

Squeak: Un Smalltalk Open Source détonnant !

Dr. Stéphane Ducasse
ducasse@iam.unibe.ch
<http://www.iam.unibe.ch/~ducasse/>

1. Squeak: Back to the Future

Squeak est le Smalltalk Open Source développé chez Disney par une partie de l'équipe originale de Xerox qui inventa à la fin des années 70 : le multi-fenêtrage, les environnements de programmation, la manipulation de bitmaps et finalement la programmation à objets [Kay77]. Steve Jobs admet avoir été fortement influencé par une démo de Smalltalk vue chez Xerox. Cette même équipe a aussi été précurseur dans l'utilisation d'écran bitmaps, l'utilisation de souris, la définition des machines virtuelles, de ramasses-miettes, de génération de byte-code, de code multi-plateformes et de conception objet de bibliothèques ... Bref de beaucoup d'aspects que l'on pense novateur en Java aujourd'hui.

Mais Squeak ne se limite pas à refaire le passé, Squeak est définitivement tourné vers le future. Il permet de programmer le Net (Pop, Html, Http,...). Il contient un mini navigateur (Scamper), un mini lecteur de mail (Celeste), serveur web (PWS), un lecteur/joueur de Flash, de nouvelles interfaces graphiques (les Morphs), un lecteur/joueur de MIDI, WAV, un digitaliseur, un très puissant reconnaisseur de caractères (Génie disponible en 3.1), un système de synthèse vocale, un moteur 2D/3D (Balloon), un environnement de script pour novice, un environnement distribué partagé (Nebraska), une machine virtuelle écrite en Squeak puis traduite en C et donc extrêmement portable... Squeak est utilisé dans de nombreux projets comme Comanche à un serveur web open source.

Comme tout, tout, ...absolument tout est objet en Squeak (la souris, le compilateur, les fenêtres, les sons, les bitmaps, les entiers.....) et que tout le code est à disposition (comme cela toujours était le cas avec les Smalltalks depuis 1980), vous pouvez lire et changer n'importe quel aspect du système.

Cet article vous propose un rapide tour en Squeak en espérant vous donnez envie d'essayer. Nous vous proposons d'interagir directement Squeak. Dans de prochains articles, nous vous montrerons comment écrire de petit scripts ou des programmes en Squeak. Le prochain article montrera les aspects que nous n'avons pas le temps de décrire dans cet article. Chacun des aspects présentés ainsi que la syntaxe et la sémantique de Squeak feront l'objet d'articles détaillés. Squeak vous propose quelques exemples : les "Play with Me" que nous allons explorer ensemble. Mais avant de commencer nous disons un mot de la motivation qui habite la communauté Squeak.

2. Amplificateur d'idées

La vision qui guide le développement de Squeak est qu'il doit être une plate-forme permettant l'exploration et la validations d'idées nouvelles. D'après Alan Kay, le père de Smalltalk et

Squeak, les ordinateurs doivent jouer le rôle d'un amplificateur d'idées [Kay77, Kay93]. Squeak est un véhicule afin de réaliser cette idée. C'est avec la vision que Squeak est un médium pour expérimenter que vous devez l'aborder. En effet, certaines applications comme Scamper (le Web browser) ou Celeste (le lecteur de emails) montrent que tous les protocoles réseaux (HTTP, Pop...) pour faire un navigateur sont là et que si vous voulez en construire un nouveau vous le pouvez (comme l'a fait une équipe de Interval dans un projet de NetPad qui fût abandonné une fois que Bill Gates ait vu le résultat et dont on peut voir des captures d'écrans à <http://www.huv.com/smalltalk/browser.html>. Notez que ce navigateur sera disponible bientôt).

