

Squeak

Markus Gaelli und Marcus Denker

Geschichte des Projektes

Ende 1995 beschlossen mehrere Smalltalk-Veteranen, die damals bei Apple unter Vertrag standen, eine Entwicklungsumgebung zur Erstellung von Lernsoftware zu erschaffen. Diese sollte möglichst plattformunabhängig, internetfähig und multimedial sein, dabei aber dennoch auch von Nicht-Technikern und sogar von Kindern programmiert werden können. Es waren zu einem großen Teil die gleichen Leute, die in den 70er Jahren in der Learning-Group am Paolo-Alto-Research-Center von Xerox für Furore sorgten.

So befand sich unter ihnen Alan Kay, der damals maßgeblich das objektorientierte Paradigma und die heute gängigen graphischen Benutzeroberflächen geprägt hat, zusammen mit Dan Ingalls, dem Chefentwickler von Smalltalk 80. Als Basissystem erwogen sie, Java oder bereits bestehende Smalltalk-Dialekte zu verwenden. Java erschien ihnen damals aber noch zu instabil und auch zu unflexibel, die bestehenden Smalltalk-Dialekte erlaubten es nicht, ihre Vorstellungen von multimedialen, über das Internet zu verteilende Applikationen umzusetzen. Squeak entstand und wird heute als OpenSource/Internet-Projekt weiterentwickelt. Editiert werden die verschiedenen Squeak-Versionen von SqueakCentral, den Männern, die unterdessen von Apple zu Walt Disney Imagineering gewechselt sind.

Das Squeak Projekt möchte eine Umgebung schaffen, der es Kindern jeden Alters ermöglichen soll, mittels eines "Medienbaukasten" kreativ mit dem Computer umzugehen. Im Zusammenhang mit dem Einsatz von Computern im Unterricht wird häufig von "neuen Medien" gesprochen. Im Angelsächsischen Sprachraum wird der Begriff "Computer Literacy" verwendet. Hier werden eindeutig Parallelen zur "Literacy", dem Lesen und Schreiben als Kulturtechnik gezogen.

Der Computer als Medium

Wenn der Computer nun ein neues Medium sein soll, so muss auch eine echte "Literacy", ein Lesen und Schreiben, für dieses Medium gelehrt werden. Die "neuen Medien" haben hier aber ein großes Problem: Alle aktuellen Multimediaformate (etwa PDF, Flash, usw. ...) erlebt der Benutzer als reine Präsentationsformate. Der Computeranwender ist einzig Konsument. Die Programme zum Konsumieren werden kostenlos verteilt, das Erstellen ist aber nur Profis mittels teurer und komplizierter Programme möglich. Dies ist vor allem auch für die "neuen Unterrichtsmidien" eine sehr negative Entwicklung: Die "neuen Medien" sind rein passiv, vollständig auf Konsum ausgelegt.

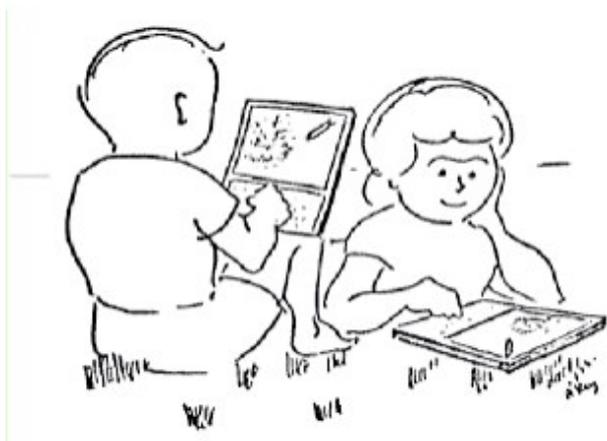
Eine "Literacy" verlangt ein System, in dem jedes Artefakt nicht nur passiv konsumiert, sondern auch aktiv kommentiert werden kann. Squeak versucht eine solches Medium zu sein: Es erlaubt dem "Leser" jederzeit, die aktive Rolle zu übernehmen und zum Autor zu werden. Das Ziel ist ein System, das keine Grenzen auferlegt: Schon Grundschüler können neue Ideen spielerisch erkunden, aber auch normale Anwender und Experten können ihre Ideen in der Squeak-Umgebung verwirklichen.

