

# Développer des logiciels éducatifs en Squeak/Smalltalk, retour sur 3 ans d'expérience

Hilaire Fernandes – [hilaire@ofset.org](mailto:hilaire@ofset.org)

CDDP des Landes – OFSET

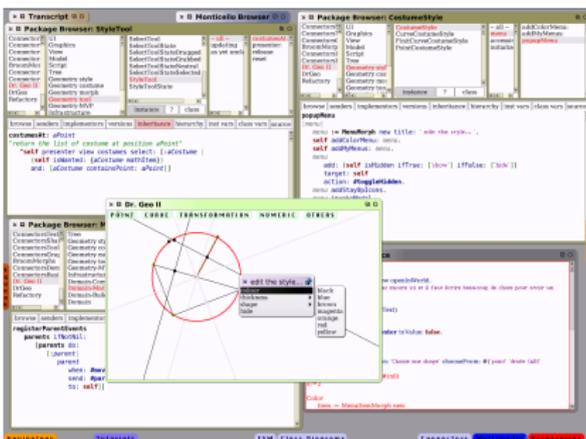
Juillet 2008

- 1 Un point de vue historique
- 2 Pourquoi Smalltalk ?
- 3 Session typique de développement en Smalltalk
- 4 Éléments du langage
- 5 Revue des outils de développement
- 6 Exemples d'applications
- 7 Conclusion

# Qu'est-ce que Smalltalk ?

## Definition

**Smalltalk** est un environnement graphique et un langage de programmation purement objet.



## D'où nous vient Smalltalk ?

**Smalltalk** fût conçu dans les '70 :

- projet de recherche des années 70 du Xerox Parc
- simplicité du langage, interface graphique avec souris
- un langage **et** un environnement utilisables par des enfants



L'ordinateur Alto



## De qui nous vient Smalltalk ?

**Alan Kay, Daniel Ingalls, Adele Goldberg** sont les principaux architectes 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, son environnement graphique et son système de pointage (la souris) ont largement inspiré Apple pour ses Macintosh.



# 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**, un état de la machine, exécutée par la VM.

# Un outil anthropocentré

# Un outil anthropocentré

- 1 environnement graphique

# Un outil anthropocentré

- 1 environnement graphique
- 2 outils de développement de haut niveau

# Un outil anthropocentré

- 1 environnement graphique
- 2 outils de développement de haut niveau
- 3 langage :

# Un outil anthropocentré

- 1 environnement graphique
- 2 outils de développement de haut niveau
- 3 langage :
  - nombre de mots clés réduit  $\Rightarrow$  faible charge cognitive

# Un outil anthropocentré

- 1 environnement graphique
- 2 outils de développement de haut niveau
- 3 langage :
  - nombre de mots clés réduit  $\Rightarrow$  faible charge cognitive
  - purement objet même dans la définition des ses *instructions* (if, boucle,...)  $\Rightarrow$  **cohérence du code**

# Un outil anthropocentré

- 1 environnement graphique
- 2 outils de développement de haut niveau
- 3 langage :
  - nombre de mots clés réduit  $\Rightarrow$  faible charge cognitive
  - purement objet même dans la définition des ses *instructions* (if, boucle,...)  $\Rightarrow$  **cohérence du code**
  - code objet beaucoup plus simple que C++, Java, C#  $\Rightarrow$  **comprendre un code source est plus facile**

# Un outil anthropocentré

- 1 environnement graphique
- 2 outils de développement de haut niveau
- 3 langage :
  - nombre de mots clés réduit  $\Rightarrow$  faible charge cognitive
  - purement objet même dans la définition des ses *instructions* (if, boucle,...)  $\Rightarrow$  **cohérence du code**
  - code objet beaucoup plus simple que C++, Java, C#  $\Rightarrow$  **comprendre un code source est plus facile**
  - cohérence du langage, pas de mélange objet et fonctionnel