Squeak est ainsi utilisé dans des projets ayant comme différents objectifs de développer de nouveaux services web, de nouvelles approches multimédia ou de travail collaboratif, de piloter des véhicules sous-marins, d'expliquer des concepts mathématiques avancés (MathMorph), de définir de nouveaux langages de programmation, d'offrir un environnement de développement pour PocketSmalltalk (un Smalltalk pour PalmPilot), de manipuler de la musique (SIREN <http://www.create.ucsb.edu/Siren/>).... En particulier, un des buts de l'équipe Disney est de définir des environnements de programmation pour les enfants. EToy un environnement de scripts est ainsi développé et utilisé dans des classes pilotes en Californie avec des enfants de 6 à 11 ans. Certains résultats seront disponibles sur <http://www.squeakland.org/> très bientôt. De plus en plus de groupes de recherches utilisent Squeak comme outil de développement pour la puissance et malleabilité disponible. Certaines companies commencent aussi à utiliser Squeak mais cela reste assez secret (<http://> montre des images d'un projet industriel défunt).

3. Obtenir Squeak

Pour pleinement apprécier cet article il est conseillé de pouvoir essayer directement en Squeak les manipulations proposées et d'interagir avec le système. Squeak fonctionne sur pratiquement toutes les plateformes, OS, certains PDA et de quelques processeurs dont voici la liste en vrac : Windows 2000, NT, 95, DOS, Acorn, Linux, LinuxPPC, Solaris, Digital Unix/Alpha, Windows Ce (Cassiopeia and HP320LX), Mac, MacOSX, SCO System V, NetBSD/Sparc, Psion 5, Rhapsody/NextStep, OS/2, SunOS, Zaurus, StrongArm, M32R/D.... Attention, les versions Unix sont un peu à la traîne et faites attention si vous n'avez pas de carte sonore, il existe un patche (<http://www.cc.gatech.edu/people/home/lex/squeak/>).

Vous pouvez trouver Squeak sur le site ftp officiel contenant les dernières versions est: <ftp://st.cs.uiuc.edu/Smalltalk/Squeak/> ou le site officiel <http://www.squeak.org/>. D'autres sites comme le site collaboratif <http://minnow.cc.gatech.edu/> contiennent de nombreuses informations.

Versions. La dernière version officielle est la version 3.0. Cependant, comme la gestion des versions ainsi que les machines virtuelles ont totalement changé entre les versions 2.8 et 3.0 donc il reste certains bugs. La dernière version stable avant la 3.0 est la version 2.8, la 2.9 n'a existé qu'en beta. C'est la version 2.8 que nous utilisons dans cet article et dont les versions PC, Mac et Linux sont fournies sur le CD.

Dans un prochain article, nous vous montrerons toutes les nouveautés présentes dans la version 3.1 comme Génie le reconnaisseur de caractères, les projets accessibles directement sur le web, un lecteur MPEG et MP3 (nécessitant une nouvelle VM).... Si vous voulez être à la pointe

vous pouvez prendre la version 3.1 alpha qui sur les serveurs. Elle contient toujours les dernières améliorations et nouveautés (attention vous devez utiliser une nouvelle machine virtuelle). D'autre part, il est possible si vous n'êtes pas derrière un firewall d'accéder en temps réel aux nouveautés et fixes (menu help... update code from server).

Lorsque vous téléchargez Squeak, une fois décompressé vous devez avoir au minimum: un exécutable (la machine virtuelle), une image (contenant du byte-code) ainsi que plusieurs autres fichiers. Si vous souhaitez pouvoir lire le code, assurez-vous d'avoir un fichier nommé SqueakVX.0.source qui représente tous les sources du code compilé présent dans l'image où X représente la version. Prenez-le s'il n'est pas fourni dans le fichier zippé. Attention le fichier image (zzzz.image) et change (zzzz.change) doivent toujours être synchronisés.

Dans certaines situations, Squeak nécessite différentes actions aussi les boutons de la souris ont les significations suivantes:

Table 1: Equivalence des sémantiques des boutons de la souris

PC	PC 2 boutons	Mac	Signification
Gauche	Gauche	Click	Sélection
Milieu	Droit	Option + Click	Menu
Droit	Alt + Gauche	Commande + Click	Halos

4. Les Morphs

Squeak utilise de nouvelles interfaces graphiques: les Morphs. Les Morphs ont été initialement développées pour le langage Self développé par Sun Labs [Morphs]. Les Morphs sont des éléments d'interfaces permettant une manipulation uniforme et directe. Squeak contient en fait deux frameworks graphiques, l'ancien et célèbre MVC et les Morphs qui sont maintenant l'interface privilégiée à partir des versions 3.0.

