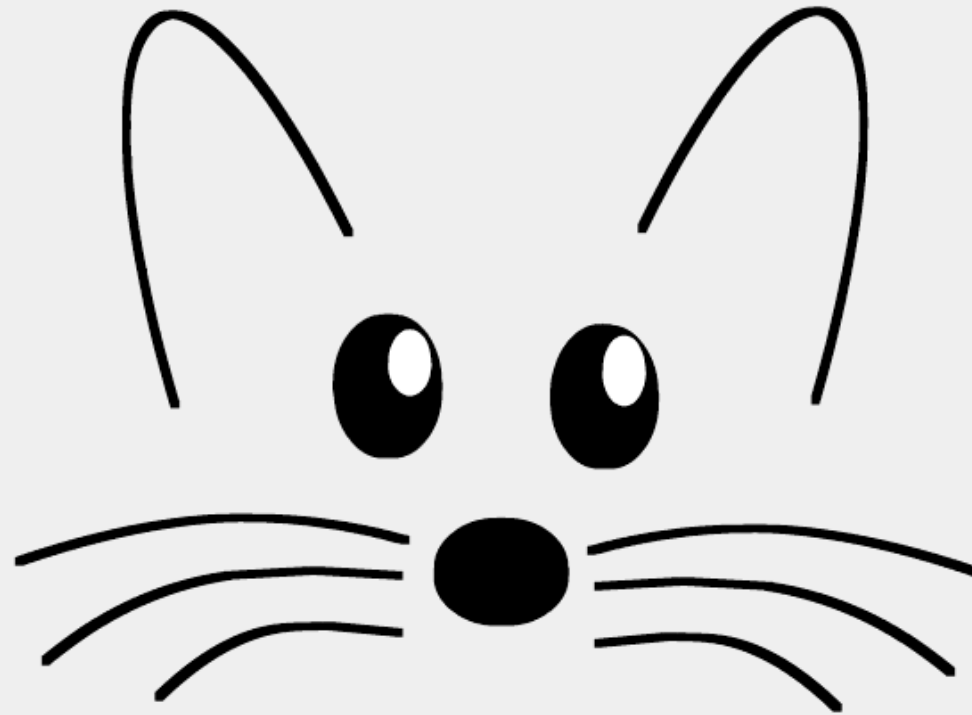


# Squeak and Croquet

- Please note:
  - The following are just screenshots from the demo made with Squeak
  - For the real experience: Download the demo image from: <http://squeak-ev.de/21c3>

# Squeak and Croquet



Marcus Denker - [denker@acm.org](mailto:denker@acm.org)

full screen on

full screen off



# Overview

Part I:

The Squeak Project

Part II:

Squeak for Kids (Small Hackers)

Squeak for older Hackers

Part III: Beyond Squeak: Croquet



# Part I: The Squeak Project

- What is Squeak ?
  - > Examples
  
- History
  - > Alan Kay's Dynabook



# What is Squeak?

- 1 Multimedia Authoring
2. Programming for Children
3. Operating System?
4. Development Environment
5. A Community



## Computers, Networks and Education

*Globally networked, easy-to-use computers can enhance learning,  
but only within an educational environment that  
encourages students to question "facts" and seek challenges*

by Alan C. Kay

The physicist Murray Gell-Mann has remarked that education in the 20th century is like being taken to the world's greatest restaurant and being fed the menu. He meant that representations of ideas have replaced the ideas themselves; students are taught superficially about great discoveries instead of being helped to learn deeply for themselves.

In the near future, all the representations that human beings have invented will be instantly accessible anywhere in the world on intimate, notebook-size computers. But will we be able to get from the menu to the food? Or will we no longer understand the difference between the two?

ALAN C. KAY has been a Fellow of Apple Computer Inc. since 1984. Before joining Apple, he was a founder and fellow of the Xerox Palo Alto Research Center and, later, chief scientist of Atari. One of the pioneers of personal computing, he is the original designer of the overlapping-window user interface and Smalltalk, the first completely object-oriented language.

Worse, will we lose even the ability to read the menu and be satisfied just to recognize that it is one?

