Alan Kay Squeakland posts

These are the messages posted by Alan Kay to the Squeakland mailing list between 2001 and 2012. This mailing list was intended for teachers and parents to discuss Etoys.

***Wed May 2, 2001**: Squeakland.org*

Squeakland.org

As all of you know, Squeakland.org is currently "under construction" and due to open "any week now". Our plans have been to have at least three sites, one for each of the authoring environments in Squeak that we've been working on. So:

* Squeakland.org is primarily for children, parents and teachers who use "etoys"

* SqueakOmni.org is for "Omniuser Squeakers" -- sort of from Hypercard to Lingo and beyond

* Squeak.org is for "Expert Level" Squeakers -- the bolts, nuts and guts of the system

The middle site and the middle authoring environment are quite a few months away from birth.

The first goals for Squeakland.org are to make sure that the plugin can be downloaded and run everywhere with as little difficulties as possible. We enlist your aid to help do these tests.

The mailing list -- squeakland at squeakland.org -- is hoping to attract people who are interested in elementary education and play and how computing might enhance them. Specifically, we are looking for enough day to day users of the site to create a forum for our next stages, which include a sample curriculum, and the next round of etoys. It would be nice to generate about10-50 emails a day about these issues. We at SqC plan to develop a trial curriculum this summer with several teachers that we've been working with, and we will do most correspondance using squeakland at squeakland.org.

Though an important part of this mailing list is to get bug reports, we plan to copy all technical emails to the regular Squeak mailing list. Squeakers, please don't scare off the parents and teachers -- after the children, they are our main intended users.

We will send out quite a bit more information about how to use the site as it gets closer to completion. For now, please try downloading the plugins and then try navigating around, both at the top HTML level of the site, and the entirely within Squeak levels below.

***Tue May 8, 2001**: Introducing children and/or school to Squeak*

John --

There is no reason (cognitive or otherwise) why a child of 9 or 10 shouldn't use Squeak on a computer.

There are many reasons why children of various ages shouldn't do "X" on a computer, but both the ages of the kids and the X's have to be taken into account.

There is quite a lot of parallelism between the desirable percentages of time spent learning from books at various ages with similar activities on computer. Basically, the younger the child, the more they should be messing about with the physical world. (Of course, most parents don't do a very good job of dealing with their children's physical world experiences either. For example, the kinds of toys that children play with in the physical world are quite important, but very little effort on the part of most parents goes into learning about desirable toys.)

But, even with young children, having them get familar with books and reading (especially via "lapware") is good for all. The same applies for thoughtful uses of computing.

Finally, though having anyone look at CRTs up close for any period of time is not terribly good for them (research supporting this was done by us at Xerox PARC in the mid70s), there is absolutely no harm incurred by having children look at the typical flat-screen XVGA screens found on current day laptops.

***Tue May 8, 2001**: Introducing children and/or school to Squeak*

John --

I just read the article in question that you mentioned below. It is really quite bogus and completely mixes up stuff that is more or less true with lots of stuff that is simply alarmist and most quite wrong. It's like blaming the printing press because of comic books or that Hitler wrote "Mein Kampfe". I could not find a shred of understanding about what children really do need to experience at an early age (it's neither hands off intellectual stuff, nor is it mini-university education).

In any case, it quite misses all of the important points about children and just about anything -- moreover, it could just as well be about books -- highly isolating (that's part of the point), "intellectual", etc. -- and musical instruments -- repitative stress injuries (you bet) -- rather than computing.

The biggest problem is that those holding these sentiments and those of the faction they oppose -- both are very large groups -- are both quite wrong about early childhood -- *and* the possible uses of computers.

I would not recommend this article to anyone (except as an egregious example of special pleading).

***Thu May 10, 2001**: Introducing children and/or school to Squeak*

Cathleen --

This is why this summer I'd like to collaboratively have us generate a sample curriculum for 4th through 6th (or possibly even through 8th) grade.

The way I have this in my imagination is to think of the deep knowledge ahead and to try to build the deep intuitions that are needed to understand that knowledge when it is encountered.

So, to me, I would like to have "math" be "real math", and (a) have the emphasis be on learning how to do mathematical reasoning on the one hand, and (b) for the kids to learn vectors and geometry as the main ways they have to think about numbers and arithmetic, and for them to learn how to use differential (tiny little) vectors that can be pasted together to make complex mathematical structures of many kinds.

Similarly, I'd like to have "science" be "real science", and to (a) have the emphasis be on learning the scientific ways to look at the world and also the limitations of trying to "know" that world, and (b) for them to make real contact with some of the deep scientific ideas that can be made completely understandable to them at various ages.

So my first pop at any curriculum design is always to think about these ideas and how they might be taught using the best pedagogy and most fruitful materials. The computer is just one of these, and it is best used for the parts of a curriculum where it is quite superior to physical media. We have a friend at the Exploratorium (Modesto Temez) who is a positive genius in organizing science learning just using easily obtainable junk in the outside world. This is where science learning has to start. The computer can be useful in motivating and being the instrument of the "mathematical music of science".

I will try to put out sketches of curriculum ideas for math and science on the squeakland list over the next few weeks to stimulate discussion.

A sketch at the computer part of the curriculum can be done by just organizing the etoys as "starters" that have progressions to more complex versions. For example, it's a good idea to do uniform motion before doing accellerated motion (and this obtains for all the different motions: in space, though images, audio samples, etc.).

In the accellerated motion examples we have experimented with,
the progression seems to be: model the dropping of a water balloon, then model shooting it (shoot the alien), then do the Lunar Lander game, then do the roller coaster. Then do Spacewar. Then do orbits of planets and spaceships. A progression like this might extend over more than one year of school, etc.

I think the tricky part of doing a math and science curriculum in elementary school that really looks ahead to the "deep content" of both these areas, is the amount and kind of teacher coaching that needs to be done to help elementary school teachers who may not have concentrated on math or science (in my experience, most have not).

***Wed Aug 22, 2001**: Successful use of in-class Active Essays*

That's great Mark!

"really natural" is what we are aiming for. All on this list should try their hands at active essays on ideas they particularly like ....

***Sat Nov 17, 2001**: Tamika Know Tutorial*

Hi Diego --

That's a good suggestion. I've been waiting until the college kids have made their project before I put my example on line (heh heh).

Also, I've been traveling extensively the last month so have been lagging in finishing the tutorial. I will try to get more done over the Thanksgiving holidays coming up. I'll probably put the example project online in December.

Meanwhile, you might just try to build it. Let me know if you have difficulties, because it will help the tutorial.

***Mon Dec 3, 2001**: I don't know*

Right now it is a "Squeak.org" thing only (not for squeakland.org folks). It was just a demo of antialiased fonts, and doesn't actually have an editor attached to it. But you can copy it using the green handle, and change the string that is displayed by clicking on the debug handle (the one with the little wrench) and choose "inspect". In the inspector that appears you can click on "string" in the left pane to see the string that is being put into TrueType. You can change this string in the right pane and "accept it" with a cmd-s. Then, I think if you move or adjust the size of the morph, the new string should appear.

Direct editing of TrueType fonts is very desirable and is "on a list" of things to do. Tansel Ersevas of SqueakNews uses these for all of his titles -- BTW very worthwhile looking at for ideas and examples of active essays -- and they are not too onerous for just that.

***Mon Dec 17, 2001**: Distances.*

How old is your son?

I did a very simple Newton's algorithm in the Etoys. This is a good project for 12-15 year olds. It is not very long and it shows them yet another way to think of feedback and searching and averaging.

***Mon Feb 25, 2002**: Hello. My name is Thom Gillespie. I write for a magazine called*

Thom --

One of the necessary parts of "literacy" is fluency. So it's not enough to read a little, or do math a little or program a little. There are important thresholds that have to be crossed. As with the older thresholds of reading and writing, most children haven't crossed the ones that would allow them to be literate.

The other consideration is that one can get fluent in lots of things that don't confer much benefit: television watching, videogames, pop culture, etc.

Taking both of these together, nothing really interesting has happened yet, but the technological parts of the new literacy are pretty close to being what is needed.

***Mon Mar 4, 2002**: Reply*

Thom --

At 12:39 AM -0500 3/3/02, Thom Kevin Gillespie wrote:
> From: Alan Kay <Alan.Kay at squeakland.org> > Subject: Reply

> Thom --

> One of the necessary parts of "literacy" is fluency. So it's not > enough to read a little, or do math a little or program a little. > There are important thresholds that have to be crossed. As with the > older thresholds of reading and writing, most children haven't > crossed the ones that would allow them to be literate.

Agreed.

> The other consideration is that one can get fluent in lots of things > that don't confer much benefit: television watching, videogames, pop > culture, etc.

Not sure if this is actually fluency in a language-like sense.

They aren't -- that was part of the point I was making.

Again the reading writing problem. I have a real interesting response from John S on the Squeak list, actually about 3 good responses from John and a good one from Michael Rosenblum from NYU who really blasts the illusion that watching TV makes anyone literate

It doesn't.

and the fact that we would never tolerate the lack of writing literacy in books that we tolerate in TV. Also a bunch of good stuff from Howard Gardner, a bunch of his grad students and Chris Crawford who as usual comes in so far from left field that he changes the game completely but in a very interesting way.

> Taking both of these together, nothing really interesting has > happened yet, but the technological parts of the new literacy are > pretty close to being what is needed.

The phrase I keep coming back to is 'mediajazz' Since this stuff shifts constantly and shows no sign of not shifting it makes a lot of sense to look at it in a jazz/improv frame and just add on the fact that it is media which is jazzing. Going back to John S I think you just shift the focus to the aesthetics and away from the differences among Squeak, html, Flash, Director, iShell, Blender, etc.

I think you are missing the difference between "productivity tools" whose main goal is to get something manifested, and "learning tools" whose main goal is for big important changes to happen in the learner's mind. Also, in the list that you gave, Squeak is the only real programming system and the only one that covers the range of what computers can do.

Big question seems to be that there are so few people equipped to deal with this combination of technologies and this combination of arts (2d, 3d, storytelling, video, animation, sound, music and flat out spacial design.)

The technologies are generally poorly done. That being said, there are fewer good drawers and painters out there than one would hope. There are fewer people who can play musical instruments *and* compose than one would hope. There are lots fewer who can do all four things mentioned above. There are an even smaller number that are fluent in math and science. And an even smaller number of those who are fluent in the arts. Since the first thresholds of fluency in most things is a 5-7 year process, we have to look to our own culture to wonder why people don't get fluent in more that a few things in a lifetime.

***Wed Jun 5, 2002**: Fill in the Blank*

I'm curious as to your prefered choice. Also you might try to fill in this blank first:

"Music is not in the piano" as "(blank) is not in the book"

You could also ask yourself what is (are) the special thing(s) about humans that the computer might be great at amplifying.

I'm terrible at titles ... but good luck and let me know of your next set of ideas about this ....

***Wed Jun 5, 2002**: Fill in the Blank*

By the way, one of the ways that I characterized the Dynabook years ago, was:

"An instrument whose music is ideas"

***Sun Jun 9, 2002**: Fill in the Blank*

... like "Idea" ...

***Wed Jul 3, 2002**: Are there any folks on the Squeak list who are either ...*

Thom --

I don't know of any, but it would be a very interesting collection of venues to investigate.

Currently, we think that most people need a fairly detailed guide about projects, the system, the significance, etc. There are interesting real-estate limitations and other distractions from today's screens that make creating a separate "good old paper" book about this to be a worthwhile project. We're in the process of doing this over the summer, and will start sharing drafts of several documents in a few months. In any case, we will also supply this material as active essays online.

Given our pretty successful school year just completed, I think the biggest gap to be closed right now is to give more people more of the information and context they need to start trying this stuff for themselves.

P.S. We are also very interested in home-schoolers, who also use libraries and online a lot ....

Wed Jul 3, 2002*: Are there any folks on the Squeak list who are either ...*

Thanks Jimmie --

I think making materials that could be used by home schoolers to teach math, science, computing, etc., using Squeak is a very good next step for all. We are trying to create a (much smaller than Squeak.org) open source list of home-schoolers who would be interested in helping make and package Squeak materials. Sounds like you might be the first of this list!

Wed Jul 3, 2002*: Laptops, Learners and Powerful Ideas Conference*

Well, I got started on this path by meeting Seymour in 1968 and many of my first insights into this area came from Seymour. We've been colleagues and friends now for more than 30 years.

Wed Jul 3, 2002*: Factors*

Edwin --

You might also be interested in the Reggio Emila schools and some of their books about learning by authoring at a very young age. One has the title: "The Hundred Languages of Children".

I don't think adults who have never programmed are challenged in the least by OOP. But the first paradigm that one learns seems to have quite a lasting effect these days. It was easier in the early sixties when I learned because there were no orthodox machine or language architectures, and one had to learn at least 20 or so. This helped quite a bit when a new idea came along .... By the end of the sixties, all had changed, and data structures and procedures had quite taken over.

Wed Aug 21, 2002*: Book for gifted 12 year old.*

The first question I'd ask is "what is she interested in?". It's almost always best to come up with projects that have some interesection with a learner's interests.

Wed Aug 21, 2002*: Link: Education in the Digital Age*

Sounds great Dave!

(It's also nice to hear from you!)

Wed Sep 4, 2002*: Helping my nephew with collision detection*

I don't think so, but it is possible to overlay embedded transparent patches (not really recommended for children's programming).

Wed Sep 25, 2002*: Is it alive?*

Yes. Kim, Roxanne Maloney and I are in Japan where the city of Kyoto is planning on putting versions of the "etoy curriculum" in 6 schools in the Spring (2 each of elementary, middle and high school).

Sat Nov 16, 2002*: Squeek for O SX?*

Joel --

Squeak creates its own world and does not use any of the host interfaces (this is one of the reasons it runs bit-identically on more than 25 platforms).

Sun Nov 17, 2002*: Squeek for O SX?*

However, you can use one of the UI skin packages to do a complete imitation of Aqua or Luna, etc. Jim Benson's Zurgle is excellent.

Mon Jan 13, 2003*: Squeak ideas for a classroom/clubhouse*

Hi Jahanzeb --

I'm going to write two replies to your very nice and interesting letter. This is the first, and I'll try to follow up with a more thoughtful one in a few days.

At 7:20 AM -0800 1/11/03, Jahanzeb Sherwani wrote:
Hello all,

I'm a research associate at LUMS University, Pakistan, and am working on a research project here where we're trying to use low threshold/high ceiling software environments to enable schoolkids to create and learn, and have fun while doing so. I'm interested in giving the fifteen kids (aged 7 thru 14) the flexibility to create whatever it is they want to create: graphics, music, games, stories, building their houses or other aspects of reality -- such that their learning is made personally meaningful (along the lines of Papert's Constructionism) in ways that are rarely found in education in the Third World. However, we'd also like to give them challenges that will bring out the most learning (such as learning about feedback through the cars, as shown in the elementary school gallery by BJ). Our main reason for choosing to do such a project is to enable the kids to think outside the hold of a curriculum that holds little to no relevance to their daily lives or the world around them, and to help them do stuff that's educational, fun, and personally relevant.

I was initially using Alice2 as the major environment to work in, although I'm running into quite a few problems (namely, it's not running on the school's PCs for some reason!) so I've been looking *at other options, and discovered Squeak (which many at the MIT Media Lab recommended when I visited a month ago), and I believe it perfectly fits the bill.*

I managed to go over parts of the mailing list archives before asking these questions, so I hope you'll forgive me if they were already addressed before. My questions are:

1) How does one act as a facilitator in such an open setting with Squeak, so as to allow diverse views of what each wants to do, but still make sure that there is some learning (and not just air guitaring) going on? There is so much you can do with Squeak (no ceiling), but how does one try to nudge it along lines that will lead to good learning -- or is that a contradiction in terms?

What we try to do is inspired by Montessori: to come up with projects that the kids absolutely treat as toys and play, that also have (we think) beneficial cognitive side effects. So all the stuff about cars and driving, the car races, etc., teach about vectors, velocity, accelleration, the idea of "random", the idea of feedback (to make a robot car that can stay on the road), etc. Our experience over the last 3 years has been that virtually every child in a classroom gets really interested in this stuff for their own reasons. Recent experiences in Japan indicate that these projects and the process are pretty independent of simple cultural distinctions.

Another area that children love is nature, especially regarding animals. And it is a source of delight to them to find out that they can make a new costume for their car and turn it into a fish or horse, etc.

Last year we tried a more ambitious project in science. This involved having the children (10 years old) first learn about velocity and accelleration using their cars. This is a very nice project all by itself. Then we had them do some measuring (for example, a bike tire circumference) using different tools. This helped them understand that measuring is likely to not produce the exact same numbers, but is likely to produce numbers similar in magnitude.

Then we showed them various objects (two shotputs of different weights, a croquet ball, a foam ball, some apples, etc.) and got them to speculate as to which would fall faster or slower. Then we took them outside and dropped the objects from the roof of the school. We took videos of these drops. The videos were imported into Squeak and the kids could look at every 5th frame and measure what a dropped object was doing. They could see that the pattern was the same one they had seen when they were doing accelleration with their cars. This led them to write similar scripts to accellerate a painted object to match the movie. (We have a nice video clip of this process.)

Thus they were able to experience a phenomenon of the real world, measure it, put a model to it, and make a mathematical simulation of it. (Most American college students are not at all successful at learning this using standard methods.)

Once they had a script that would move objects as gravity moves them, they had a new tool and toy to make gravity games, etc.

2) How does one introduce the medium as something that is infinitelly malleable, and that it is ok to add/change something if you don't like it?

Most children have a big revelation about this in their first few hours of doing stuff. We've noticed it happening many times when they put a new costume to their car object and realize that they can make anything and make it do anything.

For instance, the lack of a 'move sideways' tile (like the 'forward by') tile means that kids will have to start off controlling their game-characters with a move forward/backward, turn left/right instruction set, and so can't start off by making a Pac-Man type game (which needs to move left/right, and not turn). Should I create a 'move sideways' tile beforehand

I would suggest not.

, or try to help them make it themselves as they require it?

There are two approaches here and both are worthwhile in the end. The first is that using "object forward by" and "object turn by 90" will do what you want. The second is that there are x and y location properties in the viewer of all objects. It is very worthwhile for the children to see that:

```
       object's x increase by 10
```

will move the object 10 pixels to the right, and that this is exactly equivalent to

```
       object forward by 10
```

if the object is pointed to the right.

Similarly,

```
      object turn 5
```

is exactly equivalent to

```
     object's heading increase by 5
```

3) The main reason I pushed to have classes of diverse ages was that the young ones will be able to learn from what the elder kids are doing, and will also get a sense of self-respect by working on the same environment as elder kids. Should we be giving different problems to the younger ones to solve, or not? Could anyone on the mailing list (who has experience with such age ranges in such classes) tell me about their own experiences, if possible?

Teach the older ones a few days before the younger ones. You can start with the same set of projects, but the older children can go quite a bit further and faster. So it's a good idea to have more project ideas for them.

4) The social relevance of education is something that is touched upon most by Paulo Freire, who said that imported curricula aren't adequate because they lack relevance to the sociocultural environment, particularly in developing countries.

This is an interesting claim. I think it is true at one level, but it rapidly misses the point once education starts to happen (and this is the great difference between "education" and "training"). Perhaps a milder view is that in any kind of user interface experience, the designer has to start in the world that the endusers live in.

However, the learning of powerful ideas is not just a new tool that one wears on one's belt, but an actual change in how the world (especially of ideas) is perceived. It's a change of perspective as well as one of knowledge. A child who learns science starts to become part of a different cultural environment, and this is why scientists quite resemble each other and can easily communicate with each other all over the world regardless of their initial background.

This isn't the same thing as impressing "Western Civ" on other cultures -- it's partly an accident that science was invented in Northern Europe (it could just have easily have happened one or two thousand years earlier in the Mediterranean or in China or Japan).

The simple bottom line here is that young children especially are interested in things they can *do*. So you will have no problems. Freire was talking much more about trying to educate adults who had grown up in traditional cultures (and here, I think, he was most right).

Does anyone have any experience in addressing these concerns best through the use of environments such as Squeak?

I think Mitchel Resnick of the Media Lab has had more experience than we have, with his various LOGO in schools projects in Latin America, and his more recent computer clubhouses around the world.

Our twice-weekly sessions with the kids in their school labs begin this Tuesday, so whatever you could tell me before then would be extremely helpful.

I hope this helps. Also Kim Rose has had the many experiences teaching Squeak etoys to children and teachers around the world. She can provide quite a bit of guidance as well.

Wed Jan 15, 2003*: Squeak ideas for a classroom/clubhouse*

Hi Jahanzeb --

At 7:58 AM -0800 1/15/03, Jahanzeb Sherwani wrote:
*>The rule is: If you drag a tile out of the viewer, it *will* create a*
*>new script. If you drag it out of an existing script it will *not**
>create a new script. I know this is slightly inconsistent but it
seemed >to be the best solution considering all the other tradeoffs.

Yes, this makes sense... but the problem I was having was that dragging a tile out of the viewer was creating a script half the time, and not creating a script the other half. We (sheepishly) were running the plugin in a browser, and thus had not updated it -- could this have caused the problem?

I don't think so. I think all this is the result of a few decisions we made several years ago that made some sense at the time, but perhaps don't at all now. For example, when I first demoed etoys to Disney in '97 or so, I wanted it to look like magic, and having the tiles turn into a script when tossed on the desktop did just that. But this created a huge violation of good UI design because it produced a conflict that should never be there: at odds with how to show the scripts made by the kids. We just decided to have them open up on the desktop rather than create a new script for the tiles to go into. Thus the conflict with making a new script.

The other problem that needs to be solved and fixed is when to do something when tiles are dropped. I think what is happening to you is that you might be dropping tiles somewhere other than the desktop and the system is not reacting to this. At an earlier point in etoys all drops did the same thing and this was changed (I can't remember just why). Another example is that playfields are now made sticky and can't be picked up directly with the mouse because children would often miss when trying to touch something in the playfield and would pick it up instead. However, this means that playfields can only be moved with the black and brown handles, and this is yet another rule to be learned (amongst too many rules already).

I think the right thing to do with regard to the tile drag and drop cases is to have a better way to create a fresh script and then make all tiles open up when tossed on the desktop. For example, we could have a new handle in the halo make a script. This doesn't seem like a good idea because there are too many handles in the halo already. A better idea might be to put a new script button at the top of a player's viewer so that it is really easy to make one whenever the viewer is open. There is one of these inside the top of the viewer menu, but this seems too hidden to me. What do all think about this?

Also, I think while we're at it, we should also expose the "make a new variable" button that is also hidden in the top of the viewer menu. ??

Fortunately, our experience over the last few years has been that the children very quickly learn what to do in the UI, even with inconsistencies in the UI (I think they all have built-in talents for this since the real world of human cultures also has quite a few inconsistencies(!)). However, my belief is that the only difficulties that should be in a child's learning environment are ones that all of us have put there to help the child -- there shouldn't be any gratuitous difficulties from bad design ....

Wed Jan 22, 2003*: Etoy issues*

How old are these children? 10 year olds on up have no trouble with "x" and "y" (and that is a good convention for them to learn in math).

Pull out the

```
   car's x <- 5
```

tiles. Click on the caret in the "<-" to turn it to "increase by" so you get:

```
  car's x increase by 5
```

This is the same as

```
    car's heading <- 90
    car forward by 5
```

"Increase by" is a very powerful idea in mathematics and this is an excellent way for 10 year olds and up to start to learn it.

Don't give up on the running the maze problem. Do a few simpler things first. Get the kids to try to go around the inside of the room with a blindfold on by feel. You can usually help them discover that a good way to do this is (in English) something like:

```
  student move around room          ticking

   Forward by a little
   Test wall touch
     Yes turn a little away from wall
      No  turn a little towards the wall
```

This is a nice feedback program. Draw a road (say <brown>)with the paint system. Draw a dot of some color (say <blue>) on the front of the car. Try:

```
            car followroad

           forward by 5
           Test car's color <blue> sees <brown>          You can pick
these colors from your paintings
              Yes car turn -5
               No car turn 5
```

This will make the car move about the outside of the road. I think there are some examples of projects kids did branching out with this idea on the squeakland.org site.

A good next project is to figure out ways to make a car go down the middle of a road. Children have come up with many solutions to this that include different colored stripes, more than one sensor (sometimes they use the two headlights, etc.)

Once this is done, then one only has to realize that a maze is a kind of road ...

Then there are some very nice ways of thinking about how to get out of an arbitrary maze. A good start to this is first just make a script (or two) that will move the car through a maze and around corners. Then think about trying to explore the maze with the car.

Here's another way to think about "bounce". Ask yourself about what is the change of angle that happens when your player senses an obstacle. Do it for horizontal and vertical lines first....

Sun Jan 26, 2003*: Etoy issues -- Keyboard Control of etoys*

You could also try a joystick player. Look in the supplies bin to see what they do.

Thu Feb 6, 2003*: A Question about Croquet's Philosophy on Multi-user 3D Environments...*

Hi Darius --

This email list is for parents, teachers and children who are concerned with the "etoys" part of Squeak. Croquet stuff can and should be discussed both on its own list and on the squeak.org list.

To answer your question: remember what has happened to the "Victoria's Secret" website on the occasion of special promotions they've done, especially connected with TV. At some point capacity gets exceeded. So there is nothing new here. The first practical limit in Croquet is in the number of polygons that can be displayed by one's own 3D accellerator. This limits both the scene complexity and the number of people who can be in view. This is why Everquest, even with its farms of servers, trys to spread visitors out over the world so there are never more than a few in view at any given time.

Tue Feb 11, 2003*: Partial List of Etoy Projects*

The list below is not meant to tantalize unfairly (because we haven't yet put out hints and project books for most of them). The reason I put out the list here is to encourage people who have gotten interested in etoys to look further before trying to find out what Squeak can do outside of etoys. (But for those who do want to find out and are well versed in computing, please visit http://www.squeak.org).

Our impression is that most folks who have started in etoys have tended to stay with projects that are like the examples given on the website and have not gone much beyond those examples. This is just to point out that there are lots of really interesting mathematical, scientific and theatrical projects for which etoys are a pretty neat authoring environment.Partial List of 2D Etoy Projects for grade 5-8 ------------------------------------

Many Illusions for various ages Drive a Car Collaborative Car Races Sensing the Road Robot Car Robot Car Races Change of Position = Speed Random and its uses Change of Speed = Accelleration Simple Animations & Movies Hairdos and Faces Bouncing Mazes Sound Synthesis MPEG, MP3, MIDI, etc. A Dixieland Band Gravity and Objects Off a Cliff Water Balloon Cannon Lunar Lander Roller Coasters Vectors POVs Spaceships Spacewar Orbits Springs Weighing Gradient following - Salmon and Clownfish Tree Growing Epidemics Multiple Mentalities Grey Walter Conditioned Response Learning Circuit Models DTP Books Presentations Collaborations

--

Wed Feb 12, 2003*: problem from a beginner*

Hi Mark --

You need to return the value of answer.

You can do this by adding

```
    ^ answer.
```

at the end of the method.

(Don't forget to put a period between the "]" and before this line.)

Having said this, let me refer you to http://www.squeak.org. This site is for people who program in Squeak proper.

This http://squeakland.org site is for teachers, parents, and others who are working with the etoys part of Squeak for children.

Fri Feb 14, 2003*: Partial List of Etoy Projects*

Do you have a good algorithm or strategy for TicTacToe? Let me know and I'll see what I can do. There are some very interesting distributed system approaches that might be good here -- somewhat similar to the biological tree growing schemes in etoys that don't require recursion, but are "recursive" none the less.

***Wed Feb 19, 2003**: Repeat?*

Hi Phil --

At 11:00 AM -0500 2/19/03, Phil Firsenbaum wrote:
Thanks much for the help with repeating a script...I think I've got the idea now. Any script can be repeated by using a conditional (test) statement (if that's the right nomenclature).

Not exactly. Any script can be repeated (period). You can get it to repeat by clicking on the clock (on the script, in the viewer, or on the go, step, stop button found in "Widgets"), or by holding down where it says "normal" and choosing "ticking", or by sending it a message - to start up script "car foo":
car start script foo
These tiles are found under the category "scripting".

Looking at your picture, I see that the scripts "startRepeat" and "setup1" are both paused, which means they must once have been ticking. But neither of these scripts should be looped. They are both designed to be run once, they should be set to "normal".

So, here's my project thus far. Activating the first script moves the object (Roam1) based on the value of the slider (light blue rectangle). Now, I'd like to be able to show that value without using the object's Viewer.

Go into the viewer and look at the menu associated with the slider's "numericValue" property. The menu includes choices for a "simple watcher" and a "fancy watcher". The first just gives you a number that will reflect the the value of the property, the second will give you a UI for this value that includes a label and the ability to set the value from the UI (this is the option that is usually most useful for children).

My initial goal with this project (in addition to learning how to create it for myself) is to have 1st and 2nd graders work with estimating distance and number values...they will set the value of the slider as they attempt to get the object to land in the box at the top of the track.

forward 30 is too long a distance -- it will be about 1/3" on some screens.

I've also been trying to slow down the movement of the object so that kids can see more clearly what is happening. Haven't figured that one out yet either.

Press down on the clock in a script and hold it until a menu appears. This will allow you to set how many ticks per second the script will execute. (This can also be programmed by hand using a variable to hold a delay count.)

Another way to slow it down and get some precision would be to just do "Roam1 forward 1".

I think there might be a somewhat richer approach to this project, that allows the children to do some scripting rather than just use what the teacher has provided. Let me know if you are interested in exploring this.

***Thu Feb 20, 2003**: Repeat?*

This allows the pages in a book to be of a difference size. However, there is also a menu command on the book to make all the pages in the book the size of the page that is currently showing. I would advise looking at the balloon help in the book UI (at the top), to click on the "more options" button (diamond shaped to the right) and to look at the main menu (the big dot in the center).

If you want to have separate environments for etoys on each page, these can be obtained from the "widgets" flap and have the label "scripting".

***Fri Feb 21, 2003**: Repeat?*

Hi Phil --

At 9:22 AM -0500 2/21/03, Phil Firsenbaum wrote:
Kim, Once again your advice is invaluable, however, I also see that I could be using Stack/cards instead of Book/pages. What's the difference?

The Stack stuff is quite experimental, you are better off with Books.

I'm inclined to use the Stack since I used to work in and teach HyperCard.

Except, that the Stacks here are not enough like Hypercard to help you.

Alan has referred to the "widgets" flap several times. Is that synonymous with the Supplies flap that I see?

It's another flap that is hidden in the children's version that you are using. However, try typing a cmd-o (alt-o on a PC). This should make a "useful-object" palette visible. Click on "scripting" and you will see the object in question (called "scripting") on the right side of the second row. Drag it out to get one.

***Sat Mar 1, 2003**: Oops re the reminder for the Kay/Papert talks....*

BTW, Sheine deserves *great applause* for being such a positive force for advancing Squeak for children, and especially for doing an enormous amount of work to set up our talks and bring Seymour and me to Toronto. I was especially impressed with the atmosphere, children and teachers I met at the Don Mills School, especially Sebastian and Donna. This was great fun!

Best wishes to all,

Alan

***Sat Mar 1, 2003**: turtle trails*

This used to work. And still should be working .... Scott? Mike?

***Tue Mar 11, 2003**: I want to document but I need to learn first!*

Jerry --

I think you should first separate out the Squeak system -- an experimental version of Smalltalk that is quite beyond the scope of this list, which is for parents and teachers -- from the Squeak "Etoys" which is aimed at children and *is* discussed on this list. So complaining about 2000 posts to SqueakDev on this list is just confusing for most of the folks here -- it's like complaining that LISP is big and comprehensive -- it's not an enduser system, etc.

I will confine myself to the tradeoffs with the Squeak etoys. First, we really do need better documentation, even for a system that is still being tested by us. We have found that it takes about 3 years in a classroom to get a good set of tests and we are just now in that 3rd year. The results of these 3 years have been written up by teacher BJ Allen-Conn and Kim Rose in a "book of 10 projects" -- they have done a great job! -- and drafts of this book will be available online not too far in the future. Another terrific contribution is from Sebastian Hergott's 8th grade class in Toronto. They did lots of projects and he got them to write them up as documented examples. These two books together supply lots of examples and should help to bridge some of the gaps in documentation.

However, I should say a little about the history of etoys. They were originally not aimed at classrooms but as 10-20 minute projects supplied on the web for parents and their children to do together. I stripped out as many features as I could and tried to come up with a system that could do "100 examples" pretty straightforwardly. The documentation that was intended here was to have been to teach parents how to do the examples so they and their kids could have a good experience. For several reasons, this plan did not work out at Disney. But BJ saw it and wanted to try etoys in her 5th grade classroom. I was initially against the idea because I thought that etoys were not complete enough for that venue. But she and Kim Rose decided to do it anyway. Six weeks later they started to show me some really good results, and I realized that it would be worth doing a 3 year experiment to see how well the etoys -- even with some of their lacks -- would work out with 10 and 11 year olds.

The results have been excellent -- in the proper environment most children have no trouble getting joyously creative and fluent -- and hence the forthcoming book by BJ and Kim to help other teachers and parents achieve the same results.

Our previous plans to make a kind of "superhypercard" and then get version 2 of etoys from that much more comprehensive design did not work out at Disney, and it wasn't until recently that we've been able to get that plan going again. I think this is more like the system you want, and you'll have a chance to try it out this summer.

To zero in on a real critique of today's etoys, it is helpful to confine discussion to 10 year olds and up, since essentially all the experience that we and others have had are in this age range. The etoys have changed very little in several years, in part because of the testing that is going on, so comments such as "too fast moving" really have to do with the larger Squeak community over at www.squeak.org. Here I think the problems are not so much lack of documentation as lack of particular kinds of documentation, such as detailed tutorials and project workbooks. The user-tested books mentioned above should help this.

Let me turn to another area, and tell a story that I witnessed recently. I was visiting a classroom with a really terrific teacher, who was truly ecstatic when his children could figure out something before him (we need more of these kinds of teachers!). But he brought up a problem that he couldn't see how to do. He wanted to general random colors, and had seen that the red, green and blue blends are given in the color picker. In etoys colors are not manifested as three numbers (we possibly should, but don't) though they are in the larger Squeak system (and in many other ways). So he didn't see how to make up colors, especially random ones. My thought was to put a bunch of objects (such as ellipses) into a holder, give them different colors and then do random picking by moving the cursor

holder's cursor <- random

to get an object whose color can be gotten at.

We did that and he was happy. But then we saw a child who came
up with a much better way to do this. He just put splotches of paint on the desktop and ran a Squeak player (like a car) over the splotches in a random "drunkard's walk" and used "color under" to pick up the color as a value.