# Un outil anthropocentré

- 1 environnement graphique
- 2 outils de développement de haut niveau
- 3 langage :
  - nombre de mots clés réduit  $\Rightarrow$  faible charge cognitive
  - purement objet même dans la définition des ses *instructions* (if, boucle,...)  $\Rightarrow$  **cohérence du code**
  - code objet beaucoup plus simple que C++, Java, C#  $\Rightarrow$  **comprendre un code source est plus facile**
  - cohérence du langage, pas de mélange objet et fonctionnel
- 4 le concept d'image contenant des objets persistants (*vivants* d'une session de travail à une autre)  $\Rightarrow$  **proche de la façon de travailler de l'être humain.**

## Quels avantages ?

Et concrètement quels avantages pour Squeak/Smalltalk...

- libre
- environnement graphique et langage pensés utilisateur, ce n'est pas Emacs !
- une excellente portabilité des applications : GNU/Linux, Windows, Mac OSX
- un langage réflexif : on peut manipuler du code source comme tout autre objet
- des outils de développement qui s'appuient sur la réflexivité du langage
- un ensemble de classes pour faire du multimédia
- interfaces utilisateur originales (Morph)

## Quels inconvénients ?

Ce sont les inconvénients de ses avantages !

# Démonstration

- écrire des bouts de code dans un workspace
- trouver des traitements spécifiques sur des objets avec le method-finder
- trouver la définition d'une classe ou d'une méthode à partir du workspace : Point [Alt]+b
- naviguer dans l'interface d'un objet
- montrer une session typique de mise au point d'un programme, démo interactive :  
<http://blog.offset.org/hilaire/index.php?post/Why-Squeak-Part-III>

