Oral History of Andy Hertzfeld and Bill Atkinson

Interviewed by:
Grady Booch

Recorded: June 8, 2004
Mountain View, California

CHM Reference number: X2786.2004

© 2004 Computer History Museum

Table of Contents

Andy Hertzfeld and Bill Atkinson

Abstract: On June 8, 2004, Grady Booch interviewed Bill Atkinson and Andy Hertzfeld, early Apple developers, on videotape for the Computer History Museum's Oral History Program. This is a transcript of that session where they discuss the development of the Macintosh and its early software, particularly, MacPaint. The two programmers relate their early computing experiences, how they came to Apple, and share stories of what went on during the early '80s at Apple. They discuss the evolution of technology from the Lisa to the development of the Mac, including descriptions of the design decisions along the way. Atkinson and Hertzfeld talk about the details of creating early software for the Macintosh, and show onscreen versions of the software, including a demo of MacPaint 1.0.

Introduction

Grady Booch: Today, we have a session for the Oral History Project sponsored by the Computer History Museum. In particular, we're looking at MacPaint. There are lots of stories we can talk about within Apple itself. We're going to focus mostly upon MacPaint and what swirls around it doing double duty in this interview because you guys have just some great stories; this is also part of our effort in the preservation of classic software, and what other great software can we point out in the world than MacPaint? This is classic, classic stuff. I'm Grady Booch. I'm not on camera, so I'm just this disembodied voice on the tape. We have here, on my left, Bill Atkinson and then, my right, Andy Hertzfeld.

First, I'm going to ask you guys, what's your current computer? What are you using these days?

Bill Atkinson: I use a G5 with dual two gigahertz processors, about 23-inch cinema display that's accurately color calibrated. I recently did a comparison of the original Mac with my current one, and my current one has 64,000 times as much memory; it has 20,000 times as much processing power and 1.3 million times as much disk storage.

Booch: Oh, my goodness.

Atkinson: And I still complain that it's not enough.

Booch: Andy?

Andy Hertzfeld: I use mainly Macs and I use that one, almost an identical machine, although I think I might have a little less RAM. I only have three Gig.

Atkinson: I have eight. Not that I can use all of it.

Hertzfeld: But, I also do a lot of work in Linux and I use Intel machines, so I have Dell machines as my servers and I have Sony machines as my Intel client machines.

Early Computing Experiences

Booch: What was your first machine?

Hertzfeld: First computer I ever owned or programmed?

Booch: That you ever owned and ever programmed.

Hertzfeld: The first one I owned was an Apple II, serial number 1708, which I bought in January, 1978.

Booch: So, you actually had to buy it?

Hertzfeld: I bought it. I wasn't an Apple employee then. I was one of those people who was led to Apple like a moth to the flame; the Apple II attracted me to Apple.

Booch: Very cool. And, Bill, what was your first one?

Atkinson: My first one was a little, teeny board prototype thing, and then very soon after that I got an Altair that you had to solder in the 21L02 memory chips. Then I moved up to an MSAI which had 20 slots. I could actually put 16 memory cards in it and have a whole megabyte of RAM.

Booch: Ooh.

Atkinson: That was in the days where you had to assemble. The memory cards came with a bag of chips and the circuit board, and you had to solder them, and you had to decide whether you were going to pop for sockets or not because the memory was pretty unreliable. But the sockets were unreliable too, so it was a tradeoff.

Booch: That's great. Well, it goes without saying, that these two guys are part of the soul of the Macintosh software, and, as we talked about before we started here, Burrell Smith on the hardware side, and Steve Jobs – his spirit is very much in this. But, Bill, you're the guy behind MacPaint, so we're going to spend some time on that one. Let's go way back. Where were you guys born? Tell me a little bit about your education.

Personal Backgrounds

Atkinson: I was born in Iowa, but moved out to California before I was one-year old, so I have no memories of Iowa. I grew up in Los Gatos, which is not that far from here.

Booch: Has it changed much since then?

Atkinson: Oh, yeah. Well, when I left to go to college it was the Santa Clara Valley. When I came back eight years later to work at Apple, it was the Silicon Valley.

Booch: Where did you go to school?

Atkinson: I went to undergraduate at UC San Diego, where I studied chemistry and biochemistry, and University of Washington at Seattle for my graduate work in neuroscience. And I got sidetracked. I was using the computers to run my experiments and sort of got more involved in the computers and ended up getting lured away from my neuroscience, and my friend from San Diego, Jef Raskin, invited me down to Apple.

Booch: So, when did you first start programming then? What was the first program you ever wrote?

Atkinson: Oh, it was probably something for a beginning programming class down at UC San Diego.

Booch: Do you remember any professors in particular that were important mentors for you?

Atkinson: Don Norman, who was not only teaching programming but had a good way of looking at computers. Oh, I guess a bigger influence would be Jef Raskin. Jef really had a healthy disrespect for the computer, and I think that has served me well in designing user interfaces for people. Here's an example: Each department had a budget for computing time that they were supposed to spend at the computer center. Jef Raskin took his and bought mini computers, beanbag chairs, and video terminals and set up a little computing center with just his budget that was supposed to pay for time; he had a much better environment where students could come in and write programs and see the results right away, when everybody else in the school was submitting a deck of punch cards to an operator and coming back the next day to find one semicolon out of place. I know Jef Raskin got in big trouble over it, because I think the computer center could see the writing on the wall that, if everybody followed suit, they wouldn't have their budget.

Booch: Do you remember what that computer was?

Atkinson: Oh, it was a little Nova, and it was running a basic interpreter.

Booch: Very cool. So that brings you up to the graduate program. Andy, tell us about your history.

Hertzfeld: I was born in Philadelphia, lived there until I was 18. I went to Brown University as an undergraduate, was a student of Andy Van Damm to whom I just donated a little bit of money – his former students bought him an endowed chair at Brown. He's retiring, I guess, this year.

Booch: Very cool.

Hertzfeld: I was exposed to graphics at Brown. But I actually started programming in high school. We had just one of those TTYs, with the yellow paper and all, connected through just the ordinary phone line to a GE time-sharing service in Valley Forge, about ten miles away, so I learned to program in first Basic and then Fortran. That was, I guess, 1969.

Booch: Wow. And then how did you wind up in California?

Hertzfeld: I came out to California to go to grad school at UC Berkeley, so I was at UC Berkeley at the same time Bill Joy was there. Often he was sitting in – they were rewriting UNIX just constantly at the terminals.

Booch: Very cool.

Hertzfeld: I was out here when the micro computer revolution started. I was the kind of programmer who always used to like getting to the bottom of things, and I didn't have super-user status on the time-sharing computers there. I wanted my own computer and so checked out the Altairs and the IMSAIs, but I wasn't handy enough with the soldering iron. So, when the Apple II happened, I knew it was for me.

Booch: So, after you both graduated, what were your first jobs, then?

Early Apple Days

Atkinson: Well, I came to Apple in 1978, and I was hired as the application software department. There wasn't one. And, actually at the time, I was a little better at pushing chips than software but that's what they needed. So, okay, I can do that. And my first job was mostly kind of a librarian to take software that other people had contributed and clean it up, and publish it for other people. Fairly soon after that, I worked on the portfolio evaluator, a Dow Jones program, because Apple had this advertisement showing hubby at the table with an Apple II with all these bar graphs of stock prices, and his wife is at the sink beaming at him, and there was no such program. It was all just drawn on the screen. There wasn't a real program, and they wanted it to be real, so that was the first big program I did for the Apple II.

Booch: How big was Apple at that time?

Atkinson: When I came to Apple there were 30 people there. I was a graduate student at the University of Washington and Jef Raskin kept saying, "Oh, you have to come check this out. There's something cool starting here." And I said "No, I got stuff going up here." Finally, round trip airfare showed up, and he said, "No strings attached. Just come and visit for a weekend." "Well, okay, it's your money." So I came down, and Steve Jobs spent the better part of the day with me putting on his best sell for why I should come down and work at Apple based only on Jef Raskin's high recommendation of me.

Booch: What were your first impressions of Steve? You had never met him before, I presume.

Atkinson: I remember I wanted to show him my credentials or portfolio, and he didn't want to look at it. So, okay – he doesn't really care about me. But, what he said really hooked me. He said, "You know, you're up there in Seattle and you're reading about brand new products coming out, and you think that's hot new news, but really they were made two or three years ago. You're just at the end of a delivery chain. And, if you want to shape the future, you have to be ahead of that lag time. You have to be where the things are being made. If you come to Apple, you'll be able to help invent the future." And, I remember thinking, oh this guy's laying it on a little thick, but he wasn't. It was actually true, and the thing that drew me to Apple was this notion that you can do something with your life. Making a dent in the world is what Steve used to call it. You can have an impact for the positive if you are where things are being created. I think it was within that weekend, that I called my wife and asked, "What would you think about living in the Bay Area?"

Booch: What was her reaction?

Atkinson: Well, since she had moved up to Seattle and I followed her, it was sort of my turn, but she was graceful about it. She arranged to finish her law school at the University of Santa Clara.

Booch: So, from the time that you said yes to Steve and came on down to Apple, how much of a gap was there until you were onsite doing things?

Atkinson: Maybe a week.

Booch: Wow, very fast turnaround. So, by then the Apple II team was rolling along. What was Apple doing at that moment in terms of their plans? Had the Mac even been born as an idea yet?

Atkinson: Oh, no, no, no, no, no. I remember at a certain point we reached a build capacity of 100 Apple IIs in a day, and the big joke we said was, "That's a pretty hot Apple turnover."

Booch: Oh, dear.

Atkinson: Jef Raskin was complaining that he was supposed to be able to put his big pipe organ in the factory. The factory had a tall ceiling, and Steve wasn't letting him do that after all, or something like that. We didn't have a floppy disk. We had a cassette tape. We didn't even have what we later called the language card. We didn't have Apple Soft in a ROM card. It wouldn't fit, so we had to actually load Apple Soft into memory off of a cassette tape hoping the cassette tape wouldn't fail – every third time you try to read it, it would fail and you'd have to start all over again. The computers were very static sensitive; if you walked up to the computer and touched it, it was very likely to go zap, and there goes all the work you've been doing. So, you'd have to keep saving and saving on these little cassette tapes, because we didn't have anything else to save them on. Steve took me around to meet everybody at Apple, and I remember they all had pretty high energy, and I was impressed by that. And it also seemed to me that there really weren't any experienced programmers there. Most of the people learned backyard hacking kind of stuff, and nobody even thought about ideas like structured programming or modular programming. I remember the second major task that I did was helping to port the UCSD Pascal interpreter over to the Apple II. UCSD had done most of the work. They made a 6502 interpreter. All they needed was the BIOS – the input/output system to plug in the graphics, plug in the disk, and plug in the serial ports. So, the little bit of adaptation needed was a fairly small amount of work for the benefit that we'd get.

Booch: And that was Ken Bowles and crew wasn't it?

Atkinson: Yes.

Hertzfeld: That's right.

Atkinson: I remember I wanted to adapt to it; I thought this is exciting. Only a little bit of work would bring this whole system over, and my manager – by that time, the application software department had grown – was Roy Martin, I think. And he said, "Nope, can't do it." So I went around him. I went over his head to Steve Jobs and said, "Look, it's really important we do this." Steve didn't quite understand. He said, "Well, I think that our users are happy with Basic and assembly language, and we don't need anything else, but I'll give you two weeks to convince me otherwise." That day I was on an airplane, and in less than two weeks, I came back from UCSD with a working system. Mike Markkula sat down in front of it, started playing with the word processor, and that sealed it in. We got the UCSD Pascal system.

Booch: Very cool.

Atkinson: That was the system we built the Lisa with. Without that, we had no way to write modular code where you could write a sub-routine that would do a job, and somebody else could call it without knowing how it worked inside, because our basic had only global variables. There was no local variable. In fact, with Apple Soft Basic, only the first two characters of each variable were actually used. You could have a long variable name, but only the first two mattered. So, we actually kept these charts of A through Z by A through Z, and have we used this one up? Because you have a very different looking name but it really stomps on top of another one with the same name. So, one of the problems was how someone was going to write a program using these, when there was no way to have local.

Hertzfeld: Well, Watts' synergy basic which predated it didn't have that problem. It had up to 256 character variable.

Atkinson: Yeah, yeah. But it didn't have string operations and a lot of ….

Hertzfeld: Floating point.

Booch: So, once you ported that, is there a call because I remember using it on the Apple? That was in released by Apple?

Atkinson: Yes, that was released by Apple.

Hertzfeld: And you had to get a piece of hardware to use the Pascal.

Booch: I remember that now.

Atkinson: Right, you had to get the language card, which was basically a card full of more memory. Really all it was, was –

Hertzfeld: 16K.

Atkinson: RAM that you could –

Hertzfeld: Yeah, bank over the ROM space.

Atkinson: Right.

Booch: Did that unleash a whole set of developers who could now do cooler things?

Atkinson: Well, it allowed us to attract the programmers. We had to attract professional programmers who were used to structured languages, and we never could have attracted them if all we had was basic and assembly language.

Hertzfeld: Although the UCSD Pascal never really delivered the big applications to the Apple II, because it was straining at the memory resources of the machine and had a fairly complex run time, although Macs couldn't necessarily run it, so the killer apps when they emerged like VisiCalc –

Atkinson: There weren't any Macs then.

Hertzfeld: That's right, yeah. The killer apps were written close to the bare machine because they sort of had to be.

Booch: With the presence of UCSD.

Atkinson: You're not going to have machines that had 16K bytes of memory. You got to be careful how much you use up.

Booch: But the presence of that actually helped launch the Apple II and programming language courses, didn't it?

Hertzfeld: Absolutely, yes.

Atkinson: Yeah.

Booch: Because it was cheap and people could do cool things with it. Now, you hadn't gotten on the scene yet had you?

Hertzfeld: Yeah, I started at Apple in August of '79. I bought my Apple II in January, '78. I was a grad student at UC Berkeley, but it quickly just took over my life. The Apple II was so much more fascinating to me than my courses, and so I quickly became an assembly line rich programmer, and started writing little utilities, and it led me to the cottage industry that was the personal computer market back then and eventually to Apple to sell them a program in December, 1978. That's when I met Steve Jobs, and he tried to recruit me to work at Apple. I have a story about how I wish I had accepted his initial offer because I waited six months. At that point, I was in a Ph.D. program already for three years, and I thought I shouldn't just quit without any degree, so I stayed an extra six months to get a Master's degree, write a Master's thesis, but that cost me a four-to-one appreciation in Apple stock that I probably should have accepted it earlier.

Booch: So, you were employee number what?

Hertzfeld: I was employee number 435, but there were actually probably about 280 people at Apple when I started in August, '79. I collaborated on a program with Bill before I met him though. He had been experimenting with digitized pictures and had released on floppy disks, just to the Apple community, these beautiful pictures like Lady be Good and Winston Churchill and the rest.

Booch: Can you show us some of those?

Atkinson: I think they're in this binder.

Hertzfeld: And they were like gold to me because I had no way of making high resolution images myself.

Atkinson: I had taken a Western Union desk facsimile machine that had a rotating drum, and it was really kind of binary. It was meant to copy text, so I replaced it with an optical system that could actually sense analog and analog-to-digital conversion and an adaptor for my LSI-11 computer that I had wire wrapped. I could tape pictures on the drum and bring them in.

Booch: Oh, very cool.

Hertzfeld: Yeah, so those were the pictures.

Challenges of Images

Atkinson: With the Apple II--

Booch: Oh, my gosh, that was on an Apple II?

Hertzfeld: Yeah.

Booch: Incredible.

Hertzfeld: Well these were photographs of <inaudible>. But they were the same images.

Atkinson: Right.

With the Apple II, the challenge was you have very low-res graphics that are kind of blocky, or you have this high-res, really, as basically just a one-bit. How can you take gray scales and represent them on that? And that was always the challenge. And immediately we started into something that now they could call stochastic screening, but there we called it, I think, aero diffusion.

Hertzfeld: Well dithering, you did the dithering first. At least by the Apple II, it was just order dither and then you got the aero diffusion in there later.

Atkinson: Was that later?

Hertzfeld: Yeah.

Atkinson: Okay.

Hertzfeld: The very first ones on the Apple II were other dither.

Atkinson: Just ordered, okay.

Booch: I want to go back a little bit. Don Knuth has the story of how he was writing his multi-volume series and got tired of how books were laid out, so he stopped and created TeX..

Hertzfeld: Yeah.

Booch: You clearly have this love for beautiful images as manifested in your contemporary work as well. Where did that come from?

Atkinson: When I was ten years old my mom got me a subscription to Arizona Highways magazine, and that was the end of it because I saw these beautiful, glowing leaves, Aspen trees, and I would cut out some pictures and put them on my wall. I noticed that I got some kind of nourishment from them, and I told myself when I grow up I want to make pictures that can inspire and nourish people. So, when I was ten I made my first black and white dark room, and I would tromp around taking pictures. My mom was an obstetrician, very busy woman, but she encouraged me. She would make time in her busy schedule. Before I had a driver's license, she would drive me out to Boulder Creek and sit there knitting and reading so that I could tromp around taking pictures, and she commissioned me to make some big prints for her waiting room in her doctor's office.

Booch: Very impressive.

Atkinson: So, I was into photography long before I was into computers, and even while I was working on software, I tried to print on the weekends. It's kind of a very relaxing, in a different way. You know, programming is very much a left brain activity, and you've got to juggle 20 different things and be very attentive, because if one of them falls that's a bug and it's so much more work to find a bug, than it is to just stop it from happening. And, going in the dark room and sloshing the prints in the trays was a very, very restful thing, creative and right brain – it took me away from the stress of programming, so I always liked that. And, I guess it wasn't until maybe 1995 that I really had time to pursue my nature photography full time, and actually founded Bill Atkinson photography and pursued how to make them best. I was always frustrated at the level of quality that I could get. In high school, I realized that I was drawn to things that needed color. I think that sand dunes and nudes work great in black and white but rainforest and flowers and lots of things in the world are full of color, and it's the color that makes them, so I was really attracted to color. In high school I took a job printing in a portrait studio, Arthur Mintz Photography, so that on the weekends I could use the color dark room there to print my nature photographs.

Booch: So, for a long time you understood intimately the challenges of getting those beautiful images in nature down to a piece of paper.

Atkinson: I could make a beautiful transparency and when I went to print it, I couldn't do anything near as good as the transparency.

Booch: Very interesting.

Atkinson: And for a long time I pursued masked Ilfrachrome printing, Cibachrome printing and then even dye transfer printing with the famous landscape photographer Charles Cramer. And, I still couldn't get what I wanted. It wasn't really until the digital printing process that I was finally able to make a print where I could look at the transparency and look at the print and say, yeah, that's captured. It does it justice. It's a fair interpretation of it.

Booch: So, let me jiggle the mouse here for a minute to wake up the Mac, and I want you to put your book side-by-side that to flash it up there. Here we are with about a 20-year gap – so what's different between these two?

Atkinson: Well, let me see, one is a one bit, the other 24 bit.

Booch: Right.

Atkinson: Images that have – what screen was it – 32K bytes or something like that?

Hertzfeld: Yeah, 22K.

Atkinson: 22K on the screen here.

Hertzfeld: For the original Mac.

Atkinson: And these images are 275 megabyte original captures with no interpolation, so it's a little bit different.

Hertzfeld: Factor of 10,000 or so.

Atkinson: Yeah, 10,000 times more information. One of the things that my first wife kept bugging me about was why you want to take a perfectly good-looking photograph and scan it and turn it into this horrible approximation of it? And I never really had a good answer for that, until finally, I could use the digital printing process to give me better prints than I could ever make in my wet dark room.

Booch: Right, right.

Atkinson: So, the computer has become actually the tool of choice for any fine-art printer now. Any photographer who is doing color printing, the tool for the …..

Influence of MacPaint on Imaging Software

Booch: And I presume Photoshop is a key tool you use?

Atkinson: Yes, absolutely. Photoshop along with ColorSync, the color management that allows us to match the colors between different devices; if you move from a transparency to a monitor to a printer to a different printer, keeping the colors the same is the big job even though the numbers have to change dramatically.

Booch: And what an amazing influence MacPaint had upon Photoshop because it was in a sense the ultimate predecessor to Photoshop.

Atkinson: Well, I think MacPaint showed us a tool palette. Why a tool palette versus a menu? One of the problems was modes. When I was on a Lisa team, and they were designing a graphics, form/graphics editor, they had learned that modes were bad. Here's an example: existing word processors you would get into delete mode, and now wherever you move the cursor like a machine gun those characters would get deleted; a better solution was select, harmlessly select, the range of stuff and then say "Delete that." Well that's a lot safer. That's a modeless way. But, the Lisa editor was modeless in that you would stretch out a rectangle of marching ants, another of my inventions –

Hertzfeld: And you mean the graphics of LisaDraw.

Atkinson: Yeah, LisaDraw. You would stretch out a rectangle of marching ants, and then say "Put an oval in it" while you were stretching it out. You'd get no feedback that you were making an oval. And when you thought about making a curved line, that became ridiculous, stretching out a rectangle that was going like this and now saying "put a curved line in it," that was a joke. So, I was arguing that we needed to be in a mode for the creation stroke, like you grab a crayon and you draw with it and every stroke you make a crayon mark. You pick it up. Instead, a marker and every stroke is a marker or an eraser and these are really modes. You're in eraser mode. Larry Tesler even had a license plate "No modes." People had this dogma about no modes, so they couldn't see behind that to ask, is this actually a practical way to do things? Does this work for people? So, I had to make a little mock up. I realized, okay, if there's going to be a mode for the tool that you're in, that mode has to be in front of your face and visible, so that at least it isn't a hidden mode. If you select a tool from a menu, it's not really in front of you like a palette, and that was the origin of a tool palette; I wanted to put it out there on the screen. You can see at any moment at a glance, what tool am I in? Is this stroke going to erase some stuff or going to draw with marker or et cetera?

Hertzfeld: And even before that, you had the pattern palette because that was in even some of the very earliest prototypes, so you already had the palette. The leap was using it for the tool mode, because like you can see in some of the early pictures in the book, you already had the palette.

Atkinson: That's right, yeah.

Booch: Matter of fact, let's set the context here to bring it back up to the Lisa. So here you were in the Apple II. Tell me about when the idea of Lisa was born and how you got in the midst of that.

Birth of the Lisa

Atkinson: Well, the original Lisa project was started in late 1978, and original specifications were that it would be more for business users, and we tried to use a direct manipulation metaphor, and we tried to do it more graphically. The Lisa design goals sort of evolved over a period of time.

Booch: Who led the Lisa team?

Atkinson: Several different people. The first guy was Ken Rothmueller. There was Tom Whitney and, well, Steve was really there first.

Hertzfeld: Yeah.

Atkinson: Steve Jobs.

Hertzfeld: But he never had an organizational position with the Lisa group. He was VP …..

Booch: So he wasn't the project manager per se then?

Hertzfeld: No, but he was a big influence, especially at the shift to the graphic user interface. That's a great story. You should ask Bill about that. That happened before I happened, the famous Xerox Park visit.

Booch: Let's hear the real truth.

Hertzfeld: The most interesting thing, really, is how much we got from Xerox Park.

Booch: So, Lisa had been rolling along before Steve visited Xerox is that correct or was it –

Hertzfeld: That's right, for almost a year.

Atkinson: The visit to Xerox was in late '79.

Hertzfeld: Right.

Atkinson: And before that, our predecessors – the first thing would be Ivan Sutherland's sketchpad program in 1963. This was really the first computer graphics – I knew about that when I was at San Diego in '70.

Booch: Did you know Ivan?

Atkinson: I didn't know him. I knew of his work. I had read some papers of it and my mentor there, Kent Wilson, was a fan of his.

Hertzfeld: And Bud Tribble who was a little later, but he knows Ivan Sutherland very well.

Atkinson: Well, later Kent got this Ivan Sutherland picture system, and Sutherland came there to help install it but that was long after '63. Then, in 1968, Douglas Engelbart at Stanford Research Institute, not at Xerox but at Stanford Research Institute, invented a system that was initially called Online System NLS – it was not for beginners. This was for a professional knowledge worker. And for that, he invented the mouse and the window and wrote papers about it. In 1970, eight years before I came to Apple, I was using a mouse down at San Diego. Now, it isn't quite the same as the mice we have today. It had two wheels that were orthogonal to each other, and when you dragged it on the table one of the wheels would roll while the other was kind of skidding along behind. So, if you tried to draw a 45-degree line, you got this kind of Etch-a-Sketch weird thing. And it was okay for pointing; if you were pointing at a hunk of text, it would work fine. It wouldn't have been okay for drawing, and it really wasn't until 1983 with the Lisa mouse from Apple, that Hovey-Kelley's designers put a ball in the middle of it and redesigned the mouse, so you could draw a smooth line.

Hertzfeld: There was probably one before that. There was this guy at Xerox was making mice for Xerox before then. I think they had a ball.

Atkinson: Was it a ball?

Hertzfeld: Yeah, but his mouse was an order of magnitude more expensive. Hovey-Kelley did the first one that was mass producible, I would say. I think they had a ball.

Atkinson: I remember on the Lisa project, one day I was just talking with Steve, and I was complaining. I was saying that this our hardware manager had this kind of a no-can-do attitude. He was saying "Well, I can't build a mouse. Nobody has ever built a mouse commercially. Show me a mouse that I can buy." And I was complaining a little to Steve. The next day I came in, and he was gone, and I thought, oh God, I got this guy fired. I didn't mean to. About a week later, a new hardware manager came in to interview, and he sat down with me and the first words out of his mouth were "I can build a mouse." Oh, God!

Booch: That's wild.

Atkinson: So, I kind of throw that up to Steve – the attitude of "I can't build a mouse. Who's ever built one? Show me one for real?" And when I show Steve my ideas for a touch plate, he says "Well, anybody ever built one of these?" I say, "Wait a minute, you sound like Tom Hong."

Booch: So let me get the dates in. When did Lisa, the project itself, manifest inside Apple?

Atkinson: Late '78.

Booch: Okay, late '78, and then the famous visit to Xerox was?

Atkinson: Late '79.

Hertzfeld: December, '79. There were actually two visits.

Atkinson: Oh, yeah, and some other things in between.

Hertzfeld: Close together.

Atkinson: So, I said at '68, the windows and mouse were invented by Englebart, and I actually went to visit Englebart. It was a wonderful visit, and I think he gave a really good argument which was that when you design for the beginner, sometimes you shortchange the experienced user. If you were designing only for a walk-up-and-use-it experience, you would never design a bicycle because it takes a while to learn to use a bicycle, although it gives you good power. So, he was designing more for a professional knowledge worker, and he's willing for people to learn this five-key chord thing and lots of new ways of working with stuff. In 1973, with Alan Kay and the Smalltalk Project at Xerox Park was where another big boost of user interface development happened. In 1979, when the Lisa team went to visit, we got to see the Alto and the Smalltalk System and I think the Bravo text editor. What people misunderstand is that we didn't just copy what we saw. It gave us great inspiration and gave us great confidence that, yes, we did want to do windowing, but then we had to go incrementally, evolutionary-wise and develop this user interface a piece at a time by a lot of trial and error and a lot of stupid mistakes.

Booch: How big was the software development piece of that in terms of number of people back then?

Hertzfeld: At Xerox?

Booch: No, at Apple at this time.

Atkinson: By the time Lisa shipped it was 100 people, but at that time it was probably 15 people.

Hertzfeld: When they started the Lisa team in December, '78, I think they hired six people for that. I don't know if that counts you or not, but Tom Malloy was one of them. Chris Franklin was one of them.

Atkinson: I know Steve wanted me to go over to that, and I didn't want to because I was afraid that Pascal would get it better.

Hertzfeld: By the time of the Xerox visit, I bet you there were twice that many, maybe about a dozen. I think the best story I know about the Xerox trip is something Bill thought he saw but didn't actually see which was drawing behind Windows. They had overlapping windows but once windows overlap you have a non-rectangular area for the ones that aren't on top. And Bill was sure he saw drawing into the irregular area.

Atkinson: I had seen an existence proof that you could draw quickly –

Hertzfeld: So, he knew that it was possible to do but, in fact, that was not the case.

Atkinson: That gave me the drive and motivation to come up with the region clipping. And later, one of the people from Xerox said "Well how did you come up with that?" and I said, "I thought you were doing that!" "No, we never did that. If we drew on the back one, we just had to redraw all the windows in front and it would flicker like crazy."

Early Graphics and User Interface Design

Booch: So, your role on the Lisa team was to do what precisely? What were you tasked to do?

Atkinson: Graphics primitives and user interface design. I wrote Quick Draw, the original window manager, menu manager and event manager and I tried a whole lot of user interface stuff and that's why I brought this binder of user interface trial and errors. We only had a few of these prototypes. The prototype was basically a graphics card that would go in an Apple II that let us access a frame buffer, much bigger than a normal Apple II had. And, there were only a few of these, and I had one at my home where I did my writing. I couldn't write at Apple. It was just busy. But in order to communicate the results, I would take Polaroid pictures and bring them into the team and discuss them and talk about it.

Booch: And that's what some of those pictures are?

Atkinson: Yes, and they show all kinds of steps along the way, all kinds of blunders, and things we tried and things that didn't work. Here were the first windows and here was the first pop-up menu, and then here was the first scroll bar and here, until very late in the game, I was confused about the difference between a folder and a document. I mean now we know a folder is a container you put documents into and maybe other folders. And, generally, you might write on the tab but you don't write on the body of the folder. Well, I was confused about that so our windows, which were documents, looked like folders.

Hertzfeld: For a little while.

Atkinson: There's lot of steps along the way toward the design of the user interface there.

Booch: You had virtually nothing to guide you. You were in the wilderness here. So what influenced you to make these particular design decisions? Did you look around and say, "Hey, I'll try this?"

Atkinson: Larry Tesler was big on this, and what really helped us was user testing. We wanted a beginning person to walk up and be able to figure it out – we talked about it being intuitive. What did that mean? Somebody walking up to it and without reading a manual could figure out how to do something. So, how could we test that? We set up a room with a half silvered mirror and a dark room where the experimenters would be and a light room where the subject would be, and they'd have one video camera pointing down at the keyboard and the mouse, another pointing at the display screen such that you could see the face reflected in it. And we'd give them tasks and say "Here, edit this document and save it." And we also asked them to mutter under their breath a stream of consciousness. What are they thinking about? And that was very important because why they do something is just as important as what they do. For example, we got to a point where this person had done a bunch of editing and was supposed to choose between “do it” and “cancel” and he looked a little miffed and hit “cancel.” And we thought, what's going on here? We saw that happen with another person, and they also had this sort of reaction – they looked a little miffed and then hit “cancel.” So we turned up the volume, listened carefully. The person was muttering under his breath saying, "What's this dolt? I'm no dolt." So they hit cancel. We knew that it said "do it" and they saw it as dolt. We had a blind spot because we knew it said "do it," we couldn't possibly see that as dolt. It just didn't look like dolt to us but it did to several different users. And we hadn't wanted to use okay, because that was a colloquialism and an abbreviation and not even a correct abbreviation but the bottom line is that we changed it to "okay" and "cancel" and then everybody sailed right through it. So, thousands of these kinds of tests where you find that people made mistakes are what led us to the user interface.

Booch: Some of the major players you can think of?

Key People in Early Development

Atkinson: Well, Chris Franklin, Rod Perkins, Bruce Daniels.

Hertzfeld: Rich Page.

Atkinson: Rich Page was the main hardware guy.

Hertzfeld: And Paul Baker.

Atkinson: Paul Baker, yes. We had a place that was behind the Good Earth Restaurant. And so they called it Salt of the Earth.

Hertzfeld: Why was it called salt of the earth? They asked me that in my book when I mentioned it. Scorched earth was called Scorched Earth because all the people there smoked.

Atkinson: The hardware guys smoked. All of the hardware guys smoked and none of the software guys smoked. It was very interesting.

Booch: Was Woz on the scene at all?

Atkinson: No.

Hertzfeld: Woz actually did the initial bit slice design on the back of a napkin kind of saying it was possible, and got him started on it a little bit.

Booch: So your primary development tools at that time were still UCSD Pascal?

Atkinson: Yes, running on the Apple II.

Booch: And so you'd have a host target.

Atkinson: Using that to develop for our target, yeah.

Booch: Was it ever to a point where the Lisa was self hosted – where you could develop on the Lisa?

Hertzfeld: Yes. By the end of 1980, we had Lisas.

Atkinson: Right. We had these bug development boards to test the 68,000 processor, so we had those hooked up to one of our displays, and that was then the Lisa development environment. In fact, the UCSD Pascal disk format was used.

Hertzfeld: Yes.

Atkinson: Where block contiguous, that was called, we called that the monitor format.

Hertzfeld: Yeah, Rich Page did that.

Lisa Development

Booch: So, give me the size and shape of the Lisa. How much memory did you have to deal with? How fast was it?

Atkinson: Well, the screen was 720x320 pixels but stretched and actually in the product that they shipped, there were scan line pointers. We were worried about being able to scroll, to do a smooth scroll in real time because there's all these pixels to set and clear. So, we figured, okay, we can't do that in software. We'll have to do it in hardware. So, we had a frame buffer that had a pointer at the start of each line and just changing those 360 words of pointer would cause everything to scroll. The problem is that if you only wanted to scroll a window's worth instead of the width of the whole screen, they were useless. So I think they shipped in the original Lisa but they were never used for anything.

Booch: So it was all done in software?

Hertzfeld: That was the earliest part of the Lisa that was designed. The frame buffer never got redesigned.

Atkinson: Right. Part of the reason why the frame buffer was constrained to 720x360 is that it was 32K of memory or less. It was enough that it was one set of chips, and adding more would have blown that hardware. It had this horrible problem. I had just done the UCSD Pascal block move which was the copy bits thing for that, and either you were copying from main memory to main memory where it was linearly arranged or main memory to the screen where it was kind of haphazardly arranged, and it was kind of a weird arrangement of the bytes in memory to save some chips on why this is designed.

Hertzfeld: That's the Apple II.

Atkinson: On the Apple II. Or from the screen to main memory or the screen to the screen. There were all these different cases you had to deal with, and once we had the scan line pointers, it was going to reproduce that all over again. You couldn't just do adjacent bytes and one row bytes down. I'm glad we switched. Rich Page was really instrumental in getting us off this horrible bit slice design and onto the Motorola 68,000 processor; it was amazing that Motorola got working silicon the first turn. That wasn't par for the course in those days. And this was a processor with full 16-bit address space instead of this little 8-bit processor and bank switching things on top of it and trying to deal with all of that. It was also such that you could write assembly language routines for it, and you could add those later. There was general purpose instruction set in its own micro code-supported assembly, and that's what really allowed Quick Draw, which was originally called LisaGraf, no LisaDraw – LisaGraf?

Hertzfeld: LisaGraf.

Atkinson: It was originally called LisaGraf. It allowed it to be fast enough to support these applications without having to do a whole lot of stuff down in bit-slice land, which was a really slow development.

Booch: And you wrote all of LisaGraf, LisaDraw?

Atkinson: Yeah, I wrote all of that myself.

Booch: Assembly or Pascal?

Hertzfeld: No, mostly assembly.

Atkinson: There were a few headers that were Pascal calling framework but it was all assembly. The guts to do any of the work, all the loops and stuff had to be assembly.

Hertzfeld: And so Rich Page started that 68,000 transition in around the summer of '79 and it was kind of completed around the fall as I remember, so you could say Lisa had a false start. It went for a year before the 68,000 transition happened. The Xerox PARC visit was shortly after that, so there was a first year of just kind of feeling out what to do. But then by the end of '79, the two main things – the 68,000 and the adoption of the mouse with the PARC-like interface, windows – all happened around the end of '79. By the middle of '80, we actually had Lisa prototypes and you were programming the Lisas on Lisas. And the Mac, of course, was developed using the Lisa as the development machine.

Booch: Right, it's an interesting boot-strapping process here along the way. Now Lisa shipped on what date?

Hertzfeld: January, '83, I believe January. There's a difference between announcing and shipping. Lisa had quite a lag. I think Lisa was announced around middle of January, 1983 but it didn't ship until June, quite a lag because –

Macintosh Development

Atkinson: So, six months later the Mac was introduced.

Hertzfeld: And the Mac team didn't have much fondness for the Lisa hardware. The Lisa hardware was messed up in lots of different ways, and in some ways, the Mac was taking what they were doing –

Atkinson: Streamlining.

Hertzfeld: – and getting it right.

Booch: But they learned a lot from the Lisa team.

Hertzfeld: Yes. Well, hardware-wise, in some respects, just the broad things but not the design. The design was odd. The Mac design did not flow out of the Lisa hardware. It was more like the Apple II where you had a crazy genius coming up with very unorthodox techniques not used anywhere else, putting it together. Burrell, who designed the Macintosh digital board, really learned from Woz. The Apple II was the immediate predecessor of the Macintosh hardware, not the Lisa.

Booch: So, sketch out for me the time line there – I thought they were sequential in happenings but I'm hearing it really was quite an overlap.

Atkinson: A big overlap and kind of a race.

Hertzfeld: So, we also talked about the Lisa timing. For the Mac timing, Jef Raskin in the fall of '79 got the approval from Apple – mainly Mike Markkula gave the high level approval, then Tom Whitney, the engineering level approval – to set up a three-person team to research. So, Jef started lobbying for it before that in March of '79 but he got the resources to put the team together in September of '79, and actually Bill was the one who introduced him to Burrell. Jef needed a hardware designer, and Burrell helped Bill when they were using UCSD Pascal to write the Lisa code. They were running out of memory by that middle of '79. They had used up all 64K, all of it.

Booch: So, how did you know Burrell then?

Atkinson: Actually I think I met Burrell –

Hertzfeld: In the service department.

Atkinson: Yeah. I'll say along with the UCSD Pascal system we switched to 16-sector disk routines instead of 23-sector, which would give us more space on the disk on the UCSD system. But we had some problems with that, and unbeknownst to us, the manufacturer had changed a capacitor in the motor so everybody was blaming Woz for his new 16-sector disk routines, and that wasn't the problem. It was really a hardware problem that had been added. But, in order to decipher that, I worked with Burrell on trying to find what was going wrong with the floppy disk that was preventing the UCSD Pascal system from shipping.

Booch: You said he was in the service department then.

Hertzfeld: Yes, Burrell has no college degree. He was in one of the lowest paying jobs at Apple, just fixing broken hardware.

Atkinson: He was great. He was brilliant. I think at one point I brought him to Raskin. I said, "I think I found your hardware designer."

Hertzfeld: But in between, Burrell made the 80K language card for you.

Atkinson: Yes, that's right, which was essential also to the UCSD Pascal system.

Hertzfeld: He found a way of banking in more memory to the Apple II disk, and he wasn't even an engineer, just a guy in the service department, but he impressed Bill doing stuff like that.

Booch: Wow!

Hertzfeld: And so he introduced him to Jef.

Booch: So, ultimately, then we had the Lisa project, and shortly thereafter the three-person team to begin the Macintosh.

Booch: Were the goals of the two similar or different? How would you characterize the difference?

Comparison of Lisa to Mac

Hertzfeld: Very, very different.

Atkinson: Very different.

Hertzfeld: Similar along the dimension of easy to use. Jef had a real passion for ease of use. Lisa wanted the same thing. But other than that they were –

Atkinson: By the way, much of my passion for ease of use is attributable to Jef Raskin, to his disrespect for computers and his respect for people's time. Computer processing cycles aren't valuable. People's time is valuable.

Hertzfeld: But, in most other senses, they were diametrically opposed because the Lisa was intended to be for an elite office user. The initial Lisa's price was supposed to be $3,000 and it later shipped at $10,000. But even so, the initial Macintosh price was supposed to be $500.

Booch: I'm trying to remember where was VisiCalc in the midst of this?

Atkinson: That was long ago. That was in Apple II land.

Hertzfeld: No – the Apple II disk drive didn't happen until July, '78 and that was a big enabler. VisiCalc I think came onto the market in early '79. And then by the fall of '79 it was making Apple sales triple every month.

Atkinson: A funny story about that is that Apple was offered lock, stock and barrel exclusive rights on a spreadsheet for $50,000, and they turned it down.

Hertzfeld: Yes, and they turned it down. What would anyone use it for?

Atkinson: They could have owned the whole spreadsheet concept.

Booch: Wow! Because when Lisa shipped it had a spreadsheet application, did it not? And then a drawing application?

Atkinson: Yes.

Hertzfeld: Yeah, Lisa had seven different applications all developed by Apple. So that was another way the Lisa team diverged from the Apple II. One of the characteristic things about the Apple II, to me at least, was the third party market. The Apple II had slots so it was hardware expandable and had a vibrant third-party market. With Lisa, because it was pioneering a new way of using it, it was harder to work with the third parties, and so the idea was all the Lisa applications would be written by Apple. And so you get a different spirit, whereas the Mac brought it back home. It combined the Apple II spirit and thriving third-party, and Burrell and Woz are similar type designers – the crazy genius instead of the conservative committee.

Atkinson: It was still a closed hardware box. You know it got a lot of things for being a closed system. The goals on the two were very different. One main thing is who it was designed for. With Lisa, we were designing for an office worker and since we weren't office workers ourselves, it was kind of hard to know exactly what they wanted, and we were kind afraid to offend them. When the Mac was designed, I think we had a pretty clear picture of a 14-year-old boy using this thing, and we knew what they were like, and so we were going to have fun with that. And so, I think one of the biggest differences when the Mac finally shipped is it was the first computer that people really fell in love with, and people actually would say, "Gosh, I want one of those. How can I justify it to my spouse?"

Hertzfeld: But Jef's Mac vision was extremely consumer – computers for the masses. Let's get the price as cheap as we can and pump them out by the millions.

Atkinson: Information appliance. Also, the notion that every device would be the same. It's like all the computers would have the same amount of memory. All the computers would run the same software, and I think some of that carried over even after Steve Jobs took the project away from Jef and turned it into a volks Lisa, which is really what happened. Some of that was still left over in Lisa, left over in Steve's mind. The Mac would only be 128K ever, and that it was not okay to have it expandable to 512. Actually, Burrell went behind Steve's back to make it so there was one trace you could cut, and you could stick in the 512K chips. Steve later had to apologize and say he was wrong.

Hertzfeld: That's a really important point that the Mac was not conceived to be a little Lisa. That came later. We had a different processor in the original Mac, the 6809 chip, and even though Lisa was preexisting at the time the Mac was conceived, the Mac was conceived to be something very different.

Atkinson: Yeah, the eight bit processor would have been dead in a year.

Hertzfeld: Just like the Lisa had that false start of a year before it really found what it was, well the same for the Mac in that way. The Mac was going for well more than a year, like a year and a half, before it found what it really was, – the volks Lisa – taking the Lisa technology but making it for everybody.

Booch: Let me wrap up some questions on the Lisa and then move to the Macintosh. Reflecting back on your experience with the Lisa, what were you most proud of that you had contributed, accomplished or delivered through the Lisa? What was the real zing thing?

Atkinson: Well, one piece of engineering that I was proud of was QuickDraw's ability to draw really quickly and clip to arbitrary areas. I think I did 100 times better than anybody was doing at the time, and I came up with some clean ways to do that.

Booch: How did you come up with those?

Atkinson: Well, I thought about it a lot. I tried some things. I tried different things. I realized that if you did it by polygon clipping, which is the way that people did – they'd take polygons and then clip things to that – if you just sort of calculated out how many instructions it was going to take in the time of doing one floating point add, which was in software (remember we didn't have a hardware floating point), we needed to be able to draw five characters, so you couldn't use floating point calculations to do clipping stuff. I remember breaking down the problem into smaller and smaller pieces and saying what's really important is being able to work on parts of the problem independent of other parts. So, in the region structure I came up with, one point toggles the inner outness of everything below and to the right of it, and so just a set of these points could describe an area but in such a way that each point could be dealt with completely independently. If you're at a given place and want to know if you're in or out, you just have to look up and to the left and see if there's an even or an odd number of points there. And, instead of like tracing along the outline of a contour, you had these individual isolated points that could give you, particularly if you stored them in a raster order and you needed the clipping information in a raster sequence, tremendous efficiency that, hey, this scan line is a lot like the one above it. It shouldn't cost us anything until we get to something those changes.

Hertzfeld: QuickDraw had to be able to clip to the intersection of three regions really, really fast. Every graphics operation was clipped to three. Now, it's easy to special case them when they were just rectangles but often it had to do that compositing on the fly of three different regions.

Atkinson: So, I'm proud of that. I'm proud of the pull-down menus. I thought that the pull-down menu was a very good solution to providing visibility, spatial memory of where the commands were. I remember when I moved the pull-down menus — they were on the top of the windows and I moved them to the top of the whole screen – I got a lot of push back from the Lisa application writers. "Oh, they're going to reach farther and it's too much work." And my arguments were, well, if the window is very narrow, where did all your menu items, menu titles go? And, if the menu isn't very tall or if it's down near the bottom, where do the items go? If I put them at the top, I'll have the full width, I'll have the full height, and they'll always be in the same location. It's not like they're floating around based on where the window is.

Hertzfeld: And even better, you can just go to the top; because of pins at the top you didn't have to think about positioning.

Atkinson: It didn't always pin at the top. One of the things that made it work was pinning at the top and making it actually a gear speed change. So that if you moved the mouse quicker, it would go more inches. The ratio of millimeters on the table to millimeters on the screen would increase when you went fast. So, you could do it with a little bit of flick, reach up there quickly.

Hertzfeld: Yeah, mouse scaling is important.

Booch: And that's such a distinctive Apple feature to put that out there. Didn't Apple patent that?

Hertzfeld: Yes.

Atkinson: And the other nice thing about it was that because the titles of the menus were always much narrower than the items, it basically tripled your effective screen. You could scan along the top and see all available commands without doing any of them. You could harmlessly see what is available. But as they were flopping down, you see only one would be flopped down at a time – they were really two-thirds overlapped, but you don't really perceive that when you're working with it. It feels like you've got a lot more screen width, like three times the screen width, than you really do.

Booch: So, in the Lisa days what is it you regret not having been able to do because you were limited by hardware or software or whatever?

Atkinson: Well, a design I regret is the double click. I should have listened to Tom Malloy. We were having battles about one button. The arguments were real clear. When we tested with users, whenever there was more than one button they'd spend more time looking at the mouse than at the screen. The only way to be able to stop looking at the mouse and look at the screen was to cut it down to one button. But one argument Tom proposed that I should have listened to was having one button but having a shifter button on the side of the mouse with shift click. I think they would still look at the screen because there was only one clicker, but it would have shift click instead of double click, which was very frightening to a beginner because you had to do two within the same amount of time and two within the same amount of space. Otherwise, when you're doing click and click and click and other stuff, then it would think it was a double click. So, double click I wish I hadn't done. I wish I had gone with Tom Malloy's suggestion of a shifter.

Booch: I find it amusing that earlier this week I discovered Macintosh has patented the double click for Microsoft. For certain classes of machines.

Hertzfeld: I told Bill about that.

Atkinson: I think that in user-interface design, anytime there is a fixed time window, it's wrong. Even in movies when the credits scroll, they're always too slow for some people and too fast for other people. There isn't one right way – with any kind of a user interface, a fixed time period is always going to be a problem.

Booch: So, did the need for internationalization shape some of the design decisions in the Lisa? Was that a goal of the Lisa?

Atkinson: That was more on the Mac. I think that Joanna Hoffman was primarily responsible. I mean she speaks – what – five languages? And her passion was to make this thing so that out of the box, we can localize the software for all these different countries. She insisted that no text strings, “please enter your name,” would be in the program. They would be in a separate part called the resource board that could be edited separately by people who didn't have the source to the program. They could make up new strings in French or German or Italian.

Booch: So that's what led to the design decision of resources?

Hertzfeld: I have to give credit to Bruce Horn as well for coming up with the resources.

Atkinson: Yes, he implemented the resource management.

Booch: So the internationalization is what led to the resource distinction?

Hertzfeld: One of the big drives was Joanna's passion for internationalization, and then it was also from Bruce's experience at working on Smalltalk and Xerox and whatever. It was, in a way, just a little stab at objects and yes, the internationalization, and we ended up using resources more and more through the course of the design.

Booch: So that was a Mac thing, not a Lisa thing.

Hertzfeld: Lisa didn't have that. Lisa was more just let's write the application-- rewrite the application.

Atkinson: What we were looking at before the Lisa was really command line oriented. You had to type in Del or REM or Remove, and then try to remember what the right command was and then try to remember the final …….

Hertzfeld: So, Lisa had the function key thing so you don't have to type, but it would have a row of buttons at the bottom, right near the keys to label them ……

Atkinson: On Lisa?

Hertzfeld: Yeah, the original Lisa.

Atkinson: I'm talking on Apple – on Apple II.

Hertzfeld: Yeah.

Atkinson: And on an IBM PC there were no menus. The biggest virtue of menus was you didn't have to remember what the darn commands were – you could explore and see, that's the one. You could choose. And, likewise, when choosing a file, you could see the names of files that were there and point at one instead of having to try to remember what you saved that file as and type it and it says, "No, I can't find that file," and you try another one, "No, I can't find that one."

User Interface Development

Booch: So, all these user gestures sort of lead up to the user experience. Who began to codify the Apple interface rules? I would imagine they came as you developed them and vice versa.

Atkinson: One guy who played a big part of that was Larry Tesler. One of the jobs he did was to say, "We've got these seven applications running on the Lisa. I'd like all their menus to be similar, right? If there's a filing menu, let's make the items in it like "print" and "save" and "save as" to be the same between the different applications." That was important for leverage. You can walk up to a new Mac program you've never seen, and you know how to use half of it already because it's the same as all the others. That consistency of the user interface was one of Larry Tesler's passions and gave rise to the user-interface documentation for the Lisa, which had user-interface guidelines.

Hertzfeld: In my recollection, there were two marathon meetings that were a year and a half apart. The Lisa one was in the late summer of 1980 with you and Larry Tesler, maybe some other people – Jef says he was there and people say Jef wasn't there, whatever. Bill and Larry had meeting of the minds about the basic elements of the Lisa user interface such as the way text is selected and all of that, and that got codified and written up. But they didn't have the user interface guidelines so much because there weren't third-party developers. It was just the Lisa team, and Larry was talking to each of those people every day, so there wasn't as much of a need to formalize it.

Atkinson: They were formalized. There was a formal Lisa user-interface guidelines document, and the person who managed it and edited it was Marion …..

Booch: But you're saying they weren't really broadcast outside?

Hertzfeld: Yeah, it wasn't. It was in the company and was just a little different. For the Mac, one meeting I remember we had, and I remember you being there, although I'm not sure how much you remember. It was in the beginning of January, 1982. We were giving our first prototypes. We had a build of the first 50 Macs in February, '82. We were going to give them out to developers, and so we realized we had to write down what is the user interface. We had two-and-a-half days; we all went into some office that wasn't the normal building, and Steve was there as well. Chris Espinosa was another one. We didn't use the Lisa document as the starting point, but obviously all the concepts were from that. We just wrote them down. And Joanna Hoffman, who was the marketing person Bill mentioned, got the job of writing that, and that was the first draft of the Macintosh user-interface guidelines. After that, Chris Espinosa took it over and redrafted it and took it to shipping.

Booch: Probably much later, a guy named Tog?

Hertzfeld: Yes. Okay, here's a debunking or something. Bruce Tognazini had absolutely nothing to do with the development of the Lisa or Macintosh user interfaces initially. He was working in the Apple II part of Apple, and actually had Hodgkin's disease, and was gone from Apple during the time. But when he came back in late '83, he had a real passion for usability always, and later as it evolved, he founded the user-interface group at Apple. So, while he gets a little bit too much credit for saying he founded the user-interface group, he didn't design the user interface. The user-interface group was founded two years afterwards.

Booch: Very interesting.

Hertzfeld: So he was not involved at all with the initial creation – I think he would say that if he was here – but he was one of the main people who helped evolve it going forward.

Booch: He has very strong opinions about user interface.

Hertzfeld: Yes, and I think good opinions. I really think he's pretty smart about it. It's just that he should maybe qualify when he said he founded the user interface group. Yes, it was in 1986 two years after the Mac shipped.

Atkinson: Well it's like Jef Raskin-- Jef Raskin claims to be the father of the Macintosh, and I would give him that he's the father of the Macintosh project. But Steve was the father of the Macintosh. He basically took it away from Jef and said "We're going to make it this other thing."

Hertzfeld: Jef conveniently fails to mention that he had left Apple before most of the Mac was developed.

Booch: So, Jef provided some the DNA here.

Hertzfeld: Yes.

Atkinson: Yeah.

Working on Macintosh

Booch: So, when did you guys officially become part of the Mac team?

Hertzfeld: Well, for me and Bill it was different.

Atkinson: He was there right away.

Booch: You were always from day one?

Atkinson: I was sort of this bridge member. I was really the only person that was a member of both the Lisa and the Mac teams, mostly because of QuickDraw and because my window manager, event manager and menu manager got ported over, cleaned up and made smaller and improved on by Andy for the Mac.

Hertzfeld: Basically my job was to take what Bill had done for the Lisa and make it work in the Mac with much tighter memory constraints.

Atkinson: What portion of the original Mac ROM was code that I had at least started?

Hertzfeld: Well, if you count the tool box code it's probably more than half. Quick Draw alone was 24K, toolbox added up to another maybe 12 to 16K, but the toolbox code I essentially hand translated from Bill's Pascal into assembly language.

Booch: What was the developing environment for the Macintosh then? I was on Lisa and then--

Hertzfeld: The Lisa running, the Lisa monitor with a serial cable as an umbilical cord. That was the initial--

Atkinson: Downloaded into the Mac prototype.

Hertzfeld: -- the initial one, and then by the fall of '81 we had the monitor running stand alone on the Mac, so the Mac could boot up from a floppy and you wouldn't need the Lisa to demo it.

Booch: So, how would you characterize the software developing culture for the Macintosh?

Hertzfeld: Free wheeling. Bill was really the center of coming up with the user interface, but he worked at home so he would come in – I would say maybe two or three times a week – usually when he had discovered something new. We would all gather around and talk about it, give him feedback.

Atkinson: I'd get good suggestions from other people and say, "Oh, that's a good idea."

Booch: Then you'd go back home.

Hertzfeld: It was very loose. It was on the Lisa group that there were a lot more philosophical arguments about what is the best way to do it in the abstract, whereas with the Mac, it was much more, well, try it out and see how it feels like every single step of the way.

Atkinson: But like even on the Lisa, how do you determine which way the scrolling arrows go? Okay, if you're going to scroll a document so you see more of what's below, do you use an arrow that points downward to see more what's below? Because if you're going to see more what's below, the document has to move up to see more of what's below right? So, it kind of could go either way. Do you plan on an upward pointing arrow or a downward pointing arrow to see more of what's below?

Hertzfeld: Which is moving the viewer or the document.

Atkinson: Right. And so we just tried it. I was a big fan of “let's try it and see what people expect it to do right.” And what I found was it was really 50/50. They'd be startled about half the time. It proved the opposite way they thought. But if I put the arrows at opposite ends – they used to be right together, the verticals, and they used to be on the left – they always did it right. If they wanted to see more of what was below, they'd press on this one because it was near more of what they'd see. They always did it right, and independent of where I pointed the arrows, they'd reach to the right place. Now it was less efficient because if you overshoot, then you got to reach way the hell up here to undo it. And now our computers actually have an option. For an advanced user, you stick them together. For a beginner, we leave them apart.

Booch: Or they have a scrolling mouse as well.

Atkinson: Yeah, we tried things like that over and over; user testing was necessary. Like the grow box and a window, it used to be right up here. How it worked is you would press in it. You'd drag out a rectangle the size of the window you wanted. When you let go, it would jump to that. The window tab you could double click to minimize it down to a little window tab, and double click that to open it up. Well, if you double clicked on this, but missed and hit the little grow tab, suddenly the window would jump down only a teeny little bit and people really thought they'd broken the computer. "Oh, my God, what did I do?" Because if you had seen that happen on previous computers, that means the video circuit crunched. You lost the computer there. Well, what I found is the only way that really solved that was putting the grow bars down in the bottom right corner, which forced two very un-optimal things because we had to move the vertical scroll bar over to here, which was farther away. Text tends to be left centered.

Booch: And Smalltalk had it on the left side, too, as I remember – didn't it in its browser?

Atkinson: Well it's closer to the text. That's the practical reason – if you put it here, and if you required every window have a horizontal scroll bar. At first they weren't required because how many of them need it? A word processor might not need it. We required then both scroll bars, and they'd be grayed out if you weren't going to use them, and it gave a structure to the window, separated it from the rest. It also gave us a place to put that grow box that made sense to people. If they dragged it a little bit, it would change a little bit. When they let go, no sudden change happened. It was pretty much what they were already seeing.

Booch: Yes. Proportional scroll bars because Smalltalk would have those and the Mac had ……

Hertzfeld: Here on the original pictures, we had that first; well, we had them off and on. We had them, and then didn't have them. Here's the first scroll bar – the first scroll bar had a non-proportional ….

Booch: On the left side.

Atkinson… on the left side. A little while after that, I tried a proportional scroll bar where the amount of white here is proportional – if this is the whole document, this is the portion you're seeing. What our experience was is that people clicked in the gray part. They thought that was the active part. They didn't know that this white, big white area was something they could drag on. It didn't have affordance. It didn't have a thing that said, "Here, touch here and drag." Now we would put little lines that show it's grabbable, you know. The next week we went back to a non-proportional one because they understood what it was. Many of the design decisions were for people who had never used a window, a mouse, a menu. How do we make them walk up to this thing, and without reading a manual because they don't, and be able to figure out what it was they were facing, and just giving them a task, could they figure out how to do that task? Simple tasks should be something you can figure out without any help, and with more advanced tasks, maybe you have to learn something, then.

Steve Jobs Takes Over Macintosh Project

Booch: So, when did development really proceed at pace on the Mac where you said, "We're really going to drive this to closure here? We know we've got a product."

Hertzfeld: Oh, when Steve took over...

Booch: Which was when?

Hertzfeld: It was a research project. Steve took over January, '81. Let's just backtrack a little bit. There's just a really key connection that we've glossed – Jef hired Bud Tribble who was Bill's good friend at UCSD.

Atkinson: UC San Diego. And then when I went to Washington, he moved up there and he did his Ph.D. program there while I was doing my neuroscience studies.

Hertzfeld: Jef had originally hired a guy named Marc Le Brun, who is reasonably well known in the LISP world, to be the only software developer for the Mac, but he just didn't work out. He didn't like programming small machines, so he just kind of sat there ….

Atkinson: He was used to big LISP machines.

Hertzfeld: … with the 6809 with 64K of memory on his desk for about seven months in 1980, and nothing happened, no software development happened for the Mac besides Jef writing his thought pieces, but not really programs. The key thing was Bud starting in October of 1980.

And really the most key thing was the close personal connection between Bud and Bill. Bud lived at Bill's house, and so here he is watching Bill developing this incredible graphic code and the Macintosh having a different processor, so we couldn't even use it.

Bill Atkinson: Here's a funny picture. This one shows two phase non-overlapping clocks – Bud and Bill. His waking hours were during my sleeping hours.

Hertzfeld: Bud liked to work late.

Booch: Twenty-four seven development.

Atkinson: So we'd leave notes for each other. The Polaroids were partly for him. Hey, this is what I'm doing.

Hertzfeld: So, Bud began pushing Burrell asking him, "Hey, can you put the 68,000 in the Mac instead of the 6809 so we can use all of this great code Bill has been writing? So, Burrell can – the resistance to that was the Mac--

Atkinson: Oh, Jef didn't want that.

Hertzfeld: -- the Mac was supposed to be extremely low cost and the 68,000 required a 16-bit memory bus, doubling the RAM, and that was just thought to be no way, 16 bits, they're expensive computers. The Mac had to be an 8-bit smaller computer. And Burrell came up with this circuit he called the bus transformer, where in a very small little hardware hack, he was able to multiplex. The RAMs had-- this is getting a little too technical. The RAMs had page mode, a way of accessing. Normally to access a RAM, you have to column address then a row address, but the RAMs had a special mode where one column addressing, you gave it two row addresses really fast and could get twice as much out.

Atkinson: So, take eight chips and you could still stick 16 in and out.

Hertzfeld: He used the page mode access of the RAM to feed the 16 bus with only eight ram chips, and that was kind of key discovery. He said, "Oh, we can use the 68,000 because we can still just use the eight RAM chips."

Atkinson: Of course how many RAM chips did the 128 Mac have?

Hertzfeld: Yeah, I know, I know. Well, eventually it changed. That was later down the road. But in December of 1980, Bud had convinced Burrell. Burrell made a prototype. And then that prototype was a really hot machine because suddenly we had a machine that's more powerful than Lisa. Burrell, because he designed it a little bit later, he used an eight megahertz clock instead of the five megahertz clock. Eight megahertz, that was fast!

Booch: What's a G5 these days?

Atkinson: That was .5 mps. Our G5 is 10,000 times as fast as that. And there are two of them, so it's 20,000 times as fast as the original Mac.

Hertzfeld: And that's not even counting the difference in the buses.

Atkinson: And probably 30,000 times as fast as the original Lisa. Not counting the disk which on Lisa was five-to-one inter lead, horrible. You'd get one track of disk; it would spin five whole times. Of course, it had hardware protected memory domains. It had multi processing that couldn't step on each other. If one process dies, it can't kill the others. It had some things that were a little bit brain dead like when you asked for a hunk of memory, it would go and write zeros onto the disk. Instead of just putting flags saying, "Pretend this is zero. If anybody asks for it, read it as though it is zero without having to go to the disk and write it." So it was very slow.

Hertzfeld: The real transition point of the Mac was when Burrell did the bus transformer circuit. They used the 68,000. Suddenly we had a computer that was a quarter of the cost of a Lisa, just in the chip count, that was actually faster, and it could run the same software, and that attracted Steve Jobs. He saw this thing could be the future of not only Apple but the computer industry. Just a month or two before that, he had been elbowed out of the Lisa project because Steve has extremely strong opinions and will always create an interference with an organization. Steve doesn't respect the organizational authority, and so he was just kind of wreaking havoc there, and they told him "Please stay away." So he was kind of looking. So he found this little Mac prototype and--

Atkinson: But, of course, that made then a Lisa/Mac competition thing, and Steve was gunning to make the Lisa fail.

Hertzfeld: So, Steve's first act as taking over the Mac – even though he had just started, and there were no schedules – he bet John Couch who was running the Lisa group, $5,000 that the Mac would ship first even though the Lisa project had been going for two years. The Mac was just getting started.

Atkinson: I wonder if John ever collected.

Hertzfeld: I think he did, actually. I remember after the Lisa introduction he collected.

Booch: He could have taken it in stock.

Hertzfeld: They had a photograph with both of them.

Atkinson: Part of why the original Mac couldn't take more than 128 bytes of memory is because Lisa was this bad thing that had grown to 512K.

Hertzfeld: A megabyte when it shipped.

Atkinson: Really?

Hertzfeld: Lisa required a megabyte when it shipped.

Atkinson: Oh, my goodness. I remember at one point it was 512K, and Jobs says "This isn't craftsmanship. This is just slop and they're just"--

Art of Developing Software

Booch: So what makes software beautiful for you? Do you care about the beauty of software?

Atkinson: Oh yes, it's an art form, like any other art form, and it's not just practical, as in, does it do the job? Is it clean inside, does it look--? I would spend time rewriting whole sections of code to make them more cleanly organized, more clear. I'm a firm believer that the best way to prevent bugs is to make it so that you can read through the code and understand exactly what it's doing, that there's nothing tricky in it, that it's all very straightforward. And maybe that was a little bit counter to what I ran into when I first came to Apple. There were a lot of people who prided themselves in how this little piece does something that I can't for the life of me figure out what it is, but it's magic that it does it. I would do things like deliberately assign something into a variable, and instead of putting ten operations concatenated onto one line, I'd use intermediate variables so I could understand the purpose of each of these. And I liked it. I found that if I spent time going over the code, cleaning it up, making it sometimes tighter – I did a lot of measurement and tried to make the performance good – but also making it so that it was straightforward so another person could follow in my footsteps, then I would feel proud of it.

Booch: Don Knute was telling me that he felt that MacPaint was perhaps one of the most beautiful programs he has ever seen, just in terms of its readability. So who influenced you to develop that literary style, if you will?

Atkinson: Well, the first person was Bud Tribble. Back at UC San Diego, I would type punch-cards for him. So he would write the program and I'd type; he could write really quickly and I would go do the dirt work. We were working together on a project, so we split up the labor. But I learned that when he wrote assembly language, every line had a comment on it, and a comment wasn't just, what does this instruction do? It was "why am I doing this?" It's a bit like here, calculate word count, branch if all in one word, do left word with mask, flush left mask for N, decrement word count, branch if unmasked words. And every statement has a comment on it. It's not like you put a comment every now and then to remind you what you're doing.

Booch: To sidestep for a minute, this just triggers in my head, because I know Charles Simonyi was with you guys in the midst of this –

Atkinson: No he wasn't with us.

Booch: He wasn't?

Hertzfeld: Well no, but we interacted with him.

Booch: What was your relationship to Charles?

Atkinson: My relationship to Charles Simonyi was via Tom Malloy, who was the guy writing the word processor on Lisa and had trained with Charles Simonyi on a programming documentation style that we called Hungarian. It meant you take all the verb- all the nouns--.

Hertzfeld: The vowels – no vowels.

Atkinson: All the vowels out. Like Rgd-wd-lp-fst-lp-rlp, would mean a range of words from first to last and they all meant something, but you couldn't read it.

Hertzfeld: In his defense, he was not doing that because he was demented, he was doing it because he felt the type of each variable should be encoded in the variable name. So he came up with a discipline for having the types, in getting his language ……

Atkinson: Well, he had to do that because the language he was programming in was not structured, it wasn't typed.

Hertzfeld: Well, actually I'm not sure about that.

Atkinson: The integers and reels weren't even separated in the language he was working in.

Booch: What was that language?

Hertzfeld: He was proposing –

Atkinson: BCPL?

Hertzfeld: At least by the time he got to Microsoft he was programming in C where he did have it that way.

Atkinson: Yes, but it was BCPL that he was doing it in.

Hertzfeld: Yes, probably.

Booch: So he was actually part of the Apple team for awhile?

Andy Hertzfeld: No Charles, no, never.

Atkinson: Tom Malloy was the lead programmer on the Lisa word processor.

Hertzfeld: And Charles was his mentor. He brought the Hungarian into Apple, and Bud and Bill and a lot of the programmers at Apple hated the Hungarian but –

Atkinson: Couldn't read it.

Hertzfeld: But we did work with Charles on the Mac because Charles went to Microsoft, and we first met with them in the summer of 1981, and we gave them the prototype in February '82; we were working with them a fair bit. They were one of the first serious users of our frameworks. And Charles was one of their leads.

Booch: He'd written Bravo before those days.

Hertzfeld: Yes. And Tom Malloy worked on Bravo for Charles.

Booch: So given the Microsoft relationship you were working in, did Bill get it, and did he really understand where you guys were headed?

Hertzfeld: Yes, well he didn't get every detail but definitely, when he saw the Mac and the graphical user interface, he believed in it, put a lot of resources on it at Microsoft. Microsoft was really helpful at tweaking some of our rough edges. They had almost as many people on the Mac as Apple did, for awhile.

Atkinson: Tell him about Steve shushing you up about the graphics accelerator.

Hertzfeld: I have this story all ready to tell, about the first time Bill Gates saw the Mac. Bill doesn't like to have people explain things to him, he likes to guess the way it works, he likes to tell you how your program works, and he sometimes is right and sometimes is wrong. But the first time he saw the Mac he started telling us how the cursor works, which is what kind of sprite hardware do you have for the cursor? And of course we had no hardware at all.

Atkinson: Only software.

Hertzfeld: So I was about to tell him that. In fact, what had happened just two weeks before that visit, myself and a few other people at Apple had gone in en masse, working on an Apple II, which had 1/10th the processor power of a Mac. So I was about to tell him, "Are you kidding, you don't need hardware, we can even do it on an Apple II." But Steve knew I was about to say that and didn't want Bill to know that you could do it without special hardware. So he just said "Shut up" as loud as he could to try to drown me out.

Booch: So Bill and these guys are wondering, what's going on here?

Hertzfeld: Yes.

Atkinson: Well, Jobs didn't want Bill to know you could do this all with software, i.e., you could do it with those existing processors out there.

Booch: So this is what year?

Hertzfeld: Well, that meeting was July 1981 when we first demoed the Mac. There's another interesting part of that story at that point you needed the Lisa to download the software into the Mac. You couldn't run a Mac stand-alone yet. But we weren't allowed to show Bill Gates the Lisa because the Lisa team was developing all its own applications. He was like the last person in the world we wanted to see a Lisa, but we did want to show him the Mac. So we had the Lisa in a different room with a 25-foot serial cable, and I had to start the thing, and then start running.

Atkinson: Start the download and run around.

Booch: Right, this is one of those ignore-the-man-behind-the-curtain kind of …..

Hertzfeld: Yes.

.**Atkinson:** Bill was no dummy.

Challenges Surrounding Macintosh

Booch: Were there any dark days you experienced in the Mac world at all?

Hertzfeld: Oh, absolutely.

Atkinson: There were a lot.

Booch: What was the darkest for you?

Hertzfeld: To me the darkest days were when someone who was really important to the project quit, and so one of the darkest days was when Bud quit, in the early stages. In December of 1981, he decided to go back to Medical School. Bud was on the Mac, just on leave-of-absence from a six-year MD/PhD program that he had already done four years of. So, he was heavily invested in it, and Jef convinced him, that it'll only take a year to do the Mac. So he came down and …..

Atkinson: A couple of years later –

Hertzfeld: He starts getting these nasty letters from his advisors saying, "Should we keep your place open in the program?" And he was kind of tortured about it. He didn't want to leave, but he decided he had to leave the project. He took Earl and me out to dinner at Earl's favorite restaurant that served sushi, and broke the news to us. We were thinking the whole thing was going to fall apart without Bud. The whole Mac team at that point still only had maybe six engineers, and I thought it was hard to do it without him. And then I had my personal dark point.

Atkinson: Belleville.

Hertzfeld: Yes. In February of 1983, my engineering manager gave me a terrible review, organizational review. At the time, I was doing the best work I ever did, he said …..

Atkinson: He was the star of the team.

Hertzfeld: He said, "Your technical work is adequate but I have problems with….."

Booch: Adequate?

Hertzfeld: Yes. "…..I have problems with your attitude."

Atkinson: He was from a military background and chain of command.

Hertzfeld: Well, no, he was really frustrated because Steve would come every night and talk to the programmers to find out what was going on. So Steve knew more about what his direct reports were doing than he did.

Atkinson: And he'd come every night and talk.

Hertzfeld: He had a bunch of problems. But that one really just broke my heart. I thought I'd have to quit, and I did quit, but I stayed until the next shift, at least. So there were personality conflict things like that. Another dark moment for the Mac team, I remember, was we were using these twiggy disk drives, which were the same ones that Lisa used, and they didn't really work.

Atkinson: They never did.

Hertzfeld: They didn't work for Lisa, but Lisa was low volume enough that it was able to ship with them, kind of hoping no one would notice. But for the Mac it became clear by the spring of 1983, that we're supposed to be shipping not so many months away, and we didn't have a disk drive. The disk drive just was not shippable. And we felt the whole project might collapse in itself.

Atkinson: Oh, tell him about hiding the Sony engineer.

Hertzfeld: Yes. What saved our life was the Sony 3.5" disk, like this.

Atkinson: Really, Mac was the first computer that used those.

Hertzfeld: Well, he shipped it first.

Atkinson: Oh yes? I didn't know that.

Hertzfeld: But anyway, yes. Actually George Crow, who had worked at HP and knew about Sony drives because HP was an early adopter of these things, was saying "Oh, you got to use the Sony 3.5," and brought in the disks. Steve loved the disk, the hard plastic shell and everything. He says, "Okay, we'll engineer our own," which was not what George was suggesting, because we only had three months.

Atkinson: Typical Apple bravado- we'll make our own RAM chips.

Hertzfeld: Steve thought that we could take what we learned from the twiggy, the failing disk, and make our own 3.5 "disk from scratch in three months, which was lunatic. But he insisted that we pursue that, instead of working with Sony. Other people were brave enough to defy Steve and continue working with Sony, knowing that when we tried to make the thing on our own, we'd see at some point that we couldn't do it, and we'd have a backup plan. So, the story goes, we were meeting with Sony engineers, both in Japan and in Cupertino, but we had to hide that fact from Steve. A young Sony engineer named Mr. Komoto was there in Larry King's cubicle talking with him about the disk, when Steve suddenly appeared in the Mac software area, unexpectedly, and we asked him to hide in the closet to avoid Steve. So we all scurried about, and he went into the closet. The punch line to the story was he was able to get out of the closet after five minutes when Steve left, and he said something like, "Oh, American engineering practices are very unusual."

Booch: When did the Mac become a bet-the-company kind of thing?

Hertzfeld: Because Lisa failed …..

Atkinson: Also, when they started stomping on the Apple II – Apple II was the cash cow that was supporting the company, and Steve was pushing the Mac.

Hertzfeld: The Mac was initially a skunk works. When the Mac started, the big projects in Apple's future were the Lisa on the one hand and the Apple III on the other hand; even the Apple II was diminished in their minds, not thinking it would last as long as it did. But the Mac, Apple began to take it seriously in 1982, once we had prototypes built. But then by the spring of '83, it became clear to everyone that the Mac was the future. If the Mac failed, Apple would fail. I would say spring of '83 was when they started seeing that Lisa sales were not going to be what they anticipated.

Booch: As I recall, it was in '83 that you began the notion of having MacPaint, as part of the product – it's bundled with the Mac itself.

Hertzfeld: At the very beginning of '83 was when Bill – because he was finished with the Lisa – was able to transfer full-time onto the Mac team.

Booch: What started the decision process for what was going to go into the Mac release? Had that gotten stable by around that time?

Hertzfeld: Yes, we wanted third parties to make a big contribution, but we thought Apple should write the most crucial applications, mainly the finder, the system shell, and a word processor, because the proportional text word processing we thought was too important to trust to someone else, even though Microsoft was …..

Atkinson: But also exemplar apps to teach other developers.

Hertzfeld: Yes, we wanted just a great graphics application, for which Bill was, of course, the natural person.

MacPaint Development

Booch: So MacWrite and MacPaint were the two –

Hertzfeld: MacWrite and MacPaint – we decided that around early ’82. We knew we’d be working on a word processor and a graphics program.

Booch: So let’s talk about MacPaint for a minute. So you were just coming off Lisa, and you had all the experience of the Lisa – what was it called then, the Lisa Graphics Program?

Bill Atkinson: LisaGraf. Because it was being ported over to Macintosh, it had to change its name. And so I changed it to QuickDraw which was actually the name of Jef Raskin’s Master’s thesis at Penn State, on a graphics system that was very different; it was all vector based. I think Jef might still be mad at me that I stole his name.

Hertzfeld: No, I think it’s the other way around, it lets Jef get a lot more cache for the Mac because he said he …..

Atkinson: Invented Quick Draw?

Hertzfeld: No, he says you named it that in homage to his contributions on the …….

Atkinson: I did, but I didn’t ask him permission. I didn’t ask Ivan Sutherland permission either, for SketchPad.

Hertzfeld: I don’t think Jef would complain.

Booch: So, it was around from the spring? Oh, I guess February of ‘83 is what I had in my notes, to October of ’83 is when MacPaint had its gestation and became a little more stable. Are those dates about right?

Hertzfeld: I’d say that’s about right. There was a lot of introduction.

Atkinson: There were a lot of users. Susan Kare was my main tester, because she would use it, and I would go in and just watch her using it, and notice what she was trying to do.

Booch: Tell me about Susan because she developed some amazing …..

Atkinson: She's a graphic designer, not a programmer.

Hertzfeld: She was my friend from high school. And I was the one who figured out we needed her. I had just recently gotten back in touch with her. We both grew up in Philadelphia, and in the spring of 1982 I reconnected with her. I knew she was a very talented artist; she actually had a Ph.D. in Art History and was a sculptor and made these fantastic sculptures out of welding metal together. But I realized that this was a heavily graphic computer, and it would be great to have someone who was talented at making graphics. At first I got her a consulting job, and what I told her to do was the finest graph paper she could find and to draw pictures, just like coloring squares of the graph paper. So she spent about a month making all these fantastic pictures and that became her portfolio. Then when she came into Apple, the original idea was she would just consult, but she talked to enough people, and they liked her, so she started as a full-time team member in January '83. In fact, when she started, there was something about her hiring that was supposed to be only if she works out would she stay, but then she started doing great. Susan has a great sense of humor. Susan's incredibly smart, which really helps.

Atkinson: The first commercial digital artist, anyway.

Hertzfeld: Susan does more than people give her credit for because her main talent is drawing small bit maps. But she also helps a lot at the conceptual level, figuring out what image to draw. She has a great imagination and a sense of humor.

Booch: So tell me the interplay between Susan and the evolution of MacPaint. Because she was your main …..

Atkinson: Well, she was my main customer, and I would go and watch her using it, and see what she stumbled on or wished she had. When I'd see there was something she was stuck on, or didn't understand or she didn't have the tool that she needed to do what she wanted, that would often give me an idea for a feature to add.

Booch: Do you recall any particular stumbling block that she had that you then were able to remediate? Back bits came out of her use of it, didn't it, or was that something else?

Atkinson: No, I was going to do that.

Hertzfeld: Yes, I recall maybe a little something--. I'm not sure how much on this but …..

Atkinson: There was a menu that had extra stuff, and I was going to call it the "aids" menu. But she told me, "You know there's this thing starting to happen – there is this disease, and I think you better stay away from that word." So, I reluctantly gave it up, and I called it the "goodies" menu.

Booch: As I recall she started doing a lot of little icons in MacPaint as well.

Atkinson: She did an icon for me, and another one for HyperCard.

Booch: Yes, it's right here in fact.

Atkinson: A different one for HyperCard,

Booch: Yes, and that became a badge of honor to have a Susan icon, didn't it?

Bill Atkinson: Oh yes, people begged her to do one of them.

Hertzfeld: The Steve icon was one of the first ones she did, this icon of Steve Jobs that was mysteriously compelling.

Booch: Had this boyish grin to it. Yes, very cool. What did Steve think about it when he first saw it?

Hertzfeld: I think he liked it.

Booch: And she did this hand in MacPaint?

Hertzfeld: Well, there was actually an icon editor that used the fat bits editing style. But she did a lot of them there. As MacPaint got further along, she preferred using MacPaint to the icon editor because it had better tools.

Booch: Yes, and then ultimately that became part of the gestalt for the Mac; you'd see those little icons all over the place.

Hertzfeld: Well, we started using the icons for the finder on the Lisa.

Atkinson: The very first incarnation of MacPaint was on a Lisa prototype, and it was called SketchPad, and it took over the whole screen, it wasn't in a window, and it had one tool. It was a combination of sort of a lasso, but also you could click to make polygon regions, and you could choose a pattern.

Booch: So that's one of the first images from Mac?

Atkinson: Yes.

Hertzfeld: Yes, that's really early that one …..

Atkinson: Way before the Mac.

Hertzfeld: Two years before MacPaint really started.

Atkinson: And then later in the Lisa development. Here's SketchPad in a folder, which we would call a document today, not taking up the whole screen but being in a document. And then later, there are graphical pull-down menus to specify the patterns. It wasn't really until Lisa shipped that I had a free rein to work on MacPaint, with a lot of time.

Hertzfeld: As I recall, the very first thing you did to make it MacPaint as opposed to the sketch program was the off-screen buffers- you didn't have the off-screen buffers in the thing and that was …..

Booch: What led you to that decision?

Atkinson: Well, in order to make it smooth – first, you have to be able to undo. For a real application, undo is essential. I remember on Lisa it was difficult getting the application writers to do undo, and a chain of undos was unheard of.

Hertzfeld: Yes, you had to flip them.

Atkinson: Flip-flopping between the last two states was about all we could hope for. But that was one of the big things about the Lisa and Mac is you felt safe to try something, because you could undo. And in MacPaint in order to undo, you really needed a whole buffer the same size as what you had, so that the one here and the one there, if they zoomed into fat bits, you didn't really have this data anywhere. So you had to have off screen two buffers – the current one and the previous one. The whole computer only had 128k bytes of memory, and we had to deal with how am I going to have the screen, the full screen, plus two full buffers? I had to do a little juggling to make that happen, and I had to make the code.

Hertzfeld: Eventually, there was a third buffer that wasn't allocated all the time.

Atkinson: It had to do with selections?

Hertzfeld: Yes, that's right.

Booch: So you were pushing the limits of the hardware, and I'd imagine blowing the hardware guys' minds?

Hertzfeld: Well, every Mac program pushed the limits of the hardware because it was so limited.

Atkinson: The final version, MacPaint 1.0 that shipped, had a case that you could get to that left it with 134 bytes of memory free. Steve Capps wrote this wonderful thing called the Monkey, which is basically simulated user input at a tremendous rate – it would flail around and do all kinds of stuff. One of my requirements is that the program be able to survive 12 hours of Monkey. If there is kind of goof where you allocate something but forget to de-allocate it, or there's a way that you forget to de-allocate it because you took a sideways exit or something, it would accumulate, and you'd eventually flood the memory. I remember pushing and pushing and pushing, and I was so pleased I could get it to where it would reliably go for more than a day, with the Monkey, all in one.

Hertzfeld: I was hoping the Monkey would draw something interesting but I never really saw it do anything for savings.

Booch: So these days, we have garbage collection tools to look for leaks and the like, but I assume you had none of that? So you were in a fairly poor tools environment.

Atkinson: We also had errors in the memory manager software itself, so I would be finding them.

Hertzfeld: Yes, you have to remember that when MacPaint was written, the Mac framework was changing rapidly. It wasn't until the fall where you could really begin to get it robust.

Atkinson: I had a different method of software developing than Randy Wiggington who was developing MacWrite. We had two different philosophies. Mine was you don't get to add any new features until what you've got is working reliably, solidly; and his was, let's add all the new features and then let's debug them. I just think that that's a bad way to do it because finding bugs is a very unlikely proposition. If you search really hard you might find half the bugs that are there, and the others are going to bite you and you just don't know when. So I had a little more conservative approach to design.

Booch: What you describe is very similar to what the Agile community speaks of these days; always have something executable through incremental development. And it's very cool.

Hertzfeld: Yes, yes, that's really how the Mac was developed.

Booch: You were the first Agiles. So by my count, or others, MacPaint was about 5,804 lines of Pascal and so on; 2,738 lines of assembly language.

Atkinson: Plus all of the QuickDraw I'd built on top, because without QuickDraw you couldn't do all that stuff.

Booch: And pretty much everything had built upon QuickDraw.

Hertzfeld: Yes.

Atkinson: There were 70,000 lines of assembly language.

Booch: Now because you were worried about third parties, there was also this phenomenon of the *Inside Macintosh* documentation that popped up around the same time.

Hertzfeld: Right.

Early Macintosh Documentation

Booch: That was an important factor in getting third parties to use what you were doing as well. Did that become stable pretty much the same time the Mac itself was released?

Hertzfeld: The documentation?

Atkinson: Earl and Rose.

Booch: The *Inside Macintosh.*

Hertzfeld: Yes, we started it but there was a little bit of a lag. There was draft documentation even a few years before the Mac was released, but the documentation wasn't published in the hard bound book until about a year after the Mac.

Booch: I had one of those fix up tools.

Atkinson: The telephone book. Earl and Rose gave her fingers to them. She got carpel tunnel syndrome big time.

Hertzfeld: But we were lucky we found Caroline Rose. I worked with a documenter on the Window Manager documentation, before Caroline started, who didn't want to really understand it, just wanted to regurgitate words that I told them, and it was very, very disappointing that we had awful documentation. And then we worked with a writer to try to improve it, but she couldn't improve it, until finally Caroline showed up on the scene, and working with her was night-and-day difference. Caroline had an incredible drive to understand exactly what's going on, ended up improving my software by her penetrating questions. She'd ask me things I hadn't thought of …..

Atkinson: Here's a condition that will cause it to crash.

Hertzfeld: And so I'd say, "Come back tomorrow and ask me tomorrow," and then just change the API to improve it in between.

Atkinson: Having to teach somebody how a piece of software works forces you to clean it up.

Hertzfeld: Yes.

Atkinson: There were interesting things that happened in making the stuff public. Like, for when you're totally out of memory and have so little memory that you can't even put up an alert saying you're out of memory, we had the Deep Shit manager.

Hertzfeld: That was me – I called it the Deep Shit Manager.

Booch: How do you document this one?

Atkinson: Well, we had all these equates. There's a DS this and a DS that. So, the documentation people scratched their head and said, "Dire straights"- it's all documented as Dire Straights Manager, but we know that it's Deep Shit.

Booch: Oh the truth comes out. So let's talk about MacPaint. You were relating to me a story about Robin Williams …..

Atkinson: Okay, so a little after the Mac shipped, we had a big celebration where Rich Paige and I got anointed as Apple Fellows, and we had Robin Williams come and do a comedy routine for us, and he got a Mac as an honorarium. I remember him up on the stage talking about painting in the snow with his MacPeter. I'm hoping that maybe someday he will interview me about my book, *Within the Stone*. Shameless plug.

Shipping Version of MacPaint

Booch: Let's talk about MacPaint as it was finally shipped. I recall there were some really cool "ahas" in the user interface for it. The marching ants thing, for example: Tell us the story about how that came to be.

Atkinson: Well, if you have black and white pixels, and you can set and clear any one of them, there's nothing that you could draw that would be static. There's nothing we could show for the user interface of things being selected that you couldn't have drawn except for motion. So I figured that the way you had to display what's selected is by something moving, because that's something that the user couldn't draw, anyway. And I wanted not just rectangles with it, because you could select an arbitrary area. How do I display that? The clue that I got was when I was in a bar, and I saw one of these beer signs, maybe Hamm's Beer, that had a waterfall flowing like this, but if you looked at it carefully, there was a cut-out picture of the waterfall that had some little holes behind it, and behind it there was a sheet of plastic that had designs on it that was moving. It was actually on a motor, and it was the sheet of stripes, striped patterns flowing behind this cut-out that made the whole thing look like the waterfall was flowing. Well, that's exactly the technique I used to make the marching ants. There's actually a diagonal pattern that is drawn – when it's being drawn, I look at what time the 60-hertz clock is at. So, it's basically a moving pattern, depending on what time it's being drawn. And then I mask that with the shape of the selected image.

Booch: Who dubbed it marching ants?

Atkinson: I think that Rod Perkins did. Or maybe he called it the marquee.

Hertzfeld: You told me he called it the marquee, and I wrote it down in my …..

Atkinson: I think so, I think it is Rod Perkins.

Booch: Now there were some features that didn't make it into the shipped version – you had some character recognition?

Atkinson: Oh yes. That was a hard one to let go of. See, for an engineer, when you do something really cool, the hardest thing is to step back and look at it in perspective and say, "Is this really going to improve the product, or is this something that's actually neat and cute, but going to make the whole result of people's interaction with the product be worse?" This was a bit-map program, it's set in clear bits. But you could type some characters, and there would be bits that represented characters, but it wasn't like a structured graphics editor that really still remembered what characters were there; there were just some bits. So I looked at how could I get those back? If you typed a bunch of text, and then you could select it and bring it back into a typeable form. So I worked out ways to basically take advantage of whatever fonts were online and pre-cache hashed versions of each character. You would select it into bands of where there was text by just counting how many black dots are on the line, and here must be the baselines. So you got an idea what the spacing of this is that limits what fonts could be involved and then flatten those to find the character boxes, and then within those, make the hashes and all this.

I figured out a way that I could actually fairly quickly and efficiently recover something; in maybe a tenth of a second lag, it would be back into edible text, and you could position and type some more and then seal it back down to bits. But the problem is if there were any stray bits under there, suddenly it couldn't be recovered. If the fonts you used at the time weren't online on this computer, it couldn't be recovered, and more importantly, what it did was create people's expectations that what this thing was a word processor; that it was a word and graphics together processor. And then they wanted to have different kinds of justification, and there are a lot of properties you have in a word processor that you wouldn't have in labeled text. What it would be is a very bad word processor because you could hear a scream from across the room, "Oh no, it's gone now – I set it down on top of some other pixels, and I can't get it back." I was very proud of what I had done, and I got it to be really fast, but I realized that it would actually make the product worse because people's expectations of what they were dealing with would change; they'd think, oh, this is supposed to be a word processor, and it never was, and so I had to take it out.

Booch: That was hard, I imagine. So actually right in front of you, on the point after this, we're looking at some of the original disks from MacPaint. Tell us what we've got here and how it came to be.

Hertzfeld: These are the ones that shipped with the Mac. So they were the very first Macs.

Booch: Did you ship also with a cassette tape of music with the Mac II? Or was that the--?

Hertzfeld: No, that's the *Guided Tour*. Because the mouse was so new, we had to teach people how to use it. So we prepared what's called the *Guided Tour,* and that even involved writing some software to capture events and play them back. There was a tutorial, and a cassette tape was part of the tutorial. The Mac wasn't up to doing the sound internally yet – that would have been too much memory in those days.

Booch: So this was MacPaint 1.0 that shipped with ….

Hertzfeld: It might've not been called 1.0, the one that shipped. But I'm not sure. When I just launched the one from this disk, it said 1.4.

Atkinson: Here's MacPaint 1.3, sources.

Hertzfeld: And these are the sources we donated to the museum, unless Apple gives new ones.

Booch: Yes, it shipped on a diskette.

Hertzfeld: Yes, yes, 400k. I could tell the story of how we came to do this donation, which involves Don Knuth, one of the greatest computer scientists of all time. And it involves the Computer History Museum, because there was an event in January of this year to celebrate the Mac's 20th birthday where they had the Mac marketing team, with Mike Murray, who was the leader of the Mac marketing team leading a panel where most of the key people on the marketing team were up on the stage, and about three-quarters of the rest of them were in the audience. So that was a great presentation, and Don Knuth attended it. Toward the end he stood up and asked a question, which was, "I think MacPaint might be one of the best programs ever written and I'd love to be able to study it. Is the source code available?" And we just said no. But then at the end, Mike Murray invited all the people from Apple to come to the stage. So I went up front, and Don Knuth came up to me and asked me more about that. He said, "Well, I'd really love to see the MacPaint source code. You guys must have a copy or have it somewhere." So when I got home I called – Don Knuth is one of my heroes.

Atkinson: And mine, too. He's a grand old man.

Hertzfeld: Yes. If Don Knuth asks you, you better work hard to try to satisfy it. So, I called up Bill and said, "Bill, you got to have the MacPaint source code somewhere." He says, "No, I lost it all ten years ago when my hard disk blew up." And I said, "Oh, come up, pretty please." And he says, "Oh, okay." And he's busy trying to get his book together – I don't want to ask him to do too much.

Atkinson: So I hunted and hunted around, and I really thought I'd lost it. But I found in a box in the garage, I found the binders. This is original binder of MacPaint source. The first part of it is in Pascal and then a big part of it is in assembly language. And QuickDraw, these are the QuickDraw sources …..

Booch: These are your working documents?

Atkinson: Yes. As I was working on them, I would work on a module and replace it and punch this. For example, here's a section on performance, old, new and ratio of performance, QuickDraw performance, each round. It was always, how can I get this thing to go faster? because the applications were critically dependent on it. So I would test things like drawing a 5-character string- old 28; new 25, and draw a 60-character- old 81; new 66- et cetera. So this set represents an inside view into what I wrote. Now actually, I don't think anybody at Apple got to see this until after a Mac shipped. I worked alone on this, and there was a feeling that these were the crown jewels, and we didn't want everybody to have access to them because they might get swiped or something.

Hertzfeld: Defying the orders of his manager, he gave them to both Bud and me. I had copies of QuickDraw, but you told me I wasn't allowed to have them. So I wasn't allowed to tell anyone that I had them. Anyway, yes, this box is what Bill found. And this box had the source code of both QuickDraw and MacPaint but on this obsolete file format, the Lisa that we were talking about earlier.

Atkinson: It cross-developed.

Hertzfeld: Which the Mac never shipped with. So no Mac ever that shipped could read these disks.

Atkinson: I know Lisa could because it didn't have that shape disk.

Hertzfeld: But, luckily, the Lisa Monitor had such a simple file system. It's a continuous file system.

Atkinson: Basically, it's sort of like the UCSD Pascal system.

Hertzfeld: Macs haven't had floppy disks connected to them since 1996. But luckily I found a 1996 Mac – my constraints were I needed a Mac with a floppy. You couldn't use the third party ones because the Apple disks are strange. You had this motor speed control where they vary the motor speed to get 30% more data on the disk, so, anyway I was able to read them--.

Booch: Where did you get to the next day?

Hertzfeld: The hero of every programmer in my generation, he wanted the source code. Bill was able to come up with it looking through – was it the garage or the attic? – to find this box of the old, old source code.

Atkinson: The binders were in the garage. These were in a back cupboard in my…..

Hertzfeld: Bill was developing with the Lisa Monitor system. It used a file format that was never running on the Mac and really hard to find software; there was no way you could find software to interpret it. But luckily, it had such a simple-minded file system; everything had to be continuous on it. So I was able to read in a disk image and then just load that into the text editor, except where blocks fell short – at the end of a file you might get a little bit of garbage in there, but the rest of it was pretty straightforward. So I could pull the files out of the disk image by just selecting the text. But the problem was it looked really messy, the lines did not line up, and I had a dim recollection in the back of my mind that UCSD Pascal used some kind of funny file system. So when I looked at it closer, I saw there was a byte at the beginning of each line. We forget that now, because plain text is just so pervasive, and we wouldn't think that you would do blank compression on a text file because it's such a small amount of memory. But in those days it mattered. So it turned out there was a count byte at the beginning of each line to say how many blanks were supposed to be there. I ended up writing a little pro script to just read in the blank count and turn it back into plain text. After I did that, I was able to get the text files to look nice, put them on a CD, and send them to Don Knuth. So mission accomplished. Meanwhile I'm working on this project called Folklore, documenting the development of the Macintosh. And about a week after I sent the disk off to Don Knuth, I thought, well, the ultimate way of documenting the Macintosh is if I could put the MacPaint source code on the site. For everything we were doing, the code really is the essence of it. So, if future generations could see the insides of it, that would be so worthwhile. I started thinking about, well, do I have the guts? The intellectual property belongs to Apple, but it's been 20 years; there's absolutely no proprietary reason to keep it secret.

Atkinson: No commercial value to it.

Hertzfeld: I asked Bill first thing, I think, because Bill was really the primary author, he has the moral sway, should we try to release this or not? And Bill was encouraging that we should try to release it. I started talking to other people I knew, especially people who knew Steve Jobs, to try to determine if I could get away with just putting it on my site? Because I knew if I released it to the world, it would get copied a few times on the Internet; the genie's out of the bottle, and they'd never get it back. So, at first I thought what I should do is just put it on the site and take it down when Apple got mad. But I talked to more people, and I saw a few people said, "They might get very mad." And their argument that made sense to me finally was the slippery slope argument. Of course, there could be no damage from releasing source code that's 20 years old, but what if your act inspires some Apple engineer to release source code that was five years old? And it is a slippery slope.

Booch: Tim O'Reilly had gotten involved?

Hertzfeld: Yes, yes. I started calling basically all the smart people I know, and Tim O'Reilly's one of the very smartest, and I asked him about this quandary, should I release it or not? He came up with the idea of why don't you donate it to the Computer History Museum? And he made the initial contact with Grady and others. It took a little while but eventually we got the Computer History Museum to formally request it from Apple, and just in the last week Apple granted the request.

Booch: About a year ago, I had contacted Tim and 500 of my closest friends, saying, what do we preserve? And Tim gave me some ideas. So, it's great this all came together, and this is so exciting to have.

Atkinson: That's great.

Booch: Can you jiggle the screen here? Let's take a look at MacPaint 1.something.

Hertzfeld: Do you want to demo it?

Booch: Here we have the creator application. Let's take a peak at this one here – so draw us something Bill.

Atkinson: All right.

Booch: You're the artist in the bunch. Can you point out some of the major user interface stuff that's there? In some ways it violates the basic principles of some of the Mac user interfaces, though.

Hertzfeld: Yes. The window's not growable, for example.

Atkinson: Because I had these two-and-a-half off-screen buffers, and if you could grow it, I couldn't fit them. MacPaint was a nice cross between a tool and a toy. You could draw a diagram for business use but you could also play with it. And I put in things like brush mirrors, and you could turn on eight-way symmetry, for example, and now whatever you drew would make these kaleidoscopic patterns.

Booch: How fun.

Atkinson: People just had a lot of fun just doodling with that – when you're on the phone with somebody and you're bored, you doodle. But that was strictly a fun thing.

Hertzfeld: That one's not in PhotoShop today, right?

Booch: That's right.

Hertzfeld: And you think they have so much.

Booch: Nobody's even done a plug-in on those two like that.

Atkinson: Introduction was a little cheat sheet. You could say, there's the lasso, there's the scroll, this is area fill, and this is brush.

Booch: Why this particular set of shapes? Why did you choose those? Because I remember the rounded rectangles has a story behind it, too.

Atkinson: Why these particular icons?

Booch: No, why the particular shapes you could draw?

Atkinson: Oh these, the hollow shapes – a rectangle, and constrain it to a square, and filled or not filled, so these are filled shapes, these are hollow shapes, and an oval, an arbitrary, smooth curved thing, and a polygonal thing, and then the rounded corner rectangle. Actually, when I was writing QuickDraw, the graphics primitives, I didn't think that was general enough, that rounded quarter rectangle wasn't something that should be in the graphics primitives until Steve Jobs took me for a walk and said, "Look all around you, everything you see that's a rectangle is actually a radius direction. Look right here, here this is a rounded corner, right here that's rounded, that's rounded." And I think, finally, when he got to a no- parking sign which has a rounded corner, I said, "Okay. Once he convinced me of it, I went home and that night I implemented it. It turned out that if you were to take the oval tool, and if you just at the beginning of it, instead of starting with the edges here and here and developing them, just offset them so it starts over there – oh, then that's part of a rounded rectangle – and – in the algorithm it bumps where the next edge is – when you get to the middle, you take a break. Whoops, wrong thing. It's different than Photoshop. I use Photoshop day in and day out.

Hertzfeld: Yes, I wanted to see if you remembered to double click on the eraser, but you did.

Atkinson: So just by starting later, you end up with a round rec. I was able to implement it very quickly and really at no cost.

Booch: And some amazing art came out with just this fun, basic thing. It created a whole subculture.

Atkinson: I actually have a disk – let's see if we can read one of these. We're going to experience the Mac shuffle if we do this, but let's give it a try.

Booch: The Mac shuffle?

Atkinson: There's an artist in Paris that did this stuff for me.

Booch: Let's take a look at this. The Mac Shuffle – what a great phrase.

Hertzfeld: No, you won't experience the Mac shuffle, Bill, because I copied the MacPaint to the hard disk.

Atkinson: Oh, you did? Great.

Hertzfeld: So, it's all going to be fine. You shouldn't have any …..

Atkinson: The original Mac didn't. You could only see a piece of it at a time, and you had to scroll around because we didn't have enough memory. I wanted a printable page, and so you could see a piece of it at a time and scroll through it.

Booch: Yes, I imagine the interplay between what you were doing with QuickDraw and the Image Writers came out at the same time, too. That's a drama unto itself.

Atkinson: Yes. I can't believe Command O doesn't do open.

Booch: Wow, amazing stuff. See I was just thinking, hey, use the magnifier.

Hertzfeld: Yes, I just wanted to see – you don't have Command O.

Story Behind the Command Key

Atkinson: Oh, we should tell him about Command and Feature, how Steve didn't like the abuse of the Apple logo.

Hertzfeld: There's another good story of what that funny clover symbol is.

Atkinson: On the Lisa we used the apple as the command key symbol. So it's an apple on the key and up on the menus to do a speed up of it, it would say Apple Z, and we read it as Command Z. Well, Steve didn't like misusing the Apple logo in that way, so he ordered Susan Kare to come up with a different symbol for that. And he also didn't like Command. He thought these were features – this was going to be the feature key. So Susan Kare looked at a bunch of different maps and found this Swedish map where they used that little three-way interchange symbol to represent a remarkable feature, like campsite or something like that. And so that's how it got on there. People had learned from the Lisa, the Apple II. And then they're trying to learn the Mac, well, what's this little three-way thing? So, now we've got a situation where this is the only key on the keyboard that has two different icons on it. It's got the apple and this little three-way clover leaf. Nobody knows what the three-way clover leaf is, they can't even pronounce it. With the apple, at least you could say it's an apple key. But what key? We've gone back to calling it command, because it really is distinguishing a command from typing characters.

Launch of the Macintosh

Booch: Let me get a moment in time. Here it is the launch of the Mac, Super Bowl Sunday; the seminal ads are out there. How did you guys feel when that thing was out there? Where were you when that happened?

Atkinson: On top of the world.

Hertzfeld: Yes.

Atkinson: You know, I think the biggest thing was seeing that people actually responded to it. We would want to go wherever there was a store that had one of these. We'd like to go to the store and be anonymous there and just watch people interact with it and see how they felt about it, I think.

Booch: Where were you when the ad was displayed at the Super Bowl?

Atkinson: I wasn't at the Super Bowl and I didn't watch it when it was happening. I saw it ahead of time. I think Apple showed it at maybe the shareholder meeting the first time. Or where did they show it first?

Hertzfeld: The first showing of it was in Hawaii at the sales meeting in October of '83.

Atkinson: Oh yes. I remember seeing that and just a chill going through my whole body. It was very high key; everybody was really excited and it was a powerful event.

Hertzfeld: I had seen it a few times. The most interesting thing to me was to see the reaction of other people when they saw the ad. So, Earl Smith, Barbara Koalkin and I went to the Oasis, a hamburger place with a little bar near Stanford, to watch the Super Bowl, but really to watch people watching the ad.

Atkinson: What happened? What did you see?

Hertzfeld: We saw a few people say, “Oh, what’s that?” It was kind of anti-climatic for people were more focused on the football game or filling up their beer during commercials. But there was a hush. We thought we sensed something of an impact from it.

Booch: Oh indeed, what an impact there was, what an impact!

Atkinson: Apple management didn’t want that to air. They had ordered Steve to sell the time back, and it was only because he couldn’t find a buyer that it ended up getting aired. And then it won a Cleo.

Hertzfeld: But to me the really emotional thing was at the shareholders meeting, partly because Steve was so emotional, and it was that this thing was finally out in the world. An important thing about the Mac, unlike the Lisa, was it shipped the day it was announced, too. So right after the shareholders meeting, I went to a store and bought one, because that made it really real for me, that I can just anonymously go into a store and give them my money and get a Mac.

Atkinson: Fifteen thousand copies of this book just arrived at the Oakland dock today, and I’m really looking forward to walking into Keppler’s bookstore and buying one- then it becomes real.

Booch: What a feeling, exactly.

Hertzfeld: It was euphoric, getting it out to the world, and then there’s just this gradual process, the next few months of seeing them out in the world. There was a little design place between my house and downtown Palo Alto that Earl and I would walk by every day, and about a month after the Mac shipped, they had a Mac in that office, you could see it. Every time we walked by we noticed the Mac, if they’d left it on or not.

Booch: You remember the first time it was in a movie?

Hertzfeld: Yes.

Booch: What movie was it that you first saw it in?

Hertzfeld: Oh I can’t remember.

Booch: But it was just like, wow, there it is. So, MacPaint then – it shipped with the Macintosh for multiple years, ‘til ’86, and then what happened after that?

MacPaint's Ultimate Influence

Atkinson: Apple was responding to pressure from third parties to not ship MacWrite. They owned all rights to MacWrite, and they could ship it as much as they wanted. They owned all the rights to MacPaint, and they could ship it as much as they wanted; it didn't cost them a cent. But under the pressure to not ship MacWrite, that left the Finder and MacPaint. I got a call from the people at Apple saying that they had concluded that if they shipped just MacPaint and the Finder, it would make people think their computer was a toy. They had earlier gotten comments from IBM along the lines of, "got real work to do, get a keyboard; want to play, get a mouse; do real work, get a keyboard." So they were sensitive to people perceiving the Mac as a toy. And MacPaint, well it was one of these applications that was a toy and a tool, both, and so they said, "we're not going to put it in the box anymore."

Booch: Many of the best tools are playful in many ways. And so Claris …..

Atkinson: So then, they sold it down the river to Claris, and Claris never really maintained it.

Booch: At the time, there were some other third party product; there was FullPaint and SuperPaint, and did they have a similar code base, or were those totally parallel worlds?

Hertzfeld: They were influenced by the design but they didn't …..

Atkinson: Written as applets to the code.

Booch: But MacPaint 2.0 had the story that there was a gentleman that took over the code base.

Hertzfeld: David Ramsay.

Booch: He did all sorts of interesting things with it. Now by that time you were off on HyperCard.

Atkinson: Yes.

Booch: That is another story unto itself which we'll have to take up some day. HyperCard was so cool.

Hertzfeld: In many ways Hyper Card was a successor to MacPaint because it really had MacPaint inside it to a large degree.

Atkinson: And it was also a precursor; it came six years before Mosaic, which was the first web browser, and if you look at it, it had stacks of cards, each which could have text and graphics, and buttons that were interactive, and links that would take you to another card or another stack. And it had a user readable scripting language. In many ways it was an authoring system, and Apple didn't get it. Their first marketing was, HyperCard, but what is it? I was grossed out because they were afraid to tell people. I thought it was like a software construction kit. You could put together prefab pieces that you didn't have to have a machine shop to make them, but like a tinker toy set or an erector set, and you can bolt together pieces and make little machines without having a lathe and a machine shop and all that. But they were afraid of it; they were afraid that people would be frightened they couldn't program it. One way of think of it is as an authoring system and another way is simply as a document format that went the next notch. So we had documents that had text in them, and along came MacWrite and MacPaint, and now you could have graphics in a document. So documents now had graphics and text. I wanted interaction, so that you could click on something, it would do something, so you could make an interactive document format. I wanted it to be an open platform where everybody could exchange these things, and that's why I negotiated with Apple that for two years they had to bundle it with the Mac, give it away free. So it got a chance to get established.

When they broke out the authoring part, and it had to be bought for $400 from Claris, that killed it. Because a lot of what it was, was an offering system, and people were making stacks. We used to have integer basic, and you could write a little program to do you own thing, some little calculation you needed that you're never going to get a professional programmer in to do, but you could write it in the afternoon and have what you wanted. HyperCard was the graphical user interface equivalent of that. Now, in a graphical user environment, you can make a little piece of software that does the thing that you want. Some of these HyperCard stacks only a mother could love, but their mothers loved them, because it empowered them to do something they couldn't do before. So I was really upset when they took it.

BoocH: Three more questions and we'll wrap up here: When did you both leave Apple, what did y **h:** ou do next, and what are you doing today?

Experiences Post Apple

Atkinson: Well, we left at the same time, for the same reason.

Hertzfeld: Not exactly.

Atkinson: Oh, well, you left and came back.

Hertzfeld: I never came back.

Atkinson: You didn't?

Hertzfeld: But, I helped Bill leave.

Atkinson: Well, I left with Marc Porat – and I thought I left with Andy, but I guess he'd already left – to co-found General Magic. Our notion was what we later called the Personal Communicator, a cross between a digital camera and a pager that you could take a snap and flip it over. and there's this picture; flip it over and write on the back of it, maybe say something into it, zip it off into the ether, and in your daughter's pocket, this little card would appear. We had rather naïve notions maybe about how much these things could be made to cost, and how the data would get sent, and that's part of where we failed; the infrastructure wasn't really there to get the little telecard – we called them telecards – that just dropped out of the sky.

Booch: This was in the late-'80's?

Atkinson: 1990

Hertzfeld: Yes, May 1990 General Magic was founded.

Atkinson: Yes.

Hertzfeld: Before it gets founded, there's a few month process. We started working on it in February '90. And at first we used HyperCard. We made a little mockup to send telecards where Bill wrote a HyperCard stack for the front end, and I made a little email server on the back end.

Atkinson: Yes.

Booch: And General Magic lasted from when to when?

Atkinson: Prime versions have lasted.

Hertzfeld: It lasted 10 years. It ran its course. The initial thing we tried to do was five years. Bill left in 1995, I left in 1996. It finally gave up the ghost in 2002, after …..

Atkinson: It had too much money to die. I thought you should just take the money and give it back to the investors, and say "Sorry, we failed", but no they couldn't do that.

Booch: So what are your passions today?

Atkinson: Well, I'm a full-time nature photographer, and I teach workshops in color management, digital printing, to help other photographers be able to make fine art. I recently finished making my first book of photographs using color management techniques to get better printing than has been done before. I'm looking forward – my next major project I think will be a DVD slide show of all of my nature photographs, designed with music composed for it and my hope is that I can make something heart-opening, that people will put it in a DVD player and receive some kind of nourishment from it. And maybe it'll be available at the back of a book, but I want it available not just with a book.

Booch: Marvelous. Andy, what are you up to these days?

Hertzfeld: Well, my passions, as they've been most of my life, are literature and music, and my most recent project is writing up these old Mac stories. I made a website called Folklore.org. And I have a book coming out, published by O'Reilly, which collects the stories. So right now, I'm in the editing phase. I've done all the writing, but I have to go through all the edits and improve them or react to the editor's comments such as, I don't understand what you're talking about here. And so I'm finishing that up. After General Magic, I became really, really interested in free software, in 1998, seeing it. I've been disappointed- with the Apple II, and then especially with the Macintosh, the personal computing industry seemed to be such an empowering thing, and with the Mac we were making computers easy to use. But I think it ran off the rails at some point in the '90's, with Bill Gates and Microsoft kind of dominating it in an unhealthy way, and somehow things stagnated, and it's due to a structural problem in the industry. I thought there was no solution to it. I became aware that things were bad, and I didn't see any way out. But, then in 1998 with a Mozilla announcement, free software attracted my attention, and when I studied it, I realized that free software held out the hope for a thriving, wonderful software industry, and I decided to dedicate the rest of my career to helping make that happen. I started a company called Eazel. We ran out of money though, when the bust happened, and so that didn't happen. But I'm still going to be working on making free software take over, if I can.

Booch: Very cool. You said a phrase when you were speaking, of folklore.org, is it?

\Hertzfeld: Yes.

Booch: Trying to document the stories. Because for me, the technology is very cool, and there's beauty in the software, but software is ultimately a human experience. And there's this incredible drama that goes into it, and we've seen some of that drama today. So you guys are my heroes, absolutely. Looking back from your experience, what would you say to the software developer today, what kind of wisdom can you offer that person in terms of what should they aspire to, in today's current state of software development?

Advice to Today's Developers

Hertzfeld: Follow your heart is the first thing that pops into my head. You have to do work each day that you believe in.

Atkinson: If you want to get it smooth, you've got to rewrite it from scratch at least five times. And have a lot of user testing. Because you can't see the things that you can't see. So you have to test it on people who are novices and people who don't know what this thing is, and they'll show you. And if you're willing to be humble and say, okay, they don't get this, and now what am I going to do about it? And not try to ship that first prototype, hold off, and let it incubate in privacy so you're not forced to ship. Don't tell the marketing people you're done with it, when you've got the first fifth of it done. But really, the thing that makes software smooth and useable is user testing, user testing and user testing. So get in a development environment where you can prototype quickly, string unveiling where it doesn't have to be robust, but it has to be something you can evolve it very quickly, and try real user interfaces and change it quickly.

Hertzfeld: I just want to say, pick things to work on that you really, really want to use yourself. Because then you can close that loop with the user testing. If you're one of the users, you can just iterate in your head. So it's really, really important. Of all the things you can work on, work on the thing that isn't in the world, that you want to make in the world. And then, you can be both the user and the creator, and I think there's real power in that. To some extent, that's the secret of the Mac's success itself. We all wanted the Mac more than anyone else, so much so, we had to make it.

Booch: So what makes for a great software developer? How does one become one?

Atkinson: I don't know.

Hertzfeld: There's probably lots of different ways, but I would say the most important thing is passion, will, tenacity. There's the intellectual side of it, and then there's the personality, emotional side of it, and I think you really need both. Actually, I was just reading Paul Graham's book *Hackers and Painters*, and he has a set of essays about this exact question: How do you become a great programmer, how do you make it? He has an essay called "Feast for Bakers," and he said, "The recipe for success is to have really exacting taste and the talent to satisfy it". I thought that was a good quote.

Booch: The last question then. Here we've covered a 20-plus year span. Predicting the future has always been impossible, but what's your hope, and vision for what computers- and what's it like at 20?

Vision of Future Computing

Atkinson: The first thing is, get rid of the keyboard and the mouse, and replace them with a smooth plate called a touch plate; it's responsive to gentle, stroking touch. We can enter characters by touching on a character and stroking through others around it, and we can enter whole- instead of typing i-n-g, we're typing "ing" – we're entering text at a higher level. Shift key and the whole thing becomes a touching-pointing device. Forget this teeny little scratch pad on your portable. So, one is the touch plate. Another avenue is that as computers get smaller and ubiquitous, working with computers has to enter into a dialogue, not this thing where you're commanding the computer, but where actually you're having a conversation. The computers have to get to where they actually understand the flow of a conversation. It's not just a speech-recognition thing. It's a language understanding problem. And when it does, we can have, like in the Ender series, a little crystal in our ear that's listening to everything we're listening to and whispering little things in our ear, and it's connected into the wireless net.

Booch: You're referring to *Ender's Game*?

Atkinson: Yes, *Ender's Game* and the whole series of Orson Scott Card books. I think the smaller we go, the more we need to rely on audible interfaces. And I think that programming needs to be a conversation of what it is you're looking for. You say, well I want to do this, and then that's ambiguous, but whatever you're talking about, this other entity is asking you some questions about it, and you refine it.

Booch: Very cool. Andy, what are your thoughts?

Hertzfeld: Oh, a lot of what Bill said. I think that clearly the next really big frontier in user interface is going to be the language understanding. That really is as much of a leap as the graphic user interface was. There are lots of problems to be solved but it's pretty fair, as Bill said, for the mobile application, the keyboard. I've been experimenting with little ideas in that space myself. The voice recognition software is really good today, but it doesn't make sense to use it at your desktop because you can still do it better with the keyboard and the mouse. But suddenly, when you're standing up, the keyboard and mouse are useless. Of course, computers are going to be everywhere. You're going to need to have the computer on your person. So, the speech recognition I think is important. But I would say the next really important thing is getting the software industry on a level playing field, a place that's really open to innovation. I think the way that's going to happen is if the shared infrastructure becomes available and is owned by the community of its users and developers instead of a psychotically driven business force. And that's going to happen, and it's going to happen over the next year. So, it's not the far years. I think once we have a level playing field open to innovation, we can start to really explore the possibilities better. And then I've been thinking about agent-type user interfaces. Bill's into that too. But the graphic user interface, there's always direct manipulation.

If you’re steering it at the steering wheel, you can only do things while you’re at the steering wheel. Eventually we’re going to want to set up policies where the tireless computer can execute our policies for us, continuously, especially with the network.

Atkinson: Yes, drive me to work and let me know when we get there.

Hertzfeld: I think there’s a very fertile area right now in exploring and getting it right – like Apple more or less got right with the Mac – getting the agency-user interfaces right. Things that can happen while you’re not directly controlling it will be a frontier in the next few years.

Booch: Bjarne Stroutrup – he’s the inventor of C++ – has often said that our civilization runs on software and, increasingly, software is moving itself into the interstitial spaces of our society. I call you my heroes, and you’re heroes to many because you guys have created something that has helped that movement to the interstitial spaces. I just want to thank you for the gift to the museum. This is great stuff. I hope that I’m actually addressing generations dozens of years from now, who are looking at this stuff and saying, “Wow, that’s pretty cool.” Your passion very much comes through, and I appreciate that. Thanks so much for being here.

Hertzfeld: Thanks for having us.

Atkinson: Thank you..