

## **SQUEAK if You Love GUI**

By Jeff Read of World Tech Tribune 09/06/2002

In the 1980's, learning computers often meant learning how to program. Personal computers usually came with BASIC in ROM, so you could start programming right away, seconds after turning it on. In the back of many grade-school classrooms sat an Apple II machine, often running BASIC or some version of the popular LOGO language. It was fun to "teach" the on-screen turtle how to draw a square, star, or more complex shape by writing a small program with the instructions to do so.

With the proliferation of Microsoft Windows, learning computers means learning how to browse the Web, write e-mail in Outlook, or compose documents in Word or spreadsheets in Excel... and not much else. Ordinary users are simply not encouraged to expand their machine's capabilities through programming. In fact, the further users are distanced from programming, the better. The programming environments that are available are often goal-directed, with steep learning curves, and do not encourage the exploration and discovery that is at the heart of many people's interest in programming.

Enter Squeak. Squeak is a fully-functional Smalltalk environment that closely approximates the Smalltalk-80 standard. It was developed by Alan Kay, Dan Ingalls and associates as a platform on which to base various software research projects. But do not let the bright, colorful GUI or experimental nature of Squeak's design fool you; it is not a toy. It is a complete, robust Smalltalk development environment. It comes with a powerful graphical framework called Morphic; a 3D world builder called Alice; Web and E-mail clients; real-time sound and speech synthesis; a full suite of development tools for the Smalltalk language and environment; and a lot more. Squeak is so powerful that it's used to write its own virtual machine. A translator converts the Smalltalk into C for optimum performance, but the next Squeak VM is written, tested, and debugged using the existing one. Having been co-created by Alan Kay, the man who developed the Smalltalk language and laid the foundation upon which modern graphical user interfaces (including the Macintosh and Microsoft Windows) would be built, Squeak also offers a glimpse into the future of human-computer interactions. Both Apple and Disney have used Squeak to prototype multimedia applications with innovative interfaces.

Kids who aren't ready to handle working with Smalltalk can still use Squeak. By drawing on its surface like a Magna-Doodle, and then giving their creation commands using a graphical scripting system, they can construct "eToys", animated on-screen objects that follow their instructions. Graduating to working with the system directly in the Smalltalk language, however, is a step they should take: object-oriented programming is a marketable skill, and Smalltalk is the language that has defined what it means to be object-oriented for three decades. Programmers who know Smalltalk, when moving to languages like Java or C++, will find themselves in very familiar territory.

Junior high and high school students can create "Active Essays", or dynamic, interactive presentations with embedded text, graphics, and small programs. An example is Squeakland's Weasel Essay, which explains evolution and basic genetics by using a genetic algorithm to "mutate" a random group of letters into a phrase from Shakespeare. Examples of both eToys and Active Essays can be found at the Squeakland web site at <http://www.squeakland.org>.

Squeak is highly portable, existing on Mac, Windows, Linux, and most other popular platforms; and it has been released under an open source license for free download. It has a flourishing user community, with mailing lists and "Swikis" (dynamic workspaces on the Web, with a Squeak server powering each one) providing a forum where Squeakers can interact and exchange applications and extensions.

Squeak is a portable, extensible, fun, totally cool way to get started in programming. Even seasoned programmers like me can find much to like about it. It may be the perfect tool for putting the sense of wonder back into programming... and getting kids interested in technology beyond video games.

<http://www.worldtechtribune.com/worldtechtribune/asparticles/sv/sv09062002.asp>