



CMP

United Business Media

▶▶ PRODUCT REVIEWS ENDORPHIN 2 ▶▶ CLAYTOOLS SYSTEM

AUGUST 2005

gamedeveloper

THE LEADING GAME INDUSTRY MAGAZINE

▶▶ **SID WILL PROVIDE**
BUILDING MODDABILITY
INTO CIVILIZATION IV

▶▶ **BUSINESS LEVEL**
GROWING PANDEMIC
STUDIOS SMOOTHLY

▶▶ **HDR FOR OPEN GL**
ENLIGHTENED ME
WITH SCIENCE

POSTMORTEM:
DOUBLE FINE'S
PSYCHONAUTIC
BREAK





So real it renders fear.

Idea:

Create the most gripping and realistic stealth action game on the market.

Realized:

With *Tom Clancy's Splinter Cell® Chaos Theory™*, Ubisoft™ wanted to continue setting records for the visual gaming experience. That's why they chose Autodesk's 3ds Max software to model and animate the game's realistic characters and backgrounds. By providing a highly creative and stable platform, 3ds Max helped Ubisoft artists continue their impressive work on Tom Clancy's Splinter Cell series that has already sold 10 million copies worldwide. 3ds Max's robust, work-horse capabilities also helped Ubisoft stay on target with their grueling production schedules. As a result, Ubisoft met their highly anticipated launch and garnered a 9.9 out of 10 by *Official Xbox Magazine* because of the game's amazing looks and lifelike play. From taking out the competition to taking out the warehouse guard, Autodesk software helps today's top developers realize their ideas to compete and win. To learn more, please visit autodesk.com/3dsmax



Autodesk and 3ds Max are registered trademarks of Autodesk, Inc., in the USA and/or other countries. All other brand names, product names, or trademarks belong to their respective holders. © 2005 Autodesk, Inc. All rights reserved.



Tom Clancy's Splinter Cell® Chaos Theory,™ image courtesy of Ubisoft.

Autodesk®

FLY THROUGH YOUR CREATION



Meet SpacePilot™

So much power right at your fingertips

Navigation is a breeze- SpacePilot lets you pan, zoom and rotate in one efficient, fluid movement. Plus its customizable hot keys guarantee that modeling and animating commands are just a click away.

And since you're sharing the work across two hands, you're reducing physical stress- 50% of your mouse movement vanishes into thin air. Grab hold of your 3D world with SpacePilot and let your creativity soar!

Special offer for you at www.3dconnexion.com/fly



 **3Dconnexion**
A LOGITECH COMPANY



POSTMORTEM

30 HEAD GAMES: DOUBLE FINE'S PSYCHONAUTIC BREAK

PSYCHONAUTS, the warped action platform title which takes place entirely in a character's head, is the first effort from Tim Schafer's San Francisco-based Double Fine Productions, and went through two publishers and four years of cinched belts before seeing the light of day. What holds a development team together when the game they're making is literally straining toward psychosis?

By Caroline Esmurdac

FEATURES

15 ACT OF MOD: BUILDING SID MEIER'S CIVILIZATION IV FOR CUSTOMIZATION

It's a mod mod world—and Sid Meier's latest game was built to accommodate it; CIVILIZATION IV was designed so that players could customize nearly every bit of it. One programmer from Firaxis Games explains how such a project was planned for and pulled off.

By Mustafa Thamer

21 BUSINESS LEVEL: GROWING PANDEMIC

In this special, expanded Business Level column, an executive from Pandemic Studios, makers of DESTROY ALL HUMANS! and FULL SPECTRUM WARRIOR, shares the company vision: grow, but never at the expense of your employees.

By Josh Resnick

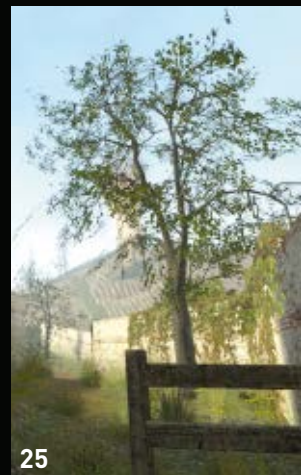
25 HDR-EVOLUTION WITH FRAME BUFFER OBJECTS FOR OPENGL

High dynamic range (HDR) rendering is a technique used to retain the color precision of a rendered scene as it goes through the rendering pipeline, recently in the news due to Valve's HDR-utilizing HALF-LIFE 2: LOST COAST expansion. This technical article talks about ways to use frame buffer objects for off-screen rendering to help facilitate HDR effects, and gives practical examples of how to do so in terms of OpenGL code.

By Allen Sherrod



15



25

DEPARTMENTS

- 4 GAME PLAN *By Simon Carless*
Psycho Kipper
- 6 HEADS UP DISPLAY
Siggraph preview, IGDA makeover, 3DS Max 8 announced, and more
- 9 SKUNK WORKS *By James Alguire and Justin Lloyd*
SensAble Technologies' ClayTools System and NaturalMotion's endorphin 2.0
- 64 A THOUSAND WORDS
Sony's SHADOW OF THE COLOSSUS

COLUMNS

- 39 PIXEL PUSHER *By Steve Theodore* [ART]
Dry Eyes: Building Eyes for Animation
- 43 INNER PRODUCT *By Sean Barrett* [PROGRAMMING]
Scanning Large Directories
- 46 GAME SHUI *By Noah Falstein* [DESIGN]
Right to Bare Arms
- 47 AURAL FIXATION *By Alexander Brandon* [SOUND]
Fun Audio Design

EDITORIAL

EDITOR

Simon Carless scarless@gdmag.com

MANAGING EDITOR

Jill Duffy jduffy@gdmag.com

ASSISTANT EDITOR

Brandon Sheffield bsheffield@gdmag.com

ART DIRECTOR

Cliff Scorsio cscorso@gdmag.com

INTERNS

Alan Bank, editorial & Rebecca Nickodemus, design

CONTRIBUTING EDITORS

Sean Barrett sbarrett@gdmag.com
 Alexander Brandon abrandon@gdmag.com
 Noah Falstein nfalstein@gdmag.com
 Steve Theodore stheodore@gdmag.com

ADVISORY BOARD

Hai Barwood	Designer-at-Large	Dave Pottinger	Ensemble Studios
Ellen Guon Beeman	Monolith	George Sanger	Big Fat Inc.
Andy Gavin	Naughty Dog	Harvey Smith	Midway
Joby Otero	Luxoflux	Paul Steed	Microsoft

ADVERTISING SALES

NATIONAL SALES MANAGER

Afton Thatcher athatcher@cmp.com • 415.947.6217

SENIOR ACCOUNT MANAGER, EAST COAST, EUROPE & EASTERN CANADA

Ayrien Machiran amachiran@cmp.com • 415.947.6224

ACCOUNT MANAGER, SO. CALIF., SOUTH WEST, CONTRACTORS & MARKETPLACE

Susan Kirby skirby@cmp.com • 415.947.6226

ACCOUNT MANAGER, NO. CALIF., NORTH WEST, ASIA & WESTERN CANADA

Nick Geist ngeist@cmp.com • 415.947.6223

ACCOUNT MANAGER, GLOBAL RECRUITMENT/EDUCATION & TEXAS

Aaron Murawski amurawski@cmp.com • 415.947.6227

ADVERTISING PRODUCTION

ADVERTISING PRODUCTION COORDINATOR Kevin Chanel

REPRINTS Julie Rapp jarapp@cmp.com • 510.834.4752

GAME GROUP MARKETING

DIRECTOR OF BUSINESS STRATEGY Michele Maguire

DIRECTOR OF MARKETING Tara C. Gibb

MARKETING MANAGER Kathleen Cheng

CMP GAME GROUP

VP, GROUP PUBLISHER APPLIED TECHNOLOGIES Philip Chapnick

CONFERENCE DIRECTOR, GDC Jamil Moleldina

ASSOCIATE CONFERENCE DIRECTOR, GDC Susan Marshall

EXECUTIVE WEB PRODUCER Peter Leahy

EDITOR, GAMASUTRA.COM Simon Carless

FEATURES EDITOR, GAMASUTRA.COM Quang Hong

CIRCULATION

CIRCULATION DIRECTOR Kevin Regan kregan@cmp.com

CIRCULATION ASSISTANT MANAGER Jessica Ward jward@cmp.com

CIRCULATION COORDINATOR Miguel Mendiola mmendiola@cmp.com

CIRCULATION ASSISTANT Michael Campbell mcampbell@cmp.com

CIRCULATION ASSISTANT Adrea Abidor aabidor@cmp.com

SUBSCRIPTION SERVICES

FOR INFORMATION, ORDER QUESTIONS, AND ADDRESS CHANGES

• 800.250.2429 • 847.763.9606 • gamedeveloper@halldata.com

INTERNATIONAL LICENSING INFORMATION

Mario Salinas

msalinas@cmp.com • 650.513.4234 • 650.513.4482

CMP MEDIA MANAGEMENT

PRESIDENT & CEO Gary Marshall

EXECUTIVE VICE PRESIDENT & CFO John Day

EXECUTIVE VICE PRESIDENT & COO Steve Weitzner

EXECUTIVE VICE PRESIDENT, CORPORATE SALES & MARKETING Jeff Patterson

EXECUTIVE VICE PRESIDENT, HUMAN RESOURCES Leah Landro

CHIEF INFORMATION OFFICER Mike Mikos

SENIOR VICE PRESIDENT, OPERATIONS Bill Amstutz

SENIOR VP & GENERAL COUNSEL Sandra Grayson

SENIOR VP, COMMUNICATIONS Alexandra Raine

SENIOR VP, CORPORATE MARKETING Kate Spellman

VP/GROUP DIRECTOR INTERNET BUSINESS Mike Azzara

PRESIDENT, CHANNEL GROUP Robert Faletra

PRESIDENT, CMP HEALTHCARE MEDIA Vicki Masseria

VP, GROUP PUBLISHER INFORMATION WEEK

MEDIA NETWORK Michael Friedenbeck

VP, GROUP PUBLISHER ELECTRONICS Paul Miller

VP, GROUP PUBLISHER NETWORK COMPUTING

ENTERPRISE ARCHITECTURE GROUP Fritz Nelson

VP, GROUP PUBLISHER SOFTWARE DEVELOPMENT MEDIA Peter Westerman

VP, DIRECTOR OF CUSTOM INTEGRATED MARKETING SOLUTIONS Joseph Braue

CORPORATE DIRECTOR, AUDIENCE DEVELOPMENT Shannon Aronson

CORPORATE DIRECTOR, AUDIENCE DEVELOPMENT Michael Zane

CORPORATE DIRECTOR, PUBLISHING SERVICES Marie Myers



PSYCHO KIPPER

QU'EST-CE QUE C'EST? THIS ISSUE OF *GAME DEVELOPER* COINCIDES WITH THE YEARLY DIGITAL GRAPHICS FEST THAT IS SIGGRAPH 2005, AND SOME OF YOU MAY BE LUCKY ENOUGH TO PICK UP YOUR FIRST COPY OF THE MAG THERE. IF SO, PLEASE TELL KEYNOTE SPEAKER GEORGE LUCAS THAT WE'RE EAGERLY AWAITING KNIGHTS OF THE OLD REPUBLIC III, ALREADY.

to allow wholesale modification ("modding") of the title in ways that are often verboten in many games. It's obviously important to Firaxis that modification of its game be simple and flexible, as the shelf-life of many titles has been elongated by fan-made expansions and enhancements, and by allowing everything from world building, to basic XML characteristic modding, to Python scripting changes, all the way to the ability to change the game's artificial intelligence DLLs. Author Mustafa Thamer shows why mutability is vital to plan for. Oh, he also discusses how to make planet Earth round, in an oddly Slartibartfast style.

BRAIN DRAIN

I was particularly pleased to see the CIVILIZATION IV team supporting modding in such a positive way, since I'm also co-chair of the Independent Games Festival (www.igf.com), which has just announced its eighth Annual Awards. A new Modding Competition category is among the changes for this year (alongside an increased Grand Prize in the main competition).

By the time you read this, though, public voting for the games to be included in the mod category will have ended. There's one Grand Prize for the best total conversion for each of four games. Looking over the votes cast so far, the one notable thing is the diversity of the games that people love to mod. From racing titles, through RPGs and strategy games, to the inevitable FPS onslaught, millions of hours of extra gameplay for your favorite games are created every year by their users, and we felt that was something the IGF should honor as well.

OPENGL PANDEMIC?

But the core of the IGF is still independent games, and with ever more distribution opportunities for indie games, both online and even on consoles (Microsoft's forthcoming Xbox 360 Live Arcade service is rumored to include at least one IGF entrant from previous years), there are ever more opportunities for independent creators to make a living doing what they love. Alongside the polished efforts of the big publishers, innovation on the indie fringes is something vital and visceral to the game industry, too.

Hand of Sid
 We're also delighted to host a second technical article (pg. 15) from the development team of the much-anticipated Sid MEIER'S CIVILIZATION IV over at Firaxis Games, discussing the system put in place

S!

Simon Carless, editor



IT WORKS FOR ME

"ATI's FireGL™ graphics have enabled PSA Peugeot-Citroën to solve problems and complete complex design projects that before would not have been possible."

ALAIN GONZALEZ, TECHNICAL ARCHITECT, PSA PEUGEOT-CITROËN

PSA Peugeot-Citroën, Europe's second-largest auto manufacturer, began using ATI's FireGL™ graphics accelerators in their workstations five years ago and continue to trust them today as an integral part of their 3D design process.

ATI's FireGL graphics are optimized and certified for professional CAD as well as DCC applications and are supported by a unified driver. Advanced features include dual display, dual-link support, stereo 3D output, and PCI Express® high-bandwidth performance.

Find out how ATI's FireGL graphics can work for you at: ati.com/FireGL.

ATI™ FIREGL™

©Copyright 2005, ATI Technologies Inc. All rights reserved. ATI and FireGL are trademarks and/or registered trademarks of ATI Technologies Inc. All other company and/or product names are trademarks and/or registered trademarks of their respective owners.



ati.com

SIGGRAPH 2005 PREVIEW

THE SIGGRAPH 2005 INTERNATIONAL CONFERENCE

on Computer Graphics and Interactive Techniques, held from July 31 to August 4, is the 32nd iteration of this preeminent graphics-related event. Last year, the show in Los Angeles was attended by 27,000



game, science, and film professionals, and featured such major topics as mesh editing, physics simulation (centering around the newly-forged idea of PPU's),

3D displays, and other related hardware innovations. The 2005 Los Angeles show promises to be even bigger, with more than 230 companies exhibiting and 30,000 attendees expected.

Of particular interest is the keynote address by

Star Wars creator George Lucas, titled "A Keynote Q&A with the Father of Digital Cinema." Elsewhere in the program, astute attendees will find panels on the future of computer graphics and interactive technology, such as, "Believable Characters: Are AI-Driven Characters Possible, and Where Will They Take Us?" which includes speakers from Electronic Arts, Quinnipiac University, and Procedural Arts, the company responsible for *Façade*, a character AI-based interactive drama.

Other panels under this banner include discussion on outsourcing CG work, in "International CG Collaboration: Good, Bad, or Just Impossible?" and a talk on open source graphics software, entitled "The Open-Source Movement and the Graphics Community: How Can Open-Source, Third Party, and Proprietary Software Models Coexist?"

This year's Special Sessions theme is "From the Past to the Future of Digital Technologies," and includes the roundtable "Jump! Shout! Dance! Sing! An Interactive Conversation About Games, Game Art, and Play That Goes Way Beyond the

Joystick." This session about alternative gaming input methods features Richard Marks, creator of the EyeToy hardware, *KARAOKE REVOLUTION* developers from Harmonix and Konami, and Fred Swan from peripheral developer Logitech.

Also important to developers is the forum on education and games, including the talk "Lessons Learned From Games for Education." The panel will discuss the most important features of games for promoting learning, and perhaps most timely, how game-based learning transfers to the real world, in a crossover with the so-called serious gaming topics.

New to Siggraph is The Incubator, a showcase for the interactive educational products of today, as well as ideas for the future. This year also marks the first for all high-definition digital projection at Siggraph. The usual conference features will also return, such as the expo floor, the Art Gallery, the Best of Show Awards, the Computer Animation Festival, and the always-popular Cyber Fashion Show.

—Brandon Sheffield

LIBERATE 8

AUTODESK UNWRAPS NEXT 3D STUDIO MAX

GAME ARTISTS AND ANIMATORS ATTENDING

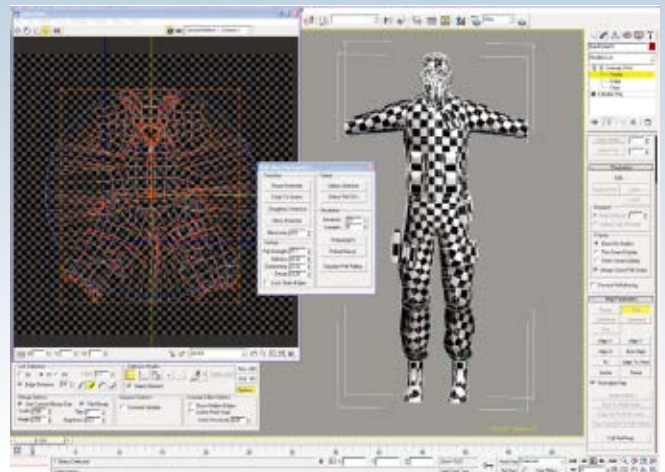
Siggraph will be among the first to see the new 3D Studio Max release, version 8. It's become tradition for the company to unveil a full point release at the show, and this year, game developers can expect to see a handful of valuable features targeted to improve their workflow and speed.

The big new feature for game artists in the next version of Max is Pelt Mapping, a tool whose name is fitting because it lets artists simply slice, stretch, and relax the UV coordinates of a figure, much like automatically opening and stretching an animal's pelt. I recently watched a biped become splayed into a pelt in less than eight seconds, a feat almost as monumental as normal mapping, Autodesk's signature advancement last year when version 7 was released.

In this release, 3DS Max has also turned to improving workflow. For

example, Autodesk is increasing visibility of its Autodesk Vault server—a data management system more well recognized by CAD users—by adding compatibility features to Max, such as an Asset Tracker tool. The Vault server can also assist users in emailing assets to their colleagues quickly and efficiently.

Among other handy updates in 3DS Max 8 are an added Motion Mixer, improvements to biped characters and their animation, a load/save improvement, and a new debugger. The Motion Mixer lets artists load animation clips such that they can be edited for any rig in a non-linear fashion. For bipedal animation and structures, F-curves have been added and more bones can twist than before, giving you the option of creating an Elvis-pelvis character, should your game require one.



Another time saver is a load/save feature that gives users new, open, custom file-formats to save, load, remap, and retarget animation data. And now that Autodesk has added a debugger—Maxscript Debugger—to Max, artists can debug custom scripts which greatly aids in writing and testing the functionality.

3DS Max 8 is expected to be available in the fall for \$3,495. The upgrade from Max 7 will be \$795.

—Jill Duffy

Pelt Mapping lets artists quickly unwrap the UV coordinates of geometry using custom-placed seams, and new Relax methods remove distortion for stretch-free mapping.

Alias Releases SketchBook Pro 2



With a clean and sleek new interface, the new SketchBook Pro 2 (tablet-dependent drawing and painting software from Maya makers Alias) adds three particularly improved features: a quick new way to pan and zoom, a coin-sized brush resize button, and a steering wheel-like toolbar that's rather suggestive of an Apple design. Combined, these features are intended to maximize surface area, letting artists work on a truly clean canvas, stylus and tablet in hand. SketchBook Pro 2 is the first full-feature release since 2002.

Among other improvements, a layers window, used to work with a myriad of layers, eases the use of Marking Menus

to manipulate the transparency, movement, rotation, scaling, mirroring, or flipping of layers. You can now import images into a layer stack and manipulate them to fit, while layered PSD files can be saved for use in Photoshop. Version 2 also allows for 75 brushstroke or action undos with a one-click feature in the tool palette.

SketchBook Pro 2 costs \$179 (download) or \$199 (shrink-wrapped) and is available on Windows XP Tablet PC edition, Windows XP or 2000, and Mac OS X (v10.3 or higher). A nonconsecutive 15-day trial version is available from Alias' web site.

—Alan Bank

GDC EUROPE KEYNOTES ANNOUNCED

THE ORGANIZERS OF THE 5TH ANNUAL GAME

Developers Conference Europe (GDC Europe) have announced the keynote and conference line-up for the event, due to take place between August 30—September 1 at London's Le Meridien Hotel.

The two keynotes will be given by Sony Computer Entertainment Europe's executive vice president of development Phil Harrison and KATAMARI DAMACY creator Keita Takahashi. Harrison will give his keynote session on Wednesday, August 31, when he will discuss next-generation development and the state of the European interactive entertainment community in a Q&A format.

Namco's Keita Takahashi will present the second keynote, previewing new gameplay innovations in the forthcoming sequel to his critically beloved

(and still to be released in Europe) "object aggregation simulator" KATAMARI DAMACY—the sequel is named WE LOVE KATAMARI outside Japan, and Takahashi will discuss his game design philosophy and approach to the sequel in an extension of his popular Game Developers Conference talk.

"Through industry-defining keynotes to next-generation content, GDC Europe provides the European game development community an essential forum in which to learn and network," said conference director Jamil Moledina. "Visionary speakers such as Phil Harrison and Keita Takahashi form the nucleus of current game creation, and are ideally positioned to inform and influence."

Other conference sessions will feature the developers of titles such as KILLZONE, SINGSTAR,



Keita Takahashi

PROJECT GOTHAM RACING 3, with five conference tracks: Production and Management, Tools and Tricks, Case Studies, Vision, and Best of GDC.

The main two days of lectures, panels, and tutorials will be preceded by GDC Europe Mobile on Tuesday August 30, a one day seminar focused exclusively on mobile game creation.

Note: GDC Europe is presented by the CMP Game Group, organizers of the Game Developers Conference and Serious Games Summit, and publishers of *Game Developer* and *Gamasutra.com*.

—Simon Carless

PHOTO BY ROBIN DIAL

IGDA ELECTS NEW BOARD

THE INTERNATIONAL GAME

Developers Association (IGDA), a non-profit membership organization that advocates globally on issues related to digital game creation, has welcomed game attorney Tom Buscaglia as its newest board member, along with Ageia's Kathy Schoback, who has been re-elected to the board.

The IGDA has also appointed new officers for the 2005-06 term. Bob Bates, the Legend Entertainment co-founder and author of *Game Design: The Art and Business of Creating*

Games, was appointed as the new Chairperson, in the place of Carnegie Mellon University's Jesse Schell, who is now officially chairperson emeritus. Existing board member Mitzi McGilvray was appointed secretary, and Schoback will serve as treasurer. Board members Tom Buscaglia (attorney, T.H. Buscaglia & Associates), Jason Della Rocca (executive director, IGDA), John Feil (mission designer, Microsoft Game Studios), and Brian Reynolds (president, Big Huge Games) remain.

"I am honored and proud to be

chairperson of the organization that is the voice of the developer," said Bob Bates. "Making great games has always been about people working together and the IGDA is the ideal organization to help strengthen the community of game industry professionals everywhere."

The IGDA was at one time associated with and partially managed by CMP Media, the company which owns and operates *Game Developer*, but is self-managed as of January 2005.

—David Jenkins and Simon Carless

CALENDAR

EDINBURGH INTERACTIVE ENTERTAINMENT FESTIVAL

Edinburgh International Conference Centre, Scotland
August 10-14, 2005
Cost: £3 per session
www.eief.co.uk

GAME DEVELOPERS CONFERENCE EUROPE

Café Royal, by Le Meridien Piccadilly, London, UK
August 30-Sept. 1
Cost: £75—£600
www.gdceurope.com

Metal Collides

with concrete, wood and glass calculated for force, velocity and speed. Massive explosions throw off shards that decimate everything in their path.

(Processed while hundreds of enemy warriors scatter for cover)

[realism]



AGEIA's Novodex SDK provides the most powerful and advanced toolset for creating dynamic physical environments. With unmatched support for collision detection, fluids and advanced multi-threading for tomorrow's massively parallel architectures, your games will feature the next level in immersive game play.

With AGEIA's PhysX technology, including Novodex and PhysX processor, we've got you covered for today's software physics effects and tomorrow's game play with powerful hardware-based physics. Learn why more than 60 game development studios use AGEIA's PhysX technology. Visit us at www.ageia.com to learn more.



CLAYTOOLS SYSTEM 1.0

BY JAMES ALGUIRE

CLAYTOOLS SYSTEM 1.0



STATS

SensAble Technologies, Inc.

15 Constitution Way
Woburn, MA 01801
781-937-8315
www.sensable.com

PRICE

\$2,795

MINIMUM SYSTEM REQUIREMENTS

Intel Pentium single 2.0GHz or Dual 933MHz or faster CPU, 1GB RAM, FireWire port (IEEE—1394), qualified graphics card.

Operating Systems.

Windows 2000 SP3 or SP4, Windows XP Professional SP1 or SP1A.

Software. Autodesk 3DS Max version 6 or higher, or Rhino. (Maya-compatible version due out soon.)

PROS

1. Short learning curve.
2. Excellent control for manipulation and navigation.
3. Fun to use.

CONS

1. High price point and weak documentation.
2. Users may need to add Firewire card, as some PCs do not have them built in.
3. Works with a limited number of 3D programs.

IMAGINE IF CREATING COMPLEX 3D

models for cinematic effects or game projects were as easy as sculpting a lump of clay. Wearing virtual reality gloves, you could reach into a holographic display and simply grab a primitive object and start molding, pulling, smoothing, and even animating fully realized 3D objects with the same ease of kids playing with Play-Doh.

That vision is closer than you think, thanks to SensAble Technology's ClayTools System 1.0. Available since March, the ClayTools hardware/software combination incorporates haptics (the study of technology and the sense of touch) to bring the sense of touch to creating, modifying, and animating 3D objects.

ClayTools is currently available for Autodesk's 3DS Max and Robert McNeel & Associates' Rhino, with a version for Alias' Maya scheduled to ship this month. This review focuses on the 3DS Max version.

A TOUCH OF CLASS

Being able to virtually touch and feel 3D objects can potentially improve the efficiency of existing 3D development workflows and make 3D development more accessible to artists eager to enter the digital realm but daunted by the complexity of existing tools.

The Phantom Omni Haptic Controller (the hardware component, see image) consists of a stylus attached to what appears to be the shoulder and arm assembly of a robot. Using it is like drawing in the air. The Phantom Omni has six axes of motion: up/down, left/right, forward/back, and three rotations, which provide a surprising range of movement for manipulation, suitable for either right- or left-handed people. Two buttons activate various software functions or act as standard mouse buttons. It can be slightly awkward to use the Phantom Omni to perform standard mouse controls. Several times, I had to perform some hand-based gymnastics to get the cursor to reach the menu item I wanted to access. But it's the built-in force-feedback capabilities that make working in 3D wholly new.

The bundled software includes Windows drivers for the Phantom Omni Haptic Controller, the HapticExtender/MX plug-in for 3DS Max, and SensAble's own ClayTools 3D Modeling software. The installation procedures provided in the printed manual and the PDF documentation on CD-ROM differ somewhat, which can cause needless confusion and can easily stall the user from getting into the tool.

And once you're all set up, you'll also need to give yourself a clear area—about 12 inches on either side of the Phantom Omni—to avoid damaging the device by bumping into objects. Another much-needed hint: Move the Phantom Omni out of the way when not in use to avoid catching the stylus on shirtsleeves and elbows.

FEET OF CLAY

I found the ClayTools software to truly be as easy to use as advertised. Instead of having to deal right away with the technical aspects of modeling, spline construction, or Boolean operations, I simply grabbed a primitive object from one palette and a tool from another and started the 3D creation process. The force-feedback features let you feel the objects that you're creating. Pressing against a 3D object causes resistance in the Phantom Omni as if you were pressing against a real object, making it almost feel like you're working in real clay. Different tools can carve, pull, and smooth geometry with incredible ease. The size of the tool used and the hardness of the virtual clay are easily adjusted on the dynamic toolbar, or better yet, using keyboard shortcuts. It took about 20 minutes to rough out a humanoid head with basic features in ClayTools when a similar project would have taken me a couple of hours or more in another 3D program—and the process was actually fun.

The simple and uncluttered interface organizes tool functions and primitives within floating palettes, and settings can be adjusted in the dynamic toolbar. The



The Phantom Omni Haptic Controller hardware, used with ClayTools, mimics the physical properties of clay by creating resistance.

status bar below the toolbar provides additional information whenever the cursor is positioned over a tool or function.

The Phantom Omni also gives a whole new (I daresay more natural) feel to navigating the 3D space and changing the view of 3D objects being modified. You can turn a 3D object around with just a turn of the hand or twist of the wrist. You can start the modeling process in 3DS Max and add detail to the models in ClayTools, or you can create detailed organic objects in ClayTools to render and animate in Max.

SensAble provides PDF documents outlining the process of moving between Max and ClayTools. Once you understand the process, you can create low-poly 3D objects suitable for use in video games with levels of detail that would be difficult to achieve using other methods, and in many cases, in less time.

EXTENDING YOUR REACH

While the Phantom Omni can replace your standard mouse for 3DS Max, a plug-in (the HapticExtender/MX) can also add to Max many of the cool force feedback features that you find in ClayTools. Actually touching the 3D models as they are created brings a whole new dimension to the program. You can work by touch outside an object, or with a gentle push, you can punch through and work from the inside of an object, again all by touch. You can even make true 3D selections by following the geometry of an object by feel.

Collision detection allows you to place objects on or in other objects with ease. Putting a tool into a game character's

hand becomes a much simpler task when the tool stops moving once it's in the hand. This also comes in very handy when animating. But collision detection isn't limited to objects in 3DS Max. The Phantom Omni also detects the edges of the viewport you're working in, requiring a bit of extra effort to move between viewports. This takes some getting used to and was somewhat annoying at first.

To activate the Phantom Omni's force-feedback enhancements in 3DS Max, click on the Utilities button and then the More button. Select HapticsExtender/MX from the Utilities list and click OK. The Utility Panel displays parameter roll-ups for modifying the behavior of the Phantom Omni device. You can adjust for friction (making objects more or less slippery), touchable range (how far the Phantom Omni reaches in the 3D space), and selection behavior.

Expect to spend a few days using the Phantom Omni device with ClayTools and the HapticExtender/Mx plug-in to become familiar with the nuances of force-feedback modeling, but once you get the hang of it, you'll wonder how you ever worked without it.

THE FINAL TOUCH

The documentation provided with the ClayTools System is adequate, but feels like it was the last item on the list to be completed. The bulk of it is electronic, either as PDF documents or html-based

help files. You'll definitely want to read through the help files, as they provide a good rundown of ClayTools' different functions. Game developers will especially want to become familiar with the workflows for moving between ClayTools and 3DS Max. The workflows do assume that users are well versed in Max and could be confusing for newer users. I would prefer to see the documentation amplified somewhat to give more detail while explaining the process of using the ClayTools System.

SensAble's application of haptics technology brings virtual reality to the mainstream of 3D modeling and has created, in my opinion, a worthy successor to traditional mouse control in 3D applications. The pricing is geared toward the professional 3D market, so it's a bit high for individual use. There are still a few wrinkles to iron out, but for a version 1.0 product, ClayTools is definitely one of the easiest 3D programs I've used. Once you've tried it you'll agree, no other tool gives 3D the same feel.

JAMES ALGUIRE is a Mac professional and Apple Certified Trainer with more than 20 years experience in the computer industry. You can email him at jalguire@gdmag.com.

NATURALMOTION'S ENDORPHIN 2.0

By Justin Lloyd

NaturalMotion's endorphin lets animators direct an artificial human to perform stunts and actions at a higher level than the kinematics and key framing that they are used to. Although endorphin has matured enough to reach version 2.0, it still doesn't really have any competitors; it's the only software package of its type commercially available to animators.

Endorphin uses bio-mechanic techniques along with artificial intelligence (AI) to simulate the motion and reactions of a human body in a physical environment. The AI arrives pre-programmed with the rules of the human body's mechanics: how a body moves, joint limitations, natural positioning, and limb mass, coupled with an easy-to-understand physics simulator.

Endorphin is not a 3D rendering package—it doesn't replace your regular 3D package. It's a stand-alone software package that works in conjunction with your familiar environment. In fact, with the new ability to change key layouts and mouse controls, you can make it work just like Maya or Softimage or 3DS Max. Endorphin is used for creating human character animations which you then import into your scene just as if you were using Character Studio or motion-capture data.

To use endorphin, animators first employ ordinary animation techniques, focusing on the gross motion of an action, for example, getting a character to look realistic as she pitches backward down a flight of stairs. Endorphin puts in the gross motion for you and allows you to shift your focus to the smaller details, fussing over how the hands are positioned, the fold of the clothing, or where the character is looking. In addition, animators can import animation data created with ordinary animation techniques, and continue or modify it in endorphin.

ADAPTIVE BEHAVIORS

NaturalMotion's software is designed to control a human body within a physical environment to make it perform simple, mostly reactive, actions (called Adaptive Behaviors) such as falling or reacting to a physical impact. These types of reactions can be directed by the animator from a



NaturalMotion's endorphin 2.0 can add Adaptive Behaviors to character animations.

ENDORPHIN 2.0



STATS

NaturalMotion
33-35 George Street
Oxford OX1 2AY
United Kingdom
+44 (0) 1865 250 575
www.naturalmotion.com

PRICE

\$12,795, or monthly license

MINIMUM SYSTEM REQUIREMENTS

Pentium or Athlon 1.7 GHz CPU, 512MB RAM; Nvidia GeForce 2 or ATI Radeon 7000.

Operating Systems.

Windows 2000, XP.

PROS

1. Allows the creation of natural looking bipedal animations very quickly.
2. Integrates well with existing animation pipelines.
3. Very easy to learn and begin using.

CONS

1. The small details of an animation still require the fine tweaking that they always did.
2. Only provides play back for "canned animations" in your game engine or cut scene.
3. Prohibitively high price.



>80%

25 of the top 30 media & entertainment companies trust Akamai.

They trust Akamai to deliver more than content. To deliver billions of downloads daily and give users what they want when they want it. Why? Because Akamai provides instant scalability to handle spikes in traffic, delivering positive user experiences and increased customer loyalty.

Start learning more and get your copy of *Online Games: 10 Tips to Increase Play and Profitability*.
Call 888-340-4252 or visit www.akamai.com/gaming



simple user interface.

If you've worked with motion-capture data, you're probably a little more predisposed to understand endorphin than animators who haven't. You'll get the essence of the animation quickly, leaving you free to define the particular nuances of a performance. Endorphin lets you give a character high-level directions, such as "protect your head with your hands, tuck into a ball, and do try and land on your feet this time." Given the constraints of the environment, the character will attempt to comply with your requests.

Endorphin characters are pose-able at various time increments within the animation and will attempt to comply within the physical constraints of the environment. For example, you pose the character's arms and hands, but there isn't quite enough room, so she does the best she can.

But unlike rag doll simulators, which every major 3D animation package provides, endorphin gives animators more control over the character's reactions. When you pitch a rag doll character down a flight of stairs, the forces acting on her are gravity and the force of how hard you pushed. Once she is set on her way down, what unfolds is mostly out of your control. She acts just like her namesake, a rag doll.

Endorphin, on the other hand, gives you the ability to make the character react more intelligently, more like a real human, to actively respond in near realtime to her environment rather than let her body be flounced around passively.

TIME'S MAKING CHANGES

NaturalMotion has added a number of changes since the first iteration of endorphin, but two are noteworthy: improved behaviors and overlapping behaviors.

First, the behaviors have been vastly improved and extended over earlier attempts. New behaviors in version 2 are more fluid and natural compared to the slightly rag doll-ish responses in earlier versions. Second, animators now can add multiple, overlapping behaviors to a character, which can further be ordered

hierarchically so the character reacts as if one type of response is more urgent than another. Behaviors can also be assigned to different body parts. As each behavior becomes more or less urgent than the others, you come to see what fine control you actually have over the main action being performed.

Here's an example: You have a character who, while simultaneously trying to catch a ball, is being tackled by another character. If you set the ball-catching behavior as the more urgent of the two, the character will both dive to catch the ball and respond to the tackle appropriately.

MOCAP SUPPORT

Endorphin is not a motion-capture playback package, even though it offers mature mocap tools that support most of the major formats. You can import a mocap file and endorphin will intelligently retarget the data to your character's skeleton.

In the new version, you can transition instantaneously and seamlessly from mocap to behavior simulation and back to mocap very easily by manipulating the time graph. The power of the new mocap system really shines when you integrate behaviors in addition to mocap data. Now, you can have mocap only affecting certain parts of a body as a simulated behavior or active pose controls the other parts. So you can take a run-cycle and integrate it with a behavior for reaching out and catching a ball. Even if your mocap data set doesn't contain the exact sequence you need, you can easily create one from a single performance and two or three behaviors.

NOT SO ADAPTIVE BEHAVIORS

Endorphin v2.0's Adaptive Behaviors only work on human biped figures. I was unable to get the AI to handle either a quadruped or a hulking man beast that ambulated the way a gorilla does—on feet and knuckles. You can make a bent-over humanoid figure that drags his knuckles, but I never managed to get one ambulating using arms as well as legs. The software doesn't restrict the type of skeleton you can built or import, but it does eliminate your ability to use Adaptive Behaviors on these figures.

FEEL THE MAGIC

As a programmer, I'm interested in how NaturalMotion created the AI for the endorphin characters. They toss around terminology like so much technological confetti—neural networks, genetic algorithms, learning behaviors, biomechanical neural models—that you cannot be exactly sure what techniques they're using. As an artist making use of endorphin, I just need to scratch out those phrases on my brain and instead focus on the phrase that counts: "magic happens here."

And magic really does seem to happen in endorphin. The ability to give high level direction to a character, mixed in with your motion capture data, makes animating people occur a lot faster and easier.

There will be a few new pieces of terminology to learn with regard the physics, e.g. how slippery to make a surface—concrete or ice rink? How much emphasis should the character place on a particular pose you want it to make compared to what else is happening—should it strike a dramatic victory pose or watch out for the 300-pound fullback about to introduce its face to Mr. Astro turf?

How long will it take you to really get inside this package? For me, a regular Maya and Softimage user, about two days total to become really comfortable with it—but it took less than a full week to begin pushing the boundaries.

The user interface is clean and easy to understand for anyone familiar with a major 3D package. The learning curve is nowhere near as steep as Maya or Softimage. If you can figure out iTunes you can figure out endorphin. ❖

JUSTIN LLOYD is director of business development for Infinite Monkey Factory, a game development studio based in Los Angeles. He has worked with software and hardware for 27 years, 20 of them specifically on video games. Email him at jilloyd@gdmag.com.

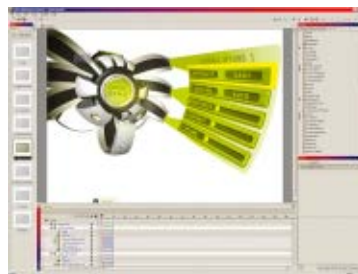
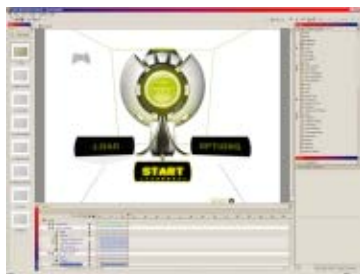
ANARK GAMEFACE

Build great UIs in half the time



Anark Gameface, the first professional solution for creating 3D game user interfaces and front ends. Cut your development time in half by building a bullet proof UI pipeline.

Now an authorized middleware tool for the Xbox 360.



Who wins?

- **Programmers:** Code the pipeline once and never touch the UI again.
- **Artist:** Use a professional authoring tool with real workflow.
- **Producers:** Save money and reduce schedule risk.

Anark Gameface Bundle Includes

- **Anark Studio:** Proven interactive 3D authoring tool
- **Anark Format SDK:** All the tools you need to import Anark UI projects into your existing game engine.

For more information go to www.anark.com/gameface or call 866.705.1010 x102.



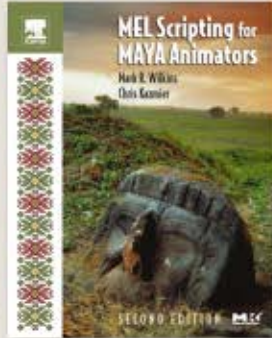
WWW.MKP.COM



WWW.FOCALPRESS.COM



Complete Maya Programming Volume II: An In-Depth Guide to 3D Fundamentals, Geometry, and Modeling
by David A.D. Gould



MEL Scripting for Maya Animators, Second Edition
by Mark R. Wilkins and Chris Kazmier



3ds max 7 New Features and Production Workflow
by Autodesk



3ds max 7 Fundamentals and Beyond Courseware
by Autodesk



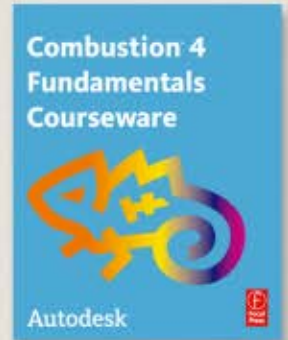
Visualizing Quaternions
by Andrew J. Hanson



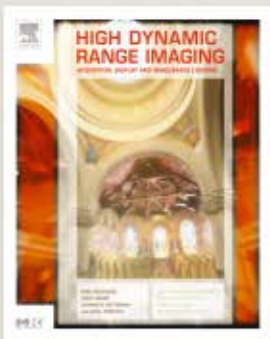
Real-Time Collision Detection
by Christer Ericson



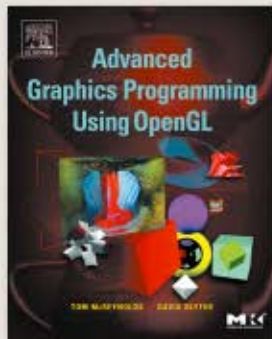
Director MX 2004 Games: Game Development with Macromedia Director
by Nik Lever



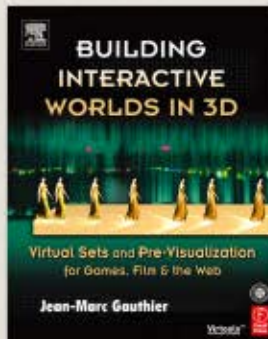
Combustion 4 Fundamentals Courseware
by Autodesk



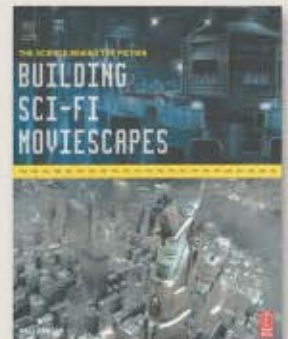
High Dynamic Range Imaging: Acquisition, Display, and Image-based Lighting
by Erik Reinhard, Greg Ward, Sumanta Pattanaik, Paul Debevec



Advanced Graphics Programming Using OpenGL
by Tom McReynolds and David Blythe



Building Interactive Worlds in 3D: Virtual Sets and Pre-Visualization for Games, Films & the Web
by Jean-Marc Gauthier



Building Sci-Fi Moviescapes: The Science Behind the Fiction
by Matt Hanson

ORDER FROM ELSEVIER AND RECEIVE A 15% DISCOUNT AND FREE SHIPPING!

Refer to offer code #80491

Mail: Elsevier, Order Fulfillment Department, 11830 Westline Drive, St. Louis, MO 63146

Phone: (800) 545-2522 / (314) 453-7010 [Intl.] **Fax:** (800) 535-9935 / (314) 453-7095 [Intl.]

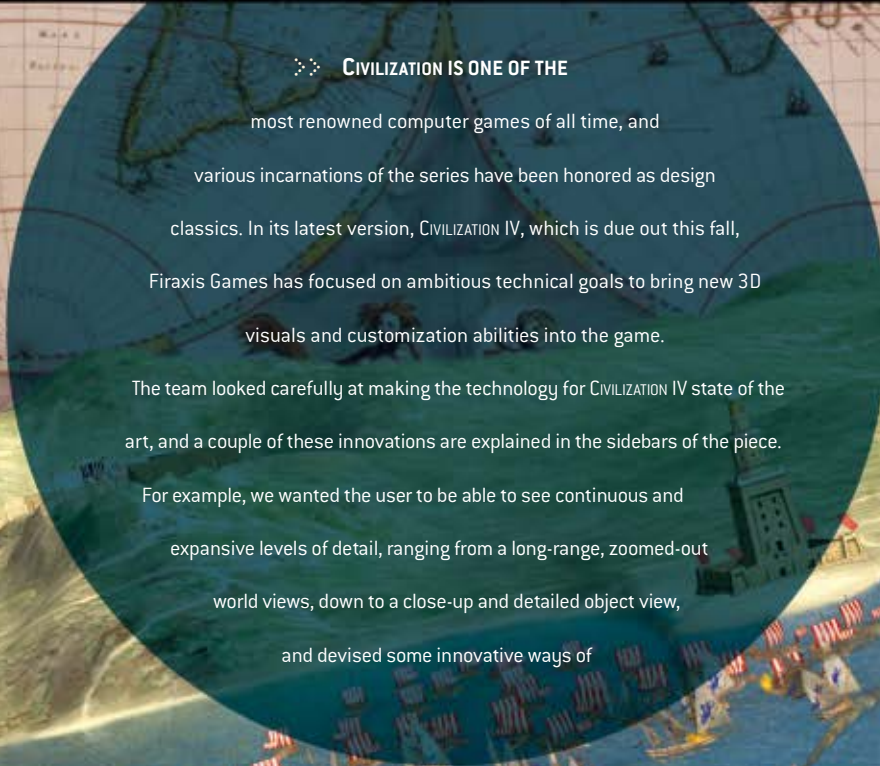
Web: www.mkp.com / www.focalpress.com **Email:** usbkinfo@elsevier.com



ACT *of* MOD

» mustafa thamer

BUILDING SID MEIER'S CIVILIZATION IV FOR CUSTOMIZATION



» CIVILIZATION IS ONE OF THE most renowned computer games of all time, and various incarnations of the series have been honored as design classics. In its latest version, CIVILIZATION IV, which is due out this fall, Firaxis Games has focused on ambitious technical goals to bring new 3D visuals and customization abilities into the game.

The team looked carefully at making the technology for CIVILIZATION IV state of the art, and a couple of these innovations are explained in the sidebars of the piece.

For example, we wanted the user to be able to see continuous and expansive levels of detail, ranging from a long-range, zoomed-out world views, down to a close-up and detailed object view, and devised some innovative ways of

MUSTAFA THAMER is a 14-year veteran of the game industry and joined Firaxis Games in 2003. He has worked for companies such as NDL, HeadSpin Technology, and Cyan Worlds, focusing on 3D environments and game engines. Mustafa is currently the lead programmer for CIVILIZATION IV and can be reached at mthamer@gdmag.com.

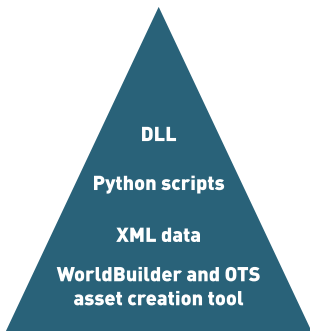


Figure 1 The tiered mod tools are shown.

THIRD-PARTY CIV IV LIBRARIES

- NDL Gamebryo
www.ndl.com
- ScaleForm GFC
www.scaleform.com
- Bink Video
www.radgametools.com
- Miles Sound System
www.radgametools.com
- Python
www.python.org
- Boost
www.boost.org

accomplishing this.

In addition, there were also some goals we set for ourselves that required a completely new engine, game architecture, and technologies. For example, we set out to craft a message-based events system for built-in networked play, as well as support for multiple multiplayer modes and stand-alone server applications.

The inclusion of these various elements led us to a natural conclusion—even though CIVILIZATION IV is not a genre of game which users naturally expect to be modifiable, such as a first-person shooter, we wanted to make an extremely ‘moddable’ title which would help the game’s longevity and allow users to change almost any aspect of gameplay and graphics. We wanted to house a data-driven engine that would allow extensive mutability, as well as hot-swapping of assets. We added a comprehensive scripting layer and modular game DLL SDK. And finally, we included support for low-end, non-gamer machines.

MODDABILITY

CIVILIZATION IV supports a tiered approach to user modification (see Figure 1). There are four basic entry points to modifying the game, ranging from novice to expert skill level.

1. *The WorldBuilder*. This in-game map editor tool lets you customize the terrain and place units, cities, buildings, resources, and other objects. Your work is saved in a text-based, code-independent format, which is parsed by a Python script on reload.
2. *XML*. All the relevant game variables, text, asset paths, and rules are located in standard XML files which can easily be modified using an XML or text editor. Using XML provides unfettered access to a wealth of data.
3. *Python*. All high-level game functionality in CIVILIZATION IV has been exposed to Python. Scripts can be written to generate maps, modify the interface, trigger game events, override AI, and so forth.
4. *GameCore DLL SDK*. The core game and AI code will be provided as an SDK, which builds a DLL that the application loads at runtime. Users with a compiler can modify, replace, or extend the low-level C++ code that plays the game.

We expect that a lot of casual users will play with the WorldBuilder, since it’s a simple tool built into the game. As the complexity of the tools increases, the number of players that use them will fall off rapidly. However, it only takes the creativity, perseverance, ability, and time of a small number of people to create excellent game modifications for others.

Looking at it another way, the tiered mod tools define the four basic data types that drive the engine as inputs. We made sure that all data inputs were hot-swappable; any changes were automatically detected and reloaded into the game on the fly. Creating a data-driven engine is generally accepted as good practice, according to Fermier (see References), since the benefits usually outweigh the cost and complexities involved in setting one up.

In our case, it helped the development team rapidly plug new art and code into the game, as well as being a necessity for the mod community. In turn, using the tools in our day-to-day development helped make them much more solid and robust, which will benefit the intended audience. Of course developing a code and data pipeline for external use is always more difficult

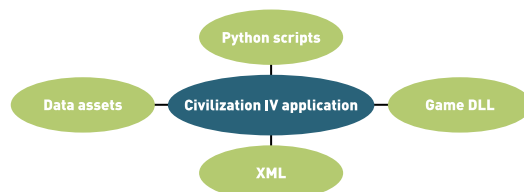


Figure 2 Data inputs are represented graphically.

than using something for internal use only. You have to go the extra mile to make sure everything is well organized, clearly documented, stable, and secure (see Figures 1 and 2). Regardless, once your system is released, someone out there will try to abuse it in a way that you never expected.

UNIT FORMATION EDITOR

One example of a tool that was easy to develop since we had extensive modding capabilities was the formation editor. Since one of our goals for CIVILIZATION IV was to make practically everything editable by a player, we had to consider this each time we implemented a feature. Things that were easy to hardcode in the past, such as where units stand within a square, became quite a bit more complicated when we considered that the user could shuffle these numbers around.

The formation editor (see Figure 3) was programmed in Python and hooks into the game’s XML data files so it can stay in synch with whatever changes the user makes in other applicable sections of the game. It gives full control over how, where and, most importantly, in what circumstances units take a particular formation.

GAMECORE DLL

We decided that it would be advantageous to isolate the CIVILIZATION IV core game and AI code in a DLL. This approach provides at least three benefits. First, the gamecore DLL code can be provided as an SDK to advanced modders who want to customize the intrinsic game code and structures. Second, other game engines, future versions of CIVILIZATION, different platforms, and other related applications and tools can easily use the DLL code. This is made possible by making gamecore DLL a pluggable module, which makes no assumptions about the underlying platform or hardware. Third, the gamecore DLL can be updated and patched at runtime, making it easy to make significant game or AI changes without rebuilding the game application.

The disadvantage of using a DLL is the additional setup and maintenance overhead that it entails. In terms of setup, the code had to be reorganized so that its interface to the application was minimal and clean, so many dependencies were removed, as well as proprietary code. We also had to be careful when adding new files and functionality to make sure that the clear delineation between the DLL and the application code remained intact.

Once we decided to move the core game and AI code into a DLL, we had a few more problems to solve. First, we had to decide what should go into the DLL and what should be kept out. Intuitively, we had a good idea of what was engine code or app-level code, and what was game code, so most things fell into one camp or another without much of a problem. While we wanted the DLL to contain only the code specific to the game of CIVILIZATION IV, it needed to be ignorant of the interface, graphics,

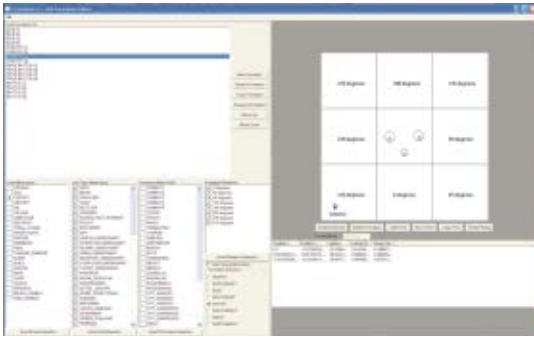


Figure 3 The unit formation editor hooks into the game's XML data files.

sound system, and application platform. This led to problems with game entities such as units and cities, whose classes had both game and engine entity code intermingled. Moving the engine entity code into another class residing outside the DLL separated the code into its logical parts, but required a solution to a simple question: How will the DLL communicate with engine-specific entities?

DLL EXPORTS AND APPLICATION INTERFACES

The DLL publishes functions to the application by using the `_declspec(dllexport)` keyword, allowing easy access of DLL code from the CIVILIZATION application (see Figure 6). In order to support reverse communication from the DLL back to the application, the application exposes interfaces for numerous subsystems, including the interface, engine, sound, entities, and network. Each accessible system provides a pure abstract base class (ABC), defining the functions that are exposed (Figure 4). By giving the DLL access to the abstract class definitions and pointers, the DLL can be built without the application source and can still access the application's key systems at runtime. In this case, polymorphism and the ABC mechanism allow us to separate the interface from the implementation.

CODE INTERFACES

One of the most critical aspects of software design is developing the correct interface for a class or system. The interface defines what can be done; the implementation describes how to do it. This "what versus how" distinction is important to understand. The interface must be flexible, yet simple, so that it can evolve, stay relevant, and be reused (open and closed).

At some point, the implementation details become unimportant compared to the contract that is described by the interface. The interface guarantees that any objects that publish it (reasonably) support the methods described by the interface, so calling objects know what to expect.

For CIVILIZATION IV, at a high level, we dealt with many different interfaces to the code, due to the different tiers of access that we provide, as well as the fact that our code was split across the application, DLL, and Python. At that level of complexity, it's easy to turn your code into a big ball of mud (see Foote in References), if you're not careful to structure things intelligently.

We managed to include a number of code interfaces in CIVILIZATION IV, such as:

- Standard public/private class interfaces.
- An exported DLL interface. A subset of the public interface (delineated with keyword `DllExport`) was exported to the

application.

- An exposed Python interface. A different, (but possibly overlapping) subset of the public interface (delineated with a `PyExport` comment tag) was then exposed to Python. This was the case for both the app and the DLL.
- A server application interface, which consisted of simple public entry points for graphics, audio, and interface classes that were stub overrides.
- An exposed application interface, which provided a set of abstract base classes that allowed the DLL to connect to it.

XML

As mentioned earlier, we used XML as our data description language. Although we had basic tools, such as ini files, and more complex tools, such as Python, XML fit the sweet spot as a simple yet powerful mechanism for getting data values into the game. Exposing code variables in an easily modifiable way is invaluable to developing, tuning, and tweaking the game, and is necessary for supporting game mods, which was one of our original goals.

Beyond game rules and variables, other XML data includes camera variables, animation paths, asset file locations and scale, input mappings, localized text, script function hooks, audio script parameters, and much more. XML is a good match for the CIVILIZATION team's needs because its data is largely self-describing, and there are many good off-the-shelf editors available.

For parsing XML files, we stuck with MSXML, Document Object Model, since it offered reasonable performance with the large amounts of XML data we had. Data objects were written so that they could read and write themselves from various data-streams, including XML. Since the data object classes were heavily inherited, super classes were responsible for serializing their own data members. We took advantage of object serialization to utilize caching.

Once data objects are read by XML, they write themselves back out using an extremely efficient native binary format. From then on, as long as the XML files aren't touched, the data

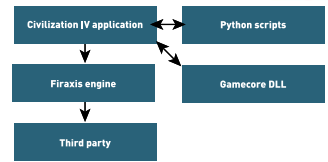


Figure 4 CIVILIZATION IV's code structure keeps Python scripts and the gamecore separate from other data.



Figure 5 Even with a trivial set of rules, it's possible to come up with moderately convincing city layouts. Guiding the application of these rules by distance and content heuristics can produce the wide variation in city configurations possible in CIVILIZATION IV.

City Layout

In CIVILIZATION IV, we have tried to bring cities to life by creating a novel look and feel for each one within the main game, allowing us to eliminate the need for the separate "city view" present in previous CIVILIZATION versions. Two constraints make city layout a nontrivial problem.

First, in order to make a city's current contents obvious by inspection, the layout algorithm is required to place specific buildings in obvious central locations.

Second, all cities must occupy a single plot, regardless of size, necessitating a very dense packing of buildings in the case of larger cities. Achieving such dense packing while also adhering to our building placement goals using pre-built city models or generic random placement algorithms is very difficult.

Cities in CIVILIZATION IV were heavily inspired by other works that used Lindenmayer systems (a.k.a. L-systems) to create their plants and trees and more recently, city layout (see Parish,

References). CIVILIZATION uses similar L-systems to perform city layout. Just as in regular L-systems, a variety of rules are specified (in XML data files) for subdividing city plots into child plots. Recursive application of these rules (see Figure 5), are used to create a variety of organic-looking block layouts into which specific buildings are packed. Rather than defining transition probabilities in order to constrain the resulting layouts (which is often done in organic L-systems) we view the city layout problem as a tree search challenge guided by simple weighting heuristics. By assigning priorities based on a rule's ability to satisfy our city placement constraints, we are able to guide the system to meet our final requirement both in terms of mandatory building placements as well as overall city size and shape. This approach is fast, artist-friendly and most importantly, gives cities that extra "kick" that they need in order to come alive.

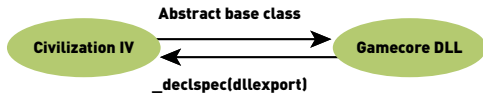


Figure 6 DLL and application communication.

REFERENCES

Dawson, Bruce. 2002. *Game Scripting in Python*. Proceedings of the annual Game Developers Conference, San Jose, CA.

Fermier, Rob. 2002. *Building a Data-Driven Game Engine: Case Study: The Age of Mythology*. Proceedings of the annual Game Developers Conference, San Jose, CA.

Foote, Brian and Joseph Yoder. 1997. *Big Ball of Mud*. Proceedings of the Pattern Languages of Program Conference, Monticello, IL.

HAP Python Remote Debugger, Humongous Entertainment. 2001. <http://hapdebugger.sourceforge.net>

The Open-Closed Principle. www.objectmentor.com/resources/articles/ocp.pdf.

Parish, Y.I.H. and P. Muller. 2001. Procedural Modeling of Cities. Proceedings of ACM Siggraph, Computer Graphics, Annual Conference Series, ACM (August): 301-308.

van Rossum, Just and Paul Moore. 2002. *New Import Hooks* (PEP 302). www.python.org/peps/pep-0302.html.

objects read and initialize from the much smaller and faster binary files, in a fraction of the time.

PYTHON

Like XML, scripting was extremely useful as both a mod tool and an internal development tool. If you don't have any need to expose code and algorithms in a simple and safe way to others, you can argue that providing a scripting language is not worth the effort. However, if you do have that need, as we did, scripting is a no-brainer, and it makes complete sense to use a powerful, documented, cross-platform standard such as Python [see Dawson in References].

Python, like many good technologies, soon spreads virally throughout your development team and finds its way into all sorts of applications and tools. In other words, Python begins to feel like a big hammer and coding tasks look like nails.

Python scripts are used in many areas of the game. Initially, we thought scripting would basically be used for map-generation algorithms as well as triggering in-game events. Eventually, we wound up using Python for interface layout, debugging tools, tutorials, the server front-end, the Civlopedia, the unit formation editor, and game pop-ups and screens. For user customization, we also exposed every major high-level game function and event; there's quite a lot that you can modify and add to CIVILIZATION IV via scripting.

Embedding Python into our app was fairly straightforward. We supplied our own hooks to access stdout and stderr along with exception output, so we could dump output to logs and raise pop-ups when there was a problem. We also integrated Python into our game console so that it could pose as a simple Python interpreter. Finally, we needed to load Python code from our own PAK files, so we registered import hooks to intercept the standard Python method of finding and



Figure 7 The CIVILIZATION IV globe view is created by using only a portion of a sphere instead of a fully rounded, morphed shape.

loading Python modules on import (see van Rossum in References).

Extending Python with our own C++ code was also not difficult, thanks to Boost Python, but it took a little bit of work. Classes were wrapped so that simple, safer versions were exposed to Python, providing a firewall layer and a decoupling between script and the application. On the other hand, structs and enums were exposed directly to keep things simple. For debugging, we customized the HAP debugger, which worked really well and was invaluable at times.

Overall, we believe that we've created CIVILIZATION IV as a truly, deeply moddable title. We look forward to having consumers get their hands on our game later in the year, and discovering the kind of amazing mods they can create modularly and easily using the tools we've provided to them. Viva la mod! ❖

The author acknowledges the technical staff at Firaxis Games and gives special thanks to Nat Duca (Globeview, City layout), Bart Muzzin (Unit Formation Editor), Nathan Mefford, Tom Whittaker, and Kelley Gilmore for their contributions to this article.

Civilization Globe View

CIVILIZATION IV can seamlessly zoom from a detailed, close-up view, all the way out to an outer-space globe view. The sole purpose of the zoom could have been just for great eye candy, but the game uses this distant zoom level to help visualize and interact with elements that are too wide-scale to be visualized at closer levels. For example cultural spread, military threats, resource locations, and city prosperity are much easier to see from a global perspective. The seamless transition from the detailed CIVILIZATION world to the globe view is achieved using a textured impostor and an incremental morphing algorithm. At a specific camera zoom level, the engine's terrain is replaced with a low-resolution approximating mesh. By texturing this mesh with an image of the actual world we are able to turn the creation of the globe from a massive and complex transformation problem into a single morphing problem.

One of the main problems with using a globe in CIVILIZATION IV is that the globe system needs to work with a wide array of different map dimensions. To understand why this poses

a problem, just try bending a piece of paper into a sphere. What you will find is that certain paper sizes are amenable to "spherization" while others require heavy deformations in order to fit the proper shape. To avoid these visual distortions altogether we came up with the virtual sphere approach to globe rendering [see Figure 7].

Rather than bending the terrain into a full sphere—that is, bending it a total of 360 degrees so that the globe's east edge touches the west—we only bent it a fraction of that amount, usually about 150 degrees, which left us with only a small segment of a full sphere. While this solves the distortion problem [because it reduces the amount of deformation applied to the mesh], a new problem emerges: The resulting world no longer looks like a sphere. The virtual sphere approach solves this problem by rendering a second copy of the partial sphere transformed such that the opposing edges of the sphere appear to line up. This illusion is quite effective, going unnoticed by all but the keenest of observers.

Alias demonstrates full range of industry workflow solutions



For more than twenty years, Alias has been developing award-winning software solutions for the entertainment and design industries. Maya® is the top choice of leading filmmakers, and has been used in many films that have been nominated for an Academy Award® for best visual effects. In 2003, for the awards year of 2002, Alias|Wavefront™ was awarded an Oscar® for scientific and technical achievement for the development of Maya software. In entertainment, Alias MotionBuilder® is an integral part of the pipeline for many film, game, video and broadcast professionals. It has long been established as the pre-eminent character performance and animation package. As for the world of design, Alias StudioTools™ leads the way and is used by most major automotive manufacturers in the world. Award-winning product designers also depend on it for unparalleled technology,

flexibility, and depth. Other software solutions include: Alias SketchBook™ Pro, Alias ImageStudio™, PortfolioWall®, mental ray® Standalone, FBX®, and HumanIK® Middleware.

Accompanying their diverse product line are industry-leading education and Alias' service solutions. From the wide range of self-directed Learning Tools that include books and DVDs, to the full-service, customized consulting solutions offered by Professional Services, Alias can offer as much support as you need. The Platinum, Silver and Bronze Memberships continue to be a huge success with computer graphics professionals around the world. And for educators, Alias has a special line of solutions geared specifically for you. Leading educational institutions consistently choose Alias solutions to prepare their graduates for successful careers.

IN GAME DEVELOPMENT THERE ARE
NO SECOND CHANCES.
MISS A DEADLINE OR SHIP A BUGGY PRODUCT AND IT'S
GAME OVER.



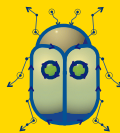
DELIVER QUALITY. DELIVER ON TIME. USE SEAPINE.

Powerful, cross-platform TestTrack Pro and Surround SCM help you manage game development from start to finish—spend less time chasing problems and more time creating quality games.

TestTrack Pro

Advanced Issue Management

- Defect linking
- Robust reporting
- Fast, secure remote access
- Beta site feedback
- Easy to use and administer



Surround SCM

Flexible Change Management

- Advanced code branching
- Changelists and atomic transactions
- Email notifications
- Fast, secure remote access
- **NEW!** Image browsing and thumbnail support



TRUSTED BY LEADING GAME DEVELOPERS

ION STORM, ATARI, CLIMAX, SPORTS INTERACTIVE, KUJU ENTERTAINMENT,
and other top game development companies trust Seapine products to help them deliver bug-free games. Is your company the next Seapine success story?

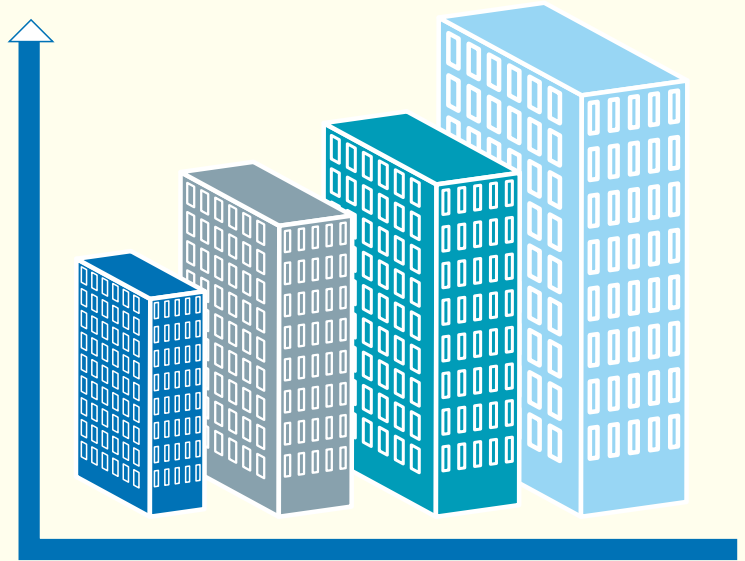
Visit www.seapine.com/gamedev to learn more about TestTrack Pro and Surround SCM and download evaluation copies.

 **Seapine Software**™

JOSH RESNICK

◆◆ BUSINESS LEVEL:

GROWING PANDEMIC



IT WAS IN THE SUMMER OF 1998 THAT

Andrew Goldman and I left Activision and headed off with 10 other brave souls into the often stormy sea of independent game development. While today that sea is as wild as ever (and our ship is a lot bigger), we've been fortunate enough to recruit some amazing talent, produce several successful titles, and build a studio that we're really proud of. And while it's true we occasionally started down the wrong path, we were able to find our way back because we always kept in mind our core values, commitments to our people, and dedication to creating industry-leading entertainment.



SCALED FOR CHANGE

From the very start, Pandemic was somewhat unusual compared to many game start-ups

because we had two projects and two production teams. This initial scale was in part a result of the opportunities we had at that time, but it was also in part an expression of our strategy. We were leaving Activision with the core teams that had developed *DARK REIGN* and *BATTLEZONE*, and in a volatile industry like ours, we believed that there was strength and stability in a multi-project slate. If

Activision had canceled one of our initial titles (and thankfully, it didn't), we wanted to make sure there was another project in production, so we could shift people to work on it for a while instead of having them sit around soaking up overhead. When that happens, you suddenly find yourself lunging at any project you can find, and quality becomes a casualty.

Not everyone knows it, but we've had the opportunity to test this strategy on several occasions when projects have been canceled. Several years ago, while carrying four small production teams, we dropped a project from our slate. After several months of work on the title, Andrew and I recognized we had the wrong team working with the wrong technology on a game deal that wasn't taking us in the right direction. When the project was finally terminated, however, we had work waiting for the people transitioning off the team because we had the scale to absorb them.

The same principle works in reverse. Sometimes it's not an issue of a project being canceled but of it needing help. When our team in Australia required additional resources to finish *DESTROY ALL HUMANS!*, we were able to offer them fresh reinforcements from the people at the Los Angeles studio coming off of *MERCENARIES*. Some of them even made the trip all the way down to Brisbane, Australia, to work on site for a few months.

As each project comes to a close, you begin to see whether your optimistic staff transition plans from a year or so before bear any resemblance to what the project ultimately required. It's a balancing act every time. Up to this point, however, things have always worked out: I'll happily report (as I knock on wood) that in Pandemic's seven years in

business, we've never had to lay off a single employee.

A ROAD NOT TAKEN

Both Andrew Goldman and I have business backgrounds, and I like to think that in a number of instances over the years, we've demonstrated that we will at least keep the studio out of serious trouble. That's not to say we haven't been tempted to wander down some blind alleys.

Most people have forgotten that, many years ago during the dot-com bubble, we briefly dabbled in online casual gaming. We even had a few of our then-tiny staff penciled into an online gaming group.

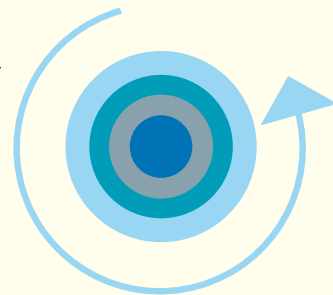
We talked about the possibilities of some sort of series of web-based games that promoted consumer products. We talked, we considered, but we also saw fairly quickly that it was a direction which was entirely disconnected from our interests and expertise.

From time to time we do get hyped up about new ideas and new directions, but we're careful not to let our passions overrule our business principles. It's a track record that's significant to our staff as well. In an industry where companies rise and fall overnight, it's very important to our people to have confidence that their jobs aren't going to suddenly evaporate because Andrew and I weren't thoughtful about our commitments.

'WE'RE GOING TO NEED A BIGGER BOAT'

While we still strive to keep our costs down and maintain our small company fiscal vigilance, the logistic and support challenges we face today are far broader in scope than in the one-coffeemaker start-up days. Not surprisingly, one of the largest of those challenges is making sure we have the physical space we need to

JOSH RESNICK is president and co-founder of Pandemic Studios. Over the past seven years, Josh and his partner, Andrew Goldman, have managed Pandemic's work on titles such as *FULL SPECTRUM WARRIOR*, *MERCENARIES*, *STAR WARS BATTLEFRONT*, and *DESTROY ALL HUMANS!* at their offices in California and Australia. Reach him at jresnick@gdmag.com.



MERCENARIES is one of the more notable recent Pandemic titles.

accommodate our growing operation. While the average game project will run one to two years, the typical office lease runs five. What game company can say with any confidence what its needs will be in five years? What company is in the position it thought it would be five years ago?

Even though the future isn't always clear, we've learned that we can take certain approaches to minimize risk and maximize flexibility. We started out in a warehouse down by the beach in Santa Monica, Calif., and while being there might appear cool in a "creative beach community" sort of way to some, it ultimately didn't offer us the space flexibility we needed to accommodate the company's growth. Once we ran out of room, we had no options to expand, and we had to move the whole studio.

Now, we're in an office tower, where new spaces frequently become available. Because of this, we didn't have to solve our most recent space crisis by moving the company again—we just leased an additional floor.

INSIDE AND OUT

An office tower may not sound like a cool and sexy place to work, but we've been able to put together a great development setting. With a little imagination (and a lot of extravagant spending by a former tenant), you can make almost any space sing. The really important factor in the equation, we believe, is location. Westwood is not the cheapest place in the Los Angeles area to quarter a company, but it's hard to match for a concentration of food and recreational

outlets. Being in Westwood allows us to offer our employees not only an office but a lifestyle.

The internal details are also extremely important for an expanding company. We believe in the open office. Although it's important to provide each of our specialized people with a personal space they can call their own, we've found that closing everyone up in private offices cuts essential lines of communication. Each section of a production team is grouped together: the designers have their whiteboard-festooned cove in one area, animators and their full-length mirrors are grouped in another area, a pair of concept artists hunch over their tablets in another section, and so on.

The challenge comes in that each of these subgroups, as well as the team as a whole, expands and contracts over the life of the project. To support this dynamic, we've purchased modular, light-weight, off-the-shelf retail furniture that can be constructed and easily reconfigured by our own in-house staff. Individual workspaces are delineated but not isolated, and the whole arrangement works together to encourage communication while facilitating individual focus.

POWER TO THE PEOPLE

More than anything else, it's been important for us to create an environment in which our people feel a strong sense of belonging. As Pandemic has grown, we've come to see that creating this atmosphere is one of our most critical challenges. Those of us who have worked for large companies know how easy it is to get lost in the crowd and feel like a cog in a machine rather than part of a community.

When we started Pandemic, there were just a dozen people in the company. We would sit around after work on our thrift-store sofas, crack a few cold drinks, and gab until we had solved a respectable number of the world's problems. Sometimes we would even wander down to the beach and play volleyball. There was a real closeness. We all felt like we

were part of something and that each of us mattered.

Now, we're a considerable operation: 200 full-time staff stretched across two continents. We have five production teams, and the L.A. studio is sprawled across three floors of an urban high-rise. For all of that, though, we still act as a community, and we're always looking for new ways to provide our staff with a supportive social context outside of work. We not only throw huge parties, but also have movie and poker nights, a monthly video game tournament, ongoing ping-pong competitions, season tickets to the Angels and Dodgers, day trips to local attractions, and we even sponsor basketball and soccer teams. To manage all these activities, we've designated a full-time quality of life manager.

These efforts are especially important to us because we recruit from around the world (about a quarter of our staff is international), and many of our employees arrive in a new city not knowing anybody. Over the years, we've found that recreation helps to fuel creation, and the bonds that are formed in casual settings outside the office support the multiple-team dynamic inside the studio.

THE ROAD AHEAD

I can point to three key approaches that have allowed us to achieve our current position. The first is flexibility: managing a balanced production slate that facilitates smooth staff transitions and following an elastic real estate plan. Another is stability: keeping costs down while maintaining a scale that absorbs staff fluctuations. The third is taking care of our people: fuelling our employees with the rewards, resources, and respect that inspire them to do great work.

With the next generation of consoles, we all know the bets are getting larger and risks are getting bigger, but today I'm more confident than ever that we have the people and the approach to stay out ahead of the pack. ❖

Adding a new dimension to mobile gaming has never been this easy.

```
figure = new Figure("/res/myfigure.mbac");  
Texture tex = new Texture("/res/mytexture.bmp", true);  
figure.setTexture(tex);  
...  
action = new ActionTable("/res/walking.mtra");  
...  
figure.setPosture(action, 0, frame);  
frame+=action.getNumFrames(0)/50;  
if(frame>=action.getNumFrames(0)) {  
    frame=0; }try{  
    g3d.bind(g);  
    g3d.renderFigure(figure, 0, 0, layout, effect);  
    g3d.flush();  
}catch(Exception e){}  
  
<<robot.jpg>> <<sphere.jpg>>
```



It's easy to make your 3D models come alive in Sony Ericsson mobile phones with Mascot Capsule Micro3D v3. For tips, tricks, code and case studies, go to www.SonyEricsson.com/developer/java3d.

With mobile plug-ins for Mascot Capsule v3 and JSR-184 (m3g) you can now use your favorite tools, like 3D Studio Max, Maya and Lighwave, to add exciting new dimensions to mobile games. And at Sony Ericsson Developer World you'll find everything you need to get up to speed in mobile Java™ 3D development. From technical documentation and the popular J2ME SDK with support for full 3D emulation and on-device debugging, to responsive tech support.

Once you've realized the possibilities, Sony Ericsson Developer World can assist you all the way from mind to market. No one is more committed to mobile Java 3D than Sony Ericsson, and our unparalleled line of Java 3D technology-enabled phones is just one proof of that.

But Java 3D doesn't just have the power to move the mobile gaming experience into new dimensions, it also has the power to energize your business. So to find out what mobile Java 3D can do for your games and your business, go to www.SonyEricsson.com/developer/java3d – your starting point in mobile Java development.

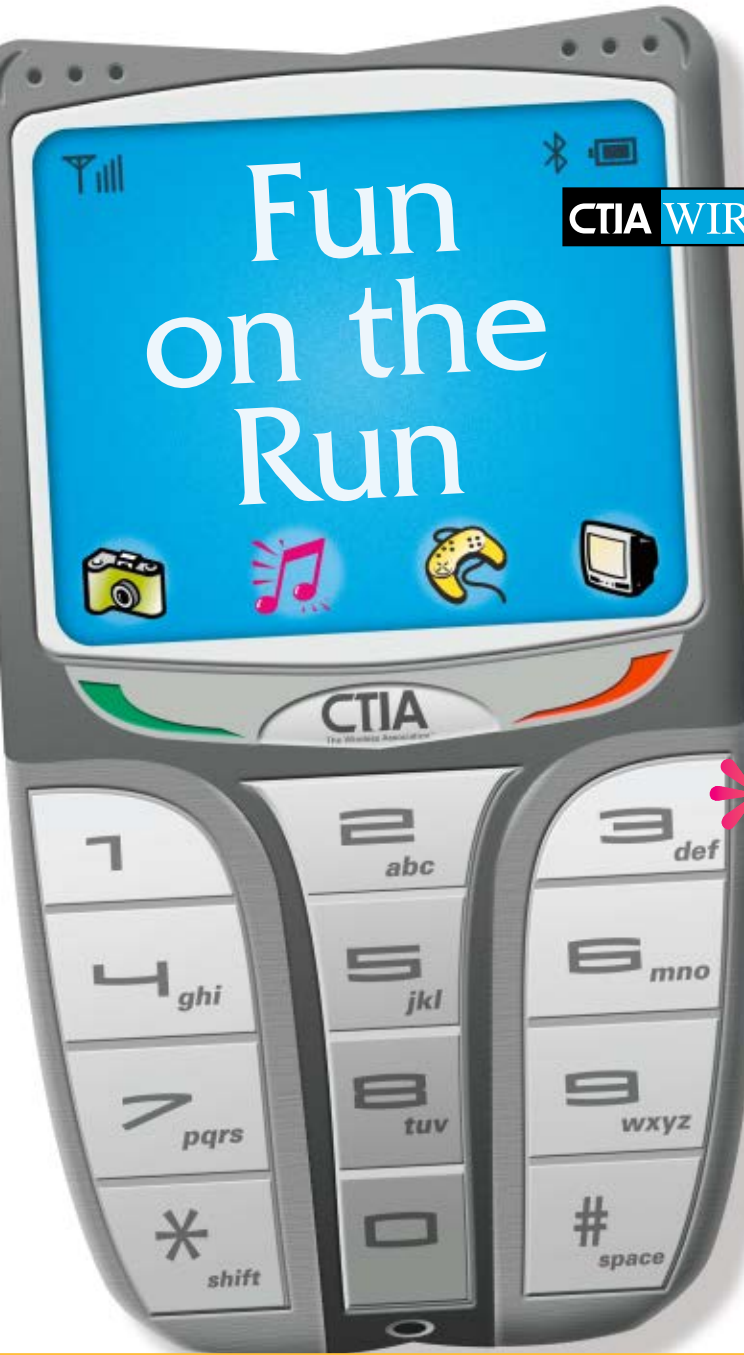


Sony Ericsson

www.SonyEricsson.com/developer

One Show. Two Personalities.

Enterprise | Entertainment



CTIA WIRELESS I.T. & Entertainment 2005

A Department of CTIA—The Wireless Association™

The Next Generation of Wireless Data

With today's competitive pricing and readily available technology—the true differentiators lie in features, enhanced content, and personalization—wireless video, mobile gaming, music, MMS and imaging applications.

CTIA WIRELESS I.T. & ENTERTAINMENT 2005 is the one industry event that effectively joins the wireless industry with the excitement of the entertainment world.

- * Over 200 Exhibitors
- * Dozens of Sessions and Special Interest Seminars on the latest wireless data issues.
- * **MECCA 2005!** Mobile Entertainment, Content, Commerce & Applications, September 26, 2005.
- * Key Networking Opportunities

*Wireless—
keeping entertainment
at your fingertips.*

KEYNOTES



Keynote Host:
Steve Largent
President/CEO
CTIA—The Wireless
Association™



Edgar Bronfman, Jr.
Chairman & CEO
Warner Music Group



Suzan DelBene
Corporate Vice President
of Marketing, Mobile and
Embedded Devices Division
Microsoft Corp.



Rob Glaser
Chairman & CEO
RealNetworks, Inc.



Joel Katz
Co-Managing Shareholder,
Atlanta Office Chair
Greenberg Traurig, LLP



Sean Maloney
Executive Vice President,
General Manager,
Mobility Group
Intel Corporation

*Stay tuned for more exciting
Keynote announcements...*

September 27–29, 2005

Moscone West | San Francisco, CA

www.ctia.org

Register today @ www.ctia.org



HDR-EVOLUTION

With Frame Buffer Objects for OpenGL

» allen sherrod



ALLEN SHERROD has a B.S. in Computer Information Systems. He also authored an upcoming book, *Ultimate Game Programming with DirectX*, to be published by Charles River Media. Allen hosts the game programming web site www.UltimateGameProgramming.com, where he shares demos and their source code to the game programming community. Reach him at asherrod@gdmag.com.



HIGH DYNAMIC RANGE (HDR) RENDERING IS A TECHNIQUE USED TO retain the color precision of a rendered scene as it goes through the rendering pipeline, and has recently been in the news for a number of reasons, not least Valve Software's embracing of the technique for its *HALF-LIFE 2: LOST COAST* expansion, as well as carrying it into some of its other titles. This article will discuss ways to use frame buffer objects for off-screen rendering to help facilitate HDR effects, and gives practical examples of how to do so in terms of OpenGL code.

For applications, especially games, this means that our scenes will be rendered in a more realistic manner in terms of lighting. Using high dynamic range rendering we can add a great deal of detail to our applications by retaining as much light information as possible. This will then cause our objects and surfaces to be displayed in a way that comes closer to resembling real life than ever before.

The problem with non-HDR games is that traditionally, the color precision of a rendered scene is lost, and the rendered display is limited to a low dynamic range of color values between 0 and 255. In the past, this limitation was mainly a result of PC or console hardware only supporting integer buffers, which has a limited range of precision when compared to floating point buffers. Thus, to perform HDR rendering we will need to render our scene to an off-screen floating-point surface, so that the data can be manipulated and made ready to be displayed on the screen.

When we display a HDR rendered scene to the screen, we need to convert it to a low dynamic range since our monitors expect it in that format. What this means for us is that we need a fast and efficient way to render to an off-screen surface multiple times for each frame, and manipulate the data on the fly.

OPENGL'S NEW FRIEND

Although there are many methods that will allow us to perform off-screen rendering, we will be using OpenGL's new frame buffer objects to do the job. In terms of performance, the frame buffer objects are certainly enticing, and since performance is a huge issue in game development, this is a great new addition to the OpenGL API that we have at our disposal.

Frame buffer objects will run on any PC video card and operating system that has driver support for the feature—as long as you have the latest drivers from your hardware manufacturer, then you should be ready to go. For example, Nvidia has now released drivers (v 77.72 or higher) with FBO support, and although ATI drivers do not support FBOs at press time, they will likely do so in the very near future.

So, the purpose of frame buffer objects is to make performing off-screen rendering easier and more efficient in terms of application speed and memory usage. Since frame buffer objects are created and handled by OpenGL internally they are easier to port than pixel buffers (also known as p buffers). These days, more and more applications rely on using off-screen surfaces to perform many

*The next movement in games,
from Tokyo to the world*

World's Largest Game and Entertainment Event

TOKYO **GAMESHOW** 2005

September 16 (Friday) ▶ 18 (Sunday), 2005

Business Day ___ Sept. 16 Open to the Public ___ Sept. 17, 18
Venue ___ Nippon Convention Center (Makuhari Messe)

TGS 2005 will attract not only Japanese game business-related people and the press but also many visitors from other Asian and Western countries. In addition to game consoles and packaged software, various platforms will be exhibited, including online games and mobile-phone games.

Join and visit the "Development Tools and Middleware Area" at Tokyo Game Show 2005!

This special area is for the vendors of development tools and middleware to address the growing need for information on these products. Please visit our website for information.



For further information please visit <http://tgs.cesa.or.jp/english/>
E-mail: tgs@asia-ad.com



LISTING 1

```
void GenFramebuffersEXT(sizei n, uint *framebuffers);
void DeleteFramebuffersEXT(sizei n, const uint *framebuffers);
void BindFramebufferEXT(enum target, uint framebuffer);
boolean IsFramebufferEXT(uint framebuffer);
where
n is the number of objects that are being created or deleted
framebuffers is an unsigned integer variable or array that will
    store the id(s) of the object(s)
target will be GL_FRAMEBUFFER_EXT
```

LISTING 2

The CheckFramebufferStatusEXT() will return one of the following values:

```
GL_FRAMEBUFFER_COMPLETE
GL_FRAMEBUFFER_INCOMPLETE_ATTACHMENT
GL_FRAMEBUFFER_INCOMPLETE_MISSING_ATTACHMENT
GL_FRAMEBUFFER_INCOMPLETE_DUPLICATE_ATTACHMENT
GL_FRAMEBUFFER_INCOMPLETE_DIMENSIONS_EXT
GL_FRAMEBUFFER_INCOMPLETE_FORMATS_EXT
GL_FRAMEBUFFER_INCOMPLETE_DRAW_BUFFER_EXT
GL_FRAMEBUFFER_INCOMPLETE_READ_BUFFER_EXT
GL_FRAMEBUFFER_UNSUPPORTED
GL_FRAMEBUFFER_STATUS_ERROR
```

different effects, which can seriously affect the rendering time of your program.

IF NOT FBOS, THEN WHAT?

Currently there are a few ways of rendering to a texture without FBOs. For one, we can use `glReadPixels()` and `glTexImage()` functions. This method is extremely slow and isn't really recommended. Second, we could use `glCopyTexSubImage2D()`, which is much faster than using `glReadPixels()`. Yet another method we could use would involve setting up and rendering to p buffer objects, which works reasonably well, because rendering to p buffers and creating a texture out of the results is much faster than the previous methods. Another benefit of p buffers over `glReadPixels()` and `glCopyTexSubImage2D()` is that unlike the previous methods, we don't always want our textures to be the width and height of the window; we might want to use a different pixel format, or we might not want to be slowed down by a lot of OpenGL state juggling.

However, p buffers aren't a perfect solution. They suffer from memory issues, and switching between p buffer contexts is expensive. They also require a lot of management, and porting p buffers between operating systems can be a pain.

Although using p buffers can cause a lot of negative issues to arise, there is one strong benefit of using them: You can abstract the code to set up and use a p buffer into a reusable class. This will allow a programmer to write the p buffer code once and then use that same code in multiple projects, instead of having to keep rewriting the same information.

FRAME BUFFER OBJECTS

Now, we have the new OpenGL frame buffer objects, and I can think of at least five reasons why frame buffer objects are a great solution. First, they allow the results of the render to be directly read as a texture. Second, using them requires a single context, which makes switching between frame buffer objects rather fast. Third, frame buffer objects use less memory than p buffers by sharing textures and buffers between objects. Fourth, they work on any hardware that has driver support, and finally, they are easy to set up.

Frame buffer objects use the `EXT_framebuffer_object` extension. Two new objects are introduced with this extension: frame buffers and render buffers. Frame buffers are a collection of buffers (e.g. color, stencil, depth) and render buffers are simple 2D images that store the result of a rendering. When you bind a frame buffer, its attached images are the source and destination of all fragment operations until you unbind it.

SETTING UP FRAME BUFFER OBJECTS

Before you can start coding, you need to make sure you have met all of the requirements. To start, you'll need the `EXT_framebuffer_object` extension.

Creating and destroying frame buffer objects is a lot like creating and destroying textures. Listing 1 shows the necessary functions.

When you bind a frame buffer object using `BindFramebufferEXT()`, all operations will occur on the images attached to the object. Sending 0 to this function will return OpenGL to its default buffer and stops all rendering to the frame buffer object. `GenFramebuffersEXT()` and `DeleteFramebuffersEXT()` work like the functions `glGenTextures()` and `glDeleteTextures()` by allowing you to create or delete a frame buffer object. You can check whether an object is a valid frame buffer object with a call to `IsFramebufferEXT()`.

Textures can be attached to a frame buffer object with a call to `void FramebufferTexture2DEXT(enum target, enum attachment, enum textarget, uint texture, int level)`, where

- the target is `GL_FRAMEBUFFER_EXT`
- the attachment is `GL_COLOR_ATTACHMENT0_EXT...`, `GL_COLOR_ATTACHMENTn_EXT`, `GL_DEPTH_ATTACHMENT_EXT`, or `GL_STENCIL_ATTACHMENT_EXT`
- the textarget (short for texture target) is either `GL_TEXTURE_2D`, `GL_TEXTURE_RECTANGLE`, `GL_TEXTURE_CUBE_MAP_POSITIVE_X`, `GL_TEXTURE_CUBE_MAP_NEGATIVE_X`, or something similar
- the texture is the texture object that was created with a call to `glGenTextures()`
- the level is the mipmap level of the texture that's being attached.

The status of a frame buffer can be checked with a call to `enum CheckFramebufferStatusEXT(enum target)`. This function can only be called after a frame buffer is created and will return one of the values cited in Listing 2.

RENDER BUFFER

Next up is the render buffer. The render buffer API looks similar to the frame buffer and is made up of the functions depicted in Listing 3.

`GenRenderbuffersEXT()` and `DeleteRenderbuffersEXT()` are used to create and delete render buffer objects. `BindRenderbufferEXT()` is used to bind a render buffer, much like



`glBindTexture()` is used to bind a texture object. `IsRenderbufferEXT()` is used to check if the object id that is passed into the function is a valid render buffer object. `RenderbufferStorageEXT()` is used to define the format and dimensions of the render buffer. Last but not least there is `GetRenderbufferParameterivEXT()`, which is used to get a parameter that defines the render buffer. If you wanted to know the width of the render buffer, you would call this function with the parameter `GL_RENDERBUFFER_WIDTH_EXT` passed in to it.

ATTACHING RENDER BUFFERS TO FRAME BUFFERS

The next thing we need to cover is how to attach a render buffer to a frame buffer. This is done with a call to void

LISTING 3

```
void GenRenderbuffersEXT(sizei n, uint *renderbuffers);
void DeleteRenderbuffersEXT(sizei n, const uint *renderbuffers);
void BindRenderbufferEXT(enum target, uint renderbuffer);
boolean IsRenderbufferEXT(uint renderbuffer);
void RenderbufferStorageEXT(enum target, enum internalformat, sizei
    width, sizei height);
void GetRenderbufferParameterivEXT(enum target, enum pname, int
    *params);
where
n is the number of objects you are creating or deleting
renderbuffers is a unsigned integer/array of unsigned integers
target must be the value GL_RENDERBUFFER_EXT
internalformat can be any normal texture formats (e.g. GL_RGB, GL_RGBA,
    etc)
width and height are the buffer's dimensions
pname can be GL_RENDERBUFFER_WIDTH_EXT, GL_RENDERBUFFER_HEIGHT_EXT, or
    GL_RENDERBUFFER_INTERNAL_FORMAT_EXT
params value is based on what pname is. If pname is
    GL_RENDERBUFFER_WIDTH_EXT then after the call params will have the
    width of the buffer's dimension, if it is GL_RENDERBUFFER_HEIGHT_EXT
    then params will have the height, etc.
```

FRAME BUFFER OBJECT TIPS

There are a few things that you should keep in mind when working with OpenGL frame buffer objects.

- For performance reasons, do not create and destroy frame buffer objects in each frame.
- Avoid using functions like `glTexCopy()` and `glCopyTexImage()` when modifying the contents of textures used as rendering destinations.
- If the frame buffer object was not successfully created, then `glBegin()` will generate an error if you try to render to the frame buffer object.
- Frame buffer objects allow for switching between multiple rendering destinations quickly.
- You can use a single frame buffer object with multiple texture attachments.
- You can use a different frame buffer object for every texture you wish to render to.
- You don't need to create a render buffer in order to use frame buffer objects, but you do need a frame buffer in order to use a render buffer.

`FramebufferRenderbufferEXT()` (enum target, enum attachment, enum renderbuffertarget, uint renderbuffer), where

- target must be `GL_FRAMEBUFFER_EXT`
- attachment is `GL_COLOR_ATTACHMENT0_EXT ... GL_COLOR_ATTACHMENTn_EXT`
- renderbuffertarget must be `GL_RENDERBUFFER_EXT`
- renderbuffer is the ID of the render buffer object you're trying to bind to the frame buffer that's currently bound.

Mipmaps can also be generated on all texture images attached to a target using the function void `GenerateMipmapEXT()` (enum target). The target parameter can have a value of either `GL_FRAMEBUFFER_EXT` or `GL_RENDERBUFFER_EXT`.

HIGH DYNAMIC RANGE EXAMPLES

In video games, or in any graphical application, the scenes displayed on the screen are limited to a low dynamic range of luminance values between 0.0 and 1.0. In reality, there's a much higher range of luminance values that our eyes can detect.

Since our monitors can't display images in a higher range of values, we must find a way to take the final output and convert it to the desirable range, which is done using tone mapping.

The basic HDR demos that accompany this article, which are available on www.gdmag.com, work by rendering a scene to an off screen floating point buffer, then tone mapping the results while displaying the scene on the screen.

For one of the demos I use a pixel buffer, while the other uses a frame buffer object. The FBO code is easier to manage and use, while also being compatible with multiple operating systems. The pixel buffer demo will only work on the Windows operating system and is somewhat less elegant than its FBO counterpart. Here, the scene is made up of a simple square that's textured with a HDR image called a light probe.

Once the scene is rendered to the floating point buffer it will need to be tone mapped before it is projected to the screen. Tone mapping maps a HDR image into a low dynamic one so that it can be rendered on the screen without looking like an over-exposed super bright image. The tone mapping technique I've used is a simple one that works by multiplying the color value by an exposure level, which is then reduced by a power function: $\{pow(\text{color}, 0.55)\}$.

The demo applications also use the OpenGL Shading Language for high level vertex and pixel shaders. When you run the application, you can control the exposure level with the up and down arrow keys. Bear in mind that this demo is intended for (and written by) a programmer, so the HDR image isn't highly artistic—but it works. ❌

RESOURCES

For further information on OpenGL frame buffer objects, take a look at a presentation created by Simon Green of Nvidia:

http://download.nvidia.com/developer/presentations/2005/GDC/OpenGL_Day/OpenGL_FrameBuffer_Object.pdf.

You can download high dynamic range texture images from Paul Debevec's web site: www.debevec.org.

You can create your own HDR images using the program HDR Shop, available at www.ict.usc.edu/graphics/HDRShop.

F U T U R E V I S I O N

LONDON
UK

30 AUG
TO
1 SEPT

GDCE
>05

Bring Your Future Vision Into Focus at GDC Europe

The fifth annual Game Developers Conference Europe provides the European game development community an essential forum in which to learn, network, and focus on the future of next-generation interactive entertainment.

GDC Europe offers unparalleled education, including:

- Two days of conference sessions focusing on game development issues of today, and next-gen strategies for tomorrow
- One-day GDC Europe Mobile seminar featuring market-defining sessions and panel discussions
- Three tutorials focused on game design, audio, and Microsoft's next-generation development

ATTENTION GAME DEVELOPER SUBSCRIBERS!

Register by 3 August, 2005 and save up to 30%!

Use Priority Code PAMAXW to activate early rate pricing.

GDC Europe Keynotes



Phil Harrison: A Keynote Q&A on the Future of Games

PHIL HARRISON

Sony Computer Entertainment Europe



Seeing What Sticks: Developing a Katamari Sequel

KEITA TAKAHASHI

Namco

GameDevelopers
Conference Europe

HEAD GAMES



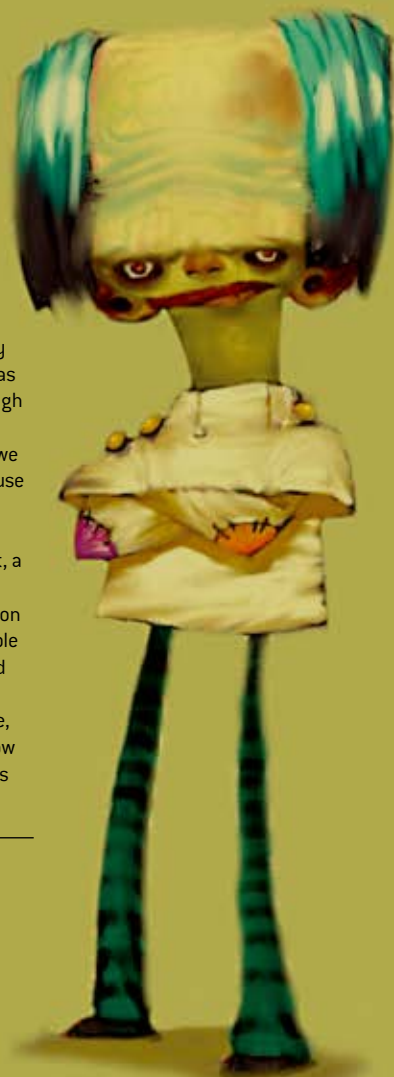
DOUBLE FINE WAS BORN TO DEVELOP

original, genre-defining games based on the imaginative outpourings from the mind of its founder, Tim Schafer—the first of which was *PSYCHONAUTS*. Double Fine did many things right, such as boundlessly tolerating creative risk and exploiting the strengths of the company in its product. But the company also suffered perilous setbacks that threatened its survival. With no precedent to guide us through the problems that arose, we relied on our prior experiences and a collective desire to be successful. Some times, however, our successes during the game's production seemed like they could be defined as "repeatedly snatching victory from the jaws of certain defeat."

In 2001, during the dot-com boom, the only San Francisco work space we could afford was a warehouse on Clara Street. There was a rough and ready start-up vibe to the place; it was really great for parties, and big enough that we could actually drive our cars into the warehouse and park next to our desks.

But the neighborhood was not the safest. Cars were broken into repeatedly. One night, a woman from the transient hotel next door jumped out a fifth floor window and landed on our roof, breaking her leg and knocking a hole in our ceiling. Another day, there was a dead body in the doorway across the street, apparently the victim of an overdose. Inside, there was no heat [space heaters would blow the circuit breakers]. Rats made themselves

CAROLINE ESMURDOC is the executive producer on *PSYCHONAUTS* and the COO of Double Fine Productions. Prior to joining Double Fine, she founded Circus Freak Studios and produced games at Accolade/Infogrames/Atari and Crystal Dynamics. She has shipped numerous titles in her career, including *SUPERMAN: THE MAN OF STEEL*, *LOONEY TUNES RACING*, *TEST DRIVE OFFROAD 3*, and *PANDEMONIUM 1 & 2*. Email her at cesmurdoc@gdmag.com.



Double Fine's Psychonautic Break



GAME DATA



RELEASE DATE

April 2005

PUBLISHER

Majesco Entertainment

GENRE

Third-Person
Action/Adventure/
Platformer

PLATFORMS

Xbox; later ported to PC
(internally), and PS2 (by
Budcat Creations)

PROJECT BUDGET

\$11.8 million

PROJECT LENGTH

4.5 years

LINES OF CODE

Game: 166,781 lines in
381 files in C++.

Game Script: 332,650

lines in 2,433 files in Lua.

Tools: 81,260 lines in 445
files in C++/C#; 11,318
lines in 51 files in Python.

PEAK TEAM SIZE

42 full-time developers,
5 contractors.

HARDWARE USED

Xbox dev kit, Dual Xeon
2.66 GHz, 1GB RAM,
Nvidia GeForce 6800GT

SOFTWARE USED

Maya, Photoshop,
Premiere, Lua, Python,
Emacs, Visual SlickEdit,
XACT authoring tools,
Perforce, Visual
Studio.Net, MoinMoin
Wiki, Bugzilla, Bink Video
SDK, Microsoft Xbox XDK



comfortable in our offices, and even worse, on rainy days the sewers under the office would expel through the latrines, onto the floors, and through the halls. What started as punk-rock charm soon became depressing, disgusting, and dangerous.

By July 2003, office space had become affordable again—outrageously cheap, in fact. So we packed up and moved into our current climate-controlled, industrial, loft-like space.

As if deplorable office conditions weren't enough, we also faced impossible deadlines. One early publisher milestone required that we demonstrate multi-pass effects before the renderer was completed. In another case, it was only after a milestone had been submitted that we learned of content that was required before the delivery would be approved and a payment released. Eventually, our schedule began to slip as well.

Just prior to our office move, we amended our publishing agreement to move out the ship date.

The new contract stipulated that within three months we hire a producer and develop a build of the game that demonstrated the fun factor of the finished product—or risk cancellation. I joined the team as executive producer in the middle of this trial in the summer of 2003.

With new management in place, and everyone focused on one game section for three intense do-or-die months, the Black Velvetopia level emerged as one of the most innovative expressions of the *PSYCHONAUTS* gameplay experience. It was well received by our publisher who renewed its green-light decision. We spent the next several months developing multiple levels of the game concurrently at an unprecedented pace.

In February 2004, at what seemed to be our peak productivity, a time when we felt most confident about shipping on schedule, Microsoft decided to discontinue its development of *PSYCHONAUTS*. Microsoft had funded years of mistakes, course corrections, and learning curves, but it drew the line at underwriting the remaining game development now that Double Fine was finally on track. When Ed Fries departed Microsoft, the new management seemed to think that we were expensive and late. The assessment was accurate, though it did not reflect the progress we were finally making toward shipping the game and recovering the development investment.

It took all of our savings, careful money management, and a little help from our friends to survive the cancellation. We continued to work hard on milestone builds, though we had no publisher to submit them to. Tim and I focused on securing new funding.

PSYCHONAUTS was met with resistance from some publishers and faced internal political struggles in getting green-lighted by others. It was a demoralizing time, compounded by the stress of being completely honest with the team while still motivating them to continue to meet scheduled deadlines. After several trying months, and with our coffers running dry, we prepared the team for the worst. Though it was hard

for them to hold out hope, they continued to toil. Our determination finally paid off. In July 2004, Majesco offered us a publishing deal.

The new publishing terms meant foregoing additional planned hires without the benefit of scaling back the design. The convergence of these factors led to the most insane crunch I have ever witnessed. We all worked ourselves beyond what was reasonable and humane—yet the team remained loyal and steadfast. In March 2005, *PSYCHONAUTS* went gold. We had managed to dodge a hundred bullets without compromising the quality of the game, losing ownership of the company, or missing a single day of payroll. Through a series of setbacks and disappointments that would have decimated other groups, the Double Fine crew displayed an unshakable spirit, resulting in the creation of one of the most cohesive teams I've ever seen. