

Universidad Católica Nuestra Señora de la Asunción

Teoría y aplicaciones de la informática II

Squeak. Un Smalltalk del siglo XXI



Humberto Aquino

2006

Tabla de Contenidos

- 1) *Introducción*
- 2) *Como se inició todo*
- 3) *Qué es Smalltalk*
- 4) *Squeak es Smalltalk*
- 5) *Qué no es Squeak*
- 6) *Qué es interesante de Squeak*
- 7) *Cómo se inició Squeak*
- 8) *Filosofía*
- 9) *Licencia y Soporte*
- 10) *Características*
- 12) *Proyectos en Squeak*
- 13) *Repositorios*
- 14) *Referencias*

Introducción

Squeak es una implementación del lenguaje de programación y ambiente Smalltalk. Es moderna, altamente portable, rápida y de código abierto.

Smalltalk fue creado en Xerox PARC (Palo Alto Research Center) por un grupo de investigadores liderado por Alan Kay durante la década de los 70. Estas investigaciones tuvieron como resultados notables Smalltalk/71, Smalltalk/72, Smalltalk/76 y Smalltalk/80.

En esa época, el mismo definió el termino orientación a objetos, y es el primer lenguaje en el cual todo esta construido a partir de objetos, los cuales se comunican a través del paso de mensajes.

Smalltalk está inspirado por las ideas de Simula, Sketchpad y Lisp, e inclusive hoy en día Smalltalk guía a los lenguajes y ambientes orientados a objetos dinámicos de tipo fuerte.

El kernel de Squeak posee:

- ❖ Un lenguaje y librería base extensa compatible con Smalltalk-80 y el ANSI Smalltalk X3J20.
- ❖ Una máquina virtual rápida escrita con un subconjunto de Squeak apodado SLang.
- ❖ Una memoria de objetos compacta de 32-bits “bit identical”.
- ❖ Un recolector de basura eficiente.
- ❖ Un sistema de plugin para la máquina virtual
- ❖ Una ejecución Bit-identical incluyendo gráficos en la mayoría de las plataformas populares y sus respectivas versiones.

Por encima de esto se encuentra la librería de clases y los plugins de la máquina virtual para multimedia de avanzada incluyendo gráficos anti-aliased 2D, gráficos 3D acelerados, sonido de tiempo real y sinterización de música, video MPEG2 y mucho más. Además, Squeak tiene uno de los ambientes de desarrollo mas avanzados, totalmente reflectivos jamás creados, con más de 650 paquetes adheribles disponibles a un click para su fácil instalación.

¿Como se inicio todo?

El Dynabook fue un sistema conceptual propuesto por Xerox PARC a finales de 1960 y comienzos del 1970. Las ideas del Dynabook permitieron el desarrollo del prototipo Alto, que poseía todos los elementos de interfaz gráfica alrededor del año 1972. El componente de software de esta investigación era Smalltalk, que cobró vida propia y prácticamente se independizo del concepto del Dynabook.



El concepto del Dynabook describía lo que ahora es una laptop pero mas parecida a una Tablet PC, con una batería casi eterna y un software para brindar a los niños la oportunidad de expresión con todos los medios digitales disponibles. Los adultos podrían usar el Dynabook pero la audiencia serian los niños, y el software “crecería con ellos”.

Alan Kay fue el precursor principal del concepto del Dynabook. Cuando apareció Microsoft con su tablet PC Alan dijo: “El Tablet PC de Microsoft... la primera computadora Dynabook-like suficientemente buena para criticar”, un comentario que había aplicado antes a las Macintosh de Apple.

El hardware en el cual corría era relativamente irrelevante.

En los 90s, Alan Kay empezó a trabajar en un sistema de programación llamado Squeak, un ambiente Smalltalk de código abierto que se podría ver como la continuación del concepto del Dynabook.

Que es Smalltalk?

Smalltalk es un lenguaje orientado a objetos puro, simple y uniforme. Smalltalk ha influenciado a la mayoría de los lenguajes “modernos” de programación, a pesar de que la mayoría del tiempo, estos pierden la elegancia y simplicidad de Smalltalk.



La filosofía fundamental de Smalltalk es la de completa apertura. Todo está disponible para ser visto, entendido, modificado y extendido.

La librería de Smalltalk ha inspirado librerías muy conocidas como la de Java debido a su robustez. Esto se debe a que tiene más de 30 años de “depuración”.

La sintaxis de Smalltalk cabe en una hoja, y el modelo de objetos es simple:

- ❖ Todo es un objeto, inclusive las clases.
- ❖ Los objetos se comunican a través del paso de mensajes.
- ❖ En las clases se describe el estado del objeto a través de las variables de instancia, y el comportamiento es descrito por los métodos de la clase y sus instancias.
- ❖ Cuando un objeto recibe un mensaje, el método correspondiente es buscado en la clase y súper clases del receptor hasta ser encontrado. En caso que esto no se cumpla, se envía al mismo receptor un mensaje `doesNotUnderstand`.
- ❖ Los métodos son públicos.
- ❖ Las variables de instancia son privadas.
- ❖ El tipo de herencia es simple.
- ❖ Palabras reservadas: `nil`, `true`, `false`, `self`, `super`, `thisContext` (solo en Squeak) y `Smalltalk`.

El siguiente es un ejemplo de Smalltalk (usando una clase de GUI Morphic de Squeak) que crea una elipse, cambia su color a azul y muestra en la pantalla.

```
EllipseMorph new color: Color blue; openInWorld.
```

Squeak es Smalltalk?