There has always been confusion between carriers and contents. Pianists know that music is not in the piano. It begins inside human beings as special urges to communicate feelings. But many children are forced to "take piano" before their musical impulses develop; then they turn away from music for life. The piano at its best can only be an amplifier of existing feelings, bringing forth multiple notes in harmony and polyphony that the unaided voice cannot produce.

The computer is the greatest "piano" ever invented, for it is the master carrier of representations of every kind. Now there is a rush to have people, especially school-children, "take computer." Computers can amplify yearnings in ways even more profound than can musical instruments. But if teachers do not nourish the romance of learning and expressing, any external mandate for a new "literacy" becomes as much a crushing burden as being forced to perform Beethoven's sonatas while having no sense of their beauty. Instant access to the world's information will probably have an effect opposite to what is hoped: students will become numb

instead of enlightened.

In addition to the notion that the mere presence of computers will improve learning, several other misconceptions about learning often hinder modern education. Stronger ideas need to replace



**STUDENTS at the Open School: Center for Individualization, in Los Angeles, are creating a dynamic simulation of ocean life (right) and doing math (above) with the help of Macintosh computers, which are set unobtrusively into the desks. In the Open School, which already had a strong curriculum before it obtained computers, the machines do not substitute for teachers. There are thought of as "just another material", like books, paints and clay, that can support the**



ALAN C. KAY has been a Fellow of Apple Computer Inc. since 1984. Before joining Apple, he was a founder and fellow of the Xerox Palo Alto Research Center and, later, chief scientist of Atari. One of the pioneers of personal computing, he is the original designer of the overlapping-window user interface and Smalltalk, the first completely object-oriented language.

frontpage Overview partI whaIsSqueak Multimedia CompNet video and audio Presentation

SqueakLanguage Class Library Portable Sam'sFaceBall Development Community history

Dynabook kid programming drive a car The Big Race New Lunar Lander partII syntax

examples seaside Seaside demo Links Croquet

etoy proje etoy projects





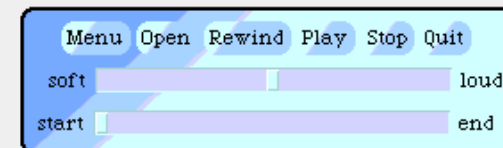
# Squeakers

A documentary film about Alan Kay, computers, and kids.



Dr. Alan Kay, known as "the Father of the Personal Computer," has devoted his life to the subject of computers and early childhood education. Thirty years of computer innovation and a deep understanding of how children learn is behind his Squeak programming language - and the new math and science projects shown here with fourth to sixth graders.

A 44-minute Hi Def film of interest to parents, educators, and anyone concerned about how children learn, SQUEAKERS features Alan Kay, Jerome Bruner, Seymour Papert, and a special appearance by Quincy Jones.



<http://squeakersfilm.org>

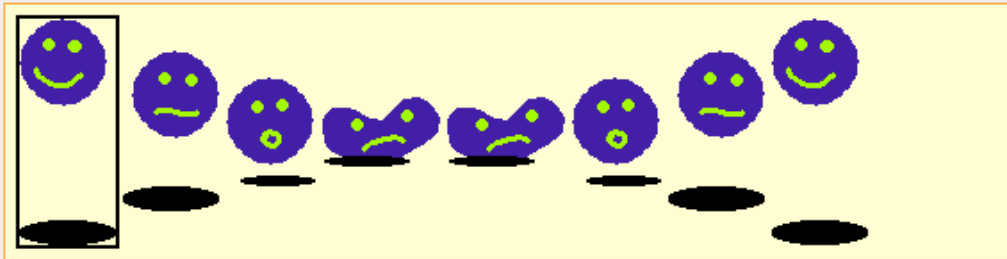
# What is Squeak?

- 1 Multimedia Authoring
2. Programming for Children
3. Operating System?
4. Development Environment
5. A Community



# Kid Programming

```
Sketch script1 - paused  
holder's cursor increase by 1  
Sketch look like holder's playerAtCursor
```



Sam L.