Verschiedene Umgebungen vorhanden

Natürlich benötigen die verschiedenen Benutzer unterschiedliche Umgebungen, die entsprechend den Fähigkeiten unterschiedliche Hilfen und Hinweise bieten. Squeak bietet Umgebungen für Kinder, Anwender und Experten. Man kann hier eine Parallele zum Erlernen der Muttersprache ziehen: Ein lernendes Kind spricht die gleiche Sprache wie die Erwachsenen, natürlich ist die Sprache des Kindes strukturell weniger komplex und auch weniger vielfältig. Aber es ist die gleiche Sprache, das Kind muss niemals etwas gelerntes wieder vergessen, wenn es die Sprache der Erwachsenen erlernt. Dieser Artikel führt hauptsächlich in die Squeak "E-Toy" Umgebung ein, die schon in der Grundschule sinnvoll eingesetzt werden kann. Am Ende des Kapitels finden sich Hinweise auf die weiteren Einsatzmöglichkeiten.

Die Zukunft erfinden

"The best way to predict the future is to invent it." (Alan Kay)

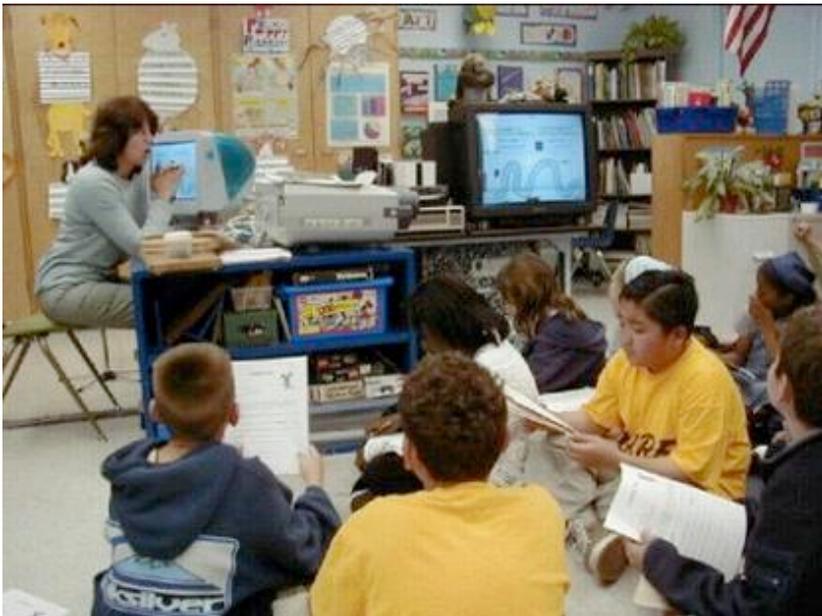


Seit Ende der 60er Jahre hat Alan Kay den Traum, einen speziellen tragbaren Computer zu bauen, das sogenannte "Dynabook". Es soll Kindern aller Alterstufen erlauben, besser zu lernen, indem sie mit ihm neue Ideen entwickeln und auszuprobieren können, seien sie künstlerischer oder wissenschaftlicher Natur.

1972 kam er zur "Learning Research Group" am Xerox Palo Alto Research Center (Xerox PARC). Dort wurden unter seiner Mitwirkung die bahnbrechenden Ideen der grafische Benutzeroberfläche, der objektorientierten Programmierung und des Personal-Computers entwickelt. 1996 startete die Apple-Forschungsabteilung "Learning Concepts Group" unter seiner Leitung das Projekt "Squeak", um damit die Software für das Dynabook weiter zu verwirklichen. Nach einigen Jahren bei Walt Disney hat Alan Kay im Jahr 2001 die non-profit Organisation "Viewpoints Research Institute" gegründet. [<http://www.viewpointsresearch.org>].

