By Craig Hickman

Ben: Dad, what have you been doing upstairs?

Dad: Writing a short history of Kid Pix.

Ben: Do you really think anyone is going to want to read that?

Maynard and Joe

Some people believe that humans are either analytical, rational and sensible or spontaneous, creative and intuitive. Having watched **too much TV** as a kid in the 50's, I thought people were supposed to be Sergeant Joe Friday (from Dragnet) or Maynard G. Krebs (Dobie Gillis). My problem was I always felt a little like both of these guys.

Sargent Joe Friday

Maynard G. Krebs

In college I decided to be a fine art photographer. But in 1972, when I was working in photography at The Evergreen State College, I heard a clattering noise coming from what was labeled the "Terminal Room." Inside, a friend was sitting at what looked like a large typewriter. He said he was **programming** and asked if I would like to see how it was done. I said "yes" and have **never been the same since**.

Me next to the computer terminal shortly after I was introduced to programming, 1972.

I bought a book explaining each command and how it was used. I started writing little programs that would use each of the commands. None of my programs tried to do anything useful, but they tended to be fun and surprising.

In 1974 I left Evergreen along with access to a computer and moved to Portland where some good friends and I started the non-profit photography **Blue Sky Gallery**. Later I went to graduate school in photography at the University of Washington, where Paul Berger, a professor of mine, had just gotten an Apple II computer. Microcomputers were new then and he explained what it could do. **Suddenly my craving to program came back**. I had to have a computer of my own.

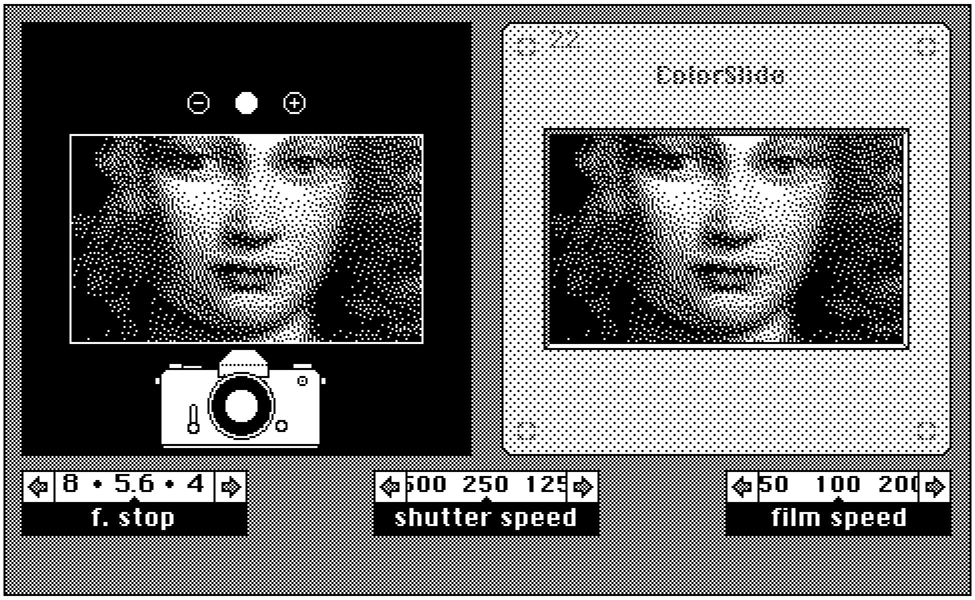
My Atari in the darkroom

A couple of months later I was busy **programming** at home with my own computer, **an Atari 400**. This was a **joyous experience**. (My wife says that the day it arrived was the happiest she's ever seen me.) Despite its membrane keyboard and TV set monitor, it was loaded with graphics, sound capability, and programming challenges. I came up with endless projects. I can remember the thrill when I discovered how to input and output signals from the joystick ports. At one point I designed a **home surveillance system** for our apartment in Seattle that used little magnetic switches from Radio Shack. It displayed a representation of our apartment on the screen and showed when a door or window was open. It worked so well I expanded the system to include little tilt switches placed on bushes outside the windows. This also worked fine until one windy night when I was away from home and it set off the alarm every few minutes. The next day my wife told me to dismantle it.

I began to be very interested in **user interface and program design**. Articles that influenced me the most were ones by Chris Crawford in *Byte* Magazine and by Alan Kay in *Scientific American*. I wanted to discover the simplest and most effective way to perform any task on the computer. This is when I began **my quest for software design beauty**.

