeak – Aprèn a Programar amb Robots*. - en preparació -, 2008.
- [9] Stuart Feldman. A conversation with Alan Kay. *Queue*, 2(9):20–30, 2005.
- [10] Adele Goldberg and David Robson. *Smalltalk-80 — The Language and its Implementation*. Addison-Wesley, Reading, MA, 1983.
- [11] David J. Malan and Henry H. Leitner. Scratch for budding computer scientists. In *38th SIGCSE technical symposium on Computer science education*, pages 223–227, Covington, Kentucky, USA, March 2007. ACM, ACM.
- [12] John Maloney, Kylie Peppler, Yasmin B. Kafai, Mitchel Resnick, and Natalie Rusk. Programming by choice: Urban youth learning programming with scratch. In *39th SIGCSE technical symposium on Computer science education*, pages 367–371, Portland, Oregon, USA, March 2008.
- [13] Seymour Papert. *Mindstorms — Children, Computers and Powerful Ideas*. Basic Books, 2nd edition, 1993.
- [14] Paolo A. G. Sivilotti and Stacey A. Laugel. Scratching the surface of advanced topics in software engineering: A workshop module for middle

- school students. In *39th SIGCSE technical symposium on Computer science education*, pages 291–295, Portland, Oregon, USA, March 2008. ACM, ACM.
- [15] Ursula Wolz, John Maloney, and Sarah Monisha Pulimood. ‘scratch’ your way to introductory cs. In *39th SIGCSE technical symposium on Computer science education*, pages 298–299, Portland, Oregon, USA, March 2008. ACM, ACM.