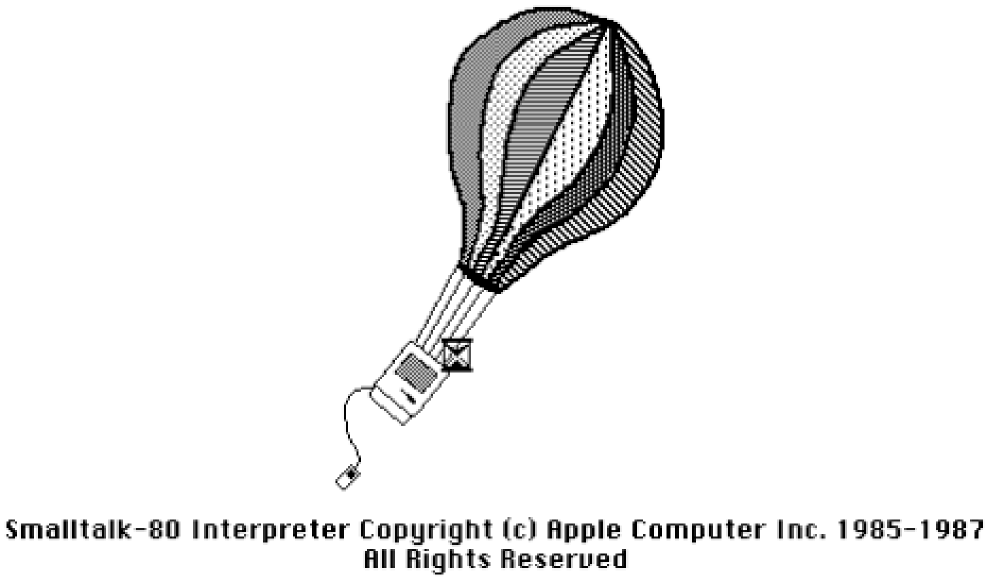
The Long View


Smalltalk-80 Interpreter Copyright (c) Apple Computer Inc. 1985-1987
All Rights Reserved

Smalltalk-80

Saturday, March 6, 2010

Everybody knows that in 1979 Steve Jobs and some Apple engineers visited the Xerox Palo Alto Research Center and were shown some of the amazing experimental hardware and software under development there. PARC was working on a number of revolutionary ideas about how to use computers, and there has been a lot said and written about what Jobs and the other saw and how it might have influenced future Apple products including the Lisa and the Macintosh. Maybe they saw an experimental version of the Star networked office system and laser printers. We know for sure that they saw the Smalltalk programming environment. We know that because we have the word of Bruce Horn, who should know (see http://www.folklore.org, article entitled "On Xerox, Apple and Progress").

Smalltalk isn't an office computer system. It doesn't include a word processor, a graph-drawing program, or a filing system using a desktop metaphor. It's a programming language, a programming environment, and a simple operating system. It’s intended for use by computer programmers, not office workers. But it clearly had a huge influence on Apple. As a programming language, it is a direct predecessor of Apple’s current favorite, Objective-C, which was designed to be a cross between Smalltalk-80 and C. As a user interface, it is a direct predecessor of the Lisa and Macintosh, and also Windows and X11. Smalltalk pioneered a lot of the key features of nearly all current computer interfaces, including the use of the mouse as a pointer, overlapping movable and resizable windows, cursors, and menus.

Smalltalk also is a predecessor of all modern Integrated Development Environments, including Microsoft's .NET and Apple's XCode. It has an integrated class browser, code editor, syntax checker with built-in spelling correction, and source level debugger. It was originally designed for use in a Xerox's networked environment, and it treated remote filesystems on servers as the equivalent of local mounted volumes. It kept managed programmers' source code documents in a built-in database rather than requiring or allowing users to organize files themselves. It presented the entire library of source code, both for the operating system and for user applications in a single browser. In this it presaged programs like iPhoto and iTunes, that relieve users of the need to name documents and organize them into directories. Smalltalk was, and still is, a little miracle. In many ways, computer systems now still fall a little short of fulfilling the promise that was represented by the Smalltalk system.