Einsatz in der Open School

Squeak wird seit zirka zwei Jahren sehr erfolgreich in der Open School (Los Angeles, USA) eingesetzt. Die Lehrerin BJ Alan Conn benutzt Squeak in der vierten und fünften Klasse, mit Hilfe von Squeak sollen die Schüler konkrete Erfahrungen mit den im Mathematikunterricht gelernten Konzepten sammeln.



In der Open School in Los Angeles

Schon nach einer kurzen Einführung sind die Kinder in der Lage, einfachere Dinge mit Squeak zu verwirklichen, sie können Objekte am Computer zeichnen, sie bewegen und die Aktionen mit Soundeffekten unterlegen.

Projekte mit Squeak

Ein Beispiel für ein großes Projekt, das die Schüler später in Angriff nehmen, ist ein Autorennen: Es werden Autos konstruiert, die selbständig über eine Rennbahn fahren können. Am Ende treten die verschiedenen Autos gegeneinander an. Wer wird gewinnen?



Squeak ist nicht nur zum Malen gut ...

Autorennen:

Basteln wir uns selbst ein kleines Autorennen und schauen wir zu. Wir werden uns dazu ein Auto und eine Rennstrecke bauen, das Auto erst selbst mit einem Lenkrad steuern und es anschließend mit Hilfe von Sensoren automatisch über die Piste manövrieren lassen. Konstruktion des Autos und der Rennstrecke:

Wir kreieren unsere kleine Rennwelt, indem wir ihre Elemente einfach malen. Dazu öffnen wir wie die Ablage (Abbildung 1) und ziehen aus ihr die Malpalette auf die Oberfläche (Abbildung 2).



Abbildung 1: Öffnen der Palette

Wir malen unser Auto und übernehmen es mit "keep".



Abbildung 2: Die Malpalette

Wir können das Auto mit einem Links-Klick anheben und beliebig verschieben, ein weiterer Links-Klick lässt es wieder fallen. Das Auto wirft nach dem Anheben einen Schatten, es ist nicht etwa ein rechteckiges Bild, sondern ein freistehendes Symbol. Alt-Klick auf das Auto lässt die sogenannten "Smarties" (Abbildung 3) des Autos erscheinen, außerdem erscheint der Name des Objekts. Zunächst ist das ein Defaultname, den wir nach Linksklick auf den Text in "Auto" ändern.



Abbildung 3: Unser Auto mit "Smarties" und Rotationszenter

Analog bauen wir uns eine Rennstrecke und setzen das Auto auf sie (Abbildung 4).



Abbildung 4: Unser Auto wird startbereit gemacht

Direkte Steuerung des Autos:

Wir haben unser Auto fertig gebaut, jetzt wird es Zeit, die ersten Fahrtests zu machen. Dazu benutzen wir die graphische Skriptsprache von Squeak namens "E-Toys". Über den "Viewer-Smartie" (das Auge in Abbildung 3) öffnen wir eine Befehlsleiste, welche die mögliche Befehle für dieses Auto enthält (Abbildung 5).



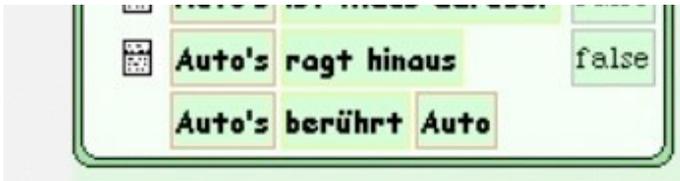


Abbildung 5: Die Befehlsleiste des Autos

Hier sehen wir unter anderem "Auto gehe vorwärts um 5" und "Auto drehe dich um 5". Durch "Drag and Drop" lassen sich diese Befehle auf die Oberfläche ziehen. Es entstehen zwei neue Skripte. (Abbildung 6, Abbildung 7)