My interest in photography began to wane at this time. The computer was just more interesting. I got one of those "Electronics Made Simple" books, figured out how to use the computer to control the enlarger, and made a rather nice program to control most of my darkroom timing and record-keeping tasks. I ended up using a "card" metaphor for this project that was sort of HyperCard-like even though it predated HyperCard by several years.

In 1984 Apple released the Macintosh. Up to this point I saw no reason to stop using my Atari (except for the fact that hardly anyone else used them and there was practically no audience for the software I was writing). For its day, it had great graphics and sound, and I knew a lot about programming it. Other computers had more serious reputations but they didn't seem as much fun to program.**The Macintosh was different**. It was totally graphical so the visual possibilities were much greater. Most of all, the Mac seemed to have people with a **consistent and enlightened vision behind it**. The user interface was intelligent and beautiful. I got a Mac and began teaching myself to program on it. Originally I programmed in Rascal, a language developed at Reed College which was sort of a cross between C and Pascal. Later I switched to Pascal.

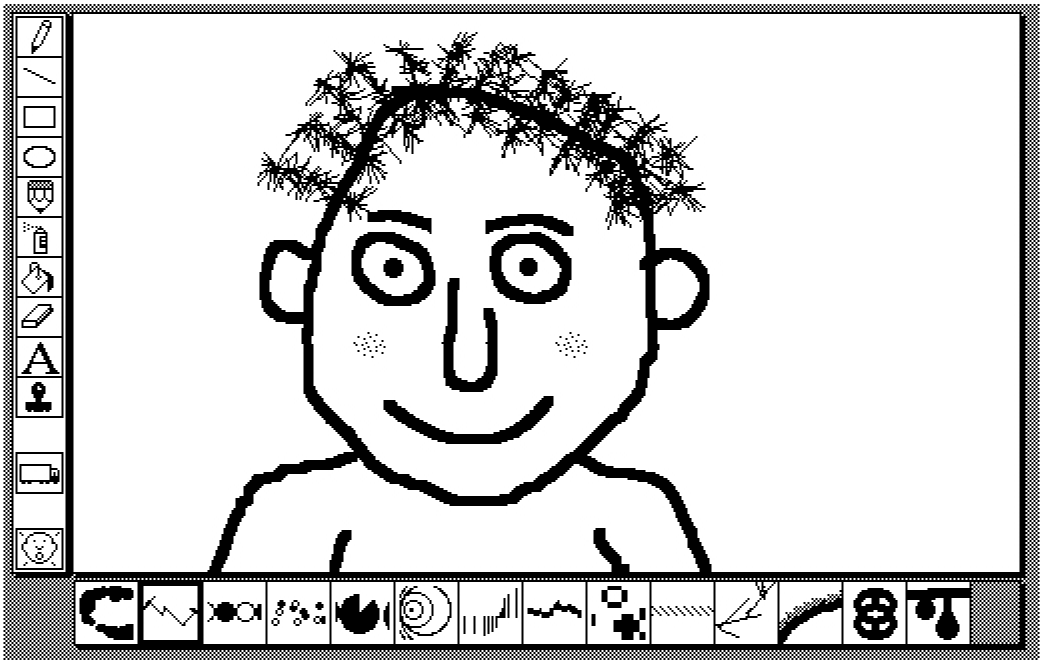


Screen snapshot of Camera, 1988

My first Mac programs, which were given away, were designed to help me learn the Mac QuickDraw routines. They did nothing useful but were mildly entertaining. **Company** was a desk accessory that displayed an animated face and a game to keep you company. **Fido**was an animated dog that talked in graphics. I also created two artist's books, ***Signal to Noise*** **and** ***Dry Reading (Available on this site)*** , that shared some of the graphics and aesthetics my software has. My most successful program from this time was**Camera**, an educational title that interactively demonstrated the relationship between f. stop and shutter speed. I put **Camera** in public domain and it was distributed widely on computer bulletin boards.

