

Squeak, un amplificateur d'idées

Hilaire Fernandes

OFSET

Automne 2006



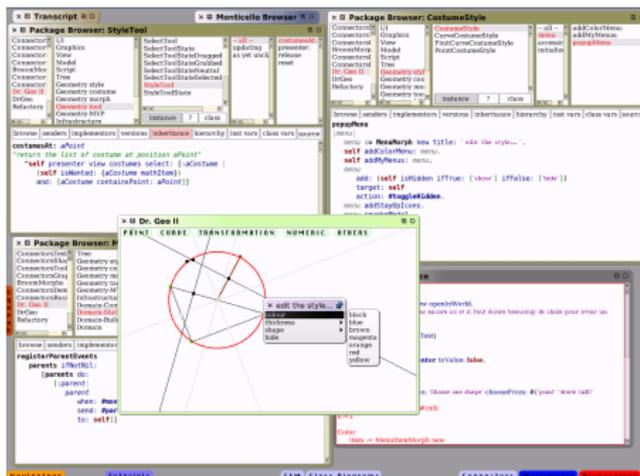
- 1 Introduction
- 2 Genèse de Squeak
- 3 Squeak, environnement éducatif
- 4 Squeak, environnement de développement
 - Le langage
 - Les outils de développement
- 5 Conclusion



Squeak, c'est quoi ?

Definition

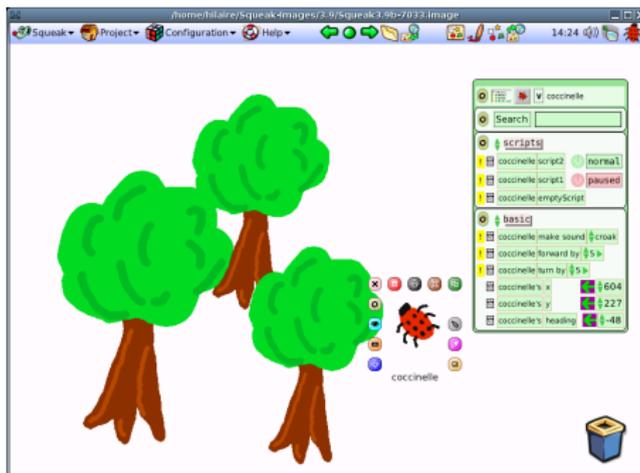
C'est **Smalltalk**, un environnement et un langage de pure programmation objet.



Squeak, c'est quoi ?

Definition

C'est **EToys**, un environnement interactif et multimédia de jouets électroniques à créer et programmer graphiquement.



Comment fonctionne-t-il ?

Comme tout environnement Smalltalk, Squeak est constitué de deux parties :

- 1 Une machine virtuelle – **VM** – qui est un programme binaire
- 2 Une **image** exécutée par la VM.



Où trouver Squeak ?

- En français : <http://community.offset.org/wiki/Squeak> ;
- En anglais : <http://www.squeak.org/> et <http://www.squeakland.org/> pour la partie éducative anglophone.



Une ambivalence éducation-programmation

Pour explorer :

Une ambivalence éducation-programmation

Pour explorer :

- de nouvelles interfaces graphiques
 - ▶ le pattern MVC vient du monde Smalltalk
 - ▶ les interfaces Morphic



Une ambivalence éducation-programmation

Pour explorer :

- de nouvelles interfaces graphiques
 - ▶ le pattern MVC vient du monde Smalltalk
 - ▶ les interfaces Morhic
- de nouveaux modèles de programmation
 - ▶ programmation incrémentale
 - ▶ écriture de code et débogage fusionnés
 - ▶ méta-programmation



Une ambivalence éducation-programmation

Pour explorer :

- de nouvelles interfaces graphiques
 - ▶ le pattern MVC vient du monde Smalltalk
 - ▶ les interfaces Morhic
- de nouveaux modèles de programmation
 - ▶ programmation incrémentale
 - ▶ écriture de code et débogage fusionnés
 - ▶ méta-programmation
- de nouveaux environnement éducatifs
 - ▶ expérimentation d'idées avec un modèle informatique
 - ▶ approche constructiviste de l'apprentissage