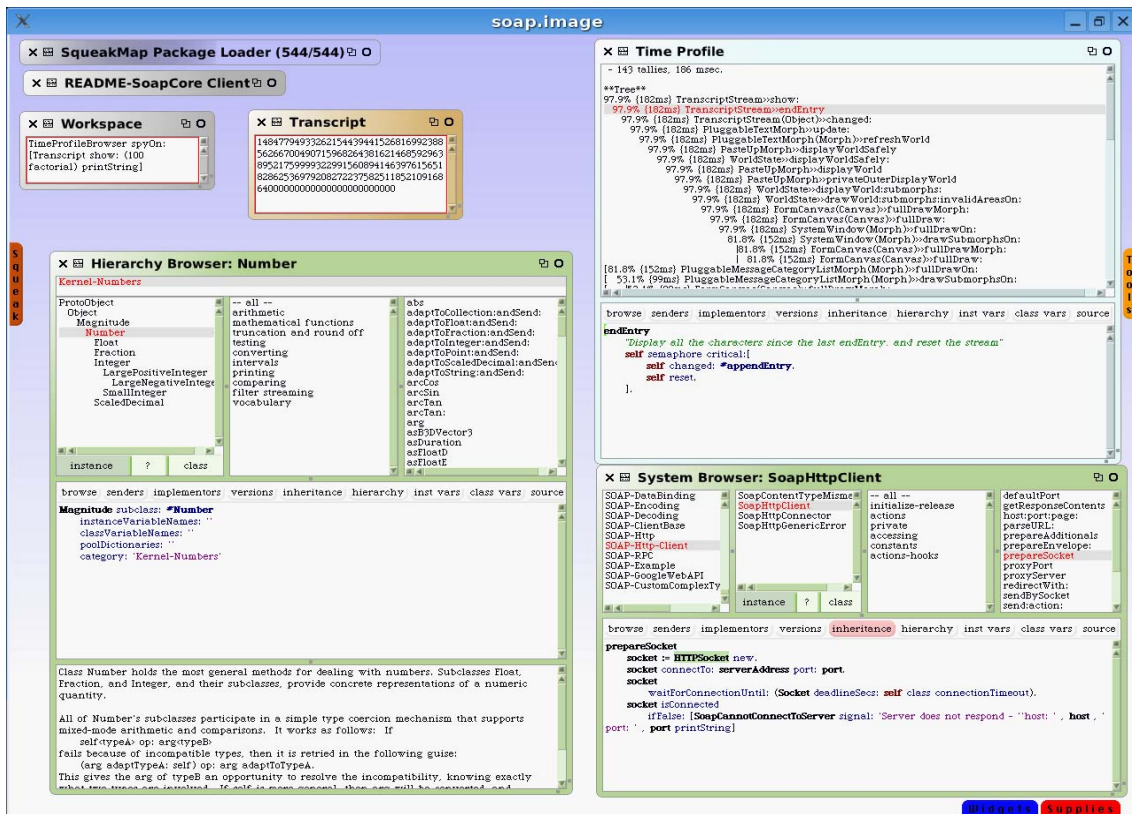
Squeak es una implementación abierta, altamente portable de Smalltalk-80, cuya maquina virtual esta escrita enteramente en el mismo. Lo que hace fácil su corrección (debugging), análisis y cambio. Squeak añade características a Smalltalk aunque se trata de mantenerlo compatible.

Para lograr un performance práctico, el traductor produce un programa equivalente en C cuyo performance es comparable a un Smalltalk comercial.

Otros aspectos notables de Squeak incluyen:

- * Sonido de tiempo real y sintetización de música. Todo escrito en Smalltalk.
- * Extensión de BitBlit para manejar color de cualquier profundidad. Además se puede rotar y escalar imagen anti-aliased.
- * Soporte de acceso a las redes, que permite una construcción simple de servidores y otros programas útiles que necesiten interconectarse.
- * Corre de manera bit-identical en muchas plataformas (Windows, MacOS, Unix, y otras).
- * Un simple aunque efectivo recolector de basura incremental para punteros directos de 32-bits.

Muchas veces se menciona a Squeak como el ambiente y a Smalltalk como el lenguaje. Aunque ya sabemos que el lenguaje y ambiente son Squeak, que posee en su gran mayoría características Smalltalk.



Que no es Squeak

El sistema Smalltalk Squeak no tiene ninguna relación con el lenguaje diseñado por Rod Pike y Luca Carnelli en 1985. Tampoco tiene relación su sucesor “Newsqueak”.

Qué es interesante en Squeak

Squeak extiende la filosofía fundamental de Smalltalk al incluir una maquina virtual. Es una genuina, completa, compacta, eficiente y robusta implementación de un ambiente Smalltalk. No es especializada para ninguna plataforma hardware o sistema operativo en particular.

El equipo principal de Squeak incluye a Dan Ingalls, Alan Kay, Ted Kaehler y Scott Wallace. Todo esto ha atraído a muchos de los mejores y más experimentados programadores de Smalltalk en el mundo.

Squeak se utiliza como un Smalltalk practico en donde desarrolladores, investigadores, profesores o estudiantes motivados pueden examinar código fuente de cada parte del sistema, incluyendo las primitivas de los gráficos, e inclusive la maquina virtual. Uno puede realizar cambios inmediatamente sin tener que ver o involucrarse con otro lenguaje aparte de Smalltalk. Squeak corre imágenes bit-identical en todas las plataformas, facilitando la colaboración en diversos ambientes. Cualquier archivo de imagen podrá correr en cualquier intérprete inclusive si fue guardado en un hardware diferente, con un sistema operativo diferente (O sin un sistema operativo!).

Squeak es un ambiente persistente. Cuando se guarda la imagen, todo es guardado en ella, todo el contexto y datos. Es como realizar una fotografía del sistema actual.

Squeak es un IDE. Se puede codificar, correr y debuguear en el mismo ambiente.

Squeak usa compilación incremental. Una vez que se guarda el código, este está disponible para ser corrido de manera eficiente. O sea, se compila a bytecodes este se interpreta por la máquina virtual, la cual puede realizar compilaciones JIT para un mejor desempeño.

Y como se inicio Squeak?

Squeak comenzó muy simple, por la necesidad de un grupo de investigación en Apple. Se quería un sistema tan expresivo e inmediato como Smalltalk para diferentes propósitos (prototipo de software educacional, experimentos de interfase de usuario). Como se puede leer en el paper OOPSLA ("Back to the Future"), se dio la idea de escribir un intérprete de Smalltalk en un subconjunto de Smalltalk, junto a un traductor de este subconjunto a código C eficiente.

Que posee la versión actual

La versión actual del intérprete de Squeak combina intérprete de bytecodes Smalltalk clásico con una simple pero eficiente memoria 32-bits de punteros a objetos y un recolector de basura incremental.

También incluye un sistema de gráficos llamado BitBlt que soporta 1, 2, 4, 8-bits de color indexado además de 16 y 32 bits de colores RGB, todo con un "warp drive" que soporta rápidas rotaciones y transformaciones. También soporta simple anti-aliasing.

Otra notable capacidad de Squeak es que incluye entrada y salida de sonido de 16-bits, y soporte de sockets y acceso general a redes.

La portabilidad y compatibilidad de Squeak, junto con su maleabilidad, han creado un gran interés en la comunidad académica, y lo que uno llamaría la "independiente" comunidad de ciencias de la computación. Con esta frase se quiere incluir a personas que no están atadas a un lenguaje, o un sistema operativo, sino a quienes tienen una pasión en particular (análisis numérico, gráficos, computación distribuida, sintetización de música, Educación, etc.) y a quienes quieren un sistema que provea el ambiente más flexible e inmediato para realizar experimentos en el campo de su interés.

Squeak es gratis, con una licencia liberal

Smalltalk fue creado en Xerox PARC en los años 70s. Apple obtuvo la licencia en 1980. Un equipo de Apple desarrolló Squeak en 1996 y está disponible gratis bajo una licencia. El acuerdo de la licencia tiene como propósito mantener a Squeak abierto y disponible a la comunidad de usuarios mientras que también permita a los usuarios hacer cosas útiles con Squeak. A continuación se incluye parte de esta licencia:

Se esta permitido cambiar Squeak, escribir extensiones de Squeak, construir una aplicación en Squeak, e incluir alguna parte o todo Squeak con tus productos. Se puede distribuir todas estas cosas con Squeak, o porciones de Squeak, de manera gratuita o por dinero. Sin embargo, se debe distribuir estas cosas con una licencia que protege a Apple de la forma descrita en esta licencia.

Si se modifica cualquiera de los métodos de la clase de objetos o sus relaciones que vienen con Squeak (contrario a construir sobre las clases en el release), se debe publicar las modificaciones en un sitio web o de otra manera hacerlas disponibles gratis a otros, de la misma manera que ha sido hecho con Squeak. Lo mismo es verdad si se porta Squeak a otra máquina o sistema operativo. Se debe portear en un sitio web o de otra manera hacerlo disponible de manera gratuita para los demás bajo los mismos términos de esta licencia.

Soporte para Squeak

Estándares oficiales y soporte a productos son enemigos del cambio. Acceso universal, maleabilidad son las figures principales con mayor mérito para Squeak. Es nuestra intención que Squeak evolucione.

Algunas personas se sienten tentadas acerca del uso de un sistema que parece ser dependiente del entusiasmo caprichoso de un grupo de magos. ¿Quién podría hacer planes para un producto en estas arenas movedizas?

La respuesta es simple: Cada release de Squeak incluye todo acerca de si mismo: la imagen, la maquina virtual, con el código fuente completo para cada uno. Si el equipo de Squeak se desvaneciera o desapareciera, probablemente el 25% de los integrantes de la lista de correo de Squeak podrían mantener la versión actual simple, prácticamente realizada a mano por los siguientes 20 años aproximadamente. Es mejor que tener soporte.

Características (Features)

Esta lista de características es impresionante. Lo que en otros ambientes tendría como extensiones, en Squeak estas ya se encuentran completamente integradas.

Los usos más comunes para Squeak son:

- ❖ Desarrollo
- ❖ Educación
- ❖ Diversión y juegos
- ❖ Investigación
- ❖ La maquina virtual de Squeak

Desarrollo

Squeak posee todas las características de un ambiente moderno de desarrollo:

- * Código coloreado y reformateado automáticamente gracias al paquete Shout. Además el código inválido es marcado por este mismo paquete.

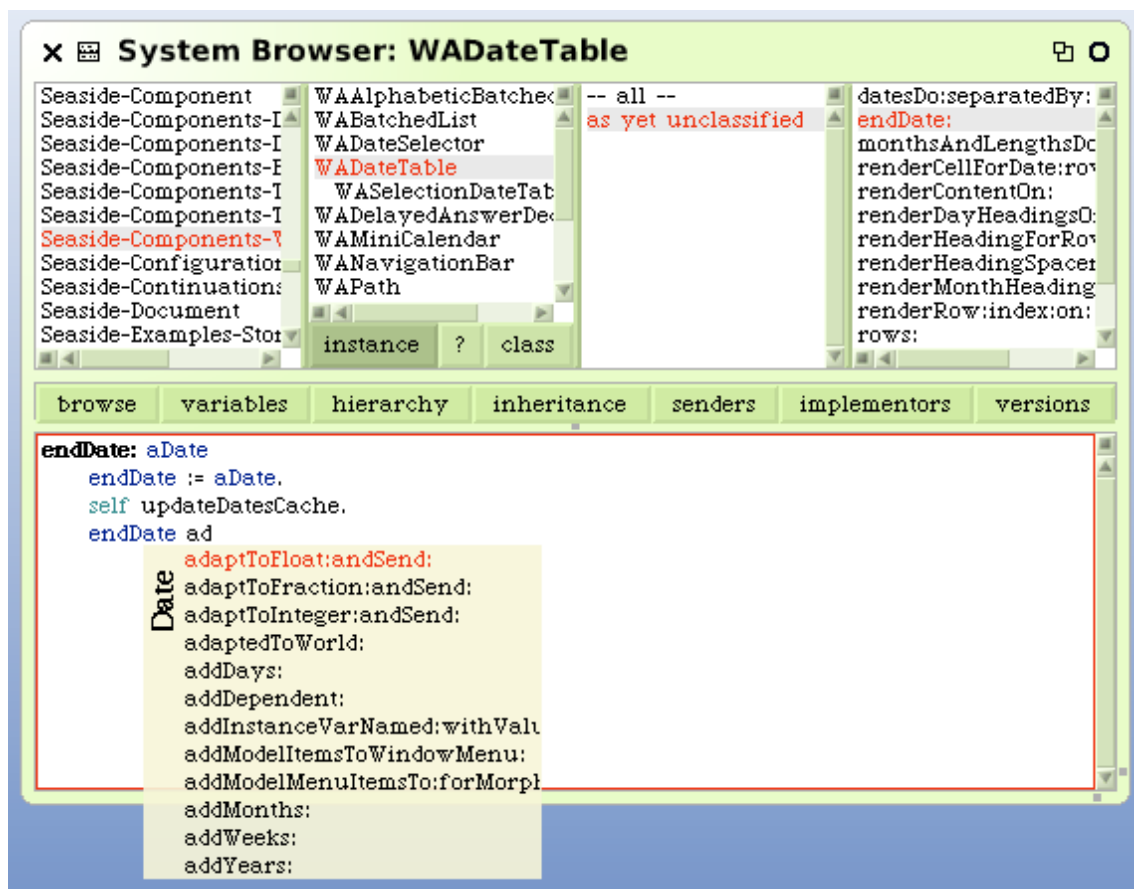
- * SLint es un paquete que testea el código a partir de un conjunto de reglas de calidad.

- * Squeak posee un Browser para realizar refactoring, a pesar de que también posee una versión básica poderosa.

- * El compilador de Smalltalk ya se encuentra integrado y realiza la compilación de forma incremental.

- * El ambiente Squeak es un sistema completamente vivo (se puede ver como un gran debugger). Se puede inspeccionar cualquier parte del sistema en cualquier tiempo, por más que el código se encuentre ejecutando.

El desarrollo usando Squeak no necesita ser realizado con el clásico ciclo de escribir, compilar y correr. En vez de esto se interactúa con la imagen de Squeak directamente, la cual contiene todo el código y cualquier objeto que este necesite instanciar. Es posible también cambiar la implementación de un objeto mientras este se encuentra activo.

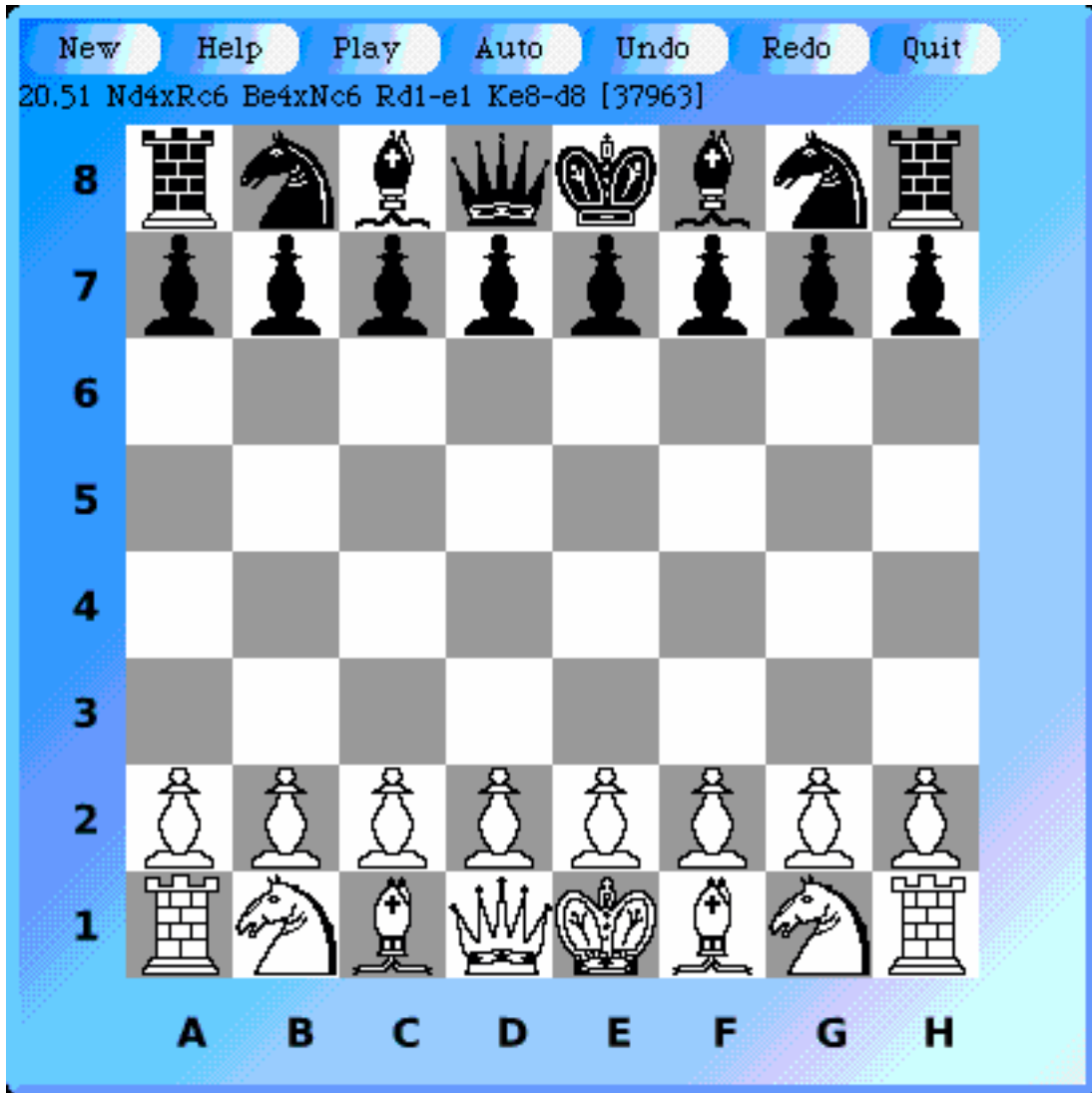


Educación

Si se es un estudiante, padre o profesor se puede estar interesado en SqueakLand.org. Este contiene proyectos educativos interesantes y el sistema E-toys posee potenciales increíbles.

Diversión y juegos

Squeak es una plataforma excelente para el desarrollo de juegos. El código fuente de todos estos ya se encuentra incluida en la imagen.



También existen otros juegos como:

- ❖ Cipher Game
- ❖ Crostic
- ❖ Mines
- ❖ Chess

- ❖ Sokoban
- ❖ War Game
- ❖ etc.

Multimedia

Squeak puede presentar información en una variedad de formatos digitales (audio, video, interactividad, gráficos y animación) lo que provee a los creadores un vehículo para que el usuario comprenda el mensaje fácilmente.

Squeak grafico

En squeak los gráficos en 2D los maneja el Framework Morphic y los gráficos 3D los hace el ambiente Alice 3D.

Gráficos 2D

El modelo de gráficos standard de Squeak esta basado en BitBlit de Dan Ingalls, que ha sido extendido para dibujar líneas, mostrar texto y realizar transformaciones con escalaciones, rotaciones, etc. para 1, 2, 4, 8, 16 y 32 bits de profundidad. Las clases incluidas pueden leer y manipular BMP, LPEG, GIF, TIFF y PNG, entre otros.

Procesamiento de Video e imagen puede ser realizado para adherir filtros a las imagines y al video (VideoFlow).

Gráficos 3D

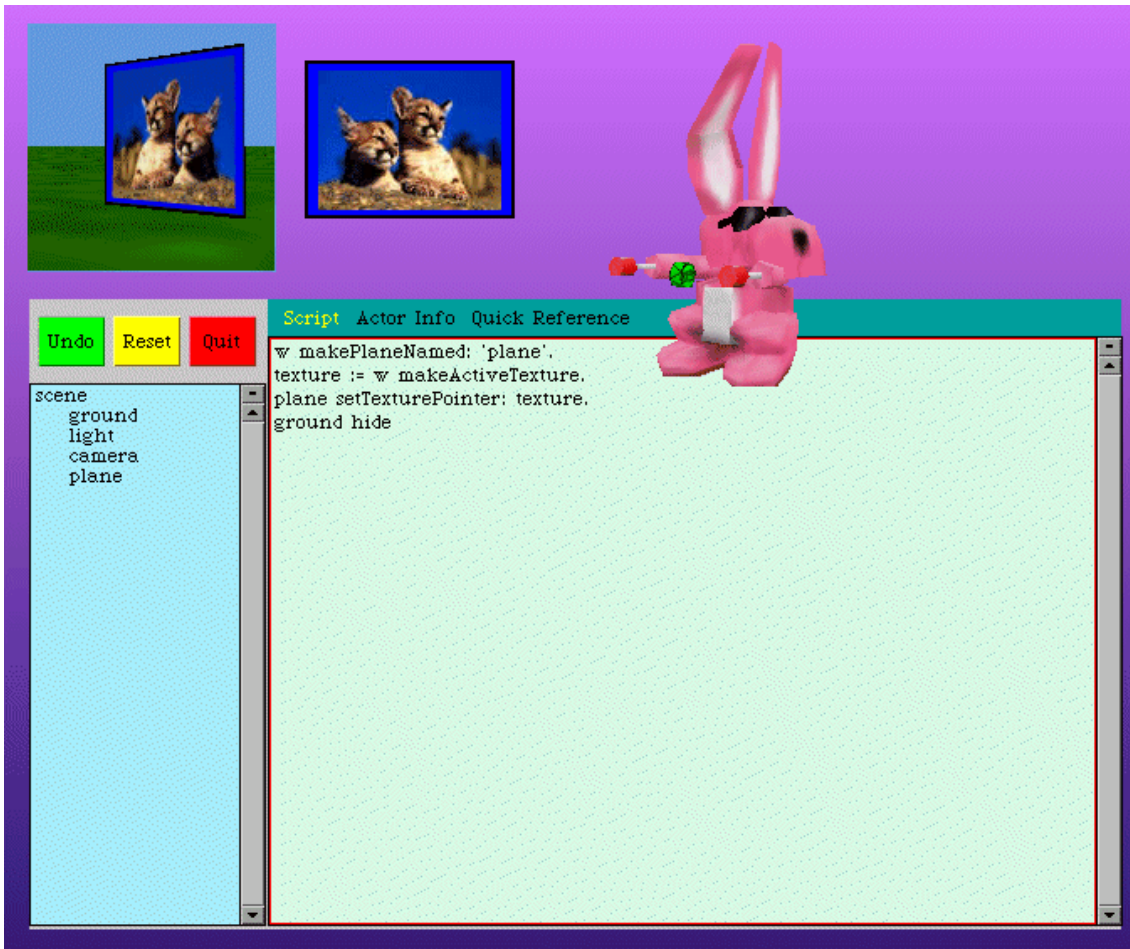
El motor de gráficos vectoriales 2D es Balloon. La extensión 3D es llamada Ballon3D. Con el mismo se pueden crear aplicaciones 3D con gran facilidad. El motor usa aceleración de hardware cuando está disponible.

Alice

Alice es una herramienta diseñada para realizar auditoria. Esta pensada para personas sin ninguna experiencia de programación. Con ella pueden crear mundos 3D interactivos. El Stage 3 Research group empezó el proyecto en la Universidad de Virginia y continúa su trabajo en la Universidad Carnegie.

Squeak-Alice trae las ideas e ideales de Alice a Squeak, con la finalidad de introducir mas personas a los gráficos 3D. Wonderland es el bloque básico de Squeak Alice y son mundos 3D interactivos. Para crear un Wonderland se puede evaluar:

Wonderland new.

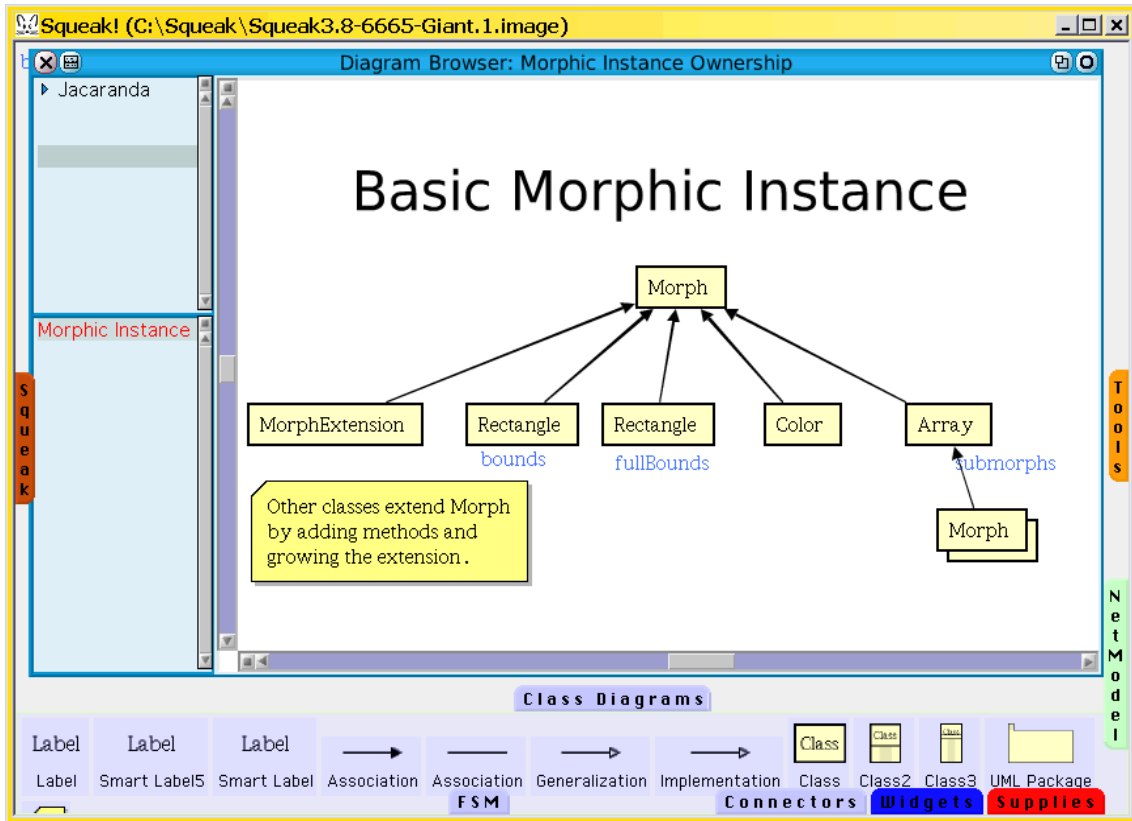


Esto crea un editor de script, un control de panel y una ventana del mundo 3D. Desde aquí uno puede empezar a construir el mundo.

Interactividad

Morphic

Morphic es un kit de construcción y manipulación de Interfase de usuario basado en display trees. Es una alternativa al tradicional MVC (Todavía disponible en Squeak).



Audio

Música

Squeak posee muy buenas capacidades para el sonido, desde reproducir archivos MIDI (a través de un sintetizador interno o un sintetizador externo compatible) para generar múltiples instrumentos y voces a la vez, además de grabar sonidos y reproducir una gran variedad de archivos de sonido (WAV, AIFF, etc)

Speech! Speech!

Squeak incluye KlattSynthesizer, un Sintetizador paralelo en cascada de estilo Klatt. Tan solo hay que fijarse en los métodos de clase “ejemplo” de la Clase DECTalkReader y Speaker.

El sintetizador esta linkeado a un sistema de generación de caras.

Video

Squeak soporta mpeg2 y un formato de compresión de video jpg interno .

Investigación

Squeak es un ambiente muy flexible que permite a los programadores realizar técnicas para explorar diferentes paradigmas de programación.

Proyectos de investigación que usan:

- ❖ LSCTalk: de Serge Stinckwich es un learning classifier system platform para multi-agent learning.
- ❖ Classboxes: es un modulo para lenguajes orientados a objetos que permite adherir métodos o reemplazarlos.
- ❖ Traits: son simples mecanismos de composición para estructurar programas orientados a objetos.
- ❖ MetaclassTalk: de Noury Bouraqadi es una extensión reflexiva de Smalltalk que provee al programador con un protocolo de meta objetos (MOP) para controlar la estructura de los objetos (Alocación de memoria y acceso a variables de instancia) y comportamiento (envío de mensajes y recepción y métodos de búsquedas (lookup) y aplicaciones
- ❖ AspectS: de Robert es una extensión del ambiente Squeak para habilitar programación orientada a Aspecto. PerspectiveS, que esta construido encima de AspectS, introduce capas de comportamiento dinámicas al ambiente de squeak.
- ❖ FracTalk: de Noury Bouraqadi es una implementación del Modelo de componentes Fractales (Fractal Component Model).
- ❖ El proyecto QuA ha hecho un prototipo de arquitectura de componentes de software de Squeak que soporta desarrollo basado en componentes para desarrollo de aplicaciones multimedia de tiempo real.

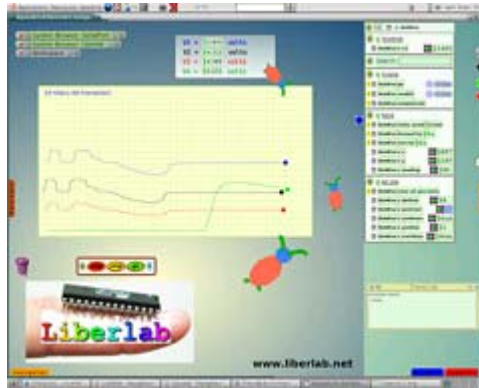
Proyectos

Desarrollo

- ❖ SqueakFoundation es una fundación que da soporte a Squeak.
- ❖ Monticello es un sistema de manejo de paquetes. Más como un repositorio. (Similar a CVS)
- ❖ SqueakMap es un catalogo global de paquetes desarrollados en Squeak. Soporta desarrollo distribuido.
- ❖ SUnit Tests es un framework para realizar Unit Testing. Este ya se encuentra en el ambiente standard de Squeak.
- ❖ Refactoring Browser es un browse poderoso que permite al programador realizar Refactoring de código fuente. Renombrar métodos de manera segura, cambiar nombres de las clases.
- ❖ Magma es una implementación de una base de datos orientada a objetos.

Educación

- ❖ Squeakland es un sitio que soporta el uso de Squeak como herramienta educativa.
- ❖ Etoy es un ambiente poderoso basado en scripts para aprender ciencias y matemáticas (entre otros).



- ❖ Small-land es un proyecto español el cual corre en más de 80.000 PC en el citado país.
- ❖ Scratch es un toolkit programable que permite que los niños creen sus propios juegos, historias animadas y arte interactivo, además de poder compartirlos a través de la red. Scratch está construido sobre el tradicional LOGO/Lego pero toma las ventajas de las nuevas ideas para que este sea más fácil de aprender y usar.
- ❖ ComiKit es un toolkit para niños que usa un lenguaje de programación visual basado en strips de comics. Con ComiKit se puede crear juegos interactivos e historias con personajes animados. Este es el resultado de una investigación con lenguajes de programación visuales en la Universidad de Linköping.

Desarrollo de aplicaciones Web

- ❖ Seaside es un framework basado en componentes y flujo de datos para desarrollar aplicaciones web altamente dinámicas y sofisticadas. Utiliza continuations y el URL para preservar el estado de la aplicación inclusive en browsers diferentes, facilitando ampliamente la complejidad. Se puede utilizar Scriptaculous para realizar aplicaciones Web 2.0.

Kongresové centrum - Rozpis pro den 23.5.2005 24.6.2005 >>

<< Zpět | Tisk

čas	ps	s2	s3	r2	sa	ms	r1	out
0								
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								

Testovací kongres

Tisk (recepce, strava, rozložení, technika) | Klonuj

Popis: Testovací kongres

Firmy: [Nová Najit](#)

Sevas, spol. s r.o.

Osob:

Stav: Předběžné jednání 16.05.2005

Platba: fakturou

Cena:

Termíny: 23. 5. 2005 9:00-18:00
24. 5. 2005 9:00-18:00
24. 5. 2005 9:00-18:00

Akce: **B I U**

23,5,2005 9-18 s1 s2 s3 ps
24,5,2005 9-18 s1 s2 s3
9-18 s1 s2 s3

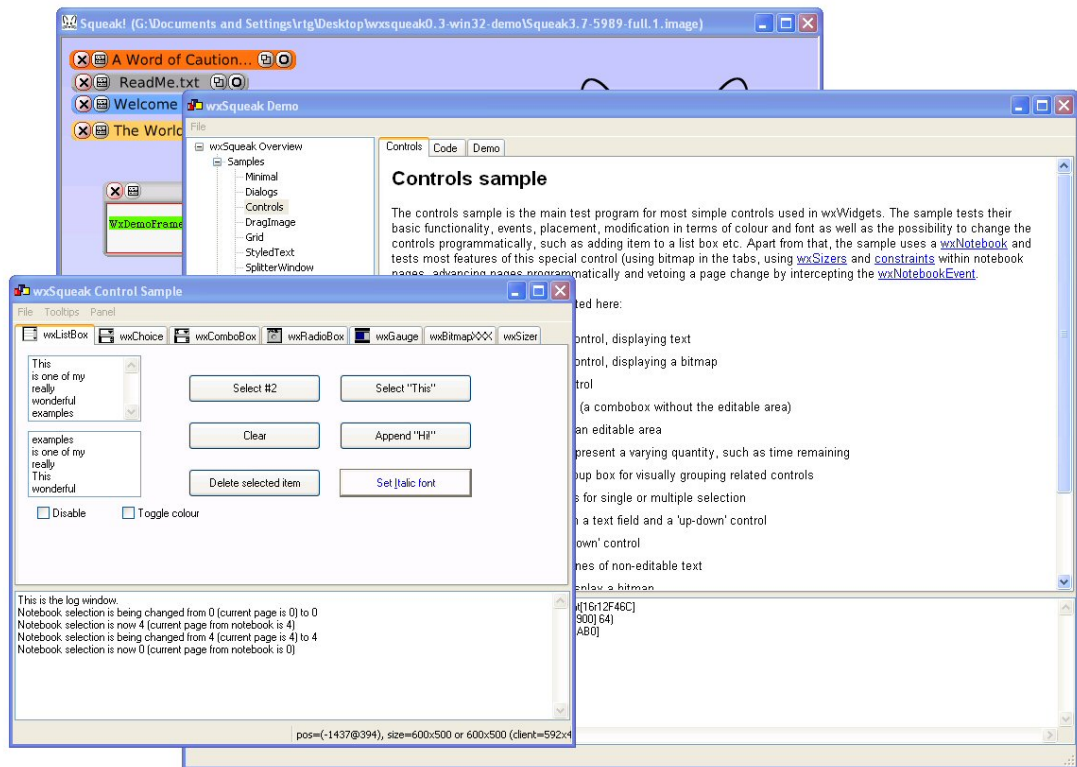
čas	ps	s2	s3	r2	sa	ms	r1	out
23. 5. 2005								
9:00-18:00								
s1 s2 s3 ps								
24. 5. 2005								
9:00-18:00								
s1 s2 s3								
9:00-18:00								
s1 s2 s3								

Firmy: [Sevas, spol. s r.o.](#)

Termín: 23.5.2005 - 24.5.2005

- ❖ SmallWiki2 es un wiki basado en Seaside
- ❖ Magritte es un meta-model framework que se utiliza para crear vistas, reportes, persistencia y editores del modelo de forma automática sin utilizar generación de código.
- ❖ Pier es un wiki extendido que utiliza Seaside y Magritte.
- ❖ ShoreComponents es un conjunto de componentes para Seaside.
- ❖ Asteroid es una implementación de Comet escrita en Squeak.
- ❖ ST2JS es un traductor de Smalltalk a Javascript.
- ❖ Croquet es un ambiente de auditoria 3D peer to peer para desarrollo simultáneo de simulaciones interconectadas y modelos, en equipo.
- ❖ Sophie es un proyecto para desarrollar un sistema de auditoria de código abierto para documento electrónicos ricos en medios.
- ❖ Tweak es un ambiente para niños de todas las edades. Es Morhic rehecho con flexibilidades adheridas, abstracciones más poderosas y claridad de código. Morhic es un kit de construcción de interfaces de usuario de manipulación directa basado en mostrado de árboles. Trabaja con objetos gráficos llamados Morphs. Reemplaza el toolkit grafico MVC original del Smalltalk 80.

- ❖ wxSqueak es una interfase Squeak para la librería wxWidgets utilizada para realizar aplicaciones GUI con widgets nativos del sistema donde corra.



Más proyectos

- ❖ Quoth es un sistema dinámico e interactivo de ficción tirando a la codificación en vivo musical.
- ❖ * ODECo es un toolkit fácil de usar para realizar simulaciones dinámicas 2D/3D basadas en ODE (<http://www.ode.org>), un simulador físico de movimiento.
- ❖ NetMorph es un agente móvil visual. En el ambiente, morphs con cualidades de red pueden moverse libremente en el World (escritorios Squeak de varias maquinas en la red) para lograr algún tipo de colaboración.



- ❖ Traits es un proyecto que adhiere capacidades orientadas a aspecto (Aspect oriented) a Smalltalk.

- ❖ BabyUML es un laboratorio para explorar nuevas disciplinas de programación.
- ❖ Spoon es un proyecto para crear aplicaciones a partir de un correo o parte esencial mínima que dinámicamente crezca su librería de clases a medida que son usadas.
- ❖ Exupery es un proyecto para mejorar la compilación y ejecución de bytecodes.

La Maquina virtual de Squeak

La maquina virtual de Squeak esta escrita en Slang, que es un subconjunto funcional de Smalltalk capaz de ser traducido a código estándar de C eficiente. Squeak usa el lenguaje C como un lenguaje ensamblador multiplataforma.

Desde que Slang es un subconjunto de Smalltalk, la maquina virtual de Squeak puede ser editada y debugueada al correrla en el propio Squeak.

La maquina virtual tambien puede ser extendida con plugins, escritos en C o Slang. Estos son usados en Squeak para reproducir MPEGs y para realizar encriptaciones de clave pública, capacidad que ya se encuentra integrada a Squeak.

A continuación se listan las plataformas que poseen máquinas virtuales para correr imágenes de Squeak:

- ❖ Windows
- ❖ Win32
- ❖ WinCE
- ❖ Mac
- ❖ MacOSX/Classic
- ❖ Unix
- ❖ VM Source
- ❖ Sparc
- ❖ FreeBSD
- ❖ NetBSD
- ❖ OpenBSD
- ❖ Linux
- ❖ Darwin

- ❖ PPCLinux
- ❖ RiscOS
- ❖ RiscOS
- ❖ Squeak Sources File

Repositorios y código compartido

A continuación se listan los tipos de repositorios de inestable a estable.

* Las imágenes son las primeras formas de obtener código, y algunas veces contenido en las manos del usuario. Son como snapshots de un sistema en ejecución. Sería como una instalación de linux mas los paquetes de interés, totalmente configurados, con una cierta cantidad de aplicaciones corriendo. Todo esto listo como para que se resuma la computación donde esta ultima se quedó. Un numero de imágenes son mantenidas y publicadas periódicamente incluyendo las oficiales (3.7, 3.8,3.9)., las imágenes squeakland y las imágenes específicas de los proyectos (croquet, tweak, seaside, smallwiki, spoon..).

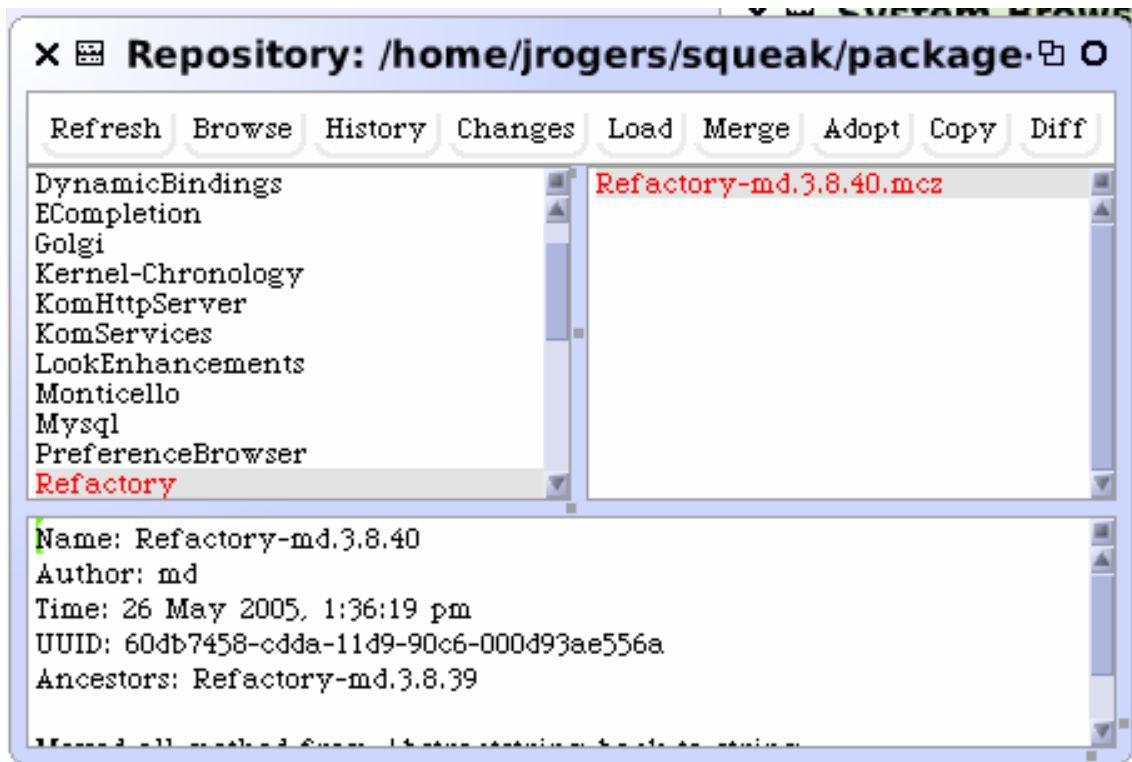
* SqueakSource es como sourceforge.net. Provee un servicio central de hosting para el código en desarrollo, con una revisión control llamado Monticello (y posee rss feeds, project wikis, etc.).

* SqueakMap es como freshmeat.net. Es un directorio central que provee información estandarizada primariamente acerca de todos los tipos de paquetes bajables, pero tambien acerca de los desarrolladores de Squeak y otros ítems de interés. Los paquetes pueden ser subidos a squeakmap o linkeados desde cualquier lugar. SqueakMap tiene un cliente Morphic sincronizado (SqueakMap Package Loader)

* Package Universe es como una distribución gnu/linux como lo es debian. Es un conjunto de paquetes que han sido seleccionados de squeakmap y han sido mas trabajados, testeados, arreglados y se ha adherido información de dependencias para que se pueda instalar cualquier paquete y estos funcionen sin inconvenientes.

* Update streams, que son proveídos por algunas imágenes squeak mayores (major squeak images), es la forma principal de instalar updates y códigos nuevos, como el update stream de debian. Son usadas mayormente para actualizaciones muy necesarias como las de seguridad

Los límites entre estos tipos no siempre están bien delimitados ya que cada uno va copiando características del otro.



Conclusión

Squeak es un ambiente abierto en el cual uno puede realizar cualquier tipo de tarea. Desde desarrollo profesional a investigación y educación. Posee grandes prestaciones y es una plataforma ideal para probar conceptos e ideas. Es fácil de aprender y muy útil.

Referencias

- ❖ Squeak License: <http://www.squeak.org/SqueakLicense/>
- ❖ Slang: <http://minnow.cc.gatech.edu/squeak/2267>
- ❖ Squeak: www.squeak.org
- ❖ Smalltalk: <http://en.wikipedia.org/wiki/Smalltalk>

Bibliografía

Programando con Smalltalk

DIEGO GÓMEZ DECK

Back to the future - The Story of Squeak, A Practical Smalltalk Written in Itself

Kay, Alan

Smalltalk with Style

Skublics, Suzanne. Klimas, Edward Thomas, David

Prentice Hall. 1996.

Smalltalk, Object and Design

Chamond, Liu

New York. 1996.

Squeak - Object-Oriented Design whit Multimedia Applications

Guzdial, Mark

Prentice Hall. New Jersey. 2001.

Squeak. Open Personal Computing and Multimedia

Guzdial, Mark. Rose, Kim