Squeak and Croquet

Marcus Denker

Squeak and Croquet

Marcus Denker

4. Juni 2005

Zusammenfassung

Squeak allows kids of all ages to be creative with their computer. The goal of the Squeak Project is to build a system without constraints: It is used at schools, universities and in industry.

Squeak is an open System: It is implemented in Squeak itself, all parts are available for learning and hacking. The whole source code is available and can be changed while the system is running.

Squeak is available on the internet under a free license, it is highly portable and currently used on over 20 different platforms.

This talk will give an overview over the Squeak Project: From the eToy kids programming environment up to the Seaside system for professional web development.

The eToys make programming fun for children from around age 8. The talk will show how to build simple eToy programs and how Squeak is used at school. But even professional developers are using Squeak; The Seaside framework shows how the openness of Squeak can help to make developers more productive.

The last part of the talk will give a glimpse into the future: OpenCroquet. The Croquet project is building a revolutionary collaborative environment based on Squeak. It provides a scalable, peer-to-peer multiuser 3D environment that is completely open for exploration and makes novel ways for communication and interaction possible.

1 Installing the LinuxTag Demo

The first thing we need to do is installing Squeak on your system. Squeak has been ported to nearly everything, from handhelds to mainframes, or even game systems like the PS2. Not all of the ports are kept alive for all releases, but for a common system like Linux, MacOS or that-which-shall-not-be-named there should be no problem.

So go to squeak-ev.org and download Squeak for your system and unpack. The files we will need for running the demo are:

- SqueakV3.sources: The source code. All in one big lump.

- Squeak.exe / Squeak.app / squeak: This is the virtual machine.

The distribution files will contain two other files: .changes and .image. To get the contents of the LinuxTag Squeak Demo, we need to get the special LinuxTag version of these two files. They are available at

<http://www.squeak-ev.de/LinuxTag05>

After uncompressing the LinuxTag05.zip, you will find these two files:

- LinuxTag.image: all the Objects that make our demo, dumped into a file

- LinuxTag.changes: additional source code

Put these files in the directory with the other Squeak files, and you are ready to start.

