

Learning with Squeak Etoys

Cathleen Galas, c.galas@uclalumni.net
 Squeakland Foundation, Galas Consulting

Rita Freudenberg, rita@isg.cs.uni-magdeburg.de
 Dept of Computer Science, Otto-von-Guericke-University Magdeburg, Squeakland Foundation

Abstract

This paper serves as introduction and overview of Squeak Etoys, an engaging computer media-authoring environment, and describes how it aids in the constructionist approach to learning, thinking, and education. Etoys development was inspired by LOGO, the constructionist ideas of Seymour Papert, and Piaget, Bruner, and Montessori. It was developed to help student exploration and discovery in learning and thinking deeply about powerful ideas in math and science.

The design of Etoys includes a powerful user interface that will run on all platforms and allow users to author in multiple dimensions. Etoys is an object-oriented system that is built in Squeak, an open-source implementation of Smalltalk. The learning environment is an open world, a sandbox, ready for the creative visions of the user, and undisturbed by buttons and toolbars. Resources are readily available, a click away, stored in flaps, hidden windows that open to many new possibilities. Beginning activities involve creating and scripting objects, building collaboration between objects, and building, exploring, and sharing projects. The user interface includes modeless editing, so that users can edit anything and at any time in project work.

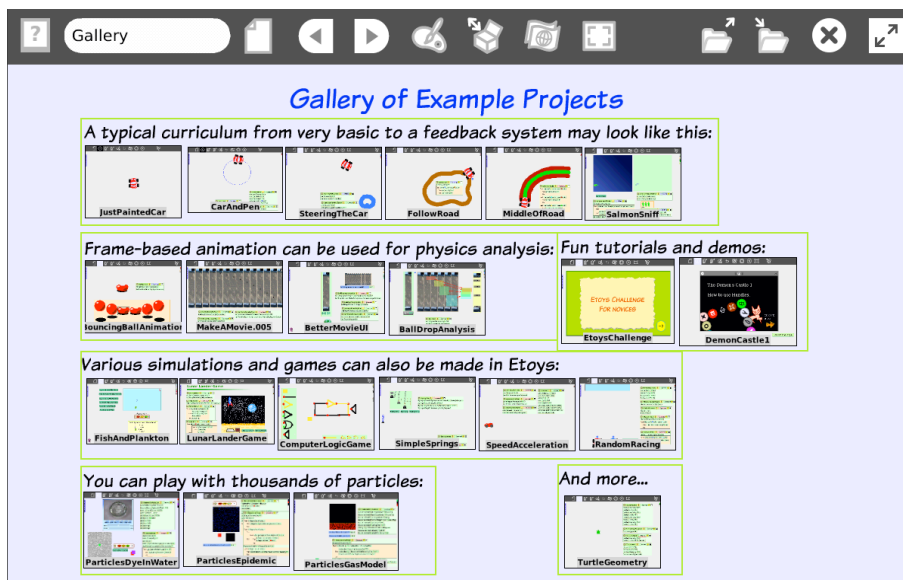


Figure 1. Gallery of example projects, which are shipped within Etoys

Keywords

Squeak Etoys, media-authoring system, constructionism, constructivism, constructionist application, objects-to-think-with, Seymour Papert, Alan Kay, Squeakland, powerful ideas

Introduction

Squeak Etoys is an engaging computer, media-rich authoring environment and visual programming system for children built to help student exploration and discovery in learning and thinking deeply about powerful ideas in mathematics and science. Squeak Etoys includes a wide variety of dynamic media that can be easily created, scripted or programmed, and shared publicly or used for collaboration with others. The “world” is a sandbox, a safe place for non-programmers, an open, flexible, environment in which to create, modify, script, and make objects interactive. It allows children to encounter and learn ideas by playing with them and constructing their own objects visually, kinaesthetically, and symbolically. Children’s representations can exceed their oral communication abilities, and Etoys lowers the threshold for exploring complex cognitive skills that are normally beyond children’s developmental stages, while also lowering the learning curve to work with powerful programming capabilities.