One day in 1988 while I was using **MacPaint**, the wonderful paint program that came with the Macintosh, my 3-year-old son Ben asked to try using the program. I was surprised at how quickly he got the knack of using the mouse and how easily he was able to select tools. The problem was that he didn't have total control of the mouse and would occasionally (like every five minutes or so) pull down a menu and bring up a dialog box that he couldn't dismiss without being able to read. Everything was fine as long as I was in the room, but if I stepped out for a few minutes I would come back and find Ben kicking on the floor in **frustration**. This was **not what I had in mind** for his introduction to the computer. As it turned out I was looking for a good programming project. I decided to write a simple paint program for Ben to use. I also decided that I would probably give it away for free. A few years earlier I had written what I thought was a rather nice darkroom timer program for the Atari computer and signed an agreement with a company to sell it. But few people used the Atari computer and almost no copies were sold. I always wished I had just **given the program** away so people could use it and I could have some satisfaction for the great deal of time I had spent writing it.

Since I was going to give the program away I decided **I could design it exactly the way I wanted**. These were its guiding principles:



Screen snapshot of the original Kid Pix

1. **(The Prime Directive)** The program should be **extremely easy to use**. No manual should be needed and program features should "explain themselves" through use. All tasks should be able to be performed in the simplest, most straightforward way. The program should go out of its way to meet the user.

2. As long as The Prime Directive is not violated, every opportunity should be taken to make the program **surprising and satisfying to use**. No opportunity should be missed. The process of making a picture should be as important as the picture produced.

3. The program should, in some way, **expand** the concept of what computer paint programs are, as well as what mark making can be.

4. The program should be **open ended**. Premade graphics should be modifiable by the user.

5. I wanted each tool to have a **large number of options** but I didn't want the user to be overwhelmed with options. I didn't want to use pull-down menus for this because, among other reasons, Ben had trouble with them. The system I developed involved displaying a selection of 14

options for each tool in a window at the bottom of the screen. **As the tool selection changed, the options changed**.

6. The first Kid Pix was designed to run on the original Macintoshwith a 9-inch, black and white screen (512 X 342 pixels). All paint programs made for the Mac at that time displayed a virtual page showing only a portion of the potential drawing area at one time. I decided this was a bad idea for my program because it is essential to see your whole picture at the same time. So I made the drawing area on the screen as large as possible but did not have the contents of the screen scroll. **"What you saw is what you got."**

7. In addition to "traditional" computer paint program tools like the brush, pencil, eraser, circle and rectangle, my program would include**tools that would be surprising and visually unusual**. Most computer paint programs were designed in a way that made them seem like second class citizens, always attempting to mimic traditional art making tools, but never doing a particularly good job of it. In fact, the computer can make marks in a variety of interesting ways that traditional tools can't. It can also do a good job of creating comic tools. This is where the "wacky brushes" came from.

The Undo Guy

8. Undo is an extremely important function that deserves a button of its own instead of just being a menu item. **The Undo Guy was born.**

Some of the stamps from the original Kid Pix.
(Notice a relative of the Undo Guy here?)

9. Children love **stickers and rubber stamps**. The program should allow for some pre-made iconic art. The stamps in the Original Kid Pix were taken from the Cairo font from Apple Computer designed bySusan Kare. Apple had stopped distributing Cairo with their computers and I thought the pictures were too good not to be seen.

10. Users are very smart and can tell quickly when someone is trying to cover up a lame program with flashy graphics. A poor program gets dull very soon no matter how trendy the interface is.

11. When Ben built something out of blocks, he enjoyed knocking his structure down almost as much as he did building it. **Getting rid of the picture should be fun.** Hence the exploding firecracker

eraser.

12. Why not have an eraser that erases your picture but **displays a hidden picture** as you erase?

13. When small children are using the program, a parent should be able to **hide the menu bar** so desk accessories, etc. can't be selected accidentally.

14. The program should, in most ways, **act like a standard program** and have a standard user interface so users new to the computer learn basic principles. Kid Pix ended up being widely used to introduce children and adults to using the computer.

15. While on the surface the program should seem to be for children, open minded **adults should get a kick out of using it too**.

As the program came closer to completion I needed to think of a name. After considerable thought I came up with **"Kid Pix."** It was**easy to say**, had a **nice punch to it** and was made up of simple three letter words. (This must be something I am attracted to: my wife's name is Kay and my children are Ben and Art). I liked the way "Pix" referred to "**Pictures**," "**Pixels**," and "**to Pick**," and I liked way the words looked graphically when stacked on top of each other. It also occurred to me that it would probably be easy to say in other languages, though at that time I never imagined that it would be sold around the world. Later, Broderbund decided to rename it and had a company-wide competition to come up with a new name. When none came up, they went to a professional name consultant. He told them that he could take their money and come up with a new name, but it wouldn't be any better than "Kid Pix."