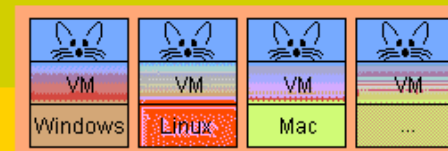
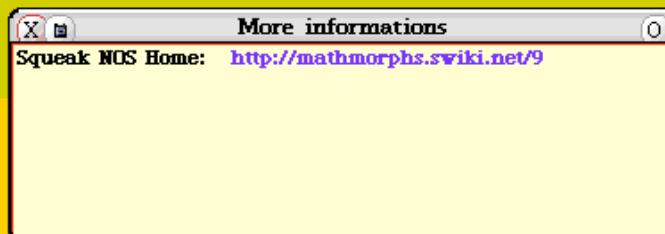
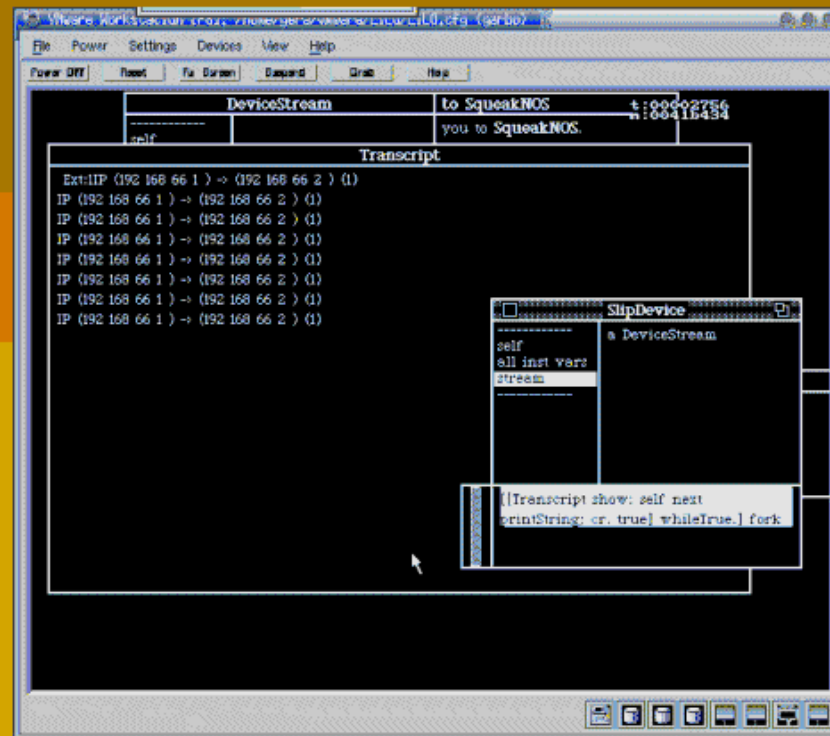
# What is Squeak?

- 1 Multimedia Authoring
2. Programming for Children
3. Operating System?
4. Development Environment
5. A Community



# Squeak NOS

Squeak NOS (No operating system) project is trying to get rid of the OS under Squeak. All needed functionality is implemented in Smalltalk. The idea is to write a really tiny kernel to boot Squeak



# What is Squeak?

- 1 Multimedia Authoring
2. Programming for Children
3. Operating System?
4. Development Environment
5. A Community



# Development Environment

**System Browser: Float**

Kernel-Chronology	Complex	-- all --	hasContentsInExpl
Kernel-Chronology	<b>Float</b>	arithmetic	isFloat
Kernel-Classes	Fraction	mathematical func	isInfinite
Kernel-Classes-Tes	Integer	comparing	isLiteral
Kernel-Contexts	LargeNegativeInte	<b>testing</b>	isNaN
Kernel-Magnitudes	LargePositiveInteg	truncation and rou	isPowerOfTwo
Kernel-Methods	instance ? class	converting	<b>isZero</b>

browse senders implementors versions **inheritance** hierarchy inst vars class vars source

**isZero**  
^self = 0.0

**a PasteUpMorph(131)**

self	#(a SystemWindow(70) a
all inst	SystemWindow(449) a
bound:	RectangleMorph<Rectangl
owner	e>(2560))
<b>submo</b>	
fullRo	

# What is Squeak?

- 1 Multimedia Authoring
2. Programming for Children
3. Operating System?
4. Development Environment
5. A Community





## Community

- Developers:
  - Mailinglist (~1000 Members)
  - SqueakParty
- Teachers and Parents:
  - Squeakland.org
- Germany:
  - Squeak e.V.  
(German Squeak Association)