Play With Me -1 et Play With Me - 2 montrent un exemple de Morphs. **Ouvrez Play with Me -1** (attention l'interaction avec les icônes des fenêtres est assez déconcertante au début). Maintenant **cliquez sur une** des listes pour obtenir les "Halos" (cf. Table1). Vous devez obtenir de petites boules de différentes couleurs autour de la liste (c.f. Figure 1). Si vous laissez la souris sur un halo un petit ballon d'aide vous indique la sémantique du halo. Par exemple, **cliquez sur le halo bleu** marine en bas à gauche (rotate) et faites tourner la liste. Notez que vous pouvez toujours sélectionner ses éléments. Vous pouvez aussi la faire revenir à la situation de départ et comme vous le voyez il n'y a aucune perte. Essayez d'autres manipulations avec les morphs de Play With Me-1. N'importe quelle morph peut être manipulée de la sorte. Certaines morphs offrent plus ou moins de possibilités.

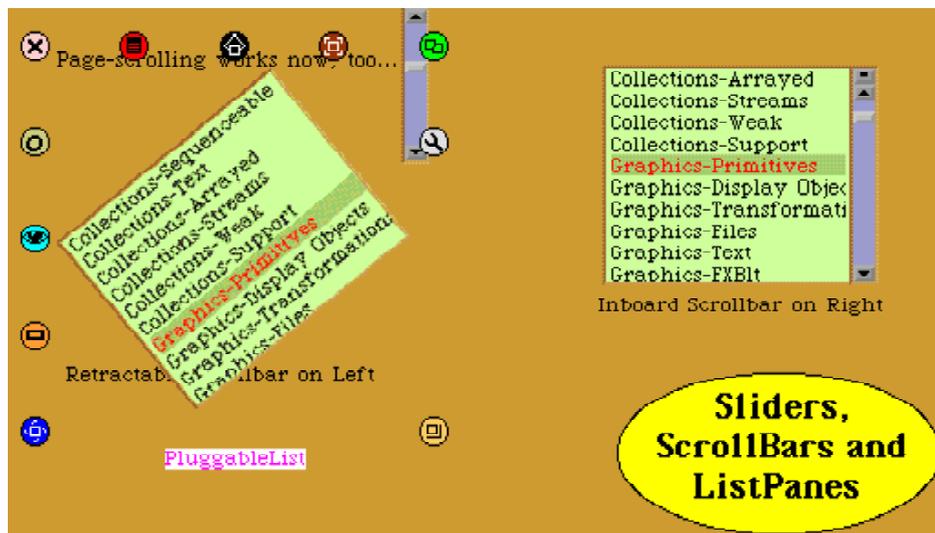


Figure 1 Un exemple de manipulation directe de Morphs. Notez que la liste continue de fonctionner.

Là sceptique vous allez dire bon et alors.... Ces exemples sont justes pour montrer la puissance des Morphs. **Ouvrez** maintenant le Play With Me- 5. Si vous **changez la taille** de la morphsupérieure vous allez voir que le texte qui la remplit se deverse dans le tuyau puis dans la morph du bas. **Changez la taille** des Morphs, et le texte la remplit (cf. Figure 2). Vous pouvez aussi écrire directement dans le tuyau comme je l'ai fait ! Vous pouvez aussi modifier la taille de l'ellipse (deux cliques pour accéder à l'ellipse + halo jaune) et voir comment le texte se réajuste automatiquement. Essayez donc de faire cela avec n'importe quel système existant.....Bon courage! Je vous expliquerai dans un autre article comment: (1) créer cet exemple par manipulation directe en une dizaine de cliques et (2) le programmer en une dizaine de lignes.

Le Play With Me - 4 montre un autre exemple : un BookMorph. En deux mots, un livre morph permet de créer des présentations ou documents qui peuvent être partagés entre utilisateurs ou non et être actifs. Je reviendrai sur cet aspect dans un autre article, pour le moment, essayez les options proposées dans le bouton <<>>. Vous pouvez aussi essayer de charger une présentation PowerPoint.