Program and document icons for the original Kid Pix

I could have run the Kid Pix name together to become KidPix, as was the style then for program names. I decided against it because I was afraid it was just a naming fad, although today if you do a search on the web the name appears as much one way as it does the other.

As I added new features to Kid Pix, I would try them out on Ben. Anything he had trouble with would be changed. **The customer was always right.** One day I took a number of computers to Ben's pre-school and tried it out on the children there, most of whom had not used computers before. Within ten minutes of having the children use the program I had accumulated a **long list of small user interface changes** for the program.

Before I'd finished Kid Pix, Apple came out with the color Macintosh. It seemed that the next logical step would be a color version of Kid Pix. My friend Ken O'Connell always supported my work and insisted that **I should sell the program rather than give it away**. I felt that as it was, Kid Pix made a **great freeware** program but didn't yet have enough capability to be a commercial program. Before sending out the finished black and white version of Kid Pix I decided to commit to writing an expanded, **color version** of the program, which I would sell for $25. In **November 1989 I started distributing the black and white version**, which included an ad for the color version stating that it would be done by June 1990. (The original black and white version of Kid Pix is still available for**downloading** today from shareware web sites like shareware.com.) Writing the black and white version was quite a challenge for me and I had no idea how to program color. I anticipated several months just to convert what I had done into color. As it turned out I figured out the basics of the conversion in about a day, which left lots of time to work on new features.

Manual cover for Kid Pix Professional

It was the style at that time to add the word "Professional" to new versions of software. To be humorous, I called the new version "**Kid Pix Professional.**" Besides adding color, I was **bursting with new ideas for the program**. Here are a few of the changes and additions I made:

1. The program adjusts to **compensate for monitor size and displayable colors**. You can make really big Kid Pix pictures if you have a large monitor, and all aspects of the program are customized to work in either black and white or color depending on the computer used.
2. **Sounds added** to the user interface make a more satisfying experience. The Alphabet tool now

reads the letters.

3. There are twice as many **wacky brushes**.

4. A new category of tool called "mixer" effects makes **changes that affect the entire picture.**

5. New, more interesting **hidden pictures** are added.

6. More **rubber stamps** are added.

7. Since my wife speaks Spanish, Kid Pix's spoken Alphabet and menus become **bilingual**.

8. There is now a **manual**.

By June of 1990 I was starting to get $25 checks in the mail for Kid Pix Professional, and the program was actually done on time. It was a real **treat to walk out to the mailbox** every day and find checks and letters. I ended up **selling about 100 copies of Kid Pix Professional** and think that if vintage software is ever collected, it should make one of the really rare finds.

Ben and me just after Kid Pix was released

Many people had been **supportive of Kid Pix** when it was being developed, among them Paul Scott, Bonnie Meltzer , Anita Best ,Richard Wanderman and Judi Mathis. Many were instrumental in helping make contacts that finally resulted in my sending a copy of Kid Pix to Broderbund Software in the summer of 1990. I really didn't expect that the program would be picked up by a major publisher, but when I received the name of someone at Broderbund to send it to, I dropped a copy in the mail anyway. About a week later I got a call from Harry Wilker of Broderbund, who said right off that he **wanted to publish Kid Pix**. He also said that they receive a lot of paint program submissions and almost didn't even look at Kid Pix. A week later I flew to down to the corporate headquarters. There I also met Leslie Wilson, Richard Whittaker and Rebecca Knievel.**Rebecca had been a student of mine** at the University of Oregon a year earlier and the fact that she was working on the Kid Pix project was a total coincidence. Richard, the publisher, was very astute in his assessment of software. I ended up working extensively with Leslie over the next few years and found her to be a first-class software designer and extremely good at interpersonal relations.

Before Kid Pix was released, **Broderbund wanted to make a number of changes**. Fortunately, I agreed whole-heartedly with them and felt that the people I was working with **really understood the program.** One of the major changes was about sound. Kid Pix Professional had incorporated some user interface sounds, including the exploding eraser and alphabet sounds. I hadn't given everything a sound because, for one thing, I couldn't get some of the more sophisticated sound techniques to work, and for another, I quickly discovered that I didn't have the aptitude for sounds that I had for dealing with the visual aspects of the program. Fortunately Broderbund had an **exceptional sound department** which created a wonderful set of sound effects for Kid Pix tools. (I also figured out how to program sound without crashing the computer.) There were others from Broderbund and elsewhere who contributed to the final program. At the bottom of this document you will find the credits as they appeared in Kid Pix 1.0.