Une ambivalence éducation-programmation

Pour explorer :

- de nouvelles interfaces graphiques
 - ▶ le pattern MVC vient du monde Smalltalk
 - ▶ les interfaces Morhic
- de nouveaux modèles de programmation
 - ▶ programmation incrémentale
 - ▶ écriture de code et débogage fusionnés
 - ▶ méta-programmation
- de nouveaux environnement éducatifs
 - ▶ expérimentation d'idées avec un modèle informatique
 - ▶ approche constructiviste de l'apprentissage

⇒ être un *amplificateur d'idées*



Qui est derrière Squeak ?

Alan Kay est le principal architecte du langage Smalltalk et Squeak. Smalltalk fût inventé dans les années 70. Squeak dans les années 90 alors chez Apple, Disney, HP et maintenant par une communauté active.



Smalltalk, l'ancêtre de Squeak

Smalltalk est un langage conçu dans les '70 :

- un *meta*-langage objet pur, simple et uniforme
- un navigateur de classe pour accéder à TOUT
- un environnement de développement
- un langage **et** un environnement utilisable par des enfants



Definition

Etoys est un framework de programmation graphique pour enfants et adultes !

L'utilisateur conçoit des objets et programme leurs comportements avec des briques graphiques enfichables.



Definition

Etoys est un framework de programmation graphique pour enfants et adultes !

L'utilisateur conçoit des objets et programme leurs comportements avec des briques graphiques enfichables.

Example

Donner du sens à la notion de nombre positif/négatif avec une voiture programmée pour avancer/reculer ou tourner à droite/gauche.



Etoys

Definition

Etoys est un framework de programmation graphique pour enfants et adultes !

L'utilisateur conçoit des objets et programme leurs comportements avec des briques graphiques enfichables.

Example

Donner du sens à la notion de nombre positif/négatif avec une voiture programmée pour avancer/reculer ou tourner à droite/gauche.

⇒ aspect constructiviste

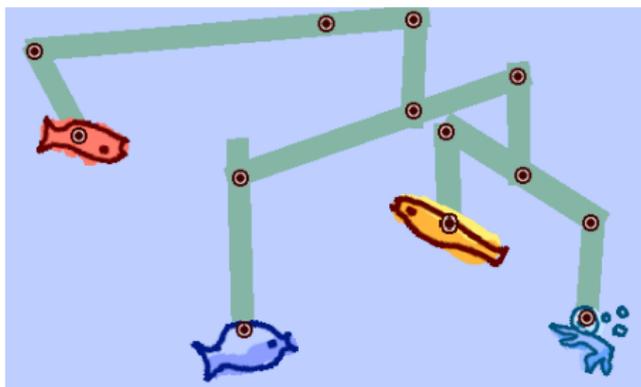
⇒ modéliser des situations



Par dessus Etoys, il existe une collections d'outils éducatifs

ODECo

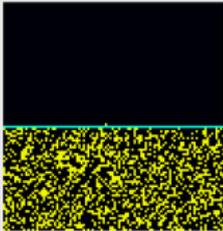
Système dynamique pour simuler des systèmes d'objets soumis à la gravité



Kedama

Système de simulation de particules

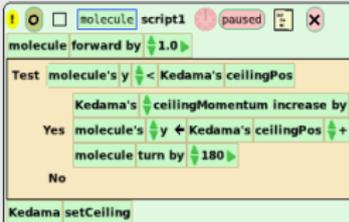
Kedama's ceilingPos2 = 46.0
Kedama's gravity = 7.0
molecule's turtleCount = 2000



Press "go" button to start simulation.

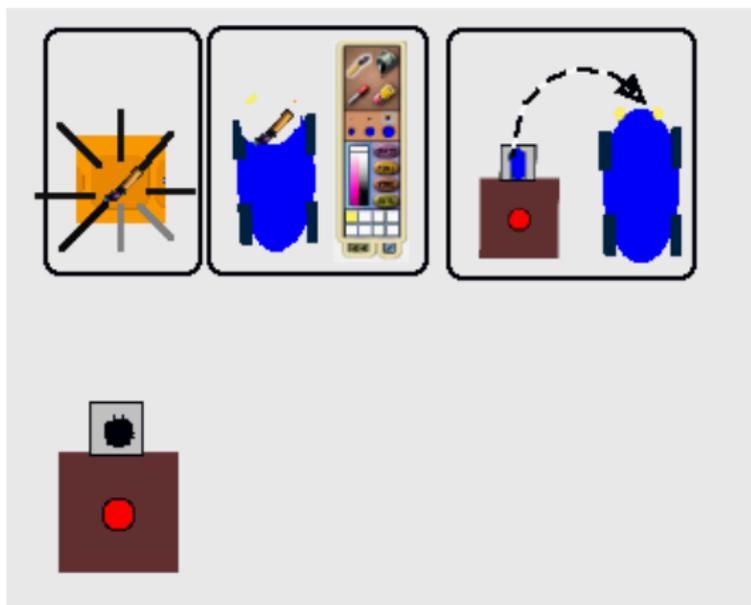
Change the turtleCount to 1000 or 500.

Then change the turtleCount back to 2000.



Vtoys

La programmation visuelle sans texte



Bots Inc

Environnement d'apprentissage à la programmation

```
Bot Workspace
Do It All Do It Clear Trails Clear Bots Clear All
| robot a hypot |
robot:= Bot new.
a:= 100.
robot go: a ; turn: 90 ; go: a.
"La longueur de l'hypoténuse d'un triangle rectangle
isocèle est égale au produit de la longueur du coté de
l'angle droit par racine carrée de deux"
hypot:= a * 2 sqrt.
"Le robot doit tourner de 90 + 45 degrés pour se mettre
dans la direction de l'hypoténuse".
robot turn: 135 ; go: hypot.
```



The screenshot shows the Dr. Geo II software interface. At the top, the window title is "x Dr. Geo II". Below the title bar, there are controls for "undo", "redo", and "toggle tip". A text box displays "pyramidShadow's pyramidHeight = 4.383" and a message: "--> Pyramid height calculus does not depend on the sun position." The main area features a 3D model of a pyramid with a sun icon above it. Dotted lines represent the sun's rays casting a shadow on the ground. Numerical values are shown: "4.29" for the pyramid's base width, "1.77" for the shadow's length, and "1.73" for the shadow's width. A vertical "Timeline" control is on the right. At the bottom, there is a "Navigator" bar with tabs for "CURVE", "TRANSFORMATION", "NUMERIC", and "OTHERS". Below this are "Widgets" and "Supplies" buttons. A control bar at the bottom right contains "stop", "step", and "go" buttons.

Un langage pour se concentrer sur l'essentiel

- tout est objet
- \Rightarrow un seul type de variable, pour référencer des objets
- un modèle consistant : toute chose dans Smalltalk est vraiment un objet !
- des *itérateurs* de haut niveau, de toute façon on peut les compléter
- des objets, des messages et des blocs de code, c'est tout !
- tous les opérateurs peuvent être étendus
- pas d'attribut public, protected, private, etc \rightarrow tous sont protected
- et bien sûr un ramasse miette – *garbage collector*



Coercition des objets

Le typage en Smalltalk est défini par les objets eux-mêmes (typage de second ordre). Il est alors aisé d'avoir des objets qui semblent muter :

```
a := 1/3.
```

```
a class. -> Fraction
```

```
a := a + (2/3).
```

```
a class. -> SmallInteger
```



Des itérateurs de haut niveau

Les itérateurs s'utilisent avec des blocs de code :

- sélectionner des nombres d'une collection

```
#(1 2 3 4 5) select: [:i | i odd] -> #(1 3 5)
```

- calculer des valeurs sur une collection :

```
#(1 2 3 4 5) collect: [:i | i * i ] -> #(1 4 9 16 25)
```



Les blocs de code

Comme promis ce sont des objets !

- C'est une fonction anonyme (\approx λ -fonction en Scheme).
- Il peut être passé à une méthode comme argument
- Comme tout objet il peut être référencé par une variable et utilisé comme une fonction mathématique :

```
f := [:x | (x raisedTo: 3)]
```

```
f value: 5
```

```
-> 125
```

```
(1 to: 10) collect: [:x | f value: x]
```

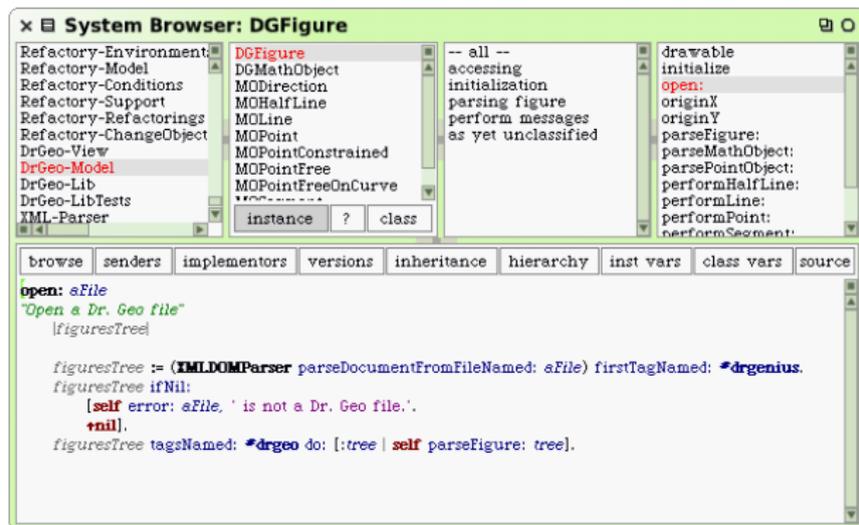
```
-> #(1 8 27 64 125 216 343 512 729 1000)
```

- **Vous les utiliserez beaucoup et vous aimerez ça !**



Le navigateur de code

La navigateur de code – ou *browser* – est l'outil central du développeur Smalltalk. Il permet d'écrire et de naviguer dans les classes et méthodes.



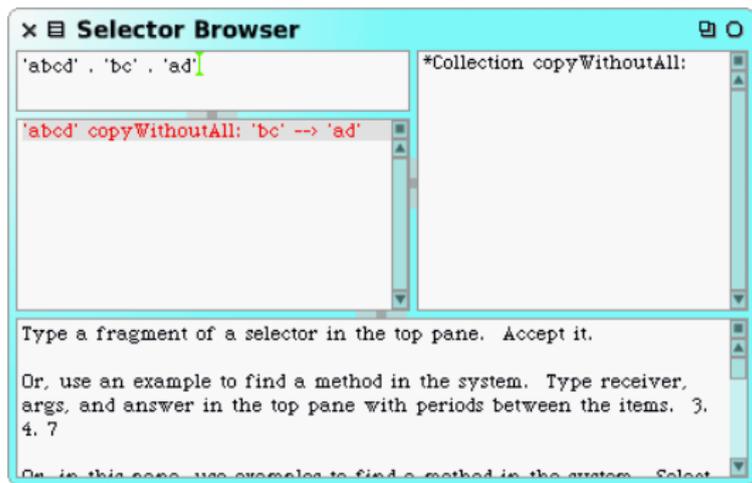
The screenshot shows the 'System Browser: DGFigure' window. The left pane displays a class hierarchy with 'DGFigure' selected. The middle pane shows the class's superclass '-- all --' and its subclasses. The right pane shows the methods of 'DGFigure', including 'drawable', 'initialize', 'open', 'originX', 'originY', 'parseFigure:', 'parseMathObject:', 'parsePointObject:', 'performHalfLine:', 'performLine:', 'performPoint:', and 'performSegment:'. Below the panes are tabs for 'browse', 'senders', 'implementors', 'versions', 'inheritance', 'hierarchy', 'inst vars', 'class vars', and 'source'. The 'source' tab is active, showing the following code:

```
open: aFile
  "Open a Dr. Geo file"
  |figuresTree|

  figuresTree := (XMLDOMParser parseDocumentFromFileNamed: aFile) firstTagName: #drgeous.
  figuresTree ifNil:
    [self error: aFile, ' is not a Dr. Geo file.'.
     +nil].
  figuresTree tagsNamed: #drgeo do: [:tree | self parseFigure: tree].
```

