Content

3 Parts:

Part I:

The Squeak Project

Part II:

Squeak: The Language

Part III:

Squeak: The IDE

About me

Name: Marcus Denker
Student (Computer Science / Karlsruhe)

Squeaker since 1998

Projects:

* Just-In-Time Compiler
* German Squeak Association
* Test Coordinator / Harvester

Disclaimer

-> I'm No Ruby Programmer

-> I know nothing about Ruby IDEs:
(So please tell me "we can do that")

-> So: This will be a Squeak Demo

Part I: The Squeak Project

Content:

* What is Squeak ?
  -> Examples

* History
  -> Alan Kay's Dynabook

* Squeak for Kids
  -> Examples

What is Squeak?

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

Multimedia

* Text and Pictures

* Presentations

* Video

* 3D

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 represent- ations of ideas have replaced the ideas themselves; students are taught super- ficially about great discoveries instead of being helped to learn deeply for them- selves.

In the near future, all the representations that human beings have invented will be instantly accessible anywhere in the world on intimate, notebook-size compu- ters. But will we be able to get from the menu to the food? Or will we no longer understand the difference between the two? 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 commun- icate 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 prob- ably 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 children's activities. In the next few years notebook-size

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. Kay has worked with children for most of his career because, he says, "the media that powerfully shape our ways of thinking must be made accessible as early in life as possible." His interests outside of computing include musical performance and instrument design and "trying

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. Kay has worked with children for most of his

Übersicht Vortrag

save

Anfang

startup

frontpage

Inhalt

ueber mich

Inhalt Teil I

Was ist Squeak?

multimedia

Sam'sFaceBall

Programmiersprache

portable

Klassen

Entwicklungsumgebung

community

Historisches

Dynabook

Squeak fuer Kinder

Auto

The Big Race

New Lunar Lander

NudiBranch3D1

Inhalt Teil III

Version 2.4

Croquet

Verein

SqueakCD

Statistik

Fragen

Teil II

Teil II

Squeak

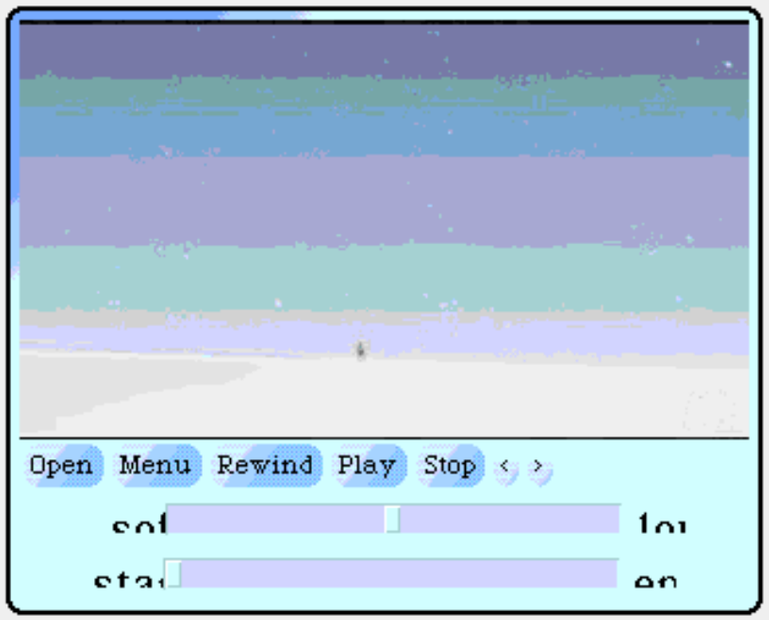
Tools

Navigator

Widgets

Supplies

Video and Audio



Squeak-Alice ReadMe

How do I create new Wonderlands?
This Wonderland was created by evaluating:
    Wonderland new.
This creates a new Wonderland, along with a camera window looking into the Wonderland, a few controls, and a scripting editor.

Speed and Color
Balloon3D can render 3D into 4-, 8-, 16- or 32-bit color. Use the screen manu to try them out. Also, the Wonderland viewing window can be made any size -- use cmd-click to get the yellow morphic resize handle. Larger windows will exhibit a lower frame rate.

