



Anecdotes about the development of
Apple's original Macintosh computer
and
the people who created it



Source

<http://www.folklore.org/ProjectView.py?project=Macintosh&detail=medium>
28 January 2004

Misc.

David T Craig added a few rather rare extras that should prove of some interest to those desiring more detailed information about Macintosh's history and technology.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION





APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



TABLE OF CONTENTS

Nybbles

Author: Andy Hertzfeld
Date: February 1981
Characters: Andy Hertzfeld, Steve Wozniak, Burrell Smith, Brian Howard
Topics: Hardware Design, Prototypes, Software Design, Early Programs,
Technical
Summary: My first job on the Macintosh project

Early Demos

Author: Andy Hertzfeld
Date: April 1981
Characters: Andy Hertzfeld, Bud Tribble, Rich Page, Bob Bishop,
Bill Atkinson, Bruce Horn
Topics: Hardware Design, Prototypes, Software Design, Early Programs
Summary: Various demos showed off what the hardware could do

Round Rects Are Everywhere!

Author: Andy Hertzfeld
Date: May 1981
Characters: Steve Jobs, Bill Atkinson
Topics: Software Design, Inspiration, QuickDraw
Summary: Steve inspires Bill by pointing out something about the real world

Apple II Mouse Card

Author: Andy Hertzfeld
Date: June 1981
Characters: Andy Hertzfeld, Burrell Smith, Dan Kottke, Bill Budge,
Steve Jobs
Topics: Software Design, Apple Spirit, Prototypes, Technical
Summary: We hooked up a mouse to the Apple II

Hungarian

Author: Andy Hertzfeld
Date: January 1982
Characters: Bud Tribble, Tom Malloy, Charles Simonyi, Bill Atkinson
Topics: Software Design, Technical, Lisa
Summary: A curious style of programming

-2000 Lines Of Code

Author: Andy Hertzfeld
Date: February 1982
Characters: Bill Atkinson
Topics: Software Design, Management, Lisa
Summary: It's hard to measure progress by lines of code



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Mi ster Maci ntosh

Author: Andy Hertzfel d
Date: February 1982
Characters: Steve Jobs, Andy Hertzfel d, Susan Kare, Fol on
Topi cs: Software Desi gn, Personal ity
Summary: Steve has a uni que idea for the software

Cal cul ator Constructi on Set

Author: Andy Hertzfel d
Date: February 1982
Characters: Chris Espinosa, Steve Jobs, Donn Denman
Topi cs: Software Desi gn
Summary: Chris tries to make a Steve-approved cal cul ator

Rosi ng' s Rascal s

Author: Andy Hertzfel d
Date: March 1982
Characters: Bill Atki nson, Wayne Rosi ng, Dan Smi th, Frank Ludol ph,
Steve Jobs, Bruce Horn
Topi cs: Software Desi gn, Li sa, User Interface, Management
Summary: The Li sa Filer was radi cally redesigned with no time to spare

I Still Remember Regi ons

Author: Andy Hertzfel d
Date: April 1982
Characters: Bill Atki nson, Steve Jobs
Topi cs: Software Desi gn, Qui ckDraw
Summary: We almost lose Bill in a car acci dent

And Then He Di scovered Loops!

Author: Andy Hertzfel d
Date: April 1982
Characters: Bob Belle ville, Ri ch Wi lli ams
Topi cs: Software Desi gn, Management
Summary: Bob has written many lines of code

Do It

Author: Andy Hertzfel d
Date: June 1982
Characters: Larry Tesler
Topi cs: User Interface, Software Desi gn, Li sa
Summary: User testing sometimes has surpri sing resul ts



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Resource Manager Countdown

Author: Andy Hertzfeld
Date: August 1982
Characters: Bruce Horn, Bob Belleville
Topics: Software Design, Management
Summary: Bob gives Bruce thirty days to finish the Resource Manager

Sound By Monday

Author: Andy Hertzfeld
Date: September 1982
Characters: Burrell Smith, Andy Hertzfeld, Steve Jobs
Topics: Hardware Design, Software Design, Management, Technical
Summary: Steve threatens to remove Mac's sound capability

Boot Beep

Author: Andy Hertzfeld
Date: September 1982
Characters: Charlie Kellner, Andy Hertzfeld, Terry Oyama, Steve Jobs
Topics: Software Design
Summary: How the boot sound evolved

Bouncing Peeps

Author: Andy Hertzfeld
Date: March 1983
Characters: Andy Hertzfeld, Steve Jobs, John Sculley, Susan Kare, Bill Atkinson
Topics: Software Design, recruiting, Early Programs
Summary: We cook up a special demo for John Sculley

Saving Lives

Author: Andy Hertzfeld
Date: August 1983
Characters: Steve Jobs, Larry Kenyon
Topics: Software Design, Inspiration
Summary: Steve wants us to make the Macintosh boot faster

World Class Cities

Author: Susan Kare
Date: August 1983
Characters: Steve Jobs, Bill Atkinson, Andy Hertzfeld, Susan Kare
Topics: Software Design, Personality
Summary: We needed to come up with a naming scheme for our fonts



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Swedish Campground

Author: Andy Hertzfeld
Date: August 1983
Characters: Susan Kare, Steve Jobs
Topics: Software Design
Summary: How we got the unusual symbol used for the menu command key

The Puzzle

Author: Andy Hertzfeld
Date: September 1983
Characters: Andy Hertzfeld, Jerome Coonen
Topics: Software Design, Management
Summary: The puzzle desk accessory became controversial

Cut, Paste and Crash

Author: Andy Hertzfeld
Date: September 1983
Characters: Larry Kenyon, Susan Kare, Andy Hertzfeld
Topics: Technical, Software Design, Bugs
Summary: We uncover a particularly bad bug shortly after ROM freeze

We're Not Hackers!

Author: Andy Hertzfeld
Date: September 1983
Characters: Larry Kenyon, Andy Hertzfeld, Bill Atkinson, Burrell Smith, Steve Capps
Topics: Software Design, Technical, Personality
Summary: We were always dealing with memory limitations

Monkey Lives

Author: Andy Hertzfeld
Date: October 1983
Characters: Steve Capps, Bill Atkinson
Topics: Software Design, Testing, Technical
Summary: The very first location in low memory

Stolen From Apple

Author: Andy Hertzfeld
Date: October 1983
Characters: Steve Jobs, Susan Kare, Steve Capps, Andy Hertzfeld
Topics: Software Design
Summary: We put a hidden icon in the ROM



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



The Mythical Man Year

Author: Andy Hertzfeld
Date: October 1983
Characters: Steve Jobs, Bill Atkinson
Topics: Software Design, Personality, Reality Distortion, QuickDraw
Summary: Steve estimates the effort that went into Quickdraw

Disk Swapper's Elbow

Author: Steve Capps
Date: January 1984
Characters: Steve Capps
Topics: Software Design, Technical, Bugs
Summary: a last minute bug causes some problems

It Sure Is Great To Get Out Of That Bag!

Author: Andy Hertzfeld
Date: January 1984
Characters: Steve Jobs, Steve Capps, Bruce Horn, Andy Hertzfeld, Susan Kare, Mike Boich, Mark Barton, Steve Hayden
Topics: Marketing, The Launch, Software Design
Summary: We need a demo for the intro

Real Artists Ship

Author: Andy Hertzfeld
Date: January 1984
Characters: Andy Hertzfeld, Bill Atkinson, Steve Capps, Jerome Coonen, Bob Belleville, Steve Jobs, Donn Denman, Randy Wigginton
Topics: Management, Software Design
Summary: The final push to finish the software

MacBasic

Author: Andy Hertzfeld
Date: June 1985
Characters: Donn Denman, Bryan Stearns, Jerome Coonen, Randy Wigginton, John Sculley, Bill Gates, Bill Atkinson, Steve Wozniak
Topics: Software Design, Management
Summary: The sad story of MacBasic

Desk Ornaments

Author: Andy Hertzfeld
Date: undated
Characters: Bud Tribble, Chris Espinosa, Donn Denman, Larry Kenyon, Steve Capps, Bill Atkinson, Steve Jobs, Susan Kare
Topics: Software Design, User Interface, Personality
Summary: A brief history of desk accessories



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



On Xerox, Apple and Progress

Author: Bruce Horn
Date: undated
Characters: Bruce Horn, Bill Atkinson, Larry Tesler, Steve Jobs, Jerry Manock
Topics: Credit, Software Design, Origins
Summary: An essay discussing where the Mac user interface came from

It's The Moustache That Matters

Author: Andy Hertzfeld
Date: September 1980
Characters: Burrell Smith, Tom Whitney
Topics: Management, Personality
Summary: Burrell wants to get promoted to engineer

Good Earth

Author: Andy Hertzfeld
Date: October 1980
Characters: Bud Tribble, Steve Jobs, Jef Raskin, Bill Atkinson, Rod Holt, Burrell Smith, Joanna Hoffman, Brian Howard, Steve Wozniak, Mike Scott
Topics: Apple Spirit, Origins, Management, Buildings
Summary: The original Mac team's original office

Black Wednesday

Author: Andy Hertzfeld
Date: February 1981
Characters: Andy Hertzfeld, Mike Scott, Steve Jobs, Dick Huston, Donn Denman, Rick Aurricchio, Jef Raskin, Burrell Smith, Brian Howard
Topics: Origins, Apple II, Management
Summary: A shakeup in Apple II engineering frees me up to work on the Macintosh

Reality Distortion Field

Author: Andy Hertzfeld
Date: February 1981
Characters: Steve Jobs, Bud Tribble
Topics: Management, Personality
Summary: Bud defines Steve's unique talent

Square Dots

Author: Andy Hertzfeld
Date: April 1981
Characters: Bill Atkinson, Burrell Smith, Rich Page, Wayne Rosing, Tom Malloy, Bruce Daniels, Paul Baker
Topics: Hardware Design, Management, Lisa
Summary: The Lisa had a different screen resolution than the Macintosh



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Bicycle

Author: Andy Hertzfeld
Date: April 1981
Characters: Rod Holt, Steve Jobs, Mac Team, Jef Raskin
Topics: Management, Apple Spirit
Summary: Rod wants to change the name of the project

Diagnostic Port

Author: Andy Hertzfeld
Date: July 1981
Characters: Steve Jobs, Jef Raskin, Burrell Smith, Brian Howard, Steve Wozniak, Rod Holt
Topics: Hardware Design, Management
Summary: Burrell tried to sneak in some hardware expandability

Signing Party

Author: Andy Hertzfeld
Date: February 1982
Characters: Steve Jobs, Bob Belleville, Jerry Manock, Mac Team, Bud Tribble, Jef Raskin, Mike Boich, Rod Holt, Steve Wozniak
Topics: Management, Apple Spirit
Summary: The artists sign their work

And Another Thing...

Author: Andy Hertzfeld
Date: March 1982
Characters: Rich Page, Steve Jobs, Bill Atkinson, Burrell Smith, Andy Hertzfeld, Larry Tesler
Topics: Lisa, Management, Personality, Personality Clashes
Summary: There was some friction between the Mac and Lisa teams

I'll Be Your Best Friend

Author: Andy Hertzfeld
Date: August 1979
Characters: Burrell Smith, Andy Hertzfeld, Wendell Sander, Steve Wozniak
Topics: Origins, Personality, Hardware Design, Apple II
Summary: Burrell Smith was creative in more than just engineering

We'll See About That

Author: Andy Hertzfeld
Date: November 1979
Characters: Burrell Smith, Bill Atkinson, Jef Raskin, Steve Wozniak
Topics: Hardware Design, Technical, Apple II
Summary: Burrell proves his mettle with the 80k language card.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Scrooge McDuck

Author: Andy Hertzfeld
Date: February 1980
Characters: Burrell Smith, Andy Hertzfeld, Jef Raskin, Tom Whitney,
Bob Bishop, Cliff Huston, Dick Huston
Topics: Origins, Hardware Design, Prototypes
Summary: The very first image on the very first Macintosh

I Invented Burrell

Author: Andy Hertzfeld
Date: 1981
Characters: Burrell Smith, Jef Raskin
Topics: Personality
Summary: Burrell imitates Jef

Texaco Towers

Author: Andy Hertzfeld
Date: January 1981
Characters: Steve Jobs, Dan Kottke, Brian Howard, Bud Tribble,
Jef Raskin, Burrell Smith, George Crow
Topics: Origins, Lisa, Buildings
Summary: The office where the Mac became real

A Message For Adam

Author: Andy Hertzfeld
Date: April 1981
Characters: Adam Osborne, Steve Jobs, Jef Raskin
Summary: We encounter Adam Osborne at the West Coast Computer Faire

Pineapple Pizza

Author: Andy Hertzfeld
Date: May 1981
Characters: Burrell Smith, Brian Howard, Dan Kottke, Steve Jobs, Bud Tribble
Summary: We stay late to bring up the first printed circuit board

PC Board Esthetics

Author: Andy Hertzfeld
Date: July 1981
Characters: Steve Jobs, Burrell Smith, George Crow, Brian Howard, Dan Kottke
Topics: Hardware Design, Personality
Summary: Steve is concerned with the esthetics of the PC board



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Shut Up!

Author: Andy Hertzfeld
Date: July 1981
Characters: Andy Hertzfeld, Bud Tribble, Steve Jobs, Bill Gates, Charles Simonyi
Topics: 3rd party developers, Microsoft, Prototypes
Summary: The first time we demoed the Macintosh to Microsoft

Donkey

Author: Andy Hertzfeld
Date: August 1981
Characters: Bill Gates, Neil Konzen, Steve Jobs, Charles Simonyi
Topics: Microsoft
Summary: We discover who wrote an awkward game for the PC

Gobble, Gobble, Gobble

Author: Andy Hertzfeld
Date: March 1982
Characters: Steve Jobs, Andy Hertzfeld, Burrell Smith, Angeline Lo
Topics: Management, Personality
Summary: We interview candidates for software manager

Software Wizard

Author: Andy Hertzfeld
Date: March 1982
Characters: Andy Hertzfeld, Burrell Smith, Steve Jobs
Topics: Personality, Apple Spirit
Summary: The Mac group gets business cards

Creative Think

Author: Andy Hertzfeld
Date: July 1982
Characters: Alan Kay, Bill Budge
Topics: Inspiration
Summary: A seminar with Alan Kay

You Guys Are In Big Trouble

Author: Andy Hertzfeld
Date: August 1982
Characters: Steve Jobs, Bill Atkinson, Bud Tribble, Andy Hertzfeld, Bruce Horn
Topics: Apple Spirit, Buildings, Management
Summary: We receive Steve's permission to dismantle the onerous burglar alarm



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Too Big For My Britches

Author: Andy Hertzfeld
Date: February 1983
Characters: Bob Belleville, Steve Jobs, Jerome Coonen, Andy Hertzfeld, Steve Capps, Larry Kenyon, Pat Sharp, Rod Holt
Topics: Management, Personality Clashes
Summary: My belated performance review is delivered verbally

Quick, Hide In This Closet!

Author: Andy Hertzfeld
Date: August 1983
Characters: George Crow, Bob Belleville, Larry Kenyon, Steve Jobs, Rod Holt, Rich Williams, Hide Kamoto
Topics: Hardware Design, Management, Lisa
Summary: Steve forbid us to work with Sony

Steve Wozniak University

Author: Andy Hertzfeld
Date: September 1983
Characters: Burrell Smith, Steve Wozniak
Topics: The Launch, Personality
Summary: Burrell's educational credentials

Make a Mess, Clean it Up!

Author: Donn Denman
Date: September 1983
Characters: Burrell Smith, Donn Denman, Andy Hertzfeld
Topics: Personality, Recreation
Summary: Burrell had a unique approach to playing Defender

90 Hours A Week And Loving It

Author: Andy Hertzfeld
Date: October 1983
Characters: Burrell Smith, Debi Coleman, Steve Jobs
Topics: T-Shirts, Personality
Status: Draft
Summary: Burrell modifies his sweatshirt

A Rich Neighbor Named Xerox

Author: Andy Hertzfeld
Date: November 1983
Characters: Steve Jobs, Neil Konzen, Mike Boich, Bill Gates
Topics: 3rd party developers, Microsoft, Personality
Summary: Steve confronts Bill Gates about copying the Mac



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Steve Capps Day

Author: Andy Hertzfeld
Date: December 1983
Characters: Jerome Coonen, Steve Capps, Bruce Horn, Larry Kenyon
Topics: Personality, Apple Spirit
Summary: We all dress up in honor of Steve Capps

Where'd That Hand Come From?

Author: Eric Barnes
Date: April 1984
Topics: Marketing
Summary: How the hand and the mouse came together!

Handi capped

Author: Andy Hertzfeld
Date: 1985
Characters: Steve Jobs, Jean-Louis Gasse
Topics: Personality Clashes, Personality
Summary: The reason that Steve parks in the handi capped space

Are You Gonna Do It?

Author: Andy Hertzfeld
Date: February 1985
Characters: Burrell Smith, Steve Jobs
Topics: Management, Quitting
Summary: Burrell quits Apple, but fails to live up to his threat

Evolution Of A Classic

Author: David Ramsey
Date: March 1985
Characters: Bill Atkinson, Gene Pope, David Ramsey, Rich Page
Topics: Maintenance, MacPaint
Summary: MacPaint 2.0

Five Different Macintoshes

Author: Andy Hertzfeld
Date: undated
Characters: Burrell Smith, Bud Tribble, Wendell Sander, Steve Jobs, Jef Raskin, Bill Atkinson
Topics: Hardware Design, Technical
Summary: Burrell actually designed 5 different Macintoshes



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



The Apple Spirit

Author: Andy Hertzfeld
Date: undated
Characters: Andy Hertzfeld
Topics: Apple Spirit, Inspiration
Summary: An essay I wrote about Apple's values that was published in MacWeek

Busy Being Born

Author: Andy Hertzfeld
Date: undated
Characters: Bill Atkinson, Bud Tribble, Steve Jobs
Topics: Origins, User Interface
Summary: A visual history of the development of the Lisa/Macintosh user interface

Origins of Spline-Based and Anti-Aliased Fonts

Author: William Donelson
Date: January 1989
Characters: William Donelson, John Warnock, Bob Sproull
Topics: Origins, Software Design
Summary: Creation of the first spline-based font system

The Father Of The Macintosh

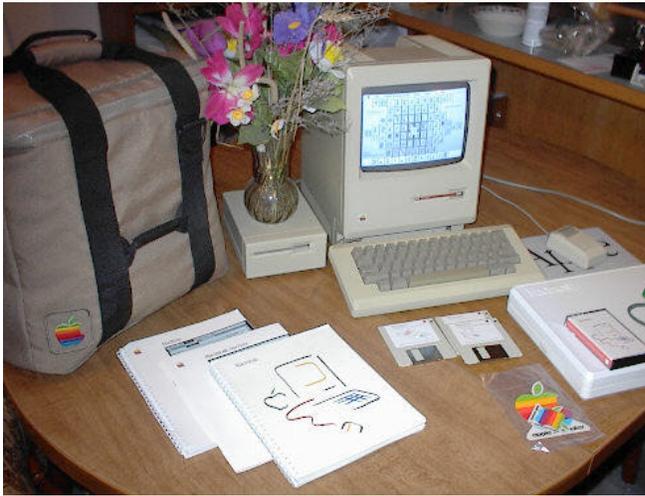
Author: Andy Hertzfeld
Date: undated
Characters: Bud Tribble, Steve Jobs, Joanna Hoffman, Jef Raskin, Rod Holt, Burrell Smith, Brian Howard, Steve Wozniak, Jerry Manock, Bill Atkinson
Topics: Credit, Personality
Summary: Who is the father of the Macintosh?

An Early 'Switch' Campaign

Author: Tom Zito
Date: September 1985
Characters: Bo Derek, Steve Jobs
Topics: Personality
Summary: Steve Jobs tries to get Bo Derek to abandon her PC in favor of a Mac



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION





APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Introduced:	Announced January 1984
Original Price:	\$2500
CPU:	Motorola 68000, 8 MHz
Memory:	128K
Operating System:	MacOS 1.0
Input/Output:	Built-in 400K single-sided floppy drive, monochrome monitor, mouse port, serial port
Bus:	N/A
Other Items in Collection:	Proper mouse & keyboard, Documentation, MacOS System 1.0, misc. software, Microsoft MacEnhancer (I/O expander), original logo carrying case
Items Needed:	External 400K drive

Originally called just "Macintosh", and later referred to as the Mac 128K, this is the original Macintosh computer. It was announced with the famous one-time-only Orwellian TV commercial during the 1984 Super Bowl broadcast.

The Mac 128 owes its existence to the Apple Lisa, which pioneered most of its graphical user interface features and other design touches (most of which were developed by Xerox PARC on a machine called the Star ***).

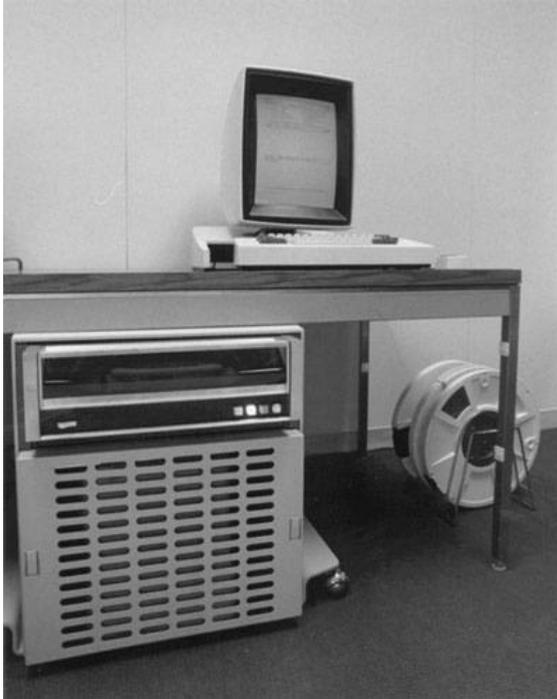
[*** Note: What about the Xerox Alto computer which pre-dated the Star by a few years and which was more of a pioneering machine in terms of new technologies than the Star ? Alto's Smalltalk programming environment which used overlapping windows and the mouse provided the main influence for non-Xerox computers such as the Apple Lisa and Macintosh.]



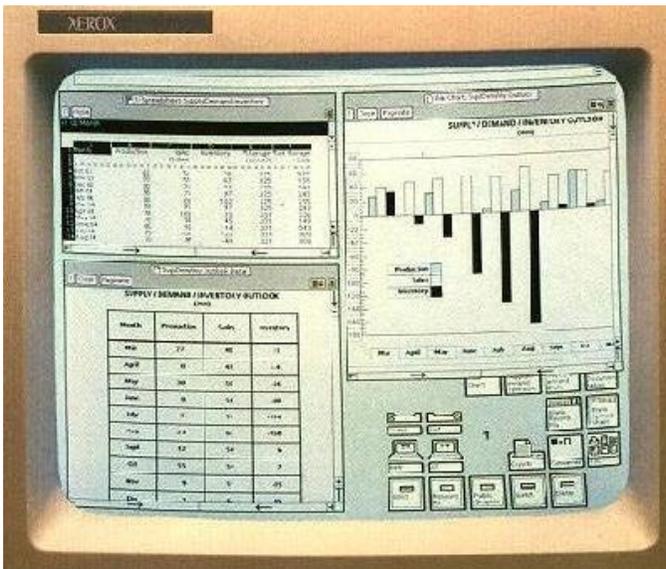
APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



[Xerox Alto computer]



[Xerox Star computer screen]

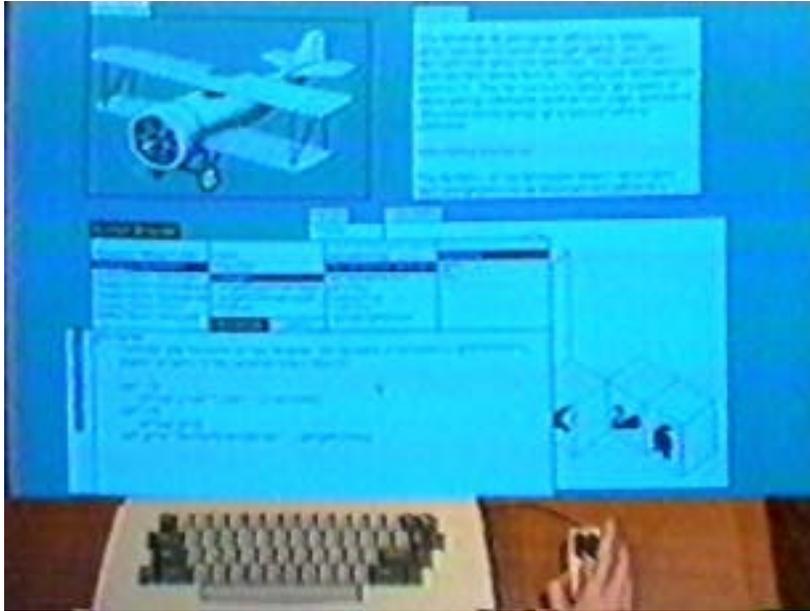




APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



[Xerox Smalltalk screen]



[Apple / Xerox / Microsoft

Date: Monday, January 22, 2001 11:53 AM
From: Frank Ludolph
Subject: Minor corrections to GUI history

As a person who moved from Xerox to Apple and worked on the Lisa Filer (desktop), I much appreciate this site.

The Alto was not the size of a large desk. It was about 2 feet on a side and slid under the standard SteelCase tables of the time.

I would expect the complications in the Apple/ Microsoft lawsuit were due to the Lisa/Apple UI being similar to the Xerox Star UI, not the Alto's. The Alto did use a mouse and bitmapped display, and SmallTalk did have overlapping windows, though most applications ran full screen. The Star's UI seemed more similar.

I was able to demonstrate the Lisa next to the Star at the CHI conference in Los Angeles a few years ago. Great fun! I have a video tape of the demo.

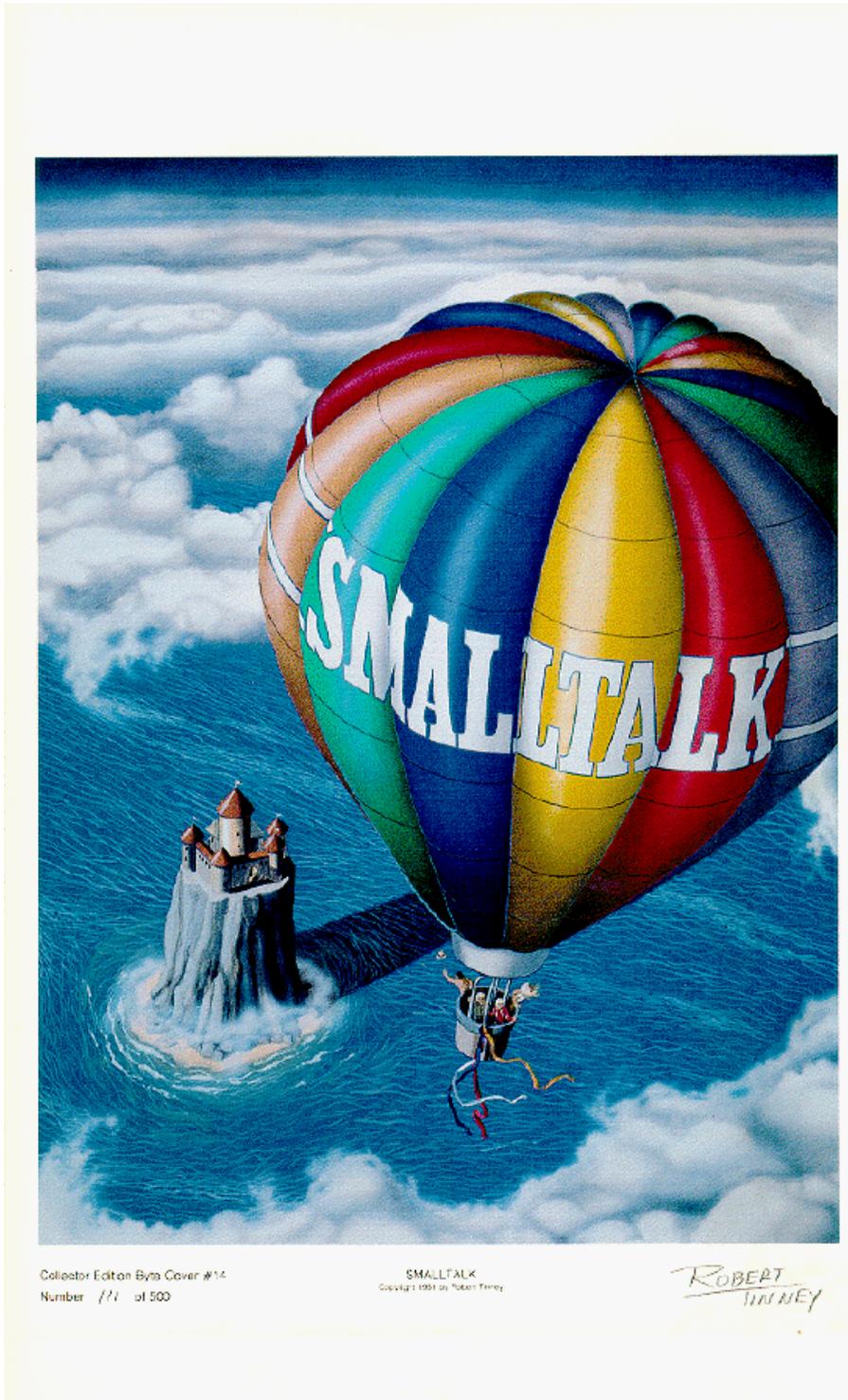
Frank Ludolph]



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



[Xerox Smalltalk BYTE cover (I really like this)]





APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



AUTHORS

Andy Hertzfeld



Andy started at Apple in August 1979 as Apple employee #435. He was one of the main authors of the Macintosh system software working on the core operating system and the User Interface toolbox, as well as most of the original desk accessories.

He later went on to co-found three companies: Radius (1986), General Magic (1990) and Eazel (1999).

He is also the main author of the Folklore website you're currently reading.



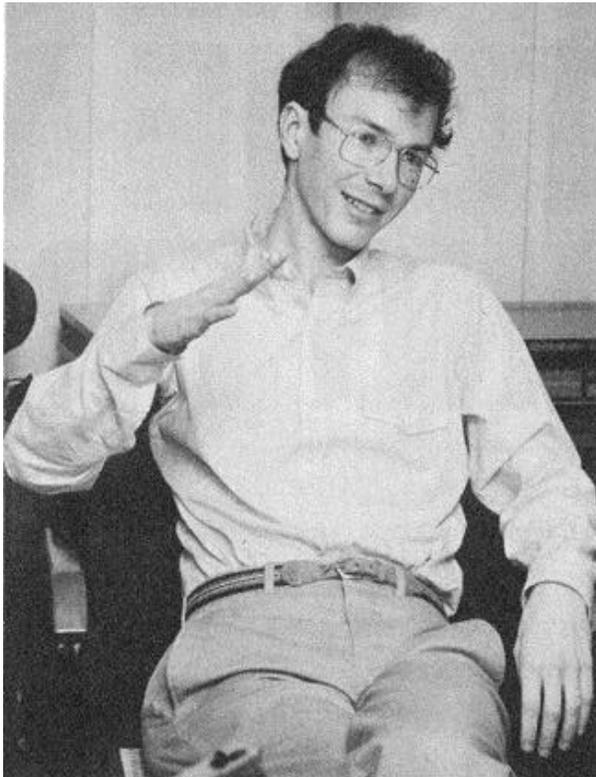
APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Burrell Smith



Bud Tribble

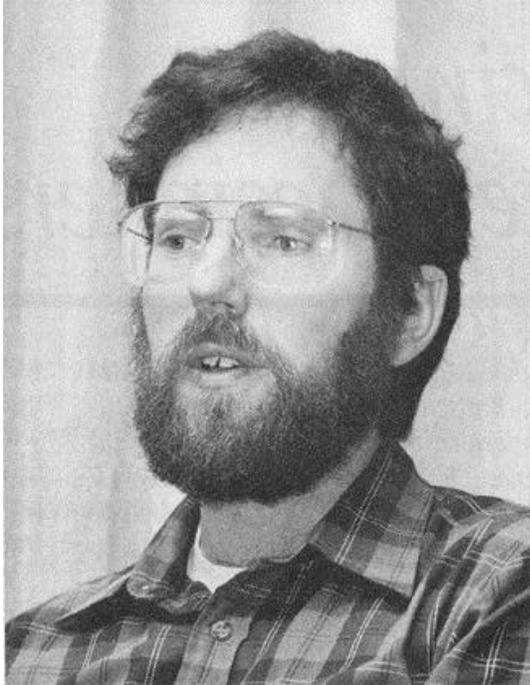




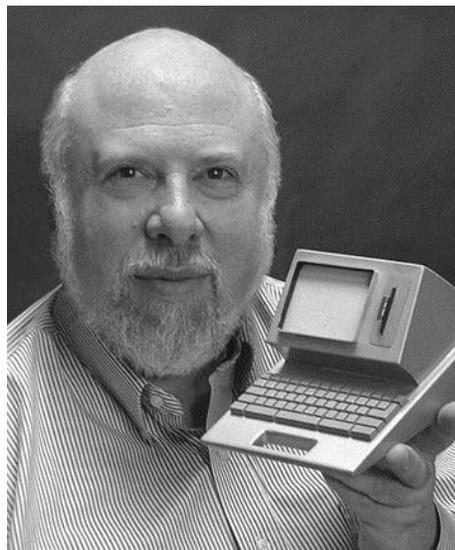
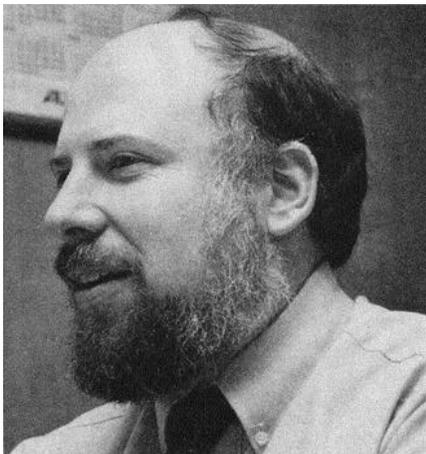
APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Brian Howard



Jef Raskin



Creator of Apple's Macintosh, the Canon Cat, click-and-drag selection and other inventions. He coined the term and the concept of "information appliances." Author of *The Humane Interface* (Addison-Wesley 2000).



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Donn Denman



MacBASIC, mi sc. software tasks.

Steve Capps



Macintosh Finder, MacMaze, Alice 3D chess, mi sc. software tasks.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION

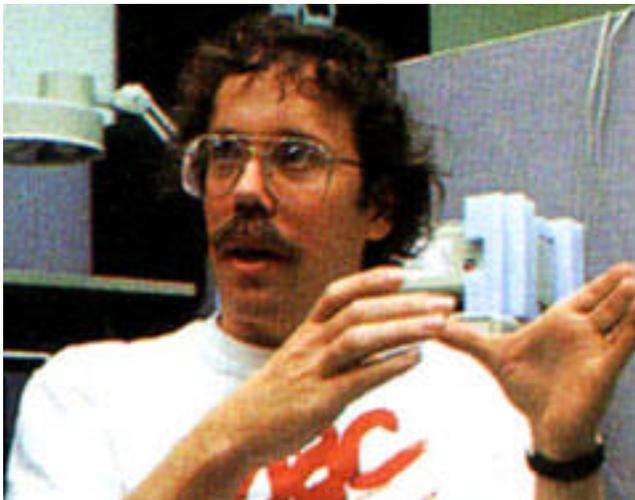


Susan Kare



Susan started working on the Mac Team in January 1983. She designed most of the icons and fonts for the Macintosh, as well as lots of the original marketing material, and helped to craft the overall look and personality of the system. After leaving Apple in 1985, she was one of the first ten NeXT employees. Since 1988, she has been a successful independent graphic designer.

Bill Atkinson



Bill's work on QuickDraw was the foundation of both the Lisa and the Macintosh, and he was really the principal designer of the Macintosh UI. Later, he single-handedly wrote MacPaint, the initial killer app for the Macintosh, followed by Hypercard. He co-founded General Magic in 1990. Since 1996, he's been a full-time nature photographer.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Steve Jobs



[Steve Jobs (left) and Steve Wozniak (right) displaying the Apple //c computer (top, white) above an Apple I computer (in wooden case). Circa 1984.]



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



[This is what Steve had to say about his visit to Xerox PARC (Palo Alto Reserach Center) in 1979:

"And they showed me really three things.

But I was so blinded by the first one I didn't even really see the other two. One of the things they showed me was object orienting programming they showed me that but I didn't even see that. The other one they showed me was a networked computer system...they had over a hundred Alto computers all networked using email etc., etc., I didn't even see that. I was so blinded by the first thing they showed me which was the graphical user interface. I thought it was the best thing I'd ever seen in my life. Now remember it was very flawed, what we saw was incomplete, they'd done a bunch of things wrong. But we didn't know that at the time but still though they had the germ of the idea was there and they'd done it very well and within you know ten minutes it was obvious to me that all computers would work like this some day. "]

[It was a turning-point. Jobs decided that this was the way forward for Apple.

In 1997 the New York Times reported:

"Steve Jobs told his audience that during his legendary visit to Xerox's Palo Alto Research Center in the late 1970's, where he first saw the technology that he commercialized in the Macintosh, he had missed some fundamental components. Mr. Jobs said he had been "blinded" by Xerox's graphical user interface and had overlooked both the company's Smalltalk programming language and its powerful office network :

"If I'd only stayed another 20 minutes," Mr. Jobs said today"

(NewYork Times 1/8/97)]



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Bruce Horn

I was fortunate enough to be a member of the Xerox Learning Research Group during the development of the SmalltalkLanguage, and was involved in several implementations including the Dorado, the NoteTaker?, and a Norwegian system called the Mycron-2000. I still wish I could be programming in Smalltalk, but for now Objective-C will have to do :-). I also was one of the original designers of the Apple Macintosh, responsible for the Finder, the Resource Manager, and other parts of the Mac OS. A few years at CarnegieMellonUniversity resulted in a CS Ph.D. and the opportunity to meet a lot of tremendous people.

Sometimes, I feel like I know or have met everybody in CS, thanks to my tenures at XeroxParc, CMU, and AppleComputer. Like the degrees of separation from Kevin Bacon. If I don't know X, I know someone who does :-).

Any number of people will try to tell you about the origins of the Macintosh, but Bruce Horn was one of the people who made it happen.

From 1973 to 1981, Bruce was a student in the Learning Research Group at Xerox, where Smalltalk, an interactive, object-oriented programming language, was developed. While there, he worked on various projects including the NoteTaker, a portable Smalltalk machine, and wrote the initial Dorado Smalltalk microcode for Smalltalk-76. At the Central Institute for Industrial Research in Oslo, Norway, in 1980, he ported Smalltalk-78 to an 8086 machine, the Mycron-2000.

At Apple (1981-1984), Bruce's contributions included the design and implementation of the Resource Manager, the Dialog Manager and the Finder (with implementation help from Steve Capps). He was also responsible for the type Framework for documents, applications, and clipboard data, and a number of system-level design decisions.

Since then, Bruce consulted on a variety of projects in the late 1980's at Apple and received a Ph.D. in Computer Science from Carnegie-Mellon University in 1993. He continues to work as a computer science consultant with Apple and other companies.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



30 Aug 83

82/BLANK:FMAIN.TEXT

Page 001

```

{
cfmain
obj:tmp
gobj:tmp
obj:fmain
{ }
UNIT FMain;
{
    The Finder--system shell program.

    Written by Bruce Horn           10 May 1983
    Copyright 1983 by Apple Computer. All Rights Reserved.

    This is the main section of the Finder. It calls procedures which
    are defined in FUtil and FObj's only. All routines are written
    in assembly and can be found in FUtil68K.

    Last modified 24 December 1983 by Bruce.
}

INTERFACE

USES { $U obj:HeapZone      } HeapZone,
      { $U obj:QuickDraw   } QuickDraw,
      { $U obj:MacFMgr     } Fonts,
      { $U obj:GrafUtil    } GrafUtil,
      { $U obj:TextEdit    } TextEdit,
      { $U obj:MacEMgr     } Events,
      { $U obj:MacWMgr     } Windows,
      { $U obj:MacIMgr     } Icons,
      { $U obj:MacDskMgr   } DeskMgr,
      { $U obj:MacDMgr     } Dialogs,
      { $U obj:MacRMgr     } Resources,
      { $U obj:MacMMgr     } Menus,
      { $U obj:MacCMgr     } Controls,
      { $U obj:MacSMgr     } ScrapMgr,
      { $U obj:OSMisc      } OSMisc,
      { $U obj:IntlUnit    } Intl,

      { $U obj:FDefs       } FDefs,
      { $U obj:FUtil       } FUtil,
      { $U obj:FObj's      } FObj's;

{ *****Assembly language procedures in FMain68K***** }

PROCEDURE DialogEvent(theEvent: EventRecord);
PROCEDURE DiskEvent(msg: LONGINT);
PROCEDURE DoClean(wind: WindowPeck);
PROCEDURE DoClear(wind: WindowPeck);
PROCEDURE DoClose;
PROCEDURE DoCloseAll;
PROCEDURE DoCopy(wind: WindowPeck);
PROCEDURE DoCut(wind: WindowPeck);
PROCEDURE DoDuplicate;
PROCEDURE DoPaste(wind: WindowPeck);
PROCEDURE DoEject;
PROCEDURE DoGetInfo;
PROCEDURE DoInitDisk;
PROCEDURE DoOpen;
PROCEDURE DoPrint;
PROCEDURE DoPutBack;
PROCEDURE DoSelectAll(wind: WindowPeck);
PROCEDURE DoSetStartup;
PROCEDURE DoScroll(theWindow: WindowPtr;
                  dh, dv: INTEGER);
PROCEDURE DoUndo(wind: WindowPeck);
PROCEDURE FreezeWindows;
PROCEDURE GetMyScrap;
PROCEDURE HandleDialogEvent(theEvent: EventRecord);

```



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



30 Aug 83

82/BLANK:FMAIN.TEXT

Page 002

```
PROCEDURE HideScrap;
PROCEDURE KillWnd(wnd: WindowPtr);
PROCEDURE LoopStuff;
PROCEDURE ObjEject(params: Ptr;
                  index: INTEGER;
                  theObject: FHandle);
PROCEDURE ObjInfo(params: Ptr;
                 index: INTEGER;
                 theObject: FHandle);
PROCEDURE ObjPutBack(params: Ptr;
                   index: INTEGER;
                   theObject: FHandle);
PROCEDURE OpenAccessory(name: Str255);
PROCEDURE SaveMyScrap;
PROCEDURE ScrollWnd(theCtrl: ControlHandle;
                  theCode: INTEGER);
FUNCTION ScICount: INTEGER;
PROCEDURE ShowScrap;
PROCEDURE WindowClick(theEvent: EventRecord;
                    WindOwner: FHandle);
{ *****Pascal procedures in FMain***** }
IMPLEMENTATION
END. { FMain }
```



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



30 Aug 83

&2/BLANK:FINDER.TEXT

Page 001

```

(
cFinder
obj:tmp
gobj:tmp
obj:Finder
({})
PROGRAM Finder;
(
    The Finder--system shell program.

    Written by Bruce Horn          10 May 1983
    Copyright 1983 by Apple Computer. All Rights Reserved.

    This is the main loop and initialization of the Finder. It calls
    procedures which are defined in FUtil, FObjz, and FMain.

    Last modified 24 December 1983 by Bruce.
)
USES ( $U obj:HeapZone      } HeapZone,
      $U obj:QuickDraw     } QuickDraw,
      $U obj:MacFMgr       } Fonts,
      $U obj:GrafUtil      } GrafUtil,
      $U obj:TextEdit      } TextEdit,
      $U obj:MacEMgr       } Events,
      $U obj:MacMMgr       } Windows,
      $U obj:MacIMgr       } Icons,
      $U obj:MacDskMgr     } DskMgr,
      $U obj:MacDMgr       } Dialogs,
      $U obj:MacRMgr       } Resources,
      $U obj:MacMMgr       } Menus,
      $U obj:MacCMgr       } Controls,
      $U obj:MacSMgr       } ScrapMgr,
      $U obj:OSMisc        } OSMisc,

      $U obj:FDefs         } FDefs,
      $U obj:FUtil         } FUtil,
      $U obj:FObjz         } FObjz,
      $U obj:FMain         } FMain;

( ***** Pascal Procedures in Finder ***** )
PROCEDURE DeskProc( switch: INTEGER;
                   theEvent: EventRecord);      FORWARD;
FUNCTION DiskProc( drive: INTEGER;
                  caller: INTEGER): BOOLEAN;     FORWARD;
PROCEDURE DoMenuCommand( mResult: LONGINT);      FORWARD;
PROCEDURE ExitToShell;                          EXTERNAL;
PROCEDURE EventFetch;                          FORWARD;
FUNCTION ForDialog( theEvent: EventRecord): BOOLEAN; FORWARD;
FUNCTION FFrontWindow: WindowPtr;              FORWARD;
FUNCTION GrowProc( needed: LONGINT): LONGINT;   FORWARD;
PROCEDURE InitDesk;                            EXTERNAL;
PROCEDURE InitHooks( deskProc: ProcPtr;
                    diskProc: ProcPtr;
                    growProc: ProcPtr;
                    resErrProc: ProcPtr);      EXTERNAL;
PROCEDURE InitResident;                       EXTERNAL;
PROCEDURE KeyEvent( theKey: CHAR;
                   theModifiers: INTEGER);     FORWARD;
PROCEDURE Startup;                            FORWARD;
PROCEDURE UpdateCursor( mousePt: Point);       FORWARD;
PROCEDURE UpdateDesk;                         FORWARD;
PROCEDURE UnloadInit;                        EXTERNAL;
PROCEDURE UnloadSeg( p: Ptr );                EXTERNAL;

( ***** Pascal procedures follow ***** )
($R- )
($D- )
($X- )

```



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



30 Aug 83

&2/BLANK:FINDER.TEXT

Page 002

```

($S BigSeg)
PROCEDURE DeskProc(*switch: INTEGER; theEvent: EventRecord*);
( This procedure must be called from the desk hook. It handles hit-testing and
  updating on the desk. )
VAR
  savePort: GrafPtr;
  deskVis: RgnHandle;
  wMgrVis: RgnHandle;
  HookUpdate,
  DskUpdate: RgnHandle;
  deskPeek: WindowPeek;
BEGIN { DeskProc }
  CASE switch OF
    -1: BEGIN
      SetPort(deskWindow);
      WindowClick(theEvent, myDesk);
      END;
    0: BEGIN
      deskPeek := POINTER(ORD(deskWindow));
      DskUpdate := deskPeek.updateRgn;
      HookUpdate := thePort.clipRgn;
      UnionRgn(HookUpdate, DskUpdate); { Union update rgns }
      EraseDPat(thePort.portRect); { Paint deskPat immediately }
      UpdateDesk;
      END;
  END; { Case }
END; { DeskProc }

FUNCTION DiskProc(*drive: INTEGER;
  caller: INTEGER*): BOOLEAN;
VAR
  theDisk: FOHandle;
  vRefNum: INTEGER;
BEGIN { DiskProc }
  IF (caller=1) { FKey } { *OR (caller=2)* } THEN BEGIN
    vRefNum := RefFromDrive(drive);
    theDisk := DiskVol(vRefNum);
    IF theDisk <> NIL THEN BEGIN
      IF ActEdit THEN
        IF IsAncestor(theDisk, objEdit) THEN CommitEdit;
        Object(NIL, 0, theDisk);
        DiskProc := FALSE;
      END
    ELSE BEGIN
      DiskProc := TRUE;
    END;
  ELSE BEGIN
    DiskProc := TRUE;
  END;
END; { DiskProc }

PROCEDURE DoMenuCommand(*mResult: LONGINT*);
VAR myBox: Rect;
    resList: Handle;
    sysEdit: BOOLEAN;
    menuStr,
    myString: Str255;
    w: WindowPeek;
    myDisk: FOHandle;
    theMenu,
    theItem: INTEGER;
BEGIN { DoMenuCommand }
  theMenu := HiWord(mResult);
  theItem := LoWord(mResult);

  w := POINTER(ORD(FFrontWindow));
  IF theItem > 0 THEN
    CASE theMenu OF
      appMenu: { Desk ornament menu }
        BEGIN
          IF theItem = appInfoItem
            THEN ShowAppInfo
          ELSE
            BEGIN
              GetItem(myMenus[1], theItem, myString);
              OpenAccessory(myString);
            END;
          END;
      fileMenu: { File menu }
        CASE theItem OF
          OpenItem: DoOpen;
          InfoItem: DoGetInfo;
          DupItem: DoDuplicate;
          PutBackItem: DoPutBack;
          CloseItem: DoClose;
          CloseAllItem: DoCloseAll;
          PrintItem: DoPrint;
          EjectItem: BEGIN
            IF evtOwner <> myDesk THEN BEGIN
              myDisk := DiskObject(evtOwner);

```



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



30 Aug 83

&2/BLANK:FINDER.TEXT

Page 003

```

        SuperSc10bjcct(FALSE, myDisk);
    END;

    IF (CountObjects(DiskFlavor, evtOwner)=0) THEN BEGIN
        myDisk := DiskSelect; { Pick any disk... }
        IF myDisk<>NIL THEN BEGIN
            SuperSc10bjcct(FALSE, myDisk);
        END;
    END;

    DoEject;
END; { Case }

EditMenu:                                { Edit menu }
BEGIN
    IF theItem<Sc1A11Item THEN sysEdit := SystemEdit(theItem-1)
    ELSE sysEdit := FALSE;

    IF NOT sysEdit THEN
        CASE theItem OF
            UndoItem:      DoUndo(w);
            CutItem:       DoCut(w);
            CopyItem:      DoCopy(w);
            PasteItem:     DoPaste(w);
            ClearItem:     DoClear(w);
            Sc1A11Item:    DoSelectAll(w);
            ClipItem:      ShowScrap;
        END; { Case }
    END;

ViewMenu:                                { Arrangement menu }
ViewObject(theItem, HndOwner(FFrontWindow));

SpecialMenu:                              { Special commands menu }
CASE theItem OF
    CleanItem:      DoClean(w);
    TrashItem:
        BEGIN
            DisposTrash(0);
            FlushAllVols;
        END;

    InitItem:       DoInitDisk;
    StartItem:     DoSetStartup;
END; { Case }

6: IF theItem = 1 THEN ExitToShell;      { Hasmig=Wood }

END; { Case }
HiLiteMenu(0);
END; { DoMenuCommand }

PROCEDURE EventFetch;
VAR
    myEvent:      EventRecord;
    isMyEvent:    BOOLEAN;
    longStr:      Str255;
    findWndCode:  INTEGER;
    whichWindow:  WindowPtr;
    rScrpSize,
    scrpResult,
    menuResult:   LONGINT;
    scrpInfa:     pScrapStuff;
    longResult:   LONGINT;
    wPeck:        WindowPeck;
BEGIN { EventFetch }

    LoopStuff;
    SyncDesk(myDesk);
    UpdateDesk;
    ReviveWindows(myDesk); { Revive any dormant windows }

    IF actEdit THEN TEIdle(TEEdit);

    isMyEvent:=GetNextEvent(everyEvent, myEvent);

    {$IFC AutoUpdate }
    { note new mouse loc + time when moved }
    IF (myEvent.where.h <> eventLoc.h) OR (myEvent.where.v <> eventLoc.v)
    THEN
        BEGIN
            eventLoc := myEvent.where;
            eventTime := myEvent.when;
        END;

    { Also if events coming, don't interrupt with update. }
    IF myEvent.what <> NullEvent THEN eventTime := myEvent.when;

    {$ENDC }

    UpdateCursor(myEvent.where);

    If ForDialog(myEvent)

```



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



30 Aug 83

82/BLANK: FINDER.TEXT

Page 004

```

THEN DialogEvent(myEvent) { Handle a dialog event }
ELSE { Handle a normal Finder event }
  if IsMyEvent THEN
    BEGIN
      CASE myEvent.what OF
        mouseDown: { Mouse Down Event }
          BEGIN
            IF
              (myEvent.when < clickTime+clickDelay) AND
              (AbsPoint(clickLoc,myEvent.where) < 4)
            THEN clickCount:=clickCount+1
            ELSE clickCount:=1;

            clickTime:=myEvent.when;
            clickLoc:=myEvent.where;

            findWndCode := FindWindow(myEvent.where, whichWindow);

            CASE findWndCode OF
              inMenuBar: { InMenuBar }
                BEGIN
                  UpdateMenus(cvtOwner);
                  menuResult := MenuSelect(myEvent.where);
                  DoMenuCommand(menuResult);
                END; { inMenuBar }

              inDrag: { InDrag }
                BEGIN
                  IF NOT featureDown THEN SelectWindow(whichWindow);
                  DragWindow(whichWindow, myEvent.where, screenRect);
                  IF wndKind(whichWindow) = finderKind THEN
                    SetWindParms(wndOwner(whichWindow));
                END;

              inSysWindow: { = inDesk with desk hook } { InSysWindow }
                BEGIN
                  SystemClick(myEvent, whichWindow);
                END;

              inContent: { InContent }
                BEGIN
                  { IF whichWindow<>FrontWindow THEN } SelectWindow(whichWindow);
                  UpdateWindows;
                  IF wndKind(whichWindow) = finderKind
                    THEN WindowClick(myEvent, wndOwner(whichWindow));
                END;

              inGoAway: { InGoAway }
                BEGIN
                  IF TrackGoAway(whichWindow, myEvent.where)
                    THEN DoClose;
                END;

              inGrow: { InGrow }
                BEGIN
                  { IF whichWindow<>FrontWindow THEN } SelectWindow(whichWindow);
                  IF (wndKind(whichWindow)=finderKind) OR
                     (wndKind(whichWindow)=scrapKind)
                    THEN GrowWnd(whichWindow, myEvent.where);

                  IF wndKind(whichWindow) = finderKind THEN
                    SetWindParms(wndOwner(whichWindow));
                END;

            END; { Case findWndCode }

          END;

        keyDown, AutoKey: { Key Down Event }
          BEGIN
            KeyEvent(Chr(myEvent.message MOD 256),
                    myEvent.modifiers);
          END;

        updateEvt: { Update Event }
          BEGIN
            whichWindow := Pointer(myEvent.message);
            UpdateWnd(whichWindow);
          END;

        diskEvt: { Disk Event }
          BEGIN
            DiskEvent(myEvent.Message);
          END;

        activateEvt: { Activate Event }
          BEGIN
            longResult := myEvent.modifiers;
            whichWindow := POINTER(ORD(myEvent.message));
            IF BitAnd(longResult, 1)=1
              THEN ActWnd(whichWindow)
              ELSE InactWnd(whichWindow);
          END;

      END; { CASE event.what }
    END;
  
```



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



30 Aug 83

&2/BANK:FINDER.TEXT

Page 005

```

        END;      { IsMyEvent }

END;      { EventFetch }

FUNCTION ForDialog(*theEvent: EventRecord*):BOOLEAN;
BEGIN { ForDialog }
  ForDialog :=
    IsDialogEvent(theEvent) AND NOT { Command key }
    (
      ((theEvent.what = keyDown) OR (theEvent.what = autoKey))
      AND (BitAnd(theEvent.modifiers, 256)>0)
    );
END;      { ForDialog }

FUNCTION FFrontWindow(*: WindowPtr*);
VAR
  x: WindowPtr;
BEGIN
  x:=FrontWindow;
  IF x = NIL THEN x:=deskWindow;
  FFrontWindow:=x;
END;

FUNCTION GrowProc(*needed: LONGINT*): LONGINT;
BEGIN { GrowProc }
  GrowProc := 0;      { Always return, without any action... }
END;      { GrowProc }

PROCEDURE KeyEvent(*theKey: CHAR; theModifiers: INTEGER*);
VAR
  theObject: FHandle;
  appKey: BOOLEAN;
  menuResult: LONGINT;
BEGIN { KeyEvent }
  appKey := BitAnd(theModifiers, 256)>0;
  IF appKey
  THEN
    BEGIN
      UpdateMenus(cvtOwner);
      MenuResult := MenuKey(theKey);
      DoMenuCommand(menuResult);
    END
  ELSE
    BEGIN { Rename an object }

      IF ActEdit AND ((ORD(theKey)=13) OR (ORD(theKey)=3)) THEN BEGIN
        theObject := ObjEdit;
        CommitEdit;
      END
    ELSE
      IF PrcEdit( POINTER(ORD(@theKey)+1), 1 ) THEN BEGIN
        TcKey(theKey, TEEdit);
        PostEdit;
        NameChanged := TRUE;
      END;
    END;
END;      { KeyEvent }

PROCEDURE Startup;
BEGIN { Startup }
  sysResident := FALSE;      { System stuff can still purge }
  FndrRes := CurResFile;     { Current resource file is FndrRes }
  SetRFAttr(7, TRUE, FndrRes); { Set Finder ReadOnly }

  InitDesk;                  { Load in desk state, find disks, etc. }
  clickTime:=TickCount;
  ViewArrow;
END;      { Startup }

PROCEDURE UpdateCursor(*mousePt: Point*);
VAR
  savePort: GrafPtr;
  lcPoint: Point;
  wndObject,
  theObject: FHandle;
  FWCode: INTEGER;
  inText: BOOLEAN;
  theWindow: WindowPtr;
  theDialog: DialogPtr;
  theKind: INTEGER;
  theItem: Handle;
  theBox: Rect;
  whichCursor: INTEGER;

  PROCEDURE TryTextHit( o: FHandle );
  BEGIN
    IF FindSubObject(lcPoint, theObject, inText, o) THEN BEGIN
      IF
        inText AND
        (TstBit(theObject.f.view, offlinView)=0) AND
        (theObject.f.RRefNum <> FndrRes) AND
        (selectCnt <= 1) AND
        (TstBit( CctAFlags(TRUE, theObject), fLocked)=0)
      THEN BEGIN
        whichCursor:=IBeamCursor;
      END;
    END;
  END;
END;

```



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



30 Aug 83

&2/BLANK:FINDER.TEXT

Page 006

```

        END;
    END;
END;
BEGIN { UpdateCursor }
    IF WndKind(FFrontWindow) < 0 THEN { System Window } Exit(UpdateCursor);
    whichCursor:=arrowCursor; { Assume the arrow cursor }
    GetPart(savePart);
    lclPoint := mousePt;
    FWCode := FindWindow(mousePt, theWindow);
    CASE FWCode OF
        inContent: BEGIN
            SetPart(theWindow);
            GlobalToLocal(lclPoint);
            CASE WndKind(theWindow) OF
                FinderKind: BEGIN
                    wndObject := wndOwner(theWindow);
                    IF (wndObject.view DIV 256 = spatialItem) THEN BEGIN
                        theBox := thePart.portRect;
                        WITH theBox DO BEGIN
                            right := right - sBarHt;
                            bottom := bottom - sBarHt;
                        END;
                        IF
                            ptInRect(lclPoint, theBox)
                        THEN BEGIN
                            TryTextHit( wndObject );
                        END;
                        END; { if spatial view.. }
                    END; { case of finder kind }
                DialogKind: BEGIN
                    theDialog := POINTER(ORD(theWindow));
                    GetDItem(theDialog, 15, theKind, theItem, theBox); { the edit field }
                    IF ptInRect(lclPoint, theBox)
                        AND (theKind=editText) THEN whichCursor:=IBeamCursor;
                END;
            END; { Case wndkind }
        END; { case inContent }
        inSysWindow: BEGIN { = inDesk with desk hook }
            IF theWindow = NIL THEN BEGIN
                TryTextHit( myDesk );
            END;
        END;
    END; { Case }
    SetPart(savePart);
    ViewCursor(whichCursor);
END; { UpdateCursor }
PROCEDURE UpdateDesk;
VAR
    deskPeck: WindowPeck;
    savePart: GrafPtr;
    updateRect: Rect;
BEGIN { UpdateDesk }
    GetPart(savePart);
    SetPart(WMgrPart);
    ClipRect(WMgrPart.portRect);
    ClipAbove(NIL);
    CopyRgn(WMgrPart.clipRgn, deskWindow.visRgn); { start with screen }
    { Find above clip in WMgrPart }
    { Set my vis for beginUpdate }
    deskPeck := POINTER(ORD(deskWindow));
    IF NOT EmptyRgn(deskPeck.updateRgn) THEN
        BEGIN
            BeginUpdate(deskWindow);
            SetPart(deskWindow);
            updateRect := deskWindow.visRgn.rgnBBox;
            EraseDPat(updateRect); { Paint deskPat first }
            DrawSubObjects(updateRect, myDesk);
            EndUpdate(deskWindow);
        END;
    END;
    SetPart(savePart);
END; { UpdateDesk }
($S )
BEGIN { Finder }
    InitResident; { Initialize all pointers }
    { InsertMenu(GetMenu(6), 0); DrawMenuBar; } { FOR HASMIG }
    Startup;

```



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



30 Aug 83

82/BLANK:FINDER.TEXT

Page 007

```
UnloadInit;                { Drop the init segment for sure }
REPEAT
  MegaMachine;             { Do file copics }
  EventFetch;              { Start finder event loop }
  UseResFile(FndrRes);
  IF NOT BigMac THEN BEGIN
    UnloadSeg(@SegBigSeg);  { Unload BigSeg seg }
    UnloadSeg(@SegOther);   { Unload Other seg }
  END;
UNTIL FALSE;
END. { Finder }
```




APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



30 Aug 83

82/BLANK:FINDER68K.TEXT

Page 002

```

@1
; Unload the scrap if inordinately large...(can't call get size here...)
      CLR      CDdeadScrap(A5)      ; set indicator to false
      SUBQ    #4,SP
      MOVE    #4,-(SP)
      JSR     GetSize              ; leave on stack
      SUBQ    #4,SP
      _InfoScrap
      MOVE.L  (SP)+, A0            ; Save space for ptr to info
                                      ; get the information
                                      ; Get ptr to A0
      MOVE.L  (SP)+,D1
      CMP.L   (A0),D1              ; if > 5000
      BGT.S   @2
      SUBQ    #4,SP
      _UnladeScrap
      TST.L   (SP)+                ; zero scrap if errors
      BPL.S   @2

; Not enough room for scrap on disk, tell user in loop
      SUBQ    #4,SP                ; room for result
      _ZeroScrap
      ADDQ    #4,SP                ; ignore result
      ST      CDdeadScrap(A5)      ; set indicator

@2
      JSR     SegBigSeg            ; Force in main segments.
      JSR     SegOther
      JSR     InitMain            ; Init the other segment
      RTS

UnloadInit
      SUBQ    #4,SP                ; get address of segment
      JSR     InitSeg
      MOVE.L  (SP)+,A0            ; get segment address from initmain
      _RecoverHandle
      MOVE.L  A0,-(SP)
      _ReleaseResource
      RTS

;=====Finder68K=====
;
;      Procedure InitHooks(deskProc: ProcPtr; 20(A6)
;                          diskProc: ProcPtr; 16(A6)
;                          growProc: ProcPtr; 12(A6)
;                          resErrProc: ProcPtr); 8(A6)
;
;      Install an update procedure into the desk hook, a disk procedure
;      into the disk-switch hook, a grow procedure into growZone, and
;      an error proc into the resErr hook.
;
;      Only the DeskProc and the ResErrProc require filters--the Grow and
;      the disk procs (eject hook) are called directly.
;=====Finder68K=====

IHResProc .Word 0,0 ; placeholder for resErr proc
IHGrowProc .Word 0,0 ; placeholder for grow proc
IHDiskProc .Word 0,0 ; placeholder for disk proc
IHDeskProc .Word 0,0 ; Placeholder for deskProc

InitHooks
      LINK    A6,#0
      MOVEM.L A2-A4,-(SP) ; save regs
      LEA    8(A6),A2 ; Point to last parameter
      LEA    IHResProc,A1 ; and first hook address.
      MOVE.L (A2)+,(A1)+ ; Save resErr procptr
      LEA    ResErrFilter,A0
      MOVE.L A0,ResErrProc
      MOVE.L (A2)+,(A1)+ ; Save growZone procptr
      LEA    GrowFilter,A0
      _SetGrowZone
      MOVE.L (A2)+,(A1)+ ; Save procptr in disk Proc
      LEA    DiskFilter,A0
      MOVE.L A0,EjectNotify
      MOVE.L (A2)+,(A1)+ ; Save procptr in desk Proc
      LEA    DeskFilter,A0
      MOVE.L A0,DeskHook
  
```



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



30 Aug 83

82/BLANK: FINDER68K.TEXT

Page 003

```

MOVEM.L (SP)+, A2-A4 ; Restore regs
UNLK A6 ; unlink

MOVEM.L (SP)+, A0 ; Get return address
ADD #16, SP ; Pop off parameters
JMP (A0) ; ...and return.

GrowFilter
MOVEM.L IHGrowProc, A0 ; Get the proc ptr
JMP (A0) ; call it instead.

DiskFilter
MOVEM.L IHDiskProc, A0 ; Get the proc ptr
JMP (A0) ; call it instead.

DeskFilter
MOVEM.L A0-A4/D0-D7, -(SP) ; MacPasLib smashes regs!
MOVE.W D0, -(SP) ; Pass switch
MOVE.L A0, -(SP) ; and possible event record
MOVEM.L IHDeskProc, A0 ; Get the proc ptr
JMP (A0) ; and call it.
MOVEM.L (SP)+, A0-A4/D0-D7 ; MacPasLib smashes regs!
RTS ; and return. Ignore hit.

ResErrFilter
MOVEM.L A0-A4/D0-D7, -(SP) ; MacPasLib smashes regs!
MOVE.W D0, -(SP) ; Pass error code
MOVEM.L IHResProc, A0 ; Get the proc ptr
JMP (A0) ; and call it.
MOVEM.L (SP)+, A0-A4/D0-D7 ; MacPasLib smashes regs!
RTS ; and return. Ignore hit.

.Seg 'Init'
.Proc Finder68k

;-----
; FUtil, FUtil68K References
;-----
.Ref CleanVals
.Ref Collect
.Ref DriveFromRef
.Ref FatalError
.Ref GetSize
.Ref OpenFile
.Ref RefFromDrive
.Ref ViewHour

;-----
; FObj, FObj68K References
;-----
.Ref ActObject
.Ref AddSubObject
.Ref DrawSubObjects
.Ref MakeWindow
.Ref NewObject
.Ref OnlineDisk
.Ref SetName

;-----
; FMain, FMain68K References
;-----

;-----
; Finder References
;-----
.Ref GrowProc
.Ref DeskProc
.Ref DiskProc

;-----
; Finder68K Definitions
;-----
.Ref InitHooks
.Ref HyperDown
.Ref SetFndrFont

.Def InitDesk
.Def InitMain
.Def InitSeg

; This returns the address of self for unloading
InitSeg
LEA InitSeg, A0 ; for unloading
SUBQ #4, A0 ; return the address of segment
MOVE.L A0, 4(SP) ; drop segment header????(MAGIC)

RTS

```



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



30 Aug 83

&2/BLANK:FINDER68K.TEXT

Page 004

```

=====F0bj:68K=====
;
;
;   PROCEDURE InitDesk;
;
;   Allocate the desk object, trash object, and anonymous disk in the
;   Finder resource file (read/only).
;
=====F0bj:68K=====
;
;   .IF   Alternate
;
IDAnonWind .Word 236, 2, 338, 428
IDAnonLoc  .Word 1000, 1000
;
;   .ENDC
;
;   ; Offsets from ScreenRect botRight (except for topLeft of wind)
;
IDTrshWind .Word -106, 2, -4, -84
IDTrshLoc  .Word -58, -56
;
IDString   .EQU    -64           ; temp string
IDStk      .EQU    IDString
;
InitDesk
;
;   LNK      A6, #IDStk           ; Link stack frame
;   MOVEM.L A2-A4/D3-D7, -(SP)   ; Save regs
;
;   ST      -(SP)                ; Unmount offline vols
;   JSR     CleanVols            ; + flush online ones.
;
;   LEA     GScreenRect(A5), A3  ; Point to screen rectangle
;   MOVE.L  BotRight(A3), D7     ; Put botRight in D7.
;
;   MOVEQ   #FldrFlavor, D3      ; Set the flavor= folder
;   CLR.L   D4                   ; location 0,0
;   MOVE.L  #$01800000+FldrIcon, D5 ; spatial+windView/FldrIcon (unused)
;   MOVEQ   #1, D6               ; Use "Desktop" for name
;   MOVEQ   #deskID, D0          ; desk object ID
;   BSR.S   IDCreate             ; Create the object.
;
;   MOVE.L  A2, A4               ; Copy desk to A4
;   MOVE.L  (A4), A0             ; dereference desk
;   MOVE.L  GDeskWindow(A5), A1  ; Get the desk window in A0
;   MOVE.L  A1, wind(A0)        ; Set deskWindow in myDesk
;   MOVE.L  A4, WRefCon(A1)     ; Set refcon to myDesk in window
;
;   MOVE.L  A4, GMyDesk(A5)     ; and set the global.
;   MOVE.L  A4, GEvtOwner(A5)   ; and set the evt owner
;
;   .IF   Alternate             ; Alternate disk?
;
;   MOVEQ   #altDiskID, D0      ; Create the anonymous disk
;   MOVEQ   #AnonFlavor, D3     ; Set the flavor= anonymous
;   MOVE.L  IDAnonLoc, D4       ; location 0,0
;   MOVE.L  #$01040000+DiskIcon, D5 ; closedView/offline/DiskIcon
;   MOVEQ   #2, D6             ; Use "Alternate Disk" for name
;   LEA     IDAnonWind, A3     ; and initial window
;   BSR.S   IDCreate           ; Create the object.
;
;   MOVE.L  A2, -(SP)           ; Push the new object
;   MOVE.L  A4, -(SP)           ; Push myDesk
;   JSR     AddSubObject       ; add it to desk
;
;   MOVE.L  A2, GMyAnon(A5)    ; and set the global
;
;   .ENDC
;
;   MOVEQ   #TrshFlavor, D3    ; Set the flavor= trsh
;
;   MOVE.L  D7, -(SP)           ; Push botRight
;   PEAK   IDTrshLoc           ; Push trash offset
;   _AddPt ; and add them to get absolute loc.
;
;   MOVE.L  IDTrshLoc, D4      ; location
;
;   MOVE.L  #$01000000+TrshIcon, D5 ; closedView/TrshIcon
;   MOVEQ   #5, D6             ; Use "Trash" for name
;   LEA     IDTrshWind, A3     ; Get offsets from screenRect
;
;   MOVE.L  D7, -(SP)           ; Push botRight, but
;   CLR.W   2(SP)              ; replace right by left(=0)
;
;   MOVE.L  A3, -(SP)           ; Push offset
;   _AddPt ; Add them, result is absolute.
;
;   ADDQ   #4, A3              ; Skip to botRight
;
;   MOVE.L  D7, -(SP)           ; Push botRight
;   MOVE.L  A3, -(SP)           ; Push offset
;   _AddPt ; Add them, result is absolute.
;
;   SUBQ   #4, A3              ; Point back to window rect
;

```



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



30 Aug 83

&2/BLANK:FINDER68K.TEXT

Page 005

```

MOVEQ    #trashID, D0      ; trash can ID
BSR.S    IDCCreate         ; Create the object.

MOVE.L   A2, -(SP)        ; Push the new object
MOVE.L   A4, -(SP)        ; Push myDesk
JSR      AddSubObject     ; add it to desk

MOVE.L   A2, GMyTrash(A5) ; and set the global

; Create the object for the boot drive initially.

SUBQ     #4, SP           ; Save space for result
MOVE.W   BootDrive, -(SP) ; Push the boot drive
MOVE.L   A4, -(SP)        ; push myDesk to be owner
JSR      OnlineDisk      ; Create the disk, and
ADDQ     #4, SP           ; ignore result.

MOVE.L   A4, -(SP)        ; Push myDesk
JSR      ActObject       ; and make it the active object.

MOVEM.L  (SP)+, A2-A4/D3-D7 ; Restore regs
UNLK     A6               ; Unlink
RTS

; Create a new object.
; D0 = object ID
; D3 = flavor
; D4 = location
; D5 = icon ID (low)+view (high)
; D6 = name ID

; A3 = rectangle address
; Returns A2 = object handle.

IDCCreate
SUBQ     #4, SP           ; Save space for result
MOVE.W   D0, -(SP)       ; Set ID
JSR      NewObject       ; Create a new object of type
MOVE.L   (SP)+, A2        ; Save it in A2.

MOVE.L   (A2), A1        ; Dereference
MOVE.W   D3, (A1)+       ; Set the flavor
MOVE.L   D4, (A1)+       ; Set the location
CLR.W    (A1)+           ; Clear lineIndex initially
MOVE.W   D5, (A1)+       ; the icon ID
SWAP     D5              ; Switch to view
MOVE.W   D5, (A1)+       ; the view
MOVE.W   #dcskID, (A1)+  ; Set homeID to desk
MOVE.L   A4, (A1)+       ; the home and homeLink
MOVE.L   A4, (A1)+       ; both = desk.
CLR.L    (A1)+           ; no chain yct
MOVE.L   Time, (A1)+     ; Set creation date
MOVE.L   Time, (A1)+     ; and modification date to now.
CLR.W    (A1)+           ; Set vRefNum = 0
MOVE.W   GFnDrRes(A5), (A1)+ ; and rRefNum = finder resource refnum.
CLR.W    (A1)+           ; flagz := 0.
CLR.L    (A1)+           ; no initial window.
MOVE.L   (A3)+, (A1)+    ; Copy topLeft
MOVE.L   (A3)+, (A1)+    ; and bottomRight of rect.
CLR.L    (A1)+           ; scrollLoc initially 0,0
CLR.W    (A1)+           ; and itemCount = 0.

PEA      IDString(A6)     ; Push string index
MOVE.W   D6, -(SP)       ; Push string index
JSR      GetStr          ; Get it

PEA      IDString(A6)     ; Push the pointer
MOVE.L   A2, -(SP)       ; and my new object
JSR      SetName         ; set the name
RTS

;=====Finder68K=====
;
;   InitMain;
;
;   Once-only initialization.  Restarts don't call InitMain again.
;=====Finder68K=====

ScrpWind .Word 246, 4, 336, 428

IMString .EQU    -64      ; temp string
IMIOBik  .EQU    IMString-IOQEISize ; io command block
IMStk    .EQU    IMIOBik

InitMain
LINK     A6, #IMStk      ; Link stack frame
MOVEM.L  A2-A4/D3-D7, -(SP) ; Save all regs

; Max out the heap

SUBQ     #4, SP           ; save for stack size
MOVE.W   #9, -(SP)       ; push index
JSR      GetSize

```



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



30 Aug 83

&2/BLANK: FINDER68K.TEXT

Page 006

```

MOVE.L    (SP)+, D0          ; Get stack size from resources.
MOVE.L    SP, A0             ; set appl limit
SUB.L     D0, A0             ; subtract stack size
    _SetApplLimit           ; leave at least 8K for QuickDraw.

; expand the heap w/o purging

SUB       #1000, A0          ; fudge factor
MOVE.L    A0, D0             ; get the appl limit
SUB.L     HeapEnd, D0        ; figure out how much
    _NewHandle              ; Get that handle
    _DisposeHandle

;
SUBQ      #4, SP             ; Save space for hndl
MOVE.L    #RLstType, -(SP)   ; Push resource list type
MOVE.W    #132, -(SP)        ; 132 includes all system stuff
;
; GetResource                ; Get it
; JSR      Collect           ; go get them in

; Flush the events and initialize the rest of the managers

    _TEInit                 ; Initialize TextEdit

CLR.L     -(SP)              ; Push NIL
    _InitDialogs            ; Initialize dialogs
    _InitMath                ; Initialize packages

JSR      ViewHour            ; hourglass during startup
    _ShowCursor              ; Show the hourglass.

; If the feature+option keys are down, trash desktop and relocate all icons.

SUBQ      #2, SP             ; save for boolean
JSR      HyperDown           ; is feature+option key down?
TST.B     (SP)+
SNE       CRestart(A5)       ; if so, trash desktop

; Figure out the boot volume and patch back for Larry.

SUBQ      #2, SP             ; Save space for reference
MOVE.W    BootDrive, -(SP)   ; push boot drive number
JSR      RefFromDrive        ; Get the refnum of the volume
MOVE.W    (SP)+, BootDrive    ; patch back in low memory.

; Set up window mgr port and screenrect.

PEA       WMgrPort(A5)       ; Point to WMgrPort VAR
    _GetWMgrPort            ; and get it.

MOVE.L    WMgrPort(A5), A4    ; Get the port
LEA       PortRect(A4), A0    ; Point to the portRect
LEA       GScreenRect(A5), A1 ; and my screen rect
MOVE.L    (A0)+, (A1)+        ; copy the rect(1)!
MOVE.L    (A0)+, (A1)+        ; copy the rect(2)!

; Create the special back window

LEA       GDeskWindowRecord(A5), A2 ; Get desk window record

MOVE.L    A2, A0             ;
MOVEQ     #WindowSize/2+1, D0 ; Get number of words +1 for DBRA
@1        CLR.W    (A0)+
DBRA     D0, @1              ; clear out the window record.

MOVE.L    A2, -(SP)          ; Push the addr
    _OpenPort                ; open the port to init fields.

SUBQ      #4, SP             ; Save space for rgn
    _NewRgn                   ; make a new region for updates

MOVE.L    (SP), A3           ; copy update rgn in A3.
PEA       PortRect(A2)       ; point to the portRect
    _RectRgn                  ; set update to portRect

MOVE.W    #FinderKind, WindowKind(A2); Set windowKind=finderKind.
MOVE.L    A3, UpdateRgn(A2)  ; set up update rgn

MOVE.L    A2, GDeskWindow(A5) ; set desk window.
MOVE.L    A2, -(SP)          ; Set the port initially.
    _SetPort

JSR      SetFndrFont         ; Use the finder font

; force the font in, while we're over system.

SUBQ      #2, SP             ; save for width
MOVE.W    #32, -(SP)         ; push space
    _CharWidth                ;
ADDQ     #2, SP              ; load the font now

; Make a new TextEdit record

SUBQ      #4, SP             ; initiate record
PEA       TempRect

```



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



30 Aug 83

82/BLANK:FINDER68K.TEXT

Page 007

```

MOVE.L    (SP),-(SP)
_TENew
MOVE.L    (SP),GTEEdit(A5)    ; save to handle
MOVE.L    (SP)+,A0             ; set up as centered
MOVE.L    (A0),A0
MOVE      #1,TEStylc(A0)

; Set up the menus

_InitMenus                ; Initialize unit menus

MOVEQ     #MLastMenu-1, D7    ; Get last menu index -1 for DBRA
LEA       GMyMenus(A5), A4   ; Point to MyMenus

@2
MOVEQ     #MLastMenu, D0     ; Get last menu
SUB.W     D7, D0             ; - current index gives 1's index
SUBQ      #4, SP             ; Save space for the menu
MOVE.W    D0, -(SP)         ; Push the index
_GctRMenu                ; and get the menu
MOVE.L    (SP), -(SP)       ; Copy for detach
MOVE.L    (SP), (A4)+       ; Install it in MyMenus
_DetachResource          ; Detach it from resources
CLR.W     -(SP)             ; push 0
_InsertMenu              ; Insert the menu
DBRA      D7, @2           ; and loop back.

LEA       GMyMenus(A5), A4   ; Point to MyMenus
MOVE.L    (A4), A3          ; Get the first menu
MOVE.L    (A3), A0          ; Dereference
MOVE.B    #ChrApple, menuData+1(A0) ; Set apple character

MOVE.L    A3, -(SP)         ; Push the first menu
MOVE.L    #DrvType, -(SP)   ; Push driver type
_AddResMenu              ; add on desk ornaments

_DrawMenuBar              ; Draw the menu bar!

; Initialize the hooks and globals

PEA       DeskProc
PEA       DiskProc
PEA       GrowProc
PEA       FatalError
JSR       InitHooks        ; Install the various hooks

SF         GNameChged(A5)   ; no change
SF         GActEdit(A5)    ; no editing yet
CLR.L     GObjEdit(A5)     ; no object being edited

CLR       MState(A5)       ; no mega move state
CLR.L     MBootBlocks(A5)  ; no boot blocks
CLR       MNFiles(A5)      ; no files
MOVEQ     #0, D0           ; allocate a dummy file list handle

_NewHandle
MOVE.L    A0, MFileList(A5)
MOVEQ     #0, D0
_NewHandle                ; and a dummy cmd safe
MOVE.L    A0, MCmdSafe(A5)

; Marks all the windows in global as free

MOVEQ     #8, D1           ; counter
MOVEQ     #-1, D2          ; handy number
LEA       GWindows(A5), A0 ; point to window array

wiLoop
MOVE.L    D2, (A0)         ; deallocate it

ADD       #DWindowLen, A0  ; next..
DBRA      D1, wiLoop

; Load the scrap and create the scrap window

SUBQ      #4, SP           ; Save space for error
_LoadScrap
MOVE.L    (SP)+, D0        ; Load in the scrap
JSR       ChkError         ; Get error code
; check for errors

PEA       IMString(A6)
MOVE.W    #4, -(SP)        ; Push "Clipboard" index
JSR       GetStr           ; Get the string

SUBQ      #4, SP           ; Save space for window
PEA       IMString(A6)     ; push name
MOVE.W    #ScrapKind, -(SP) ; Push scrap kind

SUBQ      #4, SP           ; Save space for ptr to info
_InfoScrap
MOVE.L    (SP)+, A0        ; get the information
MOVEQ     #0, D0           ; Get ptr to A0
MOVE.W    9(A0), D0        ; D0 is the scrapCount
MOVE.L    D0, -(SP)       ; Push for refcon

CLR.B     -(SP)            ; Not visible
CLR.B     -(SP)            ; no controls
PEA       ScrapWind       ; Use the default rect.

```



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



30 Aug 83

&2/BLANK:FINDER68K.TEXT

Page 008

```
JSR      MakeWindow      ; Make a generic window
MOVE.L  (SP)+, GScrapWindow(A5) ; and copy to globals.

; Hack to clean up swaps

SUBQ    #4, SP           ; get the I BEAM
MOVE    #IBeamCursor, -(SP)
_GetCursor
ADDQ    #4, SP

MOVEM.L (SP)+, A2-A4/D3-D7 ; Restore all regs
UNLK    A6               ; unlink
RTS     ; and return.

.End
```



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Nybbles

Author: Andy Hertzfeld
Date: February 1981
Characters: Andy Hertzfeld, Steve Wozniak, Burrell Smith, Brian Howard
Topics: Hardware Design, Prototypes, Software Design, Early Programs,
Technical
Summary: My first job on the Macintosh project



An Apple II Disk Drive

My very first job on the Mac project was to help Burrell Smith and Brian Howard verify that the disk controller was working properly. They had just wired up a Woz-style disk controller to the Macintosh prototype, but they had no way to test if it was working properly without writing a fairly complicated program.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



When I arrived on the scene, they were trying to debug a small 68000 routine that they had written together. They had written a simple loop to access the disk controller and were watching it execute with a logic analyzer. But neither of them were much of a programmer, so I was able to help right away, even though I had never programmed the 68000 before.

The Apple II disk controller, designed six months after the Apple II itself was complete, was Steve Wozniak's crowning achievement. His five chip disk controller card out-performed competitive controllers that were four times as expensive by shifting most of the responsibilities from hardware to software. In Woz's approach, the software was responsible for doing all of the encoding and decoding, head stepping, etc. This allowed Woz to improve the capacity and performance over standard techniques.

In those days, most floppy disks used a recording technique called FM encoding, where one clock bit would precede each data bit to improve reliability. But that was overkill, it was possible to get more data on a disk if you used some of the clocks for data. So Woz used a technique called "group encoding" (although Woz called it "nybbleizing") to get five bits out of every eight transitions instead of four. He later figured out how to use six bits out of eight (the "16 sector" format vs. the earlier "13 sector" - non-Wozian drives used 10 sectors) improving the density further.

The software stored eight transitions at a time into the disk controller's shift register. Since each transition was 4 microseconds long, each "nybble" of data had to be provided every 32 microseconds. Eventually, Woz figured out how to decode the data on the fly, in between fetching nybbles, so he could achieve the ultimate in performance, one to one interleave. But he never could figure out how to encode on the fly, since the Apple II's microprocessor just wasn't fast enough, and the timing had to be more precise for writing.

But the Macintosh's microprocessor was at least four times faster than the Apple II, more like 10 times if you were manipulating 32 bit values, so writing one-to-one interleave disk routines was simple on the Macintosh. It was fun for me to achieve the holy grail of disk performance without breaking into a sweat. But first there was a big problem to solve.

Woz's disk technology required that the software feed it new data every 32 microseconds exactly. If we were even a single microsecond early or late, it would cause a glitch in the data and ruin it. In order to write the routines, I needed to know how fast the Macintosh executed each instruction. The manual gave the number of clocks for each instruction, but I wasn't sure how long it took to fetch from memory. So of course, I asked Burrell what the timings were, but I was surprised at his response.

"I don't know. The Mac is synchronous, just like the Apple II, so each instruction has the same timing, every time you execute it, so you will be able to write disk routines that have exact timing. I don't know what it is, so we'll just measure it. Why don't you write your routine and we'll measure it with the logic analyzer."

So I spent a couple of days writing the basic routines, and then sat down with Burrell and Brian in front of the logic analyzer and we watched each instruction



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



execute, writing down how long each instruction took. They usually worked like we expected, but occasionally some things were surprising and I had to adjust the code. After a few fixes, I had the raw disk routines both reading and writing, doing the encoding and the decoding on the fly, achieving Woz's long sought after one-to-one interleave.

I expected to feel elated when I finally got the disk reading and writing, but it didn't feel that satisfying, because you couldn't really see it in action. The previous year, soon after Woz wrote his one-to-one interleave read routines, I made some fast slideshow disks for the Apple II, where the screen was filled with a new image in less than a second, twice as fast as previously possible. I thought it would be fun to use the new disk routines to read the slideshow disks on the Mac.

By this time, it was around 7pm and everyone else was going out to dinner. I was invited to come, but I was so close to getting the slideshow working, that I didn't think that I could concentrate on eating until I got it done. I was alone in the office when I finally got it working, the embryonic Mac reading and displaying images from an Apple II disc as fast as possible. It was far and away the coolest thing a Mac could do so far. It was fun to show it to everyone when they came back from dinner.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Early Demos

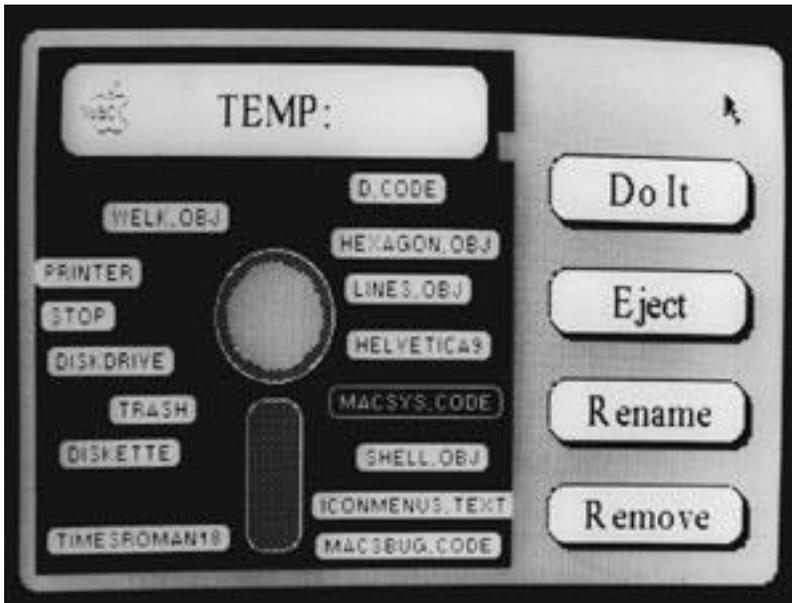
Author: Andy Hertzfeld

Date: April 1981

Characters: Andy Hertzfeld, Bud Tribble, Rich Page, Bob Bishop,
Bill Atkinson, Bruce Horn

Topics: Hardware Design, Prototypes, Software Design, Early Programs

Summary: Various demos showed off what the hardware could do



Early Finder Prototype from Feb 1982

The first demo program for the 68000-based Macintosh was written by Bud Tribble, as part of the original boot ROM. It filled the screen with the word 'hello' in



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



tiny letters, more than a hundred times. When the Mac was switched on, it performed some hardware diagnostics, filled the screen with 'hello', and then listened to its serial ports for commands to execute. The 'hellos' told us that everything was working OK.

The boot ROM allowed us to download other programs from the Lisa to the Mac over a serial line, to try out new code and test or demo the prototype. There was a ton of work to do: writing an operating system, hooking up the keyboard and mouse, getting Bill's graphics and UI routines running, and many other tasks, but we also sometimes just did things for fun.

In early March of 1981, I wrote a fast, disk-based slideshow for the Mac the same night that I got the disk routines going. It was exciting to see detailed, relatively high resolution images parading across the display so quickly.

By April, I was experimenting with writing custom graphics routines, to show off the raw graphical horsepower of the system. I had written a few ball-bouncing routines on the Apple II, and I thought it would be interesting to see how many balls could the Mac animate smoothly. I wrote some 68000 code to draw 16 by 16 images very quickly, and I found that I could keep more than one hundred balls animating smoothly, which seemed pretty impressive. I also wrote a small sketch program with a seed fill using Bill Atkinson's 8 by 8 pattern bitmaps, as well as an entertaining Breakout game, where I implemented Bud's idea of having to dodge the bricks when they fell down after you hit them.

Bob Bishop had experimented with a variety of graphical special effects on the Apple II, so I thought I'd try some of them out on the Mac. The idea was to transfer an image onto the screen in an entertaining way. The one that I liked best was a kind of waterfall effect, where you copied an image onto the screen using a varying number of multiple copies of successive scan lines, stretching the image vertically. The image looked like it poured onto the screen like water going over a waterfall; it somehow was rather hypnotic. I often used it with an image of the Muppets I converted from the Apple II, and the "Stretching Muppets" demo became pretty well known.

In May 1981, Bud stayed up all night and ported QuickDraw and some pull-down menu code from the Lisa to the Mac. For the first time, we were running mouse-based software with real pull-down menus. The best part of the demo was the pattern menu, which showed off the extensibility of the menu routines to draw an entirely graphical menu.

In June 1981, we realized that it would be worthwhile to create a stand-alone demo environment, where the Macintosh booted and ran programs from its own disk, even though we'd only use it temporarily. Our own operating system wasn't close to usable yet, but Rich Page had written a simple operating system called the "Lisa Monitor" which was based on UCSD Pascal, that was pretty easy to port - all we had to do was integrate our I/O drivers. Soon, using the Monitor, a Mac could boot up and run demos without help from a Lisa.

In the Lisa Monitor environment, it was easy for us to run QuickDraw-based programs. Soon, we had a Window Manager demo, featuring balls bouncing in multiple windows, as well as a nice icon editor and MacSketch, an early ancestor of MacPaint going.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



I think the most interesting early demo was an early prototype for the Finder, written by Bruce Horn and myself in the spring of 1982, and pictured above. Its window was filled with an image of a floppy disc, over which the files were represented as draggable tabs. You could select files and perform operations on them by selecting them and then clicking on command button. It predated Bill's design for the Finder for Lisa, which we eventually adopted instead, and is an interesting glimpse of alternative designs that we might have chosen instead of what seems so familiar today.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Round Rects Are Everywhere!

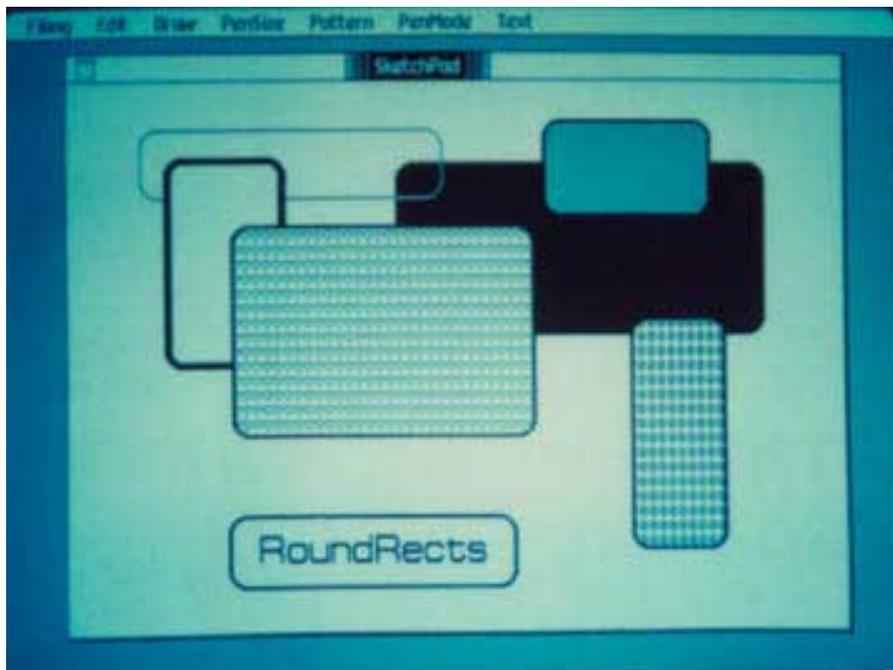
Author: Andy Hertzfeld

Date: May 1981

Characters: Steve Jobs, Bill Atkinson

Topics: Software Design, Inspiration, QuickDraw

Summary: Steve inspires Bill by pointing out something about the real world



Bill Atkinson worked mostly at home, but whenever he made significant progress he rushed in to Apple to show it off to anyone who would appreciate it. This



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



time, he visited the Macintosh offices at Texaco Towers to show off his brand new oval routines, which were implemented using a really clever algorithm.

Bill had added new code to QuickDraw (which was still called LisaGraf at this point) to draw circles and ovals very quickly. That was a bit hard to do on the Macintosh, since the math for circles usually involved taking square roots, and the 68000 processor in the Lisa and Macintosh didn't support floating point operations. But Bill had come up with a clever way to do the circle calculation that only used addition and subtraction, not even multiplication or division, which the 68000 could do, but was kind of slow at.

Bill's technique used the fact the sum of a sequence of odd numbers is always the next perfect square (For example, $1 + 3 = 4$, $1 + 3 + 5 = 9$, $1 + 3 + 5 + 7 = 16$, etc). So he could figure out when to bump the dependent coordinate value by iterating in a loop until a threshold was exceeded. This allowed QuickDraw to draw ovals very quickly.

Bill fired up his demo and it quickly filled the Lisa screen with randomly-sized ovals, faster than you thought was possible. But something was bothering Steve Jobs. "Well, circles and ovals are good, but how about drawing rectangles with rounded corners? Can we do that now, too?"

"No, there's no way to do that. In fact it would be really hard to do, and I don't think we really need it". I think Bill was a little miffed that Steve wasn't raving over the fast ovals and still wanted more.

Steve suddenly got more intense. "Rectangles with rounded corners are everywhere! Just look around this room!". And sure enough, there were lots of them, like the whiteboard and some of the desks and tables. Then he pointed out the window. "And look outside, there's even more, practically everywhere you look!". He even persuaded Bill to take a quick walk around the block with him, pointing out every rectangle with rounded corners that he could find.

When Steve and Bill passed a no-parking sign with rounded corners, it did the trick. "OK, I give up", Bill pleaded. "I'll see if it's as hard as I thought." He went back home to work on it.

Bill returned to Texaco Towers the following afternoon, with a big smile on his face. His demo was now drawing rectangles with beautifully rounded corners blisteringly fast, almost at the speed of plain rectangles. Lisagraf called the new primitive "RoundRects". Over the next few months, roundrects worked their way into various parts of the user interface, and soon became indispensable.





APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Apple II Mouse Card

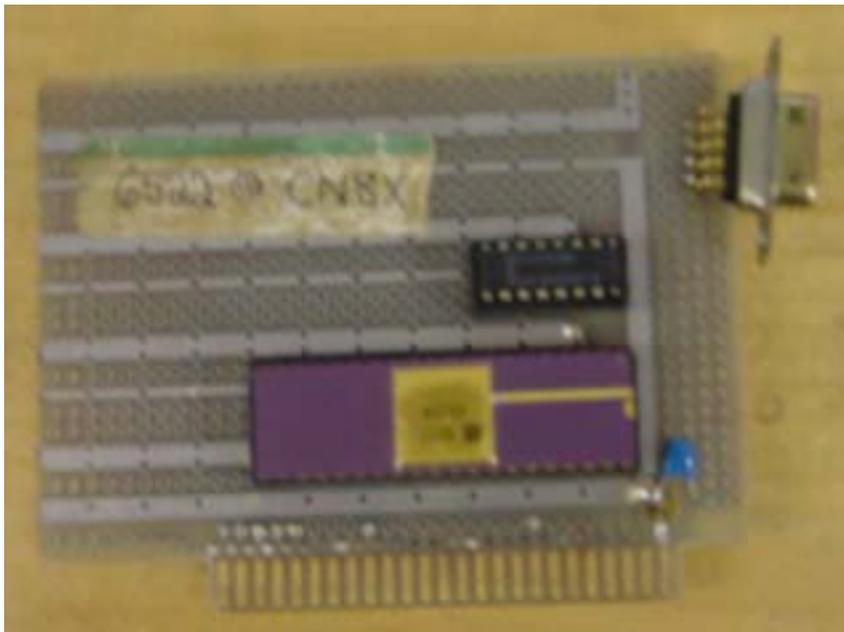
Author: Andy Hertzfeld

Date: June 1981

Characters: Andy Hertzfeld, Burrell Smith, Dan Kottke, Bill Budge, Steve Jobs

Topics: Software Design, Apple Spirit, Prototypes, Technical

Summary: We hooked up a mouse to the Apple II



Apple II Mouse Card Prototype



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Burrell's Macintosh design was elegant and clever in many different ways, always using minimal hardware resources for maximal value.

The mouse interface was a typical example, using just a fraction of the general purpose 6522 VIA chip that Burrell had incorporated. The mouse is hooked up to the 6522 so that it generates an interrupt each time the mouse moved a notch horizontally or vertically, with a one bit line to sense the mouse button. That was it - the rest was done in software.

Meanwhile, my good friend Bill Budge, who was an amazing Apple II games programmer, came up with an incredibly fast way to draw proportional text on the Apple II graphics screen, using seven pre-shifted tables. He also wrote a very fast BitBlit routine (BitBlit was the most important graphics primitive in GUIs). It was crying out for a way to interface the mouse, so you could have a complete graphical user interface on the Apple II.

Bud Tribble had written the initial code to drive the mouse on the Macintosh, but I had to maintain it, so I was familiar with how it worked. I had used a 6522 chip the previous year, since the Apple III had one, which I had programmed to shift bits out to the Silentype thermal printer. I realized that all that was required to interface a mouse to the Apple III was to make a simple connector to route the appropriate signals to the proper pins.

I asked Dan Kottke to wire up a connector. Once that was finished, it only took a few hours to write cursor routines and a driver. We had the beginnings of a full graphical user interface going on the Apple III, with no additional hardware required.

It was exciting enough that I started helping Bill Budge flesh out the program, careful to work on it only after regular hours. Pretty soon, we had pull-down menus going, and then a simple proportional text word processor. The only problem was that Apple IIIs weren't selling very well; we really needed to get it running on the Apple II somehow.

I talked to Burrell about the problem, which basically came down to making a peripheral card for the Apple II that included a 6522 VIA chip on it. Burrell came back the next day with a brilliant two chip design, just the VIA and a tiny dual flip-flop chip to synchronize it.

To move the mouse cursor without flickering, both the Mac and the Apple III code used a vertical blanking interrupt, which drew the cursor during the time interval when no video was being painted. Unlike the Mac and Apple III, the Apple II didn't have a vertical blanking interrupt. The most brilliant part of Burrell's design involved how it synchronized with the video, without running a wire to pick up the video signal.

The 6522 chip had a timer which could generate an interrupt at a specified interval. The problem was synchronizing it with the video, since the video generation was not accessible to the processor. Burrell solved the problem by wiring up the spare flip flop to the low bit of the data bus, using it to latch whatever data the video was displaying so the processor could read it.

To synchronize with the video, Burrell had me fill the Apple II's frame buffer so the low bit was on most of the time, but set off at the end of the last scan



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



line. I wrote a routine to sit in a tight loop, reading the latch. When the low bit changed, we would know the vertical blanking interval had just begun.

The Apple II wasn't fast enough to keep up with its own frame buffer. A new byte of video data flew by every microsecond, which was only one processor cycle. It took at least 10 processor cycles to iterate through a loop, so we ran the risk of never seeing the low bit change. Burrell had a way around this, though - if the loop time was relatively prime to the display frequency, it eventually had to slip into place. I wrote a 17 microsecond loop that fit the bill, and we were delighted to see it work perfectly.

Bill Atkinson had told Steve that you couldn't pull off a decent graphical user interface on the Apple II because of the weakness of its 6502 processor and the complexity of Woz's convoluted frame buffer. But, after a little more work on the software, we had a cool little GUI for the Apple II, with a proportional text word processor, in the summer of 1981. We thought that we had had a potentially valuable product on our hands.

We were reluctant to show it to Steve, knowing that he would want to commandeer it, but he heard about it from someone and demanded to see it. We showed it to him, and, unfortunately, he loved it. But he also insisted that Apple owned all the rights to it, even though we had developed it in our spare time.

Steve couldn't insist that Apple owned all of it, because Bill Budge wasn't an Apple employee at the time. But Steve could claim complete ownership of the interface card, which he said was developed with Apple resources. Burrell and I were pretty upset, because we did it on our own time and thought that we should be compensated, but it's really hard to argue with Steve, especially about money.

We ended up turning over Burrell's design to the Apple II division, but they didn't think the Apple II could deal with interrupts properly (even though we had demonstrated that it could), so they added tons of hardware and ended up with more than a dozen chips. Steve made a deal with Bill Budge that eventually resulted in MousePaint, a MacPaint clone for the Apple II which was bundled with the mouse card. Burrell and I got over it quickly, but a bit of bitterness lingered, and the whole episode whetted our appetites for eventually working on our own.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Hungarian

Author: Andy Hertzfeld

Date: January 1982

Characters: Bud Tribble, Tom Malloy, Charles Simonyi, Bill Atkinson

Topics: Software Design, Technical, Lisa

Summary: A curious style of programming



Charles Simonyi in the early days [with Xerox Alto]



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



The Macintosh used the same Motorola 68000 microprocessor as its predecessor, the Lisa, and we wanted to leverage as much code written for Lisa as we could. But most of the Lisa code was written in the Pascal programming language. Since the Macintosh had much tighter memory constraints, we needed to write most of our system-oriented code in the most efficient way possible, using the native language of the processor, 68000 assembly language. Even so, we could still use Lisa code by hand translating the Pascal into assembly language.

We directly incorporated Quickdraw, Bill Atkinson's amazing bit-mapped graphics package, since it was already written mostly in assembly language. We also used the Lisa window and menu managers, which we recoded in assembly language from Bill's original Pascal, reducing the code size by a factor of two or so. Bill's lovely Pascal code was a model of clarity, so that was relatively easy to accomplish.

The Mac lacked the memory mapping hardware prevalent in larger systems, so we needed a way to relocate memory in software to minimize fragmentation as blocks got allocated and freed. The Lisa word processor team had developed a memory manager with relocatable blocks, accessing memory blocks indirectly through "handles", so the blocks could be moved as necessary to reduce fragmentation. We decided to use it for the Macintosh, again by recoding it from Pascal to assembly language.

The primary author of the Lisa word processor and its memory manager was Tom Malloy, an original member of the Lisa team and Apple's first recruit from Xerox PARC. Tom had worked on the Bravo word processor at PARC under the leadership of Charles Simonyi, and used many of the techniques that he learned there in his Lisa code.

Even though Bud Tribble had to leave the Mac team in December 1981 in order to retain his standing in the M.D./Ph.D. program at the University of Washington, he decided that he could still do the initial implementation of the memory manager, as we were planning all along, hoping to finish it quickly after he moved back to Seattle, before classes started. He obtained a copy of the memory manager source from Tom Malloy, but he was in for a shock when he began to read the code.

The memory manager source lacked comments, which was disappointing, but the biggest obstacle was the names selected for variables and procedures: all the vowels were gone! Every identifier seemed to be an unpronounceable jumble of consonants, making it much harder to understand the code, since a variable's meaning was far from obvious. We wondered why the code was written in such an odd fashion. What happened to all of the vowels?

It turns out that Tom Malloy was greatly influenced by his mentor at Xerox, a strong-willed, eccentric programmer named Charles Simonyi. Charles was quite a character, holding many strong opinions about the best way to create software, developing and advocating a number of distinctive coding techniques, which Tom brought to the Lisa team. One of the most controversial techniques was a particular method of naming the identifiers used by a program, mandating that the beginning of each variable name be determined by the type of the variable.

However, most of the compilers in the early eighties restricted the length of variable names, usually to only 8 characters. Since the beginning of each name



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



had to include the type, there weren't enough characters left over to use a meaningful name describing the purpose of the variable. But Charles had a sort of work-around, which was to leave out all of the vowels out of the name.

The lack of vowels made programs look like they were written in some inscrutable foreign language. Since Charles Simonyi was born and raised in Hungary (defecting to the west at age 17), his coding style came to be known as "Hungarian". Tom Malloy's memory manager was an outstanding specimen of Hungarian Pascal code, with the identifiers looking like they were chosen by Superman's enemy from the 5th dimension, Mr. Mxyzptlk.

Bud decided that it would be too error prone to try to translate the Hungarian memory manager directly into assembly language. First, he made a pass through it to strip the type prefixes and restore the vowels to all the identifier names, so you could read the code without getting a headache, before adding lots of block comments to explain the purpose of various sub-components.

A few weeks later, when Bud came back to attend one of our first retreats, he brought with him a nicely coded, efficient assembly language version of the memory manager, complete with easy to read variable names, which immediately became a cornerstone of our rapidly evolving Macintosh operating system.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



- 2000 Lines Of Code

Author: Andy Hertzfeld
Date: February 1982
Characters: Bill Atkinson
Topics: Software Design, Management, Lisa
Summary: It's hard to measure progress by lines of code

In early 1982, the Lisa software team was trying to buckle down for the big push to ship the software within the next six months. Some of the managers decided that it would be a good idea to track the progress of each individual engineer in terms of the amount of code that they wrote from week to week. They devised a form that each engineer was required to submit every Friday, which included a field for the number of lines of code that were written that week.

Bill Atkinson, the author of Quickdraw and the main user interface designer, who was by far the most important Lisa implementor, thought that lines of code was a silly measure of software productivity. He thought his goal was to write as small and fast a program as possible, and that the lines of code metric only encouraged writing sloppy, bloated, broken code.

He recently was working on optimizing Quickdraw's region calculation machinery, and had completely rewritten the region engine using a simpler, more general algorithm which, after some tweaking, made region operations almost six times faster. As a by-product, the rewrite also saved around 2,000 lines of code.

He was just putting the finishing touches on the optimization when it was time to fill out the management form for the first time. When he got to the lines of code part, he thought about it for a second, and then wrote in the number:

- 2000.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



I'm not sure how the managers reacted to that, but I do know that after a couple more weeks, they stopped asking Bill to fill out the form, and he gladly complied.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION

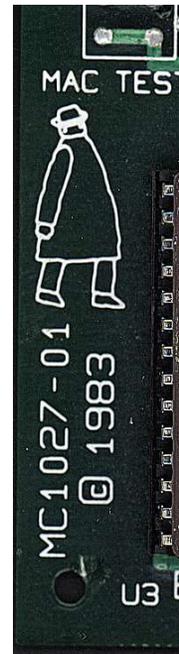


Mr. Macintosh

Author: Andy Hertzfeld
 Date: February 1982
 Characters: Steve Jobs, Andy Hertzfeld, Susan Kare, Folon
 Topics: Software Design, Personality
 Summary: Steve has a unique idea for the software



Mr. Macintosh, as rendered by Susan Kare





APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Steve Jobs often came by Texaco Towers after dinner, to see what was new, and we'd usually show him whatever recent progress we made. Sometimes he'd be pissed off about something, but other times he'd be really excited about a new idea.

I was the only one in the office one evening when he burst in, exclaiming that he had a flash of inspiration.

"Mr. Macintosh! We've got to have Mr. Macintosh!"

"Who is Mr. Macintosh?", I wondered.

"Mr. Macintosh is a mysterious little man who lives inside each Macintosh. He pops up every once in a while, when you least expect it, and then winks at you and disappears again. It will be so quick that you won't be sure if you saw him or not. We'll plant references in the manuals to the legend of Mr. Macintosh, and no one will know if he's real or not."

Engineers like myself always daydream about building surreptitious little hacks into the software, but here was the co-founder and chairman of the company suggesting something really wild. I enthusiastically pressed him for details. Where should Mr. Macintosh appear? How often? What should he do when he shows up?