# Part I: The Squeak Project

- What is Squeak ?
  - > Examples
  
- History
  - > Alan Kay's Dynabook



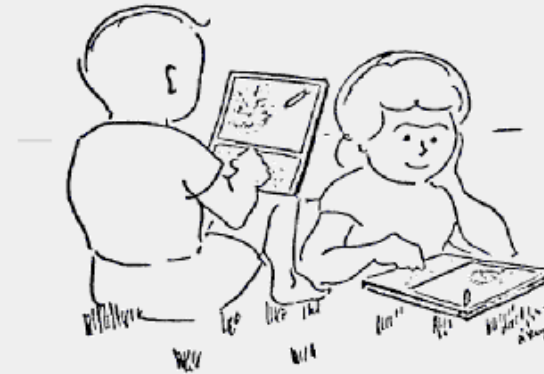
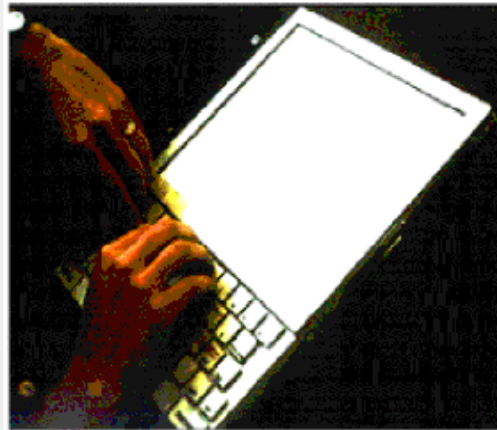
# Historical

- Alan Kay 1968: The Dynabook  
"A Dynamic Medium for Creative Thought"
- Xerox PARC 1970 - 1980
  - GUI
  - Smalltalk
- Squeak started 1996  
(Apple, Disney, now HP)
- Now working on Croquet



# The Dynabook

Alan Kay: "Ideaprocessor Vs. Wordprocessor"



## Part II: Overview

1. Squeak for Kids
  - eToys introduction
  
2. Squeak for Hackers:
  - Smalltalk in 2 minutes
  - no real time for: Seaside....



# Squeak for Kids

## Simple graphical scripting: eToys

Drive a Car:



Lunar Lander



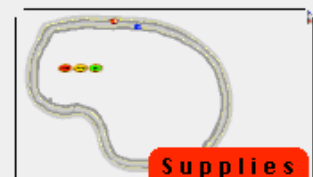
# Drive a Car



! CAR script1 normal

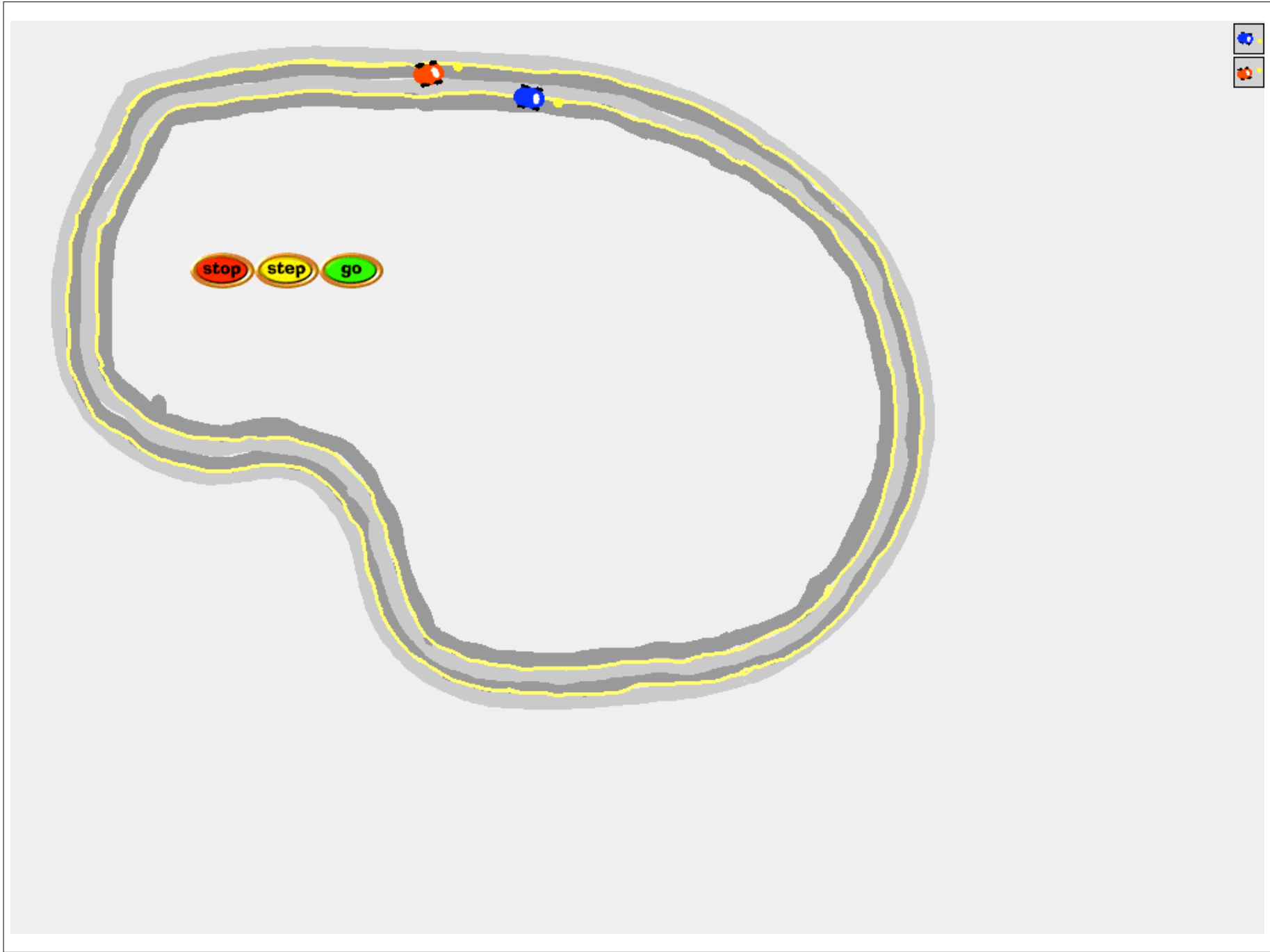
CAR forward by 5 ▶

CAR turn by 5 ▶



Navigator

Supplies





# Your Own Lunar Lander Game

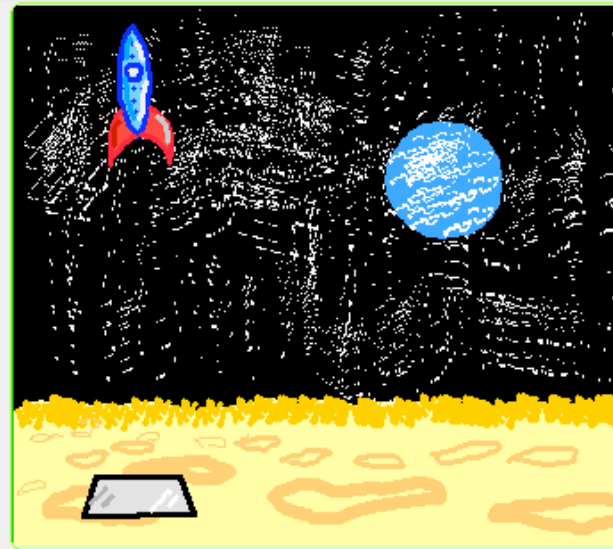
```
! [O] [ ] ship gravity [ ] paused [ ] [X]
ship's ySpeed increase by -1
ship's y increase by ship's ySpeed
```

```
! [O] [ ] ship motor [ ] paused [ ] [X]
ship's ySpeed increase by Joystick2's upDown
```

```
! [O] [ ] flame on [ ] paused [ ] [X]
Test Joystick2's upDown > 0
  Yes flame's x ← ship's x
  Yes flame's y ← ship's y
  flame show
  No flame hide
```

```
! [O] [ ] ship land [ ] paused [ ] [X]
Test ship's color sees color
  Test ship's ySpeed > -8
    Yes flame hide
    ship hide
    No ship make sound splash
    crash show
  ship allProcessespause
  No
```

```
! [O] [ ] ship allProcessespause [ ] normal [ ] [X]
ship pause script gravity
ship pause script motor
ship pause script land
```



stop step go [ ] ship's ySpeed = 0.0  
ship reset

```
! [O] [ ] ship reset [ ] normal [ ] [X]
ship's x ← 70
ship's y ← 260
ship's ySpeed ← 0
ship show
flame hide
crash hide
```

```
! [O] [ ] ship reveal [ ] normal [ ] [X]
crash show
flame show
```