My thought on seeing that was that it was the child who found the "etoys way" of solving this problem, and that the general solution in this fashion would involve using the color rainbow of a color picker to supply a wide range of colors for the car to wander about on.

My second thought was that both the teacher and I were somewhat
trapped in our pasts. The teacher had done something with color numbers in the past and wanted to do it again. I went to a table lookup solution that I had done many times in the past for other kinds of problems, and this worked. The child went at the heart of the matter with a completely simple and concrete approach that was quite brilliant and original.

One of the reasons I'm telling this story is that today's etoys -- that lack a wide and comprehensive range of features that "they should have" -- are best approached through the kinds of projects that *can* be done really nicely using the features that are there. There are more than enough such projects to occupy a full year (really more like 3 years) of work and play by children. As for the larger scope that is eventually needed, I'm hoping we can accomplish this by the time today's projects are used up.

Now to another one of your comments in yesterday's email. You wrote: At 6:44 PM -0800 3/10/03, Jerry Balzano wrote:

Get a load of these (the total "partial" list was almost 40 lines long):

Orbits Springs Weighing Gradient following - Salmon and Clownfish Tree Growing Epidemics Multiple Mentalities Grey Walter Conditioned Response Learning Circuit Models Anyone who could create projects like these in any programmable medium, I'd say, would have a serious leg up on "real" programming by anyone's hard-nosed definition of that elusive (and ever-changing) concept.

I think I agree here. I've done each of these strictly in etoys to see what the process is like and to understand how one would explain the process to both teachers and children. Most of these projects are aimed at older children (such as Sebastian's 8th graders and older), and I think are quite doable, but they haven't been tested yet with adults and children of a good age and mindset. Just to provide a few more comments on some of these:

Orbits is easily done in etoys if you understand Newton's inverse square law, vectors (and that each etoy player -- like a logo turtle -- is a vector and can do vector arithmetic). The script that does the work is about 4 lines of tiles long and is a pretty direct translation of the inverse square law using "increase by" of vectors. It's a very clean script.

Here quite a bit has to be worked up to for most teachers and other adults. There are hurdles of mathematics, science, and learning more about how to use etoys. The scaffolding would require many projects to be done earlier, including the accelleration and gravity projects that were easily done by BJ's 5th graders. I think a good next one is to do a spaceship floating in space without a gravity field to get a sense of how velocity is often (usually) in a different direction than the ship is pointing.

Springs are fun to do, and easy to script in etoys if you go through the exercise of deciding that the force on a spring is proportional to the displacement and in the opposite direction. I think there is quite a bit of scaffolding needed to do the science part.

Weighing is part of doing a real roller coaster in etoys. An insight is required here. Most people get stumped about needing sine and cosine, etc., to find the forces on an inclined plane. But in fact, you can "weigh" them using a postal scale on an inclined scale. You can make up a simple table -- using a holder -- of the forces every few degrees and this is quite good enough to make a real roller coaster in etoys.

Gradient following If you make a gradient using the graphic properties sheet you can do tests on it using "Brightness under". This allows a simple feedback program to be written (very much like the follow the road ones) that will cause a simulated object to follow and find the darker or lighter regions of the gradient. (Gradient following is a feature in starLogo, but I think people should learn about it by actually scripting it.)

Tree Growing Most people have cognitive difficulties with recursion, but one nice way to look at trees is recursively. This is a conflict. Because etoys can make new objects via copies (see below) it is possible to bypass recursion altogether in favor of a branching activation. This turned out to be a very clear script and a good model for other kinds of "recursion changed to branching activation" problems.

Epidemics have a wide range. The easiest ones are just having infected objects bump into noninfected ones and transmit the infection. This is just a few lines of script to do.

Multiple mentalities comes from the Vivarium work we did 15 years ago. Here we have separate scripts or even objects that represent parallel and mostly independent drives of the simulated animal. The main thinking that is needed is to figure out which of the drives should be allowed to control the animal. This is easy for two (a simple comparison) and needs something like a sort for more (it is actually just looking for the one with the largest "urgency", so it's a matter of using the "max" operator to perculate the largest urgency one in a holder.

Grey Walter conditioned reflex learning model. Here it is hard to guess about the appropriate age for this wonderful etoy. My guess is high school since Grey Walter's model is nicely subtle. (He did this with a single vacuum tube in 1949, so parsimony was the order of the day. He got all of his power from very careful reasoning and clear thinking about a simple model to do this.) Once you understand how he did it (I made a diagram to show the 7 steps you have to go through) it was quite easy to do in etoys and generated a nice set of dynamic graphs for the animal's "state of mind".

Circuit Models I've not quite figured out an appropriate approach here. One way is to use the connectors stuff of Ned Konz and propagate signals though his objects. Several folks have done this, most notably a high school student who is working with us -- he went to the heart of the matter and decided not to do batteries and bulbs per se but to see about simulating logic.

My students (same ones as above) wrote programs in NetLogo, Microworlds (a descendant of Logo),

This is a product

and Stagecast Creator

so is this. Etoys is an experimental system that is still quite a ways from being a finished packaged product.

, including a "Turtle Epidemic" model in NetLogo for which I wrote the tutorial (see http://ccl.northwestern.edu/netlogo/resources.shtml) and a "Food Fight" game in Stagecast Creator, for which I'd love to be able to write the "etoys tutorial", if I could only see how to do several simple things in Etoys, for example

* have an agent (smiley) create another agent (burger) in the space next to him

Let's suppose that smiley is in a playfield called "fastfood".

```
smiley create
    smiley's temp <- burger copy
    fastfood include smiley's temp
    smiley's temp's x <- smiley's x + 25
    smiley's temp's y <- smiley's y
```

I found "copy" and "include" just by going through the views of the two objects and seeing what the balloon help told me. This is the documentation that is there, but most people don't use it. I found that I could make a player valued variable by looking at the menu item "change data type", etc.

* have an agent (smiley) send a message to a counter agent (count down) each time he "uses up" a burger, and another message to a counter-scorer agent (count up) each time one of his burgers hits his opponent

```
burger scoring
  Test burger's color sees <color of boundary>
     Yes smiley's score decrease by 1
  Test burger's color sees <color of opponent>
     Yes smiley's score increase by 1
```

...just to name two.

So, speaking of "viable learning paths", does anyone have a suggestion for one for *me*? Who wants to respond to all the questions my teacher-students raised in my field notes?

I do.

Who wants to help me complete all the projects on Alan's list?

I have done these projects. I need help in explaining them in a way useful to parents and teachers.

*If *I* can't figure out how to do this stuff on my own, there's no way any of the teachers I teach -- even after they've been thoroughly Balzano-indoctrinated to the virtues of programming and completed my more-rigorous-than-99%-of-other-teacher-ed-computer-courses course -- will be able to figure it out either.*

I don't necessarily agree here, but your point is well taken. I think that quite a bit of success for different kinds of people is the match up between types of thinking, types of motivation, and the kinds of materials and scaffolding available. Some teachers have been amazingly successful with our inadequate documentation and others have been less successful that one would expect, given the amount of documentaiton that is there. Many children who like to explore and don't want to read documentation have done even better. Some children are quite stumped without explicit help (but that's what teachers are supposed to be for.)

But the clear lesson is that we need to provide enough coverage for a wide range of styles of learning. Please continue to be interested and to help.

Tue Mar 11, 2003*: I want to document but I need to learn first!*

Thanks Brent --

At 9:44 AM -0500 3/11/03, Brent Vukmer wrote:
Jerry --

Could you post your field notes from your eToys demo? Also it would be great to see what the teachers' questions were.

You may have already found this on the Web, but Alan Kay did a fairly detailed tutorial/exploration of eToys-and-Squeak for Tamika Knox's class problem. See http://www.squeakland.org:8080/super/200 .

I hate to say this but I pretty much forgot what I did here -- even that I did it -- and certainly did forget this link (life has been complicated the last 2 years ...). This is actually a pretty good start at some of the things that Jerry wants and needs. I think the reason that I didn't link this up is that I didn't get done with the general stuff and didn't hand it off to anyone else .. then it got forgotten. But, it's on a swiki so it is open to be added to and changed for the better ... Maybe we should link this into the squeakland.org site even in its unfinished state and hope someone (perhaps with the energy of Sebastian's students) will add to it.

Tue Mar 11, 2003*: I want to document...slightly off topic*

Hi John --

Again, this is really a squeak.org question, since squeakland.org is *only* about the etoys part of squeak. The short answer is the universal tiles were one of several experiments we did to investigate making an enduser scripting system of much wider scope than etoys. Some of it worked very well, but we judged the gestalt to be below threshold.

Tue Mar 11, 2003*: Etoy Viewer Commands Handout*

Hi Eric --

Most email clients have a size filter that stops large downloads until you say you want them. Does yours have this feature?

Wed Mar 12, 2003*: Volunteering for the documentation team*

You just did, and you're "hired".

Let's chat more off this list.

Fri Mar 14, 2003*: Limited palettes (was RE: I want to document but I need to learn first!)*

None of the limitations in etoys have *any* effect at all on really high quality projects by 10 and 11 year olds. That's who we wanted to test with over 3 years and (a) really good results happened, and (b) only about a third of the stuff we came up with easily covered a whole school year -- so there is plenty more that can be done.

I think one problem here is that you, like many adults, really want the next version of Hypercard with lots of features and wide range. This is good. That's what we want to do also, and we have been working on this for a few years. But this is not what etoys are about, as I've said many times over on this list. Etoys are an experimental authoring environment for kids around the age of 5th grade, done solely to allow us to test a bunch of ideas that seemed fruitful and needed testing. We made the work open source to attract potential colleagues, not to be a vendor (Squeak and etoys are not products, we are a nonprofit public benefit corporation operating on a shoestring for the public good, etc.)

Forgive me for saying this, but there's a certain amount of special pleading and rhetoric in your recent remarks. At one point you're using LOGO as "something that can't easily be learned", at another point you're invoking Seymour against BASIC. Neither of these have much to do with etoys -- in part because neither has a powerful dynamic object system with automatic graphical update. They simply aren't comparable and shouldn't be compared. The real heart of the matter is that children with pretty minimal help can do a wide range of projects that are engaging and empowering to them and that we think are intellectually interesting in the context of "real education".

The one place I agree with you is that "a new thing like Hypercard" (with even wider scope and higher ceilings) is what is eventually needed. But until that comes, a fabulous range of ideas can be pretty easily explored with children using etoys. (I.e. you shouldn't wait for the "76 Trombones" before you start a music program in a school. The children can sing and make instruments and a musical adult can bring them to very above threshold musical experiences with just that.)

Sat Apr 19, 2003*: Pendulum*

Hi Bert --

That is a very good point! I'd forgotten that embedding makes an object live in the coordinate system of its owner -- and that is what it is supposed to do. However, it's clear from this and other examples that I can think of that this perhaps should be the default, but it might be a good idea to put a flag on a player to "use the coordinate system that its owner uses".

Let me think about this over the weekend ...

Sat Apr 19, 2003*:* "*All the Real Math To Which School (Including College) Refused Yo u Access.*"

Hi Bert --

At 6:57 PM +0200 4/19/03, Bert Freudenberg wrote:
Am Donnerstag, 17.04.03 um 21:24 Uhr schrieb Alan Kay:

>I think lots of insight can be gained by seeing what the "weighing >angles" illustration is all about. > >Notice that when the angle is 90° the scale will measure the full >weight of the dumbell and wheels. When the angle is 0°, the scale >will show zero weight. In between, the scale will show the weight >of the dumbell and wheels in the direction down the inclined plane. >"Weight" is actually defined as the mass of an object times the >force of gravity on it (w = mg).

*If I were picking nits I'd point out that actually weight is a force (measured in Newtons), not gravity. Force is mass times acceleration (Newton's second law). So in this special case, weight is mass times gravitational *acceleration*. Weight is only another term for gravitational force. But you knew that ;-)*

> So what we are seeing on the scale is the differential effect of >gravity down inclined planes at different angles. > >If we use a protractor to tilt the inclined plane (say) every 5° >then we can write down the different forces down the plane. If we >divide these numbers by the maximum weight when the angle is 90, we >will get numbers between 0 and 1. These numbers can be put into a >holder as a table of values and used in a wide variety of projects, >including making a roller coaster. So there is no need to use the >idea of "sine" -- and this makes projects that need these ratios -- >like roller coasters -- much more in the range of 5-7th graders.

What do you think of measuring the forces in the Etoy itself (for example, by taking the vertical extent of a rotated line)? Of course, I can see the value of using real-world data.

I think this is really important at this stage. This is one of the relatively few phenomena that is both very interesting, useful, and measurable by the kids. This "weighing angles" idea cuts through a lot of steps and gets right to a way to determine the differential accelleration down the plane by directly referring to the phenomena.

Do you think it's too large a step to "see" the height of the angle, which is proportional to the force?

It shouldn't be too hard for adults ... heh heh. But kids are just learning about proportions (in the US they generally don't learn proportions successfully and operationally). I think this is a very good thing to point out after they have their simulated cars successfully going down different planes at the correct accellerations (Note that this can be done via one of the touch tests between the sim car and the sim plane.)

Mon Apr 21, 2003*:* "*All the Real Math To Which School (Including College) Refuse d Yo u Access.*"

Actually, what the 10 and 11 year olds do is:

1. measure from the bottom of a ball to the bottom of the ball in the next frame using the height of a rectangle that is stretched to fit.

2. They stack up the rectangles to see that the incremental change in height is constant. They know this is constant accelleration because they have played with it using their painted cars a few months previously.

3. they paint a small simulated ball

4. they then adapt the script they wrote for the cars to drop the ball in the vertical direction with constant accelleration:

```
5.    ball drop      ticking
          ball's speed increase by -1
          ball's y increase by ball's speed
```

6. then they tinker with the constant to find the accelleration that matches what is happening in the movie

7. Then they figure out a way to "prove" that they have "captured gravity"
    a. one way is to leave a little mark at each position of the real ball
    b. another way is to run the movie and their simulation frame by frame to show that the simulated ball tracks the movie

In other words, these children actually derive the dynamic relationship of (very near) constant accelleration near the surface of the earth in the form of a discrete 2nd order differential equation. It is quite remarkable to see this (especially for me, since I designed this project for 9th through 12th graders).

The key here is to spread out the various ideas and parts that have to be learned over several months in the midst of doing other things.

Thu Apr 24, 2003*: Pendulum project update*

Hi Folks --

A good side point about pendulums is that the motion is harmonic only for small excursions, since harmonic motion is roughly the spring law which is proportional to x, and pendulums are proportional to sine x. Sine x and x are close to the same values only for small angles.

All these subtle details are reasons why we don't do pendulums with 5th graders. Compare this to gravity near the surface of the earth where the accelleration is constant to about 1 part in a million (note that it isn't really constant because it is inversely proportional to the square of the distance and this is changing a little bit -- about 4 meters in the ball drop example).

***Fri Apr 25, 2003**: Pendulum project update*

Thanks Christopher --

A few years ago I did newon's method for square roots in etoys and it worked quite well.

***Fri Apr 18, 2003**: Remembering Philip Kniat*

Thank you Sheine --

When a child dies for any reason it is hard to find any words that are adequate to express the tragedy.

We plan to make a memorial for Philip on the Squeakland site. Perhaps Philip's classmates might like to help.

Best wishes to all,

Alan

***Fri Apr 18, 2003**: Pendulum*

There are many ways of doing frame by frame, but why not do it the way the children learn how to animate? Check out the project "Sam's Face Ball" on the Squeakland website. I think there is also a tutorial about how to animate.

But, again, I ask, why not do this with a real model instead of "just a story"?

***Fri Apr 18, 2003**: Pendulum*

Try this with television and see what happens ...

***Fri Apr 18, 2003**: "All the Real Math To Which School (Including College) Refused Yo u Access."*

Thanks John --

It would be great if you could list the "language stuff" that causes the glazing. Do you mean terms like "vectors"? What other terms are offputting? One of the reasons this stuff works so well with the kids is that they just do the models, we don't employ terminology with them.

***Fri Apr 18, 2003**: "All the Real Math To Which School (Including College) Refused Yo u Access."*

I think lots of insight can be gained by seeing what the "weighing angles" illustration is all about.

Notice that when the angle is 90? the scale will measure the full weight of the dumbell and wheels. When the angle is 0?, the scale will show zero weight. In between, the scale will show the weight of the dumbell and wheels in the direction down the inclined plane. "Weight" is actually defined as the mass of an object times the force of gravity on it (w = mg). So what we are seeing on the scale is the differential effect of gravity down inclined planes at different angles.

If we use a protractor to tilt the inclined plane (say) every 5? then we can write down the different forces down the plane. If we divide these numbers by the maximum weight when the angle is 90, we will get numbers between 0 and 1. These numbers can be put into a holder as a table of values and used in a wide variety of projects, including making a roller coaster. So there is no need to use the idea of "sine" -- and this makes projects that need these ratios -- like roller coasters -- much more in the range of 5-7th graders.

***Fri Apr 18, 2003**: Anyone seen Croquet working?*

Hi Jim --

Croquet is a prealpha system, but it works very well on various Windows versions (with updated OpenGL), Macs (also with latest versions of OpenGL) and Linux (you guessed it).

I pass this along to Andreas who can give more details if you'd like.

P.S. This listserv is only for the Squeak etoys. There is a croquet listserv, and also the main squeak.org one has correspondance about croquet.

***Mon May 5, 2003**: Pendulum*

Hi Phil --

At 5:38 PM -0400 4/17/03, Phil Firsenbaum wrote:
Interesting...I actually saw this message on the Squeak archive. i didn't receive it on the mailing list, though.

Anyhow, even if the pen were down and the pendulum simulated reality i think it would draw lines on top of lines unless the area under the pen was scrolling. Do you see what I mean?

Well, here's a little exercise in relativized thinking ...

Embed a round little object like an ellipse to be the bob of the pendulum, call it "bob". Make another little object called "plotter", put its pen down. See what happens when you do:

```
plotter's y increase by 1
plotter's x <- bob's x
```

***Mon May 5, 2003**: Pendulum*

There are many ways of doing frame by frame, but why not do it the way the children learn how to animate? Check out the project "Sam's Face Ball" on the Squeakland website. I think there is also a tutorial about how to animate.

But, again, I ask, why not do this with a real model instead of "just a story"?

***Mon May 5, 2003**: Importing a graphic*

A simpler way for a number of picture formats is to just drag the jpeg file from your desktop into the Squeak window.

***Thu May 8, 2003**: pen trails*

Hi Randy --

At 8:44 PM -0400 5/7/03, Randall Caton wrote:
Does anyone know how to get pen trails to write on top of an imported graphic (e.g. jpg)? Can it be done?

Yes (see Andreas' suggestion).

Interestingly, pen trails were originally on top in the first version of etoys, and I'm not sure just why they aren't now.

However, it's also pretty clear that a more appropriate implementation would be to have them actually make an object of some kind (a SketchMorph or a PolygonMorph) that works like any other object. We will probably do something like this in a future version of etoys.

***Sat May 24, 2003**: Has anyone played with Karel's World?*

Ned --

I love it! Besides being "a good thing" it is also a tour de force of etoy programming.

***Sat May 24, 2003**: Has anyone played with Karel's World?*

Squeak is very platform independent.

At 10:43 AM -0700 5/24/03, Doris Cassidy wrote:
I'm a new squeak user. Should Mac users be able to download projects and games created on computers using Windows OS?

Yes, this happens all the time.

I have tried but am unable to do so!

What are the symptoms? Can you download projects created on the Mac (how can you tell they are created on the Mac? etc.).

***Wed Jun 25, 2003**: flaps in the scripting presenter*

On the other hand, using just the etoys, you can make perfectly working flaps (nonstandard, but functional) and use them anywhere. I have done this several times with good results.

It's worthwhile to contemplate how this might be done. And it would make a nice piece of documentation of one kind of media construction.

***Thu Jun 26, 2003**: Launching squeak*

What happens when you just double click the image icon? This should work just like any other Mac app (where the image plays the role of the "document").

***Fri Jul 4, 2003**: Pen Trails and Sensors*

Thanks Ned --

At 8:27 AM -0700 7/4/03, Ned Konz wrote:
On Thursday 03 July 2003 07:20 pm, Jeff Longland wrote: > I've been using > penDown to create a pen trail behind the snake, but my major > problem is that the sensor on my object is unable to view the pen > trail.

Welcome Jeff!

Some context for the rest of the list: unless the Squeakland image has a fix that isn't in the Squeak 3.5 image, the turtle trails aren't visible in the color:sees: test.

That's odd and interesting -- they used to be, and should be -- for just some of the reasons that Jeff needed them. There are many wonderful things that can be done if the color sensing can see the turtle trails. Any idea of how this got lost? Can Scott Wallace easily put this back in?

***Fri Jul 4, 2003**: Pen Trails and Sensors*

Thanks you very much Bert!

***Mon Jul 7, 2003**: RE: "transparent skin" (new user Q)*

Nancy --

I think what happened is that we changed over to a new website for Squeakland about two weeks ago and modernized a few things but didn't catch up to all of them in the tutorials. We'll get the tutorials redone and more usable over the next week. For now you could try a new tutorial that is in HTML, so you can print it out from your browser. It's also on the Squeakland site at:

http://www.squeakland.org/school/drive_a_car/html/Drivecar12.html

Comments, suggestions and criticism welcome ...

Mon Jul 7, 2003*: Squeak 'non-starter' in U.K. schools?*

Hi Jim --

At 10:55 AM +0100 7/7/03, Jim Ford wrote:
Hi,

I'm a science technician in a U.K. Independant (non-State) Secondary School. I've had experience of several programming languages (including Logo) and when I came accross Squeak became a convert to the concept of it being an excellent learning tool - not only for children, but adults as well.

I've tried introducing Squeak to science teachers, but encountered the problem that I've come across with other ideas I've had, which is: if it's not in 'The National Curriculum', it won't get taught.

The US is definitely moving in a similar direction: towards extremely rigid national curricula.

As has been mentioned many times in the U.K. national papers, our schools are so focused on gaining good published examination results - the so-called 'League Tables' - (in spite of the protestations of some Head Teachers), that _nothing_ outside 'The Curriculum' has the remotest chance of being taught. The pressure on teaching staff to 'Deliver the Curriculum' is such that whilst they may show interest in Squeak, there is not the tiniest slot in the teaching day for it to be introduced.

I believe that as long as the U.K. education system remains tied to the stultifying influence of examination orientated 'League Tables', innovative ideas such as Squeak will never be introduced, unless (as is _most_ unlikely) as officially part of the National Curriculum.

There are various ways to look at this. In the US, it's really a mixed bag, because the "official curricula" are poorly taught and learned, and so, looking on the bright side of things, it's good that important subjects like music, art, and real math and science aren't official and thus don't get ruined for the children. However, I can't quite get myself to be that happy about the current situation, since the names of important and interesting subjects such as math and science are ruined in the children's eyes, and this taint can remain for many years.

When we started this effort many years ago in the 60s -- inspired by Seymour Papert -- pretty much everyone then thought that most gains would be somewhat subversive and outside of formal schooling, and that the advent of personal computers and the Internet (both of which were well underway) would provide something more like nonschool books, libraries, bookstores, etc., from which anyone could learn by themselves and in clubs with others. It is likely that this set of envisioned processes will be what is required -- and to have quite a bit of child to child mentoring -- in order for any real changes to happen in the next decade.

By the way, in the US at least, things would be helped tremendously if scientists and mathematicians were much more strongly involved in elementary schooling (and in clubs etc). This is one of our biggest problems: not enough people who actually understand the real content are involved and want to be involved.

Mon Jul 7, 2003*: Squeak 'non-starter' in U.K. schools?*

Hi Sheine --

I didn't actually decide on the bookstore approach to the reading list. The list is many years old (and probably needs updating). The kind folks who volunteered to do the new website took a lot of materials and decided to provide secondary and tertiary links to most of the stuff. I think this helps a lot and provides more depth to the site. The links to amazon are quite helpful even if you don't buy a book, since more can be found out about the books if you are interested. So I think this was a good idea. I find I use amazon about 60% of the time for the same purpose: to find out about a book rather than buy it. Historically, I grew up poor and thus virtually all of my reading was from the free public libraries -- so I'm a big supporter of this way of learning. In particular, I have very fond memories of the wonderfully generous folks at the Queensborough Public Library in Jamaica, Queens, who really went all out to help the young teenage me to find out about things I was interested in. A lot of our interest in making the Internet and keeping Squeak free came from my experiences with free public libraries.

Tue Jul 8, 2003*: Squeak 'non-starter' in U.K. schools?*

Hi Darius --

I'm pretty sure that Jim was treating Squeak as a medium for certain kinds of content just as you suggest, and I certainly was. In any case, as long as we are being really careful about terminology here, even "Squeak" is not quite accurate, since we are only using the very restricted etoy environment (that is one of many facilities within the Squeak system) with children to help them learn powerful ideas by authoring models of them.

I think Jim was expressing the difficulty of introducing ideas and processes (whether good or bad) that are different than the officially sanctioned ones. To me, a very important characterization of the problem in the US is that if the children were getting 100% on their tests in "math" and "science", they still would have learned almost nothing concerning "real math" and "real science". Helping the current processes won't help real education in these areas. The real difficulty is getting the real processes and ideas understood and underway. It is not at all necessary to use computers for this, but computers can be very useful "real math stuff", and perhaps they can be subversive enough to get the real ideas under the radar screens of the misled establishment.

If you are interested in the actual effects of media on thinking (they aren't neutral), McLuhan and Postman are two good places to start.

Wed Jul 9, 2003*: Squeak 'non-starter' in U.K. schools?*

Hi Darius --

Two things worth checking out here.

First, is "The Analyst" done in the early 80s by Xerox EOS in Pasadena in Smalltalk (originally for the CIA) but then sold as a product. Its main feature is just as you suggest: a spreadsheet composed of views of Smalltalk objects - it was quite nice and very powerful (and probably still exists somewhere).

Second, is "Agentsheets" done at the U of Colorado over the last few years. This is a pluggable cell spreadsheet system with complete objects as cells done for children.

There are strengths and weaknesses with this view of enduser computing. Some of the strengths are obvious, but can you spot the weaknesses?

Fri Jul 11, 2003*: Etoy: Saving, Publishing*

Hi Nancy --

At 11:15 PM -0400 7/9/03, Nancy Head wrote:

> *How do I save a squeak etoy project?*
>
> *How do I "Publish It!"? Can I place this in "my own" web space (i.e. my account on my isp's server)?*

Yes you can, and there are a variety of ways this can be organized. For example, in one of the LA schools we work in, the classroom has quite a few computers but not nearly as many as there are children. The children's projects are published to school servers and they can bring them to whichever computer they are working on that day. The basic idea is that you should be able to author, publish, and find projects anywhere on the net with as close to a "one button" UI as possible.

This also gives children something most adults wish they had, which is a WYSIWYG full media authoring and publishing system for the web that can do all from within or without a browser.

As Andreas mentioned, some care has been taken to protect projects from being used as viruses. For example, they are encrypted when sent outside the child's machine and signature protected against being corrupted. Projects brought in from the outside can be run safely in Squeak in part because they are confined as to the resource access powers they have on the child's computer.

Fri Jul 11, 2003*: decompress file*

Hi Nancy --

I don't have a clear visualization of just what you've got.

If you have a .pr file, then just put it in the Squeaklets folder where your plugin image resides and you can get it with "Find" on the Navigator bar.

When I downloaded Diego's project in IE, it automatically decompressed and made .pr files. You can just use these directly.

But remember that the plugin version has resrictions on it for what it can see in the larger file system. This is why you should put these external .prs in the local Squeaklets folder as mentioned above.

Diego's instructions don't have anything directly to do with the plugin version.

Fri Aug 8, 2003*: Logo vs. Squeak*

Hi Diego and folks --

There is no "versus" here. Logo is great, and it was the inspiration for much of what we've done over the years. In fact, I've been encouraging the Logo folks at MIT -- Seymour Papert and Mitchel Resnick have been long time colleagues and advisors -- to actually put a Logo on top of Squeak -- pretty much everything is already there except the syntax and UI.

Before I mention a few differences in point of view, let me say that the main aim here is to teach children real math and real science, and this can be done just fine with Logo. However, with both Logo and Squeak it really helps for the teachers to already understand real math and real science beforehand, or to really try to learn the real subjects and processes along with the children. (In the United States, most elementary school teachers are not fluent in real math and real science, and the official curricula are not about the real subjects, but about "rules for calculation" instead of real math, and "science facts to memorize" instead real science.)

There have been many variants of Logo, all with a fairly similar syntax. There have been several versions that are more or less object oriented, with a number of different syntaxes to deal with addressing messages or commands to different objects. The more recent versions of Logo have sprites with costumes, and these are basically objects.

In the late sixties, influenced by Logo and by some previous object-oriented work I'd done, I started thinking about object-oriented languages for children. One simplifying idea was to have everything the child encounted be an object, so there was only one coherent world view to understand and use and just one way to get objects to do things.

In Squeak, this simplification goes even further, so that every object is also "a turtle with a costume" -- (make a script in Squeak, get its halo, and look at its viewer. You will seen "forward" and "turn". Look at the "pen" category, and you will see that the script itself has a pen. Thus you can easily make a script for the script that will make it move in a circle!) The basic ideas here are simplicity, uniformity, a glimpse into the metanature of computing, etc.

Another noticable difference in Squeak is the tiles UI. This has turned out to be great for beginners of all ages. It really encourages rapid experimentation early one without worries about syntax, spelling, etc.

A current "drawback" in the Squeak etoys is that they were an experiment aimed at a particular age group -- 9 to 12 year olds -- for particular purposes -- about 50 etoys in math and science. The good news is that, in this range, they really work extremely well, and are learned by virtually all children and adults who try them. That was the experiment. The downside is that there is not a lot of extension in the current system, and it gets awkward for older children and experts.

For the last several years, we've been working on a version of this that starts out as easy to use as the Squeak etoys but is much more graceful in how it expands as a learner gets more fluent. We will put this new version out as an experiment this Fall for those who are interested. This version can carry multiple syntaxes, and it's likely that one of them will be a variant of Logo -- that would be fine with us -- this would make a large world of Logo documentation available to all.

Just a pause for a thought here ... Neither the current Squeak syntax nor the Logo syntaxes are ideal for children and other end users. We really should be thinking about what improvements in UI should be made to help them. Andreas Raab has pointed out that the syntax of a programming language is actually part of its user interface -- and I think this is a really important observation. If we look at the difficulties of having children understand (say) parameter passing in Logo, we should be thinking about how it should look.

(It should probably look more like explict assignments to the internal variables of the procedure than the blind magic that now exists. I left parameters out in the first version of etoys for this very reason. It is much easier for the children to make exlicit assignments to the local instance variables in the parent object before calling the procedure. But this doesn't work for recursion, so what should probably be in there is a tile that bundles up the assignments and the call. Etc.)

"Functions" and things like functions are a powerful idea. So we should be thinking about how to make this stuff better, not just how to make use of it or to ignore what the children have difficulty with. It's not that difficult to make languages and UIs for different ages and sophistications, but it is quite difficult to make the graceful blend and path from the simpler to the more sophisticated.

In the sixties, there were lots of computers and lots of different computer languages. Most practitioners back then quickly got "multilingual" and learned to program in many languages. Nowadays, different syntaxs seem to be a much greater barrier, and it seems worthwhile to cater a little more to this barrier in order to try to teach the underlying ideas.

Sat Aug 9, 2003*: Logo vs. Squeak*

Hi Ken --

There are definitely a lot of good ideas in ToonTalk -- and, as you know, I've been interested in various kinds of iconic programming for many years. I think some nifty combination of iconic and symbolic elements (yet to be discovered) will indeed be part of a much better authoring system for all.

Your knot example is a good one, but so is the fact that you used English to state your case below. I think you would agree that a combination of English and pictures and actual manipulatives would be even better, just as quite a bit of math is difficult to express only in pictures, though pictures and manipulatives are a great way to start off.

Sat Aug 9, 2003*: Logo vs. Squeak*

Hi Anindita --

As I said in my original email, the etoys were aimed at a very narrow experiment, range of children's ages and range of both subject and UI problems. So I would agree with many of the comments below. And some of the disagreements I have below are our own fault for not documenting well or widely enough. For example, the assertion:

At 11:55 AM -0400 8/9/03, Anindita wrote:
In Etoys, one cannot create a new class of objects-- just new instances of objects. I can create and program an object, then copy it so that it has the same characteristics, but I cannot define a class, then create and/or modify instances.

is much stronger than is the case. The object system in etoys is not the same as Squeak's (partly for good reason and partly for experiment). It's much more of a prototyping system and as such allows quite a bit of flexibility. The way to think of it is that the vanilla "Player" is like a class Object that just happens to know about graphics, etc. It would be easy to put a few more things in "Player" to make it fully general, but that was outside the particular experiment here.

Ditto "complex code". This was outside the experiment, but it really does limit older children's range.

All the stuff on the net was done via BJ-Conn's classroom, and she started her kids with a blank environment. If you don't want a blank environment, then just make one and store for children to start with. All the media stuff you mention in Squeak is there for just that purpose. Why complain about this when you can do something to help?

BTW, you can set a preference to have the halo of handles come up on mouseover: many teachers use this, some don't. There is balloon help on most things in the interface that is delayed one second so it doesn't get in the way of those who have become more expert. IOW, there are things you can do to deal with these problems. E.g. most of the many hundreds of children we've had experience with don't have any problems here, so I think it's more a style of approach that is affecting things.

I do think the tiles are a double-edged sword, especially on slower machines. I also think there are better ways to do this kind of scripting, especially when more elaborate expressions are desired. Here's a good opportunity to contribute to this opensource system. We've already gotten some good ideas from our colleagues at CMU's Alice project, who also have similar design goals and difficulties.

At 11:55 AM -0400 8/9/03, Anindita wrote:
At this point, it might be good to take a step back and rethink how to do scripting in Squeak so that children can access the powerful ideas and flexibility of Smalltalk more simply. Just as Logo serves as a simplified Lisp, there could be a simplified Smalltalk for children to use. Ken raises some good points about using icons. The two could also be combined, as Alan stated.

I think this is the right thing to do as well. We have been doing this quite deeply in the new (Tweak) system that Andreas Raab is making, and I'm hoping that the new (Scratch) system from MIT will shed light on all this as well.

At 11:55 AM -0400 8/9/03, Anindita wrote:
How can one transition from being an Etoys programmer to a Smalltalk programmer? Right now, the gap is rather wide, but ideally, the ceiling should be that high.

That is our favorite question. Let's all try to answer it in the most comprehensive and wide-perspective fashion possible.