"One out of every thousand or two times that you pull down a menu, instead of the normal commands, you'll get Mr. Macintosh, leaning against the wall of the menu. He'll wave at you, then quickly disappear. You'll try to get him to come back, but you won't be able to."

I loved the idea and promised that I would implement Mr. Macintosh, but not right away, since there were still so many more basic things to get done. Steve told the idea to the marketing team, and eventually recruited the French artist Folon to do some renditions of Mr. Macintosh. I also asked my high school friend Susan Kare, who hadn't started with Apple yet, to try to draw some Mr. Macintosh animations.

Most of the Macintosh system software had to be packed into a 64 KByte ROM, and ROM space got more scarce as development proceeded and the system grew. Eventually, it was clear that we'd never be able to fit bitmaps for Mr. Macintosh into the ROM, but I wasn't willing to give up on him yet.

I made the software that displayed the menus look at a special low memory location called the "MrMacHook", for an address of a routine. If the routine is present, it's called with parameters that let it draw in the menu box, and it returns a result that tells the menu manager if it did anything. Using this, an application or system module could implement Mr. Macintosh (or perhaps his evil twin) if they saw fit.

I'm not sure if anybody ever actually implemented Mr. Macintosh or used the "MrMacHook" for something worthwhile.

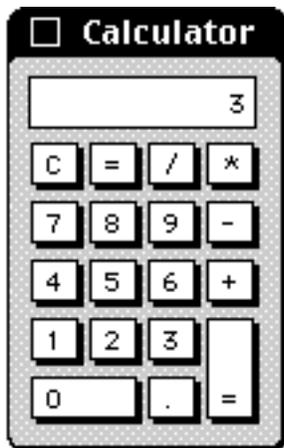


APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Calculator Construction Set

Author: Andy Hertzfeld
 Date: February 1982
 Characters: Chris Espinosa, Steve Jobs, Donn Denman
 Topics: Software Design
 Summary: Chris tries to make a Steve-approved calculator



The Calculator

Chris Espinosa was one of Apple's earliest and youngest employees, who started work for the company at the ripe age of 14. He left Apple in 1978 to go to college at UC Berkeley, but he continued to do freelance work during the school year, like writing the Apple II Reference Manual, the replacement for the legendary "Red Book".

In the summer of 1981, Steve Jobs convinced Chris to drop out of school to come work on the Mac team full time, arguing that he could go back to school anytime,



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



but there'd only be one chance to help shape the Macintosh. Chris dropped out of school to become the manager of documentation for the Macintosh, starting in August 1981.

We needed technical documentation right away, since we planned to seed third party developer in only a few months. Since the most important part of the Macintosh software was the Quickdraw graphics package, Chris decided to start with documenting Quickdraw.

Chris wanted to write a demo program using Quickdraw, in order to better understand it. He got excited about the idea of what we called "desk ornaments", which at that point were not implemented yet. He decided to work on a Quickdraw program to draw the calculator.

After playing around for a while, he came up with a calculator that he thought looked pretty good. But the acid test was showing it to Steve Jobs, in his role as our esthetic compass, to see what he thought.

We all gathered around as Chris showed the calculator to Steve and then held his breath, waiting for Steve's reaction. "Well, it's a start", Steve said, "but basically, it stinks. The background color is too dark, some lines are the wrong thickness, and the buttons are too big." Chris told Steve he'll keep changing it, until Steve thought he got it right.

So, for a couple of days, Chris would incorporate Steve's suggestions from the previous day, but Steve would continue to find new faults each time he was shown it. Finally, Chris got a flash of inspiration.

The next afternoon, instead of a new iteration of the calculator, Chris unveiled his new approach, which he called "the Steve Jobs Roll Your Own Calculator Construction Set". Every decision regarding graphical attributes of the calculator were parameterized by pull-down menus. You could select line thicknesses, button sizes, background patterns, etc.

Steve took a look at the new program, and immediately started fiddling with the parameters. After trying out alternatives for ten minutes or so, he settled on something that he liked. When I implemented the calculator UI (Donn Denman did the math semantics) for real a few months later, I used Steve's design, and it remained the standard calculator on the Macintosh for many years, all the way up through OS 9.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Rosin g' s Rascal s

Author: Andy Hertzfeld
Date: March 1982
Characters: Bill Atkinson, Wayne Rosin g, Dan Smith, Frank Ludolph,
Steve Jobs, Bruce Horn
Topics: Software Design, Lisa, User Interface, Management
Summary: The Lisa Filer was radically redesigned with no time to spare

By the spring of 1982, the Lisa User Interface was finally settling down, and the software team was working feverishly to get everything ready to ship by their deadline in the fall. Most of the applications were shaping up, although myriad problems remained, and the team could finally sense a glimmer of light at the end of the long tunnel.

Dan Smith and Frank Ludolph were working on the Lisa Filer [Desktop Manager], the key application that managed files and launched other applications. It was beginning to come together, but Dan was still unsatisfied with the current design.

The Filer was based on a dialog window that prompted the user to select a document from a list, and then select an action like "Open", "Copy" or "Discard", and then answer more questions, depending on the selected action. There was so much prompting that it became known as the "Twenty Questions Filer". Dan thought that it wasn't easy or enjoyable to use, but there just wasn't enough time left in the schedule for further experimentation, so they were pretty much stuck with it.

One afternoon, Dan mentioned his dissatisfaction to Bill Atkinson, the main designer of the Lisa User Interface. Bill suggested that they meet that evening at his home in Los Gatos for a brain-storming session to see if they could come up with a better design, even though it was probably too late to use it for the initial release.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Bill favored a more graphical approach, and wanted to use small graphical images to represent files, which could be manipulated by dragging them with a mouse. He remembered an interesting prototype that he saw at M.I.T. called Dataland, where data objects could be spatially positioned over a large area. He adapted the idea for Lisa, allowing icons representing files and directories to be positioned on a scrolling, semi-infinite plane.

After a couple of nights of fiddling around, Dan and Bill had an interesting mock-up going, with icons representing documents and folders, complete with a trash can with flies buzzing around it. The icons used a mask bitmap to define their borders, so irregular shapes could be rendered seamlessly on the gray desktop. The new design seemed to have the simplicity and elegance that they were striving for, so they began to get excited.

They were both eager and afraid to show the mock-up to the rest of the team, since the design of the Filer was supposed to be frozen, and embarking on such a major revision would surely slip the schedule, which was already precariously close to unrealizable. They gathered up their courage and approached Wayne Rosing, the Lisa Engineering Manager, and explained their dilemma.

Wayne appreciated the potential of the new approach, but wasn't ready to slip the schedule to accommodate it. He thought it was perhaps barely possible to go with both the new design and the current schedule, if they could turn the mock-up into a solid working prototype in record time. He proposed a deal: he gave them permission to work on the new design in secret for the next two weeks. If they had a robust, stable prototype by then, he promised to support it. If they didn't, Bill and Dan promised to forget it and work to finish the earlier design.

Wayne extracted one additional promise from Bill: under no circumstances was he to show the mock-up to Steve Jobs. Wayne knew that Steve would have a strong reaction and would probably wreak havoc with the schedule accordingly. He didn't want Steve to see it until they knew if they could pull it off.

Bill was used to showing off his latest advances to the Mac team, and this new, icon-based approach to file management was a particularly important one. Bruce Horn had started working on the Mac team the previous month, and he was already starting to develop our file manager, which Bud had christened "The Finder". Bruce had similar ideas about spacial filing, and he and I had created a prototype called the "micro-finder" which represented files as tabs that were spacially organized on a picture of a diskette. Bill thought it was important for us to see the new direction as soon as possible, so he left us a copy of his prototype, under strict instructions not to show it to Steve.

We had a few close calls over the next couple of weeks as we played with the prototype, frantically quitting it when we heard Steve approaching. Finally, on the last day before the deadline expired, we must have cut it too close, because Steve knew that we were hiding something from him. We explained our promise to Bill, but Steve still demanded to see it, so we had to show it to him. He immediately fell in love with it, and ran off to talk to Bill and Wayne about it, just as we feared.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Luckily, the development had gone well the last two weeks, and Wayne was ready to commit to the new approach and unveil it to the entire team. He called an all-hands meeting, to which Bill, Dan, Frank and Wayne wore newly minted T-Shirts labeled "Rosing's Rascals". Wayne explained the surreptitious nature of the two week effort to the team while Bill set up the demo. Rosing's rascals had pulled it off, endowing the Lisa with a much more intuitive file manager that quickly became a hallmark of Apple's new user interface.

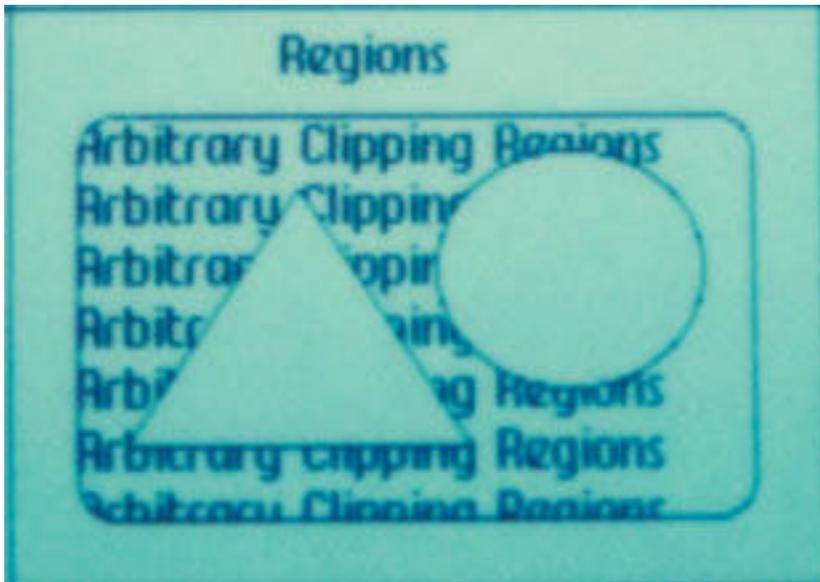


APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



I Still Remember Regions

Author: Andy Hertzfeld
 Date: April 1982
 Characters: Bill Atkinson, Steve Jobs
 Topics: Software Design, QuickDraw
 Summary: We almost lose Bill in a car accident



Regions were the heart of QuickDraw

The single most significant component of the original Macintosh technology was QuickDraw, the graphics package written by Bill Atkinson for the Lisa project, which pushed pixels around the frame buffer at blinding speeds to create the celebrated user interface. One of QuickDraw's main jobs was to provide the



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



primitives for quickly drawing text and graphics into overlapping windows, when the window that you're drawing into may be partially obscured by other windows. Applications could just draw without worrying if their window was obstructed because QuickDraw, with a little help from the window manager, would take care of the clipping to make sure pixels stayed inside in their window.

Overlapping windows can form complex shapes, especially if their corners are rounded. The key data structure in QuickDraw was called a "region", which compactly represented an area of the screen. QuickDraw provided routines that allowed the programmer to define regions by using the built-in drawing primitives, and to perform operations with them like union and intersection. Most importantly, all of the QuickDraw drawing primitives clipped to the intersection of three different regions, to allow drawing into obscured windows. We considered QuickDraw's speed and deftness at region handling to be the most significant "crown jewel" in Apple's entire arsenal.

The region data structure was a variable-sized list of what Bill called "inversion points", the coordinate values where black changed to white or vice versa. Since most regions were mostly rectangular, there weren't many inversion points, so regions were quite compact. But occasionally, there were lots of inversions, like in a circle, so regions grew as necessary.

QuickDraw was written entirely by Bill Atkinson, and in the spring of 1982 it was still evolving. He had recently sped up region operations by more than a factor of four. The concept of "pictures", a set of drawing operations grouped together for easy playback, was just added to the package, and hadn't really settled down yet. At this point, the Lisa applications were beginning to come together and Bill was changing QuickDraw in response to what they needed.

One morning, we were shocked to hear that Bill had gotten into a really bad car accident on his way into work. Apparently, he had turned a corner and not seen a parked truck, and slammed his little Corvette into the truck, shearing the roof off the top of his car. Bill was knocked unconscious and got pretty banged up, although he was still in one piece - one of the police officers who survey the wrecked Corvette commented that it was a miracle that Bill wasn't decapitated. This was a little more than a year after Woz's plane crash, but it brought back memories of that.

When Steve Jobs heard about the accident, he immediately jumped into his car and drove to the hospital where they had taken Bill. He was in a hospital bed, and had only recently regained consciousness by the time Steve arrived. He sustained a head injury and lost some blood, but luckily there were no major problems.

Steve entered the hospital room and was relieved to see that Bill had regained consciousness. "Is everything OK?", Steve asked. "We were pretty worried about you."

Bill turned his head and looked at Steve. He managed a painful smile. "Don't worry, Steve, I still remember regions."

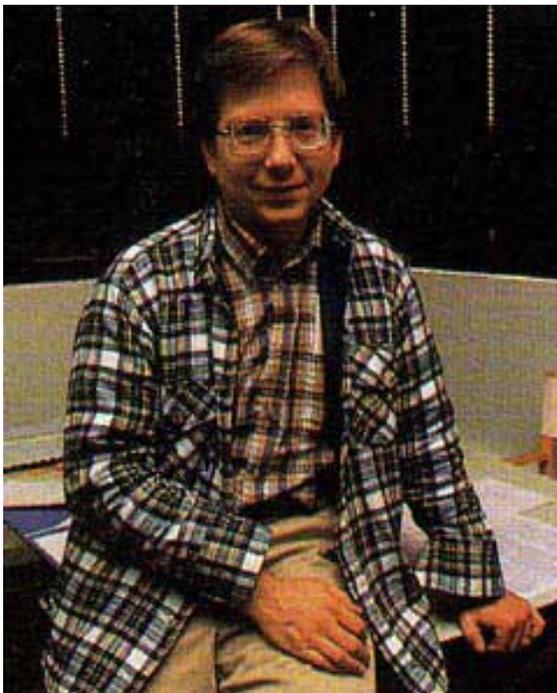


APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



And Then He Discovered Loops!

Author: Andy Hertzfeld
Date: April 1982
Characters: Bob Belleville, Rich Williams
Topics: Software Design, Management
Summary: Bob has written many lines of code



Bob Belleville in 1983



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



We interviewed quite a few candidates to replace Bud Tribble as the software manager before encountering Bob Belleville, who was one of the main hardware designers of the Xerox Star, the first commercial computer with a graphical user interface. He was intelligent and soft-spoken, and dryly skeptical about human nature. One of his many aphorisms was "The Law of Conservation of Misery" (no matter what course of action is taken, the total human misery in any given situation is maintained), which seemed particularly applicable to large computer companies.

Bob's background looked to be a lot stronger in hardware than software, so we were somewhat skeptical about his software expertise, but he claimed to be equally adept at both. His latest project was a rebellious, skunk-works type effort to make a low cost version of the Star called "Cub" that used an ordinary Intel microprocessor (the 8086), which was heresy to the PARC orthodoxy, who felt that you needed custom, bit-slice processors to get sufficient performance for a Star-type machine. Bob had written much of the software for Cub himself.

"I've got lots of software experience", he declared, "in fact I've personally written over 350,000 lines of code."

I thought that was pretty impressive, although I wondered how it was calculated. I couldn't begin to honestly estimate how much code I have written, since there are too many different ways to construe things.

That evening, I went out to dinner with my friend Rich Williams, who started at Apple around the same time that I did. Rich had a great sense of humor. I told him about the interview that I did in the afternoon, and how Bob Belleville claimed to have written over 350,000 lines of code.

"Well, I bet he did", said Rich, "but then he discovered loops!"



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Do It

Author: Andy Hertzfeld
Date: June 1982
Characters: Larry Tesler
Topics: User Interface, Software Design, Lisa
Summary: User testing sometimes has surprising results

Many of the academic types who were involved in creating the earliest implementations of the graphical user interface at Xerox PARC and various universities sort of sneered at the first generation of personal computers when they appeared in the mid-seventies, since the early personal computers were much less powerful than the machines that they were used to programming. There wasn't that much you could do with only four kilobytes of memory and no disk drive.

But Larry Tesler, who was a key member of the Smalltalk team in the Learning Research Group at Xerox PARC, felt differently. He was really excited by the potential of personal computers, buying a Commodore PET as soon as one became available in 1977. He was one of the demonstrators at Apple's famous Xerox PARC visit in December 1979, and he was so impressed by the Apple visitors that he quit PARC and started working at Apple in the summer of 1980, as the manager of the Lisa Applications team.

Larry championed consistency between applications, and made many contributions to what eventually became the Macintosh User Interface. He was also the leading advocate and implementor at Apple of user testing: actually trying out our software out on real users and seeing what happened. Starting in the summer of 1981, Larry organized a series of user tests of the nascent Lisa software, recruiting friends and family to try out the software for the first time, while being observed by the Apple designers who recorded their reactions.

The user tests were conducted in a specially constructed room featuring a one-way mirror, so observers could watch the tests without being intrusive. The



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



tests were conducted by a moderator who made sure the user felt comfortable and showed her the basics of using a mouse. Then, with no further instruction, users were asked to perform specific tasks, without help from the moderator, like editing some text and saving it. The moderator encouraged each user to mumble under her breath while doing the tasks, revealing her current thinking as much as possible. Each session was audio or videotaped for later analysis.

When the software required confirmation from the user, it displayed a small window called a "dialog box", that contained a question, and presented two buttons, for positive or negative confirmation. The buttons were labeled "Do It" and "Cancel". The designers observed that a few users seemed to stumble at the point that the dialog was displayed, clicking "Cancel" when they should have clicked "Do It", but it wasn't clear what they were having trouble with.

Finally, the team noticed one user that was particularly flummoxed by the dialog box, who even seemed to be getting a bit angry. The moderator interrupted the test and asked him what the problem was. He replied, "I'm not a dolt, why is the software calling me a dolt?"

It turns out he wasn't noticing the space between the 'o' and the 'I' in 'Do It'; in the sans-serif system font we were using, a capital 'I' looked very much like a lower case 'l', so he was reading 'Do It' as 'Dolt' and was therefore kind of offended.

After a bit of consideration, we switched the positive confirmation button label to 'OK' (which was initially avoided, because we thought it was too colloquial), and from that point on people seemed to have fewer problems.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Resource Manager Countdown

Author: Andy Hertzfeld
Date: August 1982
Characters: Bruce Horn, Bob Belleville
Topics: Software Design, Management
Summary: Bob gives Bruce thirty days to finish the Resource Manager



Bruce used one note for each remaining day

The Resource Manager was one of the cornerstones of the Macintosh Toolbox. It provided a way to manage chunks of data so they could be easily accessed by the code but be stored and edited independently from it. This was the basis of our localization, for example, since text strings could be stored as resources and translated without having to change the code and rebuild the entire application.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Resources were conceived and implemented by Bruce Horn. They came a little late in the design cycle, and we had to rewrite some other parts of the toolbox to take advantage of them, although that wasn't clear initially. But in the summer of 1982 they were definitely one of the riskier parts of the design.

Bob Belleville, our new software manager, assessed the situation and decided that the resource manager was too big a risk, and tried to convince us to drop it from the design. Bruce went apoplectic, and Bob eventually compromised on giving Bruce 30 days to implement the resource manager. Bruce agreed that if it wasn't ready within 30 days, it would be dropped from the project.

The next day, Bruce came to work with a few pads of yellow sticky notes. He tore off one note at a time, wrote a number on it, and then stuck it on the wall of the double-sized cubicle that I shared with him. Soon the cubicle was encircled with little yellow notes, each with a number from 1 to 30.

"This shows me how many days that I have left", Bruce explained. "I'll tear one down each day, and it will remind me of how much time is remaining."

For the next month, the first thing that Bruce did every morning was dramatically tear off the note with the lowest remaining number. As the days passed, he was making good progress but it wasn't at all clear that he was going to make it.

When there were only three notes left, I began to wonder what was going to happen if Bruce missed the deadline. The resource manager was mostly working now, but he didn't seem to be on a path to closure, as he continued to refine the design.

Finally, there was only one note left on the wall. But when Bruce came in that morning, he had a new pad of yellow notes, and instead of taking the last one down, he wrote another ten numbers and posted them up, before taking down the previously final note. I started to laugh.

"It's overtime", he explained. "I swear I'll really be done in ten more days."

I was waiting for Bob Belleville to confront him, and tell him that he missed the deadline, but Bob wisely saw that the resource manager was far enough along that it could stay. And Bruce really did finish in the next ten days, kind of.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



```

*****
Resource Manager for the Macintosh User Interface Standard
:
:
: Written by Bruce Horn          22 November 1982
:
: Copyright 1982 by Apple Computer, Inc. All Rights Reserved.
:
: This file contains the interface specification for the Resource Manager.
: It includes files which contain toolbox, graftype and zsysmacs equates
: and macro definitions, as well as the actual code for the implementation
: of the Resource Manager.
:
: 16-Dec-82    BLH    Added Add/RmveResource. Massive name changing.
: 21-Jan-83    BLH    Updated to new LK file system.
: 11-Feb-83    BLH    Returned error codes in InitResources,
:                    OpenResource
: 17-Feb-83    BLH    Fixed bug in CheckLoad
: 25-Apr-83    BLH    Rom 3.0 overhaul:
:                    fixed bug in UpdateResFile, optimizing
:                    the writes and making it a little quicker
:                    to call updates.
:
:                    Changed openResFile to use the resource
:                    fork, and to handle more reasonably the
:                    already-open, not-open, and other error
:                    conditions.
:
:                    CloseResFile now releases all resources
:                    in the map before closing, compacts the
:                    resources in the map, and writes out the
:                    possibly modified map and header.
:                    (@@Note--CloseResFile does not compact
:                    yet. Due to a bug in GetHandleSize,
:                    this feature has been disabled.)
:
:                    The resource file starts at ZERO instead
:                    of 512.
:
:                    Added memory full error handling in
:                    OpenResFile and CheckLoad.
:
:                    Added preLoad capability to initResources
:                    and OpenResource.
:
:                    Added procedure WriteResource.
:
:                    Added ResAdded resource attribute.
:
:                    Added MAttr, map attributes word, in map.
:
:                    MapSize and DataSize are now LONGS.
:
:                    Resource manager routines now save
:                    ALL registers except A0 and D0.
:
:                    Added SetResPurge to handle the purge
:                    hook in the Memory Manager.
:
:                    Added CurResFile.
:
: 12-May-83    AJH    fixed bug in InitResources -- shouldn't preload when
:                    you can't read the map
:
: 24-May-83    BLH    Fixed bug in NewMap--crashes if failed due to unlink bug.
: 25-May-83    BLH    Changed interface to CmpString in GetNamedResource.
: 31-May-83    BLH    Added CreateResFile, fixed AddResource bugs.
: 1-Jun-83     BLH    Added DetachResource.
: 6-Jun-83     BLH    Fixed bug in GetNamedResource (didn't search all maps)
: 8-Jun-83     BLH    Fixed bug in CloseResource (!)
:
: 10-Jun-83    BLH    Moved compacting and mapwrite from CloseResFile to Update.
:                    Added routine to close maps in RsrcZoneInit.
:                    CloseResFile closes all maps if you attempt to close
:                    System.Rsrc. UpdateResFile now compacts, since GetHandleSize
:                    finally works.
:
: 13-Jun-83    BLH    Made SetResAttr, SetResInfo NOT set the mapChanged bit.
:                    you have to explicitly call ChangedResData to make sure
:                    the change is written out to the map.
:
: 18-Jul-83    BLH    Added MapCompact attribute bit to minimize resource file
:                    compacting. Added UniqueID to return a unique ID for a given
:                    type in the current map. Also, shrunk code approximately
:                    60 bytes due to suggestions from Donn Denman.
:
: 8-Aug-83     BLH    Removed BeginSubResource, EndSubResource. Added HomeResFile.
:                    Added ResErrProc in globals. Made UniqueID return an
:                    ID which is unique in all open maps. Took out SetEOF
:                    from UpdateResFile, to allow pre-allocation of the
:                    resource fork. Put in ResrvMem to reserve low locations for
:                    locked handles. Added file-order preload and update to
:                    speed up resource I/O.
:
: 12-Aug-83    BLH    Split Resource Manager into four files: RMGr, RMGrAsm1,

```



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



```

: RMGrAsm2, and RMGrAsm3. Utilities in RMGrAsm2. Added
: CheckGrow to handle the DiskFull condition. CheckGrow
: called from ChangedResource, AddResource, and AddReference.
: Also Added ReallocRes, AllocRes, called from (surprise!)
: changedResource and AddResource to allocate space for
: either a new resource or a resource which has changed
: size and needs to be reallocated at the end.
:
: 16-Aug-83 BLH Bug fixes and space-savers.
:
: 29-Aug-83 BLH Fixed bug in SetResInfo, RmveResource (protect bit checked
: in RHnd1 instead of RAttr) and made CheckGrow handle
: memoryFull condition as well as diskFull.
: AddResource/AddReference now return ResErr=DiskFull or
: MemFull in the case that CheckGrow failed.
:
: 1-Sep-83 BLH InitResources shouldn't clear ResErrProc.
: Fixed NOP (branch to next instr!), folded com code in
: name stuff. Fixed dangling ptr problem (!) caused by
: CheckGrow. Took out dangerous LINK stuff in SetFileIO
: and just used a standard stackframe on first link in
: StdEntry. Made GetCurMap and CloseResFile handle the case
: of no maps open (set curMap to FFxx with ST). Fixed bug
: in LoadResource where errors would not get reported through
: the ResErrProc. AddResource sets the Resource bit in the
: masterPtr, RmveResource and DetachResource clear it.
:
: 2-Sep-83 BLH MAJOR CHANGE: throw away name ID stuff, made hard offsets
: to the names in the name block stored in the resource entry
: where the nameID was previously stored. This change
: fixed a time-bomb bug which would trash the resource file
: after repeated SetResInfo calls. Also, data lengths in
: the resource file are now LONG integers (4 bytes). Fixed
: CountTypes, GetIndType to return true counts and unique types
: for all open maps, to be consistent with CountResources
: and GetIndResource.
:
: 6-Sep-83 BLH Added MapReadOnly in MAttr to inhibit writing to the resource
: file and diskFull checking. Added SetResFileAttrs,
: GetResFileAttrs to allow toggling of this bit.
:
: 8-Sep-83 BLH Fixed bug in CheckGrow which could trash the resource map
: if the handle passed to ChangedResource or AddResource
: was purged.
:
: 9-Sep-83 BLH Fixed bug in CheckLoad--ResrvMem must preserve A0, so typed
: in Word $A040 to replace _ResrvMem. Fixed bug in RmveName.
:
: 10-Sep-83 BLH Fixed bug having to do with protected resources; cleaned
: up usage in UpdateResFile, ChangedResource, WriteResource.
:
: *****
: Version 7.0. Last modified 10-Sep-83 by Bruce.
: *****
:
: .Include TIAz: SysErr.Text ;System error constants
: .Include TIAz: SysEqu.Text ;System equates (memory layout...)
: .Include TIAz: GrafEqu.Text ;Basic graphics and screen driver
: .Include TIAz: GrafTypes.Text ;Offsets into LisaGraf structures
: .Include TIAz: QuickMacs.Text ;LisaGraf traps
: .Include TIAz: SysMacs.Text ;System macros, line1010 calls
: .Include TIAz: ToolMacs.Text ;ToolBox References
: .Include TIAz: ToolEqu.Text ;Toolbox equates
:
: .Proc RMGr, 0
:
: .Def InitResources ;One-time call, initializes resource manager
: ;and opens System.Resource ('refnum)
:
: .Def RsrcZoneInit ;Zero all handles in the system map
: ;which point to application heap zone
:
: .Def CreateResFile ;Given filename, create a null res file
: .Def OpenResFile ;Given filename, opens and returns refnum
: .Def UseResFile ;Given refnum, sets current res file.
:
: .Def GetResFileAttrs ;Given refnum, return resfile attributes.
: .Def SetResFileAttrs ;Given refnum, set resfile attributes.
:
: .Def UpdateResFile ;Given refnum, update resource file
: .Def CloseResFile ;Given refnum, closes the resource
:
: .Def SetResPurge ;Write resources when purging?
: .Def SetResLoad ;Auto-Load on get?
:
: .Def CountResources ;Given type, return number of resources
: .Def GetIndResource ;Given index and type, returns handle
:
: .Def CountTypes ;Returns number of types in current rsrc
: .Def GetIndType ;Given index, returns type
:
: .Def UniqueID ;Given type, return a unique ID for that type
:
: .Def GetResource ;Given an ID and type, give handle back

```



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Sound By Monday

Author: Andy Hertzfeld
Date: September 1982
Characters: Burrell Smith, Andy Hertzfeld, Steve Jobs
Topics: Hardware Design, Software Design, Management, Technical
Summary: Steve threatens to remove Mac's sound capability



Cartoon from IEEE Spectrum



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



When Burrell redesigned the Macintosh digital board in August 1982 after the Integrated Burrell Machine effort fell apart, one of the most significant improvements involved the sound generation capability. All of the extra logic available in the IBM chip allowed Burrell to implement four simultaneous channels of sound, each using a custom wavetable. That was too expensive to keep without the custom chip, but he was able to maintain the fundamental capability, DMA-fetched sound using 8 bits per sample and a clever, pulse width modulated digital to analog converter.

Burrell figured that the Mac could still have the four simultaneous voices that he had envisioned for the Integrated Burrell Machine, only now we'll do most of the work in software instead of hardware. So he promised Steve that the redesign would feature great sound, including a four voice synthesizer. Steve approved spending an extra dollar on an improved amplifier, to better match the improved sound generation capability.

But there were an enormous number of different things to work on in the summer of 1982, including getting the Mac Toolbox into good enough shape for developers to start writing applications. Burrell bugged me about writing a sound driver, and especially his cherished "four voice engine", but I wasn't able to get around to it as a few weeks passed.

Steve Jobs started to get impatient about getting to hear our new sound capability for himself. Finally, he pulled Burrell and myself aside late on a Friday afternoon.

"You told me that the new sound would be really great, right?", he asked Burrell. "Well, if I don't hear great sound coming out of that prototype by Monday morning, we're going to remove the amplifier. "

Then he looked at me. "You've had long enough to get the sound going. I want to hear great sound on Monday, or else." Then he stormed off, leaving Burrell and me to figure out what to do.

"I think he's bluffing", I told Burrell. "But what if he's not?"

Burrell didn't seem very upset, in fact he seemed kind of excited. I think he was pleased that Steve was on our case about the sound, since he really wanted me to write the sound routines as soon as possible anyway. He promised to buy me meals all weekend while we stayed at Apple to get the sound going.

I had already written a basic diagnostic that generated a simple square wave, which certainly didn't meet anyone's criteria for great sound, but at least it told us the hardware was working, and worked out the basic details of controlling the hardware. Burrell wanted me to get to work right away on the four voice synthesizer, since he wasn't sure we could actually make that work.

As usual, Burrell's new design was very clever. The Macintosh was already continuously fetching data from memory to drive the video display, interleaving memory bandwidth between the display and processor in a similar fashion to the Apple II. But every 44 microseconds, there was a "horizontal blanking interval" where no video data was needed, so Burrell used that time to fetch data for the sound. That gave us a sample rate of 22kHz, which would allow us to do frequencies up to 11kHz, which isn't too bad.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



The sound driver worked at the interrupt level, so sound generation could proceed in the background while doing some other task. It arranged to receive control at the beginning of the vertical blanking interval, which occurred every 16 milliseconds. It needed to generate all the sound data for the next 16 milliseconds, which worked out to 44 microseconds for every successive sample. If our calculation took 22 microseconds per sample, for example, the sound generation would be soaking up half of the available processor cycles.

It only took a few hours to write a driver with a simple sound generation loop. It could do two voices fine, but it didn't run nearly fast enough to do four - it took too long to generate each sample, which caused audible glitches as well as making everything else run like molasses. Burrell took a look at my code, and saw that I was using some memory locations during the sound calculation.

"Memory? Are you kidding? You can't hit main memory, you'll never make it that way. You've got to do everything in the registers!"

"Registers" are special locations that are part of the processor chip itself, where the action really happens. They could be accessed four times faster than the bulk of memory, which was in separate memory chips. The problem was that there were only 16 registers. For each voice of sound, we needed a frequency, a waveform pointer, a position within the waveform, and an amplitude, plus some housekeeping data. There weren't enough registers to do four voices.

I was able to rewrite the routine without touching main memory, but I was only able to get three voices since I ran out of registers. But that wasn't good enough for Burrell. By now it was late on Saturday evening, and I wanted to go home, but he felt that we had to get the fourth voice done before "head hits pillow" as he liked to say, or we'll never get the fourth voice.

Finally, I was able to leverage the fact that the registers were 32 bits long and we only were doing 16 bit calculations in some of them, to use some of the them to hold two different values. Each sample took about 22 microseconds to calculate, so we were using roughly half of the CPU to get the four voices at the maximum sample rate. The basic four voice capability was implemented, but we still needed an impressive demo to show it off. We went home to sleep at around midnight, after agreeing to come back around noon to work on the demo.

The next day, we decided to write a demo called "SoundLab" that would let the user control the pitch and waveform of the four independent voices. You could specify or edit a waveform by drawing it with the mouse, and control the frequency of each voice with a scrollbar. The results didn't sound like music, because there was no envelope shaping, but you could make very eerie sounding noises, which we deemed impressive enough. And it was fun to be able to hook an oscilloscope up to the sound output, and then draw a waveform with the mouse and see it on the scope.

When Steve came in on Monday, he was pleased that we could demonstrate the four voice capability, and impressed that he could edit a waveform with the mouse and see it on the scope, but I don't think he was satisfied, since he wanted high quality music. There was a lot of potential in the Mac sound capability, but it would still take years, and the efforts of many third party developers, to fully exploit it.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Boot Beep

Author: Andy Hertzfeld
Date: September 1982
Characters: Charlie Kellner, Andy Hertzfeld, Terry Oyama, Steve Jobs
Topics: Software Design
Summary: How the boot sound evolved

When you powered up an Apple II, it would make a short beep sound to let you know that it was alive. We thought that the Mac should do something similar, once it passed the diagnostics, sort of like an infant's first cry, letting the world know that you actually made it here.

The 1981 Macintosh just had a square wave sound generator, where the software controlled the frequency by loading a value into the VIA's timer. I wrote a boot sound routine that gradually incremented the frequency at an accelerating pace, so it had a whooping quality to it that was almost humorous. People generally liked it, but we knew that we'd have to do something better for the real product.

In August 1982, the Mac was redesigned with much better sound quality, so we had the possibility of a better boot sound, since we now had 8-bit samples to play with. I started experimenting a little bit, to see if I could come up with something.

Around this time, Charlie Kellner decided to transfer to the Mac group from the Apple II group. Charlie was a brilliant Apple II programmer (who wrote a classic hi-res bowling game for the Apple II before he started work at Apple), who was a multi-talented, meticulous perfectionist. He had designed a music synthesizer for the Apple II, called the alphaSyntauri, as a side project that was the basis of a small start-up company. For some reason, he got bored with the Apple II and wanted to try out working with the Mac team.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Charlie saw me messing around with sounds for a new boot beep and told me that he knew of a simple algorithm that might work pretty well. He asked me to fill the sound buffer with a simple square wave, but then make successive passes on it, averaging adjacent samples until everything reached the same level.

I coded it up and we tried it, and sure enough it had a pleasant, distinctive chiming quality. With a little bit of tweaking, it became the famous sound that the Mac made when it powered up, which lasted until the Mac II, which had even better sound capability, in 1987.

Charlie was pleased that he was able to make a significant contribution in his first week on the project. Inspired, he asked if he could take a prototype home over the weekend for testing. The next Monday he came into work very excited.

"I knew that something wasn't right!", he exclaimed. "The sound is being completely muffled by the case!. But I know how to fix it."

He had done a series of experiments with the Mac that he had taken home over the weekend, and found that the Macintosh's case was baffling and distorting the sound. He even printed out graphics showing the results of his measurements. Then, after analyzing the data, he drilled a hole about the size of a dime in a strategic place, which caused the measurements to improve dramatically.

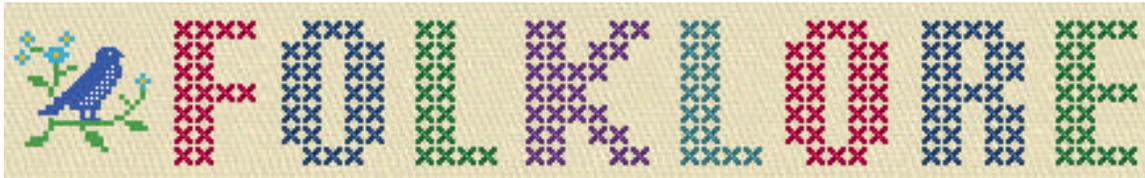
He started demoing his modified prototype, showing how the hole improved the sound quality. The difference didn't sound that significant to me, but it definitely was an improvement. He showed it to Terry Oyama, who designed the case, and asked him if he could add the hole.

The next day, Steve Jobs came by in the afternoon and asked to hear Charlie's demo. He listened to the two Macs, and then decreed "There's not enough improvement! There's no way that we're going to put an ugly hole in the case! Just forget about it!"

Charlie was pretty disappointed, and never got very enthusiastic about the Mac after that. A couple of weeks later, he transferred back to the Apple II group, leaving the boot beep as his only legacy.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Bouncing Peppis

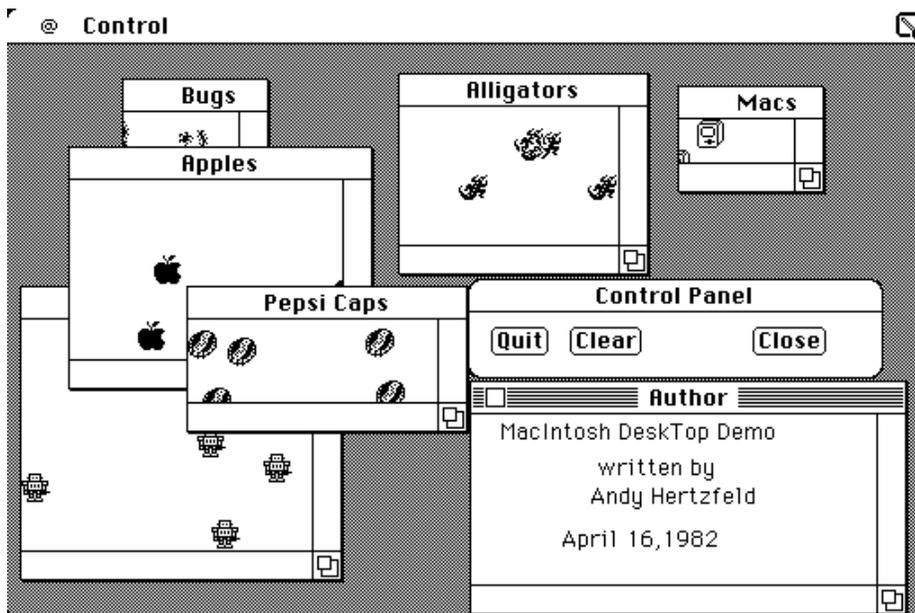
Author: Andy Hertzfeld

Date: March 1983

Characters: Andy Hertzfeld, Steve Jobs, John Sculley, Susan Kare, Bill Atkinson

Topics: Software Design, recruiting, Early Programs

Summary: We cook up a special demo for John Sculley



We added Pepsi Icons for John Sculley's visit

The Window manager was the one of the most important parts of the User Interface toolbox, and the ultimate showcase for Quickdraw's "region clipping" technology.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



The window manager had to calculate various regions for windows as they were created, moved and resized, and objects drawn inside the windows were automatically clipped as necessary.

The Macintosh window manager was based on the one that Bill Atkinson wrote for Lisa, which was written in Pascal; my job was to rewrite it in 68000 assembly language and adapt it to the Macintosh environment. The first step was to port Bill's Pascal version. I wrote a little program to test the port, which I called "Window Manager Demo", that generated some windows and put the window manager through its paces.

A year earlier, I had written a fast "ball bouncing" routine, using custom, 16x16 pixel graphics routines, that could animate hundreds of balls simultaneously, which was a fun way to show off the Mac's raw graphical horsepower. I decided to animate a few dozen balls in each window in the window manager demo, using Quickdraw, because their continuous movement would eventually cover all the bases inside a window and expose any flaws in the underlying clipping.

After Susan started in January 1983, I asked her to draw some tiny 16 by 16 bitmaps to use in the Window Manager Demo instead of the by now monotonous ball shapes. Soon, we had a variety of little objects bouncing around in the various windows, like tiny little Macintoshes, or apples, insects and alligators. I thought that the Window Manager Demo was finished, but I was wrong.

Steve Jobs came by the software area one evening a couple of months later, excited about someone he had recently met in New York City. "Hey, I want you to do a demo next week for this guy I met yesterday, John Sculley, he's the president of Pepsi. He's really smart - you wouldn't believe how smart he is. If we impress him, we can get Pepsi to buy thousands of Macs. Maybe even five thousand. Why don't you try to come up with something special to show him?".

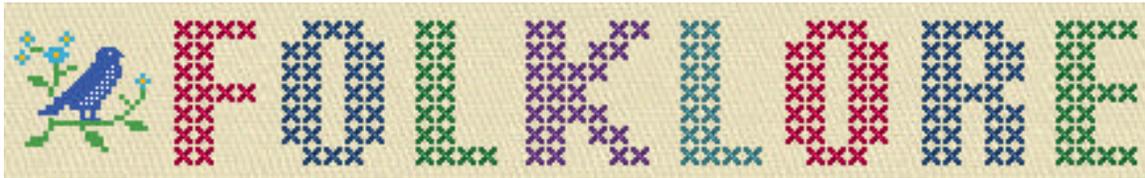
It sounded a little bit fishy to me, since we hardly ever demoed to potential customers at that point. But I asked Susan to draw some Pepsi imagery, and she came up with tiny little Pepsi caps, as well as Pepsi cans, in his honor, so I put them into the Window Manager demo.

The next week, Mike Murray led John Sculley around the engineering area, since Steve was out of town. He brought him by my cubicle to see the modified Window Manager demo. I opened the windows one at a time, saving the Pepsi caps and cans for last. He seemed genuinely excited to see the Pepsi stuff, but oddly cold for most of the demo.

A few weeks later, we found out the real story - the purpose of John's visit was to interview for CEO of Apple, and he got the job, being convinced by Steve's famous line "Would you rather sell sugar water to kids for the rest of your life, or would you like to change the world?".



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Saving Lives

Author: Andy Hertzfeld
Date: August 1983
Characters: Steve Jobs, Larry Kenyon
Topics: Software Design, Inspiration
Summary: Steve wants us to make the Macintosh boot faster

We always thought of the Macintosh as a fast computer, since its 68000 microprocessor was effectively 10 times faster than an Apple II, but our Achilles heel was the floppy disk. We had limited RAM, so it was often necessary to load data from the floppy, but there we were no faster than an Apple II. Once we had real applications going, it was clear the floppy disk was going to be a significant bottleneck.

One of the things that bothered Steve Jobs the most was the time that it took to boot when the Mac was first powered on. It could take a couple of minutes, or even more, to test memory, initialize the operating system, and load the Finder. One afternoon, Steve came up with an original way to motivate us to make it faster.

Larry Kenyon was the engineer working on the disk driver and file system. Steve came into his cubicle and started to exhort him. "The Macintosh boots too slowly. You've got to make it faster!"

Larry started to explain about some of the places where he thought that he could improve things, but Steve wasn't interested. He continued, "You know, I've been thinking about it. How many people are going to be using the Macintosh? A million? No, more than that. In a few years, I bet five million people will be booting up their Macintoshes at least once a day."

"Well, let's say you can shave 10 seconds off of the boot time. Multiply that by five million users and that's 50 million seconds, every single day. Over a year,



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



that's probably dozens of lifetimes. So if you make it boot ten seconds faster, you've saved a dozen lives. That's really worth it, don't you think?"

We were pretty motivated to make the software go as fast as we could anyway, so I'm not sure if this pitch had much effect, but we thought it was pretty humorous, and we did manage to shave more than ten seconds off the boot time over the next couple of months.

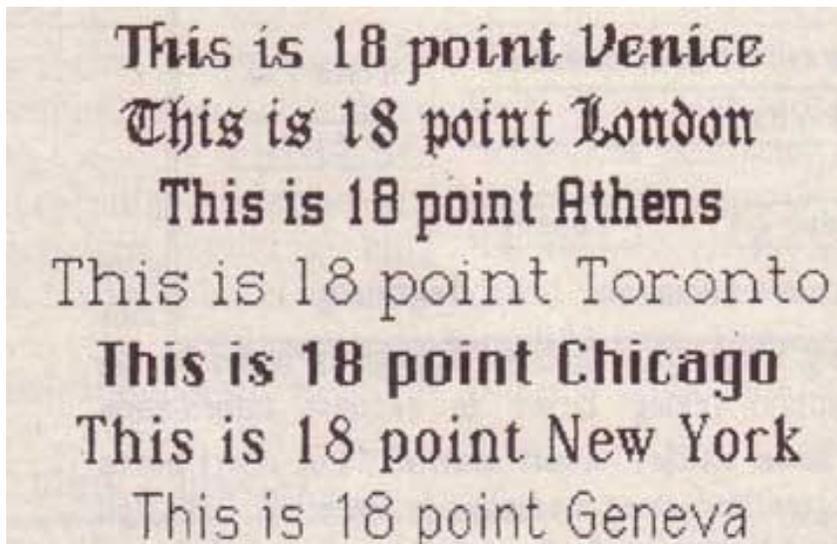


APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



World Class Cities

Author: Susan Kare
 Date: August 1983
 Characters: Steve Jobs, Bill Atkinson, Andy Hertzfeld, Susan Kare
 Topics: Software Design, Personality
 Summary: We needed to come up with a naming scheme for our fonts



Here's a sampler of some original Mac fonts

Landing in the Macintosh group as a bitmap graphic designer was a lucky break for me, and one interesting part of the job was designing screen fonts. It was especially enjoyable because the Macintosh was able to display proportional typefaces, leaving behind the tyranny of monospace alphabets with their narrow m's and wide i's.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



The first Macintosh font was designed to be a bold system font with no jagged diagonals, and was originally called "Elefont". There were going to be lots of fonts, so we were looking for a set of attractive, related names. Andy Hertzfeld and I had met in high school in suburban Philadelphia, so we started naming the other fonts after stops on the Paoli Local commuter train: Overbrook, Merion, Ardmore, and Rosemont. (Ransom was the only one that broke that convention; it was a font of mismatched letters intended to evoke messages from kidnapers made from cut-out letters).

One day Steve Jobs stopped by the software group, as he often did at the end of the day. He frowned as he looked at the font names on a menu. "What are those names?", he asked, and we explained about the Paoli Local.

"Well", he said, "cities are OK, but not little cities that nobody's ever heard of. They ought to be WORLD CLASS cities!"

So that is how Chicago (Elefont), New York, Geneva, London, San Francisco (Ransom), Toronto, and Venice (Bill Atkinson's script font) got their names.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Swedish Campground

Author: Andy Hertzfeld
Date: August 1983
Characters: Susan Kare, Steve Jobs
Topics: Software Design
Summary: How we got the unusual symbol used for the menu command key



We thought it was important for the user to be able to invoke every menu command directly from the keyboard, so we added a special key to the keyboard to invoke menu commands, just like our predecessor, Lisa. We called it the "Apple key"; when pressed in combination with another key, it selected the corresponding menu command. We displayed a little Apple logo on the right side of every menu item with a keyboard command, to associate the key with the command.

One day, late in the afternoon, Steve Jobs burst into the software fishbowl area in Bandley III, upset about something. This was not unusual. I think he had just seen MacDraw for the first time, which had longer menus than our other applications.

"There are too many Apples on the screen! It's ridiculous! We're taking the Apple logo in vain! We've got to stop doing that!"

After we told him that we had to display the command key symbol with each item that had one, he told us that we better find a different symbol to use instead of the Apple logo, and, because it affected both the manuals and the keyboard hardware, we only had a few days to come up with something else.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



It's difficult to come up with a small icon that means "command", and we didn't think of anything right away. Our bitmap artist Susan Kare had a comprehensive international symbol dictionary and she leafed through it, looking for an appropriate symbol that was distinctive, attractive and had at least something to do with the concept of a menu command.

Finally she came across a floral symbol that was used in Sweden to indicate an interesting feature or attraction in a campground. She rendered a 16 x 16 bitmap of the little symbol and showed it to the rest of the team, and everybody liked it. Twenty years later, even in OS X, the Macintosh still has a little bit of a Swedish campground in it.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



The Puzzle

Author: Andy Hertzfeld
 Date: September 1983
 Characters: Andy Hertzfeld, Jerome Coonen
 Topics: Software Design, Management
 Summary: The puzzle desk accessory became controversial



The Puzzle desk accessory

The original Macintosh could only run one real application at a time, but it could also concurrently run little programs called "desk accessories" that shared memory with the main application. Like the system itself, most of the desk accessories were written in 68000 assembly language, but in the fall of 1982, I decided to write one in Pascal as a proof of concept and to show



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



developers how to do it, by writing a tiny adaptor that the Pascal code linked with.

Desk accessories were usually utilities, like the calculator or the alarm clock, but I thought that we should also have a game or two, to show that the Macintosh was fun, too. I decided to write a "15 number puzzle", where there are fifteen numbered tiles in a four by four space that must be arranged in sequential order. If you click on a tile next to the empty space, it slides into that space. It was a fun way to waste time and build up your mouse coordination.

Since the number puzzle was written in Pascal, it had to link with the Pascal run-time, which dragged in lots of extra code that wasn't used. This made the Puzzle over 6K bytes long, even though most of it was just the run-time.

By the fall of 1983, it was time to make decisions about what to include in the shipping product. We had shown the Mac to a number of industry analysts, and while most were enthusiastic, some didn't really get the graphical user interface and thought it was "game-like", not suitable for serious computing. This made some of the Macintosh marketing folks a bit leery about the more whimsical aspects of the design, and the puzzle, being an actual game, became somewhat controversial.

Jerome Coonen, the software manager, came by my cubicle one morning to tell me that they decided not to ship the puzzle, partially because of the game-like perception, but mostly because it was just too big. Applications were very tight on RAM, and the puzzle was one of the biggest desk accessories because it was written in Pascal. At over 6K bytes, it also ate into the available disk space.

I liked the puzzle and I didn't want to capitulate to the buttoned-down, all business view of the customer, so I told Jerome, "You know, the puzzle doesn't have to be so big. I bet I could rewrite it and get it to take up less than 1K bytes. Would you keep it if I got it that small, or is it really the other issue?"

Jerome thought about it, and then told me if I could get it down to 600 bytes or so, it would be in the release. The only problem was I had to get it done over the weekend, because they had to send the manuals out to the printer soon, and there was plenty of other stuff for me to work on.

Of course, I couldn't resist a challenge like that. It only took a few hours on Saturday to recode it in assembly language, and get it down to the required 600 bytes, since it no longer had to link with the bulky Pascal code. I proudly showed it off to everyone on Monday, and it did make it into the first release, and stayed there for many years, although it was completely rewritten a couple of times, for various reasons.

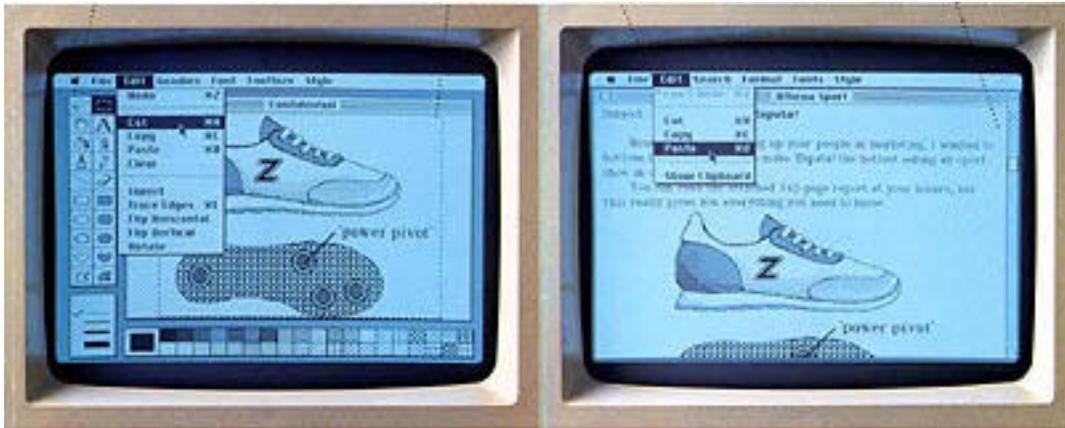


APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Cut, Paste and Crash

Author: Andy Hertzfeld
 Date: September 1983
 Characters: Larry Kenyon, Susan Kare, Andy Hertzfeld
 Topics: Technical, Software Design, Bugs
 Summary: We uncover a particularly bad bug shortly after ROM freeze



The sneaker was used to demo cut and paste

One of the last parts of the Macintosh system software to be finished before freezing the ROM in September 1983 was the "clipboard manager", which was the code responsible for facilitating cutting and pasting information between applications. The clipboard manager provided some simple calls to access and manipulate the "clipboard", a memory buffer that held the last piece of data that was cut or copied. The trickiest part of the clipboard manager was the way it managed memory when the user quit an application.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



The clipboard buffer was typically kept in the primary memory area that was available to an application, known as the "application heap". But when an application terminated, its application heap was deallocated, before a brand new heap was allocated for the incoming application. The clipboard manager had to take special measures to preserve the clipboard during this interregnum, when no application heap was available to hold it.

The first thing we thought of was to write the clipboard to disk while we were between applications, which worked but was problematic because the Macintosh didn't have a fixed disk, so there might not be a floppy in the drive, or perhaps it was write protected. We also tried to copy the clipboard into the system heap, which persisted between applications, but the clipboard was often too big to fit there. Ultimately, we solved the problem by copying the clipboard to the stack, which was a large area in high memory used to hold application variables and run-time state like return addresses, where it stayed until it was moved into the application heap of the newly launched application.

A few weeks after the ROM was frozen, Apple put together a sales "roll-out" presentation to introduce the sales team to all things Macintosh, in order to prepare them to start selling the machine a few months later. There was lots of the usual sales hoopla, but the clear highlight of the event was the first-person training on the machine itself, where each salesperson would get to try out a Macintosh for the first time. A teacher guided them through a simple demo which let them learn to use the mouse and then try out MacWrite, MacPaint and the Finder.

The climax of the demo had them loading a MacPaint document, selecting an area and copying it to the clipboard, and then launching MacWrite and pasting the image into a MacWrite document, easily mixing text and graphics, which was very impressive back in those days. Susan Kare had drawn a terrific, detailed rendering of a sneaker for the Macintosh brochure, which was used as the graphic for the demo. They were supposed to select and copy the sneaker and then paste it into MacWrite.