# 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 ( $\approx$   $\lambda$ -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
```

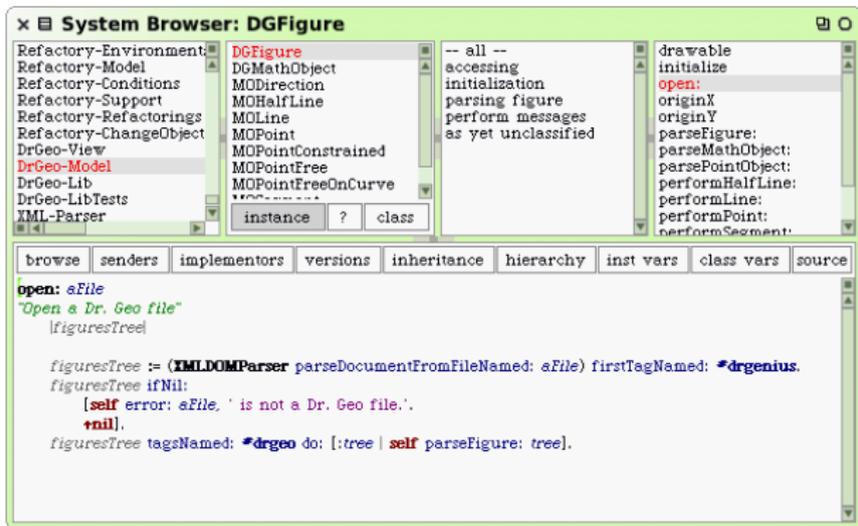
```
(1 to: 10) collect: [:x | f value: x]
```

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

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

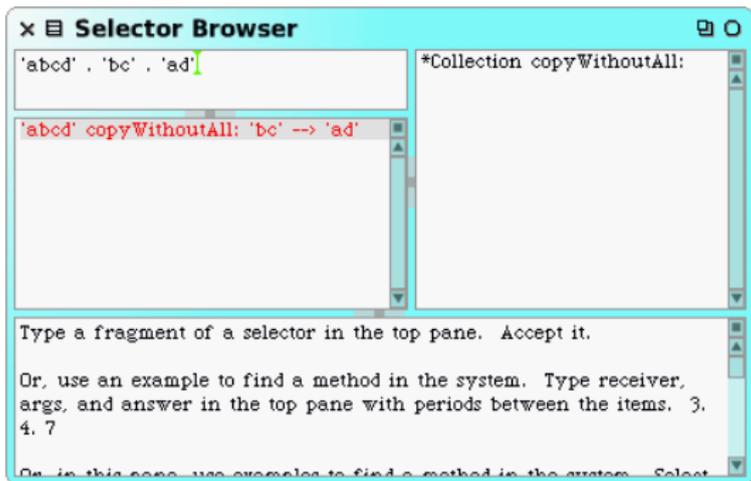
# Le navigateur de code

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.



## Le trouveur de méthode

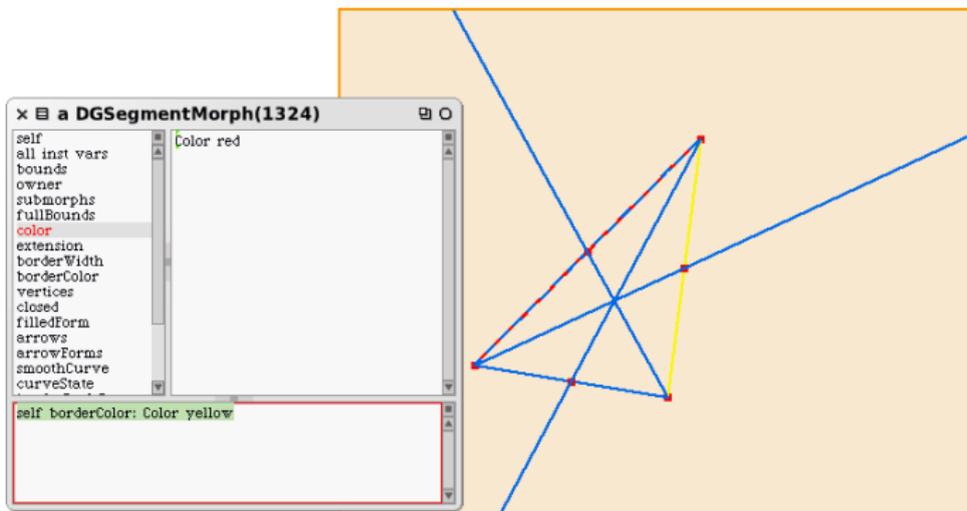
Smalltalk est un langage réflexif capable d'introspection : il peut déterminer une méthode répondant à un comportement donné. Ici avec le motif **'abcd' . 'bc' . 'ad'** nous trouvons les méthodes enlevant une sous chaîne d'une chaîne de caractères :



# L'inspecteur

Avec l'inspecteur, alors que vous programmez, 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 débogguer, c'est la moindre des choses !

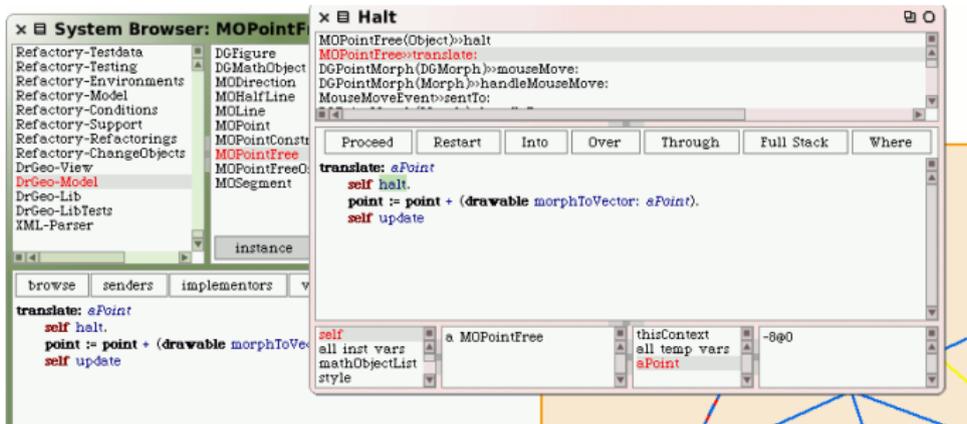
# Le debugger

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

# Le debugger

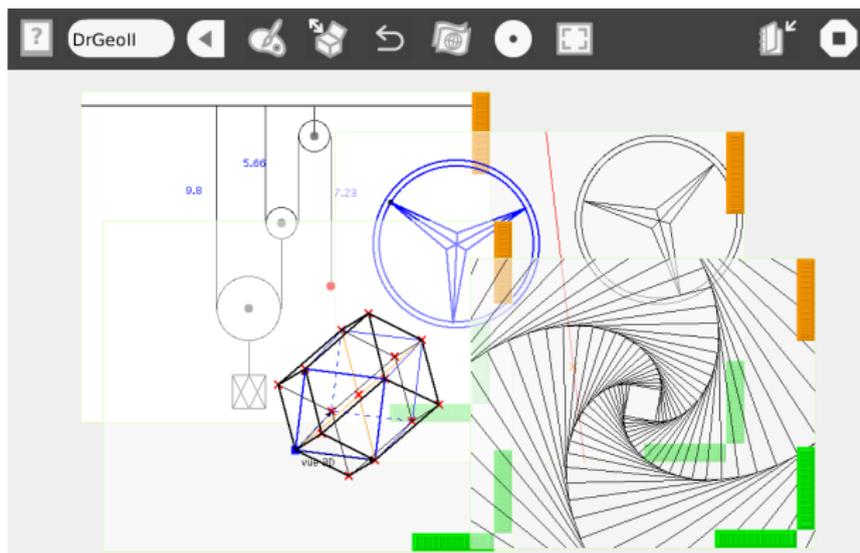
Avec le débogueur vous pouvez déboguer, c'est la moindre des choses ! Mais vous pouvez aussi corriger les bugs et recompiler la méthode *en situation*