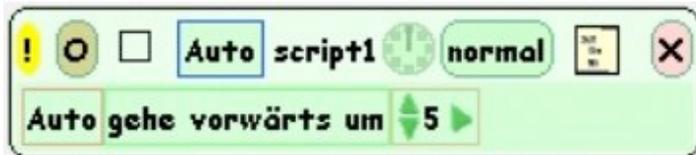


Abbildung 6: Bewege das Objekt "Auto" um fünf Pixel nach vorne

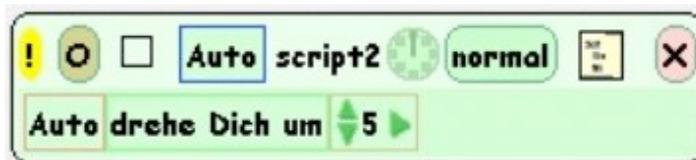


Abbildung 7: Drehe das Objekt "Auto" um fünf Grad in den Uhrzeigersinn

Durch Linksklick auf das Ausrufezeichen links oben im Skript lassen sich die Skripte ausprobieren: das Auto bewegt sich um fünf Pixel nach vorne, bzw. dreht sich um fünf Grad in den Uhrzeigersinn. Falls es sich seitlich "vorwärts" bewegt, war seine Richtung falsch eingestellt: Jedes Objekt hat eine interne Richtung, anfangs zwölf Uhr. Das Rotationszentrum eines neuen Objektes befindet sich in der Mitte. Das Zentrum kann aber visuell verändert werden, indem man im Menü des Autos (Abbildung 3) "Direction Handles" selektiert und anschließend mittels Alt-Linksklick erneut die Smarties des Objektes anzeigt. Der grüne Pfeil kann nun in die Vorzugsrichtung des Objektes gedreht werden. (Abbildung 3)

Wann die Skripte ausgeführt werden sollen, lässt sich über ein Popup-Menü einstellen, welches man über einen Linksklick auf "normal" (Abbildung 8) rechts oben in der Skriptleiste erhält. Stellen wir dies auf "ticking", wird das Skript jede Sekunde angestoßen, setzen wir aber zunächst beide Skripte auf "paused".

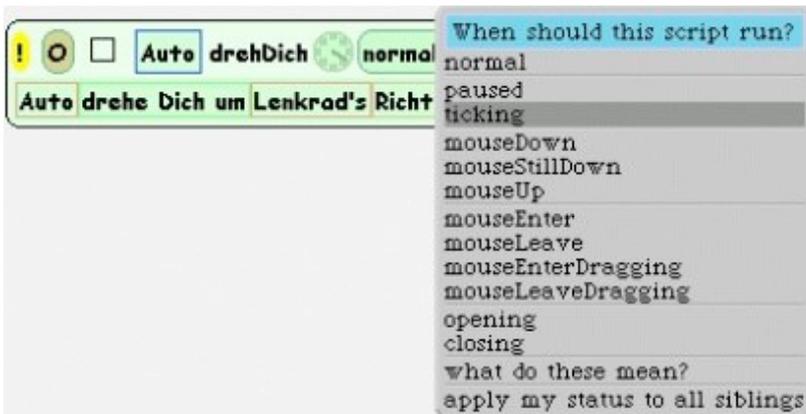


Abbildung 8: Wann soll das Skript ausgeführt werden

Um alle Skripte innerhalb dieses Projektes auf einmal zu starten, zu beenden oder schrittweise auszuführen, gibt es ein eigenes Werkzeug "All Scripts" (Abbildung 9), welches sich in der Ablage "Widgets" befindet. Starten wir alle Skripte über dieses Werkzeug, können wir beobachten, dass das Auto im Kreis fährt. Mit Pause stoppen wir das Auto.