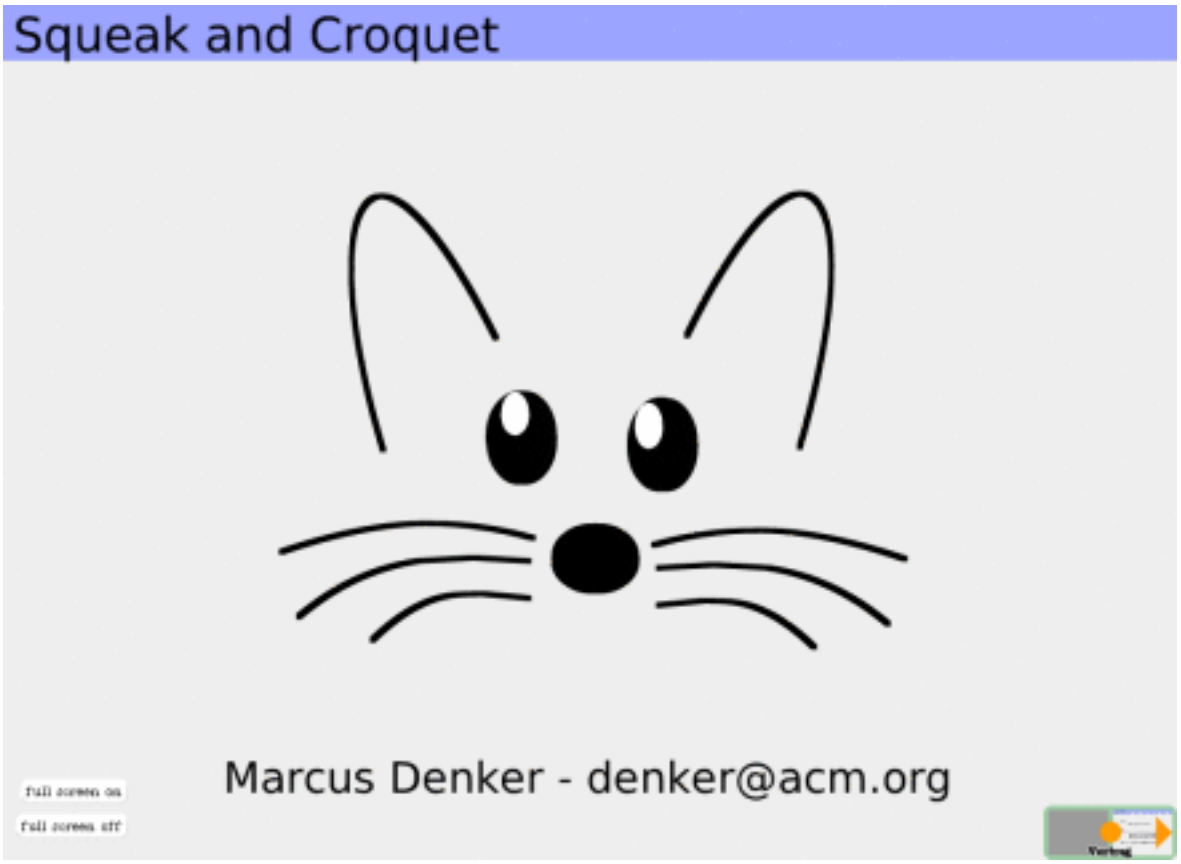
1.1 Start Up

Now we need to start Squeak. For that, just open the image file with the help of the vm. For linux, that would be something like:

./squeak LinuxTag05.image

MacOS/Win: drag-n-drop the image on the vm. Now the Squeak virtual machine will start and load the image back into memory. The image is really a image of the memory: Itquotes just a snapshot of the whole memory that the vm had allocated. It will start up exactly the way it was when I saved it.

So if I, e.g., have a editor open with the courser blinking at a certain spot, itquotes blinking right there. And this is portable across all systems: I saved on a mac, you load on Linux. Write once, run everywhere (but with Squeak, this really works! Not write once, debug everywhere like with that other system).



First slide of the demo

I saved the demo image in a way that you can start to look at the slides right now: When the window opens, itquotes around 1024x768 large, and shows a friendly mouse in the middle.

1.2 Navigating the Slides

At the lower right corner you see a thing called Thread Navigator. This handy object allows for easy navigation in the slides of the talk. Just click on the arrow, and the next slide appears. The Thread Navigator object will follow to the next slide. (He will really travel with you: He deletes himself from the old slide and hops over. So make sure not to loose him!).

Each slide is a Squeak Project. These projects are a bit like virtual desktops, but they can be saved to disk or send over the net.

You can make graphical links to projects, clicking on those will make the linked project active. To go back were you came from, you need to click on the background: A menu appears, with items for going back and lots of other entries, e.g. for opening tools.

So, now we are ready to explore Squeak.

2 What is Squeak

This question sounds simpler to answer than it is: Most Open Source projects try to recreate existing systems: Linux is Unix, Open Office is MS Office, Gimp is Photoshop... all of course much better and open. But it makes the goals of these project quite simple to describe: We want to build a free version of X.

Squeak is different: Once, there is not yet anything like it, the other problem: Squeak is lots of things at the same time. It's a programming language like Java, an IDE like Emacs or Eclipse, a media authoring system or a language for kids, depending on how you look at it.

The goal of Squeak is to create a true personal computing environment, for users of all ages, from kids in school, to students or researchers at universities , to normal „down to earth" developers.

Our number one commitment is to an exquisite personal computing environment. Imagine a system as immediate and tactile as a sketch pad, in which you can effortlessly mingle writing, drawing, painting, and all of the structured leverage of computer science. Moreover imagine that every aspect of that system is described in itself and equally amenable to examination and composition. Perhaps this system also extends out over the Internet, including and leveraging off the work of others. You get the idea - itquotes the Holy Grail of computer science. Where is Squeak Headed? – From squeak.org

2.1 History

The hero of our story is named Alan, the time is somewhere at the end of the sixties. Great time for being a computer science researcher, as the US military just pumped huge amounts of cash into basic computer research with the ARPA program. Lots of cool stuff was invented in that time, e.g. packet switching networks (ArpaNet), and Alan got to see some cool stuff: The first prototype of a flat panel display, one of the first handwriting recognizers and the drawing program SketchPad (according to some the most important program ever written). Papertquotes LOGO programing language for kids. And Moores Law.

So, he got an idea. A crazy one: Why not build a computer that is small enough to be carried around. It should have the size of a notebook, weighting around two pounds. It should have a graphical display, have enough memory to store a couple of books. It even could have a harddisc. A small one. Keep in mind that harddiscs at that time had the size of a refrigerator. A true Personal Computer. Build for kids. Codename: Dynabook.