Most **stressful** was dealing with **bugs**. Broderbund had a large department of skilled people whose job it was to find things wrong with software. And they did. I had thought Kid Pix Professional was quite bug free: after all, I had been selling it. My **first bug report was 30 pages long!** And each new feature that was added brought new bugs, which had to be fixed. Even the bug fixes often created new bugs. In the past I'd only had myself to please with my programming; now I was responsible for fixing all this myself. In the end each one got stomped out. This was also a stressful time because I was teaching full-time at the University of Oregon in a department that required many hours of teaching and committee work. Throughout this time I had many **long telephone calls with Leslie** discussing a wide variety of issues related to Kid Pix.

Phillip Ivanier and John Sculley demonstrating Kid Pix at MacWorld, 1991

Even before the Broderbund version of Kid Pix was released, **the feedback I was getting was good**. In December Stewart Alsop in*P.C. Letter* had some kind things to say about a pre-release version of Kid Pix, and to my great surprise and shock, in January, 1991**John Sculley featured Kid Pix in his keynote speech at MacWorld.** In the following issue of *Macweek* a headline read, "Kid Pix Vrooms past Sculley at Expo." This was all before Kid Pix was released. At that time Broderbund started work on Kid Pix for DOS.

Kid Pix 1.0

In March 1991 **Kid Pix 1.0 was released,** and things started to get really good. Sales were significantly above expectations and the**reviews were exceptional**. Very positive reviews appeared in*MacUser*, *The San Jose Mercury News*, *The Wall Street Journal*, *The New York Times* and *The Whole Earth Review*.
A *MacWorld Magazine*review stated, "**Pros: Brilliant; hilarious; innovative; inexpensive. Cons: None."** An article in the *Wall Street Journal*referred to it as "a startling product for the Macintosh that uses color and sound." I couldn't believe what was happening! Kid Pix also won quite a number of awards, including the 1991 *MacUser* Eddi for best Children's Program, the 1991 Software Publishers AssociationAwards for Best User Interface in a New Program and Best Early Learning Program (presented on stage by Bill Gates, who went on to create a competing product), and the *MacWorld* World-Class award for Best Education Program.

Hebru Version of Kid Pix

Hebrew and Japanese versions of Kid Pix

Kid Pix was also **sold around the world**. There were versions in about a dozen languages, including Finnish and Hebrew. Since Kid Pix was already bilingual, it was rather easy to add another language.

I found that my life was changing significantly. I was regularly being interviewed and getting job offers. People were going out of their way to **shake my hand.** I even made a trip to Japan where I was honored, interviewed and offered much **sushi**. It seemed that Kid Pix had touched people in a way the software seldom does.

Later there were add-ons to the program, and a new version, Kid Pix Studio, but that should be the topic of another story.

Japanese version of Kid Pix

Here are the credits as they appeared in Kid Pix 1.0, 1991.

Program By

Craig Hickman

Product Manager

Leslie Wilson

Graphics

Michelle Bushneff, Craig Hickman

Sound

Jonelle Adkisson, Craig Hickman

Tom Rettig

Manual

Susan Meyers

Color printing technology

Copyright © 1990 Equilibrium.

Programmed by Dave Theurer.

With Thanks To:

Nanci Buck, Dorothy Cribbs, Phillip Ivanier

Elsa Kendall-Hicks, Dick Koch, Kathryn Kruger-Hickman

Rebecca Knievel, Judi Mathis, Nancy Morris

Ken OConnell, Paul Scott, Matt Siegel, Tracy Tandy

Mike Taber, Jack Trainor, Dave Theurer

Richard Whittaker, Harry Wilker, Woods and Woods

Brøderbund Quality Assurance Staff

Brøderbund Design Services Staff

Kid Pix Fans

Alexandra McCullagh, Peter McCullagh, Jennifer Cribbs

Marcella Evans, Marianna Evans, Eric Garcia

Geoffrey Garcia, Ben Hickman, Megan Knievel

Marisol Negrete, Flannery Quinn, Carissa Picchi

Cosmo Rettig, Elizabeth Taber, Scott Taber

Henry Whittaker, Sasha Wilson

Duncan Wold