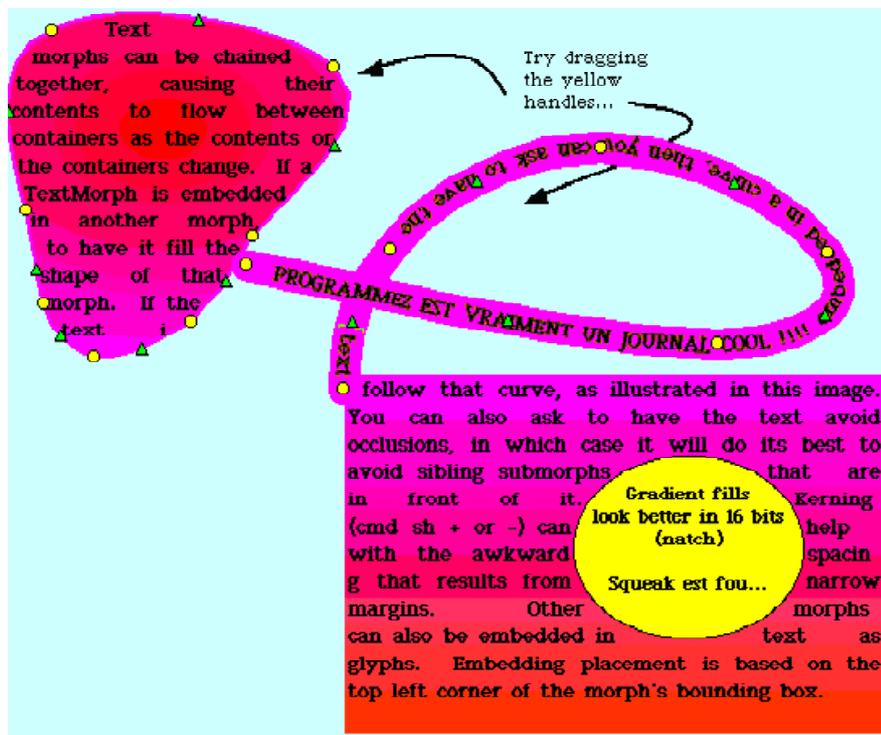


Figure 2 Illustration du pouvoir de manipulation offert: le texte prend la forme de la morph qui l'englobe puis déborde dans le tuyau pour finir dans le rectangle.

Les Morphs intègrent aussi la notion du temps et permettent aussi de faire des animations. Nous reviendrons la dessus dans un prochain article. Vous pouvez aussi créer des mondes complètement morphique :Essayez: **Open...-> Morphic Project**. Puis **cliquez** sur la petite fenêtre obtenue et **enter**. Vous voici dans un monde Morph, ici tout est morph contrairement au premier monde dans lequel les Play With Me sont.

Essayez **new morph...-> Demo -> BouncingAtomMorph**. Il s'agit d'une Morph simulant des molécules de gaz dans un espace fermé.Essayez de changer sa taille. Les Morphs peuvent aussi contenir de la 3D. Essayez les morphs B3D...Notez qu'il n'y a aucun appel à des bibliothèques extérieures comme OpenGL ou à des cartes accélératrices (bien que la communauté y pense) et que les animations sont indépendantes des machines. Voilà ce rapide tour en Morphs se termine mais bien d'autres choses pourraient être montrées sur les Morphs....

5. Flash

Squeak inclus un lecteur de Flash. Nous avons joint un fichier Flash donné par un graphiste talentueux et sympa (<http://herve.flores.online.fr/>) dans le dossier Data/flores.swf. Créez un nouveau projet morphique (**Open...->Morphic Project**) ou entrez dans un existant dj. Ensuite ouvrez un lecteur de fichiers (**Open... -> File List**) et localisez le fichier flores.swf. Sélectionnez-le, faites apparaître le menu et choisissez **open as flash**. Vous obtenez une

instance d'un lecteur Flash qui joue le fichier. En cliquant sur **stop** vous pouvez arrêter le déroulement de la séquence de morphs..... Morphs vous avez dit Morphs, et oui, les éléments constituant la séquence flash sont aussi des Morphs. Donc cela veut dire que l'on peut les manipuler, faire des rotations, grossissements, duplicas, comme le montre la Figure 3. Essayez donc de dupliquer (**halo vert**) le petit gars (pour cela cliquez de manière répétitive pour faire venir les halos, ceux-ci se déplacent de manière hiérarchique vers l'objet tant la cible du clic), ensuite vous pouvez aussi le faire grossir (**halo jaune**), et regarder la taille compressée (halo rouge -> show compressed size). Admirez la qualité du grossissement. Je vous laisse juger par vous-même de ce que l'on pourrait faire avec de telles possibilités.



