Squeak als agile Entwicklungsumgebung

Marcus Denker  IPD Universität Karlsruhe (TH)
Über mich

Name: Marcus Denker
Studiert Informatik in Karlsruhe

Squeaker seit 1998

Projekte:
* Just-In-Time Compiler
* Squeak Deutschland e.V.
Teil I: Das Squeak Projekt

Inhalt:

* Was ist Squeak?
  -> Beispiele

* Historisches
  -> Alan Kay’s Dynabook

* Squeak für Kinder
  -> Beispiele
Was ist Squeak?

1. Multimedia Autorensystem
2. Kinderprogrammiersystem
3. Betriebssystem?
4. Programmiersprache
5. Entwicklungsumgebung
6. Eine Community
Multimedia

* Text und Bilder
* Präsentationen
* 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 means 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?

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 content. Pianists know that music is not in the piano. It begins inside human beings as a special urge 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 schoolchildren, "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 act 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 to learn more about the world."
Übersicht Vortrag

Anfang

Inhalt

Über mich

Inhalt Teil I

Was ist Squeak?

Multimedia

Sam'sFaceBall

Programmiersprache

portable

Klasseneingruppierung

community

Historisches

Dynabook

Squeak für Kinder

Auto

The Big Race

New Lunar Lander

Teil II
Video und Audio
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 (shift-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."

```
```

"The bunny moved forward 2 meters. When you are finished, close the Alice window and return to the Squeak environment."

"The Alice window should still be running. Close it. Good job!"

"Now, let's try the camera. Place the bunny in the scene, and put the cursor on the bunny in the Alice window."

"The bunny is now at the bottom left of the screen. Now put the cursor in the line of code below and hit command-D (shift-D for PC users):"

```
camera move: right.
```

"The camera moved to the right. You can also move the camera up and down.

```
camera move: up.
```

"The camera moved up. Now put the cursor in the line of code below and hit command-D (shift-D for PC users):"

```
camera move: down.
```

"The camera moved down. You can also move the camera left and right.

```
camera move: left.
```

"The camera moved to the left. Close the Alice window. Good work!"

Have fun with Squeak-Alice.

Welcome to Squeak-Alice!

[Alice window]

[Scene]

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

[Tools]

- Undo
- Reset
- Quit

[Navigator]

- Squeak
- scene
ground
camera
bunny
drum
head
glasses
rightEar
leftEar
body
leftLeg
leftArm
mallet
rightArm
mallet
rightLeg

[Widgets]

- Supplies
Was ist Squeak?

1. Multimedia Autorensystem
2. Kinderprogrammiersystem
3. Betriebssystem?
4. Programmiersprache
5. Entwicklungsumgebung
6. Eine Community
Kinderprogrammierung

Sam L.
Was ist Squeak?

1. Multimedia Autorensystem
2. Kinderprogrammiersystem
3. Betriebssystem?
4. Programmiersprache
5. Entwicklungsumgebung
6. Eine Community
Squeak: Die Programmiersprache

* Vollständig objektorientiert: Alles ist ein Objekt
* Virtuelle Maschine, Bytecode
* Garbage Collector
* Grosse Klassenbibliothek
* Beispiel:

100 factorial
Klassenbibliothek

* 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
Portabel

- Windows 2000
- Windows NT
- Windows 95
- Windows 98
- Windows CE
- DOS
- Macintosh
- OS/2
- Acorn
- BeOS
- Linux/4386
- Linux/PowerPC
- Linux/Sparc
- SunOS
- Solaris
- SCO System V
- Rhapsody/Next Step
- DigitalUnix/Alpha
- NetBSD/Sparc
- NetBSD/4386
- Psion 5
- Zaurus

- Embedded Squeak
- Netscape Plugin
- IE-Plugin (Squeak as ActiveX Control)
- Multilanguage Squeak
- Bootable Squeak
- Goodies
Autonomous Controller for Microseeker built with Squeak

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

HUUV 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 FPA device called a Helio, and it will be written in Squeak. There is also a simulator written in Squeak.

Squeak for Pocket PostPet
(Popope Squeak)
EPOC 32

More informations

The port is in a very early stage (no keyboard no pointer, very slow BitBlit).

Contact: Pablo J. E. Rodriguez (pablo@rodablo.com)
http://www.rodablo.com/squeak/
http://minnow.cc.gatech.edu/squeak/1805

Welcome to Mini Squeak 2.2 for EPOC32

Much has been removed from the normal Squeak release.
We welcome your assistance in pointing out features that should not have been removed, or in cleaning up the code so that missing features do not cause errors.

Such suggestions will be happily received by Dan Ingalls, dan@wdi.disney.com, on behalf of the Squeak team.