Apple Smalltalk-80 on the Macintosh

Want to see what Smalltalk-80 looked like? It turns out to be pretty easy to do, if you have a Mac Plus or a working Mac Plus emulator like Mini vMac. Only a couple of years after the legendary 1979 visit, Apple participated in an experimental project with Xerox PARC, Tektronix, Hewlett Packard and Digital Equipment Corporation, in which they implemented Smalltalk-80 on their respective machines. There is a photo of a Lisa (the original Lisa with dual 5.25" diskettes) running Apple's implementation of Smalltalk on page 5 of Adele Goldberg's classic book "Smalltalk-80 The Interactive Programming Environment" (Addison-Wesley, 1984). I guess that it by participating in that experiment, Apple got the rights to build and distribute Smalltalk-80 (version 1) for Apple computers. I don't think that Apple ever distributed the native Lisa version of Smalltalk-80 that is shown in the photo. But on page 53 of the paper documents that came with the "May 1985" Software Supplement mailing to Macintosh developers, there was a note that said this:

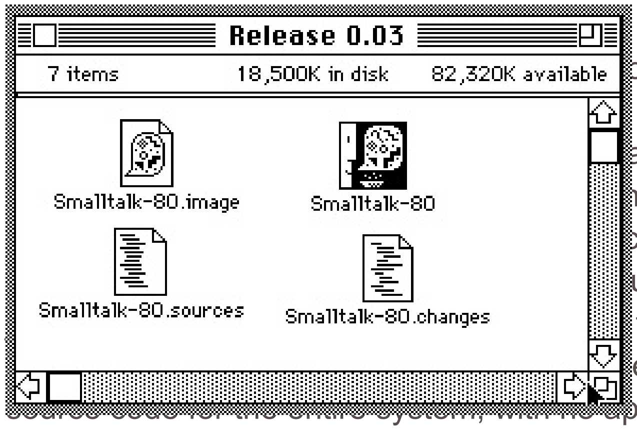
"Smalltalk
We are currently developing a version of Smalltalk tailored to the Macintosh computer. In the meantime, in response to requests from several universities, we have released a 'pre-product' version of the Smalltalk-80™ programming system running on the Macintosh and Macintosh XL computers. This is a fairly complete implementation of the Smalltalk-80 system, which we believe to be suitable for student projects and other preliminary experiments with Smalltalk."

Interested parties were instructed to contact Eileen Crombie at Apple. The cost of the program turned out to be $150.00, $50.00 for educational institutions, and it included (it was only available as) a site license. The system that was shipped was marked v0.2. Apparently there had been an earlier version (in March 1985) that ran only in the Mac OS a Macintosh XL (because regular Macs didn't have enough memory). I don't know how anybody found out about that one though, because it was not mentioned in the previous (Feb 1985) software supplement. Smalltalk-80 v0.2 was available starting in August of 1985. There were 2 configurations. The first, called Level 0, would run on a Macintosh 512K, but was only a subset of the entire system. Level 1 required 1 MB or more of memory, and could run on Macintosh XL or any Macintosh with 1 MB of memory. The Mac Plus, with 1 MB of memory standard, was released on Jan 16, 1986. Both level 0 and level 1 could be run with or without a hard disk but a hard disk was required to use the source file. The next version, v0.3, was dated October, 1986, and was the first version compatible with the hierarchical file system, and so would work properly if placed in a directory on a HFS-formatted hard disk. A final version, version 0.4, was dated June 9,1987 and was compatible with the Mac II. After that, there was no further word from Apple about Smalltalk-80 on the mac.

All Macintosh versions of Smalltalk-80 were the original version of Smalltalk-80, called version 1 at Xerox. By the time Apple was offering Smalltalk-80 for the Macintosh, Xerox was using version 2, and it is version 2 that is described in the classic books on the Smalltalk system. I suppose that Apple was restricted to distributing version 1 systems because that was the system for which they obtained a license back at the time of their participation in the experiment. There were small differences between the versions in some menus, and Apple offered a patch that could change the menus to better correspond to the system described in the book. At various times, Xerox or the spin-off Parc-Place Systems offered Smalltalk-80 as a commercial product on the Macintosh and other computers. Their assets are owned by Cincom, who still offers commercial and noncommercial versions of Smalltalk for Windows. They also have a free noncommercial version for Mac OSX, called VisualWorks. It’s a very complete and modern version of Smalltalk. Take a look at http://www.cincomsmalltalk.com.