Figure 3 Interprétation de fichiers Flash et manipulation directe d'éléments Flash

Encore une fois, il ne s'agit pas pour l'équipe de Disney et la communauté Squeak de faire un éditeur de Flash mais cette démo montre que tous les ingrédients sont là si une compagnie voulait le faire...

6. Musique

Squeak propose de nombreuses possibilités de manipulation et de génération de sons ou musiques (AIFF, WAV, U-Law, MIDI, sonogrammes, génération de voix humaines). Squeak définit les codecs (compression-décompression) suivants ADPCM, GSM, MuLaw et Wavelect. Nous nous limitons ici à une présentation du lecteur MIDI.

Pour cela, nous ouvrons un projet morphique: **Open...-> Morphic Project**, + **enter**. Ensuite nous allons lire le fichier MIDI: **Open...->File List + Sélection du fichier LETITBE.MIDI + menu**, + **load midi file**. Vous obtenez un joueur de MIDI avec lequel vous pouvez: changer les instruments associés aux notes (changer les avant de commencer), jouer le fichier (**play**), pi-

lotez un instrument MIDI (je n'ai pas essayé cela). Faites apparaître la partition en appuyant sur **Piano Roll**, vous devez obtenir une représentation de la partition (à droite sur la Figure 4). Vous pouvez aussi éditer un instrument (ratio, pitch...) via le choix **edit** du menu pour changer les instruments. Vous pouvez aussi éditer la partition à l'aide d'un clavier (faites apparaître un menu sur la partition elle-même....Bonne exploration.