Kids using the Dynabook (ca. 1968)

Today, this sounds a bit obvious, but back then, it was completely crazy. Indeed, the best way to predict the future is to invent it.

But the technical aspect is not the only interesting one: The idea was to build a computer for kids in the way that it would be a new kind of medium. For fun and learning. Not a word-processor, but an idea-processor.

The goal was to make this crazy dream a reality. Moores law gave the deadline: A sufficiently

fast computer would fit, according to Moores law, inside the required space in 1980. So Alan started to think about the software, at Xerox PARC.

The Operating System, GUI and development environment for the Dynabook was called Smalltalk. It was the first fully Object Oriented System, and the first GUI with real, overlapping windows. The famous story is that Steve Jobs did get a demo of that system....

Squeak is that Smalltalk System from the seventies, dusted off. And the Squeakers are the inventors of Smalltalk, Alan Kay of course, and Dan Ingalls and Ted Kaehler, both from the original Xerox PARC team.

Squeak was started in 1996, while the Squeak Team was at Apple. They moved on to Disney from there, and today, Alan is at HP Labs.

3 Squeak for Kids

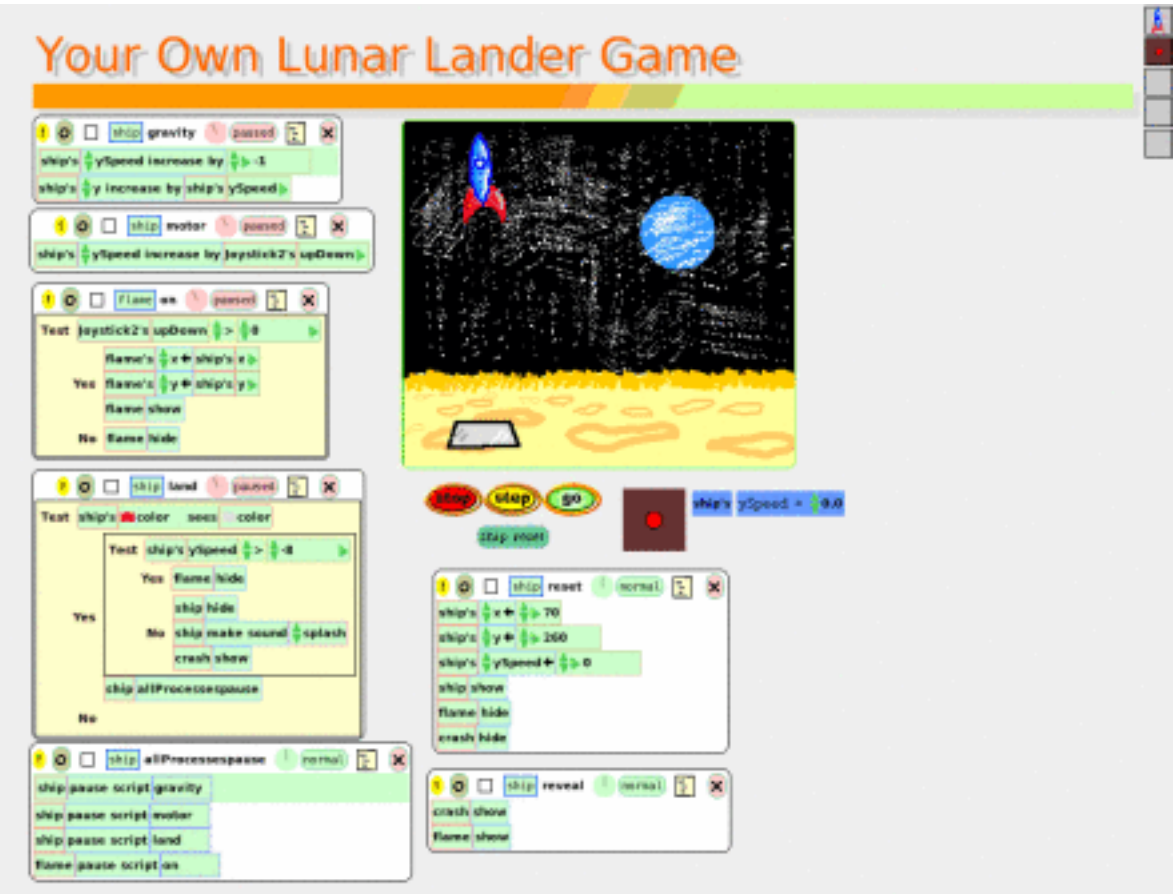
So, now go forward some slides to the Squeak for Kids project. A typical example of a project that kids will build with Squeak is the Car: A small painted car is programmed to be griven by a steering wheel and later, even has the ability to follow the road all by itself.