How do I add Actors to my Wonderland?
Currently Wonderlands prefer to create Actors from Alice .mdl files. As a service to the community, the Alice team has agreed to make

Undo | Reset | Quit

```
scene
    ground
    light
    camera
    bunny
    bunny
        drum
        head
            glasses
            rightEar
            leftEar
        body
            leftLeg
            leftArm
                mallet
            rightArm
                mallet
            rightLeg
```

Scr: Actor Quick Ref.

"

Welcome to Squeak-Alice,

an implementation of the Alice 3D authoring tool (http://www.alice.org) in Squeak. With Squeak-Alice you can build interactive 3D worlds, even if you don't know anything about 3D graphics. This little demonstration should give you an idea of what Squeak-Alice can do; read through the comments and follow the directions. Good luck and have fun!"

"Let's start out by moving the bunny. Put the cursor in the line of code below and hit command-D (alt-D for PC users)."

bunny move: forward.

"The bunny moved forward 1 meter. You can also specify how far to move actors. Click the **undo** button to move the bunny back. It is the green button at the upper left. Do this after every action, then evaluate the next line."

bunny move: up distance: 1/2.

"Note that by default everything in Squeak-Alice animates over 1 second. You can change this by specifying a duration for animations."

bunny move: forward distance: 2 duration: 4.

Squeak

Tools

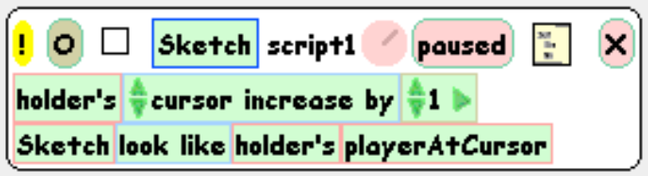
Navigator

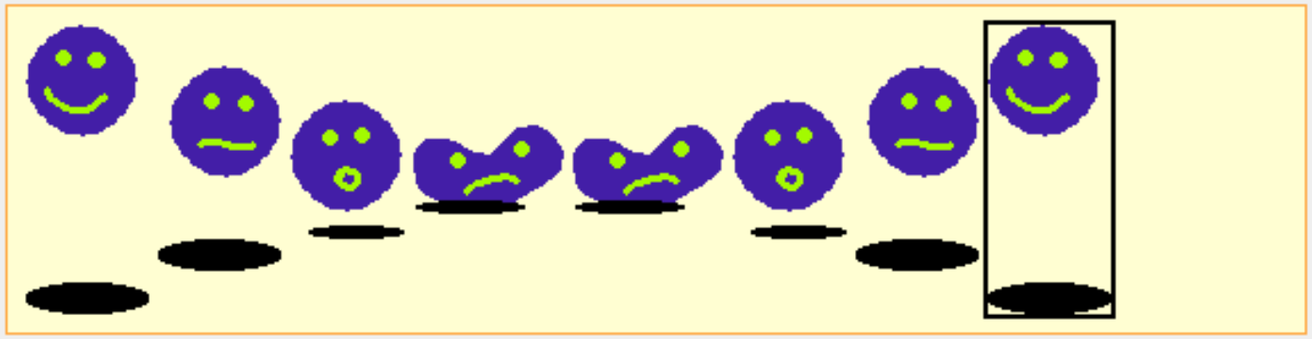
Widgets | Supplies

What is Squeak?

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

Kid Programming





Sam L.

What is Squeak?

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

Squeak: The Languange

* completly objectoriented:
  Everything is an Object
* virtual machine, Bytecode
* Garbage Collector
* Huge Classlibrary
* Example:

100 factorial

The Library

* 2D-Graphics
  - TrueType
  - Flash
  - GIF, PNG, JPEG, PCX, XBM
  - Video: MPEG und MJPEG

