

A spiral-bound notebook with a light brown, textured cover. The spiral binding is on the left side. The text is centered on the cover.

Jonathan Hoyle

Interview at the MathWorks

1/10/05

# Overview

---

- ✓ Educational Background
- ✓ Professional History
  - Early Career
  - Eastman Kodak
  - Gene Codes
- ✓ Involvement with Apple
- ✓ Skills & Achievements

# Educational Background

---

## ✓ University of Delaware

- Bachelor of Science, 1986
- Major: Mathematics
- 1st Minor: Computer Science
- 2nd Minor: Philosophy

## ✓ University of Michigan

- Graduate Studies, 1986-1988
- Major: Mathematics
- Minor: Computer Engineering

# Early Career

---

## ✓ Individual Consulting (1989-1994)

- Macintosh installations
- Mac software port to Windows 3.0
- Development on *Turbo Pascal for Windows*
- Released freeware Poker game for Mac OS

## ✓ DuPont Core Technology (1993-1994)

- Development of *HyperColor* software
- 68K port to PowerPC
- Development on *MPW* and *CodeWarrior*

# Eastman Kodak (1994-2001)

---

- ✓ Kodak's Digital Science Drivers Group
- ✓ CMM Level 2 efforts:
  - Documentation and source control
  - Acceptance & Unit tests
  - Code Reviews and Low Level Designs
- ✓ Weekly Tech meetings with presentations
- ✓ Responsible for
  - Designing objects for DGFramework
  - Universal Calibration

# DGFramework

---

- ✓ Cross platform C++ framework
- ✓ Responsible for several classes:
  - DGCStr (a detailed Unicode string class)
  - DGList<> (a templated sparse array)
  - DGPtr<> (a templated smart pointer)
  - several others
- ✓ PowerPoint presentations given at the weekly DGTech meetings

# Universal Calibration

---

- ✓ Primarily responsible for developing Universal Calibration for Kodak's printers
- ✓ Written in Metrowerks CodeWarrior C++ for Mac OS and Visual C++ for Windows
- ✓ Printer-specific data house in plug-ins
- ✓ Available for a range of thermal printers:
  - Kodak DS XLS 8600/8650/8670
  - Kodak Polychrome DCP 9000/9300
  - Kodak 4700

# Kodak Calibration Utility

Kodak Professional 8670 thermal printer

Kodak Professional 8670 thermal printer

General Media Color Management Calibration

Densitometry Input

Square	K	C	M	Y	
1					↑
2					
3					
4					
5					
6					
7					
8					
9					↓

Densitometer Device

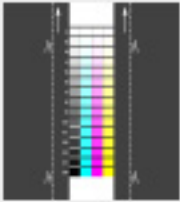
Model: X-Rite™ Digital Swatchbook

Type: Status T

Calibration Setup

Patches: 15

Algorithm: Default



Help Cancel Recompute



# Gene Codes (2001-present)

---

## ✓ Sequencher

- Overview
- Development

## ✓ World Trade Center Project (*M-FISys*)

- Timeline
- Project Philosophies
- Kinship Analysis

# *Sequencher* Overview

---

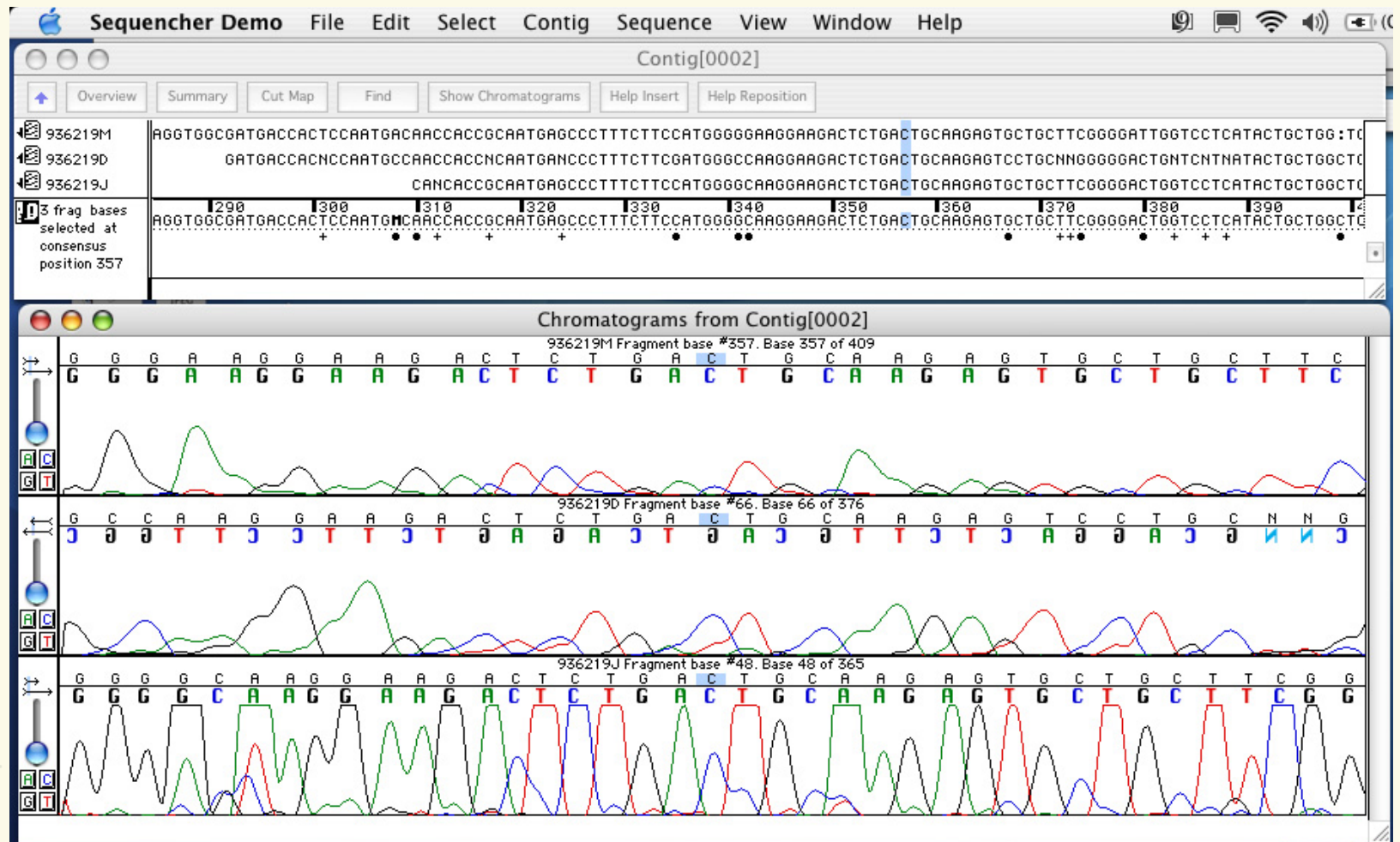
- ✓ Leading Bioinformatics software package
- ✓ DNA sequencing & analysis
- ✓ Used by:
  - Cancer/Aids research
  - Medical and drug companies
  - Universities
- ✓ Available for both Macintosh and Windows
- ✓ Specialized Forensic version for FBI, etc.

# *Sequencher* Development

---

- ✓ CodeWarrior C++ both Mac & Windows
- ✓ Written in THINK Class Library framework
- ✓ Single cross-platform code base
- ✓ Primarily responsible for replacing Classic code with Carbonized API's
- ✓ Carbonized version of Sequencher was release as version 4.2 in early 2004
- ✓ Updated version for Windows also released

# Sequencher DNA analysis



# World Trade Center Project

---

- ✓ On September 11, 2001, 2,752 people were killed in the attacks on the World Trade Center.
- ✓ Over 20,000 remains were recovered, the vast majority of which would require DNA matching for identification.
- ✓ Existing software tools for DNA identification (such as CoDIS) proved wholly inadequate for the scope and magnitude of this project.
- ✓ The entire company devoted its resources to creating the identification software need by NYC.

# 2001 *World Trade Center* Timeline

---

- ✓ **September 17:** Armed Forces DNA Identification Lab [AFDIL] asks Gene Codes to update *Sequencher* for the Pentagon and Shanksville crashes.
- ✓ **September 28:** Office of the Chief Medical Examiner [OCME] in New York City contacts us for new software.
- ✓ **October 15:** Using the *Extreme Programming* [XP] methodology, software development is underway.
- ✓ **December 13:** *M-FISys* (Mass-Fatality Identification System) has its first release to the OCME.
- ✓ **Since:** Weekly releases personally delivered to the OCME, to accommodate rapidly changing requirements.

# *M-FISys* Project Philosophies

---

## ✓ Test First

- New functionality must have Unit Tests written first
- Once the test is in place, write code to make it pass
- Single integration machine to submit and run tests

## ✓ Paired Programming

- Two eyes are better than one

## ✓ Thin Design

- The simplest thing that can possibly work
- Design to allow future refactoring

## ✓ Small Releases

- Fully functional releases every week or two

# M-FISys Forensic Software

M-FISys 4.0.2-Master List-Default

Locate ID Samples: 12 Identified Aggregates: 1 Identifiable Aggregates: 1 Unidentified Aggregates: 2

ID	RM	Likelihood	I	# M Sn	Gen	D3S1358	vWA	FGA	D8S1179	D21S11	D18S51	D5S818	D13S317	D7S820	D16S539	TH01	TPOX	CSF1PO	Penta D	Penta
RM# 5913 (6)		2.8E+017	0		XY	14/17	16/17	23/25	13	30	13/14	11	8/11	11/13	7/9	8/11	11/13	neg	-	-
RM# 4141 (4)		6.2E+020	1		XX	14/17	17	23/24	15/17	28/32	13/15	12	10/11	12	12/13	9	8/11	10/13	11/12	12/14
SP-90003-1	4141	6.2E+020		16	XX	14/17	17	23/24	15/17	28/32	13/15	12	10/11	12	12/13	9	8/11	10/13	11/12	12/14
DM0193281	4141	6.2E+020		16	XX	14/17	17	23/24	15/17	28/32	13/15	12	10/11	12	12/13	9	8/11	10/13	11/12	12/14
DM0196708	4141	6.2E+020		16	XX	14/17	17	23/24	15/17	28/32	13/15	12	10/11	12	12/13	9	8/11	10/13	11/12	12/14
VRT-DM0180017	4141	6.2E+020	I	16	XX	14/17	17	23/24	15/17	28/32	13/15	12	10/11	12	12/13	9	8/11	10/13	11/12	12/14
DM0180018				16	XX	14/17	17	23/24	15/17	28/32	13/15	12	10/11	12	12/13	9	8/11	10/13	11/12	12/14
OXO1-DM0180018	4141	6.2E+020	I	16	XX	14/17	17	23/24	15/17	28/32	13/15	12	10/11	12	12/13	9	8/11	10/13	11/12	12/14
RM# 9112 (3)		1.6E+018	0		XX	13/16	16/17	20/22	13/15	28/29	12	12/13	9/10	11/12	8/11	6/9	10	10/12	-	-
SP-80007-1	9112	1.9E+019		12	XX	13/16	16	20/22	13/15	28/29	12	12/13	9/10	11/12	8	6/9	10	10/12	-	-
SP-80007-2	9112	1.9E+019		12	XX	13/16	16	20/22	13/15	28/29	12	12/13	9/10	11/12	8	6/9	10	10/12	-	-
DM0190020		1.6E+018		14	XX	13/16	16/17	20/22	13/15	28/29	12	12/13	9/10	11/12	8/11	6/9	10	10/12	-	-
AS040004 (4)		4.9E+015	0		XY	16/17	17	20/21	10/11	30/31	15/16	10/12	8/11	11	11/13	7/9	8	8/11	-	-
DM0190066		4.9E+015		14	XY	16/17	17	20/21	10/11	30/31	15/16	10/12	8/11	11	11/13	7/9	8	8/11	-	-
DM0190221		4.9E+015		14	XY	16/17	17	20/21	10/11	30/31	15/16	10/12	8/11	11	11/13	7/9	8	8/11	-	-
DM0190881		4.9E+015		14	XY	16/17	17	20/21	10/11	30/31	15/16	10/12	8/11	11	11/13	7/9	8	8/11	-	-
DM0190880		4.9E+015		14	XY	16/17	17	20/21	10/11	30/31	15/16	10/12	8/11	11	11/13	7/9	8	8/11	-	-

RM# 4141: Simpson, Marge (Chain 1)  
Method: Test

Expand All Collapse All  Hide Identical Alleles Exclude... Merge Export  Hide Names Print Options...

STR mtDNA SNP Jobs



# Kinship Analysis

---

- ✓ Not all identifications could be made based upon personal effects
- ✓ Cheek swabs of the family members were taken to create a DNA pedigree
- ✓ Through the mathematics of Kinship Analysis, remains can be identified through a victim's family members
- ✓ Although *M-FISys* was done in XP, I was the sole author of the Kinship prototype

# Kinship Prototype on Mac OS X

**Kinship Match**

SAMPLES Victim: 17868 Kin: 6977 PE's: 8643 Valid Pedigrees: 2294 Pedigree Status: **Consistent**

RM Number #   Identified Inconsistent PE's: None

	BODE-DM8112148	S2-80999-12	<input checked="" type="checkbox"/> BF-80887 #13	<input checked="" type="checkbox"/> BM-80887 #11	PR-80887 #01
Gen	XY	XY	XY	XX	XY
D3S1358	14/18	14/15	14/15	15/18	15/18
vWA	15/16	16/19	16	15/19	15/16
FGA	20/22	20/22	20	22	22/25
D8S1179	13	13	13	11/13	11/13
D21S11	28/31	28/30	28/31	28/30	28/31.2
D18S51	12/17	12/17	12/16	12/17	12/14
D5S818	8/11	10/11	11/12	8/10	8/12
D13S317	8/9	8/9	9/12	8/11	10/11
D7S820	11/13	10/11	9/11	10/13	7/11
D16S539	9	9	9	9/12	12/13
TH01	6/8	8	6/8	6/8	7/8
TPOX	8/11	11	8/11	8/11	8
CSF1P0	10/11	10/13	10/11	11/13	11
Penta D	-	-	-	-	-
Penta E	-	-	-	-	-
Likelihood	1.58e+18	2.9e+17	1.03e+16	1.48e+18	3.54e+17
Kinship LR	>99.999999%	99.999999%	6.72e+6	1.54e+4	
w/ SNP's	>99.999999%		6.72e+6	1.54e+4	

STR  SNPs Victim PE # 1/1

Lookup Sample... Show Equations Window... Maximum Allowable Mismatches:   Hide Names

Likelihood Threshold: 10^  Match by:   Allow STR Allelic Dropout

# Relationship with Apple

---

- ✓ Continued close relations with Apple's Developer Relations community
- ✓ Attended 8 of the last 9 Apple Worldwide Development Conferences
- ✓ Other Recent Technical Meetings:
  - PowerMac G5 Optimization Course
  - Mac OS X 10.4 Tiger Talks

# Skills & Achievements

---

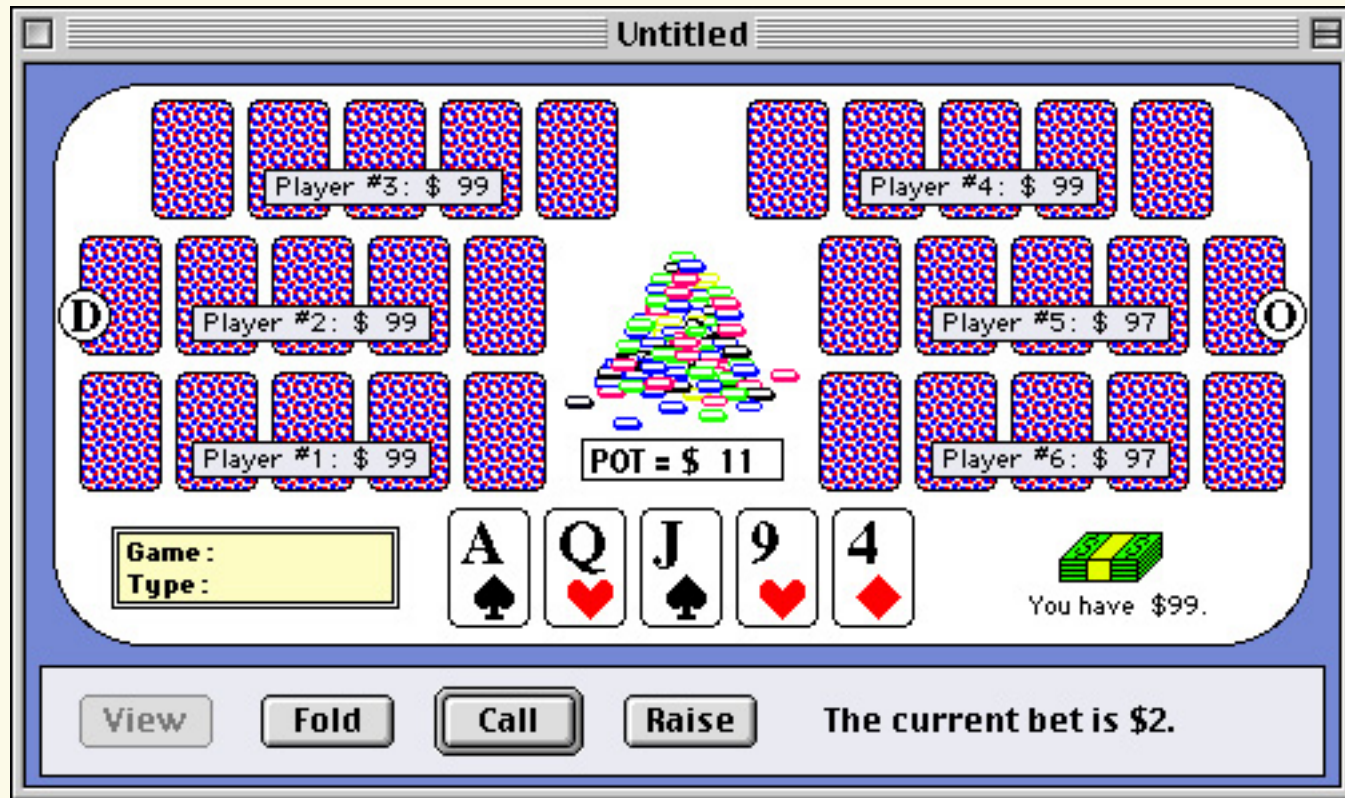
- ✓ *“Development Under Extreme Conditions”*
  - co-authored with company CEO & head of QA
  - presented at 2003 Symposium on BioComputing
- ✓ Presentation of Forensic Mathematical Analysis of World Trade Center attacks
  - presented at 2003 Academy of Forensic Sciences
- ✓ Winner of the *Most Functional Output* award of the 2004 International Obfuscated C Code Contest

# Development Environments

---

- ✓ Strong knowledge in C/C++
- ✓ Mac OS development using the Classic and later Carbon API' s
- ✓ Proficiency in *Metrowerks CodeWarrior*
- ✓ Proficiency in *RealBasic* RAD environment
- ✓ Experience with *Microsoft C#.NET*
- ✓ Some experience with Java and connectivity with C++ via JNI and RMI

# Freeware Poker Game (1993)



AOL Top 10 Download March 1993

# IOCCC 2004 *Most Functional Output*

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define _ ;double
#define void x,x
#define case(break,default) break[0]:default[0]:
#define switch(bool) ;for(;x<bool;
#define do(if,else) inline(else)>int##if?
#define true (--void++)
#define false (++void--)

char*0=" <60>!?\\";
double[0]_ int0,int1 _ long=0 _ inline(int else){int
010=!0 _ l=!0;for(;010<010;++010)l+=(010[double]*pow(else,010));return l;}int
main(int bool,char*else[]){int l=1,x=-*0;if(else){for(;l<010+1;l++)l[double-1]
=bool>l?atof(l[else]):!0 switch(*0)x++)abs(inline(x))>long&&(long=abs(inline(x)
));int1=long;main(-*0>>1,0);}else{if(bool<*0>>1){int0=int1;int1=int0-2*long/0
[0]switch(5[0]))putchar(x-*0?(int0>=inline(x)&&do(1,x)do(0,true)do(0,false)
case(2,1)do(1,true)do(0,false)6[0]case(-3,6)do(0,false)6[0]-3[0]:do(1,false)
case(5,4)x?bool?0:6[0]:7[0])+*0:8[0]),x++;main(++bool,0);}}}
```

ANSI compliant and compiles with no warnings!

A spiral-bound notebook with a textured, light brown cover. The spiral binding is on the left side. The word "Thanks!" is printed in a large, dark brown, serif font in the center of the cover.

Thanks!