A Squeak Car

There is a good tutorial on the Squeakland website about how to build a car yourself:

<http://squeakland.org/school/drive_a_car/html/Drivecar12.html>

The Lunar Lander is an example of how far you can go with eToys: This was not done by a child, but with some guidance older kids could build such a game (and learn a lot).



Lunar Lander in eToys

There are a some other examples for eToy projects in the Linux image, many more can be found at http://squeakland.org. <http://squeakland.org/>

4 Squeak for Professional Development

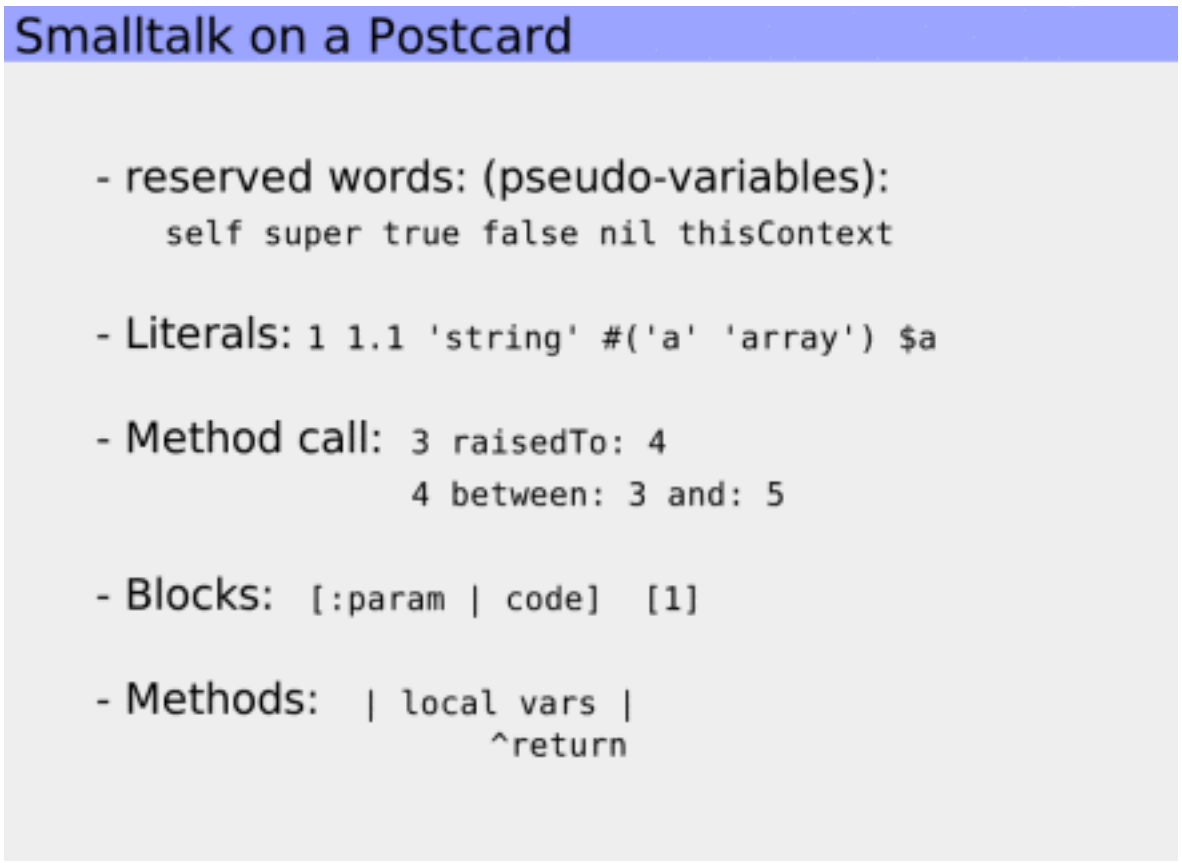
eToy is great for kids (and even for adults to learn programming). But the real hacker needs something that is a bit different. Squeak is completely written in itself, and for making that easy, it provides a object oriented programming language and a development environment.