I was invited to go to the sales presentation, but I couldn't, because I was still working frantically with the rest of the software team to finish the software in time to ship the following January. I was busy working in the afternoon when I received a panicked phone call from one of the sales guys involved with the presentation.

"We've got a problem!", he told me. "The sneaker demo is crashing. Not all the time, though, sometimes it works and sometimes it doesn't. It's impossible to predict when it will happen, but when it does, the crash is really bad - as soon as you quit MacPaint, the screen goes crazy and then it reboots. We're going to have to stop doing the cut and paste part of the demo unless you get us a fix soon."

I got off the phone and thought about it. I was describing the problem to someone when I suddenly stopped in mid-sentence, because I realized what was going on. After quitting the application, the clipboard manager would copy the clipboard to the stack. It subtracted the size of the clipboard from the stack pointer, and then moved the memory into the allocated space. I realized that around half of the time, the size of the clipboard would be an odd number, which would end up setting the stack pointer to an odd number, which was a perversely



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



horrible thing to do. The Mac's 68000 microprocessor couldn't fetch memory from an odd address; it generated an error if you tried. But the stack is used for the basic operation of the processor, including error handling. An odd stack pointer would essentially drive the 68000 crazy, recursively faulting as it tried to handle exceptions. It worked around half of the time because statistically the clipboard size would be an even number around half of the time. But the other half would cause the flamboyant crash that was being described.

Once I understood what was going on, it was easy to fix by rounding up the clipboard size to an even number. Unfortunately, the errant code in question was in ROM, which was already frozen in immutable silicon. This was the first major bug that I knew of in the ROM and I wondered if we were going to have to spin another version. But Larry Kenyon had already figured out a sneaky technique to fix ROM bugs, by patching system traps. We had always figured to replace entire buggy system calls in that fashion, but Larry thought of a finer grain way to do it, by patching the first trap called before or after the problem area. Basically, we could grab control any time the system was invoked, and then add code to fix problems. I used Larry's technique to devise a patch that fixed the odd stack problem, and then he helped me incorporate it into the System file, so it was loaded when the system booted.

We made a floppy containing the new System file, which was flown out to the sales presentation with a sales manager who was leaving the next day. But it took a while for the fixed System to proliferate to everyone, so for the next few weeks I had to brace myself every time that I saw someone about to cut and paste between applications.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



We're Not Hackers!

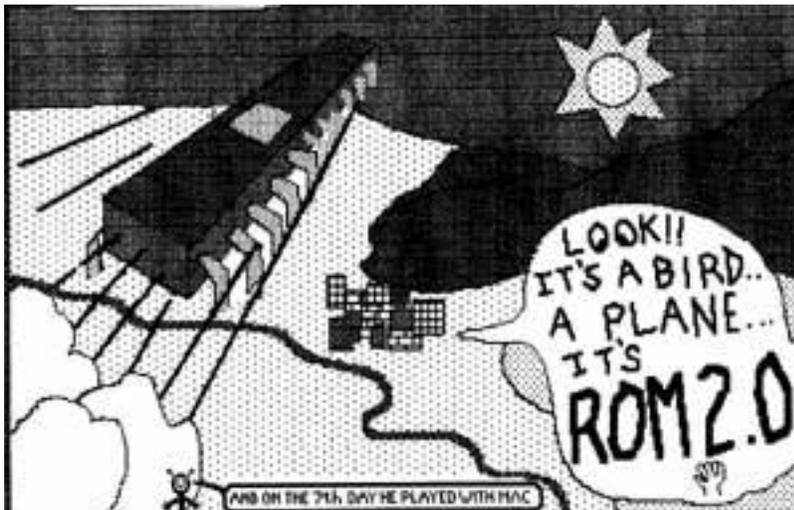
Author: Andy Hertzfeld

Date: September 1983

Characters: Larry Kenyon, Andy Hertzfeld, Bill Atkinson, Burrell Smith, Steve Capps

Topics: Software Design, Technical, Personality

Summary: We were always dealing with memory limitations



Steve Capps drew this with an early version of MacPaint, to celebrate a ROM release

From the beginning, the Macintosh was conceived to be a very low cost, high volume personal computer. It was important for the design team to keep manufacturing costs as low as possible. Since memory was relatively expensive, we were always dealing with memory limitations.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



One of the most clever parts of Burrell Smith's original, 68000-based digital board was the "bus transformer" logic that multiplexed the data bus, allowing him to hook up the 68000, which demanded a 16 bit data bus, to only 8 memory chips. He also included a single, byte-wide 64kbit ROM chip, so that first Macintosh, circa January 1981, had a total of 64K bytes of RAM and 8K bytes of ROM.

But as we started to get some software going on the prototype, it became increasingly clear that we didn't have enough RAM for the kind of graphic intensive applications that we wanted to build; after all, just the frame buffer for the bit-map display took up almost one third of the available memory. And furthermore, Bill Atkinson's graphics routines alone had recently exceeded the size of the 8K ROM. So, when the digital board was redesigned to incorporate the SCC chip in June 1981, Burrell added another row of 8 memory chips, doubling the RAM size to 128K, and added another ROM chip as well, doubling the ROM size to 16K. We vowed that we would fight hard to keep that the last increase (in contrast with the Lisa, whose memory requirement's were growing considerably faster than Moore's Law)

Even though the ROM size doubled to 16K, that was barely enough to contain our prototype environment if we included the graphics routines. Burrell figured he could add a third ROM chip, for 24K total. Two of the ROM chips were hooked up directly to the 68000's 16-bit bus, so code could run faster, while the third chip shared the "bus transformer" circuit with the RAMs.

We built 50 Mac prototypes in the fall of 1981, each containing 24K of ROM, burned into EPROM. Although the system fit readily in 24K, we were still worried that soon it would be an unbearably tight squeeze. Burrell never liked the inelegance of three different ROMs, either.

One day, in early 1982, Burrell returned from a meeting with a semiconductor company's sales representative really excited, almost running into my office. "OK, you say that you won't be able to fit in 24K, right? Be honest - how much will we really need?"

We always seemed to need just a little more ROM than we had available. "I think we'd definitely make it if we had 32K", I responded.

Burrell laughed. "No you won't. It's clear that won't be enough, since the software isn't close to being finished yet. But I just heard that the 256Kbit ROMs are really close, and they'll be ready if we ship in early 1983. So I can use two 256Kbit chips, connected up to the 16-bit bus, and we'll have 64K bytes of ROM. 64K! ROM is half the price per bit of RAM, so it makes sense to use as much as we can. I know you'll be asking for even more someday, but that should keep you busy for a while."

At first, 64Kbytes seemed boundless. We were already trying to write as tight code as we could, and it seemed like it would be plenty, since we weren't even using 32K yet. But sure enough, as the system came together in the spring of 1983, we were beginning to strain against the size limit.

Fortunately, we had started to use the resource manager to load objects like fonts and drivers, so we had some flexibility about keeping stuff on disk



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



instead of the ROM. Jerome and I designed the "PACK" mechanism, where the resource manager would load code for optional packages, like the floating point routines. But code on floppy disk is much slower to load, and it also would reduce the effective size of each disk.

Even though we tried to make our code as small as possible initially, the lack of space in the ROM made us work even harder to reduce the footprint. We developed a number of unusual space saving techniques, some of which were inspired by tricks Woz used in the Apple II ROM. For example, we'd often push parameters on the stack out of order, sometimes four times in a row, because we had a value in a register that we would need later, and we didn't want to fetch it again. We knew this made the code harder to maintain, but we thought it was worth it.

As ROM freeze time approached, the entire team started to focus on code compression, to get as much code to fit in the ROM as possible. We had a few practice sessions, where everyone explained their favorite space saving techniques, and then we all plowed through the code, saving a dozen bytes here and there. Steve Capps, as usual, came up with a good one by developing a simple way of compressing the four or five icons that were built into the ROM, saving hundreds of precious bytes in the process.

Bill Atkinson didn't participate in the code crunching marathon, and wouldn't allow QuickDraw to be subjected to it, except in a few cases, because he believed that all code should be as simple and clear as possible, and thought, probably correctly, that we'd be better off without the tricks in the long run. Just before the ROM was frozen, in September 1983, he found a bug in the memory manager, that we devised a simple fix for.

I went with Bill to Larry Kenyon's cubicle, where he was maintaining the memory manager sources. Bill looked over our shoulders as we added a little code to correct the bug. But he objected when he noticed that we used one of our coding tricks (I'll explain the exact issue in a comment below) while making the fix, and began to get upset.

"Hey, that's not the right way to code. What are you guys, a bunch of hackers? I'm not sure that I want to work with a bunch of hackers."

Both Larry and I cared more about pleasing Bill than saving every possible byte or cycle, so we changed our fix to use the slower, more conservative, Bill-approved technique. We also added a comment to the instruction in the source code, to remind us why we did it the slower way in this circumstance. The comment said "We're Not Hackers!".



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Monkey Lives

Author: Andy Hertzfeld
Date: October 1983
Characters: Steve Capps, Bill Atkinson
Topics: Software Design, Testing, Technical
Summary: The very first location in low memory

The original Macintosh only had 128K bytes of RAM (that's one eighth of a megabyte), so dealing with memory management was usually the hardest part of writing both the system and applications. We allocated around 16K bytes for system use, and another 22K bytes for the 512 by 342 black and white screen, so applications were left with only 90K bytes or so. The bigger ones like MacWrite or MacPaint seemed to be bursting at the seams.

By the fall of 1983, MacWrite and MacPaint were pretty much feature complete but still needed a lot of testing, especially in low memory conditions. MacPaint needed to allocate three off-screen buffers, with each the size of the entire screen, so it was always skirting the edge of running out of memory, especially when you brought up a desk accessory, but the specific sequences that led to crashes were difficult to reproduce.

Steve Capps had been working on a "journaling" feature for the "Guided Tour" tutorial disc, where the Macintosh could demo itself by replaying back events that were recorded in a prior session. He realized that the so-called "journaling hooks" that were used to feed pre-recorded events to the system could also be the basis of a testing tool he called "The Monkey".

The Monkey was a small desk accessory that used the journaling hooks to feed random events to the current application, so the Macintosh seemed to be operated by an incredibly fast, somewhat angry monkey, banging away at the mouse and keyboard, generating clicks and drags at random positions with wild abandon. It had great potential as a testing tool, so Capps refined it to generate more



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



semantically rich events, with a certain percentage of the events as menu commands, a certain percentage as window drags, etc.

The Monkey proved to be an excellent testing tool, and a great amusement, as well. Its manic activity was sort of hypnotic, and it was interesting to see what kind of MacPaint pictures the Monkey could draw, or if it would ever produce something intelligible in MacWrite. At first it could crash the system fairly easily, but soon we fixed the more obvious bugs. We thought it would be a good test for an application to see if it could run the Monkey all night, but usually it didn't run for more than 20 minutes, even if it didn't crash, because the Monkey would invariably select the quit command.

Bill Atkinson came up with the idea of defining a system flag called "MonkeyLives" (pronounced with a short "i" but often mispronounced with a long one), that indicated when the Monkey was running. The flag allowed MacPaint and other applications to test for the presence of the Monkey and disable the quit command while it was running, as well as other areas they wanted the Monkey to avoid. This allowed the Monkey to run all night, or even longer, driving the application through every possible situation.

We kept our system flags in an area of very low memory reserved for the system globals, starting at address 256 (\$100 in hexadecimal), since the first 256 bytes were used as a scratch area. The very first slot in the system globals area, address 256, had just been freed up, so that's where we put the MonkeyLives boolean. The Monkey itself eventually faded into relative obscurity, as the 512K Macintosh eased the memory pressure, but its memory was kept alive by the curious name of the very first value defined in the system globals area.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Stolen From Apple

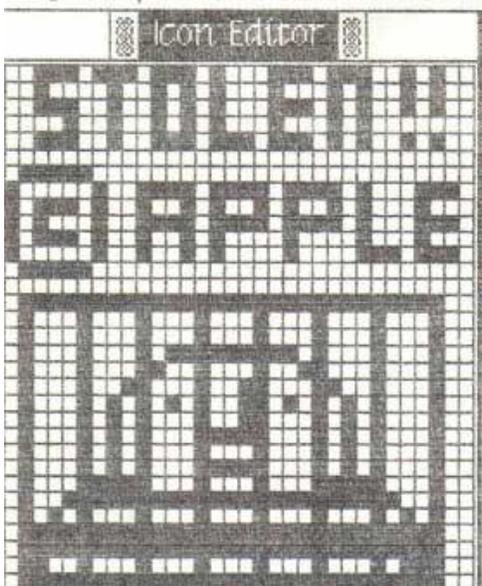
Author: Andy Hertzfeld

Date: October 1983

Characters: Steve Jobs, Susan Kare, Steve Capps, Andy Hertzfeld

Topics: Software Design

Summary: We put a hidden icon in the ROM



The secret "Stolen From Apple" icon

In 1980, a company called Franklin Computer produced a clone of the Apple II called the Franklin Ace, designed to run the same software. They copied almost every detail of the Apple II, including all of its ROM based software and all the documentation, and sold it at a lower price than Apple. We even found a



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



place in the manual where they forgot to change "Apple" to "Ace". Apple was infuriated, and sued Franklin. They eventually won, and forced Franklin to withdraw the Ace from the market.

Even though Apple won the case, it was pretty scary for a while, and it wasn't clear until the end that the judge would rule in Apple's favor - Franklin argued that they had a right to copy the Apple II ROMs, since it was just a "functional mechanism" necessary for software compatibility. We anticipated that someone might try a similar trick with the Macintosh someday. If they were clever enough (which Franklin wasn't), they could disguise the code (say by systematically permuting some registers) so it wouldn't look that similar at the binary level. We thought that we better take some precautions.

Steve decided that if a company copied the Mac ROM into their computer, he would like to be able to do a demo during the trial, where he could type a few keystrokes into an unmodified infringing machine, and have a large "Stolen From Apple" icon appear on its screen. The routines and data to accomplish that would have to be incorporated into our ROM in a stealthy fashion, so the cloners wouldn't know how to find or remove it.

It was tricky enough to be a fun project. Susan designed a nice "Stolen from Apple" icon, featuring prison bars. Steve Capps had recently come up with a simple scheme for compressing ROM-based icons to save space, so we compressed the icon using his technique, which not only reduced the overhead but also made it much harder to detect the icon. Finally, we wrote a tiny routine to decompress the icon, scale it up and display it on the screen. We hid it in the middle of some data tables, so it would be hard to spot when disassembling the ROM.

All you had to do to invoke it is enter the debugger and type a 6 digit hexadecimal address followed by a "G", which meant execute the routine at that address. We demoed it for Steve and he liked it. We were kind of hoping someone would copy the ROM just so we could show off our foresight.

As far as I know, no one ever did copy the ROM in a commercial project, so it wasn't really necessary, but it did create some intrigue for a while. We let it slip that there was a "stolen from Apple" icon hidden in there somewhere, partially to deter people from copying the ROM. At least one hacker became moderately obsessed with trying to find it.

Steve Jasik was the author of the MacNosy disassembler/debugger, which could be used to create pseudo-source for the ROM. He found out about the "stolen from Apple" icon pretty early on, and became determined to isolate it. He lived in Palo Alto, so I would occasionally bump into him, and he would ask me for hints or tell me his latest theory about how it was concealed, which was invariably wrong.

This went on for two or three years, before he finally cracked it: I ran into him and he had it nailed, telling me about the compressed icon and the address of the display routine. I congratulated him, but was never sure if he figured it out himself or if someone with access to the source code told him.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



The Mythical Man Year

Author: Andy Hertzfeld

Date: October 1983

Characters: Steve Jobs, Bill Atkinson

Topics: Software Design, Personality, Reality Distortion, QuickDraw

Summary: Steve estimates the effort that went into Quickdraw



Bill Atkinson and Burrell Smith on the cover of Byte

One of our first encounters with the press was a group interview with Byte magazine in October 1983. We wanted an article to come out concurrently with the Mac into the third week of January, and Byte had a three month lead time, so they were the first.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Byte was one of the first PC hobbyist magazines, written for a fairly technical audience of computer enthusiasts. Five or six of us were being extensively quizzed by two Byte editors, including Steve Jobs. We were talking about the Mac's graphical user interface software, and how long it took to develop.

Quickdraw, the amazing graphics package written entirely by Bill Atkinson, was at the heart of both Lisa and Macintosh. "How many man-years did it take to write QuickDraw?", the Byte magazine reporter asked Steve.

Steve turned to look at Bill. "Bill, how long did you spend writing Quickdraw?"

"Well, I worked on it on and off for four years", Bill replied.

Steve paused for a beat and then turned back to the Byte reporter. "Twenty-four man-years. We invested twenty-four man-years in QuickDraw."

Obviously, Steve figured that one Atkinson year equaled six man years, which may have been a modest estimate.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Disk Swapper's Elbow

Author: Steve Capps
Date: January 1984
Characters: Steve Capps
Topics: Software Design, Technical, Bugs
Summary: a last minute bug causes some problems

One of the more common afflictions of early Macintosh users was the dreaded "Disk Swapper's Elbow" caused by a disk copying operation run amok. Disk swapping was a necessary evil caused by having 400KB floppy disks, 128KB of RAM, and a single floppy drive. If a user wanted to make a backup of a disk, she had to eject the source disk, insert a blank one, format it, drag the source disk over the new disk, and then the Finder would copy data piece-by-piece with the necessary swapping.

A typical application on a 128K Mac had about 85K of memory available; the rest was used by the system, mostly for the bitmap display. A simple calculation shows that copying a 400K disk should involve about 5 or 6 swaps. Five disk swaps was barely tolerable, but, as early Finder users will remember, occasionally it would take well over 20 disk swaps.

You'd start a disk copy and hold your breath during the fifth, and hopefully, final swap. If the Mac dutifully disgorged the floppy that sixth time, you'd convince yourself you miscounted, cross your fingers, and hope for the best. By the seventh swap you start cursing because you knew you were trapped and you'd start wondering about investing in a second, external drive.

Even though the whole Finder was only 46K of code and had about 10K of overhead, the remaining 30K of memory was too small for practical copying. So, I had to break up the code into two chunks: the bare minimum for copying and all the rest. Then, I had to carefully flush out all data that was cached in memory, preload the small disk-copying chunk of code, and coalesce the balance of RAM.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Usually, the Finder ended up with 75K of free memory and things worked as planned. But, sometimes the system would mysteriously reload the larger chunk of the Finder code, fragment the free memory, and cause another case of Disk Swapper's Elbow.

It took me a long time to figure out what happened because we had rarely seen this in testing. There were a few bug reports of this problem that were never reproducible. The bug reports went like this: "Copied a disk, it took 20+ swaps \$%#@!!! Tried a second time, it was fine." The reason this was not reproducible was because we were all expert mouse users and usually skipped the crucial misstep.

When anybody first starts using a mouse, dragging is one of the more difficult things to do. It's actually quite awkward to click down, move the mouse while holding down the button, and then release. Beginners very often accidentally release the mouse button while dragging. In the Finder, this means you could "drop" an icon you were dragging. You rarely thought about this (unless you happened to drop it over a folder and it disappeared); you'd immediately pick the icon up and continue the drag. It turned out if you dropped the disk icon during a disk copy, you'd induce the bug. Since all of the team members had been using the mouse for years by this time, we rarely dropped icons which is why we could never reproduce the problem.

To support spatial memory, the Finder always remembered where icons were located on the desktop. When you dropped the icon -- even for a half a second -- the Finder would dutifully record its location. The routine to save the icon's location was, as you probably guessed, in the big portion of the Finder's code. When this bug occurred, the Finder would carefully massage the memory for copying and then belatedly discover the icon's location hadn't been flushed out. It would blindly call the routine to flush it and you now know what would happen...

I introduced this bug about 2AM the morning we built the final disks. This bug was caused by a fix to a much more egregious bug, so it was definitely the lesser of two evils...really!



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



It Sure Is Great To Get Out Of That Bag!

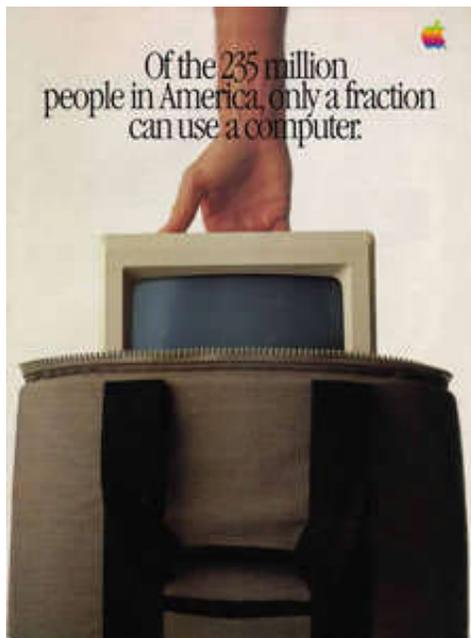
Author: Andy Hertzfeld

Date: January 1984

Characters: Steve Jobs, Steve Capps, Bruce Horn, Andy Hertzfeld,
Susan Kare, Mike Boich, Mark Barton, Steve Hayden

Topics: Marketing, The Launch, Software Design

Summary: We need a demo for the intro



Cover of Intro Brochure



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



It took a monumental effort, fueled by inordinate amounts of chocolate covered espresso beans, to finally finish the first release of the Macintosh software in time for the introduction. We finished with literally no time to spare, shipping the "golden master" of the "Write/Paint" disk to the factory at 6am on Monday morning January 16th, a week before the introduction. By that point, most of the software team hadn't slept for days, so we all went home to collapse.

I thought I would need to sleep for at least 24 hours, but I woke up after 6 hours with a desire to go back to Apple to see if the release held up, and to see how everyone else was feeling. By 5pm, most of the software team had dragged themselves back for the same reason, and we were lounging around in a tired daze, happy that we finally shipped, but still not quite believing it, when Steve Jobs strode into the software area.

"Hey, pick yourselves up off the floor, you're not done yet!"

Uh-oh, I thought, someone must have found a show-stopper in the release and we're going to have to track it down. But that's not what he meant.

"We need a demo for the intro! The Mac deserves to have a great demo for its first public showing. I want the Mac to play the theme from "Chariots of Fire" while it's showing a slide show of the apps. Plus lots of other cool stuff, whatever you can come up with. And it needs to be done by the weekend, to be ready for the rehearsals."

We moaned and groaned about being tired, but as we talked we realized that it would be fun to cook up something impressive. We were too tired to think about it right away, but when we came back the next day, a plan started to emerge.

Capps had an idea to use a gigantic font to scroll "Macintosh" across the screen, one letter at a time, to start the demo, so he worked on that, as well as the slideshow. Bruce Horn wanted to do a starry night with twinkling stars, and a skywriter writing "Macintosh" in cursive across the night sky. Susan worked on an intro graphic of the Mac sitting in its canvas carrying bag, and well as some of the other graphics for the slideshow part. I integrated all the pieces and also signed up for the "Chariots of Fire" music part, since no one else wanted to do that.

It's hard to write a music editor/player in two days, but I managed to put something together that could actually play the "Chariots of Fire" theme, but it didn't sound very good, since it used simple sine waves without any envelope shaping. Steve immediately rejected it as lousy, which it was, and opted for using a CD of the "Chariots of Fire" theme to play in the background instead.

Meanwhile, as we were working on the demo, Mike Boich came by with Mark Barton, a third party developer who we seeded with an early Mac, because he had written an impressive program for the Apple II called S. A. M. (the Software Automatic Mouth). SAM was a speech generator, which converted text to speech, with a distinctive, winning personality. I had helped Mark with sound driver issues as he developed it, and now it finally made it to fruition. SAM sounded even better on the Mac, because we had 8 bits per sample and a higher sampling rate.

When Steve heard SAM talk, he immediately decreed that we had to incorporate SAM in the intro demo. "I want the Macintosh to be the first computer to introduce



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



itself!", he insisted. He told Mike Boich to quickly cut a deal with Mark so Apple could bundle the speech generator (rechristened Macintosh) and use it in the intro.

Since my music generator fell through, I got to do the speech part, using Mark Barton's libraries. I knew that I wasn't clever enough to be the Mac's speechwriter. I think Susan had the idea of asking Steve Hayden, Chi at-Day's head writer, to do it. Steve was the guy who conceived the 1984 commercial and was as clever as they come. He was excited about helping out and got it done overnight.

Once we integrated all the pieces together, the demo didn't come close to being able to run on a standard Macintosh. Fortunately, we had a prototype of a 512K Mac in the lab, so we decided to cheat a little and use that for the demo, which made things fit.

The demo starts out with Susan's graphic of the Mac hidden in its carrying bag, on a curtained stage, displayed while the program prepared Capp's big letters. Suddenly, the music swells (from a CD, not generated by the Mac) and Capp's big letters scroll nimbly across the screen, spelling out "Macintosh". Then we transition to Bruce's skywriter, and then to various screen shots of applications, including third party applications like Microsoft's Multiplan and Chart. Finally, the music stops, the screen goes blank, and waits for Steve to press the mouse button. When he does, the Mac starts to speak, in strange but somehow endearing tones:

Hello, I am Macintosh. It sure is great to get out of that bag!

Unaccustomed as I am to public speaking, I'd like to share with you a maxim I thought of the first time I met an IBM mainframe: Never trust a computer that you can't lift!

Obviously, I can talk, but right now I'd like to sit back and listen. So it is with considerable pride that I introduce a man who has been like a father to me... Steve Jobs!



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Real Artists Ship

Author: Andy Hertzfeld
 Date: January 1984
 Characters: Andy Hertzfeld, Bill Atkinson, Steve Capps, Jerome Coonen,
 Bob Belleville, Steve Jobs, Donn Denman, Randy Wi ggi nton
 Topics: Management, Software Desi gn
 Summary: The final push to finish the software



Chocolate Covered Espresso Beans

By the fall of 1983, we had committed to announcing and shipping the Macintosh at Apple's next annual shareholder's meeting, to be held on January 24th, 1984. The failure of the Twiggy disk drive almost caused us to be late (see Quick, Hide In This Closet!) , but it seemed like the new Sony 3.5 inch drive solved all of our problems, and the rest of the hardware was ready to go. The Macintosh ROM was frozen in early September and sent out for fabrication. All that remained was finishing the System Disk, and our two applications, MacWrite and MacPaint.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



The software team worked hard over the Christmas break of 1983. The Finder still wasn't finished, and there were lots of performance problems, especially when copying files between disks, which seemed interminable. There was lots of integration testing to do, like cutting and pasting between applications, or applications interacting with desk accessories. As the New Year rolled around, it was clear that we were running out of time.

By the first week of January, the software team was working around the clock, testing and fixing problems that were found. Every employee in the building was drafted as a tester, and we held a series of dinners where Apple bought catered food for anyone who stayed late to test (see 90 Hours A Week And Loving It) .

Finally, the deadline for finishing the software was less than a week away, and it seemed obvious that there were still too many bugs for us to ship it. Late on Friday evening, we convinced ourselves that we needed an extra week or two to fix the remaining problems. Steve Jobs was on the East Coast, along with Bob Belleville and Mike Murray, doing press for the introduction, so we arranged for a conference call early Sunday morning to tell him about the slip.

Jerome Coonen, our software manager, spoke for the team, as we gathered around the speakerphone. We were exhausted, and progress was slow. There were still bugs that we hadn't gotten to the bottom of yet, and it didn't seem possible that we could make it in the time remaining. Jerome proposed that we ship "demo" software to the dealers for the introduction, and update all the customers with final software a few weeks later. We thought Jerome was pretty persuasive as we held our breath waiting for Steve to respond.

"No way, there's no way we're slipping!", Steve responded. The room let out a collective gasp. "You guys have been working on this stuff for months now, another couple weeks isn't going to make that much of a difference. You may as well get it over with. Just make it as good as you can. You better get back to work!"

We did manage to wrangle an extra couple of days, by virtue of working the weekend and moving the deadline to 6am Monday morning, when the factory opened, instead of Friday afternoon. We agreed to go home and rest up, and then come into work on Monday ready for the final push.

The final week was one of the most intense I ever experienced. Steve wanted Bill Atkinson and myself to fly to New York to present a Mac to Mick Jagger, but I decided that I needed to stay in Cupertino to help with the bug fixing. Some of us were pausing work to get photographed for magazines like Newsweek and Rolling Stone, which made others on the team feel terrible that they were being left out. At times, the atmosphere got pretty tense.

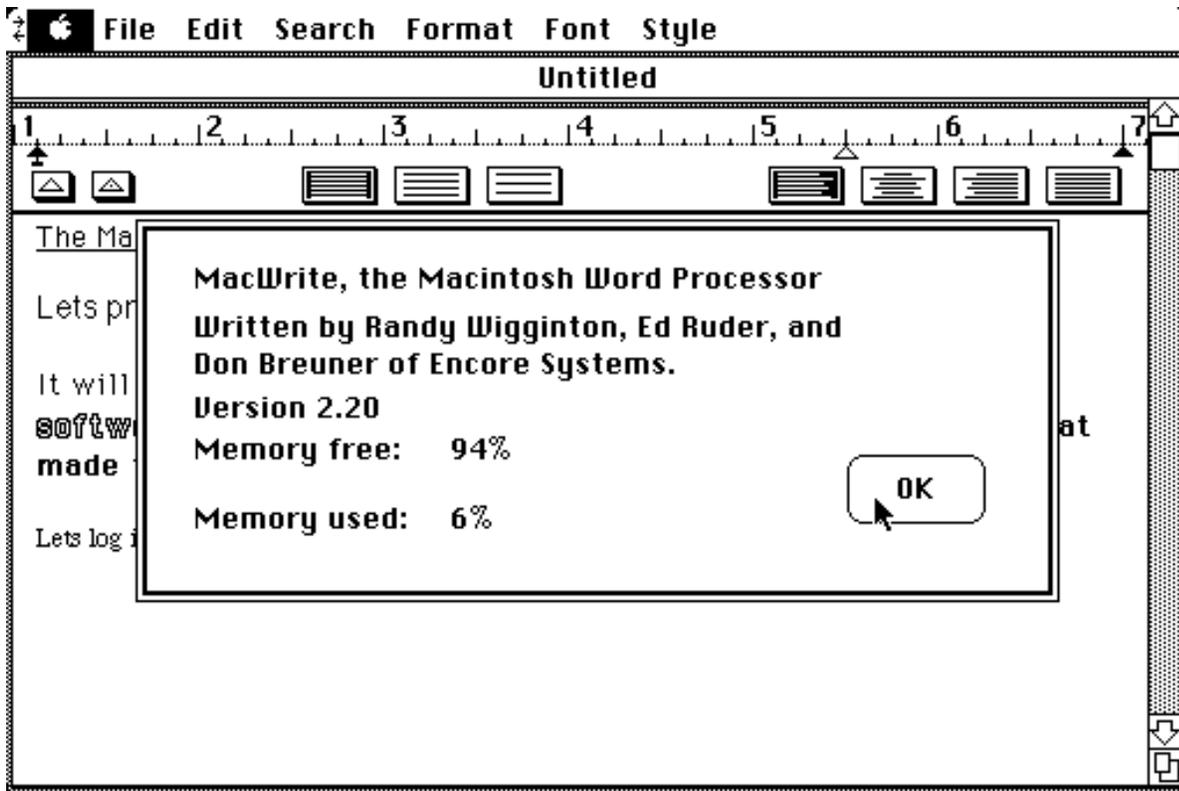
Friday finally rolled around and it was clear that there were still too many bugs in both the Finder and MacWrite. Randy Wigginton brought in a gigantic bag of chocolate covered espresso beans, which, along with medicinal quantities of caffeinated beverages, helped us forgo sleep entirely for the last couple of days. We starting doing release cycles that were only a few hours apart, re-releasing every time we fixed a significant problem.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



[MacWrite v2 screen shot]



When a new release was ready, we would all grab it and start testing again. At one point, around 2am on Sunday night, I stumbled across a bug in the clipboard code. I thought I knew what it might be, but I was so tired that I didn't want to deal with it. I tried to pretend that I didn't see the problem, but Steve Capps was watching my expression and knew there was something wrong. I also was too tired to sustain a pretense; he grilled me about the problem and then helped me craft a fix, since I was too tired to do it on my own.

Around 4am, we had a release where everything seemed to go wrong - even MacPaint was crashing, which was usually rock solid. But our final release, around 5:30am seemed to be much better; the worst problems seemed to have receded and we thought we might actually have a decent release candidate.

We all focused on testing the final release as much as we could until 6am, when Jerome would have to leave to drive it to the factory. It looked pretty good, but soon someone found a potential show stopper - the system seemed to hang when a blank disc was inserted during MacWrite - the disk didn't start formatting like it should. I realized that it was probably hung up waiting for an event, so I reached out and tapped on the space bar, and formatting commenced. Jerome thought the bug was bad enough to hold up the release, but he left to drive it to the factory anyway, figuring they needed to start duplication even if it was just going to be a demo release.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION

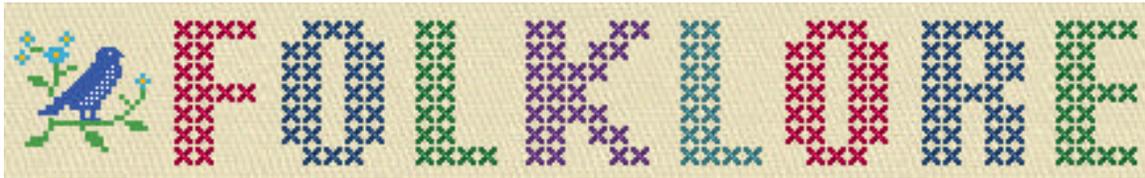


The sun had already risen and the software team finally began to scatter and go home to collapse. We weren't sure if we were finished or not, and it felt really strange to have nothing to do after working for so hard for so long. Instead of going home, Donn Denman and I sat on a couch in the lobby in a daze and watched the accounting and marketing people trickling into work around 7:30am or so. We must have been quite a sight; everybody could tell that we had been there all night (actually, I hadn't been home or showered for three days).

Finally, around 8:30 Steve Jobs arrived, and as soon as he saw us he immediately asked if we had made it. I explained the formatting bug to him, and he thought that it wasn't a show stopper, which meant that we were actually finished. I finally drove home to Palo Alto around 9am and collapsed on my bed, thinking that I'd sleep for the next day or two.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



MacBasic

Author: Andy Hertzfeld

Date: June 1985

Characters: Donn Denman, Bryan Stearns, Jerome Coonen, Randy Wigginton,
John Sculley, Bill Gates, Bill Atkinson, Steve Wozniak

Topics: Software Design, Management

Summary: The sad story of MacBasic



Applesoft Basic Cassette

When the Apple II was first introduced in April 1977, it couldn't do very much because there were few applications written for it. It was important to include some kind of programming language, so users, who were mostly hobbyists, could write their own programs. "Basic", which was designed for teaching introductory programming by two Dartmouth professors in the 1960s, became the language of choice for early microcomputers because it was interactive, simple and easy to use. The Apple II included a Basic interpreter known as "Integer Basic", written from scratch by Steve Wozniak, which was almost as idiosyncratically brilliant as his hardware design, stored in 5K bytes of ROM on the motherboard. It also came with Microsoft's Basic interpreter, dubbed "Applesoft Basic", on cassette tape. Sadly, Applesoft eventually displaced Integer Basic in ROM in the Apple II

ANECDOTES ABOUT THE DEVELOPMENT OF APPLE'S ORIGINAL MACINTOSH COMPUTER



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Plus because it had the floating point math routines that Woz never got around to finishing.

Donn Denman started working at Apple around the same time that I did, in the summer of 1979. His job was to work with Randy Wigginton on porting Applesoft Basic to the Apple III. They needed to rewrite parts of it to deal with the Apple III's tricky segmented memory addressing, as well as porting it to SOS, the new operating system designed for the Apple III. It was easy for me to track Donn's steady progress because he sat in the cubicle across from mine when we moved into Bandley III in the spring of 1980.

By the summer of 1981, the Macintosh project was beginning to hit its stride, and we started thinking about the applications that we wanted to have at launch to show off the unique character of the Macintosh. Besides a word processor and a drawing program, we thought that a Basic interpreter would be important, to allow users to write their own programs. We decided we should write it ourselves, instead of relying on a third party, because it was important for the Basic programs to be able to take advantage of the Macintosh UI, and we didn't trust a third party to "get it" enough to do it right.

I still had lunch with some of my friends in the Apple II group a couple of times a week, and I started trying to convince Donn to join the Mac team to implement our Basic. He was reluctant at first, since the Mac project was still small and risky, but he was pretty much finished with Apple III Basic and was full of ideas about how to do it better. He eventually couldn't resist and joined the Mac team in September 1981.

A Basic interpreter consists of a text editor for inputting your program, a parser to translate it into a series of byte codes, and an interpreter to execute the byte-coded instructions. Donn wrote the interpreter first, and then hand-coded some byte codes to test it. He implemented some graphics primitives early on, since they were nice to demo. In a few months, he had a pretty impressive demonstration program going that drew elaborate graphical trees recursively, in multiple windows simultaneously, showing off the interpreter's threading capabilities.

By the spring of 1982, it was apparent that Donn needed some help if we wanted Basic ready for the introduction, which at the time was supposed to be in January 1983. We decided to hire Bryan Stearns to help him, who Donn knew from the Apple II team. Bryan was only 18 years old, but he was excited about the project and Donn thought they worked well together, so we gave him a chance.

But Basic still had a hard time getting traction, especially since the system was evolving rapidly beneath it. After six months or so, I was surprised to hear that Bryan was quitting the project to work at a tiny start-up founded by Chuck Mauro, who I had helped with his 80 column card for the Apple II. I tried to talk him out of it but he left anyway. By the spring of 1983, it was so obvious that Basic wouldn't be ready for the introduction that the software manager, Jerome Coonen, pulled Donn off of it to work on other parts of the ROM and the system. Donn worked on desk accessories and wrote the alarm clock and notepad, as well as the math guts of the calculator.

After the Mac shipped in January 1984, Donn went back to work on Basic with renewed vigor, determined to get it finished. Apple brought in some free-lance



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



writers to write books about it (including Scot Kamins, who was a co-founder of the first Apple users group in the Bay Area). But Microsoft surprised us, and released a Basic for the Macintosh that they didn't tell us they were developing. It was everything that we expected and feared, since it was essentially console-based - it didn't really use the Mac user interface. Donn was making good progress and looked to be on track to ship in early 1985; we were excited to show the world what Basic should really look like on the Macintosh.

Unfortunately, there was another problem on the horizon. Apple's original deal with Microsoft for licensing Applesoft Basic had a term of eight years, and it was due to expire in September 1985. Apple still depended on the Apple II for the lion's share of its revenues, and it would be difficult to replace Microsoft Basic without fragmenting the software base. Bill Gates had Apple in a tight squeeze, and, in an early display of his ruthless business acumen, he exploited it to the hilt. He knew that Donn's Basic was way ahead of Microsoft's, so, as a condition for agreeing to renew Applesoft, he demanded that Apple abandon MacBasic, buying it from Apple for the price of \$1, and then burying it. He also used the renewal of Applesoft, which would be obsolete in just a year or two as the Mac displaced the Apple II, to get a perpetual license to the Macintosh user interface, in what probably was the single worst deal in Apple's history, executed by John Sculley in November 1985.

When Donn found out that MacBasic had been cancelled, he was heart-broken. His manager told him "it's been put on hold indefinitely" and instructed him to destroy the source code and all copies, but refused to answer Donn's questions about what was going on. Later that day Donn went for a wild ride on his motor cycle and crashed it, returning home scraped up but with no real damage, except to his already battered ego.

Bill Atkinson was outraged that Apple could treat Donn and his users so callously, and let John Sculley know how he felt, but the deal was done and couldn't be reversed.

Donn quickly filed for a leave of absence, but eventually returned to Apple to work on various projects, including AppleScript.

The Beta version of MacBASIC had been released to interested parties, including Dartmouth University which used it in an introductory programming class. Apple tried to get back all the copies, but the Beta version was widely pirated, and two books on MacBASIC were published, and sold quite well for several years.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Desk Ornaments

Author: Andy Hertzfeld
Date: undated
Characters: Bud Tribble, Chris Espinosa, Donn Denman, Larry Kenyon,
Steve Capps, Bill Atkinson, Steve Jobs, Susan Kare
Topics: Software Design, User Interface, Personality
Summary: A brief history of desk accessories

One of the first architectural decisions that Bud and I made for the Macintosh system software in the spring of 1981 was that we were only going to try to run one application at a time. We barely had enough RAM or screen space to do even that, and we thought that we'd benefit from the resultant simplifications. Besides, multi-tasking was supposed to be Lisa's forte, and we didn't want to usurp all of the reasons for buying a Lisa.

Bud Tribble was usually on an even keel, but one afternoon in the fall of 1981 he came into my office, unusually excited. "You know, I've been thinking about it. Even if we can only run one major application at a time, there's no reason that we can't also have some little miniature applications running in their own windows at the same time."

That sounded intriguing to me. "What kind of little programs? How are they different?", I wondered.

Bud smiled. "You'd want tiny apps that were good at a specific, limited function that complements the main application. Like a little calculator, for example, that looked like a real calculator. Or maybe an alarm clock, or a notepad for jotting down text. Since the entire screen is supposed to be a metaphorical desktop, the little programs are desk ornaments, adorning the desktop with useful features."



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



"But where do we draw the line?", I asked. "What are the differences between an ornament and a real application?"

"Well, ornaments have to fit into the world of the main application", Bud responded, "but not the other way around. The main application owns the menu bar, for example, but maybe the ornaments can have a menu when they're active. The main application would still run its event loop, but it would occasionally pass events to the little guys. And of course you can cut and paste between them."

That sounded good to me, but we had plenty of other stuff to work on first. Bud left Apple in December 1981 to return to medical school, but I explained his concept to new team members and considered them to be part of our plan. Chris Espinosa liked the idea when we explained it to him, and he worked on a visualization of the calculator as part of learning to program with QuickDraw, culminating in a calculator construction set (see Calculator Construction Set). Eventually, after getting the initial implementations of the window, menu and event managers going, by May 1982 we had enough of the rudiments done to give desk ornaments a try.

The nascent Mac operating system already had an implementation of loadable device drivers, where installable code could be loaded from disk to manage optional peripherals. I decided to cast desk ornaments as a special kind of driver, adding a few extra calls so they could maintain a window on the screen, receive events from the host application, and occasionally get some processing time to accomplish something. Applications were required to make a system call periodically to yield some time for ornaments to run, and had to occasionally pass events to them and follow a few other conventions for co-existing. The very first desk ornament was a boring one, called "TestOrn", that displayed a rectangular window with an incrementing counter, showing that it was capable of doing background processing, even when it wasn't the top-most window. It wasn't very impressive, but it became the template from which all the others sprung.

In the fall of 1982, the first pass of the User Interface Toolbox was completed, so I had time to work more on ornaments, which by this time were renamed "desk accessories" by the pubs group who thought that the term "ornament" was too ornamental. I wrote a simple clock, and got the calculator going, modeled after Steve Jobs' design from the calculator construction set. The most interesting one that I wrote in that spurt was a desktop pattern editor, which was a sort of miniature icon editor that let you edit an 8 by 8 pixel pattern by clicking to toggle the dots on and off. When you made a pattern that you liked, you could make it the desktop pattern, filling the background of the screen with your new creation instead of the standard gray.

Within a couple of months, other people wanted to take a crack at writing desk accessories. Someone asked me if they could write one in Pascal, which wasn't easy because you had to pass parameters to the underlying driver in registers, which the Pascal compiler couldn't manipulate directly. I realized that you could write a little bit of standard glue code in assembly language to do the dirty work, and have most of your desk accessory written in Pascal. I wrote a number puzzle in Pascal as a test case, to prove that it could be done (see The Puzzle).



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



But once again I had to get back to other things and the desk accessories languished. But by the spring of 1983, the ROM was beginning to stabilize, and we realized that we had to get serious about the desk accessories that would ship with the initial system. Jerome assigned Donn Denman, who was taking a hiatus from working on Basic to help us finish the system, to help with desk accessories. Donn and I collaborated on finishing the calculator, with Donn doing the back-end that interfaced with the floating point routines. Donn was also responsible for the alarm clock and notepad.

Now that we had cut and paste working, I realized that it would be useful to have a desk accessory that could hold commonly used snippets of text and graphics, so they could be pasted into any application. I wrote a desk accessory called the "Gallery" that allowed you to manage a set of clippings using cut and paste, building it on top of the resource manager so it was easy to implement. Everyone seemed to like the functionality but not the name. I was explaining the situation to Larry Kenyon when he suggested that we call it the "Scrapbook", which was perfect, so the Scrapbook joined our growing armada of useful desk accessories.

The Macintosh's character encoding scheme allocated the upper half of the 8-bit character set to occasionally useful but obscure characters that were not portrayed on the keys, accessed with the option key, that were nearly impossible to remember. Steve Capps conceived of a desk accessory called "Key Caps" (named partially after himself?) that displayed a picture of the keyboard, with the keycaps changing depending on the meta-keys that were pressed, which allowed you to hunt for the special characters visually. He coded it up quickly and it became another great addition.

We needed a way to control various system parameters like the sound volume, or the mouse scaling parameters. We decided that a desk accessory would be perfect for that, since it would be easy to access no matter what application you were in. So the last desk accessory that I worked on before shipping, in November 1983, was the Control Panel. Susan Kare came up with a beautiful, highly graphical design that I implemented using a separate purgable resource for each section, so they didn't have to be in memory at once. It had a little rabbit and tortoise to represent a range of speeds, and lots of other graphical embellishments; after the Mac was released, one review described it as a crib toy, which I took more or less as a compliment.

The most controversial part of the Control Panel was the desktop pattern editor, which I had rescued from its earlier stand-alone incarnation. Users could select from a couple dozen prefabricated desktop patterns or edit their own in fat bits. It was this latter capability that caused problems, as it was pretty easy to create ugly, abominable patterns.

Bill Atkinson complained to me that it was a mistake to allow users to specify their own desktop patterns, because it was harder to make a nice one than it looked, and led directly to ugly desktops. But I thought that users should be free to do as they pleased, since it was their desktop, and it was easy to revert to one of the built-in patterns. Bill cared most about MacPaint, and didn't want a potentially ugly desktop pattern marring his creation. So he made MacPaint allocate a window that was the size of the screen when it started up, and filled it with the standard 50% gray pattern, making his own desktop



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



covering up the real one, thus protecting the poor users from their rash esthetic blunders, at least within the friendly confines of MacPaint.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



On Xerox, Apple and Progress

Author: Bruce Horn

Date: undated

Characters: Bruce Horn, Bill Atkinson, Larry Tesler, Steve Jobs,
Jerry Manock

Topics: Credit, Software Design, Origins

Summary: An essay discussing where the Mac user interface came from

This essay was written by Bruce in 1996, and is reprinted here with his permission. Bruce was one of the main designers of the Macintosh software, and he worked at Xerox for years before that, so he's uniquely qualified to discuss their relationship.

Where It All Began

For more than a decade now, I've listened to the debate about where the Macintosh user interface came from. Most people assume it came directly from Xerox, after Steve Jobs went to visit Xerox PARC (Palo Alto Research Center). This "fact" is reported over and over, by people who don't know better (and also by people who should!). Unfortunately, it just isn't true - there are some similarities between the Apple interface and the various interfaces on Xerox systems, but the differences are substantial.

Steve did see Smalltalk when he visited PARC. He saw the Smalltalk integrated programming environment, with the mouse selecting text, pop-up menus, windows, and so on. The Lisa group at Apple built a system based on their own ideas combined with what they could remember from the Smalltalk demo, and the Mac folks built yet another system. There is a significant difference between using the Mac and Smalltalk.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Smalltalk has no Finder, and no need for one, really. Drag-and-drop file manipulation came from the Mac group, along with many other unique concepts: resources and dual-fork files for storing layout and international information apart from code; definition procedures; drag-and-drop system extension and configuration; types and creators for files; direct manipulation editing of document, disk, and application names; redundant typed data for the clipboard; multiple views of the file system; desk accessories; and control panels, among others. The Lisa group invented some fundamental concepts as well: pull down menus, the imaging and windowing models based on QuickDraw, the clipboard, and cleanly internationalizable software.

Smalltalk had a three-button mouse and pop-up menus, in contrast to the Mac's menu bar and one-button mouse. Smalltalk didn't even have self-repairing windows - you had to click in them to get them to repaint, and programs couldn't draw into partially obscured windows. Bill Atkinson did not know this, so he invented regions as the basis of QuickDraw and the Window Manager so that he could quickly draw in covered windows and repaint portions of windows brought to the front. One Macintosh feature identical to a Smalltalk feature is selection-based modeless text editing with cut and paste, which was created by Larry Tesler for his Gypsy editor at PARC.

As you may be gathering, the difference between the Xerox system architectures and Macintosh architecture is huge; much bigger than the difference between the Mac and Windows. It's not surprising, since Microsoft saw quite a bit of the Macintosh design (API's, sample code, etc.) during the Mac's development from 1981 to 1984; the intention was to help them write applications for the Mac, and it also gave their system designers a template from which to design Windows. In contrast, the Mac and Lisa designers had to invent their own architectures. Of course, there were some ex- Xerox people in the Lisa and Mac groups, but the design point for these machines was so different that we didn't leverage our knowledge of the Xerox systems as much as some people think.

The hardware itself was an amazing step forward as well. It offered an all-in-one design, four-voice sound, small footprint, clock, auto-eject floppies, serial ports, and so on. The small, portable, appealing case was a serious departure from the ugly- box-on-an-ugly-box PC world, thanks to Jerry Manock and his crew. Even the packaging showed amazing creativity and passion - do any of you remember unpacking an original 128K Mac? The Mac, the unpacking instructions, the profusely-illustrated and beautifully-written manuals, and the animated practice program with audio cassette were tastefully packaged in a cardboard box with Picasso-style graphics on the side.

Looking Back

In my opinion, the software architectures developed at Xerox for Smalltalk and the Xerox Star were significantly more advanced than either the Mac or Windows. The Star was a tremendous accomplishment, with features that current systems haven't even started to implement, though I see OpenDoc as a strong advance past the Xerox systems. I have great respect for the amazing computer scientists at Xerox PARC, who led the way with innovations we all take for granted now, and from whom I learned a tremendous amount about software design.

Apple could have developed a more complex, sophisticated system rivaling the Xerox architectures. But the Mac had to ship, and it had to be relatively



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



inexpensive - we couldn't afford the time or expense of the "best possible" design. As a "little brother" to the Lisa, the Macintosh didn't have multitasking or protection - we didn't have space for the extra code or stack required. The original Macintosh had extremely tight memory and disk constraints; for example, the Resource Manager took up less than 3,000 bytes of code in the ROM, and the Finder was only 46K on disk. We made many design decisions that we regretted to some extent - even at the time some of us felt disappointed at the compromises we had to make - but if we had done it differently, would we have shipped at all?

The Past and Future

In many ways, the computing world has made remarkably small advances since 1976, and we continually reinvent the wheel. Smalltalk had a nice bytecoded multi-platform virtual machine long before Java. Object oriented programming is the hot thing now, and it's almost 30 years old (see the Simula-67 language). Environments have not progressed much either: I feel the Smalltalk environments from the late 1970's are the most pleasant, cleanest, fastest, and smoothest programming environments I have ever used. Although CodeWarrior is reasonably good for C++ development, I haven't seen anything that compares favorably to the Smalltalk systems I used almost 20 years ago. The Smalltalk systems of today aren't as clean, easy to use, or well- designed as the originals, in my opinion.

We are not even close to the ultimate computing-information- communication device. We have much more work to do on system architectures and user interfaces. In particular, user interface design must be driven by deep architectural issues and not just new graphical appearances; interfaces are structure, not image. Neither Copland nor Windows 95 (nor NT, for that matter) represent the last word on operating systems. Unfortunately, market forces are slowing the development of the next revolution. Still, I think you can count on Apple being the company bringing these improvements to next generation systems.

I'm sure some things I remember as having originated at Apple were independently developed elsewhere. But the Mac brought them to the world.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



It's The Moustache That Matters

Author: Andy Hertzfeld
Date: September 1980
Characters: Burrell Smith, Tom Whitney
Topics: Management, Personality
Summary: Burrell wants to get promoted to engineer

Burrell Smith was hired into Apple in the fall of 1978 in the lowly position of service technician, one of the lowest paying jobs at the company. Even though he'd been doing genius quality work as a hardware designer on the Macintosh project for a while now (more than nine months), and he was even filling in for Woz on the low cost Apple II project, he still wasn't officially promoted to engineer as he requested, which was getting pretty frustrating.

Burrell started thinking about what it would take to get promoted. It obviously wasn't a matter of talent or technical skill, since he was already far more accomplished in that regard than most of the other hardware engineers. It wasn't a matter of working harder, since Burrell already worked harder and was more productive than most of the others. Finally, he noticed something that most of the other engineers had in common that he was lacking: they all had fairly prominent moustaches. And the engineering managers tended to have even bigger moustaches. Tom Whitney, the engineering VP, had the largest moustache of all.

So Burrell immediately started growing his own moustache. It took around a month or so for it to come in fully, but finally he pronounced it complete. And sure enough, that very afternoon, he was called into Tom Whitney's office and told that he was promoted to "member of technical staff" as a full-fledged engineer.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Good Earth

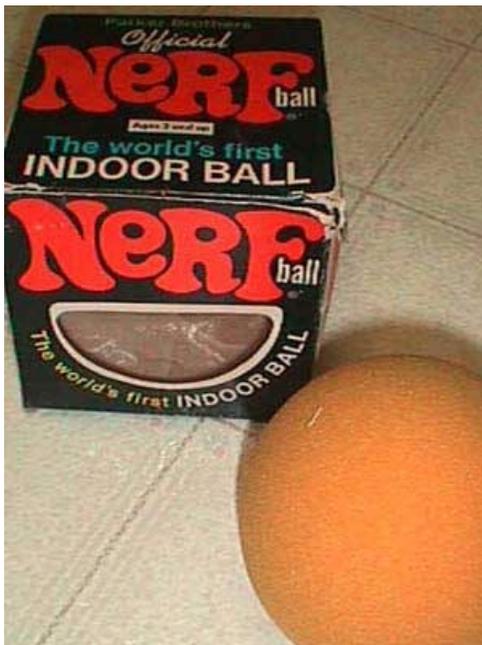
Author: Andy Hertzfeld

Date: October 1980

Characters: Bud Tribble, Steve Jobs, Jef Raskin, Bill Atkinson,
Rod Holt, Burrell Smith, Joanna Hoffman, Brian Howard,
Steve Wozniak, Mike Scott

Topics: Apple Spirit, Origins, Management, Buildings

Summary: The original Mac team's original office



Nerf balls were a favorite toy



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



In 1979 and 1980, Jef Raskin's Macintosh project was a four person research effort with a tenuous existence. It wasn't considered to be very important within Apple, and was almost cancelled a couple of times. When Apple had another major reorganization in the fall of 1980, it was terminated again, but Jef pleaded with Mike Scott and Mike Markkula for more time, and was granted three more months to show that he was really onto something. As part of the re-org, the four person Macintosh team (Jef Raskin, Brian Howard, Burrell Smith, Bud Tribble, soon to be joined by Joanna Hoffman) relocated to a small office building a few blocks from the main Apple campus.