Abbildung 9: Starte oder stoppe alle Skripte dieses Projekts

Steuerung des Autos mit einem Lenkrad

Malen wir zunächst ein Lenkrad mit dem Malwerkzeug. Das Auto mit dem Lenkrad zu steuern, bedeutet, es um soviel Grad zu drehen, wie das Lenkrad ausgeschlagen ist. (Abbildung 10)



Abbildung 10: Wir drehen das Lenkrad mit seinem Rotations-"Smartie"

Auch auf das Lenkrad öffnen wir eine Befehlsleiste und ziehen mittels "Drag and Drop" die Richtung des Lenkrads auf die Gradangabe (momentan noch 5) des Skriptes, mit welchem wir das Auto rotieren lassen (Abbildungen 11 und 12).



Abbildung 11: Auch Variablen können aus den Vorlagen kopiert werden

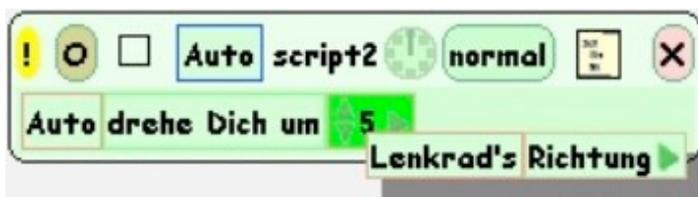


Abbildung 12: Drehe dich um die Auslenkung des Lenkrades

Starten wir wieder die Skripte und versuchen das Auto auf der Strecke zu halten, indem wir das Lenkrad drehen. Schwierig, nicht? Was könnten wir tun, um die Steuerung zu verbessern? Vielleicht sollten wir eine Übersetzung einbauen und nur einen Bruchteil der Rotation des Lenkrads zur Steuerung unseres Autos nutzen: Dazu klicken wir auf das Dreieck rechts von "Lenkrads Richtung". Die Kachel erweitert sich um "+1". Wenn wir jetzt auf das "+" klicken, wählen wir aus dem aufgehenden Menü "/" aus, die "1" erhöhen wir mit den Pfeilen auf "5" (Abbildung 13).



Abbildung 13: Das Auto soll sich nur um einen Bruchteil des Lenkradausschlags drehen. Das Auto lässt sich nun wesentlich leichter manövrieren.

Automatische Steuerung des Autos

Stoppen wir die Skripte zunächst wieder mit der Skriptsteuerung aus Abbildung 9 und drehen das Lenkrad auf seine Grundposition von 0 Grad. Wie halten wir das Auto automatisch auf der Strecke? Eine Idee wäre, dem Auto Sensoren zu geben und es nach rechts gegensteuern zu lassen, wenn sein linkes Auge eine Farbe ungleich der Streckenfarbe erkennt, analog nach links, wenn sich sein rechtes Auge nicht mehr auf der Strecke befindet.

Öffnen wir dafür zunächst wieder das Malwerkzeug des Autos mit Hilfe des "Neumalen-Smarties" (Abbildung 3) und malen dem Auto links einen grünen und rechts einen blauen Sensor ein. (Abbildung 14)

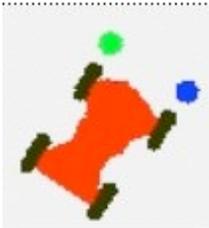


Abbildung 14: Unser Auto bekommt Sensoren...

Anschließend ziehen wir den Rotationsbefehl des Autos aus seiner Befehlsleiste auf die Oberfläche und erzeugen so ein neues Skript. In diesem Skript wollen wir programmieren, dass sich das Auto nach rechts, also in den Uhrzeigersinn drehen soll, wenn sein linkes grünes Auge eine Farbe ungleich der Pistenfarbe sieht.