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



# Dr. Geo II

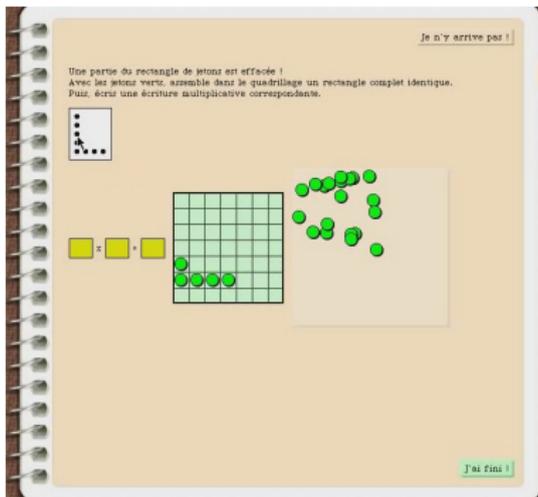
## Géométrie interactive



<http://wiki.laptop.org/go/DrGeo>

# iSTOA.net

Plate forme interactive de soutien scolaire, brique interactive élève :



<http://www.istoa.net>

## iSTOA.net

Plate forme interactive de soutien scolaire, brique web enseignant :

**iSTOA.net**  
fichier pédagogique interactif par internet

**AJOUTER OU ÉDITER UN DEVOIR**

Description

Pour le

Activités

1. Utiliser la table de Pythagore pour trouver le 2e facteur d'un produit. ==
2. Faire l'égalité de la somme de 2 ou 3 cubes. ==
3. Approche à la décomposition en base 10 d'un nombre. ==
4. Utiliser l'écriture a x b pour désigner le cardinal d'une collection d'objet indépendamment de leur position. ==

**ARBRES DES COMPETENCES**

Nom	Description
Mathématiques	
Calcul	
Calcul mental résultats mémorisés, coordonnées...	Utiliser la table de Pythagore pour écrire un produit.
Additionner ou soustraire mentalement des décimaux...	Utiliser la table de Pythagore pour trouver le 2e facteur d'un produit.
Multipliciter ou diviser un nombre entier ou décimal...	Utiliser la table de Pythagore pour trouver le 1er facteur d'un produit.