The new office, located at 20863 Stevens Creek Boulevard, was called the "Good Earth" building, because it was adjacent to a Good Earth restaurant. In fact, the office used to be Apple's very first office in Cupertino, after they moved out of Steve Jobs' parents' house, and was later used as the first office of the Lisa project, when the Lisa team had fewer than ten employees. The Mac team moved in, outfitting it with lots of bean bag chairs and all kinds of interesting toys.

Jef was very playful and always encouraged his team to express themselves creatively, so the office quickly began to look more like a day care center than an engineering lab, overflowing with playthings. Periodically, work would cease and the entire team, plus any visitors who might be on the premises, would play some organized game, usually led by Jef and Brian.

The favorite game, which was usually played at least once a day, just after lunch, was a form of tag played with Nerf balls. There were dozens of brightly colored Nerf balls scattered around the office. The rules would be improvised, but usually the person who was "it" had to confer it-hood on someone else by hitting them with a Nerf ball. This inspired everyone to surround their work area with barricades made out of cardboard, to provide cover during the game, making part of the office look like a cardboard maze.

Jef and Brian were both serious musicians, so the office was also littered with a variety of musical instruments, sometimes erupting in spontaneous concerts. Another one of Jef's interests was model airplanes and automobiles, especially radio controlled ones. It wasn't unusual to see a radio controlled car dart underneath your desk, and occasionally everyone would go outside to see the maiden voyage of the latest plane.

Jef was writing his "Book of Macintosh" during much of 1979 and all of 1980, articulating his vision in ever finer detail. Burrell's 6809-based prototype came alive in the early part of 1980, but then he went off to work on the low cost Apple II project. Jef hired Marc Lebrun to write software in early 1980, but Marc was more interested in Lisp machines than a limited memory microcomputer like the Mac, so nothing much happened until he was replaced by Bud Tribble in September 1980.

Bud knew Jef from USCD, and was also good friends with Bill Atkinson. They had a part time, two person consulting company together in Seattle called Synaptic Systems while they were both graduate students. Bill and Jef convinced Bud to take a one year leave of absence from the M.D./Ph.D. program he was pursuing at University of Washington at Seattle. Bud was in the fifth year of a seven M.D./Ph.D. program. Instead of returning to med school, Bud moved into a spare room at Bill Atkinson's house, and started work on the Mac project at Apple. He



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



quickly began to breathe life into Burrell's languishing prototype, writing some graphics routines for the 6809.

So even though the Mac project had been going for more than a year, the move to Good Earth in October 1980 came at an interesting time, with a new but limited lease on life, and software finally starting to happen. But the Good Earth era was rather short lived.

Around two months after the move, Bud convinced Burrell to consider using the 68000 processor instead of the 6809. Burrell came up with a brilliant design, catching the attention of Steve Jobs. Steve took over the project and quickly recruited most of the early Apple II crew that he trusted, including Steve Wozniak and Rod Holt, and moved the project to larger offices a half mile away, in Texaco Towers.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Black Wednesday

Author: Andy Hertzfeld
Date: February 1981
Characters: Andy Hertzfeld, Mike Scott, Steve Jobs, Dick Huston,
Donn Denman, Rick Aurricchio, Jef Raskin, Burrell Smith,
Brian Howard
Topics: Origins, Apple II, Management
Summary: A shakeup in Apple II engineering frees me up to work on
the Macintosh

I could tell there was something wrong from the moment I stepped into the building, on the morning of Wednesday, February 25th, 1981. Instead of the normal office buzz, there was a muted sadness hanging in the air. People were standing around, huddled in small groups. I ran into Donn Denman, who had a cubicle near mine, and asked him what was going on.

"Didn't you hear? Scotty fired almost half of the Apple II engineering team this morning. He started calling people into his office since around 9am, one at a time, and telling them that they were being fired. I think over thirty people have been fired so far. No one knows why, or who's going to be next. There's going to be a meeting out back around noon when he's supposed to tell us what's going on."

Apple had just gone public a couple of months ago, and it was still growing at a frenzied pace. Sales were booming and there was no financial reason to pare back. I wondered what was going on.

"Do you know who they fired?", I asked Donn.

"Yeah, it's amazing. Scotty fired three out of the four managers, so almost everyone's boss is gone. And believe it or not, they fired Rick Aurricchio."



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



I thought that the managers were more or less incompetent, so that didn't bother me, but the Rick Aurricchio part was shocking, since Rick was clearly one of the most talented programmers in the Apple II division. He would usually do a week's worth of work in a day or two, and then spend the rest of the week messing around with whatever caught his fancy, usually one of the latest games. I understood how he could be a management challenge, but it made no sense to fire him. He was my partner on the new DOS 4.0 project, which was just getting underway, and the only other programmer besides me that was working on it, so it was especially distressing that they would fire him so abruptly.

So I joined the ranks of the shell-shocked, and listened numbly to the outdoor meeting where Scotty explained his rationale. He said that the company had grown much too fast over the last year or so, and had made a few key bad hires, who themselves had hired even worse people. He thought the Apple II division had become too complacent, and that we had lost the start-up hustle that was the basis of our success. He wanted to shake us out of our complacency and prune out the bad hires, so we could start growing again in the right direction.

Scotty himself seemed a little shaken and unsure. Some of the other senior executives were standing off to the side, but they didn't participate in the meeting. Woz was there, but didn't look too happy about things. There was a Q&A session at the end of the meeting where a couple employees told Scotty how horribly he handled the situation, but in general everyone seemed listless, as if we didn't know how we should react. Within a few days, everyone was referring to the incident as "Black Wednesday".

Later in the day, I talked to Dick Huston about what had happened. Dick was an early Apple programmer who had written the boot ROM for the disk controller card, who was an astute observer of Apple politics and was friendly with Scotty. He told me that he knew that the purge was going to happen and had even met with Scotty a couple of times in the last week to help him draw up the list of dead weight. He also told me that Scotty had asked for the approval of Mike Markkula and the board of directors, and hadn't received it yet, but decided to go and do it anyway.

I told Dick that I agreed that Apple had made some poor hires over the last year, especially some of the managers, but a Stalin-like purge was not a valid way to run a company. I complained about Rick's firing and told him that the situation made me feel alienated from the company. I was the type of programmer who had to believe in what I was doing, and I wasn't so sure about Apple's values anymore.

When I came in to work the next morning, there was a message on my desk from Mike Scott's secretary, saying that he wanted to talk to me; obviously Dick must have talked to him. I called her back and arranged to show up at his office in an hour. Scotty looked harried, and our conversation was interrupted a few times by various phone calls. Scotty told me that he had heard that I was upset, and thinking about leaving, and wanted me to know that he wanted me to stay. He asked me what he could do to get me excited about Apple again. I told him that I might like to work on the Macintosh, with Burrell and Bud.

Later that afternoon, Scotty's secretary called to tell me that she arranged for me to talk with Steve Jobs. Steve had been involved with the Mac project for more than a month now, and, although I didn't know it at the time, had dismissed



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



the founder of the project, Jef Raskin, the day before, making him take a mandatory leave of absence after Jef had complained about Steve's leadership.

Lots of people at Apple were afraid of Steve Jobs, because of his spontaneous temper tantrums and his proclivity to tell everyone exactly what he thought, which often wasn't very favorable. But he was always nice to me, although sometimes a bit dismissive, in the few interactions that I had with him. I was excited to be talking with him about working on the Mac.

The first thing he said to me when I walked into his office was "Are you any good? We only want really good people working on the Mac, and I'm not sure you're good enough." I told him that yes, I thought that I was pretty good. I was friends with Burrell, and had already helped him out with software a few times.

"I hear that you're creative", Steve continued. "Are you really creative?"

I told him that I wasn't the best judge of that, but that I'd love to work on the Mac, and thought that I'd do a great job. He said he'd get back to me soon about it.

A couple of hours later, around 4:30pm, I was back to work on DOS 4.0 for the Apple II. I was working on low-level code for the system, interrupt handlers and dispatchers, when all of a sudden I notice Steve Jobs peering over the wall of my cubicle.

"I've got good news for you", he told me. "You're working on the Mac team now. Come with me and I'll take you over to your new desk."

"Hey, that's great", I responded. "I just need a day or two to finish up what I'm doing here, and I can start on the Mac on Monday."

"What are you working on? What's more important than working on the Macintosh?"

"Well, I've just started a new OS for the Apple II, DOS 4.0, and I want to get things in good enough shape so someone else could take it over."

"No, you're just wasting your time with that! Who cares about the Apple II? The Apple II will be dead in a few years. Your OS will be obsolete before it's finished. The Macintosh is the future of Apple, and you're going to start on it now!".

With that, he walked over to my desk, found the power cord to my Apple II, and gave it a sharp tug, pulling it out of the socket, causing my machine to lose power and the code I was working on to vanish. He unplugged my monitor and put it on top of the computer, and then picked both of them up and started walking away. "Come with me. I'm going to take you to your new desk."

We walked outside to Steve's silver Mercedes and he dropped my computer into the trunk. We drove a few blocks to the corner of Stevens Creek and Saratoga-Sunnyvale, to a non-descript, brown-shingled, two story office building next to a Texaco station, while Steve waxed eloquent about how great the Macintosh was going to be. We walked up to the second floor, and into an unlocked door. Steve



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



popped my system down on a desk in an office near the back of the building and said, "Here's your new desk. Welcome to the Mac team!", before darting off.

I started looking around the office, and saw Burrell Smith and Brian Howard in the next room, huddled over a logic analyzer connected to a prototype board. I told them what happened and they said Steve had been over earlier, asking them if they thought I was any good. They were happy that I joined the team.

After helping them a bit with the disk diagnostic routines they were trying to debug, I returned to my new desk and looked inside the drawers. I was surprised to see that it was still full of someone else's stuff. In fact, the bottom drawer had all kinds of unusual stuff, including various kinds of model airplanes, and some photography equipment. I later found out that Steve had assigned me to Jef Raskin's old desk, which he hadn't had time to move out of yet.

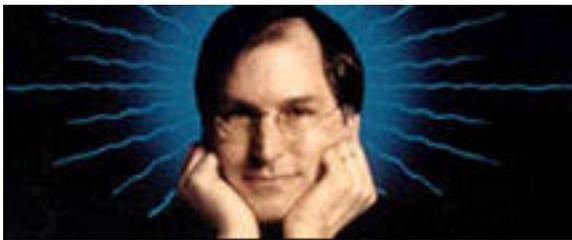


APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Reality Distortion Field

Author: Andy Hertzfeld
Date: February 1981
Characters: Steve Jobs, Bud Tribble
Topics: Management, Personality
Summary: Bud defines Steve's unique talent



Steve Jobs

I officially started on the Mac project on a Thursday afternoon, and Bud Tribble, my new manager and the only other software person on the project, was out of town. Bud was on leave of absence from an M.D.-Ph.D. program and he had to occasionally return to Seattle to keep up his standing in the program.

Bud usually didn't come into work until after lunch, so I met with him for the first time the following Monday afternoon. We started talking about all the work that had to be done, which was pretty overwhelming. He showed me the official schedule for developing the software that had us shipping in about ten months, in early January 1982.

"Bud, that's crazy!", I told him. "We've hardly even started yet. There's no way we can get it done by then."

"I know," he responded, in a low voice, almost a whisper.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



"You know? If you know the schedule is off-base, why don't you correct it?"

"Well, it's Steve. Steve insists that we're shipping in early 1982, and won't accept answers to the contrary. The best way to describe the situation is a term from Star Trek. Steve has a reality distortion field."

"A what?"

"A reality distortion field. In his presence, reality is malleable. He can convince anyone of practically anything. It wears off when he's not around, but it makes it hard to have realistic schedules. And there's a couple of other things you should know about working with Steve."

"What else?"

"Well, just because he tells you that something is awful or great, it doesn't necessarily mean he'll feel that way tomorrow. You have to low-pass filter his input. And then, he's really funny about ideas. If you tell him a new idea, he'll usually tell you that he thinks it's stupid. But then, if he actually likes it, exactly one week later, he'll come back to you and propose your idea to you, as if he thought of it."

I didn't believe that at first, until I saw it happen a couple of times. I guess that's part of the reality distortion field, too.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Square Dots

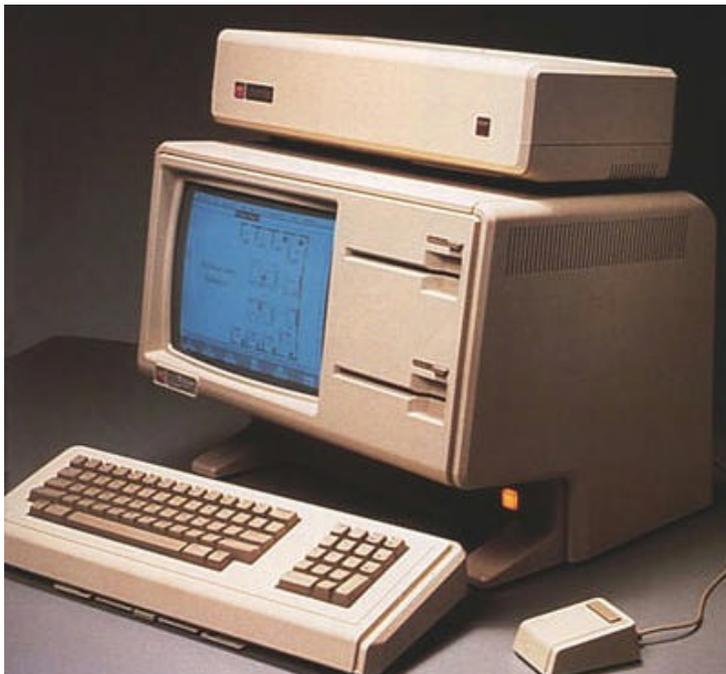
Author: Andy Hertzfeld

Date: April 1981

Characters: Bill Atkinson, Burrell Smith, Rich Page, Wayne Rosing,
Tom Malloy, Bruce Daniels, Paul Baker

Topics: Hardware Design, Management, Lisa

Summary: The Lisa had a different screen resolution than the Macintosh



The Apple Lisa 1 [with Twigggy disk drives]

ANECDOTES ABOUT THE DEVELOPMENT OF APPLE'S ORIGINAL MACINTOSH COMPUTER

<http://www.folklore.org/ProjectView.py?project=Macintosh&detail=medium>
28 January 2004 :: Page 134 of 254



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



From the very beginning, even before it had a mouse, the Lisa was designed to be an office machine, and word processing was considered to be its most important application. In the late seventies, the acid test for an office computer (as compared with a hobby computer) was the ability to display 80 columns of text.

The Lisa team decided to optimize their display for horizontal resolution, in order to be able to display 80 columns of text in an attractive font. The vertical resolution wasn't as important, because vertical scrolling works much better for text than horizontal scrolling. The designers decided to endow Lisa with twice as much horizontal resolution as vertical, using a 720 by 360 pixel display, with pixels that were twice as high as they were wide. This was great for text oriented applications like the word processor, but it made things somewhat awkward for the more graphical applications.

When Burrell redesigned the Macintosh in December 1980 to use the same microprocessor as Lisa, the Motorola 68000, it set off shock waves within Apple. Not only was Burrell's new design much simpler than Lisa, with less than half the chip count, but it also ran almost twice as fast, using an 8 megahertz clock instead of a 5 megahertz clock. Among other advantages was the fact that the Mac's 384 by 256 pixel display had the identical horizontal and vertical resolution, a feature that we called "square dots". Square dots made it easier to write graphical applications, since you didn't have to worry about the resolution disparity.

Bill Atkinson, the author of QuickDraw and the main Lisa graphics programmer, was a strong advocate of square dots, but not everyone on the Lisa team felt the same way. Tom Malloy, who was Apple's first hire from Xerox PARC and the principal author of the Lisa word processor, thought that it was better to have the increased horizontal resolution. But Burrell's redesign moved the debate from the theoretical to the pragmatic, by creating a square dots machine to compare with the Lisa.

The Lisa hardware was scheduled to go through a final round of design tweaks, and Bill tried to convince them to switch to square dots. He mentioned his desire to Burrell, who responded by working over the weekend to sketch out a scaled up version of the Macintosh design that featured a full 16-bit memory bus with a 768 by 512 display and square dots, that would also run twice as fast as the current Lisa design. Bill convinced the Lisa engineering manager, Wayne Rosing, that he should at least consider adopting some of Burrell's ideas, and arranged for the leadership of the Lisa team to get a demo of the current Macintosh, and learn about Burrell's new scaled-up design.

Wayne Rosing led a delegation of his top hardware and software guys over to Texaco Towers for a demo on a Monday afternoon, including hardware guys Rich Page and Paul Baker and software manager Bruce Daniels. We ran various graphics demos, with Bill Atkinson doing the talking, and then Burrell gave a presentation on the Mac design, and his ideas for scaling it up to 768 by 512 display. Everyone seemed pretty impressed and Bill was optimistic that they would make the change.

After a few days, Bill told us the disappointing news that Wayne had decided that there wasn't enough time to embark on such a radical redesign, since at that point the Lisa was scheduled to ship in less than a year. It ended up shipping around two years later with the original 720 by 360 resolution and



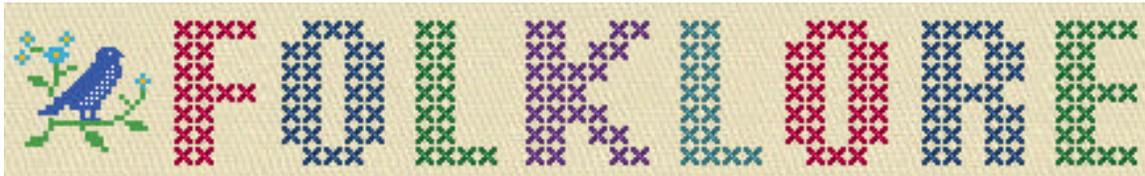
APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



relatively slow microprocessor, which became a problem when Apple decided to offer a Macintosh compatibility mode for Lisa in 1984. The emulation software didn't try to compensate for the different resolutions, so the applications were distorted by the resolution disparity, almost like looking at a fun-house mirror. It would have continued to be a problem for Apple if the Lisa hadn't been discontinued in 1985.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Bicycle

Author: Andy Hertzfeld
 Date: April 1981
 Characters: Rod Holt, Steve Jobs, Mac Team, Jef Raskin
 Topics: Management, Apple Spirit
 Summary: Rod wants to change the name of the project



Logo for Mac University Consortium

Jef Raskin chose the name "Macintosh", after his favorite kind of apple, so when Jef was forced to go on an extended leave of absence in February 1981, Steve Jobs and Rod Holt decided to change the name of the project, partially to distance it from Jef. They considered "Macintosh" to be a code name anyway, and didn't want us to get too attached to it.

Apple had recently taken out a two page ad in Scientific American, featuring quotes from Steve Jobs about the wonders of personal computers. The ad explained



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



how humans were not as fast runners as many other species, but a human on a bicycle beat them all. Personal computers were "bicycles for the mind."

A month or so after Jef's departure, Rod Holt announced to the small design team that the new code name for the project was "Bicycle", and that we should change all references to "Macintosh" to "Bicycle". When we objected, thinking "Bicycle" was a silly name, Rod thought that it shouldn't matter, "since it was only a code name".

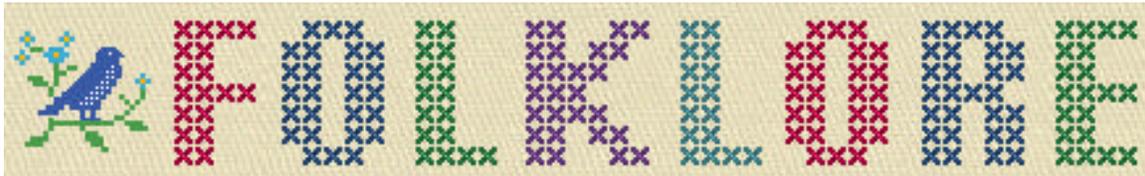
Rod's edict was never obeyed. Somehow, Macintosh just seemed right. It was already ingrained with the team, and the "Bicycle" name seemed forced and inappropriate, so no one but Rod ever called it "Bicycle". For a few weeks, Rod would reprimand anyone who called it "Macintosh" in his presence, but the new name never acquired any momentum. Finally, around a month after his original order, after someone called it "Macintosh" again, he threw up his hands in exasperation and told us, "I give up! You can call it Macintosh if you want. It's only a code name, anyway."

But it was a code name that proved to be sturdy and resilient. In the Fall of 1982, Apple paid tens of thousands of dollars to a marketing consulting firm to come up with a themed set of names for Lisa and Macintosh. They came up with lots of ideas, including calling the Mac the "Apple 40" or the "Apple Allegro". After hearing all the suggestions, Steve and the marketing team decided to go with Lisa and Macintosh as the official names. They did manage to reverse engineer an acronym for Lisa, "Local Integrated Systems Architecture", but internally we preferred the recursive "Lisa: Invented Stupid Acronym", or something like that. Macintosh seemed to be acronym proof.

But there was still a final hurdle to clear - the name was too close to a trademark from the McIntosh stereo company. I'm not sure how the situation was resolved (I suspect that Apple paid them a modest amount), but toward the end of the retreat in January 1983, Steve announced to the team that we had gotten rights to use the name. He dashed a champagne bottle against one of the prototypes, and declared, "I christen thee Macintosh!"



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Diagnostic Port

Author: Andy Hertzfeld
Date: July 1981
Characters: Steve Jobs, Jef Raskin, Burrell Smith, Brian Howard,
Steve Wozniak, Rod Holt
Topics: Hardware Design, Management
Summary: Burrell tried to sneak in some hardware expandability

Expandability, or the lack thereof, was far and away the most controversial aspect of the original Macintosh hardware design. Apple co-founder Steve Wozniak was a strong believer in hardware expandability, and he endowed the Apple II with luxurious expandability in the form of seven built-in slots for peripheral cards, configured in a clever architecture that allowed each card to incorporate built-in software on its own ROM chip. This flexibility allowed the Apple II to be adapted to a wider range of applications, and quickly spawned a thriving third-party hardware industry.

But Jef Raskin had a very different point of view. He thought that slots were inherently complex, and were one of the obstacles holding back personal computers from reaching a wider audience. He thought that hardware expandability made it more difficult for third party software writers since they couldn't rely on the consistency of the underlying hardware. His Macintosh vision had Apple cranking out millions of identical, easy to use, low cost appliance computers and since hardware expandability would add significant cost and complexity it was therefore avoided.

Apple's other co-founder, Steve Jobs, didn't agree with Jef about many things, but they both felt the same way about hardware expandability: it was a bug instead of a feature. Steve was reportedly against having slots in the Apple II back in the days of yore, and felt even stronger about slots for the Mac. He decreed that the Macintosh would remain perpetually bereft of slots, enclosed in



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



a tightly sealed case, with only the limited expandability of the two serial ports.

Mac hardware designer Burrell Smith and his assistant Brian Howard understood Steve's rationale, but they felt differently about the proper course of action. Burrell had already watched the Macintosh's hopelessly optimistic schedule start to slip indefinitely, and he was unable to predict when the Mac's pioneering software would be finished, if ever. He was afraid that Moore's Law would make his delayed hardware obsolete before it ever came to market. He thought it was prudent to build in as much flexibility as possible, as long as it didn't cost too much.

Burrell decided to add a single, simple slot to his Macintosh design, which made the processor's bus accessible to peripherals, that wouldn't cost very much, especially if it wasn't used. He worked out the details and proposed it at the weekly staff meeting, but Steve immediately nixed his proposal, stating that there was no way that the Mac would even have a single slot.

But Burrell was not that easily thwarted. He realized that the Mac was never going to have something called a slot, but perhaps the same functionality could be called something else. After talking it over with Brian, they decided to start calling it the "diagnostic port" instead of a slot, arguing that it would save money during manufacturing if testing devices could access the processor bus to diagnose manufacturing errors. They didn't mention that the same port would also provide the functionality of a slot.

This was received positively at first, but after a couple weeks, engineering manager Rod Holt caught on to what was happening, probably aided by occasional giggles when the diagnostic port was mentioned. "That thing really a slot, right? You're trying to sneak in a slot!", Rod finally accused us at the next engineering meeting. "Well, that's not going to happen!"

Even though the diagnostic port was scuttled, it wasn't the last attempt at surreptitious hardware expandability. When the Mac digital board was redesigned for the last time in August 1982, the next generation of RAM chips was already on the horizon. The Mac used 16 64Kbit RAM chips, giving it 128K of memory. The next generation chip was 256Kbits, giving us 512K bytes instead, which made a huge difference.

Burrell was afraid the 128Kbyte Mac would seem inadequate soon after launch, and there were no slots for the user to add RAM. He realized that he could support 256Kbit RAM chips simply by routing a few extra lines on the PC board, allowing adventurous people who knew how to wield a soldering gun to replace their RAM chips with the newer generation. The extra lines would only cost pennies to add.

But once again, Steve Jobs objected, because he didn't like the idea of customers mucking with the innards of their computer. He would also rather have them buy a new 512K Mac instead of them buying more RAM from a third-party. But this time Burrell prevailed, because the change was so minimal. He just left it in there and no one bothered to mention it to Steve, much to the eventual benefit of customers, who didn't have to buy a whole new Mac to expand their memory.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Signing Party

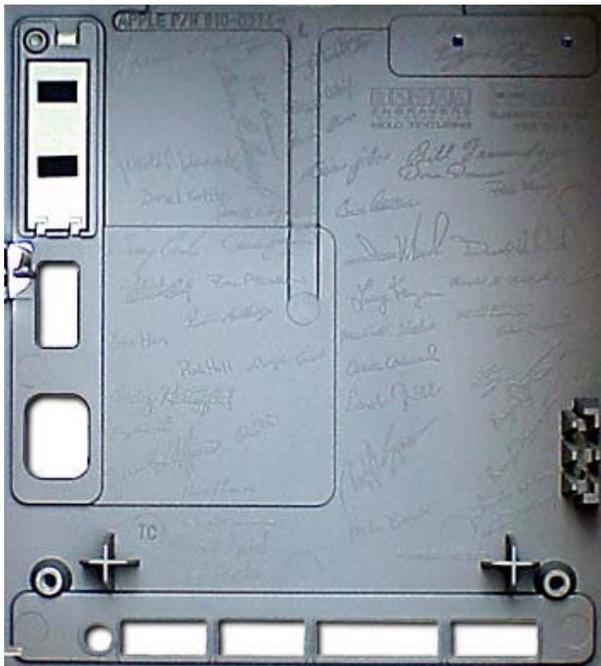
Author: Andy Hertzfeld

Date: February 1982

Characters: Steve Jobs, Bob Belleville, Jerry Manock, Mac Team,
Bud Tribble, Jef Raskin, Mike Boich, Rod Holt, Steve Wozniak

Topics: Management, Apple Spirit

Summary: The artists sign their work



The back side of the Macintosh case



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



The component of the Macintosh hardware that had the longest lead time was the hard tool that molded its distinctive plastic case. After tweaking the case design for more than six months and building a small production run of 50 units with a soft-tooled case, the final design was ready to go out for hard tooling toward the end of February 1982, so we could meet the ship date that we were aiming for at the time, which was January 1983.

The Mac team had a complicated set of motivations, but the most unique ingredient was a strong dose of artistic values. First and foremost, Steve Jobs thought of himself as an artist, and he encouraged the design team to think of ourselves that way, too. The goal was never to beat the competition, or to make a lot of money; it was to do the greatest thing possible, or even a little greater. Steve often reinforced the artistic theme; for example, he took the entire team on a field trip in the spring of 1982 to the Louis Comfort Tiffany museum, because Tiffany was an artist who learned how to mass produce his work.

Since the Macintosh team were artists, it was only appropriate that we sign our work. Steve came up with the awesome idea of having each team member's signature engraved on the hard tool that molded the plastic case, so our signatures would appear inside the case of every Mac that rolled off the production line. Most customers would never see them, since you needed a special tool to look inside, but we would take pride in knowing that our names were in there, even if no one else knew.

We held a special signing party after one of our weekly meetings toward the end of February 1982. Jerry Mannock, the manager of the industrial design team, spread out a large piece of drafting paper on the table to capture our signatures. Steve gave a little speech about artists signing their work, and then cake and champagne were served as he called each team member to step forward and sign their name for posterity. Burrell had the symbolic honor of going first, followed by members of the software team. It took forty minutes or so for around thirty-five team members to sign. Steve waited until last, when he picked a spot near the upper center and signed his name with a flourish.

We were aware that the team was still growing rapidly, and in a few months there would be a new crop of key contributors that also deserved to sign the case. We decided to draw the line at the date of the signing party, and not to let new signatures come in later, but we knew it would be tough to stick to that. We also wanted to add the signatures of a few major contributors who had left the project: Steve Wozniak, Jef Raskin and Bud Tribble. But that was supposed to be it.

Over the next few months, a few more signatures of people who weren't on the team at the time of the signing party managed to make it into the case. For a while Rod Holt held the line, but eventually Bob Belleville, who hired on in April 1982 as the software manager but soon became the overall engineering manager when Rod Holt retired, decided to add his own name. He also snuck in a few other key people, like marketing manager Mike Murray and original evangelist Mike Boich, who started before he did and who otherwise would have just missed the cut-off.

And then, over time, names gradually began to disappear for practical reasons, as Apple changed the case to make it easier to manufacture. Some details were changed even before first ship, partially obscuring some of the signatures. Each



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



time the case was revised, more names were left off, as dictated by the nature of the revision, until a substantial number of them were gone. I'm not sure which model was the last to have any names at all, but I'm pretty sure that the Macintosh Classic, from the early nineties, didn't have any left.

Jimell A. Azama
 Gerald C. Marock
 Daniel Kottke
 Barry Carl
 Charles J. Wynn
~~Walt Bull~~
 Bruce Horn
 Andy Hertzfeld
 Angelic L.
 Joana Kainoff
 HAP HORN
 M.E. McCammon
 Burrell Smith
 Jef Raskin

Stephen J. Jobs
 Bill Atkinson
 Vicki Milledge
 George Crow
 Rod Holt
 EDEN
 Steve Jobs

Susan Robertson
 Joan Mach
 Larry Kenyon
 Martin P. Haller
 Calotte Askeland
 Larry J. Judd
 Christopher Young
 Mike Boich

Stuart Hill
 Roger Alejo
 Patricia Sharp
 Bill Fernandez
 Don Demmer
 Patti King
 David H. Roth
 Ronald N. Nicholson
 Matt Carter
 Robert J. Bellville
 Randy Wigginton
 Linda Wilkin
 Michael Murray

Lynn Sakahadi
 Benjamin L. Perry

FEBRUARY 10, 1982



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



And Another Thing . . .

Author: Andy Hertzfeld
Date: March 1982
Characters: Rich Page, Steve Jobs, Bill Atkinson, Burrell Smith,
Andy Hertzfeld, Larry Tesler
Topics: Lisa, Management, Personality, Personality Clashes
Summary: There was some friction between the Mac and Lisa teams

By early 1982, the Macintosh was beginning to be acknowledged as a significant project within Apple, instead of a quirky research effort, but it still remained somewhat controversial. Since the Mac was sort of like a Lisa that was priced like an Apple II, it was seen as potential competition from both groups. Also, our leader Steve Jobs had a habit of constantly boasting about the superiority of the Mac team, which tended to alienate everybody else.

Larry Tesler, who came to Apple from Xerox PARC in the summer of 1980, was the manager of the Lisa Application Software team. He understood and appreciated the potential of the Macintosh and was very supportive of the project. He was concerned that some of the Lisa team didn't share his enthusiasm, and thought that it would be helpful for us to demonstrate the Mac to his team and talk about our plans with them. He arranged for Burrell Smith and me to give a demo during a lunch-time meeting.

By this point, we had stand-alone Macintosh prototypes that no longer depended on an umbilical cord to a hosting Lisa. We didn't have the real plastic cases yet, but we were able to house the prototypes in plastic boxes of around the same size that were a passable imitation. The demo software environment was based on the "Lisa Monitor", a simple operating system cooked up by one of the main Lisa architects, Rich Page, that I got running on the Macintosh. The monitor was based on the UCSD Pascal system Filer and offered a simple, menu-based UI. We were able to boot the Mac into the monitor from an Apple II floppy, and then use it to launch various demo programs.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Burrell and I set up the prototype in a large conference room in the Lisa building. The Lisa applications team was seated around the table, but quite a few other Lisa team members had also gathered around, standing room only, perhaps twenty-five people in all. Larry Tesler gave us a nice introduction, and then we booted up the prototype and started to run through various demos while explaining the capabilities of the machine. Everything seemed to be going pretty well, when suddenly there was a loud, insistent knock at the conference room door.

Before anyone could respond, the door was flung open, and in strode Rich Page, the systems wizard who was one of the main designers of the Lisa. Rich was a tall, bearded, ursine engineer, equally adept at hardware and software, who was responsible for getting Lisa to use the 68000, and had personally ported or created many of the tools that both the Macintosh and Lisa teams were using. But I had never seen him looking as angry as he was at the moment.

"You guys don't know what you're doing!", he began to growl, obviously in an emotional state of mind, "The Macintosh is going to destroy the Lisa! The Macintosh is going to ruin Apple!!!"

Burrell and I didn't know how to respond, and neither did anyone else in the room. Larry Tesler gave me an embarrassed glance, trying to figure out what to do. But Rich wasn't particularly interested in a response, he just wanted to vent his frustration.

"Steve Jobs wants to destroy Lisa because we wouldn't let him control it", Rich continued, almost looking like he was going to start crying. "Sure, it's easy to throw a prototype together, but it's hard to ship a real product. You guys don't understand what you're getting into. The Mac can't run Lisa software, the Lisa can't run Mac software. You don't even care. Nobody's going to buy a Lisa because they know the Mac is coming! But you don't care!"

With that, he turned around and strode out of the conference room, as quickly as he had come in. He slammed the door as he left, with the noise reverberating ominously in the stunned silence. There was some nervous laughter, but nobody knew what to say. Larry Tesler started to apologize, explaining that Rich didn't speak for most of the Lisa team, when suddenly the door was flung open again and Rich Page was back, just as angry as before.

"And another thing...", he said, before pausing to look directly at Burrell and myself. "I don't have any problem with you, I know it's not your fault. Steve Jobs is the problem. Tell Steve that I think that he's destroying Apple!" Once again, he turned around and left abruptly, slamming the door for a second time. We steeled our ourselves, wondering if he was going to return for a third round.

But this time, Larry was able to finish apologizing, and then we finished the demo quickly and held a brief question and answer session, with everyone still a bit shell-shocked from the unexpected outburst. We told Steve Jobs about Rich Page's oration later that afternoon, and he just shrugged, "That's Rich Page for you. He'll get over it."

The next morning Bill Atkinson called and told me that Rich Page felt bad about what happened and he wanted to take Burrell and me out to lunch to apologize. So



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



that afternoon, the four of us went out for a long lunch, where Bill explained that Rich was just trying to do what he thought was right, and he didn't intend to get so emotional. Rich told us that he really appreciates that Burrell and I were doing great work for the company, but he was frustrated that Steve was such a loose cannon, and wasn't working for our mutual success. We left on decent terms, but in the back of my mind I was still worried that such obvious resentment would be a problem for us in the future.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



I'll Be Your Best Friend

Author: Andy Hertzfeld

Date: August 1979

Characters: Burrell Smith, Andy Hertzfeld, Wendell Sander, Steve Wozniak

Topics: Origins, Personality, Hardware Design, Apple II

Summary: Burrell Smith was creative in more than just engineering



Andy and Burrell in January 1983



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Toward the end of my first week as an Apple employee in August 1979, I noticed that someone had left a black binder on my desk, with a hand-written title that read, "Apple II: Principles of Operation". It contained a brilliant, concise description of how the Apple II hardware worked, reverently explaining details of Woz's epic, creative design hacks, in a clearer fashion than I'd ever read before. I didn't know who left it there, but the title page said it was written by "Burrell C. Smith".

Later that day, in the late afternoon, I was approached by a young, animated, slightly nervous guy with long, straight, blond hair, who entered my cubicle and walked right up to me.

"Are you Andy Hertzfeld? Wow, it's amazing to meet you. I read your articles in Call A. P. P. L. E. and Dr. Dobb's. Apple's lucky they got you to work here. I want to shake your hand."

With exaggerated formality, he extended his right arm stiffly, almost in a parody of a handshake offer. "I'm Burrell. Burrell Carver Smith. Pleased to meet you. I wrote that manual I left on your desk.", he said, pointing to the black binder on my desk. We shook hands and then he suddenly turned around nervously and darted off, without explanation. "See you later!", he said, without looking back.

My cubicle in Bandley 1 was in the hardware engineering section, since my first project was writing the firmware for the Silentype thermal printer. It was across the aisle from Wendell Sander's office. Wendell was the designer of the Apple III and an extremely brilliant and seasoned engineer, who used to design RAM chips for Fairchild and understood the Apple II hardware design inside out. All the other hardware engineers on the team, except for Woz, usually came to Wendell for advice.

I began to notice that Burrell, even though he was supposed to be working in the service department in a different building, often hung around outside of Wendell's office, sometimes for hours at a time, waiting for Wendell to have a free moment, so he could ask him to verify his latest insight about the Apple II timings. Sometimes, when Wendell was busy, he would try the insight out on me instead, or discuss a fine point of the Apple II firmware. Soon, we started to occasionally go out for lunch together.

The first time we went to lunch, I found out that Burrell's creativity extended beyond his engineering work. He would often try to convince our waitress to concoct variations of the standard fare on the menu, thinking of something different every time.

For example, after he successfully persuaded a waitress to divide his pizza toppings into thirds, he asked her to do fifths the next time. Or he would sometimes try to order mixed sodas as if they were cocktails, in ever varying proportions, like three quarters Coke, and one quarter Sprite. Often, the waitress would balk but Burrell was sometimes charming enough to eventually convince her to comply. He would also obsess on certain foods, becoming fixated on Bulgarian Beef sandwiches from Vivi's for a while, and then a Pineapple Pizza phase (see Pineapple Pizza), evolving to his most enduring favorite, sushi, which provided a new range of interesting choices and combinations.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Burrell also had a distinctive way of expressing himself, applying technical jargon to ordinary life (like a situation that was "meta-stable", or someone being a "state machine"), mixed with a dash of baby talk (like adding plurals to people's names). An attractive woman was referred to as a "good prototype", or a "good proto" for short. Burrell had a great sense of humor and periodically performed hilarious impressions of everyone else on the team, caricaturing their personality quirks with an incisive phrase or nickname. He also liked to make fun of various language conventions; for example, if he was pleased with new software, he'd say "Happiness comma software".

One of his favorite expressions was "I'll be your best friend". He offered "best friendship" for a wide range of activities, like making some change in the software for him or getting him a Coke from the gas station. "Best friendship takes place", he would declare if pleased with the results. He also had a habit of reducing things to initials, like "B. F. R." for "Best Friendship Relationship".

Once, right after Burrell conferred best friendship upon me, I heard him offer best friendship to someone else, for a different favor. "Wait a second", I challenged Burrell, "How can you give out best friendship to someone else? There can be only one best friend at a time, can't there?"

Burrell had a quick reply, delivered with a smile. "Of course there can be only one "best" friend at a given instant of time. But best friendship relationships may be highly dynamic. The average length of a best friendship is three to five milliseconds. So there's no problem in having a new B. F. R. a second or two later."



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



We'll See About That

Author: Andy Hertzfeld
Date: November 1979
Characters: Burrell Smith, Bill Atkinson, Jef Raskin, Steve Wozniak
Topics: Hardware Design, Technical, Apple II
Summary: Burrell proves his mettle with the 80k language card.



Apple II Language Card

Burrell Smith was a 23 year old, self-taught engineer, without a college degree, who was drawn to Apple by the sheer elegance of the Apple II design. He was hired into Apple in February 1979, as Apple employee #282, a lowly service technician responsible for fixing broken Apple IIs that were sometimes returned by customers. As he debugged broken logic boards, sometimes more than a dozen in a single day, he began to develop a profound respect and empathy for Steve Wozniak's unique, creative design techniques.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Meanwhile, the Lisa team had been writing their first code in Pascal, running on Apple IIs, because the Lisa hardware wasn't ready yet. They had been at it for almost a year, and they had written more code than would fit in the 64 Kbytes of memory in a standard Apple II. In fact, the Apple II only had 48 Kbytes on its main board, but it used a "language" card to give it an extra 16 Kbytes used to run Pascal. To accomplish this, the language card had to "bank switch" its RAM over the ROM on the Apple II motherboard.

Bill Atkinson was the main programmer for both the Apple II Pascal system as well as the new Lisa system. He was in the service department picking up some extra language cards when Burrell heard him lamenting about overflowing the Apple II's memory limitations.

"Well, why don't you add more memory to the language card?", Burrell suggested.

Bill was intrigued, but he complained, "You can't add any more memory because we're out of address space. 64K is the limit of what we can address."

Burrell had already thought of that. "Well, the language card is already bank-switching the RAM, even double-banking the last 2K where the monitor ROM is. We'll just make it bank-switch another bank."

Bill was enthusiastic, so Burrell built him a prototype while Bill modified the Pascal run-time to support the extra bank switching. It worked like a charm, so soon Burrell was busy manufacturing 80K language cards for all the Lisa programmers.

Around this time, Bill ran into Jef Raskin. Jef had been writing a series of papers about a consumer-oriented computer that would be extremely inexpensive and radically easy to use. He was ready to start building a hardware prototype so he was looking for a talented hardware designer who could pull off his vision of a brutally simple, ultra low cost machine.

"I've got someone who you ought to meet", Bill told Jef. He made arrangements to bring Burrell over to Jef's house in Cupertino over the weekend.

Bill and Burrell showed up at Jef's house at the appointed time. Bill introduced Burrell to Jef, saying. "Jef, this is Burrell. He's the guy who's going to design your Macintosh for you."

"We'll see about that", Jef replied. "We'll see about that."



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Scrooge McDuck

Author: Andy Hertzfeld

Date: February 1980

Characters: Burrell Smith, Andy Hertzfeld, Jef Raskin, Tom Whitney,
Bob Bishop, Cliff Huston, Dick Huston

Topics: Origins, Hardware Design, Prototypes

Summary: The very first image on the very first Macintosh



This image is similar to the first image ever displayed on a Macintosh



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Burrell Smith liked to do intensive design work over the Christmas break, so the very first prototype of the very first Macintosh sprung to life early in the first month of the new decade, in January 1980. It wasn't really a stand-alone computer yet, as the prototype resided on an Apple II peripheral card, but it already contained the essential hardware elements of Jef Raskin's Macintosh dream: a Motorola 6809E microprocessor, 64K of memory, and a 256 by 256 bit-mapped graphic frame buffer, which was hooked up to a cute, tiny 7 inch black and white display. Burrell used the Apple II host to poke values into the memory of the prototype, so he could initialize the control registers and run small programs with the 6809.

I went out to lunch with Burrell a few weeks later and, knowing my appreciation for Woz-like hardware hacks, he explained the crazy way that he contrived for the Apple II to talk with the prototype. He didn't want to waste time designing and wiring up hardware to synchronize the memory of the two machines, since that wouldn't be needed by the real product. Instead, he delegated the memory synchronization to the software, requiring the Apple II to hit a special memory address to tell the prototype how many microseconds later to grab data off of the common data bus. It was weird enough to make me interested in seeing if it really worked.

By now, Burrell thought that he had the graphics running properly, but he wasn't really sure; he still needed to write some software to try it out. I told him that I'd look into it when I had some time. He gave me a copy of a handwritten page that contained the magic addresses that I'd have to use, hoping that I'd get around to it soon.

Burrell's DMA instructions I was used to coming back to the lab at Apple after dinner, to see if anything interesting was going on and working on various extra-curricular projects. I had some spare time that night, so I got out Burrell's instructions and wrote an Apple II (6502) assembly language routine to do the necessary bit-twiddling to transfer whatever was on the Apple II's hi-res graphic display to the Mac prototype's frame-buffer, using Burrell's unusual synchronization scheme.

One of my recent side projects involved using Woz's new, one-to-one interleave floppy disk routines to make very fast slideshow disks on the Apple II. I had just made one full of Disney cartoon characters, that were scanned by Bob Bishop, one of the early Apple software magicians. Bob adored the work of Carl Barks, the Disney artist who specialized in Donald Duck, and he had scanned dozens of Barks' Donald Duck images for the Apple II. I selected an image of Scrooge McDuck sitting on top of a huge pile of money bags, blithely playing his fiddle, with a big grin on his beak. I'm not sure why I picked that one, but it seemed to be appropriate for some reason.

Even though it was starting to get late, I was dying to see if my routine was working properly, and it would be very cool to surprise Burrell when he came in the next day with a detailed image on the prototype display. But when I went to try it, I noticed that Burrell's Apple didn't have a disk controller card, so there was no way to load my program. Damn! I couldn't shut the computer down to insert the card, because I didn't know how to reinitialize the Macintosh board after power-up; Burrell hadn't left the magic incantation for that. I thought I was stuck, and would have to wait until Burrell came in tomorrow.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



The only other person in the lab that evening was Cliff Huston, who saw the trouble I was having. Cliff was another early Apple employee, who was Dick Huston's (the heroic programmer who wrote the 256-byte Apple II floppy disc boot ROM) older brother and an experienced, somewhat cynical technician. I explained the situation to him and was surprised when he started to smile.

Cliff told me that he could insert a disk controller card into Burrell's Apple II with the power still on, without glitching it out, a feat that I thought was miraculous - you'd have to be incredibly quick and steady not to short-circuit any of the contacts while you were inserting it, running the risk of burning out both the Apple II and the card. But Cliff said he'd done it many times before: all that was required was the confidence that you could actually do it. So I crossed my fingers as he approached Burrell's Apple like a samurai warrior, concentrating for a few seconds before holding his breath and slamming the disk card into the slot with a quick, stacatto thrust.

I could barely make myself look, but amazingly enough Burrell's machine was still running, and the disk booted up so I could load the Scrooge McDuck image and my new conversion routine. And even more suprising, my routine actually worked the first time, displaying a crisp rendition of Uncle Scrooge fiddling away on the Mac's tiny monitor. The Apple II only had 192 scan-lines, while the embryonic Macintosh had 256, so I had some extra room at the bottom where I rendered the message "Hi Burrell!" in a nice-looking twenty-four point, proportional font.

By the time I came in the next morning, an excited Burrell had already showed the image to everyone he could find, but then he accidentally reset the prototype somehow, and didn't know how to get the image back on the screen. I loaded it again so he could show it to Tom Whitney, the engineering VP. I think Jef was pretty pleased to see his new computer start to come alive, but I don't think he was very happy about me giving the demo, since he thought I was too much of a hacker, and I wasn't supposed to be involved with his pet project.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



I Invented Burrell

Author: Andy Hertzfeld
Date: 1981
Characters: Burrell Smith, Jef Raskin
Topics: Personality
Summary: Burrell imitates Jef

Burrell Smith had a great sense of humor, and he was capable of performing devastating impersonations of everybody else on the Mac team, especially the authority figures.

Whatever idea that you came up with, Jef Raskin had a tendency to claim that he invented it at some earlier point. That trait was the basis of Burrell's impersonation of Jef.

Jef had a slight stammer, which Burrell nailed perfectly. Burrell began by folding his fingers together like Jef and then exclaiming in a soft, Jef-like voice, "Why, why, why, I invented the Macintosh!"

Then Burrell would shift to his radio announcer voice, playing the part of an imaginary interviewer. "No, I thought that Burrell invented the Macintosh", the interviewer would object.

He'd shift back to his Jef voice for the punch line.

"Why, why, why, I invented Burrell!"



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Texaco Towers

Author: Andy Hertzfeld

Date: January 1981

Characters: Steve Jobs, Dan Kottke, Brian Howard, Bud Tribble,
Jef Raskin, Burrell Smith, George Crow

Topics: Origins, Lisa, Buildings

Summary: The office where the Mac became real



The main Apple buildings on Bandley Drive in Cupertino had boring numerical appellations (Bandley 1, Bandley 3, etc.), but from the beginning the Lisa team gave the buildings they inhabited more interesting names. The original office for the Lisa team was adjacent to a Good Earth restaurant (in fact, it was Apple's original office in Cupertino), so it was called the "Good Earth" building. When the team grew larger and took over two nearby office suites, they



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



were designated "Scorched Earth" (because it housed the hardware engineers, who were all smokers) and "Salt of the Earth".

When the Lisa team became a separate division in 1980, they moved to a larger, two-story office building a block or two away from the main building on Bandle Drive. Everyone was so impressed at having two stories (all the other Apple buildings were single story) that the building was dubbed "Taco Towers", although I'm not sure where the "Taco" part came from.

In December 1980, the embryonic Macintosh team was residing in the Good Earth building, which was abandoned by the Lisa team for Taco Towers earlier in the year. When Steve Jobs took over the project, he moved it to a new building that was large enough to hold about fifteen or twenty people, a few blocks away from the main Apple campus at the southeast corner of Stevens Creek Boulevard and Saratoga-Sunnyvale Road.

There was a Texaco gas station at the corner, and a two-story, small, brown, wood paneled office building behind it, the kind that might house some accountants or insurance agents. Apple rented the top floor, which had four little suites split by a corridor, two on a side. Because of the proximity of the gas station and the perch on the second story, as well as the sonic overlap between "Taco" and "Texaco", the building quickly became known as "Texaco Towers".

Burrell Smith and Brian Howard took over the side of the building closest to the gas station and built a hardware lab, while Bud Tribble and Jef Raskin set up shop on the other side, installing desks with prototype Lisas to use for software development. Bud's office had four desks, but he was the only one occupying it at first. Steve didn't have an office there, but he usually came by to visit in the late afternoon.

In the corner of Bud's office, on one of the empty desks, was Burrell's 68000 based Macintosh prototype, wired-wrapped by Burrell himself, the only one currently in existence, although both Brian Howard and Dan Kottke had started wire-wrapping additional ones. Bud had written a boot ROM that filled the screen with the word "hello", rendered in a small bitmap that was thirty two pixels wide for easy drawing, which showed off the prototype's razor sharp video and distinctive black on white text.

When I started on the project in February 1981, I was given Jef's old desk in the office next to Bud's. Desk by desk, Texaco Towers began to fill up, as more team members were recruited, like Collette Askeland to lay out the PC boards, or Ed Riddle to work on the keyboard hardware. When George Crow started, there wasn't an office available for him, so he set up a table in the common foyer and began the analog board design there.

Burrell and I liked to have lunch at Cicero's Pizza, which was an old Cupertino restaurant that was just across the street. They had a Defender video game, which we'd play while waiting for our order. We'd also go to Cicero's around 4pm almost every day for another round of Defender playing; Burrell was getting so good he would play for the entire time on a single quarter.

In May of 1981, Steve complained that our offices didn't seem lively enough, and gave me permission to buy a portable stereo system for the office at Apple's



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



expense. Burrell and I ran out and bought a silver, cassette-based boom box right away, before he could change his mind. After that we usually played cassette tapes at night or on the weekends when there was nobody around that it would bother.

By early 1982, the Mac team was overflowing Texaco Towers and it was obvious that we'd have to move to larger quarters soon. Steve decided to move the team back to the main Apple campus, into Bandley 4, which had enough space for more than 50 people. The 68000 based Macintosh was born in the Good Earth building, but I still think of Texaco Towers as the place where it came of age, transitioning from a promising research project into a real, world-changing commercial product.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



A Message For Adam

Author: Andy Hertzfeld

Date: April 1981

Characters: Adam Osborne, Steve Jobs, Jef Raskin

Summary: We encounter Adam Osborne at the West Coast Computer Faire



The Osborne 1 portable computer

The Apple II was officially introduced at the First West Coast Computer Faire in April 1977, one of the very first trade shows dedicated to the newly emerging microcomputing industry. I loved the Computer Faires because they were attended by passionate hobbyists in the days before commercial forces completely dominated.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



In April 1981, a few members of the Mac team took off the afternoon and drove up to San Francisco to visit the seventh West Coast Computer Faire at Brooks Hall. The biggest splash at the show was the unveiling of the Osborne I, from a brand new company named Osborne Computer, which was touted as the world's first portable computer.

The Osborne I was the brain-child of Adam Osborne, who was a well known figure in the world of early microcomputers. Adam was a technical writer who founded a publishing company to publish crucial information about microprocessors and software that was sorely lacking at the time, which was eventually sold to McGraw-Hill. He became a controversial columnist, opining on the industry from his pulpit in InfoWorld and other publications. He had a populist vision of computing, touting a no-frills, low cost, high volume approach to the business.

In 1980, he decided to put his theories into practice, and founded the Osborne Computer Company to design, manufacture and market the Osborne 1, which was a low cost, one-piece, portable computer complete with a suite of bundled applications. He recruited Lee Felsenstein, already a microcomputing legend as the master of ceremonies of the Home Brew Computer Club, to design the hardware. Now, they were introducing the fruits of their labor at the West Coast Computer Faire, as Apple had done four years earlier.

The Osborne 1 was on display at their crowded booth near the center of Brooks Hall. It looked a lot like an oversized lunch box, with a keyboard on the back of the lid, crammed full with two floppy drives and a tiny, 5 inch monitor in its center. We were a little surprised, because it looked uncannily like some of Jef Raskin's early sketches for the Macintosh, which Steve had recently abandoned for a vertically oriented design. Portable was sort of a euphemism as the thing weighed around 25 pounds, but at least it fit under an airline seat, barely. As Macintosh elitists, we were suitably grossed out by the character-based CP/M applications, of course, which seemed especially clumsy on the tiny, scrolling screen.

We worked our way up to the front of the crowd to get a good look at the units that were on display. We started to ask one of the presenters a technical question, when we were suprised to see Adam Osborne himself standing a few feet from us, looking at our show badges, preempting the response.

"Oh, some Apple folks", he addressed us in a condescending tone, "What do you think? The Osborne 1 is going to outsell the Apple II by a factor of 10, don't you think so? What part of Apple do you work in?"