* 3D-Graphics
  - 3D-Graphics Subsystem
  - VRML import

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

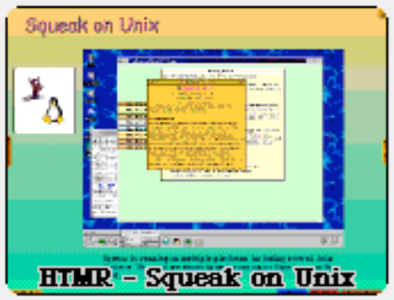
* Networking:
  - HTTP, FTP, POP, SMTP
  - Mailreader, Webserver
  - Groupware-features:
  - Chat (voice, text)

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

Portable

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

- Embedded Squeak
- Netscape Plugin
- IE-Plugin (Squeak as ActiveX Control)
- Multilanguage Squeak
- Bootable Squeak
- Goodies

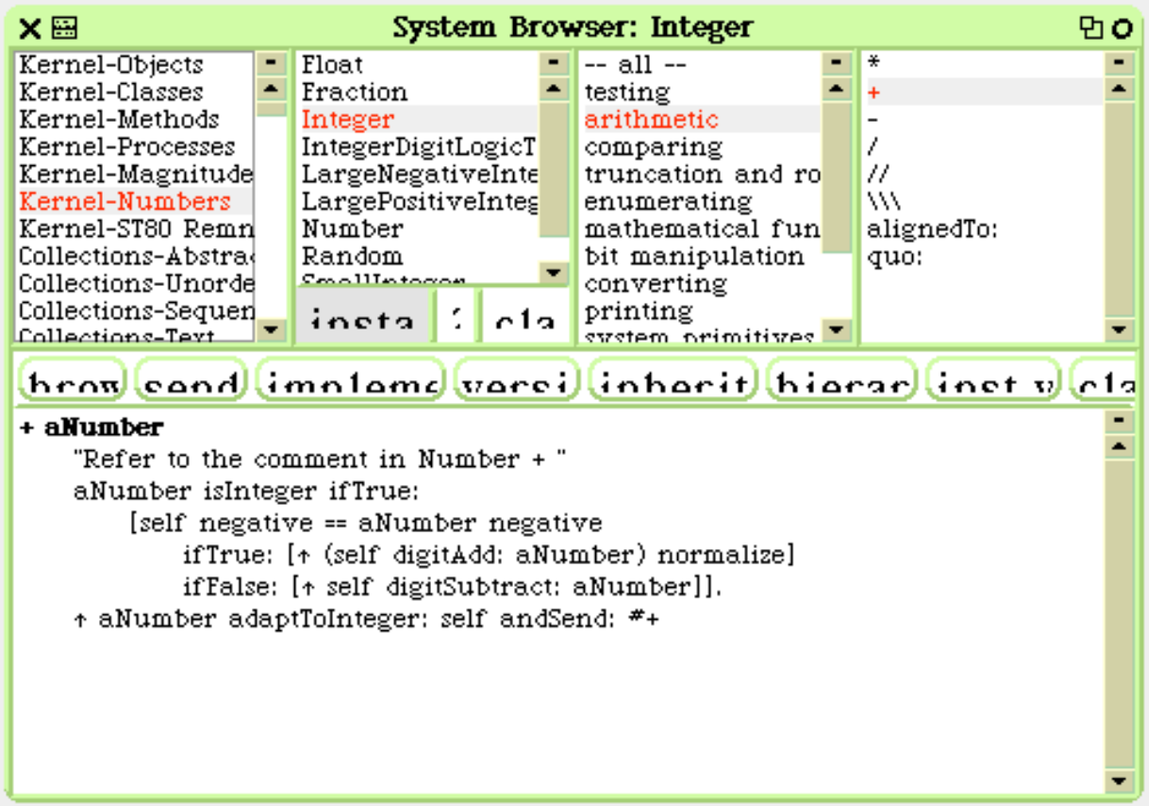


What is Squeak?