Savoir passer mentalement le résultat multiplié...	
Connaitre les tables d'addition (de 1 à 9) et...	

**DEVOIRS**

**ÉLÈVES**

<http://www.istoa.net>

# Sophie

Le futur du livre interactif, présentation le 4 juillet ici au RMLL

File Edit Help

Books Using Text.sophie (published, author mode)

# Sophie TUTORIALS

## USING TEXT

- How do I get text on a page in Sophie?
- How does editing text work in Sophie?
- How do the text halos work?
- What does the character halo do?
- What does the paragraph halo do?
- How can I change linespacing on paragraphs?
- How can I automatically indent a paragraph?
- What if I don't want my paragraph to be left-aligned?
- What if I want to change the character settings for the paragraph?
- How do I change the indents on my paragraph?
- How do I save and use a paragraph style?

## HOW TO USE THIS TUTORIAL

Each lesson in this tutorial has a video demonstration, which you may watch in a separate window by clicking the 'Start Video' button on the right-hand side of the page.

If you would like to work along with the lesson, you may do so by clicking the 'Open Workspace' button. This will open an alterable workspace in another window.

If you would like to comment on a page, passage, or lesson, please click the 'Add Comment' button at the top of the desired page. Note that any comment you leave will be visible to all other users.

100% search page frames test mode

Text Frames: cor, cor, flow, non, non

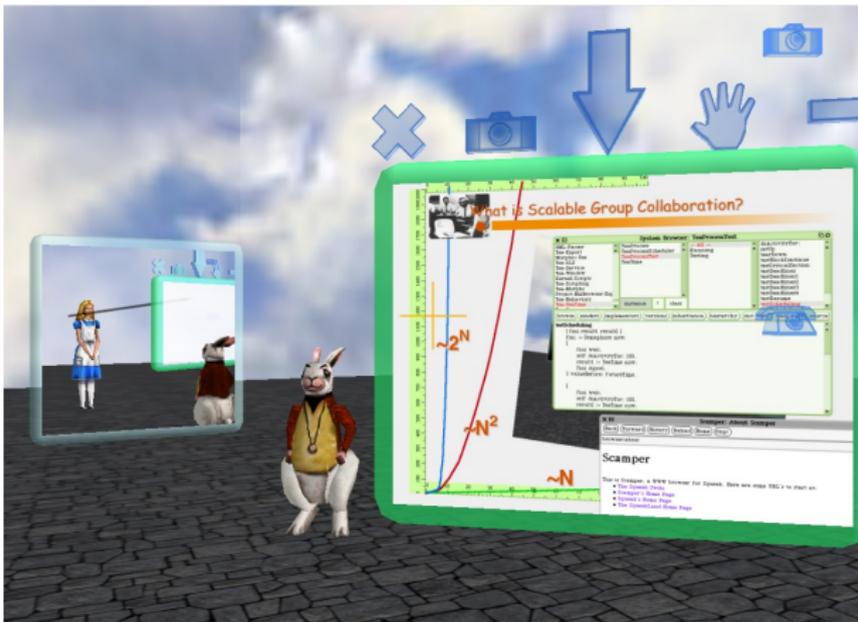
Page Templates: defau, tutori

Book Templates: Boo, Boo, Boo, Boo

<http://sophieproject.org>

# Croquet

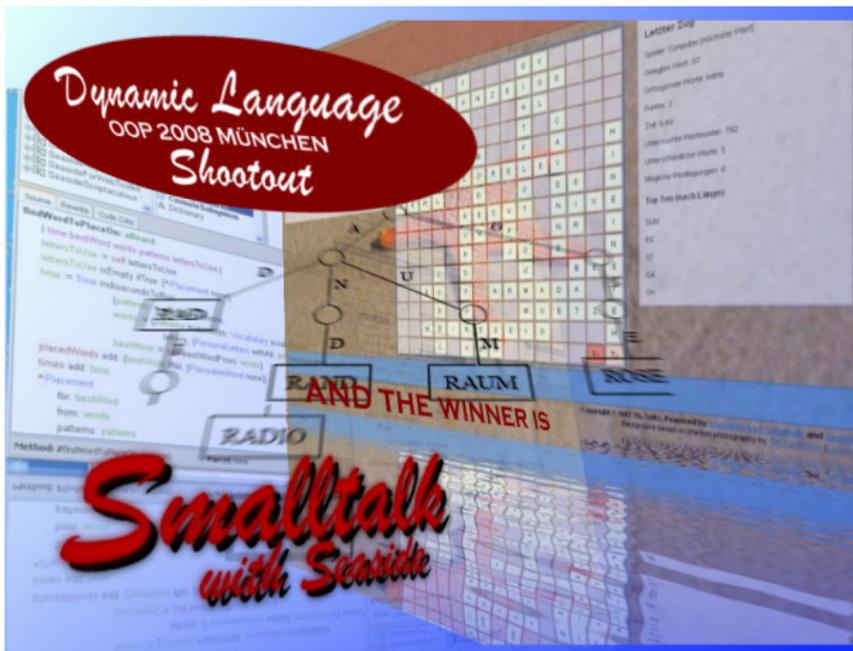
Un environnement pour créer des applications collaboratives en ligne dans des mondes virtuels.



<http://www.opencroquet.org/>

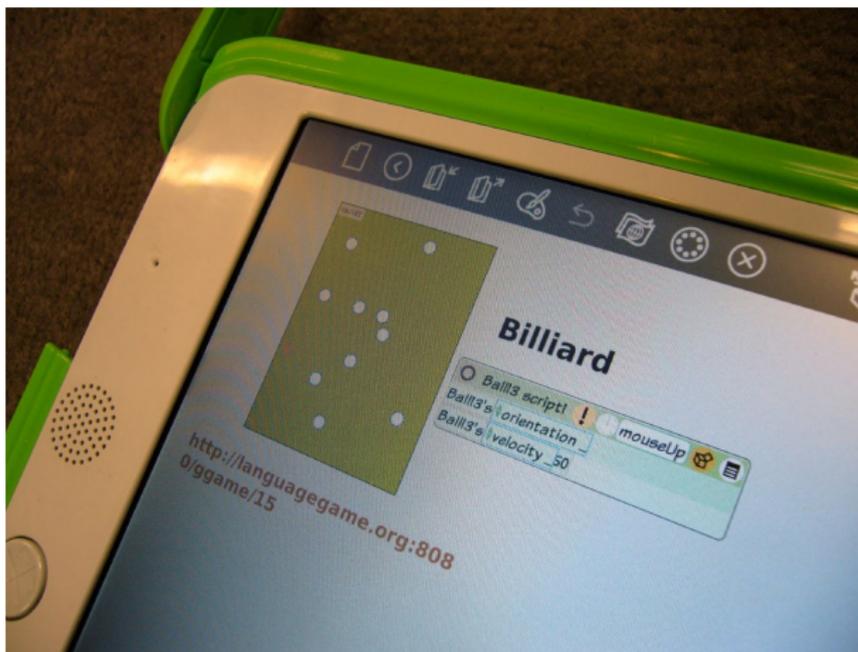
# Seaside

Un framework pour écrire des applications web de haut niveau, ateliers ici au RMLL



# EToys

Pour créer, modéliser. Intégré dans le portable XO du projet OLPC



<http://wiki.laptop.org/go/Etoys>

## Mais aussi...

- **BotsInc**, programmer des robots virtuels  
<http://smallwiki.unibe.ch/botsinc>
- **SqueakBot**, programmer et contrôler de vrais robots  
[http://www.planete-sciences.org/robot/boiteabots/index.php?option=com\\_content&task=blogcategory&id=14&Itemid=39](http://www.planete-sciences.org/robot/boiteabots/index.php?option=com_content&task=blogcategory&id=14&Itemid=39)

## En quelques mots...

La suite sur



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