The basic idea behind the design of Etoys is the powerful user interface that will run on all platforms and allow users to author in multiple dimensions. Etoys is an object-oriented system that is built in Squeak, an open-source implementation of Smalltalk. Everything in Etoys is an object, the system comes with objects, and objects can be easily made, copied, manipulated, changed, and scripted with behaviors. The learning environment is an open world, ready for the creative visions of the user, and undisturbed by buttons and toolbars. Resources are a click away, stored in flaps, hidden windows that open to many new possibilities. Beginning activities involve creating and scripting objects, building collaboration between objects, and building, exploring, and sharing projects. The user interface includes modeless editing, so that users can edit anything and at any time. Projects can be published to keep locally, or to share directly to the Squeakland website.

Etoys is an educational tool for use in both classroom and informal learning environments. In the classroom, Etoys can be used on any point on the constructionism continuum, allowing purely learner-centered activities, or providing a context for teacher-directed instruction. Completed projects, available from the Squeakland website Showcase, classroom peers, teachers, or prior projects of the same user, offer opportunity for deconstructions, as children can open the viewer and see how the project is made, make their own changes to objects or scripting, and build upon others’ or their own work.

Brief History of Squeak Etoys development

Squeak is an open-source, object-oriented, Smalltalk language implementation that was derived from Smalltalk-80 at Apple Computer by Dan Ingalls in 1996, with Alan Kay and his Apple research group. (Ingalls, 1996) Squeak then migrated to Disney Imagineering Research as an open source project. Alan Kay was influenced by Seymour Papert and LOGO and directed the Etoys development to support constructivist learning. Etoys is an object-oriented application written on top of Squeak that describes object behaviors by tile-scripting and drag-and-drop operations.

Etoys influenced the development of another Squeak-based educational programming environment known as Scratch. Scratch was developed at MIT with Mitchell Resnick, and John Maloney of the original Etoys development team at Apple.

Squeak Etoys migrated to Viewpoints Research, Inc. in 2001, <http://www.vpri.org/>, a research foundation to improve education for the world’s children and advance the state of systems research and personal computing, when Alan Kay left Disney Research. Seymour Papert, Nicholas Negroponte, and Alan Kay worked toward furthering the constructivist learning in the developing world through the One Laptop Per Child, OLPC, initiative in 2005. In 2006-2008, Etoys was adapted to Sugar, the platform of the OLPC project. It is an integral part of Sugar and comes preinstalled on all OLPC XO-1 educational machines (B. Freudenberg, 2009).

In 2009, the Squeakland Foundation, www.squeakland.org, was created by Viewpoints Research, Inc., as an initial step in launching the foundation to continue encouraging development and use of Etoys as an educational medium. The Squeakland Foundation is a non-profit organization with a board of thirteen members, including educators, computer scientists, business, and researchers. The Foundation strives to build community around Etoys as free, open-source software, encourage deep learning worldwide using Etoys as the medium, and support, promote, and improve Squeak Etoys and related educational media.

The current version of Squeak Etoys was recently released as Etoys 4.0. This version contains three major improvements in the environment: Children can directly share projects from Etoys to the Squeakland website, "Etoys-To-Go" is a portable Etoys that works on any computer and is easily copied to different machines without installation, and other steps have been taken to make Etoys a completely free and open source system.

Educational Underpinnings

Etoys as an environment for education was originally inspired by the ideas of Seymour Papert (Papert, 1980), Piaget, Montessori, Dewey, Vygotsky, and Jerome Bruner. In 1968, Alan Kay developed the idea of the Dynabook for children after meeting Papert at MIT and watching Papert work with children in LOGO. At Xerox PARC, Kay and Dan Ingalls conceived and built Smalltalk, the first object-oriented language. When Ingalls later built the first implementation of Squeak in 1996, Kay saw this as an opportunity to build on the ideas of Papert and LOGO for a more sophisticated educational environment for children to explore deeply powerful ideas in mathematics and science.

Etoys is both playful and serious; it is not simply a tool to learn by making, but an environment providing opportunities for thoughtful, deep learning. Etoys allows users to create, publish, and share their creation with others, and to collaborate with users around the world.

Squeak Inheritance

Many systems influenced the development of Etoys, including LOGO, HyperCard, Smalltalk, and StarLogo. Once Squeak itself was developed, the development of a simpler authoring tool for object environments including tile scripting became possible. Squeak Etoys is similar to LOGO, but includes costumes and multimedia objects. It is similar to StarLogo, but includes levels of scale. It is like HyperCard or PowerPoint, but it is simpler to use and richer in the multimedia objects it contains. It is like Smalltalk because it is Squeak, a Smalltalk implementation, underneath (Kay 2007).

The LOGO language, especially turtle graphics, influenced Etoys through the idea of visualising algorithms while drawing vector graphics. Every newly created Etoys object owns a pen, which can be set up and down to draw a line when the object is moving. Turtle graphics is part of the Etoys system and creating an object and putting the pen down is a good beginning exercise to situate the user in the world.

The HyperCard system, released in 1987, introduced an easy-to-use programmable environment, described as "Object-Like Programming". This system allows its users to author their applications without typing any commands. Users design new cards and model dependencies between cards using functions from a menu. HyperCard gives the user a lot of flexibility, freedom, control, and power (Shafer 1988). Etoys user interface allows users to author and edit anytime and anywhere in the project; it gives all the power to the user, since it allows them to completely reprogram the system.

StarLogo is an agent-based system developed by Mitchell Resnick at MIT. StarLogo allows users to explore decentralized systems like ant colonies or bird flocks through the modelling environment. StarLogo is a version of the LOGO language that allows the control of multiple

turtles in parallel. (<http://education.mit.edu/starlogo/>) StarLogo was the inspiration for the particle system in Etoys called Kedama.

Basics: Drive a Car

Etoys is a constructionist tool built to help children construct, reconstruct, and deconstruct their understandings of the world in ways that incite curiosity, questioning, and amplification of learning big ideas in powerful ways. The simple, yet powerful user interface, provides and extends most personal computer needs, to: “runs everywhere on everything” and allows end-users “authoring and access to all things”. (Kay, 2005)

In the traditional, Alan Kay “Drive a Car” Etoys curriculum, playing with a purpose to create and drive a car offers opportunities to interact with mathematics of the real world, an unfortunately rare occurrence in many traditional educational settings. Students don’t necessarily reflect on their learning by describing the fact that they are programming on the computer, or that they are learning math, but rather, they are making a car and driving it!

The drive a car curriculum provides children with authentic problems to be solved so that something works better. The idea of using gear ratios to make a smoother car ride is an example. One must have an understanding of how the concept of gears is applied to the problem. By experimenting with different ratios, and likening it to the child's world using bicycles and gears climbing a hill, students develop visceral kinaesthetic understandings of big concepts. The problem that they can't control the car is an authentic problem within Squeak Etoys. Driving the car successfully is personally motivating and authentic to the user. (Galas, 2001) It is the car they have taken time to create and now they will be motivated to spend time solving the problem of how to drive it. The educational system or the teacher does not impose this problem; the user wants to make their own construction driveable. Solving these problems requires users to make decisions, make choices and test out their hypothesis to solve the problem. They spend time, thought, and energy in the pursuit of “hard fun”. (Papert).

Creating Objects

The Etoys system comes with an assortment of objects the user can readily use, and users can create new, similar kinds of objects anytime and anywhere in the system. Users can take an included object and easily change the color, size, and costume, a scripting term used to describe the objects appearance. The objects scripts can be easily changed, as well as its interaction with other objects.

To create a new object, the user just makes a painting and clicks on “Keep”. The painting tool provides a new, blank sheet of paper, to draw your idea. Each new object can immediately be scripted, and many objects can be scripted separately, and set to run separately, or at the same time. Existing objects can be easily redrawn; the user can “change the costume” of the object at whenever desired. Objects include everything in Etoys, from the supplied objects to interface objects, text, fonts, shapes, arrows, balloon help, sound recorders, movies, and menu items. This is one of the powerful ideas of the structure of Etoys, the idea that you can handle every object, including all media, in Etoys, in the same way, and with the same power.

In contrast to many other systems, the creations of the users are always the focal center. The entire screen, in Etoys called the “world”, is available for creation. The user Interface is built so that the functionality of an object and opportunities for object manipulation will be revealed when right-clicking on an object. This allows Etoys to be modeless, so users don’t need to switch between “editing mode” and “running mode”.

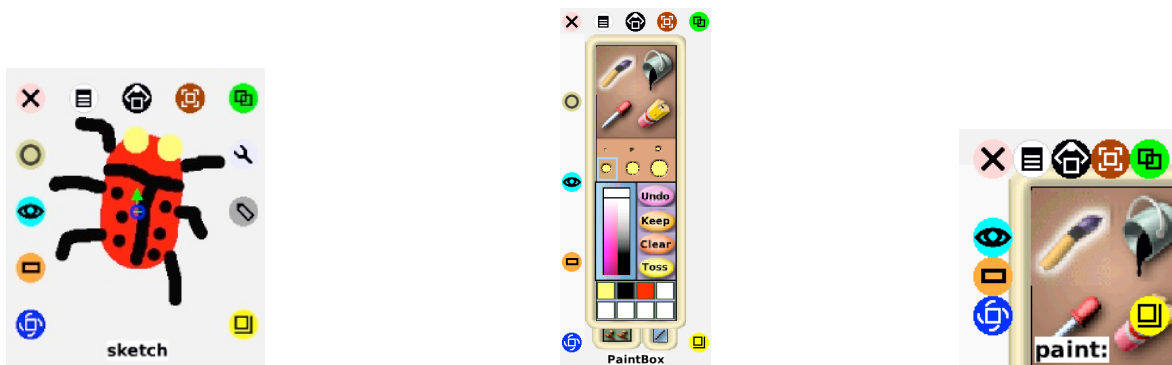


Figure 2. Halo for painted object, PaintBox and Brush

Scripting

Scripts bring the objects the user has created to life by telling the object what to do. Besides copying, rotating, resizing, embedding, or deleting, the most powerful way to manipulate the objects is to script them. Dragging the tiles onto the desktop, and then dragging the tiles into the script makes scripts. By scripting in this way with tiles, the syntax is always correct and will always work when the script ticks, or is started. The turquoise halo opens the viewer, which shows properties and behaviors of the object, and is always available for analysis or change.

Anytime the user wants to view object attributes, they can open the object's viewer; it will reflect the current state of its object. For instance, the property "heading" shows the direction an object is pointing, and the number in the viewer corresponds with the actual graphical direction of the object on the screen. Changing the number or rotating the graphical object will always update each other. The powerful idea here is to learn that there can be different representations of the same underlying data. Exploring with objects using heading is a powerful idea that bridges the known reality of the user to the computer, allowing direct manipulation and change in perception of the real world. In this process of exploring and discovering, users begin to comfortably represent their world symbolically.

To build scripts, the user arranges tiles from the viewer in a scriptor. The numbers in the tiles can be changed, and several tiles can be combined. The script itself is a Squeak Etoys object, so the user can get its handles and access the script's viewer, and use a script written for another object like the car. The user could make the scriptor, an object itself, leave a pen trail and make a circle or show any other path the user scripted.

There can be several scripts for every object, and all scripts can be executed at the same time, timed to start at different times, or manually started at different times. Child users often have an easier time with parallel processing than adult users. It is also possible to stop a script, make changes and start it again, while other scripts are still running.

When 10-year-old children create scripts similar to the example shown in Figure 3, they begin to gain fluency and appreciation of the power of symbols. The pen down function makes forward and turn visible to the user; they can visualize this familiar event in their world. The discovery process first involves the use of some known reality, moving into direct manipulation that transforms the learner's perception into meaning, and lastly moving into the symbolic mode of representation. Bruner's work also suggests that even very young learners can learn complex ideas as long as the instruction is organized appropriately. (Bruner, 1966) The repetition of visual manipulations, like drive a car scripts, over time, lead into the development of abstract ideas. Alan Kay expanded on Bruner's ideas to refer to images in the world representing and modelling the internal thoughtful, or the reflective symbols of the mind. (Kay, Google video) The

Etoys user interface gives young children the appropriate instructional environment to explore, build, and eventually reflect on their constructions and powerful mathematics ideas.

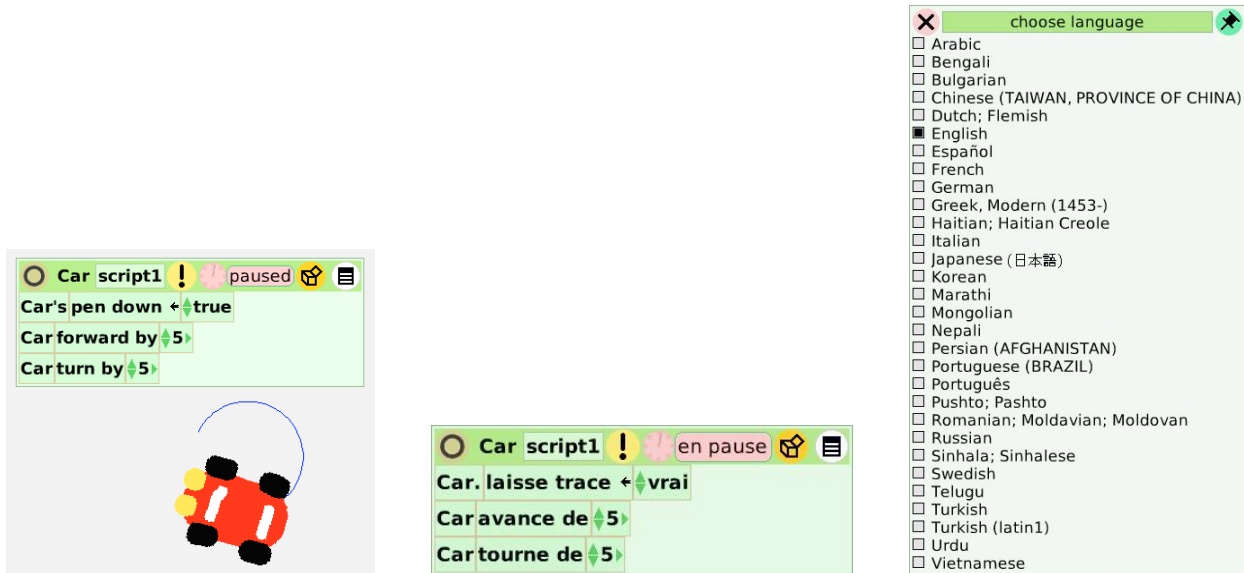


Figure 3. A script for moving the car with pen down and the same script when language is changed to French. A list of available languages in the current version of Etoys.

Etoys is a “live” system, allowing the user access to objects and scripts and possibilities to make new objects or script old or new objects while the system is running other scripts. This immediate and always available live system, makes variables and feedback more accessible to the user. Creating robotic car scripts that drive themselves on a path uses feedback. The powerful idea of feedback to an object is scripted using conditional statements, if-then statements, requiring the user to think again in new ways about their world.

Collaboration between Objects

While creating one object and scripting that same object is easily accessed, creating objects that interact with each other requires the user to think about more than one viewer while scripting. While a script belongs to one object, you can easily add tiles from other objects into each script. The user will need to understand how the movement of one object is related to the scripting in another object. In the Figure 4, you see a car and a steering wheel. The script says, that the heading of the car should result from the heading of the steering wheel. When you set the script “ticking” and use the blue halo of the wheel to rotate it, the car will also rotate. By exploring and playing with these objects and their interaction, users can discover how the feedback from the steering wheel’s heading can steer the car. This activity again requires the user to apply their understanding of the real world through symbolic representations in the Etoys world. Numeric feedback, in the form of positive and negative numbers, is used to drive the car. Through continued exploration, users gain fluency in driving by manipulating the numbers to drive straight, or to the left or right, as desired. The computer representation is an extension and symbolic representation. Fluency in the manipulation of the car is a kinaesthetic manipulation of variables, feedback, positive and negative numbers, and the beginning of modelling, and reflecting on, these powerful ideas through computer play. (Kay, Google video)

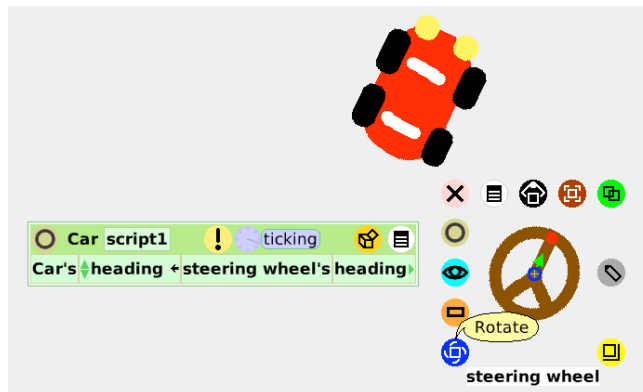


Figure 4. Car's heading is retrieved from steering wheel's heading

This is a simple example of collaboration; you can of course build scripts that are more complex and create simulations and games. Using a tile of another object instead of a number also introduces the concept of a variable: that a name stands for a number.

Animations

Animations are quickly and easily made. The user paints several “costume changes” for an object and drags tiles to a script to move the cursor and change the costume. Movies and videos in Etoys are just an implementation of the animated objects. Frames from a movie are dropped in a holder, the user drags the tiles for the script, and the movie will play! The Squeak Etoys book has pages that are just different costumes on the Etoys object! Books and power-point like presentations are just animations of objects in the system.

Extending Etoys: Kedama

Kedama is a system within Etoys that was inspired by work on StarLogo. StarLOGO is a LOGO dialect for the modeling of massive parallel agent based simulations. It allows graphical programming and inspired the development of Kedama. The actual Etoys version of Kedama is called Particles and is the environment for massive parallel simulation within Etoys, which can be scripted in the same way as other Etoys objects.

Learning and Teaching

Constructionism and Constructivism

Etoys provides an environment for Piaget's constructivist and Papert's constructionist learning. Teaching for constructivist learning requires the teacher to view the learner as unique, and take into account the culture and background of experiences of the learner. The responsibility of the learning is the learners, although the motivation for learning is reflected in the student's confidence in their learning. Vygotsky's ZPD (zone of proximal development) is the area between the things the learner can do by themselves and the things they can do only with the aid of a peer, mentor or teacher. The larger the child's ZPD, the more they will be able to do. Constructionist learning further requires that students are actively engaged in some kind of product construction, such a LOGO project, or an Etoys project.

Learning Etoys

Earlier Etoys versions started with a blank screen and offered the same power as a blank page of paper and a pen – but also the same challenges. Now, Etoys starts with a pre-loaded project, showing an open script and a moving car, and links to tutorials and example projects, so users can start exploring existing projects. Users that have not had experience in open constructionist learning environments may have initial difficulty getting started. Learning through play and

exploration is part of the Etoys experience and users can be encouraged to spend some time just exploring. However, complete freedom to discover doesn't give users the competence that creates the greater ZPD and confidence to pursue project construction.

Etoys provides scaffolds

Using the drive a car curriculum, users learn the basics of objects, scripting, turtle graphics, pen down visualizations, and interacting objects while learning to "pay attention" to some of the powerful ideas they are exploring. The drive a car curriculum is available to users in informal settings, in Etoys tutorials on the Squeakland website (www.squeakland.org), and can be used for beginning Etoys in a formal classroom environment. Etoys provides opportunities for Papert's "hard fun", but attention to scaffolding support can help build the user threshold for dealing with difficult problems, thinking through and exploring possible solutions, and generally building tenacity and resilience in thinking.

The user interface is "user friendly" in that it allows the user to access the program and the computer at all times. This provides support for the neophyte user in gaining competence. The user interface is representative of the system itself; the tiles that you drag and drop are visual representations that look like phrases or sentences and clearly show that the objects are getting the "messages" that instruct their behaviors.

Teachers in learner-centered classrooms can introduce Etoys by observing their students individually and in groups, and providing the "next step", or what is sometimes called "just in time" learning. Teachers in this setting should be ready to support learning through expert peers, expert aids or parents, or they provide the next step to give students in their Etoys experiences. (Galas,1999). Usually, students will ask for their next step because of their own deep engagement and motivation to continue to construct a project. Scaffolding learning and support for building projects is imperative, as different users have different thresholds (ZPD) for feeling competence and success. Extending the ZPD of users by scaffolding supports to students can be done with expert peer assistance, expert teacher assistance, apprenticeship learning set-ups, or encouraging and teaching to the use of the balloon help and the tutorials in the Squeak Etoys systems.

Squeak has no ceiling

Squeak Etoys is a powerful educational tool that can extend the range of user creations and the powerful ideas that users can explore. The underlying programming of Squeak Etoys is Squeak, a powerful, full-featured, object-oriented programming environment.

Etoys is developed in Squeak, an open-source Smalltalk implementation. This is immediately obvious for beginning users, who would be overwhelmed with an expert-level programming environment, but this is Squeak and Etoys. You can go beneath the Etoys interface, or as Kay and Ingalls say, "look under the hood", and access the underlying system using a special key. You then have immediate access to the full power of the Squeak programming language.

In Figure 5, you can see the similarity of the two representations of the script. There is an option to show the text code version of a script within Etoys, but to be able to write textual code, you need to actually open the system browser.

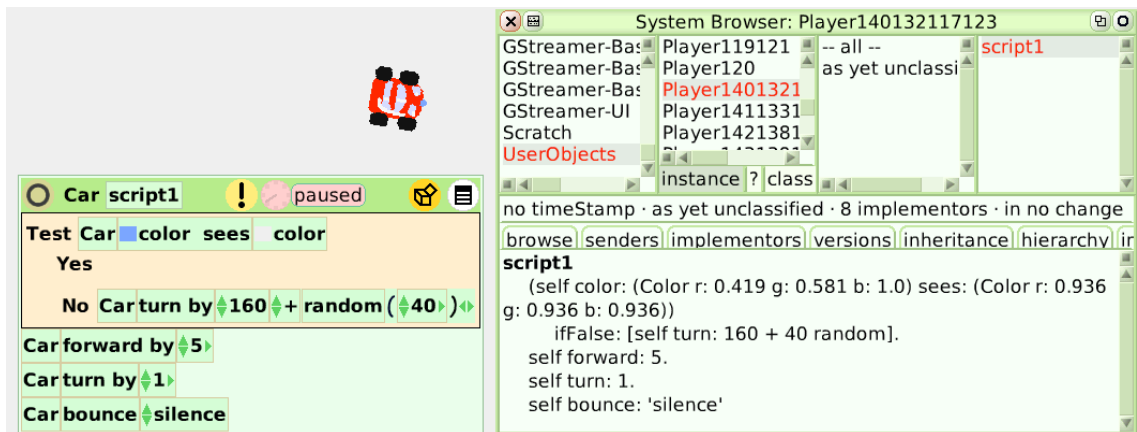


Figure 5. Tile version of Car's script and text code of the same script in the system browser

Construction and De-Construction

In Etoys, users can construct their own projects, and can explore and deconstruct projects from others to learn how these are built. You may need to view several individual scripts that are part of larger projects, and discover how the scripts work together. This ability is not only a powerful possibility of the system, but it also lets users know there is no “magic” in the computer; users can find out how everything works and work to understand or recreate it.

Another Etoys project use is exploration of prepared educational projects. There are projects built by educators or parents to teach specific concepts to children. These projects include the main idea, a simulation of a phenomenon, a description of the content, labels, explanations,, and, text fields to change parameters,. There are readily available objects in the Etoys supplies bin and object catalogue to enrich projects created by teachers or parents for these purposes.

In directed teaching environments, teachers can provide a “prepared environment” for students to experiment and explore specific topics, such as measurement (Pyramid Challenge, NASA Connect). Etoys provides teachers an easy-to-use tool for preparing lessons, thereby familiarizing teachers with Etoys use and possibilities. For students, these teacher-made projects are a resource to explore and expand their learning. (R. Freudenberg, 2009).

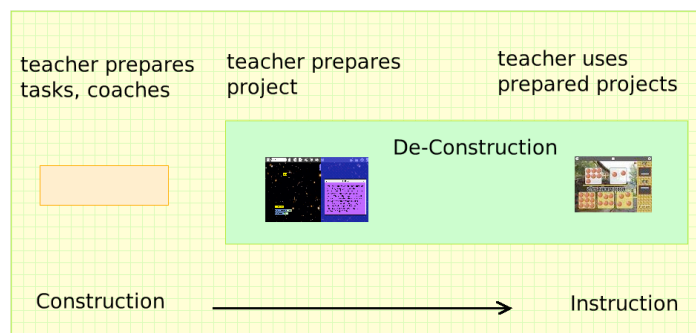


Figure 4. Different ways to use Etoys in the classroom

Conclusion

Squeak Etoys is a powerful constructionist tool that allows students on multiple platforms to publish and share projects and explore the world of powerful ideas by building and manipulating interactive simulations as active, not passive learners. Learning through play and exploration in the open world is an important part of the Etoys experience. The user interface is designed to openly allow the user access to play with powerful ideas in mathematics and science. Etoys is

an educational tool for use in both classroom and informal learning environments. Opportunistic teachers can utilize learning opportunities as they occur in student exploration, using “just in time” learning and teaching practices. These teachers can question, help students notice powerful connections, mediate classroom discourse, and nurture the development of deep thinking about powerful ideas (Galas, 2001).

Peer projects and teacher projects or a users saved projects, are all available from the Squeakland website Showcase, and offer opportunities for deconstructions, as children can open the viewer and see how the project is made, make their own changes to objects or scripting, and build upon others’ or their own work. Squeak, the underlying system in Etoys, is available for users wishing to move into programming modes involving greater expertise than the basic Etoys system. Squeak Etoys is a powerful system that can help young children learn mathematics and science powerful ideas that are not easily accessible in other learning environments.

References

Bruner, J., *Toward a Theory of Instruction*, Harvard University Press, 1966.

Freudenberg, B., Oshima, Y., Wallace, S., “Etoys for One Laptop Per Child”, Seventh Annual International Conference on Creating, Computing, Connecting, and Collaborating through Computing, Kyoto University, Kyoto, Japan, January 2009. (www.vpri.org/pdf/tr2009001_etoys4olpc.pdf)

Freudenberg, R., “Lernen mit Etoys”, Proceedings 13th GI-Fachtagung Informatik und Schule, Berlin, LNI P-156, Bonn, 2009.

Galas, C., “School Squeaking”, Squeak News, Volume 1, Issue 4, October 2001. (www.squeakland.org/resources/articles/article.jsp?id=1012)

Galas, C., “The Never Ending Story, Questioning Strategies for the Information Age”, Learning and Leading with Technology, ISTE, April 1999”, Learning and Leading with Technology, ISTE, April 1999.

Ingalls, D., Kaehler, T., Maloney, J., Wallace, S., Kay, A., “Back to the Future: the story of Squeak, a practical Smalltalk written in itself”, OOPSLA, Atlanta, Georgia, 1997. (<http://portal.acm.org/citation.cfm?id=263754>)

Kay, A., “Children Learn by Doing: Squeak Etoys on the OLPC XO”, VPRI, 2007. (www.vpri.org/pdf/rn2007006a_olpc.pdf)

Kay, A.C., “Squeak Etoys, Authoring and Media”, VPRI Research Note RN-2005-002 (www.vpri.org/pdf/rn2005001_learning.pdf)

Kay, A.C., “Doing with Images Makes Symbols: Communicating with Computers”, <http://video.google.com/videoplay?docid=-533537336174204822>, Apple Computer.

Papert, S., *Mindstorms: Children, Computers, and Powerful Ideas*, Basic Books, 1980.

Paper, S., “Hard Fun”, <http://www.papert.org/articles/HardFun.html>.

Shafer, D., *HyperTalk Programming*, Hayden Books, Indianapolis, 1988.