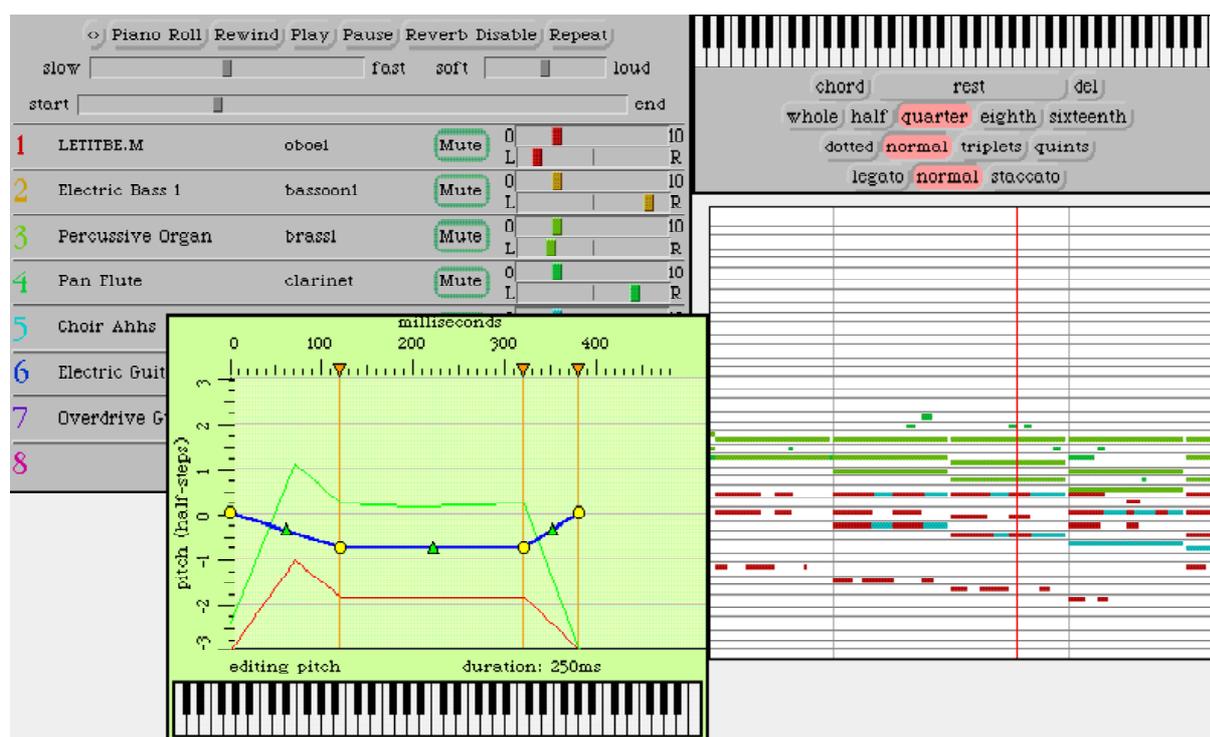


Figure 4 Un lecteur MIDI ainsi qu'un éditeur de sons.

Notez qu'il est possible de synchroniser l'interprétation musicale avec le déroulement de films (movie) avec juste du drag and drop. Mais cela sera aussi pour une autre fois....

7. Squeak et la 3D: Squeak Alice

Squeak contient un moteur 2D et 3D nommé Balloon. Ce moteur est utilisé par le lecteur de Flash ainsi que dans Squeak Alice qui est le port d'une partie d'Alice (un environnement pour novice www.alice.org). Le Play With Me 7 illustre cet aspect de Squeak. Suivez les indications pour manipuler avec le lapin.....Amusez-vous bien. Notez que le Undo ne fonctionne pas très bien dans cette version.....

Mais vous pouvez faire mieux. Agrandissez le carré représentant le monde du lapin (**halo jaune**). Puis sélectionnez via le **halo rouge** le choix **open tout drag and drop**. Ensuite créez une nouvelle instance du BouncingAtomMorph comme expliqué plus haut et lachez la sur une partie du lapin. Celle-ci doit se mapper sur la surface de celui-ci et continuer à être animée comme le montre la Figure 5.

Que dire de plus....Encore une fois, il ne s'agit pas de développer un système 3D complet, le moteur 3D pourrait inclure les collisions, les interactions entre les objets. Mais cet exemple,

montre la puissance créatrice et le potentiel de Squeak. Balloon est indépendant de la machine et fonctionne donc sur n'importe quelle machine.



Figure 5 Illustration de Balloon, l'engin 2D/3D de Squeak et exemple de mapping de Morph

8. PDA et taille réduite

Le fait que la machine virtuelle de Squeak soit écrite en un sous-ensemble de Smalltalk qui peut être traduit en C, assure une grande portabilité à Squeak. De plus, la machine virtuelle peut être réduite ou étendue suivant les besoins. Divers ports sur des PDA ont déjà eu lieu (Helio, IPac,...). Une adaptation des Morphs aux écrans minimalistes de ces machines a été développée. La taille des fichiers Squeak que vous avez est relativement grande car le fichier image contient tous les objets couramment créés et le fichier changes contient **tous** les changements effectués. En fait, la taille de Squeak incluant le framework MVC, le compilateur, des navigateurs, le débogueur et les classes principales peut être réduite à 800 k. Certains programmeurs expérimentent afin d'obtenir une machine virtuelle de l'ordre de 70K, d'autres sont en train de créer SqueakOS un OS entièrement écrit en Squeak. (Notons que l'architecture de Smalltalk est basée sur celle d'un OS et contient un ordonnanceur de tâches (scheduler), des threads que l'on peut complètement reprogrammer).

On peut par exemple penser que Disney ou d'autres sociétés pourraient proposer aux enfants des applications développées en Squeak et qui fonctionnerait sur des télévisions munies de processeurs.

9. Une première conclusion

Bien sûr comme tout le code de ce que je viens de montrer est disponible et définit sous la forme d'objets, il est possible d'écrire des scripts ou des applications. C'est ce que je vous montrerai lors d'autres articles.