# Squeak: The Language

- Completely object oriented:  
Everything is an object
- Virtual machine, bytecode
- Gargage collector
- Huge class library

100 factorial



# The Class Library

## \* 2D-Graphics

- TrueType
- Flash
- GIF, PNG, JPEG, PCX, XBM
- Video: MPEG und MJPEG

## \* 3D-Graphics

- 3D-Graphics (OpenGL)
- VRML import

## \* Sound:

- Recording, Playback
- ADPCM, AIFF, GSM, MuLaw,
- MP3 decoding
- FM-Synthesis
- MIDI

## \* Networking:

- HTTP, FTP, POP, SMTP
- Mailreader, Webserver
- Groupware-features:
- Chat (voice, text)

## \* Misc:

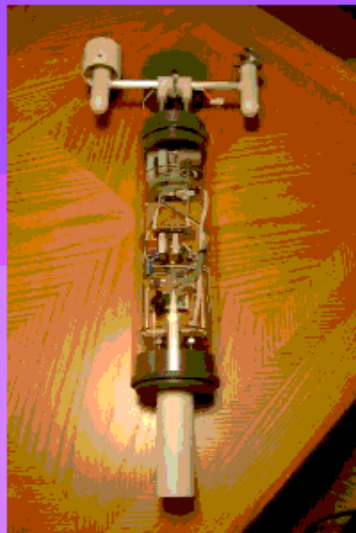
- Digital Signatures
- Compression: ZIP, gzip
- Postscript export

# Portable

- Windows 2000
- Windows NT
- Windows 95/98
- Windows XP
- Windows CE
- DOS
- Macintosh
- OS/2
- Acorn
- BeOS
- Linux/i386
- Linux/PowerPC
- Linux/Sparc
- SunOS
- Solaris
- SCO System V
- MacOS 9
- MacOS X
- DigitalUnix/Alpha
- NetBSD/Sparc
- NetBSD/i386
- Psion 5
- Zaurus



## Autonomous Controller for Microseeker built with Squeak



**More informations**

HUV is a small, privately-owned research & development company that focuses on very small autonomous underwater vehicles (AUVs), and the software that runs them.

HUV is currently involved in the design and construction of MicroSeeker, a simple, small, proof-of-concept AUV. There are two distinct systems in MicroSeeker. The first is the low-level, real-time data acquisition and control system. This is run by a network of PIC microcontrollers that run PIC/Smalltalk. The second is the high-level autonomous control system. That level will be running on a PDA device called a Helio, and it will be written in Squeak.

There is also a simulator written in Squeak.

More Infos: <http://www.huv.com/>

## Smalltalk on a Postcard

- reserved words: (pseudo-variables):  
self super true false nil thisContext
- Literals: 1 1.1 'string' #('a' 'array') \$a
- Method call: 3 raisedTo: 4  
4 between: 3 and: 5
- Blocks: [:param | code] [1]
- Methods: | local vars |  
^return



## Examples

```
(1<2) ifTrue: ['true'] ifFalse: ['false']
```

```
 #(1 2 3 4) + 3
```

```
 #(1 2 3.2 4) select: [:each | each > 2]
```

```
 #(1 2 3.2 4) collect: [:each | each class]
```

```
 #(1 1.1 'hallo') do: [:each | each class browse]
```

```
 #(1 1.1 'hallo')
```



## Seaside: Web Applications with Squeak

- Models an entire user session as a continuous piece of code
- Pages can call and return to each other like subroutines
- Complex sequences of forms can be managed from a single method
- Back-button works!



# Seaside Demo

Counter/MultiCounter:

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

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

Sushi Shop:

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

Presentation:

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



## Links....

Squeak for Kids:

<http://squeakland.org>

Squeak Hacker:

<http://squeak.org>

Squeak Germany:

<http://squeak-ev.de>





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?