Setting Up Smalltalk-80 in Mini vMac

Apple’s original Smalltalk-80 is available at various places on the web. So far as I can tell, all of them are exactly the same and so probably come from one source. It is either called Apple ST80 or Apple_Smalltalk_80_Betas. The first has only the v0.2 and v0.3 distributions. Probably you want to use v0.3, because if its HFS compatibility. The key files are Smalltalk-80.image, Smalltalk-80, Smalltalk-80.sources, and Smalltalk-80.changes. It is amazing that these are the only essential files for the entire Smalltalk-80 system. Because the sources file is about 1.4 MBytes, it could not fit onto a single 800K disk, and was distributed on several disks, with the sources file broken into

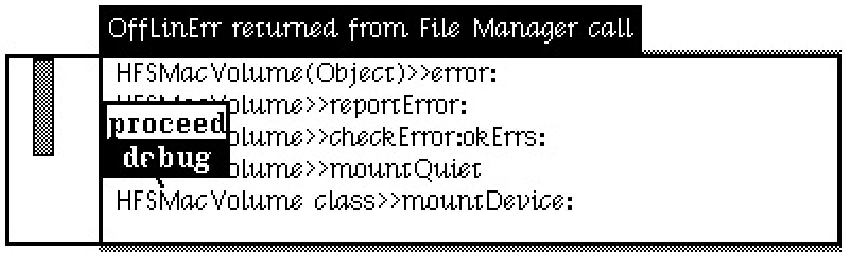


DivJoin utility (that was provided with
original source code for the Smalltalk-
It is actually optional, because
and reconstruct the source code. But
ments that can only be seen if it is
ce code for all the code you write, and
urce and changes files are text, with
to actually look at it with a text editor.
ese two files to show you the current
source code for the entire system, with no apparent difference between the original source and the code you have added.

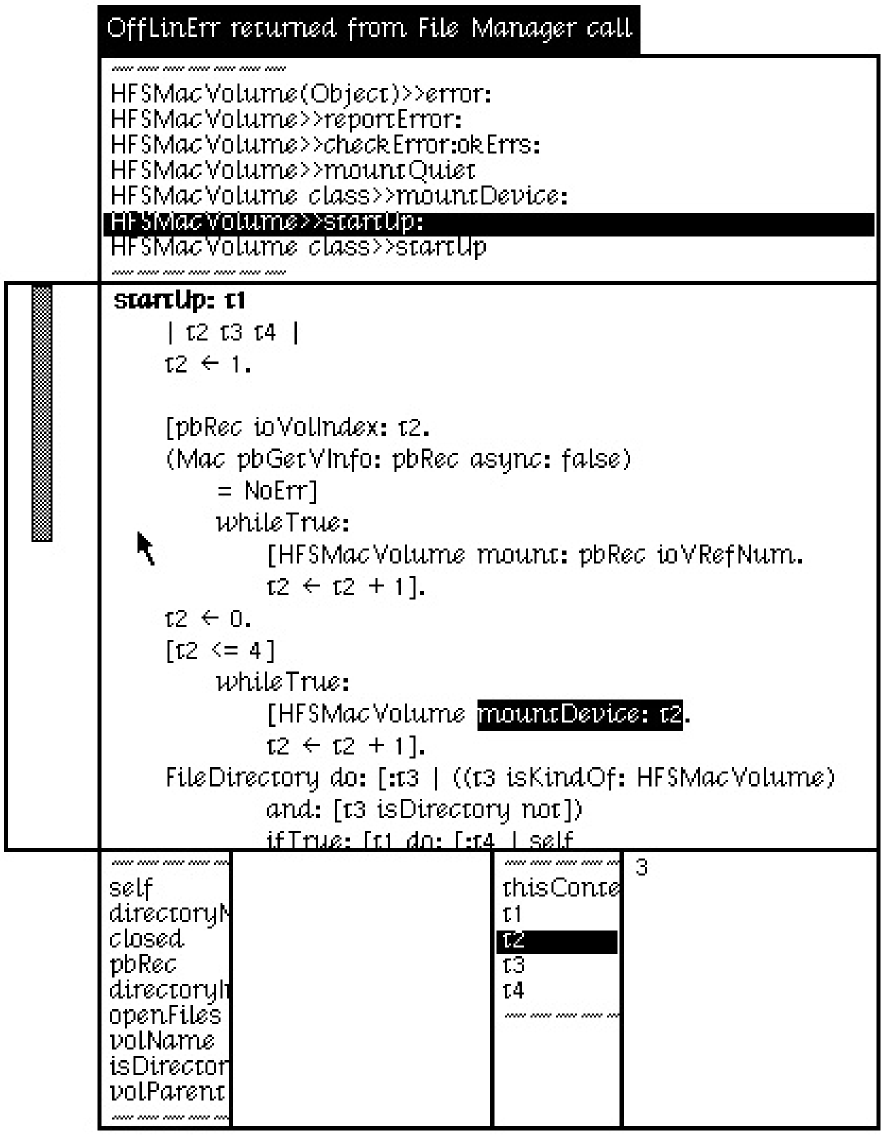
To start Smalltalk, double click the Smalltalk-80.image file, not the interpreter, Smalltalk-80. The interpreter is the only Macintosh executable file and it has to be there, but you never directly use it for anything. The image file is a complete representation of the entire state of the system. Everything is there, down to the positions of windows and current values of all the variables. If you update the image file before quitting Smalltalk, when you start it again it will appear just exactly as it did at the moment you saved the image.