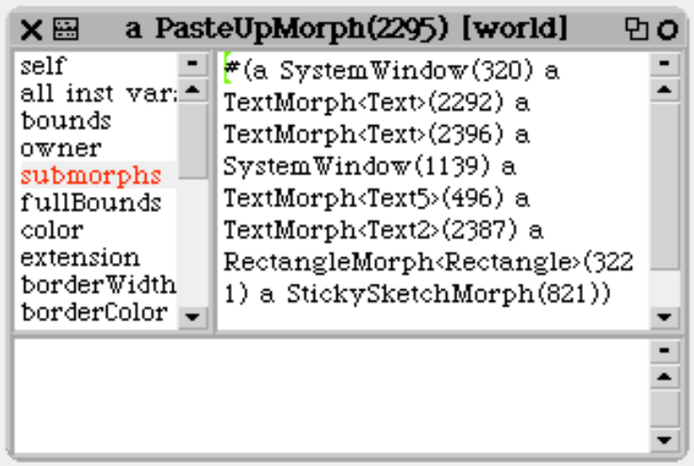
1. Multimedia Authoring
2. Programming for Children
3. Operating System ?
4. Programming Language
5. Development Environment
6. A Community

Development Environment

* ClassBrowser:



* Inspector:



What is Squeak?

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

The Squeak Community

* Developers:
  - Mailinglist (~ 1000 members)
  - SqueakNic

* Teachers and Parents:
  - Squeakland.org and mailinglist

* Germany:
  - Squeak Deutschland e.V.
    (German Squeak Association)

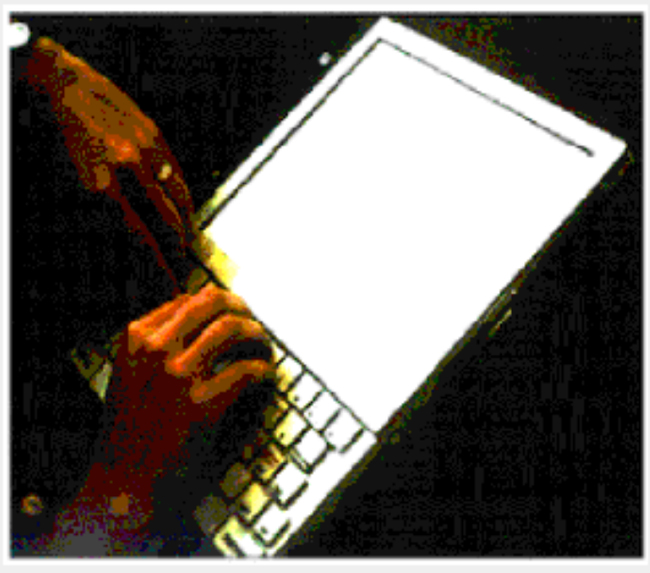
Historical

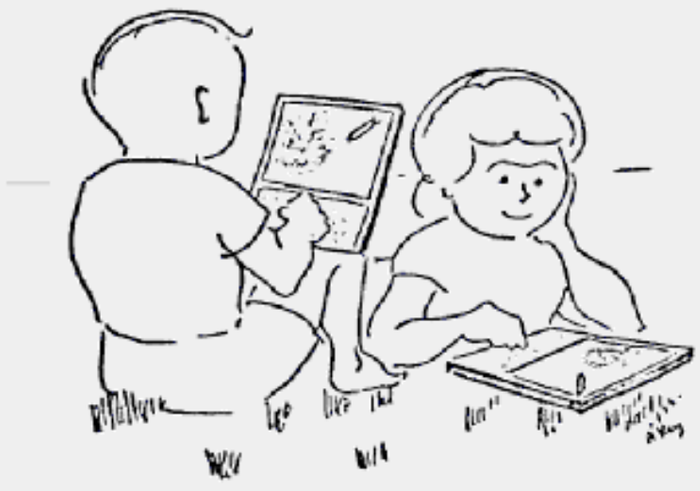
* Alan Kay 1968: The Dynabook
"A Dynamic Medium for Creative Thought"

* Xerox PARC 1970 - 1980
  - GUI
  - Smalltalk

* Squeak started 1996
  (Apple, Disney, now HP)

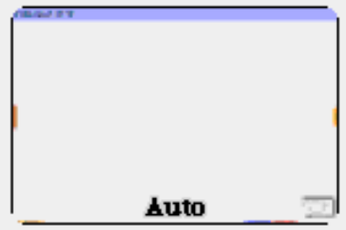
Alan Kay: "Ideaprocessor Vs. Wordprocessor"



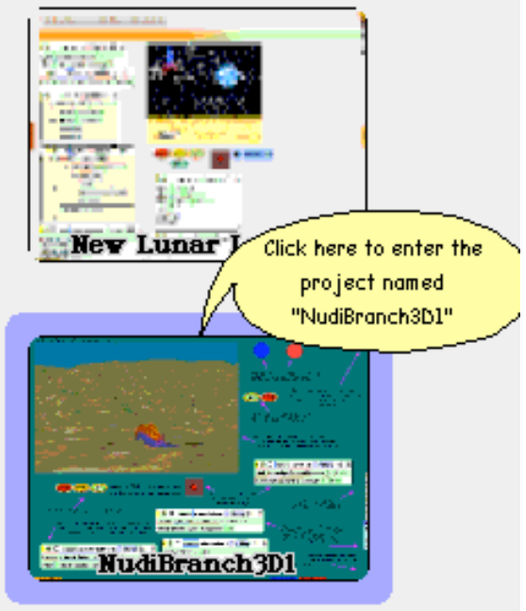


Squeak for Kids

* simple, graphcal
  scripting: eToys

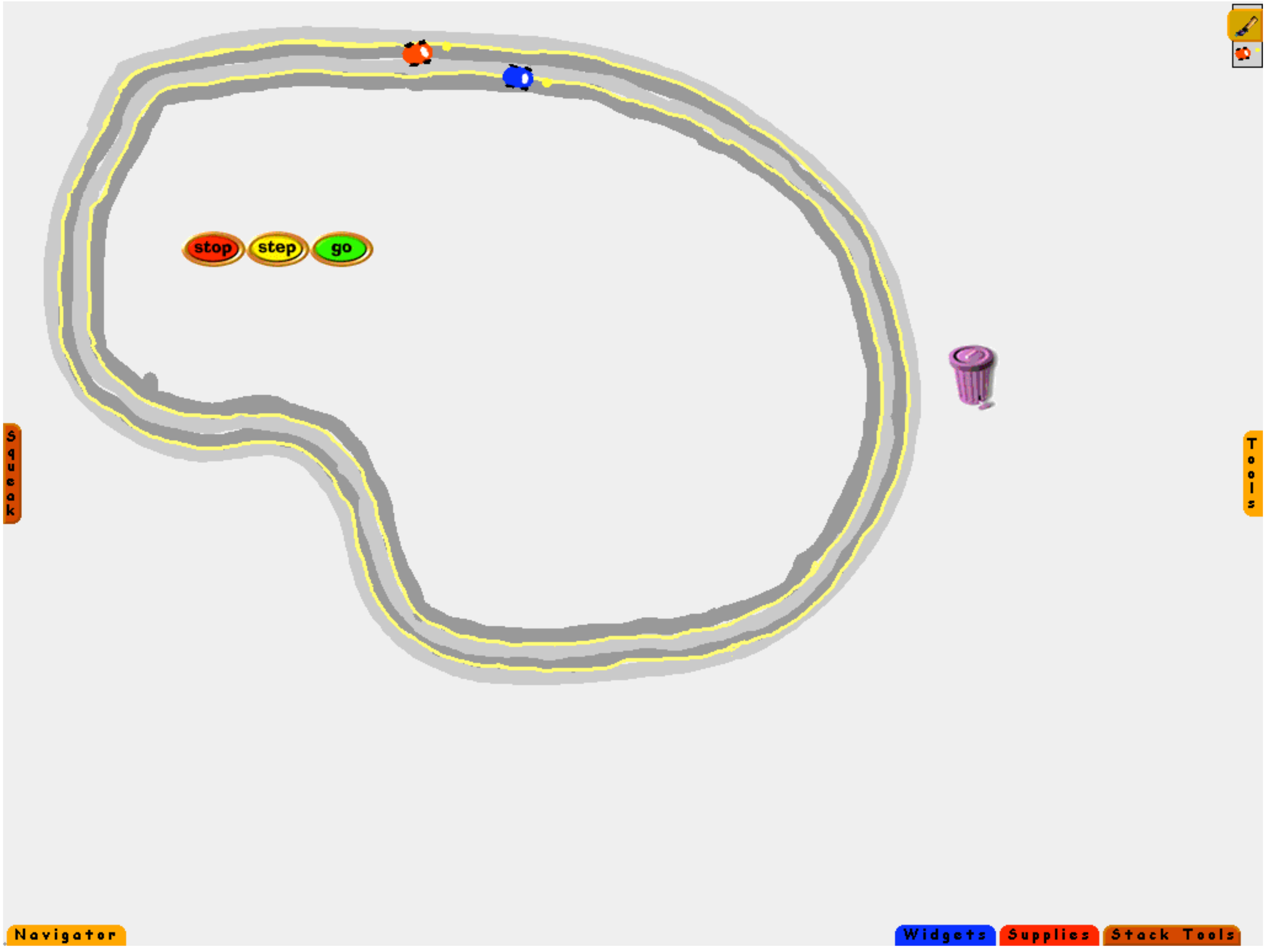


* Example 1: Drive a Car

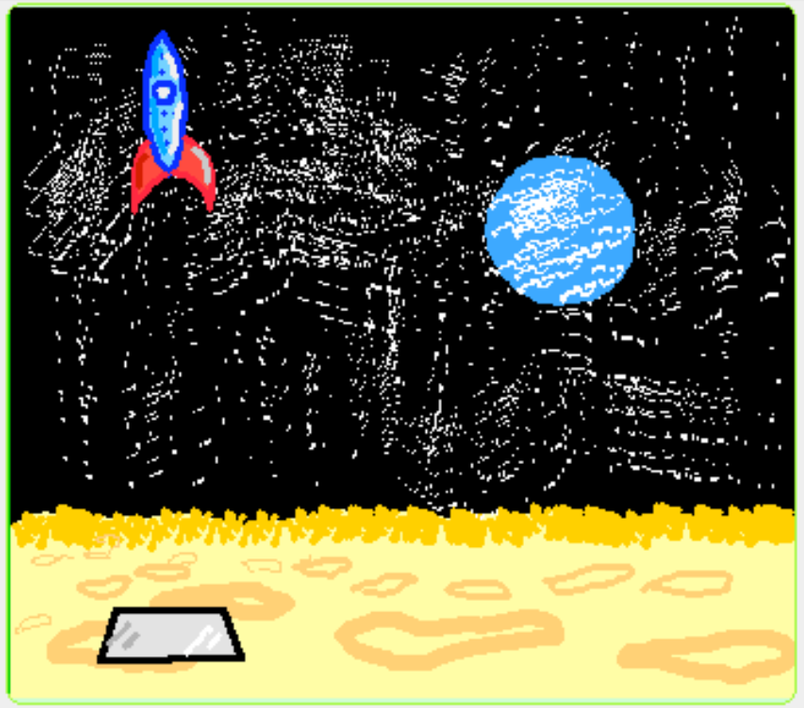


* Example 2: Lunar Lander

* Example 4: Underwater World


stop
step
go
Squeak
Tools
Navigator
Widgets
Supplies
Stack Tools

Your Own Lunar Lander Game



stop step go

ship reset

ship's ySpeed = 0

Squeak

Tools

Navigator

Widgets Supplies Stack Tools

Part II: Overview

* Squeak: The Language

  - Smalltalk in 2 minutes
  - Demo: Browser
  - Demo: MehtodFinder

Smalltalk in 2 minutes

* reserved words: (pseudo-variables):

  self super thisContext true false nil

* Literals: 1 1.1 'String' #('ein' 'array') #symbol

* Blocks: [:param | code] [1]

* Assignment: a := 1

* Methods:

```
| lokale Variablen |
^ Ergebnis
```

* Methode call: 3 raisedTo: 4

Beispiele

```
(1 < 2) ifTrue: ['wahr'] ifFalse: ['falsch'].

#( 1 2 3 4) + 3

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

#(1 2 3 4) collect: [:each | each class].

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

#(1 2 3 4 5 6) inject: 0
                    into: [:sum : each | sum + each].
```