***Sat Aug 9, 2003**: Don't forget about Boxer*

As long as we are being ecumenical, don't forget about Andy diSessa and his (and earlier Hal Abelson's) Boxer work that has been going on for many years at Berkeley. A recent deep book about this is:

Changing Minds: Computers, Learning, and Literacy

There are many great ideas and valuable insights here. --

***Tue Aug 12, 2003**: Logo vs. Squeak*

Hi --

What Andreas meant *starts* with the Hypercard Level 5 and works its way up to much deeper ideas. This is the part that hasn't yet been done well (or pretty much at all).

***Thu Aug 14, 2003**: Logo vs. Squeak*

HI Mikael --

Thank you very much for your email. I have several somewhat overlapping reactions to this.

First, I think it's great, and is a great approach in general -- especially for early learning stages and for explanations. The idea of using a comic book style has come up several times over the years, but we've never pursued it -- and it's terrific to see what you've done so far.

Second, my larger opinion about this is pretty similar to my larger opinion about the place and relationship of comics to more text-based writing. On the one hand, I think comics are an art form all of their own, and they can have quite a bit of evocative power. On the other hand, I think there is something special and unique about "disembodied text" and what is required to read it and produce one's own internal realizations of the ideas. So I'm always very interested in helping children learn the harder stuff because I think quite a bit of it is really good for them.

In the world of programming, we have more than one kind of goal for the endusers, and this definitely influences the kind of designs we put before them. Two of the most important partially overlapping goals are (a) as a productivity tool, and (b) as a tool to shape the learner's mind.

For (a) we would generally try to maximize quick success in a project, often by putting in lots of built-in functionality that can be hooked together to make the end result. For (b) say, for math learning, we might rather want to have fewer and more primitive building blocks and try to motivate harder, more difficult work and play from the learner because our main goal is not just getting a project done, but to effect a real and often qualitative change in the learner's mind.

These two areas overlap a bit, because a lot of the motivation in both areas for the enduser is "reasonable success for reasonable effort" -- so it would be ridiculous to make (b) painful, or to have stupid gratuitous difficulties that have nothing to do with the learning we're hoping for. But I do think that a lot of making a good (b) is about "finding good difficulties" whose surmounting will help the learner and motivating the learner to surmount them. And, in the end, the productivity tool approach in (a) starts to fall down because it is very difficult to provide all the plugin features (and very little deeper learning is happening while the features do cover the space).

An interesting example for me is Photoshop. It has a zillion features and is very useful. But, the user never learns anything about image filtering and is simply blocked if the filter they want isn't there. Mitchel Resnick and his group at MIT are doing a system for teenagers in Squeak (called Scratch) that tries to bridge this gap by e.g. combining the use of filters and the learning of the "computer math" of filters (by making their own filters from the start) as the kids make projects. In other words, the idea here is to keep the children "in the conversation" of the tools they are learning, not to "help them to death" by providing opaque features. This seems very good and important to me, and is an important part of how real computer literacy will eventually be learned.

These transitions are part of the ongoing conversations about how to help learners move from one comfort zone to a further higher level of sophistication. The relation of picture books for the very young to comic books and other picture and text books to mostly text books is a possible model to look at. The relation of your work to "computer math" seems quite important to me because, at least right now, the target for most children is a visual scene and objects that are usually manipulated kinesthetically. A lot of the power of the model is in symbology of one kind or another, and it is also helpful to have comments and other expressions in one's native language. This begs for a little more form and organization. I'm very interested to see what the next stage of these ideas will be.

***Thu Aug 14, 2003**: Book and DVD*

Hi Jimmie --

Kim's and BJ's book is on schedule and is being printed as we write. It is supposed to be available on Aug 20th.

***Thu Aug 14, 2003**: Book Information: Powerful Ideas...*

Hi Jimmie --

I think homeschooling is generally a very good idea -- and we have in our plans to try to do quite a bit more to help homeschoolers over the next several years.

I think any parent and/or teacher should be able to start from scratch with this book -- that is the intent. All the projects can be done by a single student.

One of the things you might consider as part of homeschooling is to set up community clubs in certain areas (like science) where a number of children can learn and do projects together.======

Fri Aug 15, 2003*: Logo vs. Squeak*

Hi Folks --

Mitchel Resnick just showed me "Howtoons". Check this stuff out, it's really terrific!

http://www.zeroprestige.org/craftoons/

Fri Aug 15, 2003*: Music and Squeak*

"Real Guitar" is ultimately more fun to learn and play than "Air Guitar" (and much better for all learners, especially children).

At 12:12 PM -0700 8/15/03, Doug Wolfgram wrote:
I just got back from Tech Fest and was fortunate enough to spend some time with Tod Machover of MIT Media Lab. I was completely impressed with Toy Symphony's HyperScore program (www.toysymphony.net) and have been composing music with it ever since downloading it.

Basically, it is an application that allows children to 'paint' music. One does not need to understand music notation or theory. It is all done visually and aurally.

And, to make an analogy, the invention of the tape recorder and other audio technology would allow children to just "paint their voice. One does not need to understand writing notation or the theory of how to use written language. Etc."

So, the other way to look at this is to try to understand why it might be beneficial for children to learn certain difficult arts: to read and write and do mathematics and play and compose music. What good is there in surmounting the difficulties of these arts? What are the real trade-offs here between ease of initial use and actual level of understanding after a while? Do we want a person+tool to be just the simple combination of the two, or do we want to qualitatively change the person for the better and the deeper?

Sat Aug 16, 2003*: Music and Squeak*

Thanks Ken --

For those who are interested, you could try to see what you can do with the very easy two line script that will sample in real-time at any pitch. You can also try to use "color sees" and other devices to get parameters from the world that can be used to control sound. Squeak itself can synth about 100 parallel real-time stereo tracks, and there are synth methods in there for sampling, FM, wave shaping, etc. There is the ScorePlayerMorph which can play midi files and allows you to both orchestrate and to compose, etc.

This is a big interest of ours, but we didn't quite get the music part of Squeak up to being smooth enough and documented enough for general consumption and play. However, I've done lots of music stuff in demos, and lots can be done. One example that was particularly interesting was to play Beethoven's Fifth with normal instrumentation and then replace all the instruments with the clink sound (that has just enough pitch to work). The result is mostly the rhythmic part of this great piece and many nontrained musicians have found it particularly insightful as to how Beethoven goes about his art.

Tue Aug 19, 2003*: A somewhat silly suggestion for Andreas*

Why not call a Tweak object an "Idea"? This would be even more Greek, and more appropriate.

Wed Aug 20, 2003*: A somewhat silly suggestion for Andreas*

John --

Actually, our main partners are children, teachers, and parents, and none of them will be interested in knowing how Tweak differs or is similar to other object systems. They just want to do stuff, and we need to show them and help them. Computer jocks can ask questions about the model on the Squeak list.

This is why my answer was facetious .....

Thu Sep 4, 2003*: Maze constructor*

It might be a little more intuitive if you put in a special player that signifies "empty" -- this was often used in the sixties for stacks, especially in mathematical explanations of stacks, etc.

Sat Nov 8, 2003*: Squeak For a First Grader?*

Hi Dave --

Six is pretty young -- not so much intellectually, but in terms of "what seems neat". Some six year olds love "camoflage games" (like a painted animal in an environment where it can't be seen until it moves) and visual illusions (like a dynamic version of the Mach contrast illusion, etc). Many of these have been done in Squeak. Take a look at an illusion book and see if she likes them. If so, I can suggest how to do dynamic versions in Squeak.

Mon Jan 5, 2004*: Q: Leverage Overlapping Player to Etoys?*

Hi Ned --

At 11:13 AM -0800 1/5/04, Ned Konz wrote:
On Monday 05 January 2004 8:25 am, Markus Gaelli wrote:

> Did we overlook some other way to find out with Etoys which of the UFOS > have been hit by our shot?

Well, there's the colorSees: test.

> If not, should be put it to miscellaneous?

> Is it a problem that we only > return one of possible many overlapped players?

How would you return more than one?

I've been thinking about making a CollectionPlayer that can apply commands and slot setters to its members; if we had one of these it could be returned. However, there's a problem with slot getters, and with the display of such a collection (what should it look like if it's visible?).

It should look like (and be) a Holder.

> Should I send the changeset?

Of course, though it may be of more interest to squeak-dev

> Another question: How do you destroy Players with Etoys? Is there > something there and hidden?

You'd have to get rid of all the costumes.

In the case of players with identical behavior, you'd probably want them to be siblings, so there isn't the problem with class duplication. The "copy" pseudo-slot getter gives you one of these.

> We created the bombs of the UFOS by copying the "mother of all bombs". > But we also wanted to > destroy them, if they hit the ground.

You could do
theBomb doMenuItem: 'delete'

This is a good way.

The way I do this is to do everything in a Playfield (which is a kind of Holder or vice versa). Then I set up another playfield "PF" to use as temporary trash. So:

```
        PF include theBomb
        PF removeAll
```

When I'm doing stuff that results in lots of "atoms" in a Playfield, I reset by taking everything out of the playfield that I want to save (like the mother of the bomb)

```
            PF include bombMother
```

and then do a Playfield removeAll.

_Thu Jan 8, 2004__: New User_

Hi Pat --

The "Powerful Ideas In The Classroom" book that is available on the website is a tested curriculum of the first dozen or so Squeak Etoys projects for elementary aged children. This would be an excellent place to get started, and please don't hesitate for a second to ask for more help if needed.

The DVD might give you some insights into the physical processes and children's reactions to this kind of "hard fun".

_Wed Jan 14, 2004__: Some questions about scripting an eToy player_

Hi Darius --

Here are some answers from the enduser's point of view.

At 1:07 PM -0800 1/14/04, Darius Clarke wrote:
Dear Squeakland team,

I've developed a couple questions recenty:

- Which Morphs/eToy players have extra scripts attached to them? Here is such an example, the Text morph whose menu is attached here to this message.

- Where is this documented so I can find out such things for myself?

For endusers, it isn't. It's assumed that the EU will get a viewer for the object in question and look at the bottom of the category menu. This was more of an experiment than a feature, and we intended to make such a category for most morphs. We will do this in the next version of the system.

- How can I change the #rotationCenter in an eToy player's script? - The the #rotationCenter in relation too what point in the player?

I'm not sure what you mean here -- change the rotation center using an etoy script? If so, there is no route for this. To change the rotation center "by hand" you can shift drag it.

- Does the Croquet mailing list still work?
http://mail.opencroquet.org/pipermail/croquet-user/

I think so.

- Where do I send my "SqueakDebug-3251534861.log" files?

Michael Reuger or Scott Wallace.

_Mon Feb 2, 2004__: Re: Squeaking Homeschoolers?_

Hi Aaron --

This would be a nice email to send to the squeakland.org list. I'm copying over there.

At 1:20 AM -0500 2/1/04, Aaron Lanterman wrote:
On Sat, 31 Jan 2004, David T. Lewis wrote:

> You are on the right list for the kinds of student you have in mind, > but you should also be aware that Squeak is being used for some very > innovative educational purposes for younger kids. See www.squeakland.org.

When I picked up the Squeakers DVD, which shows BJ Allen-Conn's work with Squeak, I also picked up a couple of copies of their "Powerful Ideas for the Classroom" book.

Just curious: has there been any effort to "market," so to speak (in the sense that one markets something free) Squeak to the homeschooling population? (My wife and I don't have kids yet, but *when we do we plan to homeschool. My wife taught 7th and 8th grade French for three years, and then went into tech support to get good pay and respect... after she left teaching she started researching homeschooling.)*

I've read every one of Alan Kay's essays on the squeakland.org site. I doubt he intended this, but if you really put together his arguments, they form an incredibly powerful argument for homeschooling, which emphasizes personal exploration of ideas on one's own timetable. Homeschoolers tend to be all about "getting their hands dirty" - i.e. if you want to learn how to make pots, you don't really get far by reading about making pots, or even by watching someone else make pots. You won't really get anywhere until you stick your fingers in the clay - and I think Squeak could be characterized as a "Clay of Ideas."

The first aims for the Squeak etoys were at what we called "lapware" (a parent with a child on their lap). And we have been wanting to do a major push for homeschoolers for quite a few years.

I think we now have enough content for the K-5/6 range, but we really lack enough of the direct and ancillary materials that parents need. This is especially acute in the earliest ages, where the advice to parents needs to be particularly rich in order for them to be good scaffolders (this is also true for teachers in the earliest grades). We've found that moderate levels of help (such as BJ's and Kim's book) seems to work fine with 4-6 graders. To take a later example, 7th and especially 8th graders can cover a tremendous amount of ground, and here again we need lots more scaffolding for the helpers in areas such as science and math. So the effort graph kind of looks like a smile with 5th grade being a pretty good sweet spot in which to get started.

I have been toying around with notes for a more complete "5th grade for homeschoolers" curriculum but even this is daunting with our small resources (also we like to test our curricula for 3 years before thinking we have results that can be trusted).

When I showed my wife the Squeakers video, she said two things that really struck me:

1) "Wow, if someone had taught me math _that_ way, I would have really gotten into it!" (she historically hated math in school, all the way through)

That's because most schools in the US simply don't teach math. They try to teach pattern-matched calculation skills which are not much fun. Math is beautiful and fun.

2) "They could NEVER have done that at a public school like the one I taught at." That was in reference to the gravity experiment - she said in disgust that at the school she taught at, one could never get permission from the various powers that be to do such a thing. In particular, there was a kid who stayed writing down his obervations once the experiment was over. My wife said that at the school she taught at, and probably most public schools, someone would have been screaming at the kid to get his butt inside along with all the other kids according to schedule.

A neat side note is that the Open Charter School is a public LAUSD school, but was set up as a Magnet school when busing was a big issue. And Magnet schools in LA were allowed to have much more control by the parents, teachers and principal as to how they did their thing.

It seems to me that what lets Squeak take hold in BJ Allen-Conn's world is the philosophy underlying the Open School that she teaches at. In a world of public school mediocrity, particularly in the realm of math and science - which Dr. Kay so eloquently exposes in his essays and in the speeches I've seen on line - a world strangled by things like the No Child Left Behind act (i.e. Every Teacher Screwed Over Act) - the power of Squeak will have trouble taking hold.

Sorry, I got a bit overexcited there... it's just so thrilling to find people so dedicated to creating new educational experiences to share ideas with. I was very glad to meet Mark Guzdial, who was the first person I met at Tech that I could seriously sit down and talk about these issues with.

Nice to be working and playing with you!

***Tue Feb 24, 2004**: Re: suggestions for pedagogical examples in Squeak*

Here's an email that's been waiting for me to add some examples. But I won't be able to get to this until next week, so am sending now.Hi Richard and Marsha --

Here are a few suggestions for using etoys.

First, though I'm guessing (hoping) you already are using "Powerful Ideas in the Classroom" by BJ Allen-Conn and Kim Rose, I need to mention it just in case. It contains ideas and directions for about a dozen projects that children like to do, are "good" for them epistemologically, and have been thoroughly tested. Another very useful collection of projects that have been written up nicely are from our friends in Toronto in the 8th grade of Don Mills School with teacher Sebastian Hergott <sebastian.hergott at tel.tdsb.on.ca>.

Here are some comments on Marsha's email.

Squeak etoys are in the form of a scriptable multimedia environment, so what gets authored can range from presentations (such as in powerpoint), stories and games (such as in MS Word, Director and Flash), to mathematical and scientific simulations. Each of these can have a little (to a lot) of overlap with various kinds of educational goals (including good ones). This is rather like introducing a word processor into a classroom "plus plus". That is, the authoring system needs to be really open-ended to deal with all of its genre (a word processor in which one could only write stories but not essays about important ideas would be a terrible use of technology). To continue the analogy, writing is much much more than just putting words down on paper, and having paper and pencils (or hightech word processors) is not much help if the teachers don't know what writing is and have some ideas about how to teach it.

Children don't know much about writing or math or science, but they do know a lot of stories and games, and a bit about stories and games, so they tend to plunge fearlessly into using a dynamic medium like Squeak etoys to make representations of stories and games of many different kinds. This simply follows previous observations of children with LogoWriter and Hypercard. There is nothing wrong with this, it's an easy way for them to learn the mechanics of using the system, and they can occasionally learn something beyond stories and games (e.g. a little about math) in the process. There is a part of NSF and the National Research Council that thinks this is worthwhile all by itself because learning to "do multimedia", especially with scripting is thought to be an important part of "technological literacy".

At 6:06 AM -0600 2/17/04, RATZEL, MARSHA wrote:
I'm now working with the head of the math department on the sligh to create a summer workshop....it just seems to me that I understand enough that I could help them do some neat things when they teach the "Moving Straight Ahead" module of the Connected Math Program module. This seems like a perfect language and integrated piece of technology for helping kids to "get" linear equations. They could experiment and see the what "if"s of all that. (I taught math and science before becomeing the computer teacher).

This is a very good thing to do, and there are really two parts to it: "real math" and "school math", which are not the same thing in most school systems in the US. It is critically important for the children to have some "real math" and "real science" experiences -- which include actually being mathematicians and scientists themselves -- doing the stuff -- as opposed to the "math appreciation" and "science appreciation" that is most of school math and science (analogous to the difference between playing and composing music and "music appreciation", which is about what other musicians have done).

Real Math is the easier of the two because "math is about itself", and a full experience can be had with very simple media (including simple computer media, such as LOGO). One really important point about real mathematics is that it is about completely understanding, deriving and reasoning about relationships. "School math" tends to be about "remembering results and methods" not about understanding, deriving and proving. So to most real mathematicians like me, "school math" isn't actually math in any important sense. This is a serious problem because it pits large systems of millions of nonmathematical adults who are committed to a particular theory of schooling against a few thousands of people who actually do and understand mathematics (we have been losing badly for more than a hundred years).

The two big things that have to be done to help children with mathematics are to (a) have the mathematical experiences be real and above threshold, and (b) to have the mathematical experiences be consonant with children's abilities and motivations to represent and reason in beyond commonsense ways.

So, as Papert pointed out in the 60s, one good thing about children learning how to program a computer is that they are actually doing real math: they are representing ideas in formal structures, learning some tricky tools (such as functions with arguments), reasoning about ideas using representations, using the computer to help debug their reasoning, and exhibiting the equivalent of constructive demonstrations and sometimes proofs of what they are trying to do. What seems to be a bit of a struggle here is actually a virtue: an embodiment of the real process of mathematics that requires the practitioners to understand the relationships and be able to say why.

However, quite a bit a computer programming can be done that, while "real math", is not above any worthwhile threshold. For example, the LOGO turtle can be used to make simple drawings (and this is real, though pretty trivial math), or it can be used as Papert intended to have children gracefully and easily learn about the differential geometry of vectors (the main mathematical language of science that is full of profound and above threshold ideas -- and is rarely even touched on in school math an any level). In spite of many books about how to teach this and other kinds of above threshold math using LOGO, most K-8 teachers did not understand what it was about and did not make the effort themselves to learn enough to help the children.

In the US, the further the children progress in school, the harder it is to help them learn real math, both from the interference with the non-math they've already learned, and also the interference with the high stakes tests they have to take in high school. This is why we concentrate most of our energies with 4-6 graders where there is still a little flexibility. For example, many of the most important ideas in mathematics -- such as counting and arithmetic, calculus, vectors, geometry, probablity, feedback and control theory, etc. -- do not require algebra to understand or work with, and important parts of these can be well taught much earlier than "school math" supposes. For many of these, a computer can really both enhance the experience *and* also connect motivations from stories and games, etc., to motivations for learning math.

What we've done in the Squeak etoys is to take as many of the ideas we think are great for children that have appeared over the years from many different sources and combine these with extensive multimedia to make a kind of a "superhypercard" that we hope will appeal to many different kinds of children and adults for their own reasons. This seems to have worked for our target group of 4-6th graders (more needs to be done for both older and younger children). A good project in Squeak etoys is one that first "appeals as art" and then has some serious nontrivial content that has to be worked out to get the whole above threshold. The "serious nontrivial content" could be mathematical, scientific, theatrical, musical, visual, etc., or some mixture. We've concentrated on mathematical and scientific, in part because these two areas are the most weakly represented in schools today.

For those who are interested in creating this kind of content, please let me point you to books and papers by Seymour Papert and many other LOGOites, Jerome Bruner (especially his ruminations on trying to make an intellectually honest version of cultural anthropology for 5th graders: a masterpiece), etc.

*****

Now for Richard's email.

First, I think pretty much everything you will need for your three day experience is in the "Powerful ideas in the classroom" book.

In what follows, I should mention that I was the one who made up the unfortunate term "object-oriented" in the mid60s (I should have called it something else). In any case, I have some strong ideas about what this term means and should mean.

You say: At 9:21 AM +0100 2/16/04, Richard Borge wrote:
We have decided not to focus on code as we feel there is a risk this can get boring. Instead we are focusing on general OO understanding and the use of a graphical tool therefore seems like the way to go.

I'm not sure what this means. 5th graders typically do the car and steering wheel project in less than an hour and they do it by writing code. And it is definitely not boring to them. My suggestion here is to avoid "music appreciation" ("object-oriented appreciation") and just have the children make interesting and above threshold things using objects.

Another suggestion is to take into account the way children think and do and know, and how these are different from adult thinking and doing and knowing. There is quite a bit known here, and this context can be found in the works of Montessori, Piaget, Vygotsky, Bruner, Papert, Brown (Ann), etc. For example, it seems much less important to me for children of this age to know *about* objects than to see computers as "personal powerful artistic material" for their ideas that they can shape using language, planning, reasoning, esthetics, etc. In other words, what "personal computing" was supposed to be about when we invented it many years ago.

It's very hard to understand a framework without having something to contrast it with, so I think I would avoid trying to get your 11-12 year olds to think categorically about objects in their first encounter. See if you can get them to love the experience, feel the power of expression, delight in the reasoning, and take happiness from being able to track down an bug and fix it, etc. The rest will surely take care of itself.

As there are already many 10-12 year old Squeakers all over the world, another part of your experiment might be to see what they think they are doing. Etc.--

---

***Fri Feb 27, 2004**: collaborative uses*

Hi Randy -

There are also a number of immersive UIs in Squeak. All are experimental (but have been successfully used at long distances (such as from LA to Japan and to Germany). We will shortly be releasing a new version of all of this and will let those who are interested on this list know when this happens.

***Sun Feb 29, 2004**: Charles Stark Draper Prize*

Thanks Sheine!

***Sun Feb 29, 2004**: drive-a-car newbie questions*

Hi Ken --

What tack did you take in your thesis (Andreas -- Ken`s thesis, called Director, was one of the first to use various kinds of clocks to tick processes along)?

Also, if you look in the menu in the etoy scriptor, there is an entry called "fires per tick". This is a somewhat kludgey but useful way of relating the animation time (ticks per second) to the number of times through a script per tick.Normally, these are 1:1, but can be as fast as 10000:1, which is fast enough to write an etoy that will step through and play samples at audio rates (see the Sampling etoy).

And, the Croquet stuff we are doing uses some of Dave Reed`s ideas (he is one of the 3 main folks), and this involves some interesting relationships between pseudotimelines and realtimelines.

***Tue Mar 23, 2004**: getting code from a .txt file into a script*

Hi Gary --

You are between two different worlds. The etoys part of Squeak has been artificially restricted and carefully designed to make it easier for children to make *etoys* using various kinds of media. There is a fair amount of design thought and testing in those decisions, and some of these are reflected in Kim's and BJ's book "Powerful Ideas in the Classroom". My strongest advice is to work through these projects.

(For example, you say "This is an etoy. How do I point it at the car object?" But it isn't an etoy. In an etoy you get an object by drawing or from one of the parts bins, get its halo, and click on the light blue "viewer" button, to open its viewer on the right hand side of the screen. This viewer holds all the phrases that have meaning to your object, and these phrases are already "pointed to the object". If you make scripts from these phrases, they will work. This is how etoys are programmed in the world of the child.)

Trying to do "other things" is quite possible, but I wouldn't advise trying to do it in etoys. The full blown Squeak can run the gamut from the very simple to writing its own OS, so it ranges from about 5 years old to Computer Science PhD projects. For example, it is quite easy to implement LOGO or any other language system in Squeak -- but there is still a fair amount of work and thought involved.

For most users, including many experts, it is the UI that determines in their minds just what they think they can do and whether they think they will like the experience. The UI in the full blown Squeak is *not* set up for children, and the first thing to do here if you wish to go your own way, is to make a suitable UI environment for children that gives them access to your ideas in a way that works for them. This is quite a design and testing task, but is definitely doable. However, if this is not done, then the children wind up with the worst of both worlds.

BTW, the best way to get children attracted to any of this stuff is to first get them to make a project of their own -- such as the "Drive a Car" project in etoys. There is a certain kinesthetic and visceral satisfaction to making things on the computer, especially things that involve programming. This is what the kids need to experience first and foremost. Showing them what other children have done should come later.

***Tue Mar 23, 2004**: getting code from a .txt file into a script*

But what does this have to do with children and the UI they need?

***Tue Mar 23, 2004**: Question: delays, collections, object instance variables*

Hi Thomas --

At 7:28 PM +0100 3/23/04, Thomas Oeding wrote:
forward for Poster at Kinetium dot Com <poster at kinetium.com>:

Hi.

Five questions from a newby.

1. It appears that tiles placed in a script are executed as sequential Smalltalk statements. Is this correct?

Sort of. The general intent is that everything done in one tick of a script is one set of mostly parallel effects and what you will see is the results of these at the end of the tick. But this was never carried through all the way, so ....

Statements that write into variables are executed sequentially.

For moving objects you have a choice of "batching" or not. If batching is turned on (it's a preference that can be specific to individual playfields), then all movements of an object are done and then the placement is shown. This is most noticable when the pen is down. For example, if you are plotting a curve by changing x and y in separate tiles then you will get the expected staircase with "batching off", whereas you will get a single line to the endpoint reached if batching is on.

2. Some tiles (e.g. make sound) appear to involve asynchronous calls to outside services, returning before completion of that service. For example, if I put 2 make sounds in sequence, they get garbled together.

Actually, they are just mixed.

Is my understanding correct, and is there a way to synchronize on the completion of the sound or to insert something like a delay between the calls so they are heard as two sequential sounds?

I'll defer to Scott Wallace here. This is certainly possible (and sometimes even desirable), but I don't know if it is currently a feature.

3. Is it possible to use collections? e.g. can I make a collection of ladybugs and then use collection methods? I don't mind using text instead of tiles.

As I said in a previous email today, the etoys were made to do etoys things. As some of the Squeak hackers have mentioned, there are various ways to get past the fences, but then you are in a UI territory that is not particularly good for 10 year olds.

That being said, if you look in the Supply bin, you will find a Holder, and this has a collection protocol that can be found by looking in its viewer. Read Kim's book to see how a few things are done using the Holder. If you look in the viewer of a text object you will see what can be done there. I.e. look in the viewer of an object to see what has been set up for etoys. ....

4. If I paint two eToys objects (sorry, don't know the right term), how can I set an instance variable of one to point at the oher?

Make the instance variable of type "Player". Look in the halo of an object and look at the balloon help on the different halo items. The brown one on the left hand side will give you a reference to the object. This can be used to set an instance variable to the this object. (We have never done this with children. It is generally much better to not use indirection when the children are learning.)

5. Lastly, I noticed that if I view the text version of a script and edit it, then switch back to the tile view, I lose the edits I made. Is there a gradual way to transition from the tiles, drag-n-drop programming version to full(er) Smalltalk or is it an all-or-nothing abrupt jump?

We should turn this off, since it really has nothing to do with etoys. It is there for certain adults to play around with, but it isn't a supported feature of etoys. I don't think this is the best route to make the transition to full Smalltalk. One of these days -- not too far off but still a little unpredictable -- we will release a system that is inbetween the etoys and full Squeak, that should be nicely suitable for most adults and teenagers, etc.

Wed Mar 24, 2004*: Re: Squeakland Digest, Vol 11, Issue 17*

Hi --

At 7:28 PM +0100 3/24/04, Thomas Oeding wrote:
Forward for Steve Gutierrez <sjg9360 at lausd.k12.ca.us>:

I've been following this discussion with interest. I'm very interested in getting sketch morphs to follow sequential commands a la the Lego Brick rather than running Forward 5 and Turn 5 simultaneously, so an object could complete a series of steps and then based on the results fire off a new script. Would this be done with text scripting?

No. As I mentioned, just turn off "batching" in the playfields you want to affect. The world is a playfield and you can turn off batching for your project by:
- getting the world menu
- clicking on "playfield options"
- make sure that "batch pen trails" is set to off.

This can also be done for individual playfields.

Also, it would be good if you could say more specifically what your larger goal is here. There might be a nice way to do it in etoys .... Are you trying to run a maze, etc.?

Thu Mar 25, 2004*: Real beginner's questions!*

Hi Shelagh --

At 1:53 PM +1100 3/26/04, Shelagh Manton wrote:
In the various books on Squeak and Smalltalk that I've read, they use a vertical stroke to delineate variables. where is this on my keyboard?

It's usually on the right hand side above the return key. It should show on the keyboard of any PC or Mac.

Is it a character that changes in Squeak when you type it like the underscore?

No. It's a regular character.

I can tell by your question that you are not looking at the Squeak Etoys, which is what the Squeakland site and this mailing list is about. If you haven't done much or any programming before, you might find a really great start by doing the etoys stuff. Get the book

"Powerful ideas in the classroom" or look at the tutorials on the site and try the examples.

If you have had some experience programming and want to learn the full blown Squeak, then the site is Squeak.org, and there are many books available about how to program both in Smalltalk in general, and in Squeak in particular.

Fri Mar 26, 2004*: Real beginner's questions!*

Hi Shelagh --

At 4:48 PM +1100 3/26/04, Shelagh Manton wrote:
Thankyou, I found it. It looks like a colon on my keyboard.

Shelagh

PS. I know I should really look at the e-toys world. But I'm a compulsive reader and wanted to get a little background on the language itself before I got in the deep cold water.

The etoys world is pretty separate from the larger Squeak world and the water is kept warm there. You can get started without worrying about the colder waters behind the dam ...

Fri Mar 26, 2004*: Assessment*

Hi Doug --

Interesting comments ....

The simplest thing I can say here is that there are now enough examples from the last 100 years or so to convince at least me that children are generally capable of much much more than most adults (and especially most schools) suppose. Thus, there is a very sad sense in which "adults are children's worst enemies" since the adults tend to control the environments in which children can learn things.

I see the Squeak etoys as trying to build on the idea that children are capable of much much more. However, I think there are many routes, including low-tech ones, in which much better assessment of what children are capable of learning can be done.

Another truly important idea about children which should be part of any learning environment, is that different children learn differently and for different reasons. Though this seems like an unremarkable observation, most learning environments do little to nothing to deal with these most important facts.

At this point in time it would be great to have either much better teaching (which includes much more understanding by teachers of the arts they are supposed to be teaching) or much better computer environments that can help children better than most adults can. Right now, we have neither, and there is great need for work and resources for both.

Sat Apr 10, 2004*: Computer as Tutor*

Hi Doug --

Al Bork is very well known in this area going all the way back to the 60s. There is a great old book called "The Computer as Tool, Tutor, and Tutee" which contains seminal papers by Bork, Papert, and others.

I couldn't find "Blowing Learning to Bits" on Amazon.

One of the original ideas about all this stuff back in the 60s was that some form of AI would develop enough to allow the computer to "understand" enough of a subject to be able to gently correct and steer. This just didn't happen. Some of the near misses (many done at CMU) are quite interesting. Plato (at the U of Illinois) was a huge system in the 60s and 70s that did a kind of tutorial on many subjects. It's worth studying, but it never got up to what Seymour and I thought would be at all reasonable.

There have been some proposals for making a tutorial interface for the Squeak Etoys that use a number of techniques to handle the detecting and gentle correction of errors. I'm hoping to get at least one of these started towards the end of the year.

It would be great to hear from people on this list just what "the computer as tutor" means to them.

Sat Apr 10, 2004*: Computer as Tutor*

Hi Gary --

Where is the furrowed brow or the cheer for comprehension in a printed book? I don't think you can get rid of the dedicated educator (and I don't want to), but I learned a very large part of what I think about from reading well written and not so well written papers and books. And, I would also say that a good book beats the average not-so-good and not-so-dedicated educator hands down if one has gotten fluent in reading and learning from prose.

So, I think there is a very important role for much better computer tutors than we now have. For example, today one could really do such an intermediary for playing a musical instrument -- especially for classical music.

An interesting setup would be to see one's human teacher about once a week and be able to practice all week with one's "practice helper". The state of the art is high for computers being able to flexibly listen to music, to follow the human player's changes of tempo, to note various kinds of phrasing, etc., and would be especially useful for practicing chamber music where the computer takes the other parts in a flexible manner.

Of course, this would not at all replace playing the piece with human players -- computers don't and won't feel music (at least not in my lifetime) -- but musicians use metronomes quite a bit of the time when they are practicing, and a flexible computer rendition of the other parts beats a metronome any time.

The reason this works for music (especially classical music) is that many (but not all) of the important goals can be characterized well enough for the computer to notice what is going on, and also quite a bit of what it means to be flexible about these goals also can be characterized. Once you decide to use it for practice and not performance, you've found a sweet spot where most of the computer involvement is overwhelmingly positive.

We can constrast this with programming (which is a bit more like creative writing). There have been several computer tutors for teaching programming, and even the best one's I've seen feel crushingly oppressive (basically like a bad teacher with Skinner box approach to teaching). In one of the earliest etoy classes with 20 children, in one of their "figure this problem out for yourself" sessions (creating a road and a car that will drive down the center of it) we got at least 7 distinct workable solutions to this, 2 of them extremely elegant.

Now, it's easy in this case to imagine a computer tutor that could watch to see if the car did indeed stay on the road, but right now, giving good advice about what the children actually did do (instead of trying to get them to do a mythical "standard good solution" (which I hate)) is beyond what anyone knows how to do with a computer tutor.

But there is one area in which a really great job can be done, and this is on some "nugget of goodness" (especially in the beginnings of learning) in which "everything is known". For example, the "Drive a Car" project is an excellent way to start learning etoys. There are about 30+ things that are learned, there are quite a variety of routes, and there are lots of known snarls that beginners need help with. Years ago there was a tutor for positional notation subtraction that really worked extremely well, and this was because the designers made a net of every possible route the kids could take and every possible bug they could encounter.

This works on a 15-30 minute project that is deemed important, but is much too much work and much more difficult in other ways for even a weeks or months long set of ideas.

So one of the things that I think would be interesting to do, and that would help people all over, would be to simply do such a brute force but nicely flexible job on "the first experience with etoys". Most people finding this stuff on the net don't have your "dedicated educator"s to ask for help, so a computer tutor that was "pretty darn good" just to get people well started would be a tremendous aid all over the world.

***Sun Apr 11, 2004**: Computer as Tutor*

Hi Diego --

Then what does the phrase "book as <fill in the blank with a word like tutor>" mean to you?

The *love* idea is really important (I loved my mentor Dave Evans), but he didn't "tutor me", he "illuminated me" on how and why to be a scientist in the service of humanity. Most of the "simple knowledge" I've picked up has come from reading, thinking and doing by myself.

But, I would really agree that for most personality types, the interest of other humans, especially adults, is one of the most important motivators for putting in effort for learning. I was already extremely interested in learning, and had been doing it for my own reasons for many years by the time I got to grad school, but still, having Dave be genuinely interested in me made a tremendous difference, especially in loosening the amount of self-criticism I was inflicting on my ideas.

So, I think the best environment involves people who really know and care, and various kinds of tools. The miracle of the book is how well it can work when one is not lucky to be around people who know and care (and even when one is). This is in part because a good writer can transmit more that just information, a lot of their basic humanity and caring glows through their words (good examples are the writings of Seymour Papert and Jerome Bruner which have affected millions of people they have never met -- these are "good guys" with "good ideas" and both shine through).

I still think that helping children to be fluent readers is the top priority in education, both because of the change of access and who can now do what, but also because of the change in how people think that comes with fluent reading.

It has been known for some time what has to be done with a computer screen to allow it to be as readable and portable as a book, and the actual technology does exist today in labs. So we are within a few years of having this next level of display be able to encompass all previous paper works without loss of legibility. However, just as there are tactile differences between vellum and paper that we've lost since printing got cheaper and superceded writing by hand, there will still be some tactile differences between books and the next generations of personal knowledge machines. The ones that are important will need to be dealt with.

But the most important thing is to discover what new special content can be manifested because we have the new medium of the computer to help us represent special ideas, and then to put forth this content so that as many people in the world as possible can think their own thoughts after "reading and doing" with it.

***Mon May 24, 2004**: very new etoys user question*

Hi Sue --

All the images in Squeak can overlap.

So they should paint their rockets separately from their backgrounds -- these should be separate objects. Then just picking up the rocket and putting it on the background painting will get the rocket "on top"..

***Tue Jun 15, 2004**: The "Triple Crown" for Alan Kay!*

Thanks very much Sheine!

***Sat Jul 24, 2004**: Video Streaming w/ Squeak Client?*

Hi Joseph --

Squeakland is for teachers and parents. Check out http://www.squeak.org which is for questions like yours. The simple answer is "yes", Squeak has a multiplatfrom mpeg player. Its best trait is that it is multiplatform. However, it is more limited than most of the commercial players, and needs to be worked on to keep pace with the commercial world.

Tue Dec 7, 2004*: spiral*

Hi Randy --

Try first with circles and changing the angle and then the distance. Note that the smaller the angle, the large the circle (the smaller the curvature of the circle). The longer the distance traveled the larger the circle. You will probably want to use the little menu in front of the property "heading" to give it more decimal places. Make variables to hold the changing quantities. Try:

forward by 5 turn by angle angle decrease by 0.1

and see what happens.

This is a constant decrease of the angle.

Try

forward by distance turn by 5 distance increase by 2

and see what happens. This is a constant increase distance.

Try accelleration in each of these examples and see what happens.

Then try a scheme in which the angle and the distance are both changed in the same script.

This should provide some insight about spirals.forward by 5

Tue Dec 7, 2004*: spiral*

To John and Randy

By the way, headingTheta is part of the vector category in etoys, has a different purpose and orientation than "heading", and is not needed for making spirals or other LOGO like geometric figures. "headingTheta" (and the rest of the vector stuff) is usually not visible in a vanilla Etoy system. How did you get it to be visible in yours?

Tue Dec 7, 2004*: spiral*

Thanks Kim --

... we should probably write up something about what they mean and how they can be used ...

Sat Dec 11, 2004*: spiral*

Hi --

Generally speaking, 5th graders get along very well without a repeat tile (but more and more older children are using etoys and thus we will include a loop construct some time this year).

But all the different kinds of loops are easy to make from two scripts, one to initialize, and one to do the loops and terminate. Use a variable if you are doing a "for" type loop.

So for player foo, "For i from 1 to 100 do mumble" would be:

```
foo loopInit
    i <- 1
    foo start script loopBody

foo loopBody
    Test foo's i > 100
       Yes   foo stop script loopBody
        No    mumble
               foo's i increase by 1
```

This is somewhat cumbersome, but is quite clear about what it does and when it does it. It has not come up as an issue with 5th graders because they stuff that we are encouraging them to do has either unbounded looping (the normal case) or the looping is stopped by some test of an external condition (as Phil suggested).

Sun Dec 12, 2004*: 2 questions - embedding text box and keystroke listener*

If you put the object to be embedded on the object which you want to be the holder, then the embed popup will show you all the layers you can embed in down to the world. If you want to be completely clear about what is going where, you can do one of two things.

. you can name the object that will be doing the holding by clicking on
the name that shows up with the halo, typing and hitting return (and this name will then show in the embed popup), or

. you can look at the red menu for the holder object and choose "open to
drag and drop". Now it will act like a playfield. The object will now capture any object dropped on it (but this can be too much of a good thing, so it is often good to turn this option off after you've done your desired embedding.

Perhaps Scott or Ned can tell us about the current state of keyboard listening in the etoy system (the feature is there but turned off, again for good reasons).

Sun Dec 12, 2004*: spiral*

Shashank --

See below ...

At 10:42 AM 12/12/2004, Shashank Date wrote:
Hello Alan,

Alan Kay wrote:

>Hi -- > >Generally speaking, 5th graders get along very well without a repeat tile >(but more and more older children are using etoys and thus we will >include a loop construct some time this year).

Great ! Yes, we were doing fine without the repeat tile so far, but now our kids (6th and 7th graders) want to do more :-)

>But all the different kinds of loops are easy to make from two scripts, >one to initialize, and one to do the loops and terminate. Use a variable >if you are doing a "for" type loop. > >So for player foo, "For i from 1 to 100 do mumble" would be: > >foo loopInit > i <- 1 >   foo start script loopBody

And here we had to add:
foo stop script loopInit

without which the clock kept ticking and the loopInit scipt kept executing over and over again. (There is an implicit infinite-loop on all the scripts).

No, there isn't. E.g. if you just say in another script
foo loopInit
this will fire this script once (this is why there is an explicit "start script" to start a script looping).

If you want to do this by hand, just click on the (!) on the left top edge instead of the clock.

***Mon Dec 13, 2004**: cannot publish a file anymore*

Hi Arvind --

Squeak has an interface to multiple external joystick ports (which could be made available in Etoys). This would be a lot better than trying to drive things around with keys. Mike Rueger is the source here.

***Mon Dec 13, 2004**: spiral*

At 04:18 PM 12/13/2004, Shashank Date wrote:
Hello Alan,

>>without which the clock kept ticking and the loopInit scipt kept >>executing over and over again. (There is an implicit infinite-loop on >>all the scripts). > > >No, there isn't. E.g. if you just say in another script >   foo loopInit >this will fire this script once (this is why there is an explicit "start >script" to start a script looping).

Oh, yes. Thanks for correcting me.

>If you want to do this by hand, just click on the (!) on the left top >edge instead of the clock.

I keep forgetting about the execute once (!) option. Is there any way to NOT have the clock icon on the script?

Why?

Alternatively, is there a more elegant solution to "stop script" regardless of which icons we click on (clock on the !)?

Why? (I don't understand your question here. Why would you want a clock on the (!)? It will only run the script once and then stops by itself.)

You can stop any script with the "stop script" command in "scripting". You can stop any ticking script by hand by clicking on the clock that is visibly showing ticking. You can do the same in a viewer. You can stop/start all scripts that are ticking by using the stop/go buttons (and these will also show you all the scripts in the environment, etc.).

***Tue Dec 14, 2004**: Script control (was Re: spiral)*

Hi --

Having a script pause for a certain number of ticks is the same as having the script run for a certain number of ticks. If you use the looping mechanism I mentioned a few emails ago, you can see that a counting loop in etoys is the same as a counting clock or timer. Each tick is one clock time, and you can keep track of how many in a variable. You can test the variable and make another script start or stop ticking when the variable gets to a certain value.

***Mon Dec 27, 2004**: Where can I find a description of make siblings*

Hi Folks --

Actually, this is all being thought through once again as we try to design and build the "Omniuser Authoring System" of which the next version of the kids' etoys system should be a subset.

I think that copy actions are too cavalier on the one hand, and classes are too top down and rigid on the other. We need something in between these two extremes. E.g. I would like to be able to "promote" an example player to being a "source" for instances. I think this will call for several UI changes, including ways to mark and find the "sources".

Ideas?

***Tue Jan 25, 2005**: Drawing lines*

Hi Gustavo --

Etoys has a preference for this: called "Batching" and "unBatching".

There is a global one that affects the world, but each playfield can be set one way or the other. Get out a playfield from supplies, go to its red handle menu, look for "playfield options" to get a submenu, and then reset the batching item.

Both of these are useful. However, it's clear that this preference should be on each line drawing object, and the set and reset should happen in the "pen use" category. We will make that change sometime this year.

***Tue Feb 1, 2005**: reflection symmetry*

Hi Randy --

I think this one is a bug (or in this case more accurately: an unimplemented feature). It should perform the way you want (and I think we should put in this feature). But there is a way to script in etoys what you are trying to do.

Tue Feb 1, 2005*: PPT presentation*

Thanks Bob!

Sun Feb 6, 2005*: selecting colors*

The color picker will pick from anywhere.

Sun Feb 6, 2005*: selecting colors*

There are several ways of accomplishing this. First, the paint box is supposed to put recently selected colors at the bottom. So the color you used for one object should be available. Also, the "paler shade" is an onion skin object that can be moved out of the way, resized, etc. There is also a preference for whether this onion skin is full screen or only part screen. In any case, with the Squeak you are using, you should be able to alt-click the onion skin to get its halo, and thus be able to move it or resize it. Also, let me know if the paint box is NOT caching recently selected colors.

Hint: this is an object oriented system in which every graphic object shares many properties. So if you can see it, you should be able to select it, get its viewer, etc. Also you can change colors (so the onion skin can be made transparent like any other object, or more opaque).

Mon Feb 7, 2005*: joining objects*

Joining is done by embedding.

Objects can be set to automatically embed dropped objects (see red halo menu). Playfields are already set that way. So a transparent playfield can hold many objects. If the objects are made sticky, then ordinary mouse clicks will not pick them up (but atl-clicks will still get the halo). If the objects are locked, then they will be indifferent to the mouse (and will have to be unlocked to be accessed). Finally, if you drop any object on any other object, you can embed it via the embed button in the red halo menu. The popup will give you choices if there are various stacked objects to choose as the holder, etc.

Sun Feb 13, 2005*: resizing objs in playfield*

Randy --

Try exploring the resources more. For example, under geometry in a viewer, you will find many properties and behaviors. One of these is called "scale". Try changing the number in there to 0.5 ... etc.

Fri Feb 25, 2005*: Some questions regarding 'supplies'*

I'll be glad to answed your questions, but for some of them I need to know what version of Squeak/Etoys you are using.

At 09:25 PM 2/24/2005, Suna Ryu wrote:
1. How can we combine two tools in supplies flap?
e.g. Text and line, or Text and rectangle..so on.

I think I understand this question. If you want to put a rectangle that contains text into a supplies flap, you first get out a rectangle object, and then a text object. Place the text object over the rectangle, and look in the text object's red dot menu. You will see "embed ...". Choose this, and then choose "rectangle". Now the text object is a part of the rectangle object, but can still be edited. Now pick up the rectangle object and drop it into the supplies bin. It will show up there in a reduced size. Now you can drag copies out. (BTW, any playfield can be converted into a supplies bin. Look in its red menu for "playfield options".

2. How can we make tables and graphs with using squeak?

This is a good thing for children to learn. Each object in Etoys has a pen, so you can construct a grapher by making a small paint dot or pulling out an ellipse from the supplies bin (make it small). Put the pen down and choose the size and color of the ink. A simple graphing script is then:

```
    dot graph  ticking
        dot's x increase by 1
        dot's y <-   <whatever value you want to graph, etc>
```

3. What is the "graph"'s function?
I can see only one graph looks like a sine or a cosine graph.

This is a different kind of graph that is used for sound. These can be made in etoys but are more cumbersome than above.

4. What is the "ruler"'s function?

We haven't used this with children.

5. I tried to use "arrow editor", but i don't know exact function of "sample".

Are you using Etoys or regular Squeak. If Squeak, then these questions are best asked on the Squeak list (you can sign up on www.squeak.org).

How can we drag out a shown arrow to the world?
I can drag out a word to the world though.
6. How can we make screen print?

Each object has an "export" in its red menu.

7. In "starsqueak", how can we use "slimemold" and "trees"?
are they existed a sort of model? or can we use them for other projects?
It seems to me that the couple of models looked like from logo

StarSqueak was derived from Resnick's StarLogo. We now use Yoshiki Ohshima's "Kedama", which is a particle system that can be programmed almost identically to the Etoys.

***Tue Mar 1, 2005**: Ubuntu Squake issue*

Hi --

The Squeakland list is for teachers and parents and children who are learning to use Etoys. I think that you might want to go to

http://www.squeak.org

to ask these technical questions about running Squeak on a version of Linux.

***Sat Mar 19, 2005**: Squeak competitor analysis (once more, please help me to improve my research)*

Hi Andre --

This could be Apples compared to Oranges.

The Squeakland etoys system is not a productivity tool for animation, but a learning environment for children.

A productivity tool -- like Flash -- would have features for for many of the elements that we wish children to create for themselves -- for example, we have the children actually write the script that does the animation, whereas this routine is built into animation productivity tools.

So it is a little more work for children to make an animation, but they wind up understanding how computers are able to animate.

This general approach is followed in the other curricula done with etoys.

***Wed Mar 23, 2005**: EToys Experience Report*

Hi Kevin --

Let me see if I can help.

There are two somewhat opposite observations about Squeak/Etoys.

First, it is a demo system aimed at 9-11 year olds for helping them understand math and science (and a little bit about computing) via investigations and authoring. So some things are quite easy and some are awkward. Later this year we plan to release a larger scale version of this kind of authoring that should fit wider needs better.

Second, there is much more that can be done with the current Etoys than easily meets the eye or is easy to find in our inadequate documentation. But a little sleuthing will help.

For example, your first category has as an example a need for collections. If you had tried one of the animation examples on the squeakland website, you would have found an object called a holder, that functions as a kind of flexible array. You would also have seen how to do these animations in The Powerful Ideas in the Classroom book, that is available on the website, etc.

You mention the need for 10 projects that are ordered. That's what the Powerful Ideas in the Classroom book contains. You said that you ordered it, so this will help.

I don't have much of a comment about teaching computer science concepts to a 9 year old (you should be able to if you want regardless of whether this is a high priority for a 9 year old's time).

From your email, I think you will have a better time with the larger scale
version to be released this year, since you have already picked up many habits from your previous experiences with computing. I think you realize that these are a double edged sword, particularly when dealing with children. If you are really trying for learning then you want an environment that is less like a productivity tool (with lots of prefabbed parts) and more like a classic tinker toy with a few parts that can be assembled in lots of ways. This should be combined with a sense of what can be really understood by the child vs recipe following. For example, an enormous number of interesting things can be done with simple addition and this will help the child get much more numerate about how numbers can be taken apart and put together.

It can be difficult for many adults (who are interested in learning things themselves) to scale back the number of tools and principles (but not the depth) to a useful level for children. We use the old Model T Ford as a motivating artifact. It was a real automobile, but anyone could take it apart and put it back together.

For example, one thing you might think about is all the ways "increase-by" can be used to model phenomena over time, from simple counting at different rates, to uniform and accellerated movement on the screen, to sampling images for animation and sound samples for synthesis, etc. All of these are easily doable by the current system. "Increase-by" is the powerful idea of a first order differential equation and it is a great "powerful idea" for a child to absorb deeply. We have found that 5th graders can easily deal with 2nd order DEs and thus be able to model accellerated motion of various kinds.

So one way to get the fullest quality from what is available is to aim at curriculum ideas that the current system is set up to handle (and hope that we can make good on our plan to release a first look at the larger scope version later this year.

***Wed Mar 23, 2005**: EToys Experience Report*

Good suggestions!

There are also some very good project write ups done by an 8th grade class in Toronto (pointer on the Squeakland site).

***Mon Mar 28, 2005**: Recursion supported in etoys script ?*

Hi Folks --

This brings up an interesting pedagogical question about recursion and objects.

Smalltalk (and hence Squeak) has full capabilities for recursion, and etoys could (and maybe should) also.

On the other hand, let's take an example, such as drawing/building a tree shape. A LOGO-like recursive approach would be to make a procedure that draws a V shape, and then write a recursive procedure that invokes this V at different sizes and angles to draw the tree.

Another, more object oriented, approach that is a little more biological is to make a branch object with a non-looping method that creates sprouting further branches (copies of itself) of the tree and gives them "life" to create limbs of their own using the sprouting method. This is much simpler to think about and do, it actually creates a tree of branch objects, and the technique generalizes much better to massively parallel structures that are more complicated than simple nested structures.

The interesting dualism here was noted when objects where invented, that because objects have a full copy of state (whereas procedures don't), there often no good reason for using an invisible stack of contexts where a visible collection of objects would better serve ....

Not that there aren't some nice and neat things that can be done with recursion, but it seems to me, where kids are concerned, there should be big payoffs for each concept that is trying to find a place in the kids' 7+-2 chunks used to think with -- and recursion doesn't make the initial cut in my opinion.

***Wed Mar 30, 2005**: Recursion supported in etoys script ?*

There is also an example in a project called "Biobob", probably on BSS somewhere ...

***Mon Apr 11, 2005**: Random*

This is why there is no parameter on the random tile. It is good for children to realize that they can get a random number in any interval by simple addition.

The random number tile generates numbers between 1 and the number that is set on it. So if we want to generate random numbers between 15 and 21, then we can do this by the sum: 14 + random 7. Often it is a good idea to make a new variable to hold such calculations:

```
                newRandom <- 14 + random 7
```

would be a typical line in a script.

In general it is good to help children to do "automatic arithmetic" whenever they see numbers. There are other places in the etoys where one might also put "a feature", but where having the children do a simple relationship from scratch is better for them.

***Tue Apr 12, 2005**: Random*

Ah, now I understand what you were asking! Good question!

I'll ask Scott Wallace for his opinion here also.

I think the simplest answer is that this should be like the feature "forward" (which can be made from simpler stuff in etoys but is not worth putting the children through the process early on -- so it is provided as a feature). That we can't simply drop a variable name on Random (and extend it for arithmetic) is a bug and should be fixed in this version of etoys or the next.

It is possible (and illuminating) to make a random number generator by hand in etoys, but I don't think this should be an early burden on 11 year olds!

What do you think Scott?

***Wed Apr 13, 2005**: (no subject)*

Hi Aimee --

Not to put you off here, but this list is for the etoys that are aimed at 10 year olds, and my guess is that you are not trying to use that, but Squeak itself (which is aimed at fairly sophisticated programmers and has its own website http://www.squeak.org).

However, I think I remember that Dan Ingalls, one of the creators of Squeak, did the Crostic stuff, and I'm copying him here.

***Wed May 4, 2005**: Beethoven*

This can be done by finding a MIDI score on the net, playing it with the Squeak MIDI player, and then using it with the clink sound on all of the parts.

There used to be some excellent timbres around for the Squeak player. We should put together another batch. Volunteers could look at the net to see about free sampled timbres. Should be quite a few that we can use.

***Thu Jun 2, 2005**: clear object-specific pen trails*

Hi Folks --

You know, in Tweak, we should probably just extrude real objects for pen trails (or maybe paint a separate object ....)

***Fri Jun 3, 2005**: clear object-specific pen trails*

Tweak is the next version of etoys, and will have considerably more range of users and use. We have been working on this for the last several years.

***Fri Jun 3, 2005**: clear object-specific pen trails*

Suggestions for better UI would be most welcome.

***Mon Jun 20, 2005**: Multiplayer network game programming with Squeak*

Also look at www.opencroquet.org

I haven't tried the Etoys screen sharing stuff recently, but it could be used for simple shared Etoy games (it works similarly to Timbuktu, but has "badges" to make rendezvous easier).

***Mon Jul 18, 2005**: random numbers in etoys*

If the variable is called "car's foo", the min is 6 and the max is 17, then how about

car's foo <- 6 + random 17

?

I'm not sure what version you are using, but you can find the random tile in every scriptor's menu, and very often there is one in the Supplies bin.

***Thu Aug 4, 2005**: Squeak Etoys and preschoolers*

Hi Leonel --

We have done little in this age group with Etoys.

The UI should be changed for this age children in a number of important ways (some of this was in the original Etoys design but didn't get implemented).

You are probably aware of Radia Perlman's work in the 70s at MIT with a "button box LOGO"? This was pretty interesting, and we duplicated her equipment and did many parallel experiments. I have heard recently about very young children and Etoys (some of them will be at SqueakFest next week) but I don't know the details. Kim Rose might. We also have a grad student working with us who is interested in this age group.

***Mon Aug 8, 2005**: make sibling from script*

Hi Randy --

Just put the copy in a variable, and then you can tell it what location to have ....

***Mon Aug 15, 2005**: Pausing animation?*

Not an accident, heh heh ... All fractions work here. This is used in the Sound Synthesis project to get all the pitches.

***Wed Aug 31, 2005**: 1st Grade*

They love illusions. Get an illusion book and think of ways to make simple projects for them to do.

For example, the Mach illusion has two a large white square next to a large black one. Two smaller same gray squares are placed in the middle of the larger ones. The grays suddenly seem quite different. Etoy version can use copying. Start with a rectangle, make it into a large square, copy it and then use the color tool to make one white and one black. Copy another one, make it smaller, and color it gray. Copy it. Put the small squares in the middle of the large ones and you will see the perceptual difference even though the two gray squares were copies. Now write a script for one of the gray squares to move horizontally at some speed and use "bounce" to reflect. Make a similar script for the other gray square with a different speed. Start them moving. They will intersect with each other and prove they are the same color, and they will move into the larger squares and seem to be different colors.

Another one for 6 year olds is the camouflage one that is on squeakland, with a grasshopper and grass. Make the hopper and grass the same color and the grasshopper will be hard to see unless it is moved. This works for fish, etc.

***Mon Oct 10, 2005**: Question about mouse on Windows*

Another option is to have "mouse over" bring up the halo. Many elementary schools use this.

***Tue Oct 18, 2005**: a question about siblings*

Hi Randy --

The idea of siblings is to make parametrically similar objects. So it's an interesting question just which parameter values should be transmitted or remain distinct. If the "graphic" is repainted on a sibling, we certainly don't want the change to be automatically propagated to the rest (any more than we want the position or heading be automatically transmitted). OTOH, we might want to selectively transmit some of the values from one object to all. E.g. we might want to say "let all my siblings have my graphic".

Right now we are pretty much restricting this to control state rather than property values ....

***Thu Oct 27, 2005**: Folder of friends?*

Squeak has a Timbuktu-like screen sharing facility with a chat and VOIP interface, and "badges" that can hold the pictures and addresses of others who are using Squeak on the net. It works and has been demoed many times, but we've never been satisfied with it for a number of reasons, and are currently in the process of replacing this using the mechanisms of Croquet (http://opencroquet.org which is much better supported, deals with transmitting events rather than screen bits, can be used on very low bandwidth connections, etc.)

My advice would be to treat this as a currently unsupported feature and not try to use it.P.S. A more supported use of networks in Etoys is Netmorphs, which logically connect up remote PCs using Squeak so each screen is a tile in a much larger virtual screen. So, e.g., a child can make a car and drive it off the screen and it will appear on someone else's computer screen. Here is a recent message from the author, Umezawa-san:

From: Masashi Umezawa <umejava at mars.dti.ne.jp>

Hello all,

NetMorph 0.3 for Squeakland 05 is finally available.

Pre-installed image:
ftp://swikis.ddo.jp/NetMorph/demoImages/NetMorph-Sq05.zip

How to use:
http://swikis.ddo.jp/NetMorph/12

This is the first official version for Squeakland 05 image.

If you found some bugs, please send a report to me!

Enjoy! --- [:masashi | ^umezawa]

***Thu Oct 27, 2005**: Copying from one project to another?*

I thought you had already asked this...

Any object can be copied by using the red halo menu. In the 2nd project, use the "new morph ..." menu item in the world menu.

BTW, pretty much every feature you have asked about so far are things we've put in the system to experiment with but are not generally used with children.

***Sun Oct 30, 2005**: Question about Powerful Ideas Project 5*

Hi Mark --

Every Etoy object has a pen (they are all "turtles with costumes"). So instead of having an opaque plotting feature, we just have the kids use a small object (like an ellipse), put the pen down and write a ticking script like

```
ellipse's x increase by 1
ellipse's y <- <the value to be plotted>
```

This is nice and simple, and also makes plotting understandable.

***Mon Oct 31, 2005**: player variable*

Hi Randy --

At 10:57 AM 10/30/2005, Bert Freudenberg wrote:

Am 30.10.2005 um 16:33 schrieb Randy Heiland:

> *When I have a variable that's of value type 'Player', is it possible*
> *to get a 'handle' to that player/object and use its tiles in a script?*

You have to built the script using the tiles from a concrete player, and later substitute the references to that player in the script with your variable.

And this is pretty ugly from a UI perspective. This is why we don't really mention it. But it does work. Ted Kaehler has been working on a much better UI for complex expression building with tiles in Tweak.

***Wed Nov 23, 2005**: un-embedding*

The black handle will "lift" out of an embedding.

***Thu Nov 24, 2005**: How can I use Bouncing Atoms?*

Hi --

Bouncing atoms was a long long ago hack, But look in the red halo menu ...

A better start with particle systems in Squeak Etoys are the Kedama objects by Yoshiki Ohshima, which are an Etoy slant on Mitchel Resnick's starLogo.

Linda Kao did some terrific documentation for Kedama at:
http://www.squeakalpha.org/fun_projects/kedama/kedma_welcome.htm

---

***Sun Dec 4, 2005**: feature request: color selection*

This used to happen. Looks like a broken feature to me.

***Wed Dec 7, 2005**: Squeak & Mobile Phones*

I think Yoshiki Ohshima did do a port to a mobile phone a few years ago. The current screen sizes are not very good for development ....

***Thu Dec 8, 2005**: resizing doesn't stick in holder animation*

Instead of resizing the drawings in the holder, try resizing the player that is changing its costumes. I.e. if the script looks like:

holder's cursor increase by 1 ball look like holder's player at cursor

then resize "ball". Whatever scaling ball has is applied to its costumes.

***Fri Dec 9, 2005**: resizing doesn't stick in holder animation*

What happens if you change the size of the ball (or change its scale factor)?

***Tue Dec 13, 2005**: multiple authors?*

The chat badges are actually functional, but they are "not civilized". I've been giving demos for years using them. And they do much more, they can also do real time screen sharing with multiple cursors. There is a form of VOIP that can be used to talk back and forth.

But this is not currently a "supported feature". There is not a good UI for setting up the badges. And the badges need the IP address rather than a name in a registry, etc. We expect to have a much more flexible and supported version of these features by the end of 2006.

The second paragraph indicates that your students might be working in Squeak rather than in the children's etoys system which is on top of Squeak. If so you should probably go to squeak.org instead of using squeakland for these kinds of questions. Squeak has an extensive development system, and there are some interesting packages for it, including one by Umezawa-san in Japan that is set up for joint development over a network.

For the ultra version of all this (and a peek into the future a few years from now) look at http://opencroquet.org .

***Wed Dec 14, 2005**: multiple authors?*

HI Bob --

The Etoys are at their best for 5th graders. Our experience with 8th graders is that they can do most of the existing 5th grade centric curriculum in 6 weeks or so. And then they need a superset of either or both of curriculum and what the language can do. We've been working on the supersets for a while (and are a little behind our plans as to when they will be supported.)

But now I'm curious as to the development model you'd like to use. If it is ABABABABA then the kids can simply email the existing project back and forth or store it on the web (there are even swikis (e.g. from GaTech) that allow projects to be uploaded and downloaded).

If you are talking about kids independently working on the "same" project at the "same time", then they are either working on different objects -- in which case you would like to merge the changes -- or they might be modifying the same objects (more likely some of the time) -- in which case forking is really going on and the merge is not trivial.

I'd like to hear how you think this would work with your kids.

In the latter case of forking it is probably more useful for them to be sharing the very same single project so the changes are going in directly and the clashes will be quick and usually obvious. The "Nebraska" sharing mechanism (through the badges) *can* do this since it is essentially a Timbuktu like client-server on the same project on one of the machines. It might be interesting to try some experiments to see if this works.

"In the future" this will be solved using the undermechanisms of the Croquet system (http://opencroquet.org)operating through the superset.

***Thu Dec 15, 2005**: multiple authors?*

It needs a better interface and a name server, etc.

***Tue Feb 14, 2006**: generate a movie?*

But, if I read the original question -- it seems that it is about making movies that are playable in Squeak from other formats, including those that cameras use.

There are a variety of SW apps that can do this, and some of them are free, depending on which platform you have.

***Tue Feb 14, 2006**: generate a movie?*

Oh ... Well, Takashi Yamamiya has done this, and I think he did use Wink.

***Fri Feb 17, 2006**: feature request: rotation center help*

For older children, you should do real inverse square orbits. Squeak Etoys does these very well. Don't forget that you can choose to have the "origin at center" in any playfield. This means that any player put in a playfield will also act as a vector, and this can make it very easy to model 2D accelleration, velocity and position. (There is also a vector vocabulary for players which can be turned on to allow simple + and - vector arithmetic to be done between players, etc.)

A fun thing to do (especially since it is intractable with classical math) is to do the 3 to n body calculations. Turn the pen on for the third body and you get beautiful chaotic curves, etc.

Also, when using embedding, think about what it means for each player to be a coordinate system for the players embedded in it.

***Fri Mar 3, 2006**: Copy an object programmatically.*

Also consider that you can put objects into a playfield both by hand and by program and clear the playfield.

I usually put my seed object into a separate little playfield before doing this.

One way to look at this is that the playfields act as variables for single objects and sets of objects. (If you put 0 at center of a playfield and drop in a player, then you have a very nice visible representation of a vector -- and there is a vector arithmetic trait that you can turn on, etc.) 03:56 AM 3/3/2006, Randy Heiland wrote:

***Fri Mar 3, 2006**: graduated fill*

Hi --

07:44 AM 3/3/2006, Sholom Eisenstat wrote:

With some experimenting in class, we've determined that a morph created with the draw tools does not have the same set of color/border tiles that a morth grabbed from the supplies tab has. The latter's tiles enable the fill stuff whereas the former's don't.

So, that appears to be the end of that thread though I don't understand why the difference is programmed in as such.

Yep, it shouldn't be. The plan for the new system is for everything 2D to be made from polygon/curves, and that bit-map painting will essentially be a texture. We have also been experimenting with freehand painting that results in vector graphics objects on the fly ...

Tue Mar 28, 2006*: Linking projects*

The link that Randy points to below discusses how to use Squeak Etoy projects as "better than Powerpoint" slides.

But there are three other ways to make links to projects. As usual, the UI could combine this better .... there could be better documentation ...

1. Look at the submenu New Morph from the world menu. The last item is Make link to Project. This will give you a list of all the projects in your image and choosing from this list will give you a thumbnail of the project that can be resized and used.

2. Any player can be a button. Go to "Extras" in the World menu and choose "Mouse Up Action". In here you can type any Squeak expression. The one you want here to e.g. go to a project named Gas Tank is: (Project named: 'Gas Tank') enter: false. This will produce a surrounding rollover frame and will take you to the named project on mouse up.

3. You can take any text object and choose part of it to make a hyperlink. Suppose you have "go here to see this" and you want to sensitize "here". Type "go here<(Project named: 'Gas Tank') enter: false> to see this". Select "here<(Project named: 'Gas Tank') enter: false>". Alt (or control if you are on a Mac) 6 will bring up a menu for text options. "Choose Do It". The result will be "go here to see this" with the "here" colored blue, pressing on this will execute the Squeak expression "(Project named: 'Gas Tank') enter: false" and will take you there.

Sun Apr 9, 2006*: I want to introduce squeakland to my Chinese friends*

Hi Jim --

Squeak runs exactly the same on more than 25 platforms, including MS.

Sun Apr 9, 2006*: I want to introduce squeakland to my Chinese friends*

Oh,.... well, we have the same problem in the US at very deep levels ....

Sun Apr 23, 2006*: I want to introduce squeakland to my Chinese friends*

Hi Milan --

Yes, what you describe is what I've called the "driver's ed" (DE) view of computing -- and this goes back at least as far as the "Nation At Risk" manifesto in (I think) 1983. It's the simplest way for school people and parents to feel they are doing something modern and relevant with computing.

The kind of computing that Seymour and (a few years later) I have been espousing since the 60s is in the same epistemological camp as real math and real science -- and most school people and parents don't understand what these are and why they are important.

I think people who are interested in Seymour's insights will have a simpler time if they just lump real math, real science and "Seymour Computing" (I'll call this s-comp) into one composite subject that is not associated with DE-computing. My generic term for this would be "real science" -- the reason for this is that "school math" has been aimed at simple arithmetic (the "driver's ed" of math) and there are now huge schooling standards and testing for this, just as with DE-computing.

Science is a little more vague for most people (and a little scary for others) so there is much less force behind standards and testing right now. This allows much more of the real stuff to be done (and combined with r-math and s-comp) if we could get parents and teachers to understand it better.

So I would advise focusing on r-science as a way to help teach children thinking (and debugging of thinking) and powerful ideas and ways to represent them (including r-math and its sibling s-comp).

Cheers

Alan

Sun Apr 23, 2006*: I want to introduce squeakland to my Chinese friends*

Hi Bob --

At 08:38 AM 4/23/2006, Robert Parks wrote:
Is anyone in the Squeak community developing tools for language literacy.

We should be. One of the more interesting and early attempts at this was an Apple II program designed by Chris Cerf (son of Bennett, working with Sesame Street at the time). It was a sentence maker for young children and each sentence was then carried out by animated figures. We thought a little about this when we designed the scripting language for Etoys, but never got around to making the very young child's version. This has come up again wrt to the 100$ laptop (where it would be a very useful part of "scripting as math".

I have developed a children's dictionary (http://www.wordsmyth.net) and
am working on an early literacy dictionary. In particular, I'm

interested in the intersection of tools for programming with variants of controlled English, and tools for teaching reading and writing.

Any good sources for "controlled English for children"?

Teaching the "debugging of thinking" would be easier if we started with the core tool - language.

More like vice versa. Human language is wrapped in metaphor and allegory, and it has been shown that most people have as little sense of how imprecise they are in language as they do of what grammatical components they are using. One of the reasons math notation moved away from attempts at careful use of language (for a good example of the "before" see Newton's Principia which is pre-algebraic) was that the way meaning has to be inferred from math is quite different that ordinary use of language. Scripting is somewhere in between. Many people tried to use Hypertalk in an imprecise way (like their normal use of language), but there was some anecdotal evidence that they learned to get more precise in both Hypertalk and normal language as they did more scripting.

I'm reminded of one of the first Apple II programs for children - Rocky's boots.

One of my favorites of all time.

It involved creation of a logic circuit for distinguishing shapes and colors. Too bad it wasn't developed further to find other areas for applying core logical concepts in the context of analogical reasoning.

Actually it was, in it's follow on "Robot Odyssey".

More is needed here.

***Sun Apr 23, 2006**: Looking for good souls*

Hi Markus --

I don't think this would be a great idea. Squeakland is explicitly for teachers and parents (and they are very shy as it is). I think it would be very confusing to convolve discussions about Etoys with discussions about Squeak (especially since most of these users think Etoys is Squeak).

Let's set up a separate list please.

***Mon Apr 24, 2006**: Looking for good souls*

How about Logo? It had a real vogue in the UK some years back.

Cheers, Alan

***Tue May 16, 2006**: TinLizzie wiki*

Hi --

Good prowling.

We plan to make this new plugin available sometime this summer -- right now it is "experimental".

The current link to this on the LogoWiki site says:

"TinLizzie" is a WYSIWYG wiki that implements Etoys using a special document format (and it needs a plugin that is not yet generally distributed). To test it go here.

I put this link in to have a nice jumping off place for some demos I'm giving over the next few months.

TinLizzie is essentially a re-engineering of important parts of the Etoy environment to try to put together a more efficient and accessible architecture for the $100 laptop project, and a big yet to be started educational project in South Africa.

***Wed May 24, 2006**: Etoys: The Good, the Bad, and the Ugly (was Re: Whither Squeak?)*

Hi Markus --

Good list!

But a better way to look at Etoys is that it was supposed to be a subset of a more comprehensive system for all programmers (as comprehensive as Smalltalk but a more advanced set of ideas). So, if we were to make a better system and do a special children's interface for part of it, that would be a better way to deal with "bad and ugly".

***Sun Jun 25, 2006**: Question about how to run an animation only once*

Hi --

The basic idea is that animation in Etoys is something for the children to learn how to program it (it's one of many uses of the powerful idea "increase by"). So it's not a "feature" or "productivity tool". If you look in the "collections" role of a holder, you will find a property "Holder's count" which keeps track of the number of elements that are in the particular holder. You can test this against the "Holder's cursor" position to see if you have gone through all the contents of the holder. The control of scripts is found under the role "scripting".animateOnce.jpeg

---

***Fri Jul 7, 2006**: Can EToys Teach Me How to Program in Squeak?*

Hi Greg --

Please tell me more about your aspirations. There are a number of styles of programming, and there are a number of programming languages, each of which addresses one or more styles.

Squeak Etoys is a style that we made up based on 35 years of experience working with children. We have had very good results with 8-12 year olds over the last 10 years, and this has accounted for its spread around the world. If you did a few things in Etoys, you would be (a) programming, and (b) get some of the feel of being able to make dynamic constructions via programming (c) be learning a few things that would transfer to other programming languages (the overlap is not large though).

I strongly suggest that you get the book "Powerful Ideas in the Classroom" by B-J Allen-Conn and Kim Rose (available through the website or at Amazon). This plus other materials on the website should help get you launched.

Squeak is an open source version of the Xerox PARC Smalltalk (from the late 70s) that we made as a general tool for constructing large scale designs. It is very powerful, but the introductions are certainly more geeky than you might like.

Tue Jul 11, 2006*: Computer Language Definitions and Intelligibility*

Hi Greg --

Here's how I think of "extreme late-binding".

Suppose you are working on something and at some point you realize that you'd like to change it or some part of it. If you can do that pretty easily, then that change would be called "late-binding". If you can't do it easily, then whatever it was would be called "early-bound". Some materials are erasable and some not. Some forms of images are more erasable and changeable on a computer than on standard physical media, etc.: they are more "late-bound".

For example, many operating systems and applications will require you to reboot your machine after certain kinds of updates. (I just got a new MacBook and was surprised that after it updated its apps from the net that it required a reboot -- this would be more expected on Windows than on a Mac.) These updates were "early-bound" and something major had to be done to rebind things.

Lots of programming and apps 40 years ago were early bound (one had to go back to source text, recompile, reload, rerun, etc.). Interactive systems started to try to late-bind as much as possible, so that a change by a user would be immediately reflected. Compare the late-bound changing of a picture in a graphics app or some text in a word processor or a Hypercard script to the to the much more tedious task of using a blog or wiki which requires the text to be typed one way, and only later do you see how it turned out. That this situation obtains today in the web is terrible (most especially since there is no good reason for this, just really bad design by the people who did the web browsers).

Lisp was one of the first programming languages to experiment with late-binding much more than had previously been done. And we took this up as one of our goals at Xerox PARC in the 70s: to see how much you could allow to be changed on the fly without killing the entire system. The Smalltalks went rather far in this direction (and could go further). (Squeak is a Smalltalk.)

For kids, we wanted them to have instant feedback on everything they did, so we took the Hypercard model and tried to remove its various modes, enrich the graphical landscape, and simplify the programming. We aimed at 8-12 year olds, and Etoys works pretty well for them.

Etoys was a demo that was supposed to be reimplemented as a wider ranging system for children from about age 5 into high school. But this didn't happen, and the result is that Etoys remains mostly useful for the original age group. For example, it would be pretty frustrating for you to try your project in Etoys.

Squeak on the other hand is a full blown programming environment (like Java) and your project could definitely be done in it). But it is much less suited to the kind of user you say you are. (I think you sell yourself short a bit because anyone with a good command of writing skills -- and you certainly have these -- can learn to program in the general non-iconic forms used today.)

The biggest problem in programming is not so much the strange seeming nature of the raw materials, but that as things scale up, architecture dominates the materials. I.e. design starts to become more and more of a factor. And design is not learned in a day, even with the best materials and environment. The very best programmers and computer scientists I know -- who have absolutely no problems with raw materials -- still have great difficulties with design for most systems that are worth doing. This is one of the reasons we like to make things late-bound: we don't know what we are doing half the time, and are constantly finding out things that we needed to know earlier.

One analogy (that might be unsatisfying) is that many people have complained about the ad hoc nature of standard musical notation and of the layout of the piano keyboard (which leads to lots of scale patterns, etc.). And, it's true they are a pain when starting, and do turn lots of beginners away. Many suggestions have been made to improve both of these.

Once one gets into the stuff, one realizes right away that real fluency doesn't depend much on the actual notation or keyboard layout. This is because fluency in the human brain is done by flattening structures into thousands of special cases. There are real similarities here to reading and spelling. It helps to have phonetic spellings in the beginning, but they are completely bypassed by fluent readers.

In the case of designing computer stuff, there really isn't enough of a body of great design yet to provide thousands of applicable patterns, and so even seasoned professionals tend to flounder. And, again, better late-binding of everything (extreme late-binding) really helps us flounder our way towards some of our goals.

Mon Oct 30, 2006*: Test if script is running ?*

Both of you are right ... it would be a good addition to have an Etoy test for the status of a script.

Tue Oct 31, 2006*: Lack of documentation frustrating*

Hi --

Yep, we should have more complete documentation in English, especially on the media objects. (E.g. the Japanese and Spanish documentation includes translations of ours, plus some very good stuff done in those languages in those countries.)

But the book "Powerful Ideas in the Classroom" is still a very good idea as a starting point because most good Etoys are thought up as a way to help children learn an idea (and generally not as a way to learn a particular programming technique -- the programming style in Etoys is different).

The tradeoffs are interesting. On the one hand it is pretty easy to get 7th and 8th graders to do a dynamic ecology of fish and food sources from scratch, whereas the AP version of this in high school in Java is deemed difficult enough that the high schoolers don't get to program it but are giving the programs to study and change parameters on.

Etoys is all about children being able to make fun working versions of interesting ideas from scratch, and learning much more about the ideas than when force-fed with them. Considerable thought on the part of the children's mentors is often required to set up a curriculum that is a nice balance between the way children think and do, the ideas, and what is most natural to do in Etoys.

Basically, every object in Etoys "is the same" for most things, plus, for some objects (such as playfields, etc.) there will be a few extra categories in their viewers for idiosyncratic behaviors. Just poking around and trying things (which is what the kids do) will reveal a lot.

For example, the "world" (the desktop object) is a kind of playfield, and all playfields have a "playfield" category, and in this category are variables that hold the mouse coordinates relative to that playfield.

Every script can do a loop in parallel with other scripts, so there are an unlimited number of parallel behaviors possible. These scripts can be controlled by other scripts (look in the "scripting" category for each object). Typical loops can be done by initializing in one script and then telling another script to start ticking (and this script can have a test tile that can decide if the loop should stop, etc.).

A little context: Etoys is kind of a "demo that wouldn't die". It was originally aimed for a particular age of children (from about 7-8 to about 11-12) to be an authoring medium for fun projects that could have an underlying "powerful idea" or two, that could be absorbed Montessori style. So the goal was not to teach anything like standard programming, but to make it easy for children to e.g. use and learn the ideas of vectors, calculus, feedback, systems ecologies, media models, etc., while pursuing projects that seemed fun to them.

Human beings (even really smart ones) have a hard time coming up with ideas that are better than mediocre. For example, if you put a piano in a classroom, the children will explore it, and develop a "chopsticks culture" with it, but they won't invent for themselves how to play a keyboard instrument (it took centuries for experts to work it out). But every child can be taught to play the piano. Similarly, the children will not invent or discover important ideas in mathematics by themselves. But every child can be taught a powerful version of the calculus of vectors, and many other kinds of advanced mathematics. And both of these can be taught as a kind of play.

If you give children a medium to explore, they will generally wind up doing stories and games with it (in large part because that is how nature has set all of us up to learn when we are children). For example, Etoys is used widely in a number of places in the world. The places that emphasize "creativity", "discovery learning", "free exploration", etc., all wind up with lots of stuff done by children, but virtually all of it uses simple animations and multiple tasking to act out stories and games. This is no surprise (it took humans 100,000 years to invent math and another 2000 to invent science). If we are interested in having children learn non-obvious powerful ideas -- e.g. in math and science -- we have to scaffold their learning and discovery by careful curriculum design.

This teaching doesn't have to feel like the kids are being put in a lock-step chain gang. It can be much more like teaching and learning an established sport or musical instrument. There are parts that are almost impossible to invent, and thus have to be shown and practised. But with these parts there are large elements of free joyful play.

We suggest using at least 3 phases for each idea. - The first is a guided creation of something interesting -- for example, how to make a robot vehicle on the screen that will follow edges. This can be done in a number of ways including Socratic leading questions, but basically it is giving the children something they would not think up for themselves. But as David Ausubel pointed out "People learn on the fringes of what they know". - Now that the children know something, they can be given a specific challenge -- such as "Come up with a car and a road where the car will stay on the road". There are 5 or 6 ways of doing this and most children working singly or in pairs will find one of them. A few of these are elegant, and a few children will find these. Sharing the solutions as demos gives the children a sense that such problems are not only solvable, but there is more than one solution. - The third stage is open play, where the children now know enough to think of many different fun ways to use what they've just done (and many of their ideas will be in the forms of games or stories). For example, some of the "middle of the road" solutions lend themselves to making a multilane racing track with multiple vehicles and using the random number tile to generate random speeds to make the race difficult to predict.

The way we've set up Etoys is with a uniform rich object model and a very simple set of scripting abilities, but with easy multiple tasking. From this we've asked ourselves what projects that involve powerful ideas can be relatively easily made from the simple ingredients. There are lots, and they fill more than a school year's worth of time. This is why the project based documentation is not such a bad idea. It's worth while to look at the kinds of things that have been done with the current ingredients, and this will help with the different style of programming that is used.

However, children younger than 7 really need a somewhat different interface. And children older than 11 need more ingredients (both scripting features -- such as case based control structures, better event structures, etc. -- and expression building features -- e.g. to more easily build complex expressions from left to right instead of just from the top down, etc.). We are going to do the latter over the next year, and the former a year after that. But for (say) 5th graders, the current set of materials seems rich enough (for powerful idea purposes).

The documentation is going to get a little more useful and detailed because Etoys will be on the "$100 Laptop" project of the One Laptop Per Child organization (http://www.laptop.org) . The test builds of this machine are just starting to happen, and we are starting to write more detailed documentation on the OLPC wiki (http://wiki.laptop.org/go/Sugar_EToys). This is not worth looking at today, but should have quite a bit of useful stuff a few weeks from now.

Please don't hesitate to ask more questions. We are happy to help.

Thu Nov 9, 2006*: Making a Copy of an eToy from a script and using the mouse pointer to activate an eToy script*

Hi --

First, how old are these children?

Tue Nov 14, 2006*: Making a Copy of an eToy from a script and using the mouse pointer to activate an eToy script*

Hi Offray --

Etoys was designed for 9-11 year olds, and it has worked very well with them. However, even 8th graders are able to use more features, and high schoolers should be using a pretty complete programming environment (like Squeak).

You are not the only one to want to do more with Etoys, and I think this is because the basic notion of Etoys' universal objects, easy scripting and multitasking are things every programming environment needs. In "a year or so" we will have something more like a full spectrum system that can be used by a much wider range of users.

Squeak itself is very easy -- as Tony Hoare once said about Algol, Smalltalk was a great improvement, especially with its successors! -- the problem for beginners is that the library has only one organization, which kind of lumps everything together, and this is a large lump. This is powerful but can be quite frustrating (other comprehensive systems have similar problems). Etoys is kind of a demo that quite a bit can be done with much much less in the end-user's world, but the aim at 10 year olds limits its range. Our original plan was to wind up with a more Hypercard-like system, still small and simple, but much more comprehensive.

But, since Squeak is "live" and includes all of its parts and tools written in itself, much can be done with the existing system. For example, it is quite possible to make up an end user view of a system by making a special code browser for it that would only show the relevant useful classes. Morphic itself was designed to be quite simple and to allow quick naive presentations driven by Squeak methods. Etc. E.g. instead of trying to make the tile scripting system do things it wasn't designed to do, you might look at how morphs and players could be programmed in the Squeak code browser, etc.

This would produce a system with a simple and easy to learn syntax, huge expressive power, and a number of useful graphics classes to do things with on the screen. Not the very best thing that could be done for high schoolers, but would still allow a lot to be learned without huge expeditions into the library.

Mon Dec 11, 2006*: Help needed on reply about squeak*

Hi --

We could use lots more documentation in lots more languages most certainly. It would be nice to have all the writing systems of the world available and usable (the OLPC machine will probably force us to do that).

I think he missed the book "Powerful Ideas in the Classroom" which would give him a start with his daughter (and for himself). Etoys is not at all about widgets, just the opposite. He also seems to have missed the tutorials that are on the website.

I don't understand the comment, "squares aren't resizable".

He should be encouraged to try a little harder.

Tue Jan 9, 2007*: Bouncing?*

Well ...

Perhaps what should happen is that when you drag the "bounce" tile line to the desktop it should show its meaning as an Etoy script.

Bounce is more or less writable directly in Etoys so it would be easy for a user to see just what it does and how. Interesting question would be what to expose about "forward" in order to make this happen. Probably the Snell's law way of looking at things is the most intuitive for end-users.

It could also be renamed to avoid bad guesses about what it might do (but simply showing an Etoy version of it would be good).

The simplest thing to do would be for this scriptor to resist editing and just tell the user so -- or, it could even allow itself to be edited (with the modifications running much slower than the internal def of bounce did).

Mon Jan 15, 2007*: e-toys project*

Hi --

Etoys uses 2D graphics. Are you in 3D? If so, you aren't in Etoys. Can you tell me how you got to where you are?

Wed Feb 7, 2007*: Squeak fails to run after install: security problems?*

Hi Simon --

Just for a little context ...

The Squeak you have been using (probably 3.6 or so) has a complete Etoys in it. It might still be the most useful vehicle for what you are trying to do. A shift-alt on any graphical object will bring up an Etoys halo of handles, and the blue eyeball will open a viewer for that Etoy player. Dragging out a behavior tile (like "forward 5") onto the desktop will make a script and put the "forward 5" into the script.

You can also get an Etoys developer image. This, again, is the regular Squeak with certain preferences set. Let us know if you are interested.

The Etoys version (from Squeakland) is aimed mostly at ages 9-12, but has worked well for younger children, and to a lesser extent for older children (they could use a few more facilities). Many teachers (especially non-technical teachers) have enjoyed using EToys.

Etoys and Squeak have no external security models, so the difficulties you are encountering are solely due to some combination of MS and the sysAdmins in your shop, plus where the Squeakland installer puts things so it can be run as a plugin.

Because the Squeakland version is also set up to be able to run as a plugin, it is sandboxed for safety. It is possible that you would like more control over things. Basically, the Squeakland version is pretty much a regular Squeak with a number of preferences set to limit the view that the end-user takes of what's available. You may very well want to relax these preferences. Let us know and we'll explain how to do it.

Basically, to take more control, you need to gather three or four files:
 - the image, which contains the system and the objects
 - the VM, which contains the equivalent of the Squeak OS and the interpreter, graphics kernel, sound, sockets, etc.
 - the sources, which contains the indexed text of all the source code in the system
 - the changes, which contains the incremental additions that have been made to the system and allows the Squeak developer to revert to earlier versions, etc. The Squeakland image is set to not write to changes, but this can be changed.

If you drop the image on the VM, then Squeak will start up. MS can also be told to use a particular VM as a default for a double click on an image.

You should be familiar with these already from your experience with Squeak.

We are interested in what you are trying to do and would like to help. Please ask.

***Mon Feb 26, 2007**: Doing a presentation of Squeak with Squeak as a presentation tool*

Hi Hilaire --

Yes, I have given every one of my presentations since about 1998 only using Squeak. And, I mostly use the ThreadNavigator to sequence through sorted projects. This allows me to have an image with many projects in them and to make up a thread for each talk. Sometimes I will use a BookMorph in a project.

I don't use templates, but templating should be there in a more complete way than it is. Scott Wallace did quite a bit of work many years ago to investigate better template mechanisms that would do quite a bit more than Hypercard, but they only work in BookMorphs. We had planned to merge BookMorph pages and Projects but the lack of funds after 2001 prevented it. Your method of saving a template as a project and reloaded is the best way to go now.

It is not possible to merge two projects (and it's not completely clear what that would mean in all cases). Why not just have the 3rd party project be one of the projects in your presentation and go to it in sequence? That's what I do.

In any case, now that we have a little more funding, we are working on a better project architecture which will at least have a good template mechanism and (probably) allow very simple merging.

***Mon Feb 26, 2007**: Doing a presentation of Squeak with Squeak as a presentation tool*

Please let me know any questions you have about talks in Etoys, the thread navigator, etc. We should document these (and actually are in the process for OLPC -- a good forcing function for many things).

***Mon Feb 26, 2007**: Squeakland Digest, Vol 46, Issue 11*

Thanks Valerie --

Yes, and something like this "should" also be a simple feature of Projects as well ... (maybe this year ...)

***Tue Feb 27, 2007**: Squeakland Digest, Vol 46, Issue 11*

The OLPC version of EToys (and I think the Squeakland version -- Scott will correct me if I'm wrong) has an option in BookMorphs (the dot menu) that will make a new page as a "scripting area". This sets up a little name space to make it more convenient to have a number of separate scriptign regimes on different pages (so they are a little more like a project). However, what we really need to do is to unify projects and pages (and hope to this year).

***Sat Mar 3, 2007**: How to learn squeak*

Hi -

What kind of educational material are you interested in making?

Squeakland is set up for children who are 8-12 years old.

There are tutuorials, examples, etc.

Take a look at http://www.squeakland.org/content/articles/attach/etoys_n_learning.pdf for a white paper about the educational approach.

Take a look at http://www.squeakland.org/content/articles/attach/etoys_n_authoring.pdf for a white paper about the media approach

There is a book for teachers: Powerful Ideas in the Classroom.

http://www.squeak.org is the main site for Squeak for adult computer folks. You will find quite a bit of stuff there.

Sat Mar 3, 2007*: E-Toy: How to hook a button to other object*

Hi --

Actually, any script can produce a button to fire it, and also, any object in Etoys can serve as a button (including text objects).

1. Suppose you have a text object named Text that contains some text: 'This is some text'.

Drag out Text's characters <- This is some text onto the desktop to make a script.

Do it again to make a second script. Name the first script "reset" and the second one "change".

Change the text in the "change" script.

These scripts can be tested by clicking on the (!). A script will give you a button to fire it. Look at the menu item in a script []

and choose the item button to fire this script.

Do this for the other script. Both of these buttons can be relabeled.

Here is what this example looks like []

2. You can also use any object as a button. Get out a rectangle. Put an empty rectangle script on the desktop. Put the tiles: Text's characters <- This is different text in this script. Now look at the normal menu and choose mouseUp. This script will fire when you do mouseUp on the rectangle object.

[]

You can make a toggle by adding a variable to hold true or false, etc.

Sat Mar 10, 2007*: group objects*

And of course, you can get out a playfield, fiddle with its options to allow it to be picked up, make it transparent, etc., then put it back in supplies, and it will form a prototype of this kind of playfield.

You can also use shift-drag to pick up a group of graphic objects to put them in the playfield. (This could be a smoother smarter operation than the current situation in Etoys.)

Mon Mar 26, 2007*: Bug on Bookmorph and a Question about Weasell Esay*

Thanks Scott --

Hello Offray --

On the other question about Weasel ... the existing version was an experimental "active essay" that Ted Kaehler and I did quite a few years ago to experiment with the form and see about a next level version of Etoys. We are in the process of redoing this in the OLPC Etoys version and will replace it "pretty soon".CheersAt 01:00 PM 3/25/2007, Scott Wallace wrote:

Tue May 8, 2007*: Croquet fails to initiate OpenGL*

Hi Eric --

The Squeakland list is about Etoys and is for school teachers and parents. The Croquet list is at opencroquet.org and the Squeak list is at Squeak.org.

Wed May 9, 2007*: MixedMorph for kids*

Thanks Karl --

A nice simple fun Etoy!

(We definitely have to make the poly control points bigger for the OLPC screen.)

Wed May 9, 2007*: the concept of mass*

But consider this Etoy (run in the OLPC version of Etoys) and also take a look at Takashi Yamamiya's site and work over the last few years.

Thu May 17, 2007*: presentations / slide shows.*

Hi Benedict --

Yes, all of the "slides" (they were actually Squeak/Etoys projects) were completely done in Squeak (I think only using the Etoys part of the environment).

Sun May 20, 2007*: When to define a local variable*

Hi Folks --

This list is mostly for teachers, parents and children, and is about the Etoys part of Squeak. Please use either the Squeak.org or the opencroquet.org (or one of the other Croquet lists) for these kinds of issues.

Thank you,

Alan

Mon May 21, 2007*: ideas to promote squeak in telecenters in Brazil*

Yes, Scratch is another Squeak based authoring system for young people. MIT did a nice job with it. It is aimed at teenagers and is more of a productivity tool than an educational authoring environment.

***Fri May 25, 2007**: Animation with displacement*

Hi --

What do you want to do?

***Fri May 25, 2007**: Lack of documentation frustrating*

Let's see, Oct 31st 2006 to May 25th 2007 seems to be 7 months, not 18 ...

But you are certainly right that much more documentation is needed. And, it is actually happening, just slower than hoped. But more of various kinds is close to being postable, and will be this summer.

There is also a lot more documentation than you think on the
http://www.squeakland.org
site. The site needs to be reorganized, but for example, there are literally hundreds of pages of documentation, tutorials, curricula, etc., done by the very active group at the U of Illinois. And there is quite a bit of "other" documentation. There is a book "Powerful Ideas in the Classroom" with a dozen or so sequential projects for 5th graders, etc.

Now, please tell me what it is that you would like to do in Etoys?
P.S. I don't hate to write documentation, but I'm not very good at it and hence painfully inefficient.

***Fri May 25, 2007**: Weird Heading numbers*

Our choice, which indeed goes against several longer standing conventions (like 0=East, +=counterclockwise, or 0 = N to 360 clockwise), was adopted because we were looking for something that would be close to how a 7 year old child thinks and had a symmetric use of positive, negative and 0 (the original Etoys was for a younger age group). This has worked very well and has many merits for all the ages of children we work with.

And, of course one can assign e.g. 245 to heading and the player will point as expected.

I think the preference you suggest would be a good idea. There is already one for all playfields (including the world desktop) as to whether (0,0) should be in the lower left corner or the center of the rectangle ([] origin at center). Both have their merits. Similarly, both negative degrees and staying positive around the circle have their merits. And we should probably (and likely will) put in a preference for the school math system, now that we are starting to work with older children who are somewhat thinking along those lines.

I think starting with other than 0 would be a bad idea (and would also violate the compass convention). Similarly, school math starts with 0 pointing East, not 360.

If you want to see the regular compass headings you can write a simple script that ticks once or twice a second that does the easy conversion into a variable, then you can use a watcher to see the value on the screen.

The debates we have are not about your issue but the more critical one of whether we should have gone with the "school math" conventions for more coherence later on. I still like starting with the compass with N pointing up and clockwise for +. But there is a good QWERTY argument for the school math conventions. (This gets more pernicious with older children who have to learn the school convention -- and when vectors are used -- should they be in school math or in compass?)

***Fri May 25, 2007**: Lack of documentation frustrating*

Hi --

Alan Kay wrote:
>
> Let's see, Oct 31st 2006 to May 25th 2007 seems to be 7 months, not 18 ...
>

Sorry, Never was great at math ;)

...

As I mentioned in another message, I was playing with an animation tutorial that uses the "holder" morph. Well I've searched through this forum and the wiki and still haven't found anything that documents what a "holder" does or the fields that are exposed by it's viewer.

If you want to learn ETоys, I strongly suggest going to the http://www.squeakland.org site and look at the tutorials. There are also some sample chapters from the "Powerful Ideas in the Classroom" book, and quite a bit of stuff from the active group in Champagne-Urbana. (I think I and several others have already mentioned this.)

Your example below is not in EToys but is in the underlying Squeak. There is a lot of Squeak documentation, but not much on how Etoys is implemented. But most people who program directly in Squeak use the Morphic graphics directly.

Etoys is really a different scheme with a different object and graphical semantics. It has some real strengths for children, but some real weaknesses for adults, especially somewhat sophisticated adults. Since it is aimed at children, this has not been a big problem.

One good ploy (which the children use all the time) is to exploit the fact that the underlying player object is essentially the same (i.e. highly polymorphic) for all graphical objects in Etoys. So exploring the standard viewer categories can be very rewarding. For example, one of the things you seek will be found in the viewer category called "Scripting". There are script lines to control ticking behavior, etc. for all the Etoy objects.

However, as I said, Etoys is not particularly complete -- it's more set up to have learnable structures by children. The way I program in it (and advise adults to program in it) is to use its features as the building blocks available, and this will bring to mind lots of simple fun structures, including many important parts of math and science. The way to be frustrated in Etoys is to bring a set of C-like or Java-like conventions from the outside and try to find the equivalents in EToys -- they might or might not be there.

(This was easier advice in the 60s when there were literally about 3000 different programming languages and hundreds of different ways to program. Most people were not surprised if a particular language had a different paradigm and were used to simply learning it.)

Sat May 26, 2007: *Weird Heading numbers*

Yes -- thanks Paulo.

Logo uses all child centric references such as right and left with the reference being up = N. One of the experiments early on in Etoys was to see how well + and - could be used in place of right and left with ~ 7 year olds. This was going to be part of a whole vectorized cuisinaire rods approach to numbers (that used some results we got in the Vivarium) that were unified with the number line way of thinking of + and - as directions (also right and left) of one dimensional vectors.

When the Etoys demo started being used in schools (instead of the home as originally planned) we found that grades 4-5-6 were a better fit with the tradeoffs between what children can do and what adults want to learn. So we never carried through the concrete number representations originally planned.

(However, now we are going to because the OLPC XO needs to have a K-12 range. This actually requires a somewhat different approach to Etoys than the demo version we have now, and we are working on it, which supporting current Etoys for the various XO builds.)

Most adult conventions and forms have huge QWERTY components which make learning more difficult for children. However, eventually the conventions need to be added in. It's imperative to start children thinking in the strongest and most intuitive way -- then we can figure out how to merge in the somewhat ad hoc conventions that adults have devised. Various ways of thinking about numbers, lengths, directions, magnitudes etc is a ideal way to eventually get to some of these conventions. But, e.g. trying to get children started into real numeracy with positional notation is really bad, even though it is a mainstream convention ... the general result in America is that children don't get numerate, even though they are forced to learn how to parrot a few of the conventions.

Wed May 30, 2007: *Discovering Pi in Squeak*

Hi --

Bert points out that it is easy to use forward by, turn by to make pologons whose diameters can be measured.

For example, you can make a big circle with a turn by 1 and sum the forwards, and also remember max y and min y to get the diameter. This will give you a pretty good value for Pi.

You didn't mention the ages of your children.

But it is always good to get them to do some reasoning about measures of various kinds and areas. I think that the manipulation of the strings, etc., might be too awkward (but see the discussion in the "Powerful Ideas" book about measurement).

I would just give them squares of different sizes and see if they can work out how a side might relate to the perimeter, and if so, why something like that would also work for a diagonal. The idea that the relationship is the same regardless of scale is a biggie for children. Discovering the relation for the area is even bigger.

Before that I would use rulers with different scales to make similar figures (starting with triangles), and get them to see that nature doesn't care how our rulers are laid out (a measurement taken with one ruler can be used to make a similar figure of a different scale using a different ruler). This is a very good way to show how and why proportions work (and many studies have shown that proportions and the normalizations associated with them are not learned well by most children).

Thu May 31, 2007: *Discovering Pi in Squeak*

Hi Subbu --

And of course Pi is not at all required for "turn by" or "forward". It is one of many constants in the init method for float but is not needed for these behaviors. "foo turn by ang" is easily written in Etoys as "foo heading increase by ang" where "heading" is a simple property, so there is no Pi involved here. "foo forward by 5" is more complicated (because it involves a vector addition to foo's location) but still requires no Pi.

The other thing to think about that if it was "vectors all the way down to the display" (as indeed it was in the old days of calligraphic displays) then one would have a direct analog to a coordinate free system in the hardware. The artifact today of having to supply the display machinery with a 2D array of pixels forces a kind of transformation and a kind of coordinate system, but still doesn't require Pi.

Thu Jun 14, 2007: *Value as Text*

Hi --

There are two ways to do this. Texts have a setter for "numeric value".

[]

Even simpler (though a little more kludgy) is to get a "simple watcher" from the viewer that holds the property you wish to see, click to the "Updating String" (called "Readout String" in OLPC version), and use red menu to make the changes.

In "regular Etoys" a shift-click will get you this component, and you can use the red menu to change the font and size, etc.

In OLPC Etoys, use the red menu to "Unlock Readout String", then select it and use its red menu to change font and size.

Mon Aug 6, 2007*: OLPC squeak on OSX*

And, of course, the OLPC Etoys runs fine on a Mac or Windows OS ...

Sat Aug 11, 2007*: Demoing Etoys to kids*

Hi Luke --

Kim knows where this stuff is better than I, but there is quite a bit of material about introductory use on squeakland.org. The "make a car and drive it" is always a good one to start off with. Most of these exercises are in Kim's book, and she can send you a pointer to some pdfs.

You could also look at the half-finished (but still 50 page) doc I did for OLPC. It has lots of different examples.

This is at http://www.vpri.org/pdf/rn2007006a_olpc.pdf .

Also, we just met some folks from Nepal who have been actively doing Etoys there for a few years. Kim might have their email addresses....

Sun Aug 12, 2007*: More thoughts - Re: Demoing Etoys to kids*

In July I traveled around to a number of workshops in different parts of the US. While at CMU in Pittsburgh, I got a surprise invitation to talk to 11-14 year olds who were attending a summer computing day camp. There were about 25 kids and this was small enough for me to show them things and ask questions, etc. For projects, the lack of prep time led me to just use what I had been showing the teachers minus the philosophy slides. Here's the list that we went through in about an hour.

Most of the them are the standard examples I show. There were a few new ones.

Make a car and steering wheel and drive it around. Use a property (Wheel's heading). Use a scale (Wheel's heading/3)

Kedama epidemic while graphing the percentage of infected villagers: 2, 1000, 500, 100. As what the diseases on the extremes look like. Answer from the kids: the fast acting one was like Ebola, and the slow one was like AIDS.

How to follow a road with a robot car Ashley and Janae's middle of the road robot car Middle of the road car with two headlights as sensors Jenny's Pig Race

Salmon Sniff -- gradient following feedback with single and multiple salmon Fish and Plankton (this could be omitted, it was just in this sequence) Ant colony -- the classic ants and food feedback system

Speed and acceleration reflection on increase by (using dots and arrows to get visual picture of S and A). Animation using holder and increase by Bouncing ball animation (change "speed" increment to change rate through the images) A movie is an animation (example is how the upcoming ball drop movie was made) A music synthesizer is an "animation" (change "speed" increment to change pitch, etc.)

Computer Logic Game (Alex Warth's way to use costumes as visualizers of state and for state). Wires, Not, And, Or, etc. gates. How to make a script (a visual Logo interpreter that uses Etoys polymorphism to do the interpretation) A rule based programming system (another way to do interpretation to make a StageCast like interpreter and then do the epidemic sim) ToyLog (a visual animated Prolog with English syntax and using the Simpson family as a database)

==== That was all we had time for ======== Naturally, they loved it and wanted to do things with it.

Some advice about teaching geometry to children. It's possible that the teacher may not understand geometry very well (and you didn't indicate what she meant by "geometry").

In any case, in general (and especially for 3rd graders) I would not center the learning on the computer to start with. There are three or four things that children this age can learn deeply that using the computer in an ancillary fashion with help make stronger. One theme to try is "map-making" starting with making models of objects and progressing to models of classrooms, school-yards, surrounding neighborhoods, etc. Our experience with this came from a highly successful adaptation of Doreen Nelson's "City Building" Curriculum for 3rd graders that we did in the LA Open Magnet School some years back.

Scaling and proportion are approached by having the children make "object costumes" (taking household objects and blowing them up into wearable costumes) and then having an "object parade". The Open School had made an garden and it was traditional for the 3rd graders to design it. So they had to measure it and make a model to let them think about it in the classroom. This got them to think about scaling the other way. One way to do both scaling is to have two measuring systems with equally spaced tick marks but with different distances between the ticks, etc. Make the ration between the tick marks something the children can compute in their heads (they don't need this to do the scaling but they do need it to start understanding ratios).

The scaling ideas can start to be used on the computer, etc.

This part of the curriculum leads to a very rich city of the future design effort stretching over many months that combines many kinds of math, design, and systems thinking.

Another thing that goes strongly with this way of thinking about measurement and scaling above is that nature doesn't care what rulers we use. We can use this idea to start talking about and using similar triangles to do scaling and indirect measurements (heights of school buildings, long distances, etc.). Forget about trig terms, etc. and concentrate on similar triangles as one of the most powerful ideas of all times.

For example, if we occlude a quarter with a dime and measure this carefully, we see that the distance in diameters has to be the same. []

And if we then occlude the moon with a coin (as Aristarchos of Samos indeed did!) we will find that it takes about 110 coin diameters, and this means that the moon is 110 moon diameters away from us!

[]

Children love this (too bad adults don't, or they would know about this and teach it to children).

Sun Aug 12, 2007*: More thoughts - Re: Demoing Etoys to kids*

Hi Stephane --

At 12:11 PM 8/12/2007, stéphane ducasse wrote:
Hi alan

I always liked the way ancient measured pyramid using the shadow of a know piece of wood and use Thales theorem. At least it was a really practical example, I used to teach Thales beauty.

Yes, and it is even simpler for the children to think just in terms of similar triangles.

>QQ:For example, if we occlude a quarter with a dime and measure this >QQ:carefully, we see that the distance in diameters has to be the same. >QQ:<dimeQuarter.png>
Now I do not understand the "distance in diameters"

Check the picture. If it is 9 dimes from the eye to the dime, it will be 9 quarters from the eye to the quarter. If it is 110 quarters from the eye to the quarter that occludes the moon, the moon is 110 moon diameters from earth.

Which > >And if we then occlude the moon with a coin (as Aristarchos of >Samos indeed did!) we will find that it takes about 110 coin >diameters, and this means that the moon is 110 moon diameters away >from us!

How do we get 110?

You measure the number of coin diameters from the eye to the location of the coin that occludes the moon.

Mon Aug 13, 2007*: the non universals*

Hi Bill --

There are various sources for "universals" on the net and off. Quite a bit more has been found out about these since the days of Lorenz and Tinbergen. One of the several fields that studies these as scientifically as possible is called "NeuroEthology" and there are a number of good books on the subject. T.G.R. Bower was one of the first to study very young humans specifically. An ancillarly field that has appeared in the last few decades is called "bio-behavior", and there also a number of illuminating books there.

I picked some of the "non-universals" that I thought were important (and some particularly to contrast items in the universal list).

To answer your question marks ...

"Theory of Harmony" is kind of like "Deductive Abstract Mathematics" in that most traditional cultures have some form of counting, adding and subtracting -- and some make music with multiple pitches at once (as did Western Culture before 1600). But the notion of harmony before 1600 was essentially as a byproduct of melodies and voice leading rather than a thing in itself in which chords have the same first class status as melodic lines. How and why this appeared is fascinating and is well known in music history.

Some of the most interesting composers in the Baroque period (especially Bach) tried to make both the old and the new schemes work completely together. Bach's harmonic language in particular was an amazing blend of harmonies and bass lines with voice leading and other contrapuntal techniques (quite a bit of his vocabulary is revealed in his harmonized chorales (some 371 or 372 of them)). That these two worlds are very different ways of looking at things is attested to by a wonderful piece by Purcell "The Contest Between Melodie and Harmonie".

As with "Greek Math", history doesn't seem to have any record of a separate and as rich invention of a harmonic theory. So it is really rare.

"Similarities over Differences" was to contrast with the standard processes of most nervous systems of most species to be more interested in "differences over similarities" (which is listed on the universal side). At most levels from reflexes to quite a bit of cognition, most similarities are accommodated and normalized while differences to the normalizations have a heightened significance (of "danger" or "pay attention").

Paying attention to differences is good for simple survival but makes it hard to think in many ways because it leads to so many cases, categories and distinctions -- and because some of the most important things may have disappeared into "normal" (in particular, things about oneself and one's own culture). So we unfortunately are much more interested in even superficial differences between humans and cultures and have a very hard time thinking of "the other" as being in the same value space as we are....

Part of the invention of modern math by the Greeks was their desire to get rid of the huge codexes of cases for geometry and arithmetic. This led to many useful abstractions which could be used as lenses to see things which looked different to normal minds as actually the same. For example, the Greek idea that there is only one triangle of each shape (because you can divide the two short sides by the long one to make a standard triangle of a given shape). This gets rid of lots of confusion and leaves room to start thinking more powerful thoughts. (The Greeks accomplished the interesting and amazing feat of using normalization to separate similarities and differences but paid attention to the similarities.) Calculus is a more subtle and tremendously useful example of separating similarities and differences. Convolution theory is yet more subtle ...

One way to think of my chart is that a lot of things we correlate with "enlightenment" and "civilization" are rather un-natural and rare inventions whose skills require us to learn how to go against many of our built in thought patterns. I think this is one of the main reasons to have an organized education (to learn the skills of being better thinkers than our nervous systems are directly set up for).

History suggests that we not lose these powerful ideas. They are not easy to get back.

The non-built-in nature of the powerful ideas on the right hand list implies they are generally more difficult to learn -- and this seems to be the case. This difficulty makes educational reform very hard because a very large number of the gatekeepers in education do not realize these simple ideas and tend to perceive and react (not think) using the universal left hand list .....

Wed Aug 15, 2007*: the non universals*

Hi Bill --

I'm in a rush, so will reply more extensively later.

But, of course, the non-universals are easy for anyone who understands a fair number of the universals and who reads a little. Most cultures on Earth have not had writing systems (and probably still most today). Etc.

Wed Aug 15, 2007*: the non universals*

OK, a few more minutes ...

At 08:05 AM 8/15/2007, Bill Kerr wrote:
hi alan,

Thanks for extensive clarification of the items which I had left question marks on

From what you say the "non universals" group originates from you (!) which sort of explains why I couldn't find other references to it on the net

Gathering the behaviors that can scientifically be claimed to be universals is what requires diligent work by experts, since literally thousands of cultures need to be perused -- and a fair amount of experimentation with early childhood behaviors is critical. Once, gotten we can easily claim that opposites (like writing and reading) are not universal. Deductive math and model based science are also easy. As is "equal rights", etc.

I have used your lists at a few meetings and it has provoked a response of sorts. On the one hand some people say the "non universals" is an interesting list. However, I've also noticed some reluctance or inability to discuss the items on the list in any real detail or to discuss the implications for the formal education system. ie it seems to come at people from left field

It requires some "perspective and knowledge" (which was what education used to be correlated with) to do the discussion. E.g. if one is not fluent in math and/or science it is difficult to understand just how qualitatively different are the modern versions of these. Most educational reform stalls in large part from the below threshold educations of the adults in the system. For most people, most powerful ideas come at them from left field.

In his dissertation on the history of the Dynabook John Maxwell asks "what is a powerful idea, anyway?" and also argues that there has been a decline of powerful idea discourse

Well, "powerful ideas" is a nice metaphor that Seymour made up to heighten people's understanding (and the significance) of the relatively few and rare inventions that have made huge differences in how humans meet and think about the world. It is normal behavior to accommodate to what is present (especially what one was born into) and so most people think of the powerful ideas as part of normal, and since most Americans have not traveled in a way that gets them to appreciate the wide range of situations that humans are in around the world, they completely miss the wonder and mystery of "better intellectual architectures". This is why there are so few scientists (because most people take things as they seem and completely miss what deeper curiosity and better methods can find out).

What I'm noticing in educational discussion groups, blogs etc. on the web of late is much talk about "web 2.0", "school 2.0" but this tends to take place outside of a framework that maybe there are powerful ideas that really do have to be taught in some way.

You do say that the major stakeholders don't get it. What I see there is curriculum frameworks being used as blunt instruments of control. I'm suggesting, too, that many of the "radicals", who describe themselves as "web 2.0" are not getting it either.

Most of this is just a cargo cult.

In this context I like the idea of your list of "non universals" and John Maxwells' idea of the need for more powerful idea discourse. However, I'm also left feeling a bit unsure of the status of the "non universals" list, eg. how complete is it? have people argued about it and disputed it?

Neither list is complete. But the important property of the universals list is that most the items are well vetted. The importance of my non-universal list is just that 5 or 7 items are the most important changes that humans have made in their 200,000 years on the planet (and most of these came very recently (even agriculture)). What more do people need to start thinking with? What more arguments about modern science need to be made? (And if they do need to be made, then what new kind of argument would work?)

In other words, if the items on my list are ignored then it really doesn't matter much what else could be on the list. For example, the notion that there are "powerful ideas" could be the number one powerful idea, since it should lead to trying to understand powerful ideas, and to trying to find more of them.

I could think of some non universals / powerful ideas that are not on your list, eg. Darwinian evolution, computer-human symbiosis for starters ...

Sure. There are lots (and they should be paid attention to). But certainly Darwinian Evolution (and a lot of other things fall under Science), etc. Computer-human symbiosis falls under the larger topics of how human thinking can be changed by the use of media (for better or for worse), etc, For a short list, it's best to use the biggies. Similarly, if we listed every built-in human trait (especially the zillions of bad ones), the list would be too long for any discussion purposes.

I'm also curious about its connection with using computers in learning. Clearly etoys and logo can be used to assist teaching some of those concepts in constructionist fashion, esp maths and science. But for others I don't see a close connection at the moment (eg. equal rights, democracy) - although the OLPC project is becoming a part of that.

Some music needs to be sung by human voices, and the best instruments in the world won't help (and will usually detract). Similarly, some theatrical expressions have to be done directly in person live, and will be diminished by inserting even high resolution media. Similarly, even the wondrous nature of mathematics often is too visible so it can obscure what's most interesting about what is being described.

We are talking about human thinking and perspective, not computers here.

However, consider this wonderful phrase from Marshall McLuhan "You can argue about a lot of things with stained glass windows, but Democracy is not one of them!" He meant that not just non-visual oral language, but only written, even printed language was disembodied and abstract enough to handle the critical issues and subtleties of this discourse. This itself is a powerful insight and a powerful idea that most adults in the world today have no notion of, and most would find it almost crazy.

In other words, as Neil Postman liked to point out, the relationship between human thought and the languages/media used to form and express it is not a separated one, but is a non-linear ecology. Dropping something like TV into a society is like introducing rabbits into Australia. Not really thinking about computing and networking as new kinds of rabbits for good and ill can lead (and has led) to disastrous effects already. On the other hand, geniuses (like Montesorri, Papert, Bruner, McLuhan, Engelbart, etc.) who have thought about how environments of any kind condition "normal" and thus much of human thought and behavior have come up with very powerful and positive ways to use new and old environments to help humans more successfully struggle with their less well fitted internal behavior patterns.

The basic approach here is to hold focus on what is really important, and to design new media and environments to help people learn what is important. As Seymour pointed out long ago, not even in the educational swamps of America do they use the phrase "paper based education" since it is patently ridiculous. But because people don't understand computers (and because magical thinking is one of the human universals) any new technology is treated as a talisman -- and they have no trouble in generating phrases like "computer based education" or "computer based curriculum" or "web based learning" etc. This is also cargo cult behavior.

Most people "take the world as it seems" as I mentioned above, and so they completely miss most of the important properties and issues. This is why having general discussions about powerful ideas often leads nowhere.

(And most discussions on the web similarly get nowhere -- opinion gets exchanged, but opinions have always been exchanged for 200,000 years with nothing much happening. The concatenation of opinions almost never leads to a better set of ideas -- this is a big bug in "web myth" and "collective behavior myths" in general. This is because the opinions in order to be understood have to share quite a bit of the same outlook, but progress usually comes from big changes in outlook. What we need are not more opinions and endless discussions, but more hooks to find stronger outlooks (aka "powerful ideas").

***Wed Aug 15, 2007**: the non universals*

Hi David --

Someone once asked Mohandas Gandhi what he thought of Western Civilization, and he said he "thought it would be a good idea!" Similarly, if you asked me what I thought of University Education, I would say that "it would be a good idea!".

There seems to me a desire among educators to help as many children and young adults as possible make the leap from arithmetic to geometry and calculus, from literacy to literary analysis, or indeed from melody to harmony. So where is the difficulty? A lack of proven agreed teaching methods, a perception of elitism, or the competing desire we all feel to make sure everyone leaves school with basic literacy and numeracy?

My perception of your first sentence is very different than yours. Most educators in K-8 do not seem to know anything about calculus and precious little about geometry or algebra (and their knowledge of arithmetic is rule-based not math-based) so I don't see whatever desires they might espouse about these progressions as having much substance. I do think that one is likely to get much better instruction and coaching from music teachers and sports coaches -- in no small part because they are usually fluent practitioners, and do have some real contact with the entire chain of meaning and action of their subjects.

I don't have deep direct scientific knowledge of the nature of the difficulties, just thousands of encounters with various educational systems around the world and educators over the last 35+ years. So I could have just been continually unlucky in my travels....

In the early 80's I went to Atari as its Chief Scientist to try to get some of Papert's and my ideas into consumer electronics. The Atari 800 and especially the 400 were tremendous computers for their price, and Brian Silverman made a great version of Logo to go on these machines. (There were also Logos on most of the other 8-bit micros.) And, there was a Logo-vogue for a time, both in the US and in the UK. Many early adopter teachers got Atari's or Apple IIs in their classrooms and got their students started on it.

This was exciting until examined closely. Essentially none of the teachers actually understood enough mathematics to see what Logo was really about. And for a variety of reasons Logo gradually slid away and disappeared.

We should look a bit at three different kinds of understanding: rote understanding, operational understanding, and meta-understanding. If we leave out the majority of teachers who don't really understand math in any strong way, we still find that the kinds of understandings that are left are not up to the task of being able to see the meaning and value of a new perspective on mathematics. For example, it is possible to understand calculus a little in the narrow form in which it was learned, and still not be able to see "calculus" in a different form (even if the new way is a stronger way to look at it). Real fluency in a subject allows many of the most powerful ideas in the subject to be somewhat detached from specific forms. This is meta-understanding.

For example, the school version of calculus is based on a numeric continuum and algebraic manipulations. But the idea of calculus is not really strongly tied to this.

The idea has to do with separating out the similarities and differences of change to produce and allow much simpler and easier to understand relationships to be created. This can be done so that the connection between one state and the next one of interest is a simple addition. Actual continuity can be replaced by a notion of "you pick and then I pick" so that non-continuities don't get seen. This other view of calculus as a form of calculation was used by Babbage in his first "difference engines" because a computing machine that can do lots of additions for you can make this other way to look at calculus very practical and worthwhile. The side benefit is that it is much easier to understand than the algebraic rubrics. If we then add to this the idea of using vectors (as "supernumbers") instead of regular numbers, we are able to dispense with coordinate systems except when convenient, and are able to operate in multiple dimensions.

All of this was worked out in the 19th century and quite a bit was adopted enthusiastically by science and is in main use today.

To cut to the chase, Seymour Papert (who was a very good mathematician) was one of the first to realize that this kind of math (called "vector differential geometry") fit very well into young children's thinking patterns, and that the new personal computers would be able to manifest Babbage's dream to be able to compute and think in terms of an incremental calculus for complex change.

Any one fluent in mathematics can recognize this (but it took a Papert to first point it out). But, virtually no one without fluency in mathematics can recognize this. And surveys have shown that less than 5% of Americans are fluent in math or science. Many of the 95% were able to go through 16 years of schooling and successfully get a college degree without attaining any fluency in math or science.

This is not a matter of intelligence at all, but is more of a "two cultures" phenomenon. So I am not able to agree with this sentence of yours:

This barrier is puzzling to me, as the key gatekeepers in education (teachers, head teachers, inspectors, government education departments) are products of the university system, which seems to me to exist to propagate and build on the hard ideas (greek math, relativity, quantum theory, sociology, musical harmony ...)

It is possible to learn about these ideas in university (and outside of university), but I don't know of any universities today whose goal it is to invest its graduates with fluency in these ideas or any other powerful ideas. That is, the concept of a general education for the 21st century that should include these ideas doesn't seem to be in any American university I'm familiar with.

If Logo, Etoys and OLPC can teach calculus to 10-year-olds, and calculus is essential to every engineering craft, and teachers love encouraging students' creativity, why are so many schools teaching pupils to use word processors instead?

The problem is that Logo, Etoys and OLPC can't teach calculus to 10 year olds. The good news is that adults who understand the subject matter can indeed teach calculus to 10 year olds with the aid of Logo, Etoys and OLPC.

If you put a piano in a classroom, children will do something with it, and perhaps even produce a "chopsticks culture". But the music isn't in the piano. It has to be brought forth from the children. And the possibilities of music are not in the children, but right now has to be manifested in the teachers and other mentors. (It took several centuries to develop keyboard technique, and much longer than that to invent and develop the rich genres of music of the last 6 centuries.)

Math and science were difficult to invent in the first place (so Rousseau-like optimism for discovery learning is misplaced), and both subjects have been developed for centuries by experts. Children need experts to help them, not retreaded social studies teachers.

One of the goals of 19th century education was to teach children how to learn from books. This was a great idea because (a) oral instruction is quite inefficient (b) you can get around bad teachers (c) you can contact experts in ways that you might not be able to directly (especially if they are deceased) (d) you can self-pace (e) you can employ multiple perspectives on the subject matter (f) you are not in the quicksand of social norming, etc. A small percentage of children still are able to learn from books, and similar small percentages of children can and do learn powerful ideas by themselves without much adult aid.

But since general education is primarily about helping to grow citizens who can try to become more civilized, the big work that has to be done is with those who are not inclined to learn powerful ideas of any kind.

Thu Aug 16, 2007*: the non universals*

Below is a recent article from Education Week. In (only) my opinion, it should be impossible for 93% of American teachers to like their jobs if they had any perspective on what they are doing, how they are doing it, and what they are supposed to do. There are a few other mildly interesting tidbits at the very end of the article.

To me this is an example of how a field can and does select the personalities and skills that fit to its actual mission. I saw this very strongly when I was in the Air Force (whose general way of doing things I really did not like). I left after my required term, but many re-upped, and they were the ones that fit into that particular scheme.

Another example of the ecological power of environments and the co-evolution and selection of environments and traits.--------------
Education Week

Published Online: August 1, 2007

Teachers Tell Researchers They Like Their Jobs

By <http://www.edweek.org/ew/contributors/vaishali.honawar.html>Vaishali Honawar

Ninety-three percent of teachers reported satisfaction with their jobs 10 years after entering the field, according to a new survey that also found attrition rates for teachers were actually lower than for other professionals.

The <http://nces.ed.gov/pubsearch/pubsinfo.asp?pubid=2007163>report, released this week by the National Center for Education Statistics, surveyed 9,000 graduates who received their bachelor's degrees in various disciplines in the 1992-93 school year. Nearly 20 percent of those graduates entered the teaching profession.

The findings from the survey debunk several long-held views on teacher pay, turnover, and job satisfaction. For instance, it found that only 18 percent of those who entered teaching changed occupations within four years of getting a degree. Given that other professions experienced attrition rates between 17 percent and 75 percent during that period, the number of career-switchers from teaching was on the low end of the scale, according to the data. More than half those who became teachers were still teaching 10 years later.

Teacher advocates and unions have long claimed that turnover among new teachers ranges from 30 percent to 50 percent within the first five years.

"The take for a long time was that there is this incredibly high attrition among teachers from schools," said Mark Schneider, the commissioner of NCES, an arm of the U.S. Department of Education. The report, he said, shows that teacher-turnover rates are actually lower than those in other professions.

"I understand why schools and school districts are upset about losing teachers, but it is part of the normal sorting process" in a dynamic job market, Mr. Schneider added.

The survey also stands on their head some commonly held beliefs about teacher salaries. Teachers' unions have often cited low pay as a major reason for teacher dissatisfaction. But only 13 percent of those who left teaching by 2003 gave it as the reason for leaving. Forty-eight percent of those who remained in the profession said they were satisfied with their salaries.

Kate Walsh, the president of the National Council on Teacher Quality, a research and advocacy group in Washington, called the findings "explosive."

"What was surprising is how cheery the [teachers'] responses were," she said. Education groups, including the unions, she contended, often cite teachers' unhappiness in order to pressure districts and states for concessions.

Spokesmen for the National Education Association and the American Federation of Teachers said they were unable to comment on the report before the story was posted.

Racial Differences

The report's findings are based on the NCES' survey of baccalaureate-degree recipients conducted between 1993 and 2003. Participants answered questions via phone and the Internet and during in-person interviews. The report was prepared by MPR Associates in Berkeley, Calif.

Of those surveyed who were still teaching 10 years after earning their degrees, 90 percent said they would choose the same career again, and 67 percent said they would remain in teaching for the rest of their working lives.

The rate among African-American teachers, however, was significantly lower, with 37 percent saying they would choose to remain in the profession, compared with 70 percent of white teachers.

Nearly 20 percent of black teachers said they would leave if something better came along, compared with fewer than 10 percent of white teachers.

Ms. Walsh said the higher rates of dissatisfaction among black teachers could be due to the fact that more black teachers teach in high-poverty schools.

The study reaffirmed that attrition rates were higher among male teachers. While women (29 percent) were more likely to leave for family-related reasons, men (32 percent) usually left for a job outside the field of education.

A candidate's age when he or she attended college also appeared to play a role in attrition rates: Those 30 or older when they obtained their degrees were more likely than younger graduates to remain in teaching.

Those who earned better grades in college were more likely than those with lower grades to remain in teaching.

The study offers a window into how college graduates perceive teaching. For instance, nearly half of all bachelor's degree recipients in 1992-93 said they had never considered teaching or taken any steps to become educators.

Lack of interest, having another job in hand, and inadequate pay were the most commonly cited reasons for not pursuing teaching.

Math, science, and engineering graduates were among those most likely to leave teaching jobs to work outside education.

Fri Aug 17, 2007*: the non universals*

Hi David --

I'm not pessimistic. If I were, then I would pursue other ventures. I'm just thinking like a scientist (which is trying to figure out a near version of how things actually are). If you look closer, I think you will find that I'm being quite even-handed.

Bush and his administration (among many others) think scientists are pessimists because they don't make up stories "that are so nice they must be true" as most people do, but instead are skeptical (not the same as pessimistic at all) and try to be "realistic" (as science thinks of that term), and are certainly optimistic, since they think they can uncover mysteries and make models of important things in the universe that have baffled humans for hundreds of thousands of years. There's a certain amount of arrogation (and some plain arrogance) in science, but not a lot of pessimism.

While for the next generation primary (K-6) teachers may be a lost cause, what I want to understand is why you (Alan) don't find large *numbers of secondary (grades 7 - 12/13) math and science teachers becoming advocates and allies of the reforms you are proposing.*

If mathematics shares some traits with language and muscular learning (and there is evidence it does), then the big disaster is in K-6.

7-12 has many of its own problems, judging both from what I've read and from participating in STEM workshops in many parts of the US this summer.

Most of the 7th and 8th grade teachers we worked with were retreads from non math and non science teaching. The simplest generalization is that almost none of them showed any heuristic sense and aim for math of any kind. They knew a few facts but did not know how to think about even what they could remember.

There is a wider range in high school teachers, and we found more than a few percent (maybe 10% to 20%) who could follow the relationships between familiar ways of looking at things and other ways of looking at the same underlying ideas. This is better and lots could be done with these teachers. This is not a high enough percent to make big changes but it would be a good start if the system would allow the goals and methods to be different.

Of course, this is far from a scientific survey ....

1. Are the math and science teachers not aware that calculus is a 'powerful idea'?

Not in the sense that Seymour uses the term. This is partly because almost no math and science teachers in HS were ever practitioners, and most were never math or science majors. Some may have majored in "math education" etc., but there is a huge qualitative difference there. Remember that most HS science is done without calculus (because it is still an optional AP subject that is taken usually in the last year of HS).

Also, there is the interesting survey result which I sent out earlier this morning which indicates that 93% of the teachers in the system like their jobs. This is quite incompatible with any real understanding of math and science.

2. If they are, are they not sufficiently fluent in it ...

I think they are indeen not sufficiently fluent in it, especially in the "what it actually is" sense (as opposed to "this particular way").

to understand that their current teaching method (whatever that is) is not engaging and developing nearly as many pupils as have the potential to get it, enjoy it and use it?

This is strongly combined with the standards, SAT, and AP criteria to make the teachers who do have some sense of other ways to feel completely trapped in HS. But it's not just the teaching methods, it's the actual form of the knowledge for learners of mathematics and science.

3. Or is there some other reason, such as suspicion of new methods, waiting for something better, or insufficient time after concentrating on basic numeracy?

Sure, and etc. Pretty much everything in American High Schools has high levels of trying to reteach virtually all of what the kids were supposed to have learned in the earlier grades. Hence, the need to look earlier for solutions. Couple this with the difficulty of learning new outlooks once you have already committed to outlooks that are not so fruitful, and the earlier grades are the place to work on.

The reason for the question is my big worry, inspired by your original post: if Papert's ideas don't engage secondary school math teachers, they have few other advocates left. There is no back door to get around these gatekeepers.

That is one of the big problems, amongst a dozen others. Cargo cults are difficult to reform once they get going. But what if the secondary math teachers complained loudly? I don't think they are in any decision process that I can find.

***Fri Aug 17, 2007**: the non universals*

Hi David --

At 02:58 PM 8/16/2007, David Corking wrote:
Thanks for wrestling with my questioning, Alan (btw - it seems we forgot to share our last two exchanges with the mailing list - my fault - I refrained from repeating your responses extensively here in case it not your intent to post them.)

I didn't notice this, so just reposted the previous reply to the list.

On 8/16/07, Alan Kay wrote:
> Of course, this is far from a scientific survey ....

You clearly know far more teachers than I do. I am shocked to hear that so few US math and science teachers were math and science majors, or were even educated in any college level math and science.

As I said, my little survey this summer wasn't scientific ... it would be nice to have a much better assessment of this done in a more rigorous fashion.

But I don't think it is an exaggeration to guess that most teachers lack the kind of operational mathematical thinking that is the most important part of mathematical fluency. And it is very likely that "most" is almost total in the elementary grades and means "very sparse" in high school.

I suspect it is normal worldwide to postpone calculus until the equivalent of "Advanced Placement" courses in years 11 and 12 - I hope it is mandatory to know calculus before going to college for math, science or engineering (and perhaps for social science too.) Perhaps by the delay we then rob many kids the chance to (1) see its beauty and (2) see that it underpins so much of modern science and engineering.

"Knowing calculus" is a tricky phrase. An important idea here (that I originally got from Ivan Sutherland) is to ask whether skills are "10 hour skills", "100 hour skills", "1000 hour skills", etc. Ivan once pointed out that e.g. learning to play piano was not a "10 hour skill" no matter how much latent talent you might have. And, though talent does play a part in time to learn and get fluent at something, it takes time for people's brain/minds to build the structures needed for doing the thinking in question, and doing it fluently enough, etc.

As I recall, this discussion came up when a bunch of us grad students and Ivan at Utah were working out the mathematical transforms for 3D graphics (much of which constitutes OpenGL today). We had some terrific French grad students, who were better prepared mathematically than most of the Americans. I had an undergrad math degree, etc. Ivan is quite a bit smarter than most people, etc. Yet, this was a real struggle for all of us to "get operational" in a theory that we all "kind of knew" really well: the transformations of vectors using matrices, with the addition of the homogenous coordinates idea from projective geometry that Larry Roberts had suggested.

Ivan's observation was that we had attained the "100 hour skill" version of transformations, but not the "1000 hour skill" version. And this led to discussions of other skills in other areas. This idea is particularly striking and easy to understand in sports and music. And also overlaps with some of the findings in cognitive psychology about habit formation and habit unlearning. For example, if you put in 10 hours a week trying to learn something (2 hours a day, 5 days a week) and take a two week vacation, etc., then you will be spending about 500 hours a year doing your learning and practicing. Two years of this is 1000 hours.

Lots of good things can be learned to the solid mid-intermediate level in two years. And 18 months to 2 years is also the time that cog psych says it takes to form a solid habit (or unlearn one). Also very interesting are the results from many studies of the attempts at educational reforms in the 60s which showed that one year of a super enriched experience didn't stick, but two or more years did.

So the concept of a "1000 hour skill" is worth contemplating when looking at instructional systems.

Back to calculus for a second: I really didn't understand calculus in worthwhile ways (that is, to think in terms of what it meant rather than just trying to apply the techniques) until I took Advanced Calculus with a very good prof (who was a fabulous mathematical thinker and teacher). This was quite shocking to me (because I didn't really have a sense that I didn't understand calculus until I understood it so much better). And all of us working with Ivan a few years later had a similar sense about transformations. We only thought we understood them until we understood them deeply enough to think in terms of them, not just try to use them.

I think there are also real analogies here to stages of learning a foreign language. Seymour and I have talked a lot about this, and he thinks so also. The differences between being able to use another language a little and being able to think in "its perfume" are profound.

This is where more longitudinal approaches and immersion are critical. One of the reasons I loved Seymour's ideas and approach was that it would be possible to have children immersed in "CalculusLand" in ways meaningful to them for years so they could gradually build up real "CalculusThink".

As you point out, the algebraic model of calculus is not interesting to many people, but the difference model would, I imagine, be useful to every aspiring mechanic, lab technician or customer service supervisor.

Its not that the algebraic model of calculus is not interesting (it really is) but it is much further removed from most people's fluencies. The difference model is just simple accumulation by addition, and the equivalent of higher order differential equations is just more accumulations by addition lined up. I have written about how Julia Nishijima (the first grade teacher who had a real mathematical sense) could set up projects that would induce the children to discover and derive second order discrete DEs (first order is steady growth, second order is quadratic, etc.), These are the very same progressions that can be used for velocity and acceleration, F = ma, Galilean gravity, etc. so it is terrific to get started with these as tools one has derived in first grade.

We have a nice way to (later, perhaps in 7th or 8th grade) reconcile the easier incremental approach to the algebraic formulas by deriving the latter from the former. This is not just pro forma but is a very useful way to start thinking about what it is that is being said (and how universally) using quantification. It's also very illuminating to start thinking about integration and what it means in the universally quantified world (leading to the fundamental theorem of calculus).

But having quite a few years of calculus thinking and doing under one's belt is a much better way (in my opinion) to approach some of the deep and initially non-intuitive properties of calculus.

> But what if the
> secondary math teachers complained loudly? I don't think they are in
> any decision process that I can find.

I don't know the US systems very well. I would like to think that school boards and education departments consult professionals first. Are there countries where that does happen?

It's very tricky in the US -- in part because there are 25,000 or more individual school districts. There are state and national standards. Professionals are consulted. Etc. I only have speculations on how the system has not managed to do better with mathematics curricula.

One thing that seems to be almost universal around the world, is that the notion of children learning some subject (like mathematics) is almost always posed as "how can children be taught the adult version of this subject?", rather than, as Montessori, Piaget, Bruner and Papert have shown "how can we find an honest children's version of this subject?".

Another important idea here is that there are likely to be other approaches that are also better than the standard ones. Seymour (and I and others in his footsteps) simply have worked out one set of insights that can allow children to actually be real mathematicians starting at an early age. This is not a religion, nor is it exclusive to "the Seymour way".

Over the last 30+ years of my own experience I have been greatly surprised at some of the things children have shown they can do (the 4 year olds at Reggio Emilia, the 6 year olds of Julia Nishijima, that 5th graders could do the Galilean gravity project I designed for 9th graders, etc.). Basically, we still don't really know what children can learn at different ages if the subject matter is properly formed. The experiments are very difficult to do, and lots of them need to be done (in part because there are so many things that can prevent a good reading of the children).

Fri Aug 17, 2007*: the non universals*

Bob Taylor (the ARPA funder who later set up Xerox PARC computing research) was an absolute master at this.

Sun Aug 19, 2007*: the non universals*

Hi --

Yes, I used to visit the IAHP quite a bit 15-20 years ago. They have some excellent insights, and are certainly optimists about what babies and young children can learn to do.05:02 PM 8/18/2007, Blake wrote:

Sun Aug 19, 2007*: the non universals*

Hi Bob --

At 02:44 PM 8/18/2007, Robert Parks wrote:
I've been listening with interest, and I've got a couple of questions and (possible) provocations.

> *1. would learning calculus as a "powerful idea" (rather than through the duller algebraic approach) be counted as "using discovery or inquiry based learning as a substitute for hard facts"?*

I don't see why it should, but there are few bounds on rhetoric and innuendo. I like Bruner's term "scaffolded learning" because real discoveries are rare -- we've learned how to teach 10 year olds a good and mathematical version of calculus but no child has ever discovered calculus without guidance (and it took 200,000 years for two smart adults to do it with hints). Much of the "discovery and inquiry learning" curricula I've seen is pretty soft.

But learning and teaching would be easy if it could be transmitted by words or actions. Instead, some changes have to happen in the learner's mind/brain through some actions on their part (which could involve doing something or just sitting in a chair pondering). Things are sometimes not obvious because they are literally invisible, or because the explanations fall outside of existing commonsense thinking patterns. Or some new set of coordinations have to be learned/built that were not there before.

These have many of the trappings of creativity and the having of ideas that are not simple increments from the ideas of the surrounding context. The phrase I use for this is "Learning a powerful idea requires a lot of the same kinds of creativity as it took to invent it in the first place". This is because it has to be invented anew by the learner.

The good news is that learners for already invented ideas almost never have to be as smart and unusual as the original inventors (calculus can be learned by pretty much everybody, but Newton and Leibniz were unusual). On the other side, some real work has to be done to "cross the barriers".

Tim Gallwey (the incredible tennis teacher) use to say: you have to hit thousands of balls to learn to play tennis -- my method gets you to hit those thousands of balls, but feeling and thinking differently. A good method in mathematics (like Mary Laycock's or Seymours) still requires you to do lots of things (to get your mind/brain fluent) but can be and feel mathematical for most of the journey rather than painful in many ways. This is what we've called "Hard fun", and it is a process that is shared by any set of arts/sports/skills that have been developed.

Another way to look at it is "If you don't read for fun, you will never get fluent enough to read for purpose".

The big problem with the "standard algebraic route" is not so much algebra, but that the standard route requires lots of work but doesn't deliver "real math" very well. It's not situated in mathematical thinking, but much more in rule learning and following. People have turned Logo (and other computing) into rule learning and following, etc. It can be done to any initially terrific subject.

> *2. What IS a "powerful idea", and how does it become powerful? I'm particularly interested in asking whether ideas get their power from abstraction (finding similarity in structure), or generalization (finding similarity in features) - or from both.*

Seymour and I have tried to characterize "powerful ideas" operationally rather than by structure. Even though there are not a lot of powerful ideas (hundreds or so) there are enough of different types to make simple structural definitions difficult. For example, "modern science" itself is a powerful idea: it is one of the greatest sets of processes ever devised for getting around many of the defects of the human mind/brain/genetic/culture system that has been so confusing and dangerous over our species time on the planet. On the other hand, "increase-by" as we use it in Etoys is the essential building block of the calculus (especially for children) and it is a "powerful idea" because it can be used in so many different kinds of "change situation" and it illuminates the change processes and makes them easier to think about and to calculate.

These two "powerful ideas" are on different scales and in different domains. But operationally they have the power to greatly amplify and channel our thinking processes. A phrase I've used in the past is "Point of view equals 80 IQ points". Choosing and using a context can be like adding an extra brain. This is why today's scientists and engineers -- who are not better endowed by nature to work in their fields -- are so much more effective than some of the great geniuses in the past.

Some of the most important "powerful ideas" can be drawn from Anthropology, Bio-behavior, Neuroethology, etc., (how History can be interpreted in the light of these, etc.) and have to do with insights about ourselves that are critical and have remained hidden for 10s of centuries. Our research project is ultimately about getting children to start learning these, but we decided that we needed to learn how to teach math and physical science (and what kinds of each of these) to children first. Jerome Bruner saw this earlier than anyone and pioneered one of the greatest curriculum designs for elementary school children in "Man A Course Of Study" (MACOS), an intellectually honest presentation of Anthropology to 5th graders. This was implemented in more than 10,000 schools in the US in the late 60s, was a masterpiece, and ultimately was destroyed by religious fundamentalists in Congress.

But it and other deep insight powerful ideas curricula need to be done again, better, and with more support.

Thu Aug 23, 2007*: the non universals*

Of course, Mark didn't look carefully enough at either the Squeakers DVD or the Kim Rose and BJ Conn book "Powerful Ideas in the Classroom" and other materials which show what we actually do with the kids (actually in 5th grade for this example).

We don't teach any abstractions, but work our way out from various kinds of animated movement in Etoys (constant velocity, random velocities, steadily increasing velocity, etc.). From a number of such examples the children gradually associate both a relationship "increase by" and a history of the movements (shown by leaving dots behind on the screen). Later (about 3 and one half months later, in the case of the first time we tried this) we got them to think about and investigate falling bodies. One example on the Squeakers DVD showed 11 year old Tyrone explaining just how he worked out and derived the actual differential equations of motion (in intellectually honest and mathematical version that computers make very practical). He did this by recognizing accelerated motion in the pattern of pictures of the dropping ball, measured the differences to find out what kind of acceleration (constant) and made the script for vertical motion partly using the memory of how he had done the horizontal motion in Etoys 3 months before. He explained how he did this very well on the video. Also, by luck, I happened to be in the classroom on the day he actually made his discoveries and derivations. Most the children were able to do this.

The important things about this experience was that Tyrone and the other children had learned a model of acceleration and velocity that was quite meaningful to them. Months later they were able to remember these ideas and adapt them to observations of the real-world. According to Lillian McDermott at the U of Wash, 70% of all college students (including science majors) are unable to understand the Galilean model of gravity (which uses a very different pedagogy in college).

The most important piece of knowledge from cog psych is a study done in the late 60s or early 70s that showed exposure to any enriched environment for less than 2 years was not retained. But two or more years of exposure tended to be retained. This also correlates to habit formation and habit unlearning.

So, I would argue that Mark's three examples are very different and don't really deserve to go together. And, in any case, all we know about the 5th graders is that using this pedagogy and Etoys they are generally able to be more successful in both the math and the science of accelerated change than most college students. This particular way of looking at differential equations has become more and more standard as computers have become more and more the workhorses of science (partly because they are in a form well set up for creating a simulation -- and for the kids, because they are much easier to understand than the previous standards for DEs).

Fri Aug 24, 2007*: the non universals*

And they are concrete in the way they are precisely because children of this age don't generalize the way older children and adults do, but by "carrying a bushel basket of 'similar things that work similarly' ". They are not patterns from the outside but are more like analogies that the child gathers together from doing many kinds of thing with a powerful idea like "increase by". Later the bushel basket starts to become an idea of its own, first as a heuristic to try when thinking in problem solving, and finally by enlarging itself into a kind of thing on its own. This is interestingly like Vygotsky's theory of concept formation in much younger children, but the resemblences could be accidental.

Fri Aug 24, 2007*: the non universals*

But here's where we should give the special cases amongst the much maligned (and quite a bit for good reason) teacher corps great credit. Every once in a while a great teacher does light the fires, and those that are affected by this never forget it. To me this is the way it should be, because many children are close to the intense interest that you describe, and contact with another person can be just what they need to give them a little more confidence and courage, to get them to look at something a little closer.

I've had just a few of these, but they were huge experiences in my life.

Fri Aug 24, 2007*: the non universals*

Hi Mark --At 05:01 AM 8/24/2007, Guzdial, Mark wrote:

Snip

Tyrone is eloquent in his explanations--I believe he understands what he's doing. Here's my concern: Does he really understand differential equations? Let me break that down into two parts.

- When Tyrone is faced with another problem related to rates (maybe disease propagation, rates of decay, etc.) in eToys, can he use those tools to analyze the new situation?

I think most of the children after a few months of using "increase by" in various ways, do recognize rates in many other contexts.

Does he recognize the situation as similar and that his same tools would apply? That would convince me that he has developed an understanding of the powerful idea of differential equations.

I would doubt that his understanding of these kinds of DEs is total or even "supremely comprehensive", but it is "operational" very along the lines that any mathematician would characterize as "mathematical thinking". Our goal was to make an environment in which more than 90% of the children exhibited real fluency in this kind of thinking. "Real fluency" implies a degree of understanding above an important threshold.

- When Tyrone gets to college and studies differential equations, will he recognize them as the same thing? I doubt that. They won't look the same.

A much more important question is "will Tyrone understand mathematics by the time he gets to college?". If the answer is "yes", then he will recognize them as the same thing. If "no" then everything will be special cases of rules (which they are to most college students).

His calculus course may not even relate to differential equations to modeling gravity. He will have too few cues to make that connection. A reasonable response to this should be that the calculus course might be taught with eToys, too, and that would help make the connection. I would agree. It's just unlikely that many (any?) college calculus courses will use eToys.

Again, the question is whether he is actually learning math or not. It has nothing to do with Etoys.

*What I do believe is that the students in BJ's course have developed an understanding of the power of computation (*programmable* computation) in problem-solving and knowledge transformation. That's tremendous, and likely will transfer to other situations using computers.*

I'd like to argue with your claim from cognitive psychology, though. "Length of exposure" is an ill-defined variable which has since been better refined and tested. What does "length of exposure" mean? One hour a day for two years? One hour a week for two years? Here's a brief thought experiment to address this point: I'll bet everyone on this list remembers exactly where they were and what they were doing when they first learned of the 9/11 attacks. That wasn't a very long exposure, yet everyone remembers it. Why?

All I can say is that this was very thoroughly studied in the 60s (as was deep habit formation). What they were testing were not memories of isolated unusual incidents (nor of "movie recognition memory" which is also from one trial). What they were doing was testing changes of paradigms in outlook, and for most children these took immersion in an environment for well over a year to be strongly detectable years later.

The two new variables that are more often studied are: - Time on task. The more time you spend on an activity, the more likely that you will remember the experience and lessons of that activity later. - Amount of reflection. The more often that you reuse an association, the more often you think about and talk about an experience, the more likely you will retain it. That's the best explanation I know for the 9/11 effect (or the Challenger effect, or the JFK assassination effect). You thought about that moment later that day, and the next day, and you've discussed it with your friends. That leads to longer term learning.

To me, these are not as interesting (nor are they parallels) to large scale epistemological shifts.

Sat Aug 25, 2007*: the non universals*

Good question --

This is not really about Etoys but about what it takes to make use of a variety of perspectives on ideas in math, science (and elsewhere).

One of the big insights of Seymour Papert was that an incremental discrete form of differential equations that is extremely simple but computationally intensive would fit very well with the kinds of thinking that children can readily do. Babbage was one of the first who proposed that "these calculations should be executed by steam" because he realized that machinery could open up this way of looking at calculus.

Gauss upped the ante considerably by being one of several top mathematicians in the 19th century who moved geometry from a global to a local perspective. Papert realized that the child had this "coordinate system" of having all changes be relative to them wherever they were. And the additive form of DEs also applied here if you used vectors (and that vectors were a very good internal way to think about numbers).

Seymour proposed that you could use an interactive computer to make a "Mathland" in which a powerful mathematics could be situated as the way to talk about and cause phenomena of interest to a child (and most importantly to start building some ways to think about things in ways different than stories that would eventually constitute a new outlook on both thinking and phenomena).

So the key idea here is made of several important insights that include new ways to look at things, but also to make them happen. This last has partly to do with emotional payoff. For example, in the case of Galilean gravity it is possible to use something like Galileo's lute strings (see the afterword in BJ and Kim's book) or e.g. rolling a toy truck carrying a baggie filled with ink with a hole in it down an inclined plane to get the constant acceleration spacings that lead to the two stage incremental relations. This can all be done without a computer, but it is much more difficult to motivate the level of precision that we want the kids to employ, and to provide a vehicle for both checking their analysis (this is supposed to be science after all), and to make really fun things that now use the gravity model (like Lunar Lander, firing a cannon, shoot the alien, etc.). This is supposed to be fun after all.

Etoys is just one of a number of approaches done by people who got really interested in Seymour's insights (and Etoys itself is actually an amalgam of the ideas from many contributors outside of our immediate research group).

Right now, to get above threshold science and math, we need highly motivated teachers like BJ. But if the highly motivated teacher does not have an environment that situates the ideas and approaches (and curricula) then many (if not most) important things won't happen (except perhaps for a very few children).

An even rarer case is the highly motivated teacher who has a deep understanding of the subject and of the learners. For example, Julia Nishijima of the Open School (of whom I've written about elsewhere) showed what could be done with 6 year olds, and it is really impressive. Her curriculum was "almost perfect" in balance and depth. A small part of this curriculum used the computer (again for what only the computer could do as an "educational material").

If we look out in the world, in the US, Europe, Asia, and much wider, we do not find enough adults who can carry the powerful ideas of math and science and help children make them their own. This is especially acute wrt parents, because here we have a much better "student-teacher ratio" and we also have a great social environment for learning. Quite a bit of success in children learning to read has quite a bit of correlation with how parents deal with reading in the home. It would be great if this could be true for "real math" and "real science".

So utopian enterprises like OLPC really need to think about using the computer not just for an environment, but as a guide (something "better than no teacher and better than a bad teacher"). This is perhaps the most important and high stakes way to interpret "the computer as a dynamic book" (that is it could be a kind of book that can also teach people how to read and write it).

I think of this as one of the great and most important "Grand Challenges" for the 21st century.

***Mon Aug 27, 2007**: size of palette in OLPC version*

Hi Randy --

You can set a preference to paint full screen. This has always been there, it's just that we set limited paint area option for the OLPC release.

***Thu Aug 30, 2007**: some comments*

Hi Bill --

What follows is not an argument against (a) using a game making approach (great for some kids) or (b) the worth of getting all children to learn to program (probably a good idea).

What we need to contemplate is the probability of "gaining enlightenment" by being in contact with various kinds of environments and epistemologies.

If we look at the class of "those who know how to program" we see a generally unenlightened group (maybe similar to humans in general, maybe even less enlightened). In any case, we have to conclude there is nothing intrinsic about learning to program that leads to deeper thoughts. One of our rueful jokes about the Logo vogue in the 80s is that everything would be OK if we could just package Seymour on the floppies!

I think it would be even easier to justify the same generalization about gamers and game makers.

Or about archers ... There was an intriguing book in the 60s zeitgeist called "Zen and the Art of Archery" by Herrigel, that made a similiar point: learning archery doesn't confer any automatic enlightenment, but it could be used as a path if much were added to standard training.

Or about any activity that requires concentration, focus and learning. My grandfather Clifton Johnson (a writer and illustrator of many books and also an early photographer) once got asked in 1904 to write an article for the Saturday Evening Post on whether photography could be an art form. He said "Art enters in when one labors thoughtfully over a goal; that is, when one cuts loose from actions that are merely mechanical". It's in that space of "laboring thoughtfully" where there are opportunities for enlightenment.

"Enlightenment about what?" brings up the environmental influences. I don't think that archery or cooking (or photography) are cosmically interesting -- so the kinds of enlightenments in these environments are likely to be personal ones, but with some flashes of "the world is not as it seems". On the other hand, if science is the environment, and one is dealing with its huge epistemological differences with commonsense perceptions -- that is: science has much of cosmic significance in its purview -- then there are great and deep opportunities for enlightenment. (But no guarantees here either, just higher probabilities.)

My interest in education is not as a form of vocational training or preparation, but in helping children to become adults who are more thoughtful, and whose perspectives for thinking are much wider and deeper than the adults of today. The built-in "universals" that are destructive to human growth can be countered to a considerable extent by a modern "real education" that includes powerful invented points of view (the "powerful ideas") that act almost as additional brain/minds and can form a much stronger and less brittle heuristic base for thinking well under wider conditions.

In the end, the epistemology of science can lead to many more and better perspectives on the human condition, and this is where I think education should and must go. Whether children learn computers or not is not the issue for me (nor even whether they gain math or science knowledge) -- it's whether they can gain clearer perspectives on "us and what to do about us" that is critical here.

The established Arts -- including writing -- have as one of their main properties to provide other perspectives and wake-up calls, but they have been less effective than one would hope: they are generally too easily overwhelmed by distracting media, and they have enough story elements that they tend to be compartmentalized (as is the natural case with stories). On the other hand, even though our brain/minds want to make stories out of everything (and judge them by how apt they seem), science stubbornly tries to rise above our "storyminds" to help us make representations of the "what's out there" that are much more accurate "maps and models" (especially including accurate maps and models of ourselves). This is what we need to concentrate on when trying to design new educational experiences.

I don't think we are doing very well at these grand goals for education at the moment, but we haven't forgotten them in all the technical flurries that accompany the invention of new media to hold new ways to look at important ideas.

***Mon Sep 24, 2007**: Randy Pausch gives last lecture*

Randy is a great guy -- no professor has done more for his students in every way.

***Fri Oct 5, 2007**: Is our fondness for Cuisenaire Rods just nostalgia?*

Hi David --

I was using the actual Cuisenaire Rods as an analogy to the need for concrete manipulables for magnitude and sense. When we found them in early classrooms we painted arrows on them to turn them into vectors and got the kids to start thinking about concrete ideas of accumulation in dimensions 1, 2, and 3. The best thing to use today is an extension of the original Rods ideas that aims very strongly at geometric vector representations for arithmetic and many other branches of mathematics.

A general comment about educational fluency is that most of the best ideas in education in powerful ideas for young minds go back more than 100 years -- and my opinion is that the best ideas of the last hundred years are enough to make great curricula for children today. This is not a conservative statement, but a radical one. (Almost none of the best ideas from the last 100 years are in most classrooms today.) 05:20 AM 10/3/2007, David Corking wrote:

Tue Nov 6, 2007*: Re: OLPC mass production started*

Yay!

Cheers to all!

Alan

Sun Nov 11, 2007*: Re: A Hero for One Laptop Per Child (RE: OLPC mass production started)*

I would be very disappointed, even appalled, to see my name as a hero here. The notion of a "Hero" seems to be partly built into human nervous systems and is a favorite trope in stories. This idea seriously distorted and masked how Xerox PARC actually worked, for example.

OLPC has been making progress because quite a few talented people decided to take responsibility for different needs of the project. I am very proud of the Viewpoints researchers who really got behind this because they believed in it.

So there are a lot of heros, if that is the word. I think of it more as "there are a lot of enlightened people" and it bodes well for humanity when they decide to take action.

Cheers.

Alan

Wed Nov 21, 2007*: Panel discussion: Can the American Mind be Opened?*

Hi David --

I think "constructivism" (like "object-oriented") has been so appropriated and redefined as to be a useless term today (I certainly don't know what it means -- and am not sure I ever did).

I think it's much better to simply try to puzzle out the nature of the desired learning, and then to find workable pathways for the different kinds of learners. This has been done quite well in (admittedly simpler) areas like sports and music, where the learner has the distinct advantage of being able to watch practitioners and gain some idea of what the subject area is all about and what might be fun and rewarding about it.

Two of the biggest barriers to math and science learner are (a) the prospective learner has very little idea what the activities actually are (though they still might think they could be "cool" because of "rocket ships", etc.), and (b) there are so few real practitioners around (available) to help give them a sense of how to get started. Schools (and many adults) introduce another barrier, which is a profound misunderstanding of what it means to be fluent in math and science (the misunderstanding is usually in the form of thinking that math and science are fact and pattern based, and that learning the facts and the patterns is what is required). I've used the " 'music appreciation' instead of 'music' " analogy for this misunderstanding.

Once we get some sense of what "the doing of math and science" is all about, the main question remaining is "for learners, what is the best balance between doing and being advised that should be set up?". This is pretty well understood for sports and music for both children and adults. As Tim Gallwey once said about teaching tennis: "the main problem with traditional tennis teaching is that the parts of your mind that learn to play tennis don't understand English!" (Of course he meant that a little English goes a long way, and a lot of English simply can't be translated into tennis action.)

It is almost certainly the case that different subject matters (and different learners) can tolerate more or less of direct instruction in English, so it is worthwhile to get a rough assessment of this when trying to invent a curriculum. However, I don't think it is controversial to state that learning to play music or tennis is really about lots of actual guided and coached (and uncoached) doing of the activities. Most mathematicians would agree about math learning, and most scientists would agree about science learning.

If we look at human history, we can see that "pure discovery" learning by children or adults usually results in weak ideas. On the other hand, rote learning usually doesn't work very well for any subject that has some art to it. (Playing lots of scales or memorizing chord progressions does not a musician make.)

So there has to be discovery and creativity of a sort, and this is done by good teachers and writers as a kind of "guided discovery" (sometimes by great environmental design as in classic Montessori education). Perhaps the most wonderful thing about human learning is that something that required a genius to invent or discover (like calculus) can very often be learned by non-geniuses if given help. One of the best accomplishments of the Etoys work over the years (and reaching back to Seymour) is that, while no 10 year old has ever invented calculus, we now know how to help most 10 year olds get fluent in a number of the most important ideas in calculus. This is real progress.

I think science is the most difficult of the "new thinking" to teach and learn because it is the farthest from normal commonsense perception and thinking. It is also the most critical of human reason because the nice crisp logic of math is only approximately mapped to considerations of the actual universe (it doesn't have to work like our current math or brains). So just what "doing science" should mean for children is not nearly as clear as for sports, music or math. I think that the "Galilean Gravity" project that is done so well by 5th graders is an excellent example of one of the "real science" activities children should be doing. But I would be surprised if it and projects like it are comprehensive enough to cover all that is needed. Part of the internalizing of the epistemology of science seems to come from so many examples from so many parts of science that show "the world is not as it seems", but also allow some pretty powerful generalizations to be drawn about many of the non-intuitive workings of the universe.

One of the paradoxes about many kinds of learning is that you can learn a lot about a subject by reading after you have learned the subject pretty well by lots of doing. But the subjects we've been discussing are not often (if ever) learned above threshold without lots of doing to provide a foundation of deep understanding for later listening and reading.

***Thu Nov 22, 2007**: Educational research*

Hi Folks --

Books are a real technology. Most people think that classrooms would be less rich without books and the literacy of reading and writing about ideas. (I do too.) And very few would disagree with the idea that the fruits of the printing press were one of the largest and most important forces in bringing forth our modern era. Yet, in the US where classrooms do have books, and there are free public libraries in most towns, education is failing. Should we blame the book or should we blame the classrooms and what's behind them?

One of the deepest built-in traits of human beings is "magical thinking" (superstitions, rituals, similarities, contagions), elements of which are found in most human behavior. This is reflected in many parts of education e.g the correct rituals will cause it to happen, or the proper effigies and/or contact with substances will cause it to happen. This is what "air guitar" (and much of fashion) is all about. It's always been a problem, and is likely worse today because the combination of media and pop culture is almost overwhelmingly focussed on form rather than content.

Some studies on the actualizations of personalities suggest that the decisive step is to take responsibility for what's necessary to turn a fantasy into actuality. In the US this has moved from a problem of individuals to a problem of the entire society.

***Fri Nov 23, 2007**: Need advice about teaching programming to children*

Hi --

The biggest difference between elementary school children - and teens and adults - is in their ability to make plans and carry them out -- quite a bit of this seems to be developmental and thus somewhat related to age (where ages 11-13 are a pivot point between one kind of planning and more elaborate plans). A second developmental difference is in how certain kinds of abstractions can be learned and used -- one could easily divide up the elementary years into 3 or 4 categories based on the kinds of abstractions and the forms for them. Some of these results have been used in both Scratch and Etoys to achieve a better cognitive fit.

The most important questions for you to ask yourself have to do with your ultimate goals for teaching programming to children. Programming can be a route towards learning lots of powerful ideas and thinking processes, but it is not sufficient all by itself (the class of programmers today doesn't appear to be necessarily very enlightened or knowledgeable about much of anything by virtue of learning to program). So you need some goals, and then some ways to possibly use programming to help children learn what you hope.

***Fri Nov 23, 2007**: Panel discussion: Can the American Mind be Opened?*

Here's what we did for City Building, Playground, and then Etoys (I've written about this before, but I don't think on this thread). "City Building" is a wonderful (at that time non-computer) curriculum designed by Doreen Nelson that is very rich and has been used successfully for many age ranges - in our case we implemented it with Doreen's help for 3rd graders - which was the youngest group tried up to that point. Google Doreen and "City Building" for a wealth of info on this terrific curriculum design.

Playground was a different way to do Etoys (similar graphics model and a different programming model). This was implemented in a grade 4-5 classroom (the school didn't have grades by age, but "clusters" by developmental level - which works a lot better).

Doreen helped in every step of introducing "City Building" to very willing "3rd grade" teachers. Still, it took 3 years before the deep quality in the curriculum was manifest in the classroom and in the students and what they did and how they did it. Photographs of each of the three years would not reveal much visible difference. It was what the children were concerned with, how they talked about it, and how they went about the processes that changed profoundly. Trying to trace all this back into "what happened?" we came to the inescapable (and not too surprising) conclusion that the teachers had also changed -- they had learned much more about design and systems over the three years, and this was manifested in a "well above threshold" assessment from Doreen and the rest of us in the 3rd year.

It's worth noting that assessments of fluency do not require control groups because what is being judged is not a teaching method or a curriculum per se, but results. Were the children doing deep "City Building"? No for the first two tries, Yes from the 3rd try onwards. Similarly, "are the children doing real math and real science or not?" Questions like these are easily answered by people who can tell the difference (just as musicians and coaches can assess their learners for degrees of fluency).

The City Building experience and our long stay in this school allowed us to try the same multiple year assessment for Playground programming and its curriculum (with similar results). Basically, there are just a lot of things that don't get normalized in single trials of even worthwhile curriculum ideas that get smoothed out over a few years. The teacher gets more knowledgeable and confident. The curriculum is improved from some of the bugs found. The software often requires tons of work over the three years before it is above threshold, etc.

When we started on Etoys 10 years ago, we had the three year trial in mind, and decided that all the initial curriculum would be tested over three years before we wrote it up (the substance of Kim's and BJ's book "Powerful Ideas in the Classroom" is about a dozen projects, each of which was tested over three years).

What we don't know from this methodology is whether there are better ways to teach Etoys and the math and science powerful ideas in these examples. And we don't know whether the choices of the math and science examples are the most appropriate. But what we do know is that the processes of their book are highly likely to result in more than 90% of a class of children getting fluent in what's in the book, and that includes strong elements of differential vector geometry, acceleration and Galilean gravity, etc.

This leads to interesting arguments, especially wrt young children, of the kind "if you can get 10-11 year olds to do real math and real science, then it doesn't much matter what the specific subject matter is". And "if the specific subject matter can be strongly related to adult uses and thinking about real math and real science, then all the better".

This bypasses the much more difficult problems of taking a given theory of subject matter (school maths, etc.) and trying to contrast different ways of teaching it. We do not do that at all, and the Etoys work was done as part of "science time" in these classrooms (a great place to teach real math given the difficulties with the school math goals and processes).

The main point here is that above threshold fluency for 90%+ of the children is one of the most important benchmarks -- and it can be done a little more easily than trying to use specific control groups if the subject matter is very different from school theories, yet still recognizable by experts.

A side comment. The reactions against "the new" take partial form in demands for "super scientific studies", and most of these are simply not feasible, if our "three years for a good experiment" is valid. But the largest most devastating studies in the US are the "whole country" results that show beyond a shadow of a doubt that the existing educational process is not resulting in more than a small percentage of children getting above acceptable thresholds in reading, writing, math and science (and thinking). This is the problem they don't want to even discuss. Contrastive studies are not interesting unless both are above threshold. If neither are, back to the drawing board. If one is, then a more detailed contrast is of little value.

Fri Nov 23, 2007*: Panel discussion: Can the American Mind be Opened?*

Hi David --

At 05:29 PM 11/23/2007, David Corking wrote:
It was not my intention earlier in this thread to challenge the work of Viewpoints.

I certainly didn't take it that way - in part because we claim almost nothing. What we have been interested in is whether 90% of the children we've worked with -- taught by a teacher, not by us -- gain real fluency in what we are trying to teach them. We found that it took 3 years to introduce each new curriculum element (as described in my last post).

Instead I wanted to get a foothold into understanding how the powerful 'progressive' and 'back to basics' movements could be rationally compared with alternatives.

I disagree with the simplistic versions of both of these. If "progressive" means what it meant long ago - "Dewey education" - then I am very much in favor of what he was trying to do and what he wrote about. If "back to basics" means "Bennet or E.D. Hirsh", then I'm very much in disagreement with what they are trying to do, and their general view of "education".

Subjects like real math and real science, with a goal to help children get fluent, are best assessed by real mathematicians and real scientists. Separate issues are: what parts of the real stuff should be taught to children, how should the teaching be done, etc. This is very important in its own right - recall the very bad choices made by real mathematicians when they chose set theory, numerals as short-hand for polynomials, etc. during the "new math" debacle. This is why Seymour Papert was so impressive -- he was that rarity, a first class mathematician who both cared about and understood important principles of how children think. He chose real math that was both deep and in rhythm with how children think about relationships.

Thank you for taking my question as a provocation

I didn't

- it is very illuminating to read the work of Rose, Kay et al justified from this perspective.

I need to confess now that I have read 'Mindstorms' but not yet 'Powerful Ideas' - does the book address whether or not there is a 'Hawthorne effect' in the trials?

"Powerful Ideas" is written to help teachers teach a dozen or so projects in real math and real science, using Etoys. It makes no claims and leaves a tiny bit of philosophy to the Afterword.
http://www.vpri.org/pdf/m2003001_human_cond.pdf

In other words, could simply the intensive attention of all involved, coupled with the novelty, willingness to persevere for the second and third year, and the involvement of real subject matter experts, have been sufficient in itself to produce a fluency result that is well above acceptable threshold?

Schools should be all about the Hawthorne Effect. The ones that aren't should be closed.

I think you misunderstood one part of my description of the process. The 3 years is with the same teacher but with three different groups of children. Each group deals with the materials and process for the same amount of time.

The other part of your question wasn't asked or answered by what we did (since we wanted the children to express the math and science they learned in terms of working Etoy models). That's what we tried to do, and that's what we assessed.

If the "it takes 3 years" story seems reasonable to you, then imagine what it would take to do a real longitudinal transfer experiment using control groups (about 7 years). We have never been able to find a funder that is willing to fund what it really takes.

Is it provable(*) that the student creation of computer models, for example, is a necessary condition of learning 'real math' fluency?

It's provable that it isn't (people have been learning "real math fluency" for thousands of years without computers). The important thing (Papert again) is what math and when? Computers make a huge difference here for pretty much everyone. Also, see the Afterword in the book for what science learning is really about (hint: computers are not at all required, but they allow more rich choices in the world of the child).

I've used many analogies to music in the past. You don't need musical instruments to teach music, they just help (and in no small part because there are lots of different kinds). A child who is not that interesting in singing might be very interested in learning the guitar, one that is not interested in guitar might be interested in a sax, etc. Different learners need lots of different entry points. Computers can provide many different entry points, and can be the medium for the kinds of mathematics that science uses. A pretty good combination.

** By 'provable', I mean: "could a future experiment be designed to prove my assertion, or, even better, could a reasoned argument prove my assertion?"*

No. But something might be done with a goal of more than 90% fluency -- computers could almost be indispensable here ...

Further, but perhaps drifting off topic for squeakland, is it provable that 'back to basics' and 'progressivism' are equally as inadequate?

I said above that the simplistic versions of both are quite wrongheaded in my opinion. If you don't understand mathematics, then it doesn't matter what your educational persuasion might be -- the odds are greatly in favor that it will be quite misinterpreted.

Or is the poor performance of public education in some countries a consequence, not of the learning theory nor curriculum, but caused by the 'received wisdom' not being applied properly, or even some external factors, such as low resources, attitudes to authority, or the currently fashionable complaint of students' learning styles not being catered for?

If you like multiple choice tests, then (e) all of the above.----------

David

***Sat Nov 24, 2007**: Panel discussion: Can the American Mind be Opened?*

Hi David --

At 02:25 AM 11/24/2007, David Corking wrote:
Hi Alan, You digressed into 'new math' and I disagree

You wrote:
>
> Separate issues are: what parts of the real stuff should be taught to
> children, how should the teaching be done, etc. This is very important in
> its own right - recall the very bad choices made by real mathematicians when
> they chose set theory, numerals as short-hand for polynomials, etc. during
> the "new math" debacle.

When I was 16 I moved schools, and joined a cohort who had been educated in the Schools Mathematics Project, a English incarnation of 'new math'. I had kind of a traditional classical math education up to that point, and I felt like a fish out of water for a few weeks. My first impression was that my new classmates thought much more like real mathematicians, and at first that seemed like a pointless stuffy homage to academia.

Of course, I was referring to elementary school new math in the US, which tried to teach arithmetic via set theory and polynomial bases for different numeral systems. It would not be at all surprising if the SMP were better.

The point is not about the worth of set theory and number theory (both good topics for high school) but about whether they are appropriate for younger children. I have degrees in both pure math and molecular biology, and I agree very strongly with Papert's view that various kinds of geometrical thinking, especially incremental, are better set up for children's minds, and also allow deeper mathematical thinking to be started much earlier in life.

One way to think about this is that "mathematical thinking" (like musical thinking) is somewhat separate from particular topics - so the idea is to choose the most felicitous ones.

Later I learned to enjoy the math for its own sake, but I had another surprise a couple of years later. The SMP kids seemed much better equipped for the world of applied math at university and technical college. Set theory and number theory are vital for computer scientists (as I understand), matrix algebra and numerical methods for engineers. So when I got to college (to study engineering), I was glad to have had a chance to try my hand at real nineteenth century math in high school.

By the way, I never learned, even today, any kind of general algebra or shorthand for polynomials, so I cannot comment on that.

I think you did, since "356" and all other numeral forms of numbers (whatever their base) are shorthands for polynomials (the 3, 4, and 6 are the coefficients for polynomials of powers of ten in this case).

It didn't hurt that in those days, most math teachers in England were math major graduates (so perhaps an example of the benefits of the Hawthorne effect we discussed.)

Why call this Hawthorne? I don't think this is what you mean here.

***Sat Nov 24, 2007**: Educational research*

At 07:32 PM 11/23/2007, subbukk wrote:
On Thursday 22 November 2007 11:26 pm, Alan Kay wrote:
> Hi Folks --
>
> .... Yet, in
> the US where classrooms do have books, and there
> are free public libraries in most towns,
> education is failing. Should we blame the book or
> should we blame the classrooms and what's behind them?
It is a quantity vs. quality issue. Books are indeed numerous and available, but well-written books are hard to find.

That's what libraries are all about, and why there are lots of books on all important topics. Different learners need different points of view.

I recollect my frustration in "understanding" electromagnetism during my school days till I stumbled on Maxwell's original article in a edited collection stashed in a dusty corner of a library. Here, at last, was a spirited presentation written without recourse to circumlocutions, jargons or acronyms.

Technology can help in keeping such "books" in active circulation and make the term "out of print" obsolete.

I agree -- and, it's worthwhile thinking of libraries as an example of such a technology as well.

Sat Nov 24, 2007*: Panel discussion: Can the American Mind be Opened?*

Thanks Bill --

I think you make the central point about all this.

Sat Nov 24, 2007*: However ...Re: Panel discussion: Can the American Mind be Opened?*

Hi Bill --

I just read Professor Wu's paper. I agree in the large with his assertion that the dichotomy is bogus, but I worry a lot about his arguments, assumptions and examples. There are some close analogies here to some of the mistakes that professional musicians make when they try to teach beginners -- for example, what can a beginner handle, and especially, how does a young beginner think?

Young children are very good at learning individual operations, but they are not well set up for chains of reasoning/operations. Take a look at the chains of reasoning that Wu thinks 4th and 5th graders should be able to do.

Another thing that stands out (that Wu as a mathematician is very well aware of at some level) is that while people of all ages traditionally have problems with "invert and multiply", the actual tricky relationship for fractions is the multiplicative one

$$a/b * c/d = (a * c)/(b * d)$$

which in normal 2D notation, looks quite natural. However, it was one of the triumphs of Greek mathematics to puzzle this out (they thought about this a little differently: as comeasuration, which is perhaps a more interesting way to approach the problem).

A few years ago I did a bunch of iconic derivations for fractions and made Etoys that tried to lead (adults mostly) through the reasoning. One of the best things about the divide one is that it doesn't need the multiplication relationship but is able to go directly to the formula. So these could be used in the 5th grade.

But why?, when there are much deeper and more important relationships and thinking strategies that can be learned? What is the actual point of "official fractions" in 5th grade? There are many other ways to approach fractional thinking and computation. I like teaching math with understanding, and this particular topic at this time - and provided as a "law" that children have to memorize - seems really misplaced and wrong. Etc.

Mon Nov 26, 2007*: However ...Re: Panel discussion: Can the American Mind be Opened?*

Hi Bill --

I think the main thing in teaching "number" is to distinguish it from "name" or "numeral" -- and I think the rush towards teaching "base 10 numerals" too early is one of the big problems in early elementary mathematics. Numbers are ordered ideas that can be put in correspondence and taken apart and recombined at will. Names and numerals are symbols for these ideas that have varying degrees of usefulness for different purposes. So one of the biggest questions any math educator should ask is: what symbols should I initially employ for numbers to help children understand "number" most throughly?

Most "child math" experts - like Mary Laycock, Julia Nishijima, etc. - would argue that a wide variety of analog (both unary and continuous) representations should be employed together (bundles of sticks, bags of objects, lengths of stuff, etc.), and each of these can have several labels attached ("one", 1, etc.). These can stay in use for much longer than is usually done in school. E.g. Some really great "adding slide rules" can be made from rulers, and then the children can make some really detailed large ones (even using their playground for baselines). These adding slide rules can add any two numbers together very accurately, whether "fractional" or not, and they can have scale changes to reveal what is invariant about two numbers (their ratio), etc. This can be used to make multiplication machines, etc.

Another use of number that uses names in a non-destructive way is the "equality game" of "how many ways can you make a number". First graders are very good at this an even though they don't know what "1000000" stands for (except it is large) they understand that they can make this or any other number many ways by a combination of additions and subtractions that add up to zero. This is a way to start algebraic thinking without needing variables. And so forth.

Wu actually makes a point against himself when he argues that phonics decoding is a good idea, even though no fluent reader decodes. This is similar to how sight reading is taught, especially for keyboards. Eventually the pattern results in a direct hand shape and mental "image" of the sound (or for text reading, a mental image of the idea). The question is how to get there, and teaching how to decode seems to help a little in early stages (maybe even just for morale purposes) rather than trying to teach either like Chinese characters. It takes 2-5 years to get fluent at such learning, so there usually need to be other supporting mechanisms (not the least is material that can be dealt with successfully after a few months or a year).

So, what Wu should be asking is "what framework do children need to get started in number and mathematical thinking about number?".

Another interesting example of what is not happening came out in a Mary Laycock workshop in which I was a "floor guy" (literally since I was on the floor with the children). One of Mary's games was to hand out a series of sheets of 10 by 10 squares, each divided in regions, with the question, "how many squares are in each region?" The 4th-5th grade children start by counting the squares in the regions. As the regions got more complicated, the children did not see that they could switch over to geometrical reasoning -- to see what fraction of the whole was occupied by each region and then divide -- instead they kept on trying to count the little squares and fractions of squares. Children who had learned to think mathematically would have had a strategy to look for the best representations for the problems, and these children had not acquired many (if any) math meta-skills.

To bring up a musical analogy again ... one of the best collections of advice about how to teach children to play the keyboard is in Francois Couperin's 1720 treatise "The Art of Playing the Harpsichord". First, he says, keep the children away from the harpsichord because it isn't musically expressive enough. And keep away from sheet music because it "isn't music". Instead, take them to the clavichord (loud, soft, and pitch modulation -- more expressive than a piano) and teach them how to play some of their favorite songs that they like to sing, and help them be as expressive in their playing as their singing is. This is music. Play duets with the children, etc. After they have done this for a sufficient time (from 6 months to several years), then you can introduce them to the initially less expressive harpsichord (which, like the organ, can only be expressive through phrasing). But they will have learned to phrase very naturally from their clavichord experience and this will start to come out in their harpsichord playing. Finally, now that they have learned to "talk" (my metaphor), they can learn to read. Now they can be shown the written down forms of what they have been playing. And now they can start to learn to sight read music.

When I was teaching guitar long ago, I used this basic scheme as much as possible, because "real guitar" has to be both music and "attention out" (so that you can mesh musically in a conversation with other musicians). Also, the guitar has some serious physical problems which have to be addressed gradually over weeks and months. Getting the students to play real stuff while all this is going on makes a foundation for the next level of much harder work. Learning to play patterns by ear allows the player to concentrate on their musicality and accuracy. Then they can be shown the patterns as both shapes and as decoded mappings in members of a key, etc.

The egregiously misunderstood Suzuki violin method also follows these ideas. (It isn't mechanical -- read his books.)

Couperin's essay is a pretty good set of distinctions concerning the general confusions between art and technique, and between ideas and media. You eventually have to get to all of these, but leading with art and ideas tends to preserve art and ideas, and leading with technique and media tends to kill art and ideas. I think it is really that simple.

Mon Nov 26, 2007*: looking for some advice on teaching Squeak to advanced high school kids*

Hi Oscar --

Let's exchange a few emails about this.

First, what would you do if the kids were University students with the same experience? You have a day and a half, and you want to get them to see what is interesting about a dynamic object environment with a metasystem.

How much time (and how to use it) would you allocate to learning the language, debugger, stuff in class library, and metastuff? What kinds of dynamic changes would you get them to do? (E.g. how about changing the shape of objects that are dynamically in use? We once added a few instance variables to Morphic, etc., and it was interesting how well this worked ... .)

A problem with the short time (i.e. let's learn to play piano in a day and a half) is that it will be difficult to come up with a convincing example that is not fairly easy to do in a static early bound language (dynamic languages excel when dealing with difficult complex systems that are hard to debug otherwise). (One of the reasons Simula was not appreciated as it should have been in the 60s was that the example in their ACM paper (that was small enough to put in a paper) was fairly easy to do in Algol -- most people missed that Simula really scaled for many important problems where Algol did not.)

What are your thoughts so far?

Thu Nov 29, 2007*: music and math*

Sounds as though the music teacher didn't understand math and what it is all about.

(By "music theory" he probably meant the kinds of ("keyboard") harmony that are covered in the first year or two of college.)

The kinds of thinking here are a lot like what is done in geometry. For example, one way to think of chords is as shapes whose "side lengths" are measured in semi-tone intervals (the smallest interval between two adjacent notes on a piano keyboard). These "distances" have been made uniform since the late 18th or early 19th century (not without musical penalty, see below).

So a major chord in closed position has the shape 4-3 and a minor chord has the shape 3-4. So a chord of any kind can be built by starting on any note and counting intervals. This scheme normalizes chords in the same way triangles are normalized by their shapes. Scales are normalized in the same way, and thus this also normalizes keys.

In harmonic theory, we are interested in how melodies can be harmonized by adding chords, how a sequence of chords (called a "chord pattern") "works" musically, how movements to other keys and returns can be made, etc. The first order and second order theories of these are very different. There is a famous 18th century piece by Purcell called "The Contest Between Melodie and Harmonie", and this sums up what Baroque music was all about. The golden age of Jazz (roughly, the 20th century until the late 50s or so) followed a very similar pathway in how melody and harmony were thought about (and not entirely by separate invention).

The first order theory is very much about how tensions are introduced and relaxed, how the notion of a "key center" can be used to provide stability (and length) for excursions, how bass lines can be used to solidify movements, etc. The second order theory was used very strongly by Bach then less so until roughly Wagner, and then in highly developed show and pop music (by stage bands, etc.) to try to intertwine melodic devices (like voice leading) with larger harmonic schemes that would force "emergent harmonizations" that are not easily described by the first order theory. What is called "chord substitution" (alternate harmonizations, sometimes of breathtaking beauty) in Jazz heavily rely on such mechanisms.

"Mathematics" is a plural because it is about many different ways of "thinking very carefully" with invented representations and inference rules. So this kind of thinking about music is mathematics (i.e. rather than "like math", it is math). And, within music, there are lots of ways of making generalizations that help with styles.

For example, my pipe organ and harpsichord have the older tuning schemes used in the 17th century. Why would anyone revert? Here's the problem (as first discovered to their horror by the Pythagoreans). Octaves are multiples of 2. The harmonic 5th is the third harmonic, which is a multiple of 3. So if you try to make a scale by running out the 5ths (of 5ths etc.) they will never come back to the original note (2 and 3 are relatively prime). One way of running out the "circle of 5ths" creates a discrepency of about 1/75th of an octave. The equal tuning system mentioned above divides out this glitch evenly by making every 5th a little bit flat (and this results in rather wide 3rds). This works (sort of) OK on a piano because it doesn't have a lot of harmonics and most people are not very sensitive to in-tune-ness. Here, every chord is equally out of tune!

On an organ or harpsichord (which are very rich in harmonics), the result of equal tuning is that major chords don't hold still, and minor chords are jangly. The older tuning schemes made some chords much more perfect and sacrificed others. This results in a harmonic theory that is partly about "sunlight" and "storms" depending on what keys you are playing in and how the harmonic progressions are devised. Because the harmonics are different on organs, harpsichords, clavichords, fortepianos, etc., it is not unusual for each to have a somewhat different unequal tuning to deal with the strengths and weaknesses provided. Some of the greatest music in the world was composed using these different bases for thinking, and much of this music loses much of its meaning in a modern tuning scheme.

And, there is a math to these other ways of thinking about how things go together, but it is a somewhat different math. The analogy is to the many kinds of geometries - all mathematical - that have been devised starting in the 19th century.

Referencing back to "art" and "technique". Learning all this doesn't necessarily make you into a composer or a better player, just as learning painting technique doesn't necessarily produce art. But if the artistic impulses are working then all this technique is tremendously helpful. Unfortunately for education, knowing technique is often all that is asked of a teacher .... oops!

***Wed Dec 12, 2007**: Re: distributing squeak on a network*

It used to work on all platforms ...

***Fri Feb 1, 2008**: Etoys - create duplicate (not sibling) from a script?*

Hi Mark --

We should have done this better.

Here is how I handle things that are "like this" this may or may not work for your case.

If you build a bunch of components on a Playfield it is smart enough to act as a little name-space, so when it is copied all the links between the components will be preserved in the new composite object. (Again, we should do this better.)

So e.g. if you have an object that is controlled by a joystick morph, you can put both of these in a Playfield, copy the Playfield, and the new object and the new joystick will be linked.

***Thu Apr 10, 2008**: Squeak's effect on Open School students*

Thanks BJ!

***Thu Aug 12, 2010**: (SQ-749) and Kathleen's question on "What do you mean by Artifacts?"*

Hi Steve,

A good slogan for teaching, pedagogy, and curriculum design is "When should it be easy, and when should it be hard?".

The notion here is that for education to be transformative, you wind up as a different "better" person than you were, and this means that certain difficulties were very important to your learning -- the stuff that was "easy" you could already do (and some of the easy stuff is not what you want to aim at). Building knowledge and skills to get fluent at handling difficulties (and in some case rendering them non-difficult) is a key for much important learning.

On the other hand, gratiutous non-productive difficulties are to be avoided because they generally both distract and occupy " thinking chunks" that one needs for the important stuff.

(I don't recognize my quote that was paraphrased)

Adults tend to be the biggest problems when trying to help children learn things. It's the adults who generally don't want to do the work and don't see many things as fun. Kids (and people in general) can spend a lot of time focussed and doing when they are having fun.

Constructivism (one of many such terms I don't use because they have lost their meanings) doesn't mean discovery from scratch (this is a huge confusion many people have), but does mean "understand and clarifying by making a careful descriptive model". This can be done with English and writing (it is what descriptive, expositional, and argumentative writing are supposed to be about). It can be done with mathematics. It can be done in many cases with physical construction materials. And a lot can be done in terms of computer programs.

I like the Montessori curriculum approach of making a carefully designed environment for the chilldren that allows choice on their part and allows limited the degrees of freedom on the educator's part. One of the best projects we've ever designed is the Galilean Gravity one -- and it illustrates what you have to do with guidance on the one hand and space to play on the other to enormously raise the probability that most children will be able to see and understand what is going on without having to give them the "answer" to memorize.

One of the keys in the early Montessori schools was the intense comprehensive training of the Montessori teachers -- and the lack of the equivalent of this in most of today's schools is a huge problem.

The scripts in Etoys are independent of the visible appearance of the object they are attached to (the objects' "costumes" can be changed at will -- and this is how "frame animation" is done in Etoys).

I'm not quite understanding what it is that you would like for teachers beyond a repository of projects with extensive notes about how they were made and how to make them. If "learning by making" is a good idea, then shouldn't teachers learn new ideas about Etoys by making them (but with lots of guidance)?

On the other hand, there are any number of things in the Etoys design itself that could be vastly improved to help both adult and child learning, and also for them to make better extensions. For example, I do all my talk presentations using Etoys and I write scripts to sequence "builds" of additive visual material to the slide (Powerpoint has a feature for doing this that is more convenient for some goals and simply won't allow others). It is very instructive to do this by hand a few times, and then gets annoying. Etoys does not have a good extension mechanism for "packaging up" a solution to "slide builds" that can then be used as a feature. This is a real sin against our own precepts. The lack of it is due to EToys being thought of as temporary and of limited scope at Disney. It is terrible that we don't have it now. Why don't we do it now? Because we've been trying to move on to the next design since before OLPC came along. And so forth.

Let me know your views.

***Thu Aug 12, 2010**: (SQ-749) and Kathleen's question on "What do you mean by Artifacts?"*

One of the "Hypercard killers" was a knockoff called HyperStudio, which replaced scripting by features.

This was very popular in schools because most people (Americans especially) are driven by pragmatism rather than interest in understanding. They don't have the desire or will to try to understand if something already does a function.

Needless to say HyperStudio created a disaster of learning even as it seemed to be a good idea to many

---

***Mon Aug 23, 2010**: (SQ-749) and Kathleen's question on "What do you mean by Artifacts?"*

Hi Steve,

Thanks for the thoughtful note!

By "when easy and when hard?" I meant parts of the pedagogical and learning processes that have been tested out ahead of time. For example, Ted Williams (one of the greatest hitters of the last 70 years) determined through experiment that using a batting tee (which he invented) would really help one very important aspect of learning to hit. On the other hand, it has also been determined by experiment that putting frets on a violin doesn't help, and is even retrograde. There are many such questions in mathematics, science, and especially computing. In writing, it is not thought to be a good thing to use a "library of already written sentences and paragraphs" to aid composing a piece. But it is done all the time in computing (and sometimes this is helpful for some of the reasons you list, and sometimes it is a very bad idea because it makes it easy to ignore building some of the important fluencies).

I still don't recognize the quote .... but if you send the Elluminate video to me, we can certainly do a transcript (there is a very speedy and reasonably inexpensive transcription company here in LA).

Criticizing teachers and adults accurately is not bashing. And the real problems have to be faced. I was very lucky to have learned to read fluently and rapidly before I entered school, and this gave me a way -- via home and town libraries -- to bypass "the barriers of adults in learning" pretty effectively except for too many unpleasant personal interactions. That being said, I also had several exemplary teachers who made a big difference in my life (and you can't substitute books for these). But the ratios of bad vs good were horrendous, and books can be read and discarded rapidly to find the good ones. The wide range of personality and motivational types we find in children suggests that "better adults" are really critical for many of them -- not all of them will be as driven as I was to get around the barriers.

The "older kids" path is a good one, and can help a lot, and much more should be done with it.

The other thing to ponder is that it is philosophically possible to make a technology that lies somewhere between what books already do -- bring us important parts of genius teachers and minds in a form that can be replicated and distributed by the millions -- and what great human teachers can do. Exactly how far this can go and what kinds of people it can work with is yet to be found out. For example, could we imagine "a book that can help kids learn how to read it"? What if we took (say) Dr Seuss books and "did something" that could help a child to read them in a variety of ways? (And by the way, here is a perfect example of "when easy, when hard?", because an overshoot of giving too much help could remove the motivation of the children to learn how to read themselves.)

To pick neutral ground, a lot of really valuable help in learning music could be given by "superbooks" -- and this is partly (a) because there is a lot known about how to teach various aspects of music (especially learning how to play an instrument), and (b) because this is one area in which "good listening" can already be done by computer programs. ("Good watching" is not so easy to do at this point.) And (c) because the computer doesn't have to do all the listening -- it can help the learner listen as well in a variety of ways. This is a key idea in several language learning environments available now.

Some subjects will be quite challenging to do "good watching" in for some time to come. But quite a bit of mathematics and especially that which overlaps with special kinds of programming have the possibility of doing good watching, so we can expect to start seeing much better environments for children over the next 10 years or so.

Re Maria Montessori -- she struggled to design her environment as well, but she stuck with it (and was a special kind of genius), so she got many good results. She is definitely a patron saint of ours ....

Re Galilean gravity ... For now, I suggest the much easier route of making "jpeg movies" (they are just a folder of images whose names sort in order). The Etoys movie player has an option for playing such jpeg files.

One of the projects for etoys that is illuminating is to make a simple movie player using animated images. (This is a perfect example of Etoys needing to have a way to package up objects (it actually does, but it is not very convenient -- the "world menu" has an option for saving morphs, and all the connections between them will be preserved if the objects are put in a Playfield first.)

That said, I completely agree that "objects should be easy to package, reuse and to share".

I think I'm not understanding your graphing example. It is "pretty easy" to position text objects containing numbers (you can adjust the centers of any object and the x and y coordinates can have some arithmetic done to them with results put in the numeric value). Please give me more info here.

I don't think "just everything" should have to be made from scratch in order to learn -- but I also think that one of the reasons that so many adults don't get fluent at this stuff is that they don't make enough things from scratch. (Analogies to drawing and painting, sports, music, etc. should be drawn here.) And, sure, they are already busy and there isn't a lot of spare time -- so the bootstrapping process is difficult when the adults start late, after they are already busy. There are chicken and egg problems also. The more fluency that is attained also changes perceptions and realities about easy and hard.

To go to science, most scientists learn most of their knowledge from reading, but it is the "real science" they do when they are starting out that changes what they are doing later from "believing in a new catechism" to "actually thinking as a scientist about scientific relationships". Similarly, there is a certain amount of "To know the world, one must construct it" (Pavese) that is absolutely critical in learning mathematics, where part of the main point is to chain together inferences and to understand how claims are preserved via that chaining.

Re: the next Etoy-like environment. The current plan is to work on a "science and systems" curriculum for ca 8th grade and to make a computer environment that can really make a difference for this curriculum. This will not be an extension of Etoys, but will be more comprehensive in many areas, and will have new abilities in others. It should wind up being better than Etoys for the 5th graders that Etoys was originally aimed at, and it should fix many of the problems that we all agree are there in the current version.

Please do follow through with your plan of asking for 3 (or 20) things -- thoughtful comments like yours are like gold for us.

Mon Aug 23, 2010*: (SQ-749) and Kathleen's question on "What do you mean by Artifacts?"*

Hi Stephen

It seems to work (see attached picture).

Remember that "text is text", if you want to use it as a label then you can center it by using the FF menu.

Mon Aug 23, 2010*: (SQ-749) and Kathleen's question on "What do you mean by Artifacts?"*

Sure ... one way to think of this is that you can start with a sample text object that you modify to make a "prototype" of the ones you want to use as numeric labels. Make them wide enough, center the text, make the background the same as the background of the screen (so it will make a nice space when it is positioned over a line, etc.)

You can put this in a variable and then make copies (instances) of it to actually use as labels.

Thu Oct 28, 2010*: (SQ-749) and Kathleen's question on "What do you mean by Artifacts?"*

Thanks Steve!

These are very helpful (and many are on our list already). Please don't hesitate to suggest more as we go forward.

Thu Oct 28, 2010*: (SQ-749) and Kathleen's question on "What do you mean by Artifacts?"*

Sounds good to me!

Sun Jan 30, 2011*: Plan Ceibal y/and General Electric*

GE is being congratulated for recognizing that the iPhone and iPad are pretty good ideas and technological realizations. But isn't this like the congratulations Bill Gates got for finally recognizing the Internet (about 25 years after it had started working)?

Seems as though Apple had a lot more on the ball than Bill Gates or GE here (they used to do computing in the 60s, but couldn't see what it was).

And most of the ideas at Apple (and for personal computing and the Internet) came from research funding that no company or government has been willing to do since 1982.

Sat Feb 5, 2011*: Plan Ceibal y/and General Electric*

Hi Chunka,

I've been challenged on this point more than once, and have challenged back to come up with one invention that was done after 1980 that matches up to the top 10 done before 1980.

This has not happened. I've been able to show the prior art for all suggestions.

Essentially everything in the last 30 years has been commercializations and other forms of "innovation" based on what was funded by ARPA, ONR, and by extension, Xerox in the 50s, 60s, and 70s.

The important point here is that there are many new inventions needed, and they can be identified, but no one has been willing to fund them. It's not that the early birds got the worms, but that most of the needed worms out there are being missed.

Sat Feb 5, 2011*: Plan Ceibal y/and General Electric*

Hi Chunka

It's basically "hunting and gathering" vs. "agriculture". Or "parisitism" vs "symbiosis". These are built into human nervous systems by genetics, but it is still surprising given that we've had agriculture for more than 10,000 years, and one would think it would be more generally noticed and understood.

Here is an example from today that is like the impulse and vision that propelled the 12 year effort that invented personal computing and the Internet.

The idea reaches back to the 60s and 70s, but an above threshold invention was not accomplished.

Children need to be helped to learn important things, such as reading and writing, mathematics and science and engineering. The helpers need to understand the subject matter, and also how to help the learning process with individual learners. Studies have shown that for many learners, just lowering the learner-to-helper ratio makes an enormous difference.

For the US, it has been calculated that it is not possible to create enough knowledgeable K-8 teachers for math and science over the next 25 years, even for the 30:1 student teacher ratios we have today. It has been estimated that this problem is much worse in the developing world.

Vision: It is a destiny for interactive computers to become sensitive expert learning helpers for many important parts of human knowledge which children need to learn.

This is an extension of what the printing press has meant for learning. There aren't enough Socrates' and other great teachers to go around, but important parts of their magic can be captured in print, replicated and distributed by the millions. This allowed more ordinary teachers plus great-books to do some of what great teachers can do. And this changed the world.

Computers can represent books and all other media, and they should be able to actively help us learn to read them (even if we start off not being able to read at all). And we should be able to go much farther beyond the book, to make computer helpers that can also understand and answer many questions in ways that extend our learning rather than undermines the growth of our minds.

These computer helpers also help the human helpers. It's not about replacing humans (even if they don't exist) with computers, but making a more powerful learning environment using technology to help.

This is a hard vision to pull off, just as personal computing was. The funding needed to be long term in the 60s because much had to be done to (a) even find a version of the vision that could serve as "problem and goals", and very importantly (b) to "grow" the grad students and PhDs, who as second and third generation researchers, were able to frame the problem and do the inventions.

The payoff has been enormous. The inventions at PARC alone have generated about $30 Trillion dollars of wealth worldwide (and yes Xerox's return on their investment in PARC has been more than a factor of 200 (from the laser printer alone).

The great funding in the 60s was done mostly by the government, and for personal computing and pervasive networks was spread over more than 15 universities and research companies who formed a cooperative research community. (The story of this is told in "The Dream Machine" by Mitchel Waldrop).

The funders today do not have a lot of vision, and they have even less courage. A new kind of user interface that can help people learn is not just for the very important needs of education around the world, but will also open up learning in business, defense, and for consumer design and products.

How much would this cost? A critical mass of institutions and researchers could be supported starting at about $100M/year. By contrast, the estimated US spending for Iraq and Afghanistan for 2011 is about $170B. So we are talking initially about less than 1/10 of 1 percent of the cost of these wars.

What's the hitch. First there is risk. It is a very difficult problem. But I think a bigger hitch is that it is likely to take more than 10 years to pull off. This is longer than any corporate or government cycle.

Perhaps a larger hitch lies in one of the biggest changes in funding today as compared to the 60s. There is no question that a funder of large research monies for high risk projects is "responsible". Today's funders are so worried about this responsibility that they confuse it with "control" and tried to insert themselves in the decision processes. This is a disaster (they are funders not researchers, and the more visionary and difficult the projects, the less their opinion can be at all germane.)

The 60s funders made no such error. They said "we can't evaluate projects behind the Beltway, so we'll fund people not projects". This required trust in both directions, but it is a proper and good allocation of expertise.

The other thing that the 60s funders pointed out when queried by worried politicians, is that they were "playing baseball" not "going to school", meaning that given the high risk and high payoff of the research, they only needed to bat .350 and "the world will be changed"). Today's funders want certainty, and this is engineering at best, and this does not change the world because the hard important problems never get worked on.

Best wishes

Alan

Sun Feb 6, 2011*: Etoys Particles/Partículas Challenge/Desafío*

The dynamics of this is pretty well modeled by using the "forest fire" particle simulation. Basically about 80% of the human population needs to feel that others around have already come to a conclusion before they will conclude. This inserts huge delays on ideas (sometimes 30 to over 100 years).

The forest fire particle simulation (originally done in Starlogo) allows you to choose the ratio of forest to clearings and start a fire to see if it propagates. One of these models requires at least 66% connectivity before the fire will spread.

***Sun Feb 6, 2011**: Plan Ceibal y/and General Electric*

I agree.

But now we have to figure out how "to help the helpers", because besides wisdom, many need much more knowledge than they have

---

***Sun Feb 6, 2011**: Plan Ceibal y/and General Electric*

I don't think of "teachers" or "teaching" as dirty words. And I don't separate them by age group, profession, or whether parents or not. (Do I have to say that good teachers facilitate learning ....?)

There are lots of poor teachers in the world (for many different reasons), but it's important to understand that no child ever invented Calculus, nor did any adult until very recently in our 200,000 years on the planet. Good teachers are vital, and most especially for the powerful invented ideas and knowledge that is less strongly built into our genetically and culturally fashioned brain/minds.

So we need good teachers from our peers, our parents, our schooling systems, our vocations, our delights, etc.

***Sun Feb 6, 2011**: Plan Ceibal y/and General Electric*

Hi Ron,

I've already played this game ad nauseum with many groups on the web. So I urge everyone to rise above the temptation to name your favorite idea that seems "new".

But do you really think there were no peer to peer and cloud computing systems already deployed before 1980? (Hint: I used both quite a lot back then, and for a short time actually was in charge of an ARPA task group to define an AI "cloud resource" for the already running ARPAnet -- the one that got built was a multiple processor system (C.mmp) designed by Gordon Bell)

For much larger issues and inventions than DHT, let me refer you to the 1978 PhD thesis of David Reed (popularly known as "the '/' in TCP/IP") at MIT.1978. If you haven't heard of David or read this thesis, then this helps make my main point.

Since it would be really improbable for me to be aware of all developments after 1980 (and even some before), I don't claim there have been none.

I simply asked for 3 (or even one) since 1980 comparable to personal computing, GUIs, the Internet, Engelbart's notion of "online system", etc. Previous essays into this yielded many suggestions, but I was able to identify prior art for all such.

For example, Tim Berners-Lee was suitably embarrassed when he found out about Engelbart - first for the web not doing as well in the design, goals and execution, and secondly, because as a physicist he would have been drummed out of Physics if he had not tried to "stand on the shoulders of giants" (as Newton said), and he had assumed falsely (and I think partly because our field is so careless about its historical great steps up) that computing had no Netwons, and the Internet had somehow just appeared without thought out purposes, and he failed to look for them.

***Mon Feb 7, 2011**: Plan Ceibal y/and General Electric*

Hi Ron,

Well, "the field" itself really doesn't have a good overview of the strongest ideas it has had since 1950. So I certainly don't get upset when anyone randomly isn't aware of something that happened 40 years ago.

But researchers and engineers need to be a lot more careful about checking out prior art. The lack of this has led to the odd phenomena since the 1980s of "reinventing the flat tire". Some of these that were really done badly (like the web browser, various bad scripting languages and UIs) have held things back for decades (and still are).

I predict that you will be amazed by Dave Reed's thesis. We implemented it a few years ago and it is now both an open source foundation (Croquet) and a startup (Teleplace).

***Tue Apr 5, 2011**: First day of Etoys class in Somerville*

Hi Derek

As noted below Squeak runs pretty much everywhere all by itself, so it somewhat defeats the purpose to tie its execution to something that might not be everywhere.

There are several ways you can do this without bringing something optional like Java into the critical path.

***Fri Sep 2, 2011**: Why is Scratch more popular than Etoys?*

Both Etoys and Scratch were done by some of the same people (especially John Maloney), and both are on top of Squeak Smalltalk. The original Etoys interface was more like Scratch's (small area for action results, most of the screen area used for showing tools, tiles, etc.). The first Etoys was aimed at the web (at Disney), and making the start up more obvious and using more screen for it is a good idea I think. The projects for the first Etoys were also like Scratch projects: effects, jokes, postcards, simple animations, etc.

The next version of Etoys was for classrooms that would have much more help and do more ambitious projects. So we went to a full screen with flaps for the tools. This worked well in this setting.

The OLPC XO presented a problem in that it had lots of pixels but a very small visual angle. We decided to stay with the classroom version, and I think this was a good idea on the one hand, but it went against the general lack of help that might be available in many of the XO's destinations.

Then we handed Etoys over to the Squeak Foundation, and the version they put out online retains the classroom UI with flaps.

Personally, I think the Scratch UI is better for many things than the Etoys UI, especially first encounters, which are so important for so many beginners these days. And I think the Scratch people have done a fantastic job on their web presence, including their gallery, the emulator for Scratch projects so you can see what they do, their online materials, etc.

On the other hand, Scratch lacks a real media system, a massively parallel particle system, and many other features that are really needed and useful for learning things beyond simple programming. Etoys is much more complete in many more ways.

Both systems have strong and weak points as to their language choices. Both lack nice extensions into more sophisticated programming. Both need to be greatly improved.

And so forth.

But I think in the world we live in, it is initial experiences that count in a non-classical culture (and this is most cultures around the world including the US). So we have to praise Scratch here, and wish that it had more depth. Etoys could easily be set up with a more useful exposed UI, and this would help tremendously in initial impressions.

As to how many features to include, this is a tricky one. Scratch has quite a few features -- such as the thought balloon one -- because it was primarily initially designed for the "Computer Clubhouses", afternoon drop in experiences for junior high and high school kids.

Etoys has fewer built in features because part of the "real deal" is to learn how to make your own features. It could have clip art, but we left it out because it is cognitively a good thing for children to learn how to draw. This is good for a "learning tool", but is not good for a "productivity tool".

There is no question that both systems could be improved along the lines of their current styles.

One could also imagine taking the lessons learned from both systems and inventing a new environment that is quite a bit better than either. I like this option the best.

Sat Sep 3, 2011*: Why is Scratch more popular than Etoys?*

Thanks Ron!

Well, there are no barriers besides some work and a few more features to make up the differences in either direction. So that is probably what should be done.

Sat Sep 3, 2011*: Why is Scratch more popular than Etoys?*

One thing that hasn't been mentioned (but I should have) is not just the initial experiences and learning curve for children/students, but also for the adults who are trying to help them. I think this is where the relative opacity of Etoys really hurts its acceptance, and why the intro UI should be set up differently. Adults can so easily bypass things that are good for children if they find them difficult to learn (consider what has happened to real math and real science).

When Mitchel, Brian and John Maloney were thinking of doing Scratch, I urged them to try a new design center that built on the Etoys experience in the hope that we could test more ideas. I think they succeeded brilliantly with both children and adults (I was only disappointed that so much of the good Etoys depth was excluded in the process).

The good news is that enough was retained to still bring real content (as opposed to e.g. the iPad, which discarded too much of what a computer is in order to be readily learnable and popular in the pop culture).

But (to me), once Hypercard appeared in the late 80s, it showed how to do "media programming for beginners" and (to me) drew a line that we should not retreat backwards from. The irony is that the media objects and tools for doing a Hypercard like experience as part of the environment are lurking below the surface in Squeak Smalltalk. Etoys exposes them wrapped in tile programming, and Scratch does not. This is a big mistake for Scratch IMO. Hardly anyone complains because hardly anyone understands what is being lost.

Given the problems with plugins, downloads, etc., one could imagine the next versions of Etoys and Scratch being done in Javascript (or less usefully in Flash). Here the temptations will be great to exclude needed features that are not already programmed in the substrate system. And we could see a further watering down of the ideas (for example it is not easy -- not possible pragmatically -- to do a particle system in Javascript). There will be many rationalizations concocted to explain away the lost abilities (just as there have been for what is still not doable in browsers after 20 years that is readily doable on the computers that run the browsers!) -- but the end result will be less for the learners, and that would be a real shame if allowed to happen.

We don't want to wind up with "Guitar Hero" here. We are trying to get children to learn powerful ideas, not just to "have a fantasy experience".

Thu Sep 29, 2011*: KIds from around the world measuring the Circumference of the Earth*

But consider a flat Earth and a low small sun directly over the well. This will yield exactly Eratosthenes' result. The key here, which I've never seen mentioned in any books for children, is that the Greeks had to have a very good set of reasons for thinking the Earth round and the sun large enough and far enough away (and they did).

I gave a talk on how they did this in the Kyoto Prize lecture followups in San Diego in 2005. Aristarchus was one of several key figures.

The shame of it is that for both math and science learning, the important heuristic of trying to identify all the possible cases for a result is never encountered by the children (or most adults) who have read about Eratosthenes.

***Thu Sep 29, 2011**: KIds from around the world measuring the Circumference of the Earth*

Hi Jason

I finally chased it down via "Alan Kay Kyoto Symposium"

https://www.youtube.com/watch?v=7heB00K2ueY

This is likely to be very frustrating because they gave me a bad wearable microphone -- it is barely understandable when I'm at the podium, and not at all when I'm moving around away from the podium mike.

However, I might be able to find the material (it was done entirely using Etoys as both the presentation and demo media).

The talk was sneakily about thinking ... via how the Greeks were able to transcend our messed up genetic brains and minds. To me, how they were able to get the first really accurate picture of our situation in the universe, not just of a round Earth of a certain size, but of the Earth's relation to the Moon and the Sun -- quite bypassing normal commonsense and cultural reasoning -- is one of the most thrilling episodes in our intellectual history. And, it was just there for an instant, roughly during the Alexandrian Greeks period.

***Sat Jul 7, 2012**: Has anyone build a set of gears in Etoys or any other freely available program?*

Hi Steve,

Yes, we did experiment with gears in Etoys many years ago, and I think we tried one or two experiments with gears at the Open School.

This is a case where lots of the goodness of the gears idea lies in the physical world, and just giving kids a simulation of gears lacks "juice".

The gear models we did actually used collision detection to drive one gear by another, and this was a good set of things for the kids to think about. This would work better today (with more computing cycles available, etc.). One of the important things we never got around to in Etoys was to make an industrial strength collision detector (like the best ones used in video games) for both the macro graphic objects and also for the particle system. Having one of these as a basic facility would make a big difference in what could be thought about and attempted.

Another fun thing at this level is to make "ratchets" and then "Feynman" engines where particle energy exchanges from random collisions will nonetheless drive the "engines" in the direction they can go.

***Sat Jul 7, 2012**: P.S. Re: Has anyone build a set of gears in Etoys or any other freely available program?*

P.S.

I vaguely recall that someone did an Etoy model of the Antikythera ... (might be findable on the web somewhere)

Going a little deeper, it would be good for children to think about: Computing is inventing a kind of machine that can carry and manipulate representations of ideas.The machine can be made from physical or symbolic materials. For example, the abacus -- which vastly predates the Antikythera -- is also a computing machine. And so is Algebra ...

***Sat Jul 7, 2012**: P.S. Re: Has anyone build a set of gears in Etoys or any other freely available program?*

"Inventing" because what we do when we write a program is precisely to invent and make a machine to accomplish the goals of the program.

It would be good to get them to think about the larger notions of "machine" and "mechanism", including ideas about how the natural world seems to operate ... not just physics and chemistry, but also biology ...