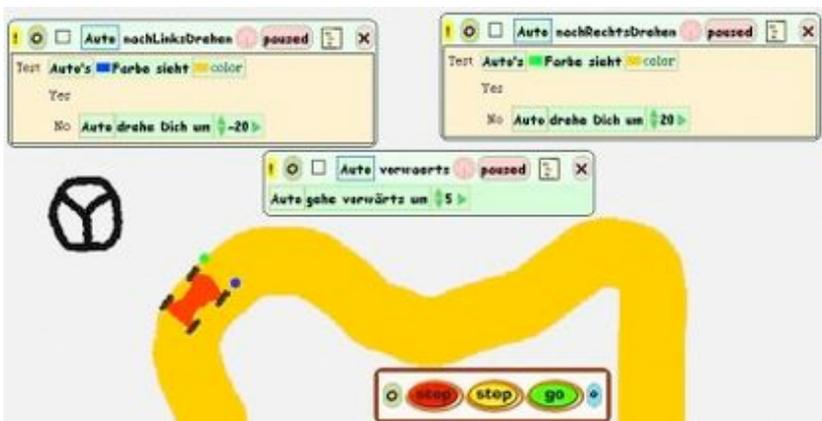


Abbildung 15: ...und fährt automatisch die Piste entlang

Dazu ziehen wir gelbe "Test-Wahr-Falsch-Kachel" ins Skript, die sich im Skript-Fenster rechts oben neben dem "Fenster-Schließen-Kreuz" befindet (Abbildung 15). Die Testbedingung ist, dass die grüne Farbe des Autos die Farbe orange der Rennstrecke sieht, wir fügen sie ein, indem wir in der Befehlsleiste des Autos die Kategorie "Tests" auswählen und dort die Kachel "Autos Farbe sieht" in die Test-Zeile unseres Skripts bewegen. Die Farben stellen wir ein, indem wir erst auf die farbig ausgefüllten Rechtecke der eben erzeugten Kachel "Autos Farbe sieht" klicken und mit der erscheinenden Pipette die jeweiligen Farben vom Auto und der Rennstrecke abgreifen. Jetzt müssen wir nur noch den Rotationsbefehl in unserem Skript in die "No-Zeile" verschieben, schließlich soll sich unser Auto nur dann drehen, wenn es nicht auf der Piste steht. Analog erstellen wir das Skript für die Rotation des Autos nach links (Abbildung 15). Wenn wir unsere Skripte starten, sollte das Auto selbstständig über die Strecke kurven, kopieren wir das Auto mit dem "Kopieren-Smartie", können wir die Modelle unterschiedlich programmieren und gegeneinander antreten lassen.

No Limits

Die Squeak Umgebung erlaubt natürlich nicht nur Autorennen. Ältere Kinder können mit Squeak komplexe Projekte verwirklichen. So kann auch ein anderes Computerspiel gebaut werden. Beispielsweise "Lunar Lander", realisiert in Squeak E-Toys. Das Ziel dieses klassischen Spiels ist das Landen einer Mondlandefähre. Auf den ersten Blick ist das nur ein einfaches Videospiel, aber wenn man es selber erstellen möchte, braucht man viel physikalisches Wissen: Geschwindigkeit, Beschleunigung, Fallgesetze, usw.



Lunar Lander

Das Vivarium Projekt ist eine biologische Simulation: In einer dreidimensionalen Unterwasserwelt leben einige Tiere. Ziel ist es, diesen Tieren arttypisches Verhalten beizubringen. Schon mit sehr einfachen Regeln lassen sich typische Verhaltensweisen modellieren.



Vivarium Projekt

Vielfältige Praxisanwendungen möglich

Squeak bietet viel für den "normalen" Anwender und den Experten. Der Anwender möchte zum Beispiel eine Präsentation für einen Vortrag erstellen. Oder eine Multimedia-Präsentation mit Text, Bildern, Video und Musik.

Auch für den Experten – bis hin zum Informatiker an der Universität -- ist Squeak ein interessantes Medium: Squeak ist komplett "in Squeak" implementiert, alles ist für Experimente und Erweiterungen zugänglich. Es eignet sich hervorragend für die Erstellung von Prototypen und zum Experimentieren. Für den Profi ist Squeak kein Spielzeug, sondern bietet eine mächtige Programmierumgebung.