Starting Smalltalk In Mini vMac Results in an Error

Double click the images file on a Mac Plus and everything works great. Starting it in a Mac Plus configuration of Mini vMac gives an error. This makes a great demonstration of how you can change anything in Smalltalk, and fix errors without missing a beat.

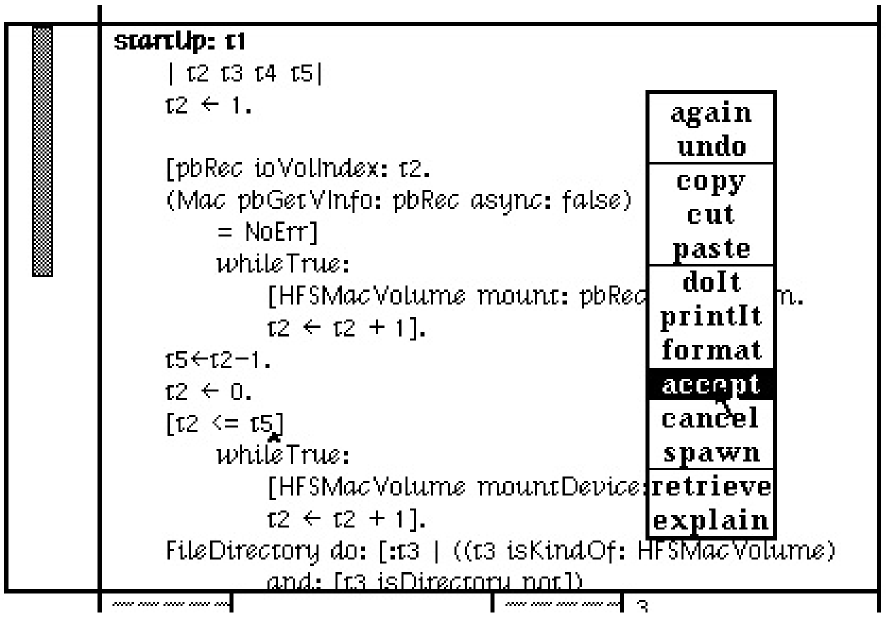

```
OffLinErr returned from File Manager call
HFSMacVolume(Object)>>error:
      olume>>reportError:
      olume>>checkError:okErrs:
      olume>>mountQuiet
HFSMacVolume class>>mountDevice:
```

Notice that Smalltalk windows have a tab-like title bar, and a scrollbar to the left of the window. The scrollbar hides and shows itself automatically to preserve desktop space. This window is a notifier that tells us about the error. It is a stack-walk of subroutine calls that led to and followed the error. Before we can do anything much, we have to choose to debug this error. Hold down the option key and click in the window to get a popup menu, and select debug. We could keep going ignoring the error by choosing proceed, but where's the fun in that? After we select debug, the system asks us to pick a place on the screen for the next window (experiment to appreciate the cool user interface device it uses for that) and we end up in a window that lets us see the source code for each of the calls in the call stack. We want to look at the first call, the one at the bottom. Probably, that's where the error happened, and the rest of the calls were all the code that got executed to cope with the error. This is a function (called messages in Smalltalk) that belongs to the class HFSMacVolume. That is, it is a message sent to the class, not to an object of that class. The message itself is called startup, so this was code that started up the class HFSMacVolume. If we don't fix this, we are definitely going to have trouble reading files in Smalltalk. HFSMacVolume sent the message startup to a particular