The IDE provides a class browser for easy coding and a really good debugger. (With edit-and-continue, something that is starting to get a hot topic for both Java and .Net).

Squeak is actually used both in research and in industry to do real stuff. This is no toy! (Even if it looks like one).

4.1 The Language

The language of Squeak is really simple. But itquotes a bit different than the other ones you might know. This slide has all the syntax of the language:



Syntax of Squeak

Thatquotes it. In the slide, you can execute and print a statement by selecting it and pressing Alt-p (or Apple-p on the Mac).

The basic idea is that everything is an object, and you can make an object do something by sending a message to it. What an object does when receiving a message is described in a method. The method is defined in the class of the object.

So if you write:

1 sin

then you have an object (the number 1) and you send a message (sin). The object 1 is of class SmallInteger, and those SmallIntegers know how to calculate the sinus.

A real smalltalk introduction is outside of the scope of this paper, see

<http://www.iam.unibe.ch/~ducasse/FreeBooks.html>

for some books.

There is a good short introduction slide-show in the current 3.7 or 3.8 Squeak release images.

4.2 Seaside

Seaside is quite interesting. Avi Bryant was using Ruby for doing web development. Then he met some Squeakers at OOPSLA (the OO Conference).

And so he jumped ship from Ruby to Squeak and invented Seaside. Seaside is interesting, as it solves the problems that any web developer faces in novel ways. The target for Seaside are web applications, that is fairly complicated interactive websites (e.g. the typical banking site), not static pages.

Seaside allows the programmer to write the web application just the way she would write a normal GUI application. Letquotes see what Avi writes:

Seaside is a framework for developing sophisticated web applications in Smalltalk. Its most unique feature is its approach to session management: unlike servlet models which require a separate hand ler for each page or request, Seaside models an entire user session as a continuous piece of code, with natural, linear control flow - pages can call and return to each other like subroutines, complex sequences of forms can be managed from a single method, objects are passed by reference rather than marshalled into URLs or hidden fields - while fully supporting the backtracking and parallelism inherent to the web browser.

Seaside also features a callback-based event model, a transaction system for auto-expiring pages, strictly compliant XHTML generation, a system of reusable and embeddable UI components, and handy web-based development tools.

The Squeak image that drives the slides has a complete seaside installation. Yes, everything is there. Just point your browser to

<http://localhost:9090/seaside/presentation>

for a Seaside demo written in Seaside.

The Counter/MultiCounter are here:

<http://localhost:9090/seaside/counter>

<http://localhost:9090/seaside/multi>

A short article about seaside: <http://homepage.mac.com/svc/ADayAtTheBeach/>

It ends with:

> We quotediscoveredquote Seaside only recently. After just one day of experimenting with it we were convinced about the dramatic improvement in abstraction and productivity it offers. We were absolutely amazed at how easy it was to do the examples described here: much, much easier than it would be in any other framework that we know of. We believe Seaside could be a killer application for Squeak.

For more information about Seaside, see http://seaside.st <http://seaside.st/>

4.3 OpenCroquet

Now to the really interesting latest and greatest in Squeak. As with Squeak itself, we need a good quote from the developers:

> WHAT IF...
>
> ...we were to create a new operating system and user interface knowing what we know today, how far could we go? What kinds of decisions would we make that we might have been unable to even consider 20 or 30 years ago, when the current set of operating systems were first created?
>
> ...we could collaborate with one another in an online dimension to create or simulate anything we wanted to?
>
> ...we had the robustness of a 3D immersive technology, the diversity of the Internet, and the degree of social interaction we have in the real world?

The LinuxTag demo image does not contain a croquet installation. Croquet is (up to now) build on a slightly older version of Squeak than the one used to build the demo.

Croquet is work-in-progress. But there is a pre-release available for download. If you have a fast computer with a good 3D graphics card, give it a try.

All the needed files and a good introduction are available at http://coquetproject.org <http://coquetproject.org/>

5 Links

- http://squeak.org <http://squeak.org/> – the main Developer Site
- http://squeak-ev.de <http://squeak-ev.de/> - German Squeak Association
- http://squeakland.org <http://squeakland.org/> - Squeak for Kids
- http://croquetproject.org <http://croquetproject.org/> - Croquet
- http://seaside.st <http://seaside.st/> - Seaside

6 Books

<http://www.iam.unibe.ch/~ducasse/FreeBooks.html>

7 Film

http://squeakersfilm.org <http://squeakersfilm.org/> - Documentation about Squeak