Le trouveur de méthode

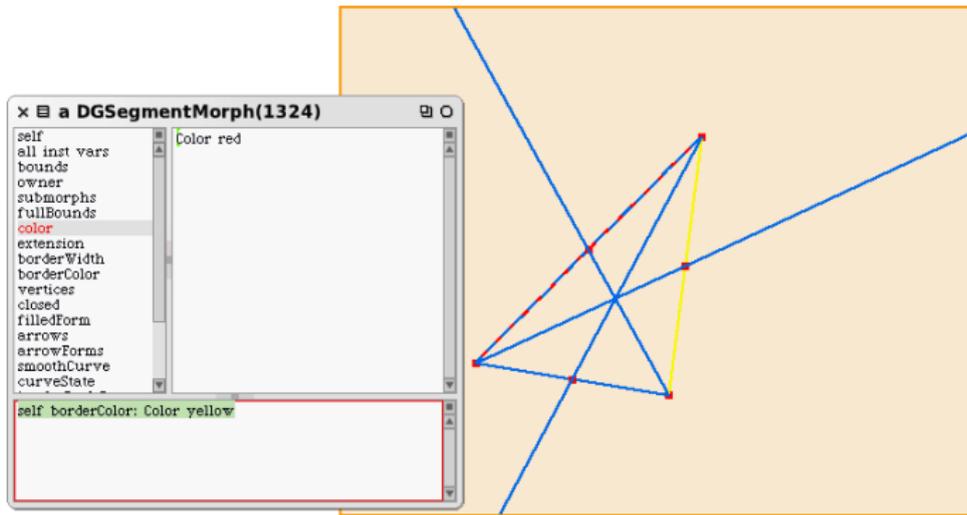
Smalltalk est un langage réflexif capable d'introspection : il peut déterminer une méthode répondant à un comportement donné. Ici avec le motif **'abcd' . 'bc' . 'ad'** nous trouvons les méthodes enlevant une sous chaîne d'une chaîne de caractères :



L'inspecteur

Avec l'inspecteur, alors que votre programme fonctionne, vous pouvez de façon *très confortable* :

- 1 inspecter vos classes et leurs attributs
- 2 modifier les attributs alors que le programme fonctionne



Le debugger

Avec le débogueur vous pouvez déboguer, c'est la moindre des choses !

Le debugger

Avec le débogueur vous pouvez déboguer, c'est la moindre des choses !
Mais vous pouvez aussi corriger les bugs et recompiler la méthode *en situation*



Le debugger

Avec le débogueur vous pouvez déboguer, c'est la moindre des choses !
Mais vous pouvez aussi corriger les bugs et recompiler la méthode *en situation*

⇒ c'est une des facettes du développement incrémentale qui divise par **deux** votre temps de développement.

The screenshot shows two windows from the Squeak debugger. The left window, titled 'System Browser: MOPointFree', displays a class hierarchy with 'MOPointFree' selected. Below the browser, the source code for the 'translate' method is visible:

```
translate: aPoint  
self halt.  
point := point + (drawable morphToVector: aPoint).  
self update
```

The right window, titled 'Halt', shows the current execution context. The stack trace indicates the halt occurred in the 'translate' method of an 'MOPointFree' object. Below the stack trace, the debugger provides a table of local variables:

Variable	Value
self	a MOPointFree
all inst vars	
mathObjectList	
style	
thisContext	-8@0
all temp vars	
aPoint	

At the bottom of the 'Halt' window, there are several control buttons: Proceed, Restart, Into, Over, Through, Full Stack, and Where.

En quelques mots...



Le symbole de Smalltalk est la montgolfière, ballon à air chaud qui s'élève dans les airs. C'est effectivement le sentiment que donne cet environnement, alors n'hésitez pas à l'utiliser, car en plus c'est libre ! Une communauté sympathique vous attend sûre :

<http://community.ofset.org/wiki/Squeak>