Hope you enjoy this mini version of Squeak, the Smalltalk written in itself.
9/29/99
Squeak running on a Hyperstone processor with SCT Expansion Board.
Squeak on VTech Helio

There is also a Squeak port to VTech helio devices.

- Squeak Wiki: [http://minnov.cc.gatech.edu/squeak/1706](http://minnov.cc.gatech.edu/squeak/1706)
- Helio Start Page: [http://pocket1.cjb.net](http://pocket1.cjb.net)
Squeak is running on multiple platforms including several Unix Systems. This picture shows Squeak running on Suse Linux with KDE.
Squeak is running on multiple platforms including all Windows platforms (Win98, Win95, WinNT, WindowsCE, Windows 2000, ...).
Was ist Squeak?

1. Multimedia Autorensystem
2. Kinderprogrammiersystem
3. Betriebssystem?
4. Programmiersprache
5. Entwicklungsumgebung
6. Eine Community
Entwicklungsumgebung

* Der KlassenBrowser:

* Inspektor:
Was ist Squeak?

1. Multimedia Autorensystem
2. Kinderprogrammiersystem
3. Betriebssystem?
4. Programmiersprache
5. Entwicklungsumgebung
6. Eine Community
Die Squeak Community

* Entwickler:
  - Mailingliste (ca. 1000 Mitglieder)
  - SqueakNic

* Lehrer und Interessiert:
  - Squeakland.org und Mailingliste

* Deutschland:
  - Squeak Verein
Historisches

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

* Xerox PARC 1970 - 1980
  - GUI
  - Smalltalk

* Squeak seit 1996
  (Apple, Disney, jetzt HP)
Das Dynabook

Alan Kay: "Ideaprocessor Vs. Wordprocessor"
Squeak für Kinder

* Einfache graphische Skriptsprache: eToys

* Beispiel 1: Auto fahren

* Beispiel 2: Lunar Lander

* Beispiel 3: Unterwasserwelt
Auto fahren
Your Own Lunar Lander Game

- Ship gravity
- ship's ySpeed increase by
- ship's y increase by ship's ySpeed
- Ship motor
- ship's ySpeed increase by Joystick2's pDown
- Flame on
- Joystick2's pDown
- x + ship's x
- Flame's y + ship's y
- Flame show
- Flame hide
- Ship land
- ship's color
- sees color
- ship's ySpeed
- Flame hide
- Ship hide
- ship make sound
- Splash crash show
- Ship all Processes
- Pause
- Stop
- Stop
- Go
- Ship reset
- Ship's ySpeed = 0
- Ship's x = 70.0
- Ship's y = 260
- Ship's ySpeed = 0
- Ship show
- Flame hide
- Crash hide
- Ship reveal
- Flame show
- Flame show
20 mSecs (47 frames/sec)

The two dots on top represent variables used in the fish animation scripts.

Here is the viewer for the nudi branch (note: the updating thumbnails are expensive so make sure you don’t have too many).

These buttons control the update cycle and should always be ‘on’.

To see the fish animations you have to open a viewer on the 'camera' to the left and look at one of the animateFishXYZ scripts.

The joystick controls the nudi branch’s motion using this script.

The flap on the right contains the fish textures.

The script to our left controls the wave animation: the holder containing the waves can be seen in the flap to the right.

The Wonderland scripting area is hidden right here.

This is the shark animation.

NAVIGATION: Click and drag in the 3D world or use cursor keys.

Here are the controls for all eToy scripts:

This script controls the animation cycle of the nudi branch itself [note: the NudiBranch model has animations for head and body separately].

The two dots on top represent variables used in the fish animation scripts.

Here is the viewer for the nudi branch (note: the updating thumbnails are expensive so make sure you don’t have too many).

These buttons control the update cycle and should always be ‘on’.

To see the fish animations you have to open a viewer on the 'camera' to the left and look at one of the animateFishXYZ scripts.

The joystick controls the nudi branch’s motion using this script.

The flap on the right contains the fish textures.

The script to our left controls the wave animation: the holder containing the waves can be seen in the flap to the right.

The Wonderland scripting area is hidden right here.

This is the shark animation.

NAVIGATION: Click and drag in the 3D world or use cursor keys.

Here are the controls for all eToy scripts:

This script controls the animation cycle of the nudi branch itself [note: the NudiBranch model has animations for head and body separately].

The two dots on top represent variables used in the fish animation scripts.

Here is the viewer for the nudi branch (note: the updating thumbnails are expensive so make sure you don’t have too many).

These buttons control the update cycle and should always be ‘on’.

To see the fish animations you have to open a viewer on the 'camera' to the left and look at one of the animateFishXYZ scripts.

The joystick controls the nudi branch’s motion using this script.

The flap on the right contains the fish textures.

The script to our left controls the wave animation: the holder containing the waves can be seen in the flap to the right.