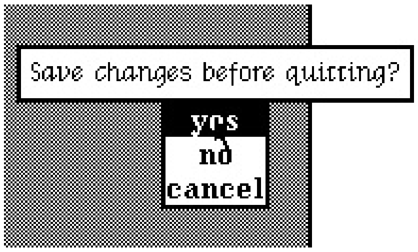
```
OffLinErr returned from File Manager call

HFSMacVolume(Object)>>error:
HFSMacVolume>>reportError:
HFSMacVolume>>checkError:okErrs:
HFSMacVolume>>mountQuiet
HFSMacVolume class>>mountDevice:
HFSMacVolume>>startUp:
HFSMacVolume class>>startUp

startUp: t1
    | t2 t3 t4 |
    t2 ← 1.

    [pbRec ioVolIndex: t2.
    (Mac pbGetVInfo: pbRec async: false)
        = NoErr]
        whileTrue:
            [HFSMacVolume mount: pbRec ioVRefNum.
            t2 ← t2 + 1].
    t2 ← 0.
    [t2 <= 4]
        whileTrue:
            [HFSMacVolume mountDevice: t2.
            t2 ← t2 + 1].
    FileDirectory do: [:t3 | ((t3 isKindOf: HFSMacVolume)
            and: [t3 isDirectory not])
            ifTrue: [t1 do: [:t4 | self

self          thisConte   3
directoryN    t1
closed        t2
pbRec         t3
directoryI    t4
openFiles
volName
isDirector
volParent
```

volume, in the next function in the chain. We've selected this function and we see the Smalltalk code for mounting a volume. This is in the heart of the operating system and the stuff we might usually be a little hesitant to mess with. Not in Smalltalk. We'll fix that thing right now. You can see that the error happened in the call to mountDevice: t2. t2 is a local variable. You can see there are several local variables created at the start of this function using the construct | t2 t3 t4 |. That's how you create local variables in Smalltalk. Notice they aren't typed. t2 starts out assigned the value 1 using the left-pointing arrow (shift-hyphen) as an assignment operator. Then it calls the pbGetVolInfo function (which is glue to a regular Mac File Manager call) for volume index of t2, then incrementing and doing it again until it doesn't work any more (it returns something other than noErr). At that point, t2 has the number of actual volumes on your mac plus 1. Inexplicably, that valuable number is thrown away and t2 is set to zero. Then the function sets out to mount the first 4 volumes on the system. On the Mac Plus, there are always 4 devices present. In Mini vMac, there may not be. There are only 2 in this case, and when it tries to open the third one, it gets an error. The values of variables are shown in the boxes at the bottom. At the time of the error, t2 equals 3. It can't mount device 3 because there isn't one. That's the problem. I suggest we add a local variable t5, and use it to store the number of devices present. Then we just mount those. Editing in Smalltalk is just like what you are used to on the mac. Note that lines of code are terminated with periods. Then option-click to get the menu that includes accept. Select accept, and the change in your code will be compiled and the system will immediately start to use your version of the function. Now we need to save an image and start Smalltalk again.



```
startUp: t1
    | t2 t3 t4 t5|
    t2 ← 1.

    [pbRec ioVolIndex: t2.
    (Mac pbGetVInfo: pbRec async: false)
        = NoErr]
        whileTrue:
            [HFSMacVolume mount: pbRec           m.
            t2 ← t2 + 1].
    t5←t2-1.
    t2 ← 0.
    [t2 <= t5]
        whileTrue:
            [HFSMacVolume mountDevice:
            t2 ← t2 + 1].
    FileDirectory do: [:t3 | ((t3 isKindOf: HFSMacVolume)
            and: [t3 isDirectory not])
```