Individuelle Zusammensetzung des Programms

Der Squeak Baukasten bietet eine Menge an Funktionen, folgende Tabelle gibt einen kleinen Ausschnitt:

- 2D-Graphik
- TrueType, Flash GIF, JPEG, PCX, XBM
- Video: MPEG und MJPEG
- 3D-Graphik
- Hardwarebeschleunigtes 3D-Graphik Subsystem
- 2D ist integriert ("Aktive Texturen")
- VRML import
- Sound:
- Aufnahme, Wiedergabe
- Komprimierung: ADPCM, AIFF, GSM, MuLaw, MP3 decoding
- FM-Synthese
- MIDI
- Einfache Sprachausgabe (Text-To-Speech)
- Netzwerk:
- Http, FTP, POP, SMTP
- Mailreader, Webserver
- Groupware-features: Chat (Voice, Text), Applikation-sharing
- Misc:
- Digitale Signaturen
- Kompression: ZIP, gzip
- Postscript export
- Portabel:
- Linux (alle Plattformen (i386, alpha, mips, ppc))
- Windows (95, 98, NT, 2000, ME, CE)
- MacOS 8,9,X
- Unix (Solaris, SGI, FreeBSD, NetBSD, OpenBSD, etc...)
- Acorn RiscOS, BeOS, NeXT, OS-2

Und diese Funktionalität ist, wie schon gesagt, frei zugänglich: Fast alles ist in Squeak implementiert und natürlich von Squeak aus nutzbar. Die "Profi"-Version der Squeak Sprache kann man vielleicht am besten mit der Sprache Python vergleichen, die schon sehr erfolgreich im Unterricht der Oberstufe eingesetzt wird. Mit Squeak wurde das kommerzielle Squeak eZine – eine Informationsschrift – professionell realisiert: Die SqueakNews erscheinen monatlich auf CDROM [SqueakNews: <http://squeaknews.com>]. Der Vorteil bei der Verwendung von Squeak liegt unter anderem auch darin, dass diese Multimedia-CD-ROM auf allen Systemen lauffähig ist, nicht nur unter Windows.

Squeak als Open Source

Squeak ist kein Produkt. Man kann es nicht im Geschäft kaufen. Squeak ist frei im Internet erhältlich und wird von einer großen Gemeinschaft von Programmierern und Anwendern weiter entwickelt. In den letzten Jahren hat Squeak eine beeindruckende Funktionalität erreicht, aber fertig ist es noch lange nicht. Die Squeak Umgebung ist noch nicht vollständig ins Deutsche übersetzt, eine deutsche Dokumentation ist noch nicht vorhanden. Wenn dieser Artikel Interesse geweckt haben sollte: Auch Sie können bei der Entwicklung helfen! Nicht nur Programmierer, vor allem Anwender sind wichtig, um aus Squeak ein Medium zu machen, das es wert ist, von Kindern benutzt zu werden. Die deutschsprachige squeakland-de Mailingliste [<http://lists.squeakfoundation.org/listinfo/squeakland-de>] hat die Anwendung von Squeak in der Schule zum Thema. Weiterführende Informationen, Links, Literatur, usw. findet sich unter der nachfolgenden Website: [<http://swiki.squeakfoundation.org/squeak-ev/24>].

Einige Artikel über die Hintergründe von Squeak finden sich auf Squeakland.org, hier kann man auch das Squeak Browser-Plugin und Beispielprojekte herunterladen. Technisch Interessierte finden sehr viele Informationen und die aktuellste Entwickler-Version auf squeak.org. Ein Förderverein für Squeak ist gegründet. Der Verein hat das Ziel, die Anwender von Squeak zu unterstützen sowie die Weiterentwicklung von Squeak zu fördern [<http://www.squeak.de/>].