When we told him that we were on the Mac team, he started to chuckle. "The Macintosh, I heard about that. When are we going to get to see it? Well, go back and tell Steve Jobs that the Osborne 1 is going to outsell the Apple II and the Macintosh combined!"

So, after returning to Cupertino later that afternoon, we told Steve about our encounter with Adam Osborne. He smiled, with a sort of mock anger, and immediately grabbed the telephone on the spare desk in Bud's office, and called information for the number of the Osborne Computer Corporation. He dialed the number, but it was answered by a secretary.

"Hi, this is Steve Jobs. I'd like to speak with Adam Osborne."



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



The secretary informed Steve that Mr. Osborne was not available, and would not be back in the office until tomorrow morning. She asked Steve if he would like to leave a message.

"Yes", Steve replied. He paused for a second. "Here's my message. Tell Adam he's an asshole."

There was a long delay, as the secretary tried to figure out how to respond. Steve continued, "One more thing. I hear that Adam's curious about the Macintosh. Tell him that the Macintosh is so good that he's probably going to buy a few for his children even though it put his company out of business!".



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Pi neapple Pi zza

Author: Andy Hertzfeld

Date: May 1981

Characters: Burrell Smith, Brian Howard, Dan Kottke, Steve Jobs, Bud Tribble

Summary: We stay late to bring up the first printed circuit board



Our reward



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



When I began working on the Mac project in February 1981, there was still only one single 68000-based Macintosh prototype in existence, the initial digital board that was wire-wrapped by Burrell himself. It was now sitting in the corner of Bud Tribble's office, on one of the empty desks, attached to a small, seven-inch monitor. When powered up, the code in the boot ROM filled the screen with the word 'hello', in lower case letters and a tiny font, rendered crisply on the distinctive black-on-white display.

Dan Kottke and Brian Howard were already busy wire-wrapping more prototype boards, carefully following Burrell's drawings. In a week or so, I received the second prototype for my office, so I could work on the low level I/O routines, interfacing the disk and keyboard, while Bud worked on the mouse driver and porting Bill's graphics routines.

The next big step for the hardware was to lay out a printed circuit board. We recruited Collette Askeland, the best PC board layout technician in the company, from the Apple II group. Burrell spent a week or two working intensely with Collette, who used a specialized CAD machine located in Bandle 3 to input the topology and route the signals, eventually outputting a tape containing all the information needed to fabricate the boards.

Burrell and Brian Howard checked and rechecked the layout, which was tediously expressed as thousands of node connections, and after a day or two they decided they were ready to send it out for fabrication. We were hoping to get the first sample boards back before the weekend, but it looked like they weren't going to make it. Finally, around 4:30pm on a Friday afternoon, they arrived.

Burrell figured that it would take at least two or three hours to assemble a board, and then even longer to troubleshoot the inevitable mistakes, so it was too late to try to get one working that evening. Maybe they would come in on Saturday to get started, or maybe they'd wait until Monday morning. While they were discussing it, Steve Jobs strolled into the hardware lab, excited as usual.

"Hey, I heard that the PC boards finally arrived. Are they going to work? When will you have one working?"

Burrell explained that the boards had just arrived, and that it would take at least a couple of hours to assemble one, so they were thinking about whether to start tomorrow morning or wait until Monday.

"Monday? Are you kidding?", replied Steve, "It's your PC board, Burrell, don't you want to see if it works tonight? I'll tell you what, if you can get it to work this evening, I'll take you and anyone else who sticks around out for Pineapple Pizza."

Steve knew that Pineapple Pizzas had recently replaced Bulgarian Beef as Burrell's current food obsession (which, as a staunch vegetarian, he thought was a positive development) and that Burrell wanted a Pineapple Pizza pretty much every chance he could get. Burrell looked at Brian Howard and shrugged. "OK, we may as well give it a shot now. But I don't think we'll be able to get it working before the restaurants close."

So Burrell and Brian got busy, selecting a board and stuffing it with sockets, carefully soldering them in place, while five or six of the rest of us,



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



including Steve, sat around and kibitzed. Burrell seemed a little tense and impatient, since he didn't like the pressure of bringing up a board in front of so many spectators. Every five minutes or so, he referred to the awaiting Pineapple Pizza, speculating about how good it was going to taste.

Finally, around 8pm or so, the board was assembled enough to try to power it on for the very first time. The prototype was hooked up to an Apple II power supply and a small monitor, and fired up as we held our breath. The screen should have been filled with 'hellos', but instead all that was there was a checkerboard pattern.

We were all disappointed, except for Burrell. "That's not too bad", he commented, "It means the RAM and the video generation are more or less working. The processor isn't resetting, but it looks like we're pretty close." He turned to look directly at Steve. "But I'm too hungry to keep working - I think it's time for some Pineapple Pizza."

Steve smiled and agreed that it was good enough for the first night, and it was time to celebrate. The seven or eight of us who stayed late drove in three cars to Burrell's favorite Italian restaurant, Frankie, Johnny and Luigi's in Mountain View, ordering three large Pineapple Pizzas, which tasted great.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



PC Board Esthetics

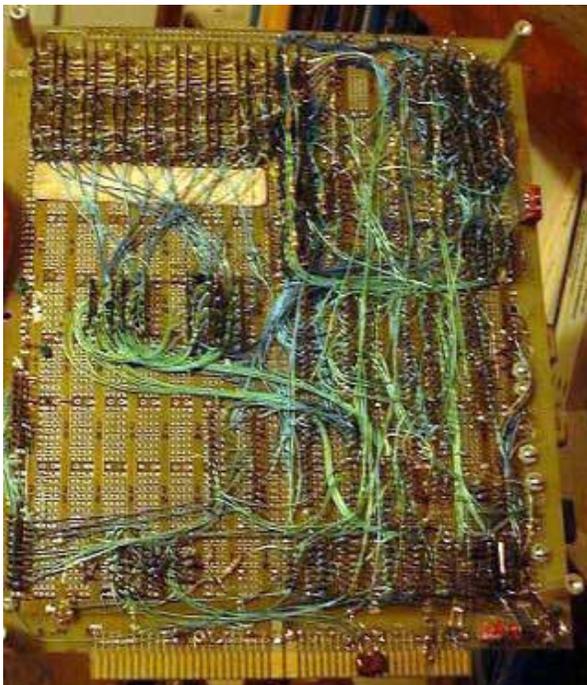
Author: Andy Hertzfeld

Date: July 1981

Characters: Steve Jobs, Burrell Smith, George Crow, Brian Howard, Dan Kottke

Topics: Hardware Design, Personality

Summary: Steve is concerned with the esthetics of the PC board



A wire-wrapped Macintosh prototype



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



The first Mac prototypes were hand-made using a technique called "wire-wrapping", where each individual signal is routed by wrapping an individual wire around two pins. Burrell wire-wrapped the first prototype himself, and then others were done by Brian Howard and Dan Kottke. But wire-wrapping is time consuming and error prone.

By the spring of 1981, the Mac's hardware design was stable enough for us to make a printed circuit board, which would allow us to make prototypes much more quickly. We recruited Collete Askeland from the Apple II group to lay out the board, and after working with Burrell and Brian for a couple of weeks, she taped out the design and sent it off for a limited production run of a few dozen boards.

We started having weekly management meetings in June 1981, which were attended by most of the team, where we discussed the issues of the week. At the second or third meeting, Burrell presented an intricate blueprint of the PC board layout, which had already been used to build a few working prototypes, blown up to four times the actual size.

Steve started critiquing the layout on a purely esthetic basis. "That part's really pretty", he proclaimed. "But look at the memory chips. That's ugly. The lines are too close together".

George Crow, our recently hired analog engineer, interrupted Steve. "Who cares what the PC board looks like? The only thing that's important is how well that it works. Nobody is going to see the PC board."

Steve responded strongly. "I'm gonna see it! I want it to be as beautiful as possible, even if it's inside the box. A great carpenter isn't going to use lousy wood for the back of a cabinet, even though nobody's going to see it."

George started to argue with Steve, since he wasn't on the team long enough to know that it was a losing battle. Fortunately, Burrell interrupted him.

"Well, that was a difficult part to layout because of the memory bus.", Burrell responded. "If we change it, it might not work as well electrically".

"OK, I'll tell you what," said Steve. "Let's do another layout to make the board prettier, but if it doesn't work as well, we'll change it back."

So we invested another \$5,000 or so to make a few boards with a new layout that routed the memory bus in a Steve-approved fashion. But sure enough, the new boards didn't work properly, as Burrell had predicted, so we switched back to the old design for the next run of prototypes.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Shut Up!

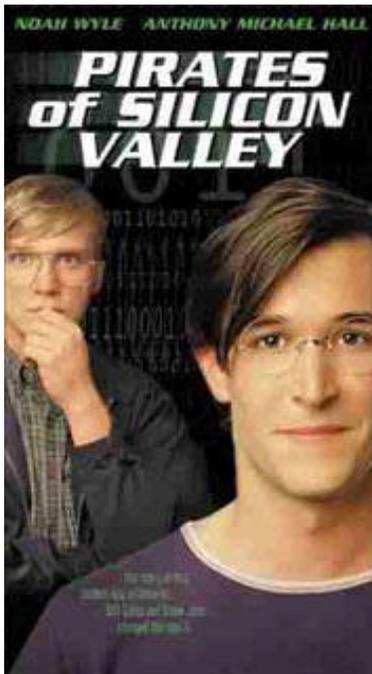
Author: Andy Hertzfeld

Date: July 1981

Characters: Andy Hertzfeld, Bud Tribble, Steve Jobs, Bill Gates, Charles Simonyi

Topics: 3rd party developers, Microsoft, Prototypes

Summary: The first time we demoed the Macintosh to Microsoft



This movie had a scene based on this story



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Apple had already learned the value of having a thriving third party software market with the Apple II, whose sales increased more than ten-fold when Visicalc, developed by a tiny company called Software Arts, caught on in the business market. The Macintosh intended to replicate the success of the Apple II as an industry standard platform, so it was very important to bring third party developers into the picture as soon as possible.

Microsoft was an obvious choice for one of the first companies for us to talk to. Both companies were started around the same time during the infancy of the personal computer industry, and they already had a business relationship, since Apple licensed Microsoft's Applesoft Basic for the Apple II. Bill Gates and Steve Jobs were born in the same year and shared a similar vision for the potential of personal computers. Each one thought he was smarter than the other one, but Steve generally treated Bill as someone who was slightly inferior, especially in matters of taste and style. Bill looked down on Steve because he couldn't actually program.

Steve had mentioned the Macintosh project to Bill at an industry conference in April, which led to a meeting in Seattle in June, where Steve spun an intriguing vision of pumping out Macintoshes by the millions in an automated factory. The plan was for Microsoft to develop a series of applications for the Macintosh, to be ready at launch. Things went well enough to schedule a meeting in Cupertino in July, where we promised a demo of the actual machine.

Unfortunately, there was one small snag. We were using the Lisa as the development machine for writing the software for Macintosh, and we hadn't yet reached the point where the Macintosh could run stand-alone. The Macintosh needed to be hooked up to a Lisa, in order to download software from it. But the Lisa group was writing all of its own applications for Lisa and didn't want Microsoft to see a Lisa. They made us promise that we wouldn't let Microsoft see the Lisa.

We finally came up with a solution where we'd use a twenty-five foot cable and keep the Lisa in a different room that the Microsoft guys weren't allowed to enter. I would start up the programs on the Lisa in the other room, and Bud Tribble would operate the Macintosh. I would usually run into the main room to see their reaction.

Bill Gates showed up in the early afternoon with three other colleagues: Charles Simonyi, who had recently joined Microsoft from Xerox PARC, Jeff Harbers, who would manage the Macintosh development team, and Mark Matthews, who was to be the technical lead on the project. They crowded around the prototype, and we started to run our various demos for them, with Steve doing most of the talking.

You could tell that Bill Gates was not a very good listener - he couldn't bear to have anyone explain how something worked to him - he had to leap ahead instead and guess about how he thought it would work.

We showed him how the Macintosh mouse cursor moved smoothly, in a flicker-free fashion.

"What kind of hardware do you use to draw the cursor?", he asked. Many current personal computers had special hardware to draw small bitmaps called "sprites", and he thought we might be doing something similar.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Of course, the Macintosh didn't use any special hardware at all. It did everything in software, which was more flexible anyway, during the vertical blanking interval to eliminate the possibility of flicker. In fact, Burrell and I had recently gotten a mouse to run smoothly on an Apple II, using a similar technique (see Apple II Mouse Card) .

"We don't have any special hardware for it!" I blurted out, probably with a proud sneer in my voice. "In fact..." I was about to mention that we got it running on an Apple II, which had one tenth the processing horsepower of a Macintosh, when Steve guessed what I was about to say.

"Shut up!", he yelled as loud as he could, looking directly at me. He yelled it again, possibly trying to drown me out in case I kept on going. But I understood what he wanted, changing what I was going to say, "In fact, doing it in software is better anyway".

The rest of the demo went pretty well, and both teams shared their excitement about how the Macintosh was going to take the industry to another level. We went out to dinner at a fancy restaurant in Los Gatos to celebrate working together, and we agreed to a framework of a deal where we would give them a stand-alone Mac prototype to develop with in the fall. We were very excited to be working with a third-party company that seemed to understand and appreciate what we were doing.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Donkey

Author: Andy Hertzfeld
Date: August 1981
Characters: Bill Gates, Neil Konzen, Steve Jobs, Charles Simonyi
Topics: Microsoft
Summary: We discover who wrote an awkward game for the PC



The original IBM PC

The first version of the IBM PC was introduced in August 1981. Apple responded by running an ad in the Wall Street Journal with the headline "Welcome, IBM. Seriously." Even though he was usually tight with money, Steve Jobs allowed the Mac team to buy an early unit to dissect and evaluate. The day it became available, we ran to the store and purchased one to take back to the lab.

ANECDOTES ABOUT THE DEVELOPMENT OF APPLE'S ORIGINAL MACINTOSH COMPUTER

<http://www.folklore.org/ProjectView.py?project=Macintosh&detail=medium>
28 January 2004 :: Page 170 of 254



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Needless to say, we were not very impressed with it. From the perspective of the Macintosh that we were already in the midst of bringing to life, it seemed like ancient history the day it came out. There was little, if any, Woz-like cleverness in the hardware design, using dozens of extraneous chips without having any cool features. The 8088 was a decent processor compared to the 6502, but it paled next to the 68000 we were using in the Mac.

But the most clunky part of the system was the software. MS-DOS seemed like a clone of an earlier system, CP/M, and even the demo programs lacked flair. It came with some games written in BASIC that were especially embarrassing.

The most embarrassing game was a lo-res graphics driving game called "Donkey". The player was supposed to be driving a car down a slowly scrolling, poorly rendered "road", and could hit the space bar to toggle the jerky motion. Every once in a while, a brown blob would fill the screen, which was supposed to be a donkey manifesting in the middle of the road. If you didn't hit the space bar in time, you would crash into the donkey and lose the game.

We thought the concept of the game was as bad as the crude graphics that it used. Since the game was written in BASIC, you could list it out and see how it was written. We were surprised to see that the comments at the top of the game proudly proclaimed the authors: Bill Gates and Neil Konzen. Neil was a bright teenage hacker who I knew from his work on the Apple II (who would later become Microsoft's technical lead on the Mac project) but we were amazed that such a thoroughly bad game could be co-authored by Microsoft's co-founder, and that he would actually want to take credit for it in the comments.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Gobble, Gobble, Gobble

Author: Andy Hertzfeld
Date: March 1982
Characters: Steve Jobs, Andy Hertzfeld, Burrell Smith, Angeline Lo
Topics: Management, Personality
Summary: We interview candidates for software manager

When Bud told us in early December of 1981 that he had to leave the Mac team to go back to Seattle to keep his place in medical school, both Burrell and I were pretty shaken. We were worried that we couldn't pull it off without Bud and that we'd get some authoritarian manager instead, who would wreck the unique spirit of our team. We expressed our concerns to Steve, who promised us that we'd have a big say in hiring the new manager, and that he would personally protect us if a situation like the one we were fearing ever arose.

In January, we began interviewing candidates for the software manager position. We had high standards and expectations and interviewed a number of outstanding people, like Ed Taft (who became one of the first employees at Adobe) and Tim Mott (who helped start Electronic Arts), who for one reason or another didn't take the job. But we also interviewed some more mundane candidates, which could sometimes get pretty wild if Steve decided he didn't respect someone.

For example, Burrell, Steve and I interviewed Angeline Lo's former manager, whom she highly recommended. As soon as the guy walked into the room, I knew it was going to be problematic, because he seemed extremely straight-laced and uptight, dressing more like an insurance salesman than a technologist. He also seemed very nervous as he fumbled at our first few questions.

I could tell that Steve was losing patience when he started to roll his eyes at the candidate's responses. Steve began to grill him with some unconventional questions.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



"How old were you when you lost your virginity?", Steve asked

The candidate wasn't sure if he heard correctly. "What did you say?"

Steve repeated the question, changing it slightly. "Are you a virgin?". Burrell and I started to laugh, as the candidate became more disconcerted. He didn't know how to respond.

Steve changed the subject. "How many times have you taken LSD?"

The poor guy was turning varying shades of red, so I tried to change the subject and asked a straight-forward technical question. But when he started to give a long-winded response, Steve got impatient again.

"Gooble, gobble, gobble, gobble", Steve started making turkey noises. This was too much for Burrell and myself, and we all started cracking up. "Gobble, gobble, gobble", Steve continued, laughing himself now.

At this point, the candidate stood up. "I guess I'm not the right guy for this job", he said.

"I guess you're not", Steve responded. "I think this interview is over."

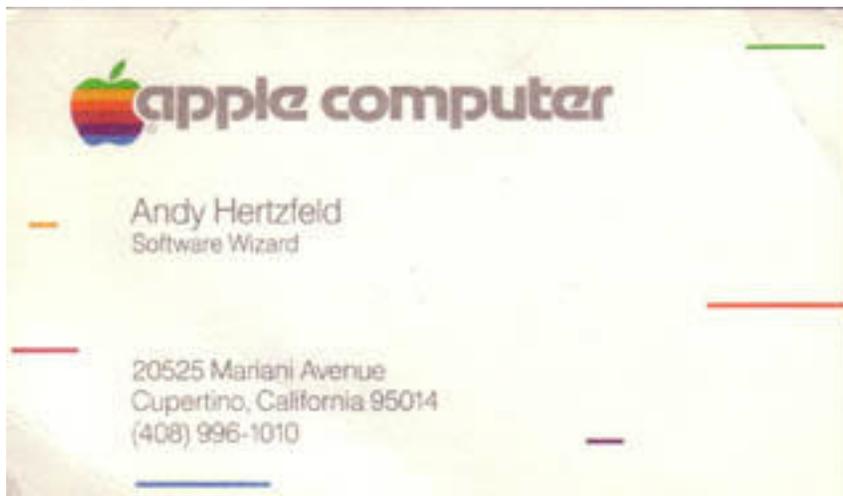


APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Software Wizard

Author: Andy Hertzfeld
Date: March 1982
Characters: Andy Hertzfeld, Burrell Smith, Steve Jobs
Topics: Personality, Apple Spirit
Summary: The Mac group gets business cards



By the spring of 1982, the Macintosh project was considered more legitimate within Apple. It was beginning to transition from a research effort into a mainstream project. We had to get more organized as the team grew.

Initially, we didn't have formal titles in the Mac group, but we needed to figure out what they were in order to get business cards made. My title with the Apple II group was "Senior Member of Technical Staff", which sounded dull to me.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



I told Peggy Alexio, Rod Holt's secretary, who was ordering the business cards, that I didn't want any, because I didn't like my title.

The next day Steve came by and told me that he heard that I didn't want business cards, but he wanted me to have them, and he didn't care what title I used; I could pick any title that I liked. After a little bit of thought, I decided on "Software Wizard", because you couldn't tell where that fit in the corporate hierarchy, and it seemed a suitable metaphor to reflect the practical magic of software innovation.

When I told Burrell about my new title, he immediately claimed "Hardware Wizard" for himself, even though I discouraged him, since it diminished the uniqueness of my title. And, as soon as word got around, lots of other folks on the Mac team started to change their titles to something more creative. The trend persisted at Apple for many years, and even spread to other companies, but as far as I know, that's how it got started.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Creative Think

Author: Andy Hertzfeld
Date: July 1982
Characters: Alan Kay, Bill Budge
Topics: Inspiration
Summary: A seminar with Alan Kay



Alan Kay, in the PARC days

In July of 1982, while I was in the midst of writing the Control Manager part of the Macintosh toolbox, my friend Bill Budge invited me to a computer industry seminar called "Creative Think", where interesting people gave purportedly inspiring talks. It was organized by Roger van Oech, a consultant who had written a book about creativity entitled "A Whack On The Side Of The Head". I usually avoided both creativity seminars and industry schmooze-fests, but my

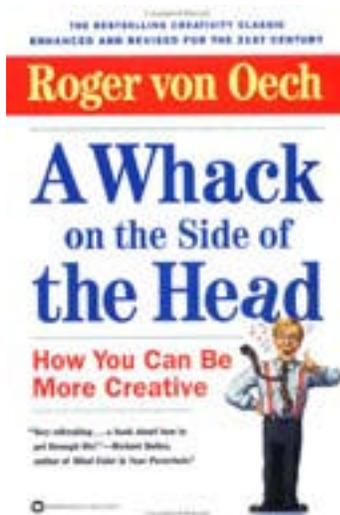


APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



friend Bill had somehow finagled free tickets and I thought it would be worth it just to see him.

Carver Mead gave an interesting talk on newly emerging VLSI technology, and some old chip industry veterans recounted amusing war stories, but the last talk of the day was the one that mattered to me. It was given by Alan Kay, the inventor of Smalltalk and the Alto, and the driving force behind Xerox PARC. I had read about Alan, and been inspired by his article in the September 1977 issue of Scientific American, but I had never seen him before in person or heard much about him.



Roger's Book

Alan's speech was revelatory and was perhaps the most inspiring talk that I ever attended. I grew increasingly excited as he made one brilliant, insightful remark after another, and took out my notebook to write as much of it down as I could. Alan was articulating the values behind the work that I was doing, even though he wasn't aware of it, in a way that really resonated with me. After I got back to my office in Cupertino, I transcribed it onto a single page, and copied it to give out to the rest of team.

I still have those notes, so I thought it would be interesting to reproduce them here, as an example of some of the thinking that inspired our efforts.

Alan Kay's talk at Creative Think seminar, July 20, 1982

Outline of talk: Metaphors, Magnetic Fields, Snobbery and Slogans

The best way to predict the future is to invent it.

Humans like fantasy and sharing:

Fantasy fulfills a need for a simpler, more controllable world.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Sharing is important - we're all communication junkies. We have an incredible bandwidth disparity (easy to take in, hard to give out); our devices have the reciprocal disparity (hard to take in, easy to give out)

Metaphors:

Computer as medium (like clay or paint)

Computer as vehicle

Computer as musical instrument

Magnetic Fields:

Find a central metaphor that's so good that everything aligns to it. Design meetings are no longer necessary, it designs itself. The metaphor should be crisp and fun.

Smalltalk is object-oriented, but it should have been message oriented.

Snobbery:

Turn up your nose at good ideas. You must work on great ideas, not good ones.

Appreciate mundanity: after all, a pencil is high technology

One goal: the computer disappears into the environment

The computer shouldn't act like it knows everything.

The whole notion of 'programming language' is wrong.

Slogans:

Better is the enemy of best

Relative judgements have no place in art

Systems programmers are high priests of a low cult

Point of view is worth 80 IQ points

Good ideas don't often scale

Remember, it's all software, it just depends on when you crystallize it.

People who are really serious about software should make their own hardware.

Final advice: content over form, go for fun.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



You Guys Are In Big Trouble

Author: Andy Hertzfeld

Date: August 1982

Characters: Steve Jobs, Bill Atkinson, Bud Tribble, Andy Hertzfeld, Bruce Horn

Topics: Apple Spirit, Buildings, Management

Summary: We receive Steve's permission to dismantle the onerous burglar alarm

By the spring of 1982, the Mac team was growing so rapidly that we had to move from Texaco Towers back to the main part of the Apple campus on Bandley Drive. We moved into Bandley 4, a medium sized building across the street from Apple's main office.

One of the things that I liked about Bandley 4 was that the software team was in the very back of the building, near the parking lot, so we could go out the back door to our cars, or to play basketball, without having to walk all the way around the building. This worked out well for a couple of months, but eventually the facilities group decided to put an alarm on the back door, so you couldn't go in and out freely. They didn't arm the alarm until 5:30pm, but that didn't help me, since I usually didn't go out to dinner until after 7pm, and then usually came back to work another few hours. The alarm became a major annoyance, since it made me walk significantly out of my way a few times every day.

Every once in a while someone would forget that the alarm was there, and walk out the door anyway. The alarm would produce a head-splitting wail, destroying any chance of concentration until a security guard arrived to disarm it, which sometimes could take more than ten minutes.

I complained about the alarm every way that I knew how, but to no avail. About one quarter of the software team worked late, and the alarm was unnecessary while we were there, so I begged the facilities supervisor not to turn it on



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



until after midnight, even offering to arm it ourselves when the last person left. But my pleas fell on deaf ears.

Every couple of months, Bud Tribble would come down from Seattle and visit with us. We'd show him the latest work we were doing so he could make his great suggestions. Late one afternoon, he showed up in the software area, and we all gathered around to demo to him, including Bill Atkinson and Steve Jobs.

Bill had done some neat hi-resolution scans with an improved dithering algorithm, and he wanted to show them to Bud. They were on his hard disk in the Lisa building, so he ran out the back door to get it. It was after 5:30pm, so that set off the alarm, and a horrible, loud, pulsating noise filled the room.

It went on for at least three minutes before Steve yelled out, "Can't someone figure out how to stop that thing?"

I saw a chance to vanquish my nemesis. "Are we allowed to damage it to get it to stop?", I asked him.

"Yes, do anything you want, I don't care if you break it," he replied, holding his hands over his ears. "Just get the damn thing to stop!"

Bruce Horn and I ran over to the nearby hardware lab and picked up every tool we could find. I got a hammer and screwdriver, and proceeded to pound the screwdriver into the center of the alarm, driving a stake through the demon's heart. The screwdriver went all the way through to the other side, but the alarm kept sounding.

Finally, Bruce took over and gave the screwdriver a mighty twist, and the whole thing flew apart into a half dozen pieces and fell to the ground. The horrible noise finally stopped.

At that very moment, a grizzled security guard entered the back door, just in time to see us cheering as the wrecked alarm finally gave up the ghost. He looked at us, our tools of destruction still in our guilty hands, and said, "You guys are in big trouble!!! Who is in charge here? You better show me your badge."

Steve stepped forward and handed the guard his badge. "I'll take responsibility for this", he told him.

The guard scrutinized Steve's badge. He looked at Steve, then back at the badge a few times. Finally, he shrugged his shoulders, picked up the pieces of the broken alarm, and walked away, without saying another word.

We were gleeful that the alarm was gone, but exactly one week later, a new one was put in its place, which stayed there until we moved to Bandy 3 a few months later. I'm not sure why, but not even Steve could get them to set it for a later time.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Too Big For My Britches

Author: Andy Hertzfeld
Date: February 1983
Characters: Bob Belleville, Steve Jobs, Jerome Coonen, Andy Hertzfeld, Steve Capps, Larry Kenyon, Pat Sharp, Rod Holt
Topics: Management, Personality Clashes
Summary: My belated performance review is delivered verbally

Apple's HR policy dictated that each employee was supposed to receive a performance review from their manager every six months, which helped to determine your salary increase or possibly an award of additional stock options. But as the end of 1982 approached, I hadn't received my review for more than eight months.

This wasn't very surprising, since Bob Belleville, who was my boss and our software manager, was not getting along very well with the software team. He thought that some of us were intrinsically unmanageable, and that we didn't sufficiently respect him. Bob had replaced Rod Holt as the overall engineering manager in August, responsible for both hardware and software, and had just hired a new software manager, Jerome Coonen, who was slated to begin in January, which would allow him to further distance himself from the software team. But he still had to deal with us directly one last time to write our reviews for 1982.

By the end of January, everyone on the team had received their review except for me. Others mentioned that Bob had acted somewhat strangely during their reviews, making cryptic remarks that they didn't understand, so I wasn't particularly looking forward to mine. I occasionally had to interact with Bob, but he was reticent around me, not saying much, seemingly hiding behind his enigmatic, tight-lipped smile. Finally, after another couple of weeks, Bob's secretary called me to arrange an appointment, presumably for my belated review.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



The meeting was scheduled for 5pm, toward the end of the day on a Thursday afternoon in mid-February. Bob was waiting for me when I entered his corner cubicle. I asked him what was up. He said that he didn't want to get into it in the office, and suggested that we discuss it on a walk around the block. That was fine with me, but now I was even more apprehensive - generally, walks around the block were reserved for firing or demoting someone, or to talk someone into staying after they had quit.

Bob waited until we were a full block away from Bandley 4 before starting to speak.

"Well, Andy, you're not going to like hearing this, but you're a big problem on the software team and I'm giving you a negative review for the last six months of 1982."

I knew that Bob disliked me, but I was nevertheless shocked. I was working my heart out for the last two years, devoting my life to the Macintosh, seven days per week, holding the project together after Bud returned to medical school. I was really doing two full-time jobs, writing the Mac Toolbox in assembly language by night and helping everybody else by doing whatever was necessary each day.

"How can you say that?", I responded, horrified. "I accomplished everything that I was supposed to, and a lot more besides". All my previous reviews from Apple were extremely positive, including the last one from Bob, so this was new to me.

Bob unfurled his mirthless grin. "Oh, don't get me wrong, I think your technical work was perfectly adequate during the review period, and I don't have a single criticism of it. That's not your problem area. I don't have a single complaint with your technical work." He paused for a moment, to take a deep breath, and then continued.

"The problem is with your attitude, and your relationship with management. You are consistently insubordinate, and you don't have any respect for lines of authority. I think you are undermining everybody else on the software team. You are too big for your britches."

At this point, as he probably expected, I broke down into tears. The Macintosh was at the center of my life, and it was suddenly clear that I was going to have to quit. I couldn't work for somebody who was saying this, no matter how much the project mattered to me.

Perhaps Bob was a little taken aback by my tears, so he tried to soften things. "Listen, this could be a very expensive conversation. It could turn out to be either very good or very bad for both of us. I'm trying to get you to see how if you listen to me, things could turn out very good for both of us."

I had no idea what he was talking about, or how a bad review could possibly be good for me. "What do you mean, undermining the team?", I managed to choke out, "I'm always trying to help everybody else on the team. Give me one example of someone that I've undermined."

"Larry Kenyon", Bob replied. "You're stifling Larry Kenyon. Now he is someone with a good attitude, and you're keeping him from realizing his potential."



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



I always thought that I had gotten along great with Larry. I recruited him to the Mac team, after working with him on Apple II peripheral cards in 1980, and then handed off the low-level OS stuff to him while I worked on the Toolbox. I thought Larry was a terrific programmer and a great all around person, and treated him with the highest respect, and always enjoyed it when we worked together. I think I knew what Bob was getting at, though, as I had reacted poorly a few months ago when Bob appointed Larry as temporary manager when he had to be gone on a short trip, probably just to irk me.

By this point, I was crying harder, and Bob looked like he might start crying at any moment now, too. We were also pretty far from Bandley 4 by now, and it was starting to get dark. The tone of the conversation seemed to shift as we both realized that we should start heading back.

"This doesn't have to be that bad", Bob said as we turned around. "All you have to do is listen to me and things will work out fine."

"What do you mean?", I asked him.

"You need to show more respect to authority. It's not just me. Jerome is still new, and I'm afraid that you won't let him do his job. He's your boss now, and you need to show him respect, and let him do his job. But that's not the main problem. What you really have to do is stop talking to Steve Jobs." Bob paused and flinched slightly, as if just mentioning Steve was difficult for him.

"Whenever there's something that you don't like, even little things, you go running straight to Steve, and he interferes. I don't have any authority with the software team, because they always hear everything from Steve before I do, and he always hears everything about the software straight from you. It's making it so I can't do my job. You should communicate through the proper channels. I can't tell Steve what to do, but you work for me, so I can tell you."

I did respect Jerome, and I was trying to make an extra effort to support him as our manager, because I knew that we really needed him. Jerome was a very smart guy, and a passionate genius when it came to numerical software - I loved to hear him elucidate the intricacies of his beloved floating point routines. But I did consider him to be more of a partner than a boss, just like I did everyone else on the team, but I didn't think that he had a problem with that. But apparently Bob did.

But the Steve issue was different. From the earliest days of the project, Steve would usually show up at the Mac building in the late afternoon, or sometimes after dinner, and ask us about the happenings of the day. We would demo the latest stuff to him, or he'd complain about something, or sometimes we'd just exchange the latest gossip. After Bud went back to medical school, Burrell and I were the only ones who would regularly stay late, but after a while, more of the team began to hang out with us. It wasn't unusual for six or eight of us to go out for a late dinner, and then come back and keep working. By early 1983, most of the software team was staying late, and even some marketing and finance people would join us, but Bob Belleville never did, since he had to get home to his wife and two young daughters.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



"I can't stop Steve from coming around", I told Bob. "If you don't want me to talk with Steve, you're going to have to tell him about it. I like Jerome, and I have no problem working with him, but now it looks like I have a problem working with you. If you think that I'm undermining the team, I'm out of here tomorrow."

Bob looked at me intently. "I don't have the power to fire you", he said. "You're going to give me power that I don't have if you quit. Do you really want to do that?"

By now, it was completely dark as we were approaching the Apple parking lot. We stopped in front of Bob's car.

"This could be a really expensive conversation for both of us", Bob muttered cryptically. "It's entirely up to you." With that, he got into his car and drove off, and I wandered back into Bandley 4, feeling stunned and drained. I got back to my cubicle, put my head down on the desk, and started crying again.

It was around 6:30pm now, and most of the software team was still around. Capps saw that I was upset, and asked me what was wrong. He began to get angry when I told him and a few others what happened, and he made me promise not to overreact until he had a chance to find out what was going on.

Larry Kenyon was still in his cubicle, so I went over and told him what Bob had said. I asked him to be honest, and to tell me if he thought I was stifling him in any way.

"You've got to be kidding!", Larry exclaimed. "I think it's really great working with you, that's the reason I'm on this team. I think it's an honor to work with you." With that, I burst into tears again, touched by Larry's support.

I was exhausted and confused, so I went home to get some sleep and to think about what I should do next. When I came in earlier than usual the next morning, there was a message on my desk to call Pat Sharp, Steve's secretary. She told me to come by his office right away because he wanted to talk with me as soon as possible.

"I can't believe Bob gave you that review", Steve started talking even before I stepped into his office. "He showed it to me a week ago, but I refused to approve it, and I told him to write something more positive. Do you have a copy of it?"

I told Steve that he didn't give me a copy, he just delivered it verbally. I told him that I couldn't work for someone who feels that way about me, and that I had no alternative but to quit.

"It's good that you don't have a copy, because that review is rescinded, it doesn't officially exist. I just got done talking with Bob, and after I chewed him out, he also quit, because he said that he can't manage the software team. And Capps came in here and told me that the rest of the software team is so upset that they're thinking about quitting, too. What a mess! You and Bob don't have to love each other to work together. We're going to sit down this afternoon and talk this thing out until it's resolved."



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



So, at 4pm on Friday afternoon, as soon as Steve was available, the entire software team, plus Burrell, filed into one of the conference rooms. We all sat in a semi-circle of chairs on the right side of the room, waiting apprehensively. Finally, Steve strode in, with his characteristic bouncy stride, trailed by a despondent looking Bob Belleville, who took a seat on the left side of the room, facing the software team.

Steve started talking first, saying that tensions had been building up for a while, and it was time to clear the air, so we could all pull together down the home stretch. All the while, Bob was staring at the floor, unwilling to make eye contact with anyone else, controlling his emotions behind a tight-lipped expression, halfway between grin and grimace.

"OK, who's going to go first? What's the problem, and how do we fix it?", Steve asked.

Capps spoke up, explaining how painful and unmotivating it was to see me broken up about an obviously unjustified review. He wanted to know how things could have gotten so screwed up.

Steve nodded to Bob, encouraging him to speak up. Bob spoke in a monotone. "I didn't give Andy a bad review. I told him that his work was fine."

I was flabbergasted. "You said I was undermining the team, and stifling Larry," I blurted out. "I can't work for you if you think I'm undermining the team."

Bob looked up, looking me in the eye for the first time. He spoke in a mild, low emotionless monotone. "I didn't say any of those things. Why are you claiming that I said that?"

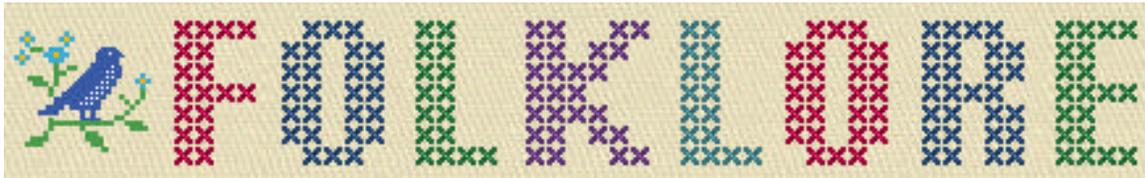
I was shocked. Bob was denying what he told me the day before, and I know that I didn't imagine it. Furthermore, he really seemed to genuinely believe what he was saying, and he looked to be in a kind of trance, both depressed and confused. If he didn't acknowledge what he said to me, there was no way to resolve it. I didn't know how to proceed, so I backed off my accusations.

A few more people spoke up, addressing other grievances, but Bob's trance-like manner persisted and eventually the meeting broke up, without anyone being satisfied. Steve tried to declare victory at the end, but no one was buying it.

I thought about things over the weekend, and realized that I cared too much about the Macintosh to quit before it was finished, managerial adversity notwithstanding. The situation that I feared when Bud left had actually occurred, in spades, and I wasn't confident that Steve would live up to his promise to protect me. I wasn't sure what was going to happen, but I knew that my blissful days at Apple were over, and that things were going to be different from now on.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Quick, Hide In This Closet!

Author: Andy Hertzfeld

Date: August 1983

Characters: George Crow, Bob Belleville, Larry Kenyon, Steve Jobs, Rod Holt, Rich Williams, Hide Kamoto

Topics: Hardware Design, Management, Lisa

Summary: Steve forbids us to work with Sony



A Twigg diskette



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



In 1980, Apple reorganized again, splitting off a new "Disk Division" headed by John Vennard, responsible for developing a hard disk code-named "Pippin" and a next generation floppy disk code-named "Twiggy". Both were intended to be used first by the Lisa project, and eventually across Apple's entire product line. At Rod Holt's request, I had written some early diagnostics for Twiggy using an Apple II, but I felt lucky that they asked Rich Williams instead of me to transfer to the disk division as their software guy, since focusing exclusively on disks seemed pretty limiting.

Woz's Apple II floppy disk design was way ahead of the rest of the industry, so Apple felt confident that it could continue to innovate to extend its lead. Twiggy was a fairly ambitious project, more than quadrupling the capacity of standard floppy disks by doubling the data rate (which required higher density media) and employing other innovative tricks like motor speed control, which slowed down the disk rotation speed on the outer tracks to cram more data on them.

The Lisa was designed to include two built-in Twiggy drives, so it made sense for the Macintosh to use Twiggy as well. Twiggy used a Woz-style disk controller, which created a problem for the Lisa designers, since that required exact timing from the microprocessor and therefore couldn't tolerate interrupts, which was perhaps OK for a simple system like the Apple II but was unacceptable for a more sophisticated system like Lisa. Instead, the Lisa hardware designers (Paul Baker, Bob Paratore and others) solved the problem by including a little Apple II, with its own memory and microprocessor (but clocked twice as fast), inside the Lisa to control the Twiggy drives.

The Lisa also supported an optional, external hard drive through a built-in parallel port. As the Twiggy designers encountered unexpected difficulties in achieving an acceptable error rate, Lisa came to rely on the hard drive instead. The Twiggy drive was also slower than expected, because of the high error rate as well as the way the variable motor speed trick increased seek times, since you had to wait for the speed change to stabilize. Besides, the Lisa operating system designers were used to working on systems that swapped memory from disk, which wasn't really feasible to pull off at floppy disk speeds. Soon, the hard disk became mandatory, upping the minimum price of a Lisa by more than a thousand dollars.

Lisa was announced to great fanfare in January 1983, but it still wasn't ready to ship. There were problems in a number of areas, but the biggest one was the low yield of the Twiggy drives, whose high error rate greatly limited production. Finally, Lisas were shipped to customers in June 1983, even though there continued to be production and reliability problems with the disk drives.

Meanwhile, the Mac team was beginning to panic. We were using a single Twiggy drive as our floppy disk, and we didn't have a hard disk to fall back on. It looked like the Twiggy drive was never going to be reliable or cost effective enough for the Macintosh, but we were stuck without an alternative. If we couldn't find a suitable replacement quickly enough, we'd have to slip the entire project indefinitely.

Fortunately, we were aware of Sony's new 3.5 inch drive that they started to ship in the spring of 1983 through Hewlett-Packard, their development partner.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



George Crow, the analog engineer who designed the Mac's analog board, had come from HP prior to working at Apple and was sold on the superiority of the Sony drives. He procured a drive from his friends at HP and proposed to Bob Belleville that we figure out how to interface it to the Mac as soon as possible, while we negotiate a deal with Sony.

The Sony drive looked really sweet, especially when compared to the Twiggy. It used the same data rate as Twiggy, but on smaller disks that could fit in a shirt pocket. Best of all, the media was encased in a hard plastic shell, making it much less fragile and more convenient to handle.

Steve Jobs was finally ready to acknowledge reality and give up on the Twiggy drive. When he saw the Sony drive he loved it, and immediately wanted to adapt it for the Mac. But instead of doing the obvious thing and striking a deal with Sony, Steve decided that Apple should take what we learned from Twiggy and engineer our own version of a 3.5" drive, working with our Japanese manufacturing partner Alps Electronics, who manufactured the Apple II floppy drives at a very low cost.

This seemed like suicide to George Crow and Bob Belleville. The Mac was supposed to ship in less than seven months, and it was preposterous to think that we could get a 3.5" drive into production by then, if we could do it at all, given the disk division's dismal track record. But Steve was convinced that we should do our own drive, and told Bob to cease all work on the Sony drive. He instructed Rod Holt, Bob and George to fly to Japan to meet with Alps to initiate a crash project to develop a workable 3.5 inch drive.

Bob and George grudgingly went along with the Alps program, but they were certain that the team would discover that we couldn't pull it off in the allotted time frame. They hatched an alternative plan to continue to work with Sony surreptitiously, against Steve's wishes. Larry Kenyon was given a Sony drive to interface to the Mac, but he was told to keep it hidden, especially from Steve. Bob and George also arranged meetings with Sony, to discuss the customizations that Apple desired and to hammer out the beginnings of a business deal.

This dual strategy entailed frequent meetings with both Alps and Sony, with the added burden of keeping the Sony meetings secret from Steve. It wasn't that hard to do in Japan, since Steve didn't come along, but it got a little awkward when Sony employees had to visit Cupertino. Sony sent a young engineer named Hide Kamoto to work with Larry Kenyon to spec out the modifications that we required. He was sitting in Larry's cubicle with George Crow when we suddenly heard Steve Jobs's voice as he unexpectedly strode into the software area.

George knew that Steve would wonder who Kamoto-san was if he saw him. Thinking quickly, he immediately tapped Kamoto-san on his shoulder, and spoke hurriedly, pointing at the nearby janitorial closet. "Dozo, quick, hide in this closet. Please! Now!"

Kamoto-san looked confused but he got up from his seat and hurried into the dark janitorial closet. He had to stay there for five minutes or so until Steve departed and the coast was clear.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



George and Larry apologized to Kamoto-san for their unusual request. "No problem.", he replied, "But American business practices, they are very strange. Very strange."

As predicted, a few weeks later the Alps team came back with an eighteen month estimate for getting their drive into production, and we had to abandon the project. When Bob Belleville revealed that he and George had kept the Sony alternative alive, Steve swallowed his pride and thanked them for disobeying him and doing the right thing. The Sony drives eventually worked out great, and it's hard to imagine what the Mac would have been like without them today.

[<http://www.macgeek.org/museum/sony400kdri ve/>]

SONY Micro Floppydisk Drive - Model OA-D30V

By Bryan G. Villados
October 27, 2002

Copyright 2002 Pearl City Networks. All Rights Reserved. This article is not to be re-published without permission of the author.

The SONY Micro Floppydisk Drive is very important to the history of Apple Computer. SONY Corporation invented the "micro" floppy disk drive, releasing its first commercially available model called the OA-D30V in 1981. The model OA-D30V you see in these photos is therefore considered the world's first flexible 400KB micro floppy disk drive to hit the market. In 1984, Apple took a more modernized model, the model OA-D34V (and later the OA-D34V-02) of the SONY mechanism and implemented it into the Apple Macintosh 128K. And, around the same time, Apple included the same part into the Apple Lisa 2 and the Apple Lisa 1 upgrade kit.

A number of things changed within the three years between the release of the OA-D30V and the OA-D34V. The OA-D30V is compatible with existing floppy disk controllers of its time (PCs used a 360KB 5.25-inch disk controller). Therefore, the disk drive required a power source, which at the time was not provided by the ribbon cable. It is the same power connector you see in today's PC 3.5-inch disk drives. The model OA-D34V Apple used later on provided power through the ribbon cable.

If you're a Macintosh fan, you should be familiar with the SONY auto-inject (MP-F51W and MP-F75W) models and manual-inject (MPF52A and MF355F) disk drives. Both models allow the diskette to be ejected automatically. The OA-D30V did not allow the diskette to eject automatically. Instead, you had to press the eject button in order to take the diskette out. Therefore, the OA-D30V was both a "manual-eject" and "manual-inject" disk drive.

The Micro Floppydisk, model OM-D30V, was also invented by SONY Corporation. Though it's basically a low-density, single-sided diskette, the disk guard (other people call it a "shutter") was somewhat of a different design. The disk guard had more area for movement than today's disk guards, which is able to move back-and-forth within a 33mm area (today's disk guards move at approx. 23mm).

Also, today's 3.5-inch disk drives automatically move the disk guard, exposing the magnetic surface. It's able to do this by placing an arm into a notch, then moving the arm, opening the disk so it can be read or written. The OM-D30V

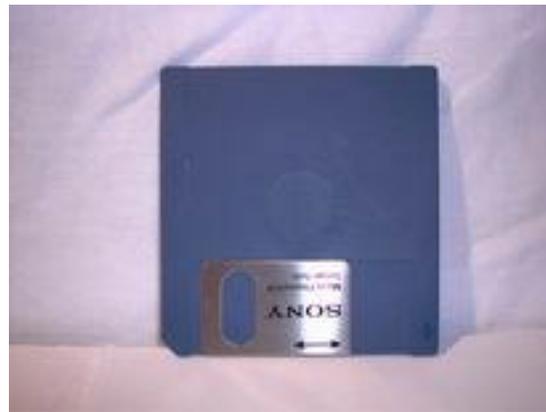


APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



required that you manually slide the disk guard to the open position, then insert into the disk drive.

And, the disk write-protect is quite primitive on the OM-D30V. The diskette was shipped with the disk in a writable state. To write-protect the disk, you would have to break the break off the disk protect tab. That means, if you were to write on this diskette again, you had to find a way to re-plug the hole. Today's 3.5-inch diskettes has a disk protect tab that can click in two positions: write and write-protect. And, the write protect hole only showed on the bottom and not the top of the OM-D30V, unlike today's diskettes.



Errata

10/29/2002 - First off, the OM-D30V diskette's disk protect tab, when removed, could actually be re-attached. Also, even though the diskette is a 400KB diskette, the format capacity is 360KB. When the drive was first released in 1981, it was reportedly used in the Hewlett-Packard 150. Later models of the 150 had double-sided disk drives. Thanks to Michael J. Mahon, Dosius, Greg Buchner for these corrected pieces of information.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Steve Wozniak University

Author: Andy Hertzfeld
Date: September 1983
Characters: Burrell Smith, Steve Wozniak
Topics: The Launch, Personality
Summary: Burrell's educational credentials

The launch of the Macintosh was finally scheduled for January 24, 1984, so we had to start preparing for it in the fall of 1983. Steve decided to anoint some of the engineers plus a few others as the official "Macintosh Design Team", and we had to be prepared to be interviewed by the press. Some magazines had lead times of more than three months, so interviews were scheduled as early as October.

As part of the preparations, the Regis McKenna press liaisons (known as the Rejettes) asked each of us to fill out a questionnaire that included the standard questions like date and place of birth, degrees, etc. They were going to be used as the basis for a fact sheet that would be handed out to the press.

Burrell, who didn't have a college degree, didn't know how to answer the "College Attended" question. He thought about it for a moment and then came up with the perfect answer: "Steve Wozniak University". In some fashion, we were all graduates of Steve Wozniak University.





APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Make a Mess, Clean it Up!

Author: Donn Denman

Date: September 1983

Characters: Burrell Smith, Donn Denman, Andy Hertzfeld

Topics: Personality, Recreation

Summary: Burrell had a unique approach to playing Defender



Defender was Burrell's favorite video game



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Working 90 hours a work week requires frequent, and highly effective, work breaks. In the center of Macintosh work area in Bandle 3 we had a ping pong table, a nice stereo system, and a Defender video game machine. We found that competitive play gave us a jolt of adrenaline, and a refreshed mind-set when we resumed work. We also learned a lot about our coworkers and how they excel during competition. While playing Defender one day I got some great insight into how Burrell accelerates his own learning process.

Andy, Burrell and I had a standing competition playing on the Defender machine. We'd challenge each other, in two or three player competitions, taking turns at the video game machine, and compare techniques and high scores. We were roughly equal in skill level, so as we'd take turns at the controls we could watch how the other player was doing, and have a gauge for who was ahead. This gave us opportunities to refine our skills, learn the other guy's technique, and show off our prowess.

The goal of Defender is to defend your humans from abduction by aliens. The evil green aliens drop down from the top of the screen and randomly pick up your humans, and try to bring them back up to the top of the screen. You control a ship and have to shoot the aliens, either before they grab a human, or during their rise up to the top of the screen. If an alien makes it to the top with a human, they consume him and become a vicious mutant, which attacks very aggressively. You start the game with ten humans, and if the last one dies, all the aliens become mutants, and they swarm in on your ship from all sides.

After a while, surviving the first few game levels was pretty easy, unless you had been up all night programming or something. The Defender machine was probably a pretty good objective measure of current mental capacity. "Gee, I can't even get through level 2! I guess it's time to get some sleep." Better to put in a bad performance on the defender game than mess up the current programming task, or start down the wrong path on some hardware design.

One day Burrell started doing something radical. Andy came by my cube and said "You've got to come see what Burrell's doing with Defender." "How can you innovate with a video game?" I wondered. I'd seen Burrell and Andy innovate on all kinds of things, but I couldn't imagine how he could somehow step outside the box of a video game - the machine controlled the flow and dictated the goals. How could you gain some control in that environment?

We started up a new competition, and when Burrell's turn came up, he did something that stunned me. He immediately shot all his humans! This was completely against the goal of the game! He didn't even go after the aliens, and when he shot the last human, they all turned to mutants and attacked him from all sides. He glanced in my direction with a grin on his face and said "Make a mess, clean it up!" and proceeded to dodge the swarm of angry mutants noisily chasing after him. "Burrell's not going to win this competition" I said to myself. "He's not going to last long with a screen full of mutants!"

Often a single mutant is enough to kill you. They move quicker, and with a different pace and pattern than the other aliens, so the normal evasive techniques don't work very well. Mutants move so quickly over small distances that they seem to just jump on top of you. Your ship is faster over the longer term, so you have to outrun them, establishing a gap, and only then do you have enough room to safely turn and fire at them.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



When Burrell's next turn came up I was surprised by how long his ship survived. He'd already developed a technique for dealing with a whole mass of mutants. He would circle around them again and again, and that would gather them into a densely clumped swarm. Then, while circling, he'd fire a burst pattern across the whole swarm, not needing to aim at individuals. He was doing really well, cutting through the swarm like the Grim Reaper's scythe. Burrell was no longer attacking individual mutants, instead he was treating the whole swarm as one big target.

Burrell may have lost that game and the next few, but it wasn't too long before he was really mastering the machine. Instead of avoiding the tough situations, he'd immediately create them, and immediately start learning how to handle the worst situation imaginable. Pretty soon he would routinely handle anything the machine could throw at him.

I was beginning to see how Burrell could be so successful with everything he does. Like many high achievers, Burrell likes challenges so much that he actually seeks them out and consciously creates them. In the long run, this approach makes sense. He seems to aggressively set up challenging situations throughout his life. Then, when life throws him a curve ball, he'll swing hard, and knock it out of the park.

Why intentionally "make a mess?" So you can get really good at "cleaning up!"



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



90 Hours A Week And Loving It

Author: Andy Hertzfeld
Date: October 1983
Characters: Burrell Smith, Debi Coleman, Steve Jobs
Topics: T-Shirts, Personality
Status: Draft
Summary: Burrell modifies his sweatshirt



Burrell wearing the sweatshirt



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Most of the Macintosh software team members were between twenty and thirty years old, and with few family obligations to distract us, we were used to working long hours. We were passionate about the project and willing to more or less subordinate the rest of our lives to it, at least for a while. As pressure mounted to finish the software in time to meet our January 1984 deadline, we began to work longer and longer hours. By the fall of 1983, it wasn't unusual to find most of the software team in their cubicles on any given evening, week day or not, still tapping away at their keyboards at 11pm or even later.

The rest of the Macintosh team, which had now swelled to almost a hundred people, nearing the limit that Steve Jobs swore we would never exceed, tended to work more traditional hours, but as our deadline loomed, many of them began to stay late as well to help us test the software during evening testing marathons. Food was brought in as a majority of the team stayed late to help put the software through its paces, competing to see who could find the most bugs, of which there were still plenty, even as the weeks wore on.

Debi Coleman's finance team decided to commemorate the effort that the entire team was putting forth in the traditional Silicon Valley manner: they made a T-Shirt. Actually, to make it a little more special, they chose a high quality, gray hooded sweatshirt. Steve Jobs had recently bragged to the press that the Macintosh team was working "90 hours a week". They decided that the tag line for the sweatshirt should be "90 Hours A Week And Loving It", in honor of Steve's exaggerated assertion.

The sweatshirt featured the name Macintosh in red letters, with the "90 Hours" tag line emblazoned in black across the front. They were produced in time for the next testing marathon, as a reward for participating. The software team wasn't all that pleased, since we felt that we really were working that hard, but most of the other sweatshirt recipients weren't even coming close. But it was a pretty nice sweatshirt, so lots of the engineers started wearing them frequently, including Burrell Smith.

When Burrell finally quit Apple in February 1985, he continued to wear the sweatshirt almost every day, but, as soon as he returned home following his resignation, he took some masking tape and made a big 'X' across the leading '9' character, virtually obliterating it from view. He proudly displayed the updated motto, reflecting exactly how he felt. It now read "0 Hours A Week And Loving It".



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



A Rich Neighbor Named Xerox

Author: Andy Hertzfeld

Date: November 1983

Characters: Steve Jobs, Neil Konzen, Mike Boich, Bill Gates

Topics: 3rd party developers, Microsoft, Personality

Summary: Steve confronts Bill Gates about copying the Mac



The Xerox Star was an influential predecessor

ANECDOTES ABOUT THE DEVELOPMENT OF APPLE'S ORIGINAL MACINTOSH COMPUTER

<http://www.folklore.org/ProjectViev.py?project=Macintosh&detail=medium>



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



When Steve Jobs recruited Microsoft to be the first third party applications software developer for the Macintosh, he was already concerned that they might try to copy our ideas into a PC-based user interface. As a condition of getting an early start at Macintosh development, Steve made Microsoft promise not to ship any software that used a mouse until at least one year after the first shipment of the Macintosh.

Microsoft's main systems programmer assigned to the Mac project was Neil Konzen, a brilliant young Apple II hacker who grew up in their backyard in the suburbs of Seattle. Neil started working at Microsoft while he was still a high school student, and single-handedly implemented the system software for their hit Z80 card that allowed the Apple II to run CP/M software.

Neil loved Apple, so it was natural for Microsoft to assign him to their new, top-secret Macintosh project. He was responsible for integrating Microsoft's byte-code based interpreted environment (which actually was a copy of a system used at Xerox that favored memory efficiency over execution speed, which was appropriate for the Mac's limited memory) with the rapidly evolving Macintosh OS, so he quickly became Microsoft's expert in the technical details of the Mac system.

By the middle of 1983, Microsoft was far enough along to show us working prototypes of their spreadsheet and word processor programs, Multiplan and Chart. I would usually talk with Neil on the phone a couple of times a week. He would sometimes request a feature that I would implement for him, or perhaps complain about the way something was done. But most of the time I would answer his various questions about the intricacies of the still evolving API.

I gradually began to notice that Neil would often ask questions about implementation details that he didn't really need to know about. In particular, he was really curious about how regions were represented and implemented, and would often detail his theories about them to me, hoping for confirmation.

Aside from intellectual curiosity, there was no reason to care about the system internals unless you were trying to implement your own version of it. I told Steve that I suspected that Microsoft was going to clone the Mac, but he wasn't that worried because he didn't think they were capable of doing a decent implementation, even with the Mac as an example.

In November 1983, we heard that Microsoft made a surprising announcement at Comdex, the industry's premier trade show, held twice a year in Las Vegas. Microsoft announced a new, mouse-based system graphical user interface environment called Windows, competing directly with an earlier environment announced by Personal Software called "Vision". They also announced a mouse-based option for Microsoft Word. When Steve Jobs found out about Windows, he went ballistic.

"Get Gates down here immediately", he fumed to Mike Boich, Mac's original evangelist who was in charge of our relationships with third party developers. "He needs to explain this, and it better be good. I want him in this room by tomorrow afternoon, or else!"

And, to my surprise, I was invited to a meeting in that conference room the next afternoon, where Bill Gates had somehow manifested, alone, surrounded by ten



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Apple employees. I think Steve wanted me there because I had evidence of Neil asking about the internals, but that never came up, so I was just a fascinated observer as Steve started yelling at Bill, asking him why he violated their agreement.

"You're ripping us off!", Steve shouted, raising his voice even higher. "I trusted you, and now you're stealing from us!"

But Bill Gates just stood there coolly, looking Steve directly in the eye, before starting to speak in his squeaky voice.

"Well, Steve, I think there's more than one way of looking at it. I think it's more like we both had this rich neighbor named Xerox and I broke into his house to steal the TV set and found out that you had already stolen it."

Unfortunately, it turned out that while the agreement that Microsoft signed in 1981 stipulated that they not ship mouse-based software until a year after the Mac introduction, that ended up being defined in the contract as September 1983, since in late 1981 we thought that the Mac would ship in the fall of 1982, and we foolishly didn't let the ship date float in the contract. So Microsoft was within their rights to announce Windows when they did. Apple still needed Microsoft's apps for the Macintosh, so Steve really couldn't cut them off.

Microsoft didn't manage to ship a version of Windows until almost two years later, releasing Windows 1.0 in the fall of 1985. It was pretty crude, just as Steve had predicted, with little of the Mac's thoughtful elegance. It didn't even have overlapping windows, preferring a simpler technique called "tiling". When its utter rejection became apparent a few months later, Bill Gates fired the implementation team and started a new version from scratch, led by none other than Neil Konzen.

Neil's version of Windows, released a couple of years later, was good enough that Apple filed a monumental copyright lawsuit against Microsoft in 1988, but they eventually lost on a technicality (the judge ruled that Apple inadvertently gave Microsoft a perpetual license to the Mac user interface in November 1985).



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Steve Capps Day

Author: Andy Hertzfeld

Date: December 1983

Characters: Jerome Coonen, Steve Capps, Bruce Horn, Larry Kenyon

Topics: Personality, Apple Spirit

Summary: We all dress up in honor of Steve Capps



The software team, all dressed up like Capps



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



The Macintosh software finally seemed to be coming together as the fall of 1983 wore on. The ROMs containing most of the system software were finished, and they seemed to mostly be holding up. Larry Kenyon devised a clever technique for fixing problems in the ROM by patching the nearest system trap to the problem, with the patch code looking on the stack for the ROM address of the caller, which allowed us to fix problems with tiny, surgical incisions, adding some code snippets here or there, without needing to replace large chunks of the ROM with precious RAM as we had originally envisioned.

By the middle of November 1983, MacPaint was already stable enough to ship, even though it was always skirting on the edge of running out of memory if we changed something in the system. MacWrite still had lots of bugs, but its core functionality seemed to be ready, as long as you weren't pushing memory limits. The only vital application that seemed like it could delay our target date in mid-January was the Finder, which was the shell application responsible for launching other applications and managing files.

Bruce Horn was the only programmer working on the Finder, and he was bogging down with a variety of problems, especially involving file copying in low memory conditions. The Finder was built on top of Bruce's resource manager, using features that were barely finished before the ROM freeze. It probably should have been at least a two man project, but Bruce was a brilliant, passionate, independent perfectionist who insisted that he would get it done in time by himself.

Jerome Coonen, the Macintosh software manager, was afraid that Bruce wasn't going to make it. Clearly, Jerome needed to assign someone else to help Bruce finish, but Bruce's perfectionism and high strung temperament made it hard for someone else to dive in and be helpful, especially this late in the game. But luckily, Steve Capps was the perfect man for the job.

Capps was an extraordinarily talented, creative and prolific programmer, who had transferred over to the Mac team from the Lisa printing group in January 1983, giving us a new burst of energy as the original team members began to tire as our sprint to finish slogged into a marathon. Stockily built, with long brown hair and a bushy beard, Capps was one of the few people around who Bruce respected enough to really listen to, and best of all, he possessed a cheerful, easy-going disposition that was the perfect complement to Bruce's high strung intensity.

To minimize distractions, which were mounting as Apple's marketing machine kicked into high gear as the introduction approached, Jerome arranged for Bruce and Capps to move into a small office a few blocks away, on Bubb Road. Capps dove in quickly and took over a bunch of tasks from Bruce. It wasn't always easy, but by the end of December it looked like the Finder was getting more or less back on track.

Jerome thought of a unique way for the software team to show our appreciation for Capps' heroic effort. Like many hackers, Capps dressed idiosyncratically, almost always wearing a long sleeved, white dress shirt with cut-off denim shorts, white socks, and a distinctive type of checkered sneakers called Vans. In fact, Capps had just given everybody on the team a pair of Vans as Christmas presents. Jerome had the idea that we should pay homage to Capps by declaring



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



"Steve Capps Day", and all come in to work dressed up exactly like him for a day.

I had to go to Macy's to buy a white dress shirt, but I managed to come up with a reasonable fascimile of Capps' attire. It was hilarious to see everyone else as we gathered in our "fishbowl" office in Bandley 3 the next morning, all dressed in Capps' unique sartorial style. Even the French and German translators who were visiting us for the month to get European versions underway joined in. The funniest person was probably Patti Kenyon, who was over 8 months pregnant at the time, wearing an extra large white shirt.

Jerome went to get Capps and Bruce from their enclave, telling them there was an important meeting that required their attendance. We could hardly hold back the laughter as an unsuspecting Capps walked in, and then we finally cracked up and gave him a round of applause when he realized what was going on.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



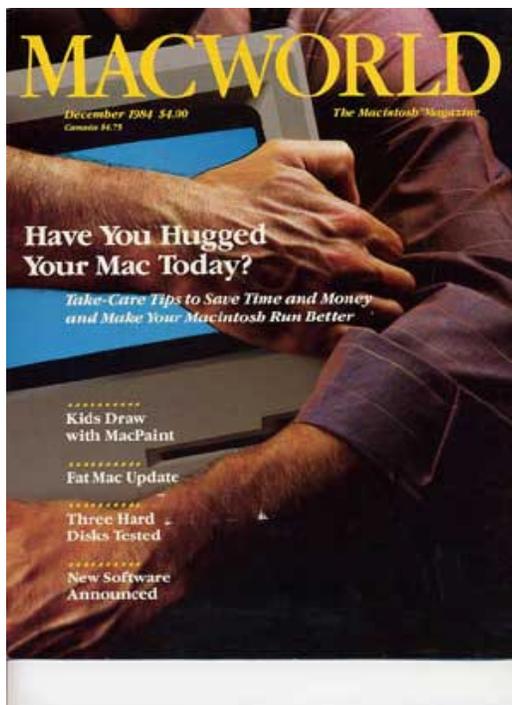
Where'd That Hand Come From?

Author: Eric Barnes

Date: April 1984

Topics: Marketing

Summary: How the hand and the mouse came together!



Early MacWorld cover, Dec 84



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



In the mid '70s, I became San Francisco's first professional male hand model. That's a whole other story in itself. But for three or four years, there were no others in the field and I built a nice (and profitable) reputation, shooting hand shots for all manner of products and companies.

Around April, 1984, I got a call from one of my favorite photographers, Dennis Gray, booking me for an hour, asking me to bring a variety of shirt colors, a couple of sports jackets and a suit jacket.

I showed up, a pale blue shirt was chosen, and we began the shoot. I was simply to hold this weird little plastic thing in a variety of ways. After about 45 minutes, Dennis thought we had the shot. I then asked him, "What *is* this thing?" Dennis replied, "Darned if I know, they call it a 'mouse.'"

A year later, I bought my first Mac, a 512. Unpacking it, I saw that virtually every surface of the boxes, plus all the manuals had this hand on it. Oddly, I didn't remember the shot amidst all the many I'd done in the meantime. It only dawned on me a few days later that it was my hand!

The connection also brought me a variety of hand shots for MacWorld magazine (Dec 84 cover pictured) and I was later told that my hand was the only 'live' thing on MacWorld covers for its first five years of publication.

So though I didn't have anything to do with the creation of the Mac, I did contribute to it's great marketing effort.

The mouse shot, btw, paid me only \$100. I don't suppose Steve would be willing to pay residuals at this late date?



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Handi capped

Author: Andy Hertzfeld
Date: 1985
Characters: Steve Jobs, Jean-Louis Gassee
Topics: Personality Clashes, Personality
Summary: The reason that Steve parks in the handi capped space

Most of the anecdotes that I've written for Folklore are based on incidents that I observed myself, but sometimes a second or third hand story is just too good to pass up. I have to issue a disclaimer here that I didn't actually witness the punch line to this one, and it certainly seems too good to be true.

Steve Jobs was not the most considerate individual at Apple, and he had lots of ways to demonstrate that. One of the most obvious was his habit of parking in the handi capped spot of the parking lot - he seemed to think that the blue wheelchair symbol meant that the spot was reserved for the chairman.

Whenever you saw a big Mercedes parked in a handi capped space, you could be sure that it was Steve's car (actually, it was hard to be sure otherwise, since he also had a habit of removing his license plates). This sometimes caused him trouble, since unknown parties would occasionally retaliate by scratching the car with their keys.

Anyway, the story is that one day Apple executive Jean-Louis Gassee, who had recently transferred to Cupertino from Paris, had just parked his car and was walking toward the entrance of the main office at Apple when Steve buzzed by him in his silver Mercedes and pulled into the handi capped space near the front of the building.

As Steve walked brusquely past him, Jean-Louis was heard to declare, to no one in particular - "Oh, I never realized that those spaces were for the emotionally handi capped...".



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Are You Gonna Do It?

Author: Andy Hertzfeld
Date: February 1985
Characters: Burrell Smith, Steve Jobs
Topics: Management, Quitting
Summary: Burrell quits Apple, but fails to live up to his threat

No matter how much resolve you could muster, it was still difficult to quit Apple if Steve wanted you to stay. You'd have to sit down with him for a reality distortion session, which was often effective at getting people to change their minds. One day, a few of us were talking about strategies to overcome Steve's persuasiveness.

"I've got it!", said Burrell. "I know the perfect way to quit that will nullify the reality distortion field."

Of course we wanted to know how he could do that.

"I'll just walk into Steve's office, pull down my pants, and urinate on his desk. What could he say to that? It's guaranteed to work." We laughed, thinking that not even Burrell would have the guts to do that.

A year and a half later, it finally was time for Burrell to quit, after months of scuffling with Bob Belleville and some of the other managers, who wanted to cancel the "Turbo Mac" project (a redesigned, faster Mac based on a semi-custom chip, with an internal hard drive) that Burrell was working on with Brian Howard and Bob Bailey.

Burrell told Bob Belleville (who was probably relieved, since he knew that Burrell didn't respect him) and the human resources department that he was quitting, and then made an appointment to see Steve that afternoon. When he walked into Steve's office, he was surprised to see Steve grinning at him.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



"Are you gonna do it? Are you really gonna do it?", asked Steve. Somehow, word about the urination threat had gotten back to Steve, and he was genuinely curious if Burrell would go through with it or not.

Burrell looked Steve in the eye. "Do I have to? I'll do it if I have to."

Steve's expression gave him the answer, so Burrell turned and walked out of the office, no longer an Apple employee.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Evolution Of A Classic

Author: David Ramsey
Date: March 1985
Characters: Bill Atkinson, Gene Pope, David Ramsey, Rich Page
Topics: Maintenance, MacPaint
Summary: MacPaint 2.0

(picture missing)

MacPaint 2.0's "About Box" with the Zebra Lady

I was hired at Apple in early 1986. Oddly, I was given nothing to do when I started, so after setting up my cube and getting my Lisa (all Mac programming was done on the Lisa back then) running, I wandered over to my manager's cube and asked if he had the source to MacPaint. He had-- he handed me a single Lisa diskette with all of MacPaint on it.

MacPaint was the first Mac program I'd ever seen; Bill Atkinson was demonstrating it at Computers Plus, a local Apple store run by Mark Wozniak and Dick Applebaum. I was completely awestruck at how fast the computer could "sling bits" on the screen. But there were still things I wanted to change: I wanted to be able to move and resize the image window, for example.

After a week or so of playing around I showed my manager (Gene Pope) what I'd done. I had just been killing time, since nobody had told me to do anything, but Gene asked me to add more features with an eye towards a "MacPaint 2.0".

And so it begins

At first I was nervous: Bill Atkinson is a hard act to follow, and I was also worried what he might think about some newbie hire taking over his baby. But he was tremendously supportive (and it probably also helped that he was deeply involved with his new project, Hypercard) and even took hours of time to go over



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



his hand-optimized 68000 assembly language routines to explain how they worked. I still remember him peering intently at me, making sure I understood the importance of what he was saying: "Unroll the loops", "keep the registers full"...

The day I finally figured out how "regions" in QuickDraw worked was an epiphany, really.

Specs? We don't need no steenking specs!

It seems strange by modern standards, but MacPaint 2.0 never had a product spec. There was no marketing requirements document. There were no design meetings. There wasn't even a product manager until the last few months of work. Many people had opinions and suggestions (and the then-new tear-off menus came directly from HyperCard, courtesy of Bill) but ultimately the product was what I wanted it to be.

Of course, I had more leeway than Bill did with the original MacPaint since I didn't have to worry about running on a 128K Mac-- a 512K computer was the minimum. Thus MacPaint gained a sizeable, movable painting window; multiple document support; large clipboards, the "snapshot" capability with the "magic eraser", and so forth.

Some features were pretty subtle: for example, in response to a request from Radius, MacPaint 2.0 would automatically move its window down if the user took advantage of a Radius monitor's capability to enlarge the standard menu bar!

Fun hardware note

Most of MacPaint was developed on a "Big Mac" prototype-- a computer that was a design study for the next Macintosh. Basically it was a 16mHz 68020 version of a Mac Plus. Cases were never made, so it was simply a 1 foot square circuit board mounted on a piece of wood, connected to a 10 megabyte SCSI hard drive.

I used the Big Mac prototype since it was faster and more reliable than the Macintosh II prototypes available. It was never produced, and designer Rich Page left Apple to work at NeXT shortly after his design "lost" to the slotted Mac II.

I still have the Big Mac I used at Apple. I wonder if it still works.

The Zebra Lady

Test versions of MacPaint had different images, generally from comic books, in the "About" box. The last beta version showed the nude upper torso of a zebra-striped woman taken from the Olivia de Berardinis painting "Zebra Lady". With the covert approval of all involved, this version made it into the release: you could show the Zebra Lady image by holding down the tab and space keys while selecting "About MacPaint".

When the artist found out, she took it well. I have a signed copy of the original print framed to this day.

Shelf Life



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



MacPaint 2.0 was introduced in late 1987 (did it really take me that long to write?) and remained on the market until fall 1998. I think this is a record of some sort for personal computer software.

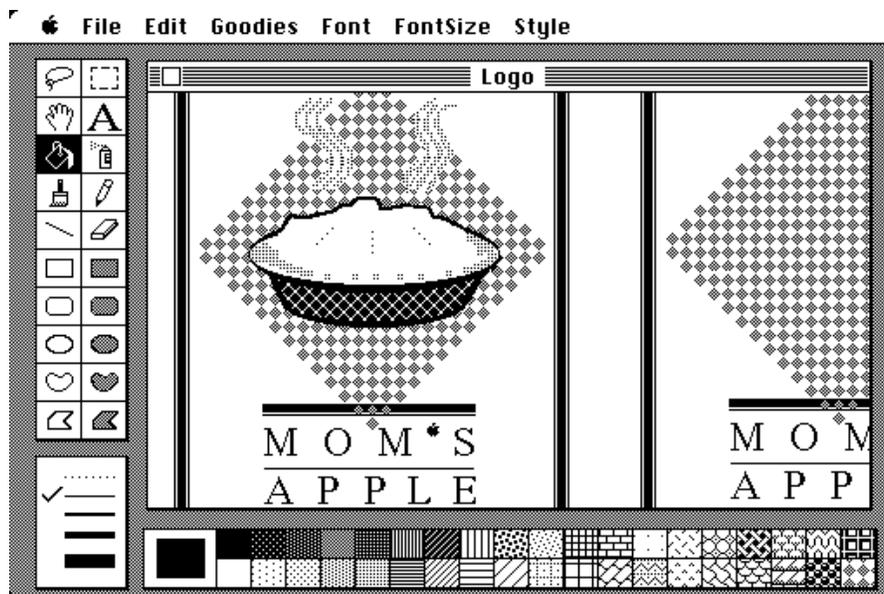
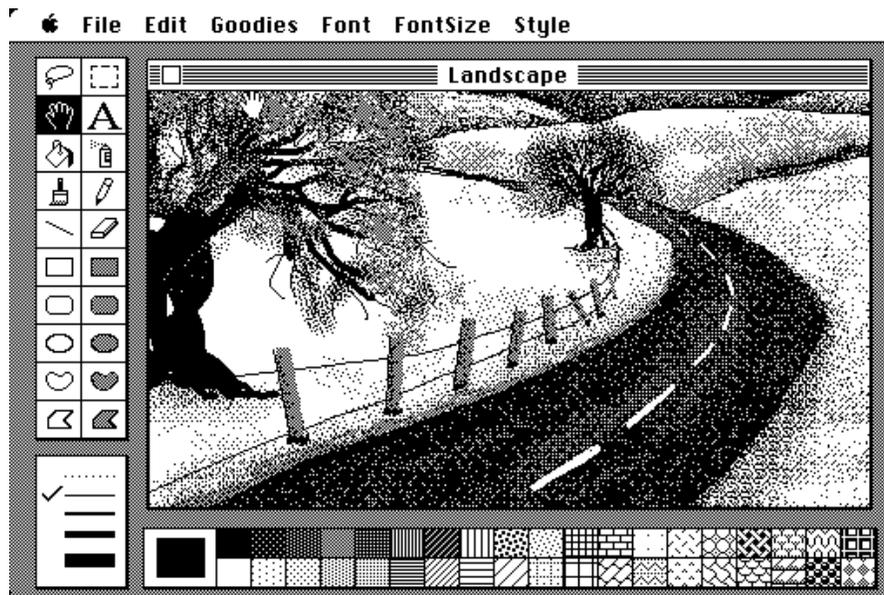
[MacPaint v1 screen shots]



Logo



MacPaint 1.5





APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Five Different Macintoshes

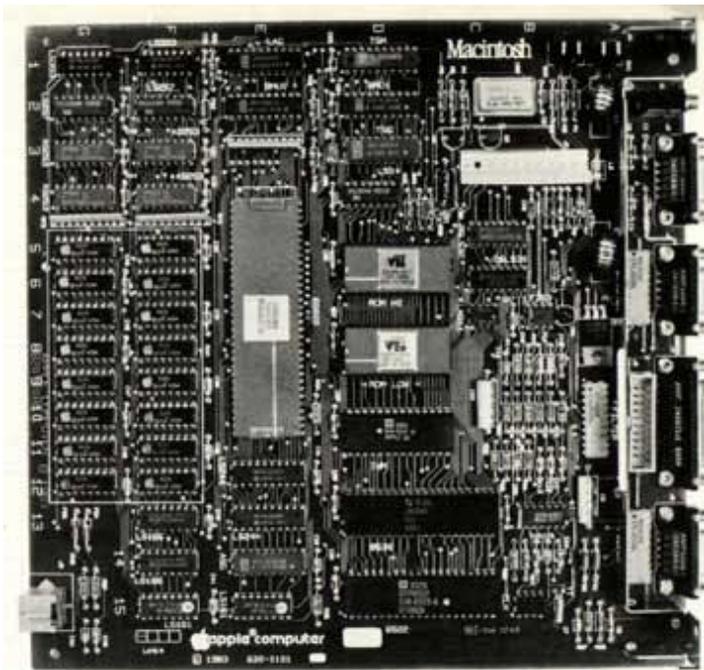
Author: Andy Hertzfeld

Date: undated

Characters: Burrell Smith, Bud Tribble, Wendell Sander, Steve Jobs, Jef Raskin, Bill Atkinson

Topics: Hardware Design, Technical

Summary: Burrell actually designed 5 different Macintoshes



The final version of the Mac digital board



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



The awesomely creative design of the Macintosh digital board was always the seed crystal of brilliance at the core of the project, but there wasn't just one design; Burrell redesigned the digital board four different times as development proceeded, finally arriving at the shipping design in the fall of 1982.

The first Macintosh digital board, designed in late 1979, was based on Jef Raskin's specifications: it had a Motorola 6809E microprocessor, 64K of memory, and a 256x256 black and white, bit-mapped graphics display. Even in this first design, Burrell was using his trademark 'PAL' chips, which were small, programmable logic arrays, to provide all of the system glue, so the system had a very low chip count.

The first Macintosh was a cute little computer, but it was fundamentally limited by the 6809 microprocessor, which only had 16 bits of address space. Bill Atkinson was doing incredible work on the Lisa project using Motorola's 68000 microprocessor, with its capacious 32 bit registers and 24 bit address space. Bud Tribble, the Mac's only software person, was living at Bill's house, and watching the extraordinary progress on Lisa's graphics package. He began to wonder if it was possible for the Macintosh to use the 68000, so it could run Bill's graphics routines.

Bud began asking Burrell if it was feasible to include the 68000 in a low cost design. The 68000 was expensive enough on its own, but its 16 bit memory bus required twice as many RAM chips as the 6809, so the overall cost was significantly more expensive. But Burrell thought about the problem and came up with a characteristically brilliant idea for his second Macintosh design.

The idea was what Burrell called a "bus transformer" circuit, built out of PAL chips, which adapted the 68000 to an 8 bit memory bus by exploiting the fast "page mode" access mode of the RAMs. The new Macintosh, designed over the Christmas break at the end of 1980, featured an 8 megahertz 68000, 64K of RAM, and a 384 by 256 bit mapped display. It was 60% faster than the Lisa (which used a 5 megahertz 68000) but a lot less expensive.

When Steve Jobs caught wind of what Burrell had come up with - an Apple II priced machine that blew away the Lisa, he became really excited, and saw that Burrell's machine could become the future of Apple. Steve's attention was the beginning of the end for Jef, though, who despised parts of Steve's personality and couldn't put up with Steve's courting of Burrell and Bud. Steve took over the project in January 1981, and the Macintosh entered the post-Jef era, on track to becoming a real product.

Burrell's third Macintosh design was done in June 1981. The main reason was that he fell in love with a communications chip called the SCC. The SCC could support a built-in local area network, making AppleTalk possible with no additional hardware, as well as providing nice buffered serial ports with interrupts and other hardware features. At this point it also was becoming obvious that we needed at least 128K of memory to support the user interface, so he also added a second row of RAM chips.

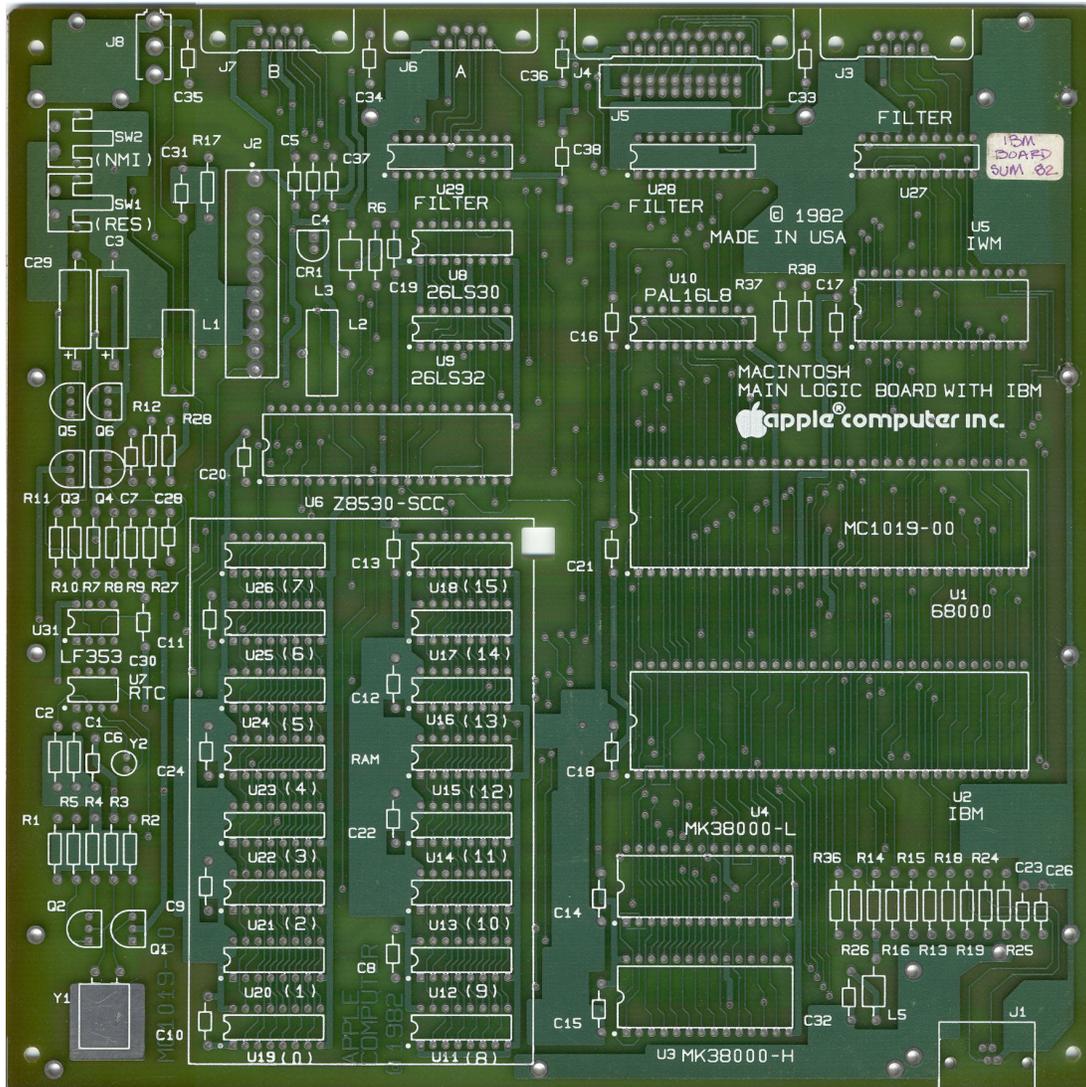
Around the end of 1981, Burrell met some engineers who were doing custom LSI chips, which were very flexible and powerful but very time consuming to design, since the software tools were still in their infancy. Burrell decided that he wanted to take a shot at building the Macintosh around a single custom chip, and



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



convinced Steve that he could pull it off. Instead of sitting around doing nothing while the software team finished the product, he would work with a couple of experienced LSI designers and redesign the Mac around what he called the "Integrated Burrell Machine".



Earlier in the year, Wendell Sander, the designer of the Apple III and one of Apple's best engineers, did a small custom chip that crammed all the functionality of Woz's disk controller into a single chip. It was called the "IWM" chip, which stood for the "Integrated Woz Machine", since Woz's disk controller is really an elaborate state machine, but it also stood for the "Integrated Wendell Machine". So when Burrell envisioned a single custom chip for the Macintosh, he called it the "IBM" chip, for "Integrated Burrell Machine", relishing the potential confusion the name might cause.

In the beginning of 1982, the original 68000 design was more than a year old, and the software was nowhere near finished, so Burrell was afraid some of the

ANECDOTES ABOUT THE DEVELOPMENT OF APPLE'S ORIGINAL MACINTOSH COMPUTER



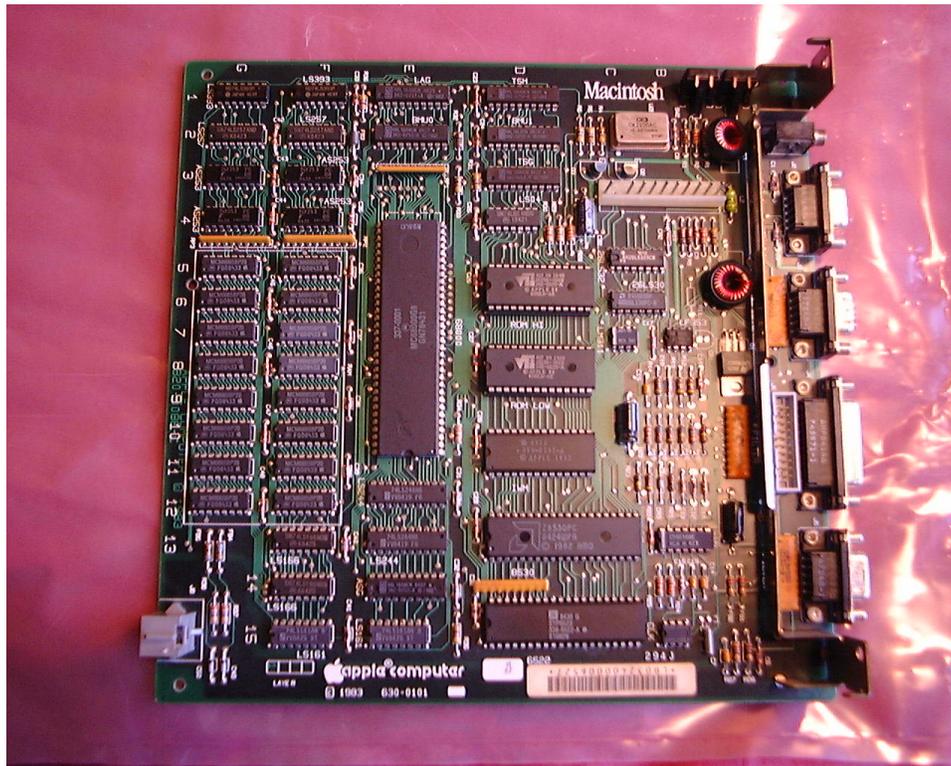
APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



trade-offs of the original design were no longer current. He used the expansive canvas of a custom chip, where additional logic was almost free, to update the architecture. The most important decision was admitting that the software would never fit into 64K of memory and going with a full 16-bit memory bus, requiring 16 RAM chips instead of 8. The extra memory bandwidth allowed him to double the display resolution, going to dimensions of 512 by 342 instead of 384 by 256. He also added bells and whistles like a fancy, DMA-fed sound generator with four independent voices. This was the fourth version of the Macintosh design.

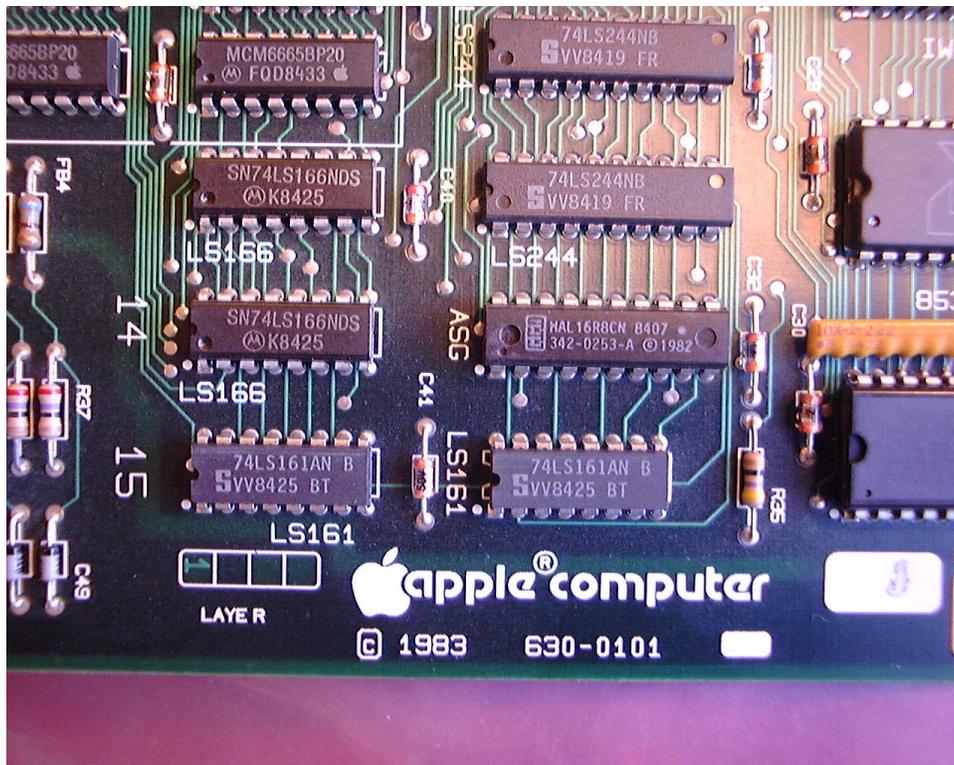
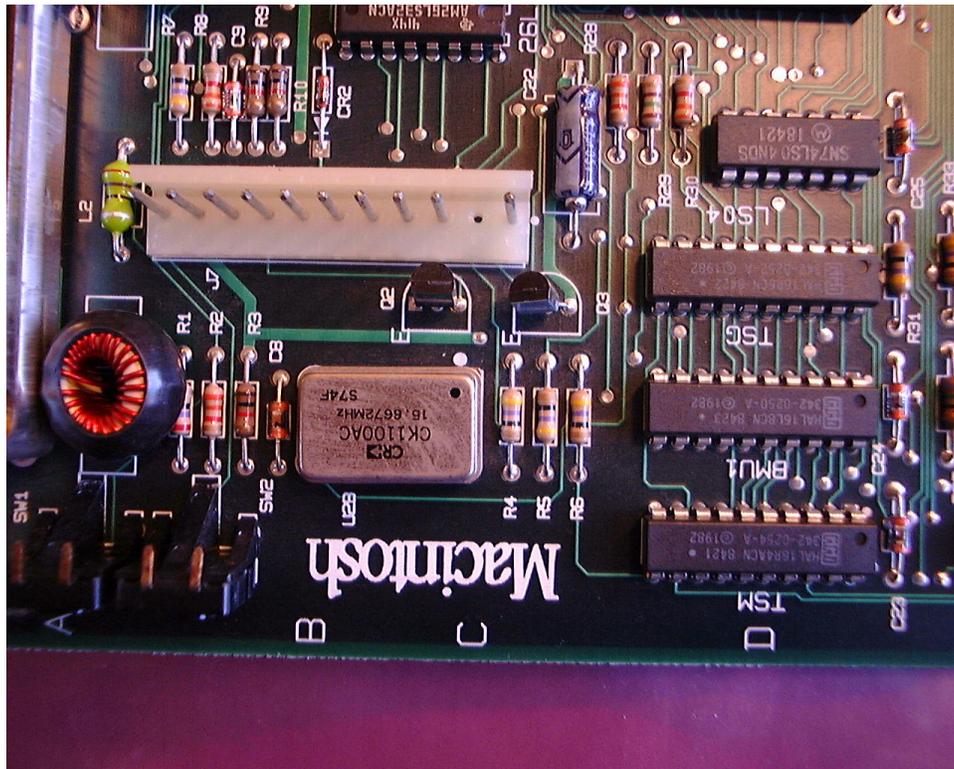
But Burrell had never done a custom chip before. He was counting on experts working with our partner VLSI Design, mainly Doug Fairbairn and Martin Haerberli, who successfully made a custom chip for an optical mouse, but they had never really worked against commercial deadlines, and a lot of tension began to build as the schedule started to slip.

Some more of the details of the Integrated Burrell Machine project are told elsewhere, but eventually, after the first silicon from an early version didn't work as expected, Burrell became frustrated and decided that it was too risky to continue. He realized that he could do a design with PALs that achieved most of the gains from the Integrated Burrell Machine, without needing the custom chip. So, in August 1982, he quickly designed the fifth iteration of the Macintosh, the one that actually shipped in January 1984, by adapting his earlier PAL-based design to one that had a 16-bit memory bus with a 512 by 342 display and some other features invented for the custom chip.



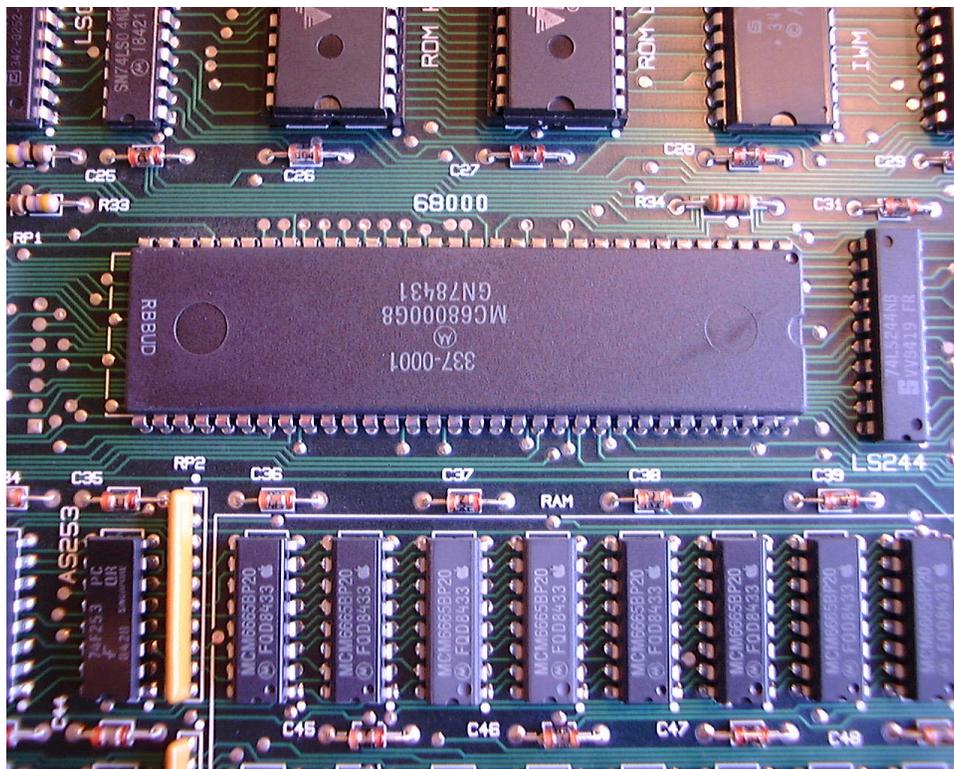
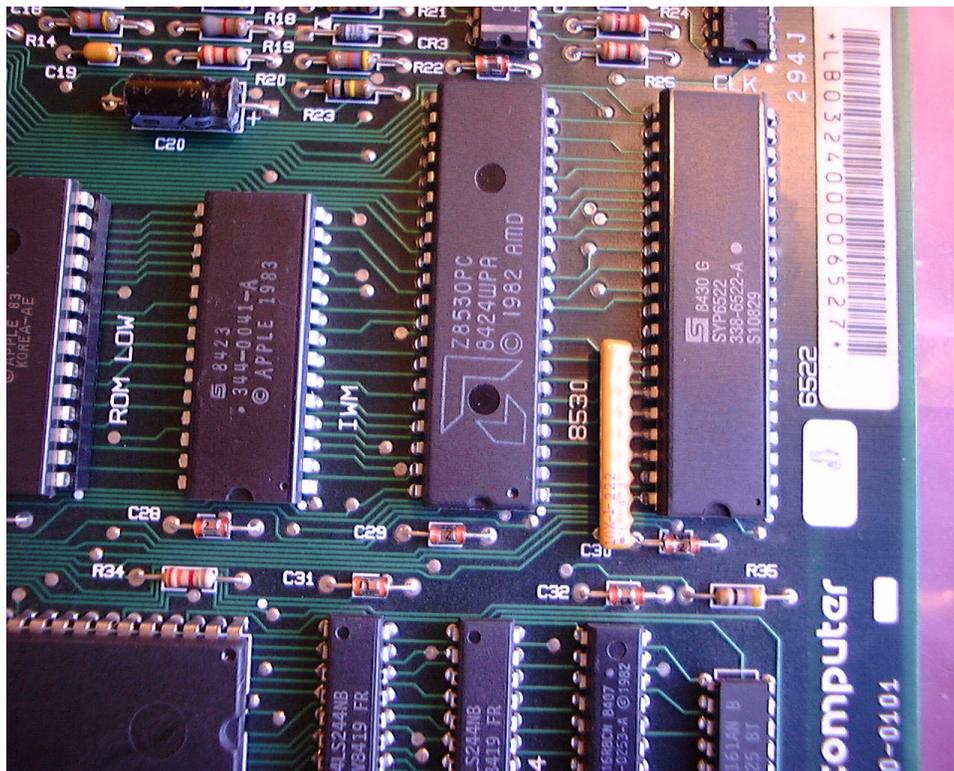


APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



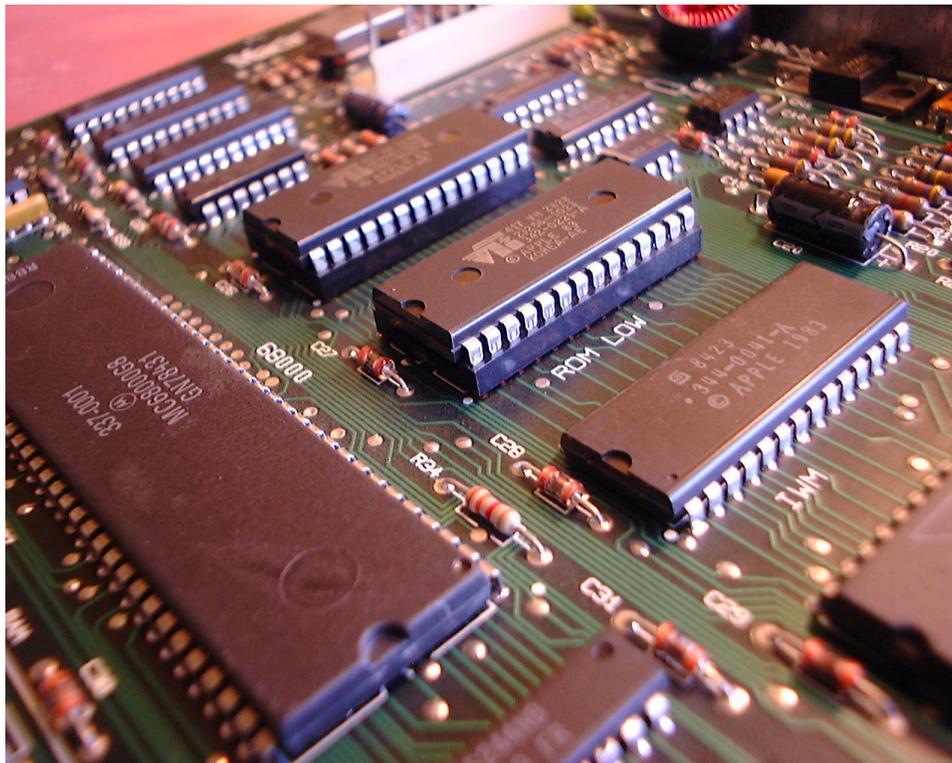


APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION





APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION





APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



The Apple Spirit

Author: Andy Hertzfeld

Date: undated

Characters: Andy Hertzfeld

Topics: Apple Spirit, Inspiration

Summary: An essay I wrote about Apple's values that was published in MacWeek

Toward the end of 1988, I wrote an essay that was published in MacWeek entitled "The Apple Spirit". It was about the creative magic that I found in the Apple II, and how we were able to transplant it into the Macintosh. The essay articulates the values behind lots of the stories collected here, so I thought it was worth including.

The Apple Spirit

November 29, 1988

The best purchase of my life occurred in January 1978 when I spent \$1295 plus tax (most of my life savings at the time) on an Apple II microcomputer (serial number 1703) with 16K bytes of RAM. I was instantly delighted with it, and the deeper I dug into it, the more excited I became. Not only could I finally afford to have my own computer, but the one I got turned out to be magic; it was better than I ever thought it possibly could be!

I started spending most of my free time with my Apple, and then most of my not-so-free time, exploring the various technical aspects of the system. As I taught myself 6502 assembly language from the monitor listing that came with the machine, it became clear to me that this was no ordinary product; the coding style was crazy, whimsical and outrageous, like every other part of the design, especially the hi-res graphic screen; it was clearly the work of a passionate artist. Eventually, I became so obsessed with the Apple II that I had to go to work at the place that created it. I abandoned graduate school and started work as a systems programmer at Apple in August 1979.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Even though the Apple II was overflowing with both technical and marketing genius, the best thing about it was the spirit of its creation. It was not conceived or designed as a product in the usual sense; it was just Steve Wozniak trying to impress himself and his friends. Most of the early Apple employees were their own ideal customers. The Apple II was simultaneously a work of art and the fulfillment of a dream, shared by Apple's employees and customers. Its unique spirit was picked up and echoed back by third party developers, who sprung out of nowhere with innovative applications.

The personal computer industry began to grow and evolve very rapidly when larger companies realized the extraordinary potential of personal computers. Apple's sales took off like a skyrocket as the Apple II became accepted as an established industry standard. By the time the early 1980s rolled around, many opportunists had come to both Apple and the personal computer industry, people whose only concern was to make as much money as possible. I started to become disillusioned when Apple hired many professional managers who didn't appreciate the magic of the Apple II; many of them would have been just as happy selling refrigerators. I probably would have left Apple sometime in 1981 if I hadn't run across a tiny, sloppily wire-wrapped digital board created by Burrell Smith, a young technician who worked in the service department.

Burrell worshipped Woz's Apple II design and had forged an idiosyncratic design style that was even crazier than Woz's, using many clever tricks to coax enormous functionality out of the minimum number of chips. Somehow, Burrell's embryonic Macintosh board reeked of the same creative spirit so prevalent in the Apple II; as soon as I saw it, I knew that I had to work on the project.

Steve Jobs also became enamored with Burrell's circuit board and quickly took over the tiny design group, moving it to a remote part of the company and inspiring us with a grand vision. The Apple II had broken through an important price barrier, making a useful personal computer affordable to ordinary individuals, but it was still much too hard for most non-technical people to master. The Macintosh would harness the potential of Motorola's 68000 microprocessor to become the first personal computer that was both easy to use and affordable. We thought that we had a chance to create a product that could make computers useful to ordinary people and thereby truly change the world.

The Macintosh design team was inspired by Woz's original design and tried to recapitulate its innovative spirit. Again, we were our own ideal customers, designing something that we wanted for ourselves more than anything else. Although Apple was already a large company by then, Steve's unique position in the organization enabled him to maintain the Macintosh group as a little island where Apple's original values could flourish and grow. The Macintosh was released in January 1984 and eventually became a very successful product.

The personal computer industry has continued to grow and change since the introduction of the original Macintosh. Apple has become [in 1988] a four billion dollar enterprise, and I often fear that they have lost touch with their original values. Yet I remember having similar worries right before starting work on the Mac. I'm sure that there are little groups at Apple right now, inspired by the Macintosh in exactly the same way that we were inspired by the Apple II. The great challenge facing Apple's management is to allow those groups to follow their hearts and imaginations, uncompromised by the inevitable



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



politics of large organizations. I hope that I will be able to buy a new Apple computer in 1991 that is not a Macintosh or an Apple II, but rather an entirely new system that once again shares the maverick spirit of its illustrious ancestors.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Busy Being Born

Author: Andy Hertzfeld
Date: undated
Characters: Bill Atkinson, Bud Tribble, Steve Jobs
Topics: Origins, User Interface
Summary: A visual history of the development of the Lisa/Macintosh user interface

The Macintosh User Interface wasn't designed all at once; it was actually the result of almost five years of experimentation and development at Apple, starting with graphics routines that Bill Atkinson began writing for Lisa in late 1978. Like any evolutionary process, there were lots of false starts and blind alleys along the way. It's a shame that these tend to be lost to history, since there is a lot that we can learn from them.

Fortunately, the main developer of the user interface, Bill Atkinson, was an avid, lifelong photographer, and he had the foresight to document the incremental development of the Lisa User Interface (which more or less became the Mac UI after a few tweaks) with a series of photographs. He kept a Polaroid camera by his computer, and took a snapshot each time the user interface reached a new milestone, which he collected in a loose-leaf notebook. I'm excited to be able to reproduce and annotate them here, since they offer a fascinating, behind-the-scenes glimpse of how the Mac's breakthrough user interface was crafted.

The images are scaled so they easily fit onto a typical screen, but you can click on them for larger versions that show more detail.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



The first picture in Bill's notebook is from Bill's previous project, just before starting work on the Lisa: Apple II Pascal. The high performance graphics routines that Bill wrote for Apple II Pascal in the fall of 1978 led right into his initial work on the Lisa.

The center and right photos, from the spring of 1979, were rendered on the actual Lisa Display system, featuring the 720 by 360 resolution that remained constant all the way through to the shipping product. No Lisa existed yet; these were done on a wired wrapped prototype card for the Apple II. The middle picture shows the very first characters ever displayed on a Lisa screen; note the variable-width characters. The rightmost picture has more proportional text, about the the Lisa display system, rendered in a font that Bill designed by hand.



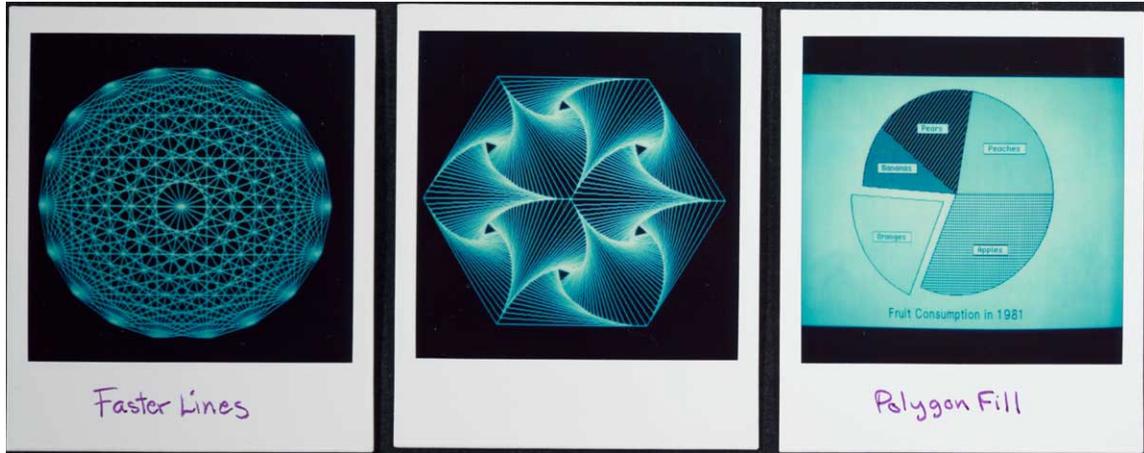
The leftmost picture illustrates the first graphics primitives that Bill wrote for LisaGraf (which was eventually renamed to QuickDraw in 1982) in the spring of 1979, rendering lines and rectangles filled with 8x8 one-bit patterns. The power and flexibility of the patterns are illustrated in the rightmost shot, which were our poor man's substitute for color, which was too expensive (at the required resolution) in the early eighties.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



The middle picture depicts the initial user interface of the Lisa, based on a row of "soft-keys", drawn at the bottom of the screen, that would change as a user performed a task. These were inspired from work done at HP, where some of the early Lisa designers hailed from.



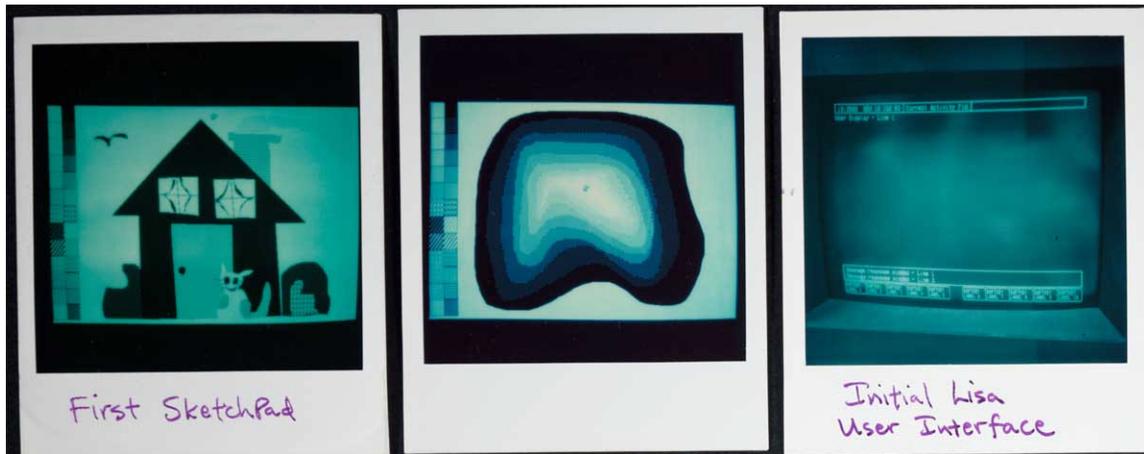
Here are some more demos of the initial graphics routines. Bill made line-drawing blindingly fast with an algorithm that plotted "slabs" of multiple pixels in a single memory access. The rightmost picture shows how non-rectangular areas could be filled with patterns, too.



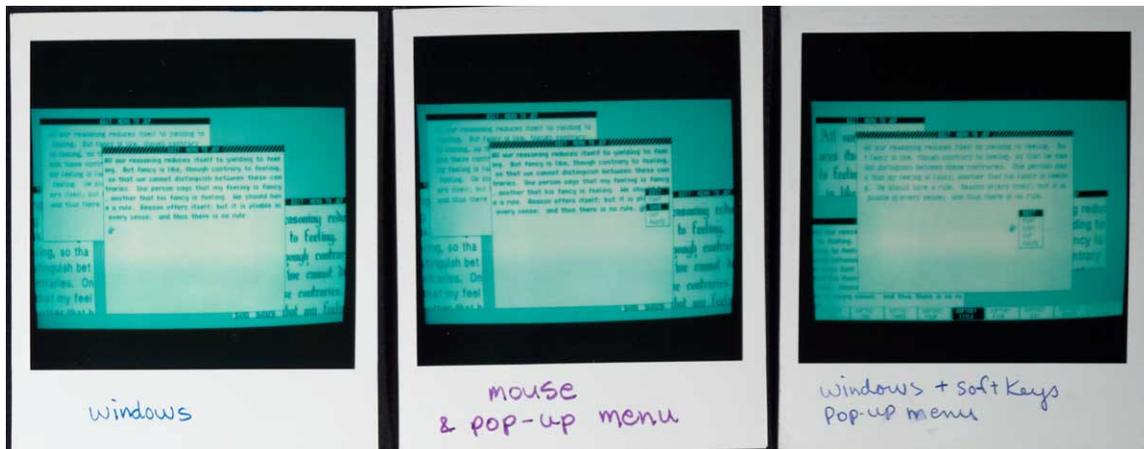
Here are some scanned images, showing off Lisa's impressive resolution for the time, which Bill scanned using a modified fax machine. He was always tweaking the half-toning algorithm, which mapped gray scales into patterns of monochrome dots. Bill had made versions of these for the Apple II that Apple distributed on demo disks, but these higher resolution Lisa versions were much more impressive.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



The left and middle pictures show off the first sketch program, an early ancestor of MacPaint, that allowed mouse-based drawing with patterns and a variety of brush shapes. I think these are perhaps a bit out of sequence, done in early 1980. The rightmost picture shows the final soft-key based UI, which is about to change radically...



...into a mouse/windows based user interface. This is obviously the biggest single jump in the entire set of photographs, and the place where I most wish that Bill had dated them. It's tempting to say that the change was caused by the famous Xerox PARC visit, which took place in mid-December 1979, but Bill thinks that the windows predated that, although he can't say for sure.

The leftmost picture shows different fonts in overlapping windows, but we didn't have a window manager yet, so they couldn't be dragged around. The middle window shows the first pop-up menu, which looks just like SmallTalk, as does the simple, black title bar. The rightmost picture shows that we hadn't given up on the soft-keys yet.

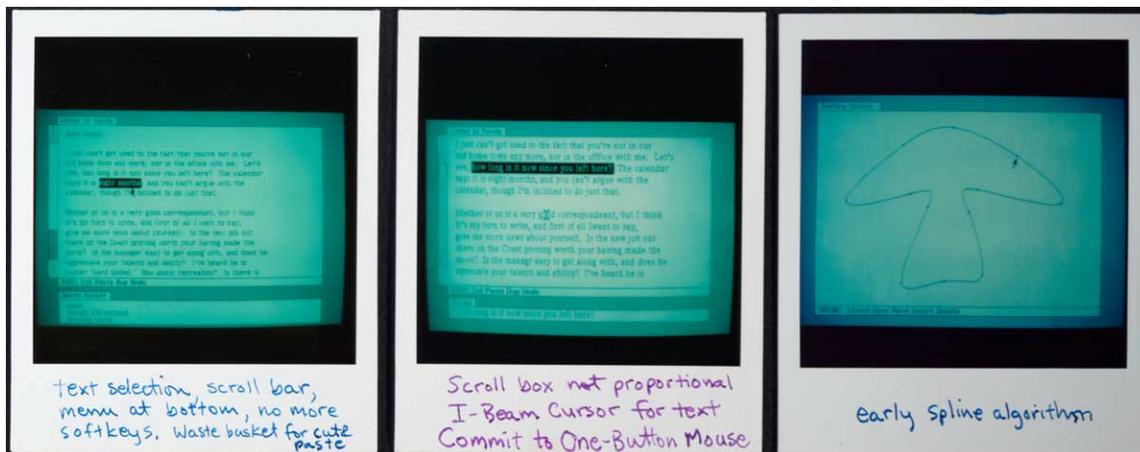


APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



By now, it's the spring of 1980 and things are starting to happen fast. The leftmost picture shows the earliest text selection, using a different highlighting technique than we ended up with. It also shows a "command bar" at the bottom of the screen, and that we had started to use non-modal commands (make a selection, then perform an action, instead of the other way around).

The middle picture shows the very first scroll bar, on the left instead of the right, before the arrow scroll buttons were added. It also has a folder-tab style title bar, which would persist for a while before being dropped (Bill says that at that point, he was confusing documents and folders). The right most photo shows that we adopted the inverse selection method of text highlighting.



By the summer of 1980, we had dropped the soft-keys. The leftmost photo shows that we had mouse-based text editing going, complete with the first appearance of the clipboard, which at that point was called "the wastebasket". Later, it was called the "scrap" before we finally settled on "clipboard." There was also a Smalltalk style scrollbar, with the scroll box proportional to the size of the document. Note there are also two set of arrows, since a single scrollbar weirdly controlled both horizontal and vertical scrolling.

ANECDOTES ABOUT THE DEVELOPMENT OF APPLE'S ORIGINAL MACINTOSH COMPUTER

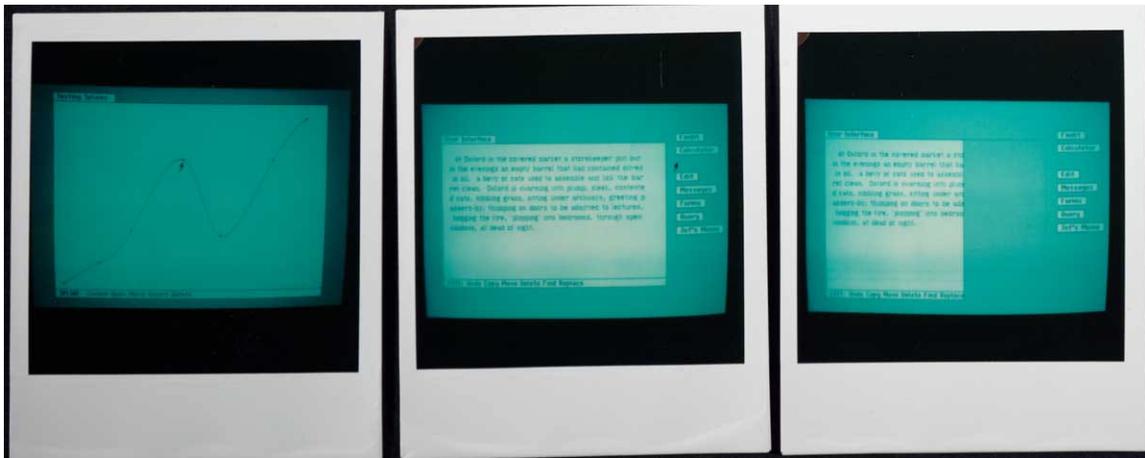


APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



The next picture shows that we dropped the proportional scroll box for a simpler, fixed-size one, since we were afraid users wouldn't understand the proportionality. It also shows the I-Beam text cursor for the first time. At this point, we were finally committed to the one-button mouse, after a long, protracted internal debate.

The right most picture shows Bill playing around with splines, which are curves defined by a few draggable control points. QuickDraw didn't end up using splines, but the picture is still notable for the first appearance of the "knobbie" (a small, draggable, rectangular affordance for a point).



By now, it's the fall of 1980. The middle picture shows us experimenting with opened and closed windows, which was eventually dropped (but it made a comeback in the 1990s and is in most systems today one way or another). The right most picture shows the first window resizing, by dragging a gray outline, although it's not clear how resizing was initiated.



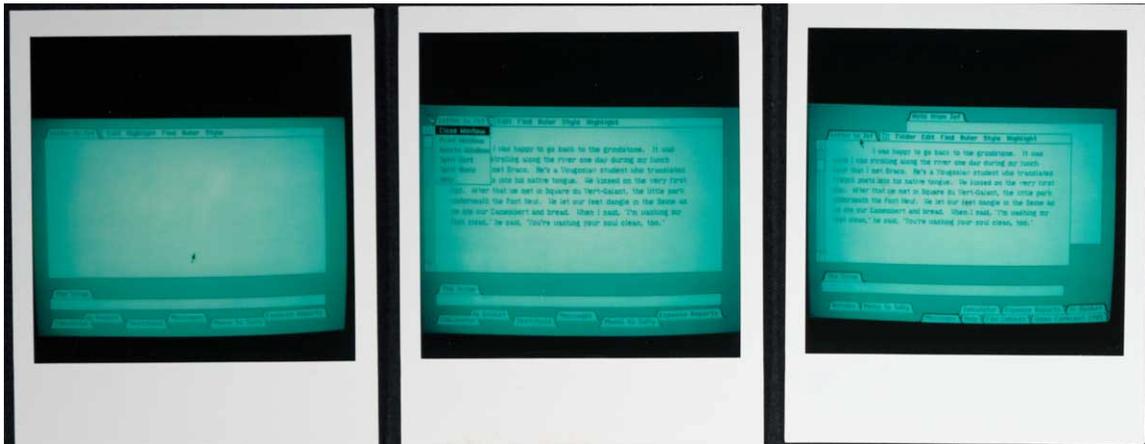
The middle picture shows that windows can be repositioned by dragging a gray outline. We wanted to drag the whole window, like modern user interfaces do today, but the processors weren't fast enough in those days. As far as I know, the NeXTStep was the first system to do it the modern way.



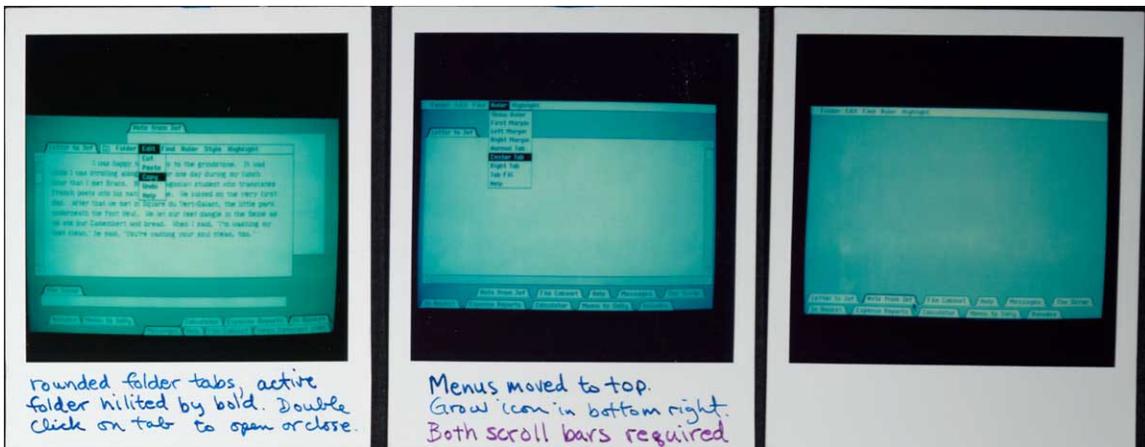
APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



The right most picture shows the first appearance of pull-down menus, with a menu bar at the top of the window instead of the top of the screen, which is the way Windows still does things. By this point, we also gave up on using a single scroll bar for both horizontal and vertical scrolling; it's looking very much like what the Mac shipped with in 1984 now.



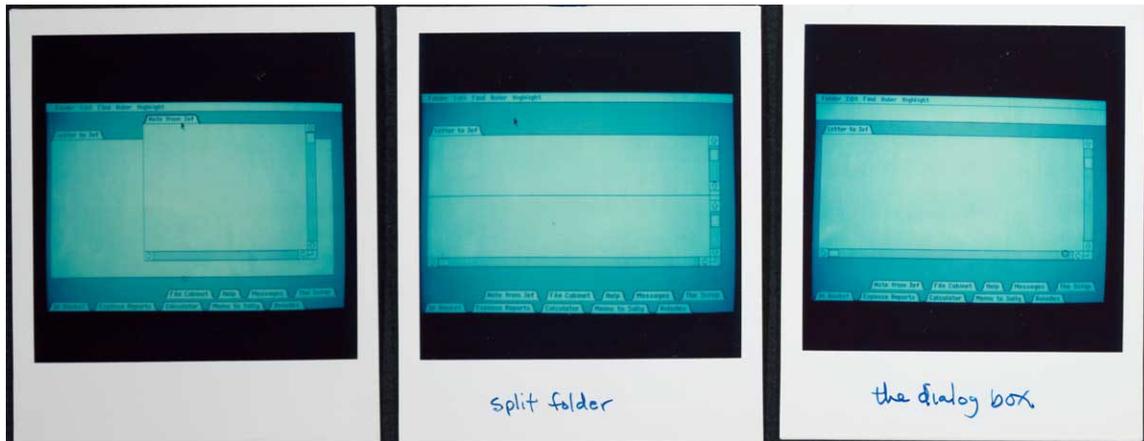
This set of pictures illustrates the Lisa desktop, circa the end of 1980, with a tab-shaped title, followed by a menu bar attached to the window. Windows could be reduced to tabs on the desktop. We've also changed the name of the clipboard to "the scrap", an old typesetting term.



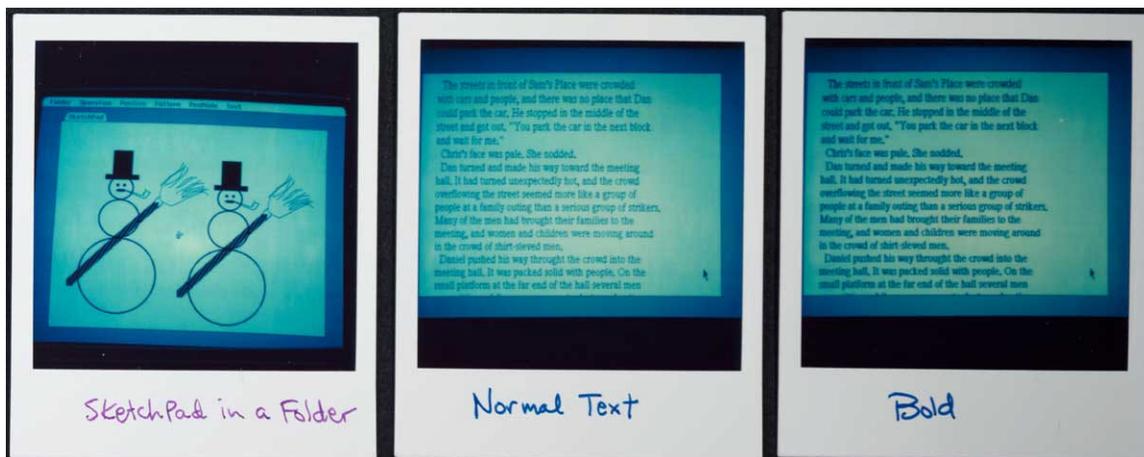
The leftmost picture mentions the first use of double-click, to open and close windows. The middle picture represents a real breakthrough, by putting the menu bar at the top of the screen instead of the top of each window. The menu bar contains the menus of the "active folder", which is the topmost window. By this point, the grow icon found its way to the bottom right, at the intersection of the horizontal and vertical scrollbars, which stuck. This is the first picture which is really recognizable as the shipping Macintosh.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



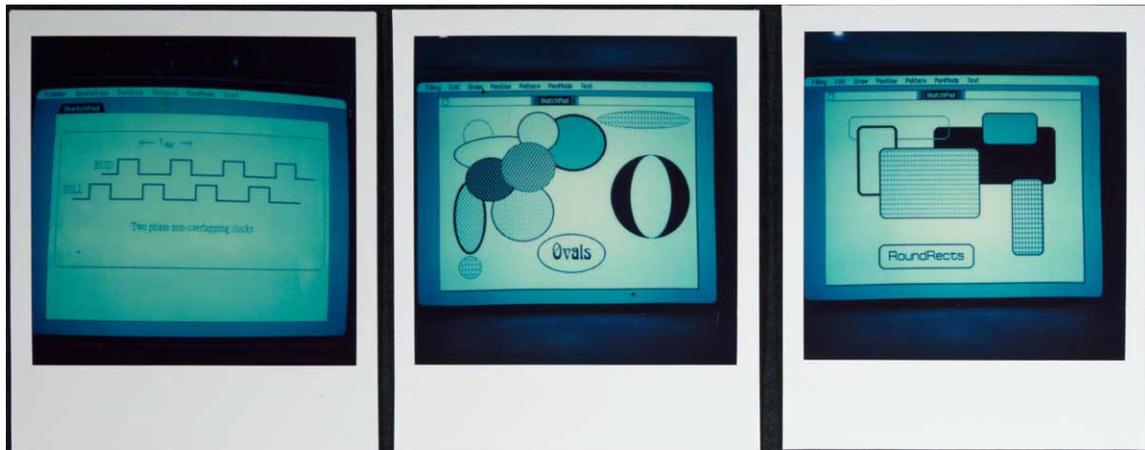
By now, it's early 1981, and things are beginning to shape up. The leftmost picture shows a window with scrollbars that look a lot like the ones that shipped. The middle folder illustrates split views, which were used by Lisa's spreadsheet application. The rightmost picture contains the first appearance of a dialog box, which at the time ran the entire length of the screen, just below the menu bar.



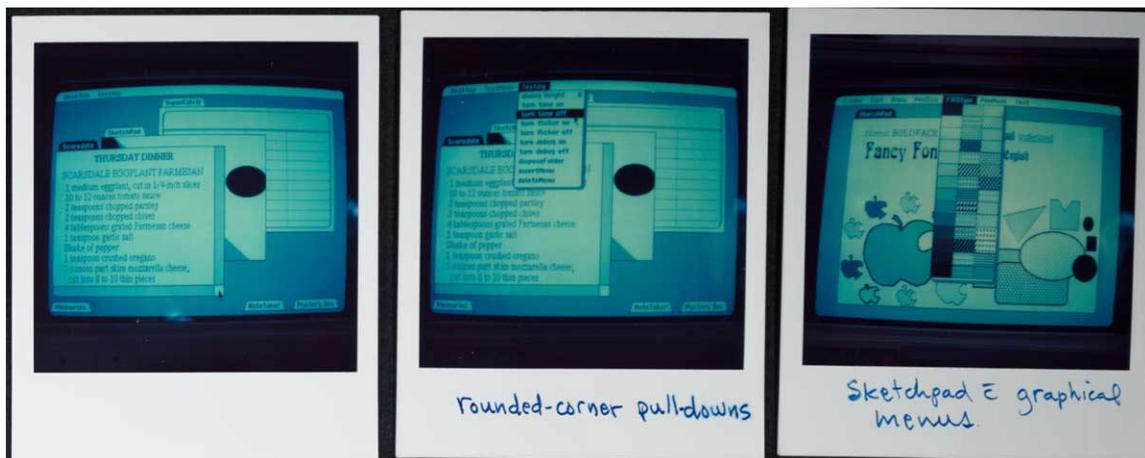
Now that the basic window structure was stabilizing, Bill turned his attention back to the graphics routines. He worked more on the Sketch program (the forerunner of MacPaint); the snowman drawing on the left is a clue that it's now Winter 1981. He added algorithmic text styles to the graphics, adding styles of bold (pictured on the right), as well as italic, outline and shadow (Bill took pictures of the other styles which I'm omitting to save space).



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



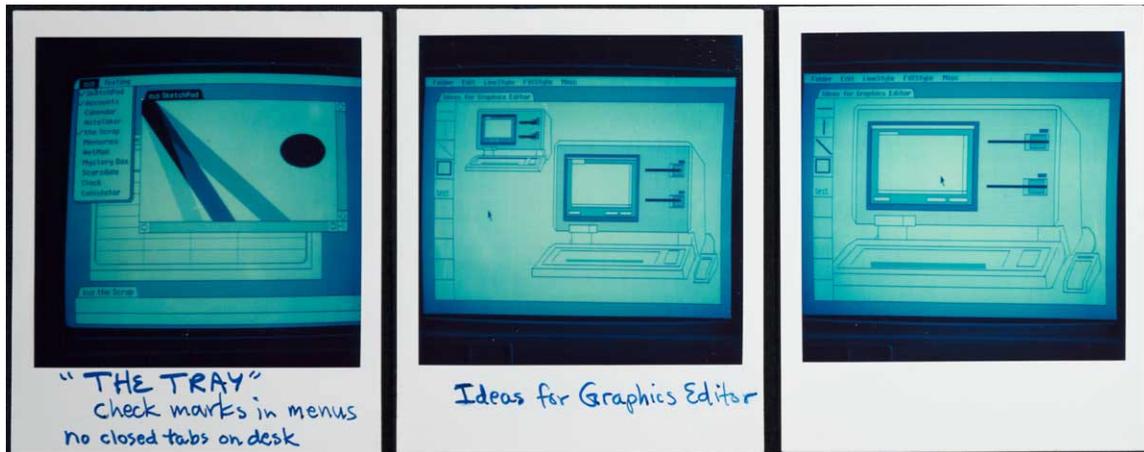
Bud Tribble was living at Bill's house now, and tended to sleep during the day and work all night, so Bill drew the phase diagram on the left with the sketch program. The middle picture shows fast ovals, which were added to LisaGraf as a basic type in Spring 1981, using a clever algorithm that didn't require multiplication. They were quickly followed by rectangles with rounded corners, or "roundrects", illustrated on the right, which were suggested by Steve Jobs (see Round Rects Are Everywhere!) .



By May 1981, the Lisa user interface is beginning to solidify. The leftmost photo shows scrollable documents of different types in overlapping windows, still sporting folder tabs for titles. The middle picture shows how roundrects began to creep into various UI elements, like menus, providing a more sophisticated look, especially when combined with drop shadows. The rightmost photo shows how menus could be also be graphical, as well as text based.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



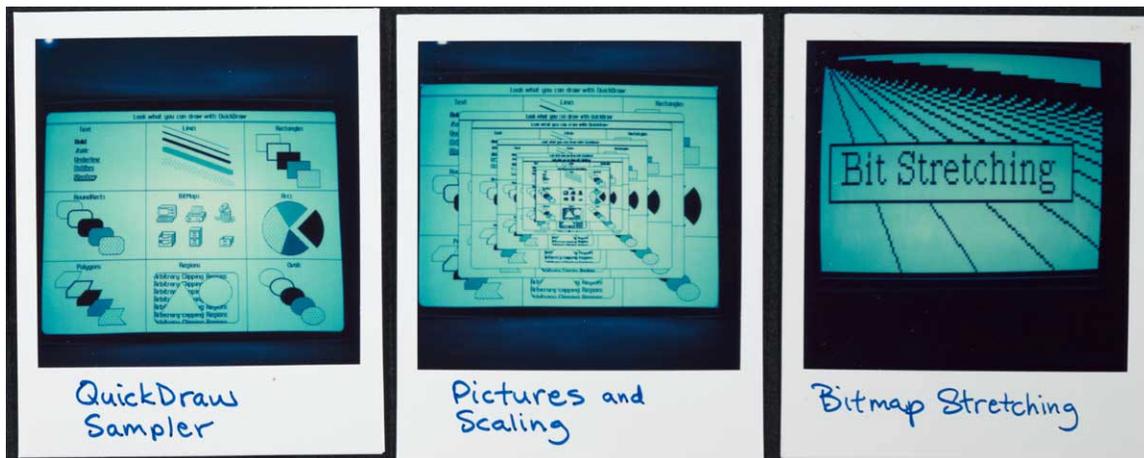
The Lisa team was worried about the closed window tabs being obscured by other windows on the desktop, so Bill added a standard menu on the extreme left called "the tray", that could show and hide opened windows. The middle and right pictures portray a prototype that Bill created for the Lisa Graphics Editor (which eventually evolved into MacDraw), to demonstrate that modes could sometimes be useful; it was the first program to select modes with a graphical palette, which eventually became the main user interface of MacPaint.



The last major change in the Lisa User Interface was moving to an icon-based file manager in March 1982. The leftmost picture was an early mock-up done in the graphics editor, using a two-level hierarchy; selecting an icon in the top pane displays its contents in the bottom one. By the middle photo, Bill arrived at something very similar to the shipping design, complete with a trash can at the lower right. (see Rosing's Rascals) . Note that the folder tab on windows has disappeared now, replaced by a rectangular title bar that's partially inverted when highlighted.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Finally, Bill renamed "LisaGraf" to "QuickDraw" in the spring of 1982, because he wanted a name that was suitable for the Macintosh, too. He added two related features to meet the burgeoning needs of the Lisa applications: pictures and scaling. Pictures were a way of recording graphics operations into a data structure for later playback; this became the basis of both our printing architecture and also cutting and pasting graphics. Since pictures could be drawn into an arbitrary sized rectangle, it also caused Bill to add bitmap scaling features as well.

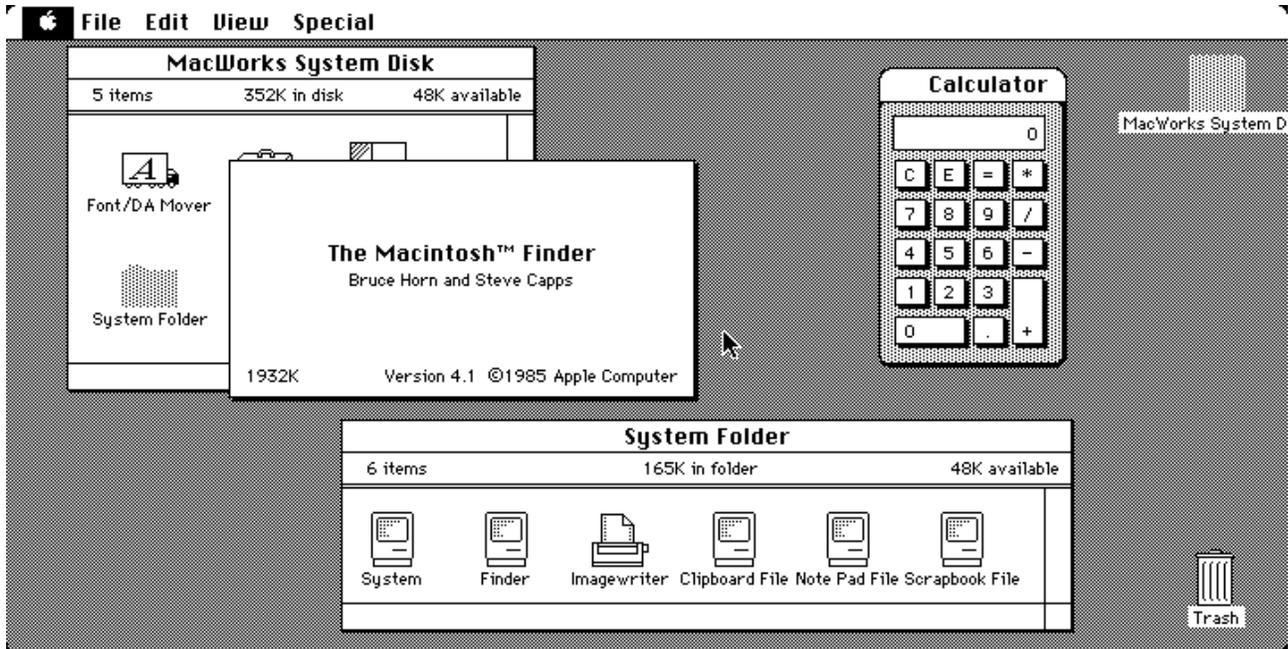
Most users and developers only experienced the user interface as a completed whole, so they tend to think of it as static and never changing, when in fact these pictures show that it was always evolving as we gained more experience and tackled more application areas. A user interface is never good enough, and, while consistency between applications is an important virtue, the best developers will continue to innovate when faced with new problems or perhaps just when they see a much better way to accomplish something. As usual, Bob Dylan said it best when he wrote in 1965, "He not busy being born, is busy dying."



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



[Macintosh Finder v4 screen shot -- 1985]

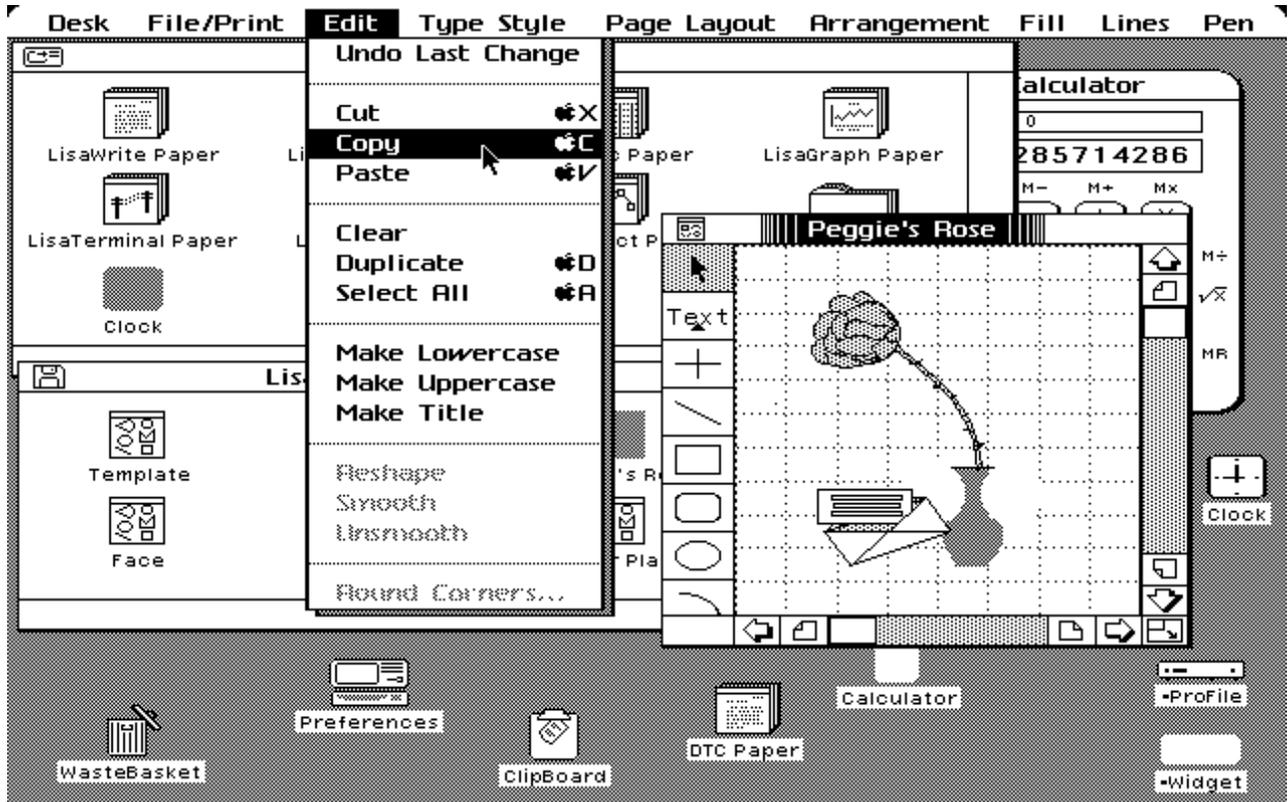




APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



[Lisa Desktop Manager v3 screen shot - 1984]





Origins of Spline-Based and Anti-Aliased Fonts

Author: William Donelson
Date: January 1989
Characters: William Donelson, John Warnock, Bob Sproull
Topics: Origins, Software Design
Summary: Creation of the first spline-based font system

While working on a spring sabbatical from high school in April and May of 1971, I worked in the computer department of the National Institutes of Health in Maryland.

I was assigned to work on the PDP-6 and PDP-10 with several people, including Dr Bob Sproull.

They had a Calcomp Film Plotter which used an electron beam to draw on film, thus allowing high resolution output, e. g. for publishing.

Bob asked me to investigate ways to make zoom-able fonts, which could also be distorted into a sort of italic, etc.

I studied the problem for a week or so, and concluded the best way to do this was a combination of straight lines and splines. I developed a list structure to store fonts. As examples, I did the letters "I" (of course) and "P" (just to prove the point).

This idea worked very well, and I gave a demo on the PDP-6's vector display system, set into raster mode. The characters could be scaled and slanted to "italic" quite easily, and could be shown as outlines or as filled characters.

(I also did a simple fighting video game on the display, "Germs" of course)

I then went off to MIT for ten years.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Fast forward to MacWorld, January 1989. I had taken a job at a Silicon Valley company doing Mac storage and video products, and was invited to a breakfast with John Warnock.

During breakfast, I mentioned to him that I was the originator of the spline-based font system. He looked shocked, and distinctly nervous! Apparently, he and Bob Sproull were associates, and he was afraid I was going to make claims or something!

I laughed, and put his mind to rest: "John, don't worry. I had an idea 18 years ago, and you and Adobe did all the real work". He laughed and relaxed a bit, but I'll never forget the expression on his face!



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



The Father Of The Macintosh

Author: Andy Hertzfeld
 Date: undated
 Characters: Bud Tribble, Steve Jobs, Joanna Hoffman, Jef Raskin, Rod Holt, Burrell Smith, Brian Howard, Steve Wozniak, Jerry Manock, Bill Atkinson
 Topics: Credit, Personality
 Summary: Who is the father of the Macintosh?

In the early days of the personal computer industry, breakthrough products could still be created single-handedly, or by very small teams. Steve Wozniak is indisputably the father of the Apple II, having designed the entire digital board himself as well as writing all of the system software, including a BASIC interpreter, most of it before Apple was even incorporated. But even Woz required help from Rod Holt for the analog electronics (the Apple II's switching power supply was almost as innovative as the digital board) and Steve Jobs and Jerry Manock for the industrial design (ditto for the plastic case).

By the 1980s, things had gotten more complicated. The Macintosh was more of a team effort, with at least a half dozen people making significant, invaluable contributions. For the launch publicity, Steve Jobs anointed seven of us (not counting himself) as the official "design team", but it could just as easily have been five or fifteen. Some people felt bad that they weren't included, and it was obvious that there was no good way to draw the line.

But if you look up the phrase "Father of the Macintosh" on Google, you get lots of links mentioning the initiator of the project, Jef Raskin. Jef was a former professor at UCSD (of both computer science and music) who started at Apple in January 1978 as Apple employee #31, after contracting to write Apple's manual for DOS with his friend Brian Howard, at their consulting firm named Bannister and Crun (a playful name appropriated from the Goon Show). Apple liked the DOS



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



manual so much that they hired Jef and Brian to be founders of their internal publications group.



Jef Raskin

In early 1979, after successfully building an outstanding pubs department, Jef turned the reins over to Phyllis Cole and started thinking about what it would take for personal computers to expand beyond the current hobbyist market, writing up his ideas in a series of short papers. He presented his idea for an ultra low cost, easy to use appliance computer to Mike Markkula in March 1979, and got the go-ahead to hire a few people and form an official research project later in September 1979, naming it Macintosh, after his favorite kind of eating apple. Most of his ideas for the new machine were collected in a set of papers he called "The Book of Macintosh".

There's no doubt that Jef was the creator of the Macintosh project at Apple, and that his articulate vision of an exceptionally easy to use, low cost, high volume appliance computer got the ball rolling, and remained near the heart of the project long after Jef left the company. He also deserves ample credit for putting together the extraordinary initial team that created the computer, recruiting former student Bill Atkinson to Apple and then hiring amazing individuals like Burrell Smith, Bud Tribble, Joanna Hoffman and Brian Howard for the Macintosh team. But there is also no escaping the fact that the Macintosh that we know and love is very different than the computer that Jef wanted to build, so much so that he is much more like an eccentric great uncle than the Macintosh's father.

Jef did not want to incorporate what became the two most definitive aspects of Macintosh technology - the Motorola 68000 microprocessor and the mouse pointing device. Jef preferred the 6809, a cheaper but weaker processor which only had 16 bits of address space and would have been obsolete in just a year or two, since it couldn't address more than 64Kbytes. He was dead set against the mouse as well, preferring dedicated meta-keys called "leap keys" to do the pointing. He became increasingly alienated from the team, eventually leaving entirely in the summer of 1981, when we were still just getting started, and the final product utilized very few of the ideas in the Book of Macintosh. In fact, if the name of the project had changed after Steve took over in January 1981, and it almost



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



did (see Bicycle) , there wouldn't be much reason to correlate it with his ideas at all.

So, if not Jef, does anyone else qualify as a parent of the Macintosh? Bill Atkinson is a strong candidate, since he was almost singlehandedly responsible for the breakthrough user interface, graphics software and killer application that distinguished the Mac. A case could also be made for Burrell Smith, whose wildly creative digital board was the seed crystal of brilliance that everything else coalesced around. But ultimately, if any single individual deserves the honor, I would have to cast my vote for the obvious choice, Steve Jobs, because the Macintosh never would have happened without him, in anything like the form it did. Other individuals are responsible for the actual creative work, but Steve's vision, passion for excellence and sheer strength of will, not to mention his awesome powers of persuasion, drove the team to meet or exceed the impossible standards that we set for ourselves. Steve already gets a lot of credit for being the driving force behind the Macintosh, but in my opinion, it's very well deserved.



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



An Early 'Switch' Campaign

Author: Tom Zito
 Date: September 1985
 Characters: Bo Derek, Steve Jobs
 Topics: Personality
 Summary: Steve Jobs tries to get Bo Derek to abandon her PC in favor of a Mac



Bo Derek, PC User

My former wife, Laura Bachko, had been a book editor in New York and had worked on a project with Bo Derek before we moved to California. One night when Laura and I were having dinner with Steve, Bo's name came up -- as did the factoid



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



that Bo was a heavy duty computer jockey, albeit of the IBM persuasion. Steve took this as a personal challenge; he was going convert Bo to a Mac user -- and who knew what else. Clearly the computer could be a foot in the door. He persuaded Laura to make him an appointment with Bo.

And so one day shortly thereafter, Steve piled into his Mercedes, along with a Mac, and drove down to Bo's Santa Barbara ranch, which she shared with her husband John Derek. Bo was cordial but unimpressed; she accepted the computer but remained a PC user. And apparently she did not find Steve as dashing as Steve expected she would.

Several weeks later, Steve was complaining to Laura about the lackluster impression he had made. "Look," she told him. "She's married. And besides, I don't know any woman who would want her name to be Bo Jobs."



APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Image Gallery



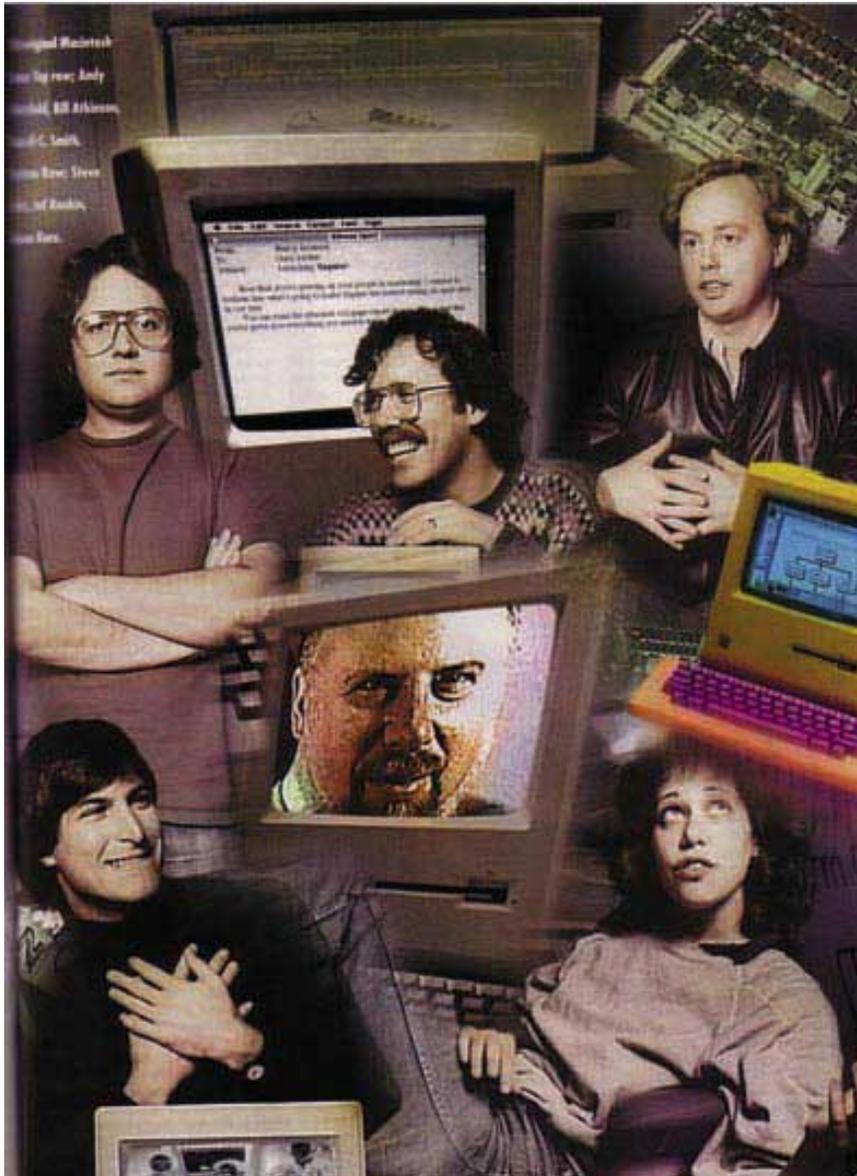


APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



The following collage is from an article by Steven Levy that appeared in the February 1994 issue of Popular Science, commemorating the Macintosh's tenth birthday. It's based on pictures taken by Norman Seiff for Rolling Stone Magazine in January 1984.

Featuring Andy Hertzfeld, Bill Atkinson, Burrell Smith, Steve Jobs and Susan Kare, with Jef Raskin in the center.





APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Here's the software team, photographed for Rolling Stone in January 1984, in the aftermath of a failed human pyramid.

From left to right, there's Jerome Coonen, Donn Denman, Andy Hertzfeld, Rony Sebok, Bruce Horn, Susan Kare, Bill Atkinson, Owen Densmore, Steve Capps, Larry Kenyon, Patti Kenyon (holding little Tracy Kenyon)





APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



A picture of the official Mac Design Team taken by Norman Seiff for Rolling Stone, in the lobby of Bandley 3. We were told to look serious for this outtake, but they ended up using a happier one in the article.

From left to right, it's George Crow, Joanna Hoffman, Burrell Smith, Andy Hertzfeld, Bill Atkinson and Jerry Manock.





APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



This is a picture of the entire Macintosh team that worked in Bandley 3, in front of the building. It was taken in December 1983 for the initial issue of MacWorld magazine. Note Larry Kenyon sitting at the front right, holding his newly born daughter Tracy, who's only a few weeks old.





APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Some Mac team members immediately following the Mac introduction at the shareholder's meeting at Flint Center. If you look closely, you can see Steve Jobs holding a bottle of water that he's about to pour on someone's head. Standing, left to right, there's Rony Sebok, Burrell Smith, Steve Jobs, Joanna Hoffman and Hasmig Seropian; kneeling is Bill Atkinson and Andy Hertzfeld.

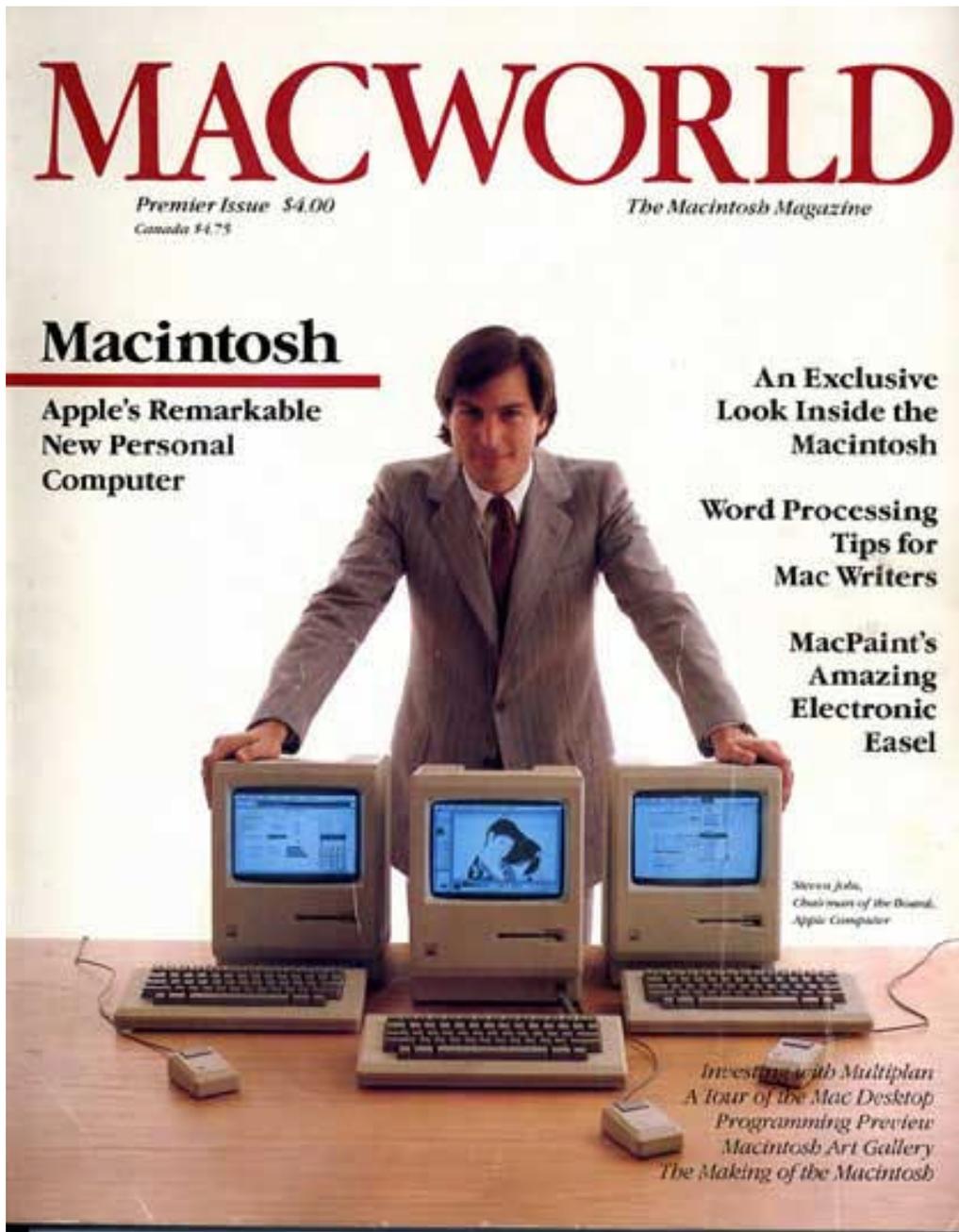




APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Here's the cover of the first issue of MacWorld magazine, featuring Steve Jobs, which was handed out at the end of the shareholder's meeting where the Mac was announced.





APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Some members of the Mac team photographed in 1989, to celebrate the 5th anniversary of the Macintosh. Everyone is folding their hands in tribute to a favorite Steve Jobs gesture.

The back row on the stairs is Patti Kenyon, Bud Tribble, Brian Howard and Steve Capps. The front row is Joanna Hoffman and Andy Hertzfeld. On the right are Larry Kenyon and Mike Boich.





APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



This photo of Steve Jobs and Bill Atkinson, which was taken by Norman Seiff in January 1984, captures something about their close relationship.





APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



This faded, scratched polaroid depicts the Mac software team in December 1983, during the final push, all dressed up in white shirts and shorts in homage to Steve Capps' sartorial stylings.

Top row: Donn Denman, Larry Kenyon. Middle row: Jerome Coonen, Hans Gerke (visiting translator), Andy Hertzfeld, a very pregnant Patti Kenyon, Burrell Smith (honorary member of the software team). Bottom row: Randy Wiggington, Rony Sebok, Susan Kare, Donatien (another translator).

Capps and Bruce Horn are missing, since this was taken right before we surprised him. Thanks to Susan Kare for this rare photograph.





APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Here are the badges of a bunch of Apple employees, that were photographed for an ad (that was never run) in the summer of 1983 by Chi at Day.

You can see Burrell and Andy's badges, plus some executives including John Sculley and Ann Bowers, the Apple Human Resources VP.

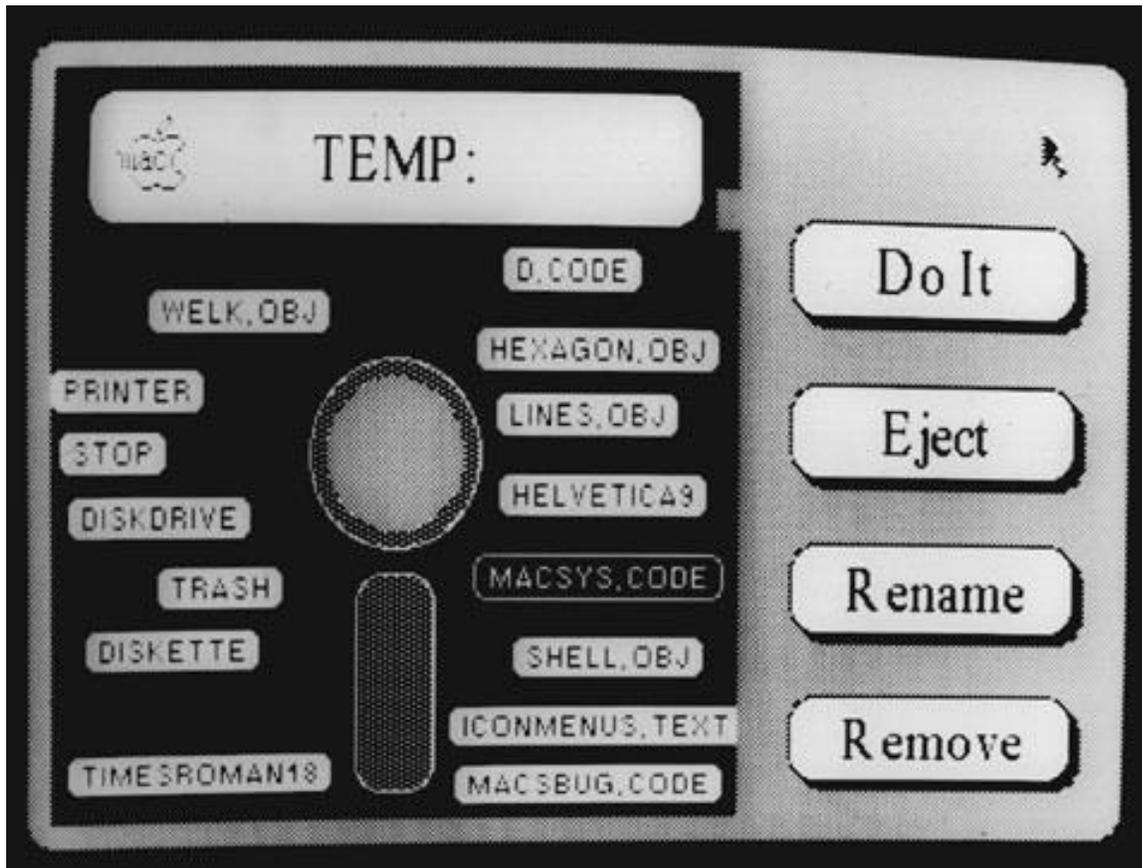




APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Here is a photo of one of two early prototypes that Bruce Horn created for the Finder in spring 1982. As you can see, it looks quite different from the icon based Finder that we ended up with, but it helped us explore ideas about special filing.





APPLE MACINTOSH COMPUTER HISTORICAL INFORMATION



Jimell d. Azama
 Gerald C. Manock
 Daniel Kottke
 Bruce Carl
 Pamela J. Wyman
 Bill Atkinson
 Vicki Milledge
 Bruce Horn
 George Crow
 Rod Holt
 Andy Hertzfeld
 Angelia Lo
 Joanna Kasprzak
 HAP HORN
 M.E. McCammon
 Burrell Smith
 Jef Raskin
 Guy Stewart
 Roger Alejo
 Patricia Sharp
 Steven Jobs
 Brian Robertson
 Joan Marsh
 Larry Kenyon
 Martin P. Hales
 Collette Askeland
 Carlos J. Fidel
 Christopher
 Mike Boich
 Lynn Sakahachi
 Benjamin L. ...
 Bill Fernandez
 Donn Denman
 Patti King
 David H. Reed
 Ronald W. Nicholson
 Matt Carter
 Robert L. Bellville
 Randy Wigginton
 Linda Wilkins
 Michael Murray

FEBRUARY 10, 1982



THE END