Squeak est relativement stable si l'on considère que tout Squeak est écrit en lui-même!!! Cependant, comme de nombreux mouvements open source, Squeak évolue vite. D'autre part, comme pour tous les Smalltalks, la première marche est assez haute. Si l'envie vous prend de programmer avec, ce que nous vous souhaitons, soyez tenace et n'hésitez pas à demander de l'aide dans la mailing-list. En effet, Squeak n'est pas encore prêt pour Mr tout le monde. Cependant, l'équipe Disney est consciente de cet aspect et travaille à la mise au point de trois environnements: Novice, OmniUser et Avancé. La version de Squeak 2.8 est malheureusement dans cette dernière catégorie et donc assez rugueuse.

Squeak est un environnement très malléable, par exemple un étudiant est actuellement en train de réduire la machine virtuelle pour quelle n'occupe que 60 K et expérimente de nouveaux modèles d'objets. Le future de Squeak n'est donc pas toute tracé et dépendra de ce que les gens feront avec. D'autre part, Squeak est encore jeune. Commencé en 1996, il a été développé à partir du code original du premier Smalltalk et a connu de très nombreuses améliorations et nouveautés. Cependant, certains aspects du système n'ont jamais connu de refonte. Afin de pallier à ce problème et faire de Squeak, un Smalltalk comme les Smalltalk professionnels permettant à des sociétés de développer des applications industrielles, un mouvement nommé Stable Squeak s'est créé. L'objectif de Stable Squeak est de refondre certaines parties souffrant de malformations, d'offrir une librairie de base d'excellente facture et des outils de développements comme un système de versions intégré. Une Fondation Smalltalk va être créée, des sociétés se forment afin d'utiliser Squeak, donc le code ne peut que s'améliorer.

10. Pour en savoir plus

Si vous voulez en savoir plus voici quelques informations utiles. Vous pouvez aussi me contacter mais je suis souvent assez d bord et r pond lentement.

- <http://www.squeak.org/> est le site officiel qui est actuellement en plein changement.
- <ftp://st.cs.uiuc.edu/Smalltalk/Squeak/> est le serveur officiel contenant les différentes versions.
- <http://minnow.cc.gatech.edu/> est le wiki de la communauté (wiki = un serveur de pages web que tout le monde peut éditer), il regorge d'informations.
- <http://titan.cnds.unibe.ch:8080/SmalltalkWiki> est un meta wiki utilisant Comanche contenant des synthèses des liens et d'autres liens sur Smalltalk.
- <http://software.freepage.de/cronos/squeak/SqOverview.morph.gz> est un livre morph (bookMorph) qui contient une présentation de Squeak.
- Le guide rapide de la syntaxe Squeak <http://www.mucow.com/squeak-qref.html>
- European Smalltalk User Group est gratuit: <http://www.esug.org/>

La mailing-liste (accessible via www.squeak.org) est très chaleureuse mais attendez-vous recevoir environ 50-60 emails par jour si ce n'est plus.

- <http://www.create.ucsb.edu/mailman/listinfo/squeakaudio> propose une mailing liste dédiées aux aspects musicaux.

Un seul livre sur Squeak est actuellement disponible, plusieurs autres sont en préparation (certaines premières versions sont disponibles <http://www.cc.gatech.edu/~mark.guzdial/drafts/> et <http://coweb.cc.gatech.edu/squeakbook/> ainsi que sur le CD).

- Mark Guzdial, *Squeak: Object-Oriented Design with Multimedia Applications*, Prentice-Hall, Dec, 2000, ISBN: 0-13-028028-3 (les premières versions sont aussi sur le CD.)

Sur Smalltalk les livres suivants permettent un bon début:

- X. Briffault and G. Sabah, *Smalltalk, programmation orientée objet et développement d'applications*, Eyrolles, Paris. ISBN: 2-212-08914-7
- A. Sharp, *Smalltalk by Example: The Developer's Guide*, McGraw Hill, ISBN: 0079130364, 1997
- T. Hopkins and B. Horan, *Smalltalk: an Introduction to application development using VisualWorks*, Prentice-Hall, 0-13-318387-4, 1995

11. Références

[Kay77] Alan Kay and Adele Goldberg, *Personal Dynamic Media*, IEEE Computer, Mars, 31, 41, 1977

[Kay93] Alan Kay, *The early history of Smalltalk*, *History of Programming Languages (HOPL-II)*, ACM, 69-95, 1993.

[Morph] <http://www.sun.com/research/self/papers/self4.0UserInterface.html>