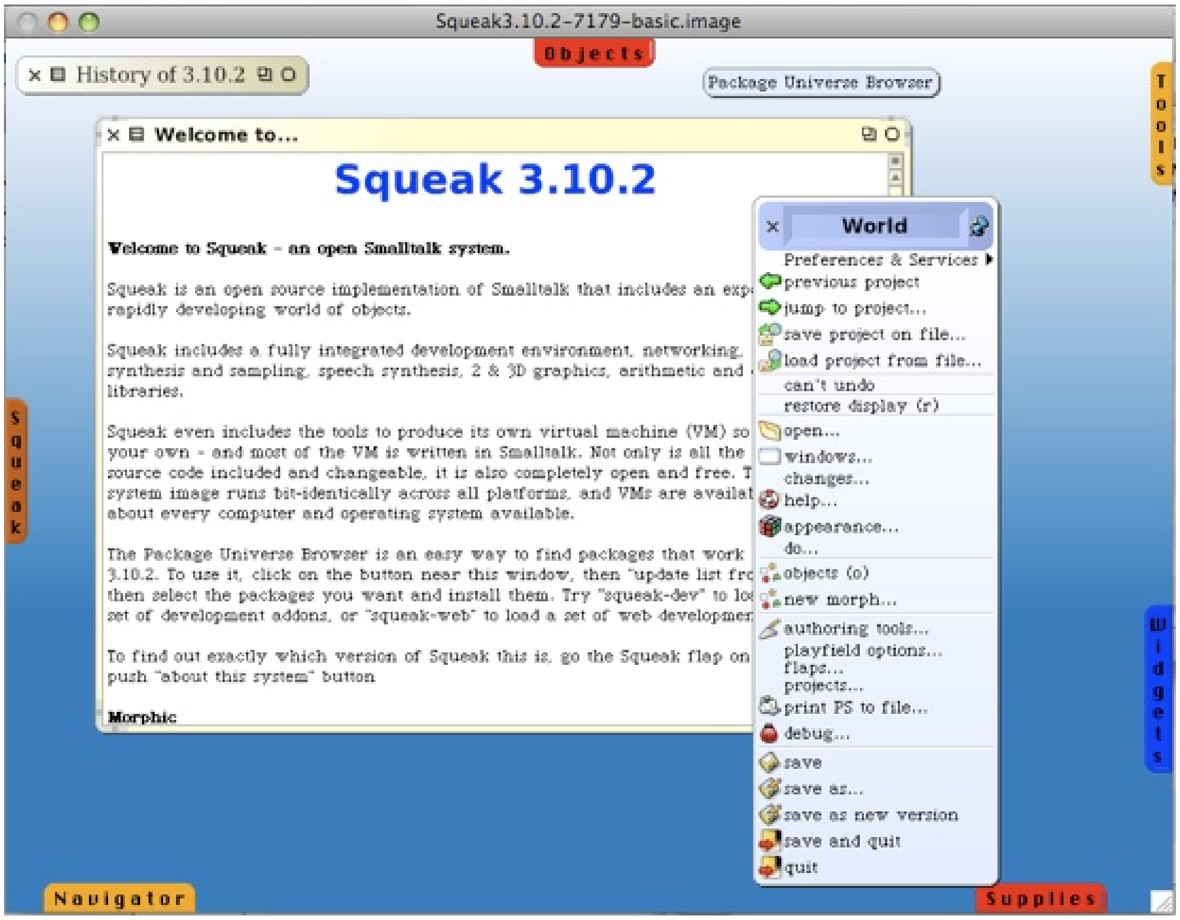
So option-click on the desktop, to get the desktop menu, and select quit. This familiar kind of menu appears. Say yes and Smalltalk will remember your changes in the image file. Start Smalltalk again and the error is gone. You can now open files, including the sources file. Actually, through all of the changes we made, we were working just in the Image file. We couldn't read the sources because we hadn't successfully initialized our file system. Isn't that Smalltalk really something?



Save changes before quitting?

Smalltalk on Modern Macs

Smalltalk is still around. Alan Kay, one of the original Smalltalk engineers at Xerox PARC, and later at Apple, and then at Disney, continued to work on the Apple version of Smalltalk. Yes, the same version we've just been fixing. He and others ported the interpreter to many operating systems to make a very portable version of Smalltalk. You can get it for free from Squeak Smalltalk. There have been a lot of changes though. Much good stuff has been added to Squeak, including support for sound, color, and the internet. On the other hand... Well take a look. Ugh. What happened to the simple elegance of Smalltalk? Oh well, don't



Squeak 3.10.2

Welcome to Squeak - an open Smalltalk system.

Squeak is an open source implementation of Smalltalk that includes an exp
rapidly developing world of objects.

Squeak includes a fully integrated development environment, networking,
synthesis and sampling, speech synthesis, 2 & 3D graphics, arithmetic and
libraries.

Squeak even includes the tools to produce its own virtual machine (VM) so
your own - and most of the VM is written in Smalltalk. Not only is all the
source code included and changeable, it is also completely open and free. T
system image runs bit-identically across all platforms, and VMs are availat
about every computer and operating system available.

The Package Universe Browser is an easy way to find packages that work
3.10.2. To use it, click on the button near this window, then "update list fro
then select the packages you want and install them. Try "squeak-dev" to lo
set of development addons, or "squeak-web" to load a set of web developme

To find out exactly which version of Squeak this is, go the Squeak flap on
push "about this system" button

Morphic

-- BG (basalgangster@macGUI.com)

< previous next >

Made on a Mac