The Wonderland scripting area is hidden right here.

This is the shark animation.

NAVIGATION: Click and drag in the 3D world or use cursor keys.

Here are the controls for all eToy scripts:

This script controls the animation cycle of the nudi branch itself [note: the NudiBranch model has animations for head and body separately].
Teil II: Inhalt

* Grundlagen
  - Smalltalk in 2 Minuten
  - Demo: Browser

* Seaside
  - Überblick
  - Demo: Counter
* Reservierte Wörter: Pseudovariablen
  self super thisContext true false nil

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

* Blöcke: [:param | code] [1]

* Zuweisung: a := 1

* Für Methoden: | lokale Variablen |
  ^ Ergebnis

* Methodenaufrufe: 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].
+ aNumber
  "Refer to the comment in Number +"
  aNumber isInteger ifTrue:
    [self negative == aNumber negative
     ifTrue: [^ (self digitAdd: aNumber) normalize]
     ifFalse: [^ self digitSubtract: aNumber]].
  ^aNumber adaptToInteger: self andSend: #+
Seaside: Überblick

* Grundlegende Informationen
* Eigenschaften von Seaside
* Demo: Counter
* Sessions und Components
* Interaktionen zwischen Komponenten: 
  #call und #answer
Seaside: Squeak Enterprise Aubergines Server

* Entwickler: Avi Bryant, Julien Fitzell
  Beta4 Productions (beta4.com)

* Buchungssystem Whistler.com

* Ursprünglich für Ruby: "IOWA"

* Link: http://www.beta4.com/seaside2

* Version 2 "Borges"
  - komplette Neuimplementierung.
  - noch nicht ganz vollständig
Seaside: Eigenschaften

* Session Management:
  - User Session ist ein Block zugeordnet
  - Controll-Flow wie in GUI-Applikation

* Callback basiertes Eventmodell

* HTML Generierung mittels Smalltalk-Klasse oder Template System

* Wiederverwendbare UI-Komponenten

* Optional: Entwicklertools im Web-Browser

* Verwendet Squeak Webserver "Comanche"
Counter Demo

* Seaside starten:
  WAKom startOn: 9090

* http://localhost:9090/seaside/counter

3

++ =
Seaside: Sessions

* In URL codiert
  
  http://localhost:9090/seaside/counter/2;730aba8c-286d-11d7-91ae-003065b55c16

* Jeder Session ist Block zugeordnet:

* alle Daten werden von Seaside in einem Cache vorgehalten (LRU)
Seaside: Components

* Zustand:  ("count")

* Aktion:  #increment, #decrement

* Display:  #renderContentsOn:

* Response Loop:
  - WACounter erzeugen. Initialisierung.
  - #renderContentsOn:
  - warten auf Eingabe
  - User Eingabe: Callback ruft #increment auf
  - #renderContentsOn:

* Komponenten können geschachtelt werden
Seaside: `#call` und `#answer`

* Neue Komponente aufrufen mit `#call`:

* Resultat zurück: `#answer`:

* Beispiel 1: `#call`:

```ruby
decrement
count = 0
ifFalse: [count := count - 1]
ifTrue: [self call: (WADialog new message: 'Ups!')]`

* Beispiel 2: `#inform`:

```ruby
decrement
count = 0
ifFalse: [count := count - 1]
ifTrue: [self inform: 'Ups!']```
Teil III: Überblick

* Neu: Version 3.4
* Die Zukunft: OpenCroquet
* Der Squeak Verein
* Statistik
* Fragen?
Version 3.4

* 3.4
  - Modularisierung
  - Squeak Kern + SqueakMap Archive

* Weiter:
  - Squeak Just-In-Time VM
  - VI4: neue Version 4
Croquet: Squeak in 3D

* Peer2Peer Virtual Reality System
* leider: nur Bild
Der Verein

* Gründung LinuxTag 2002
* Noch in Gründungsphase:
  - als gemeinnützig anerkannt
  - Eintragung bald

* Mitglieder gesucht!
  - Siehe http://squeak.de
Die Squeak CD-ROM

* Squeak für alle Systeme
  - Version 3.2 + 3.4b

* OpenCroquet 0.1 preAlpha

* Dokumentation und Tutorials

* http://squeak.de
  - CD-Image kostenlos
  - gebrannte CD ab 10 EUR
Statistik

* Anzahl Klassen: Smalltalk allClasses size

* Anzahl Methoden:
  CompiledMethod allInstances size

* Anzahl Objekte:
  Smalltalk allClasses inject: 0 into:
      [:sum :each | each allInstances size + sum] 672255

* Speicher: vm statistics

* LinesOfCode:
  Smalltalk allClasses inject: 0 into: [:sum :each | each linesOfCode + each
class linesOfCode + sum] 370240
Das wars!

FRAGEN?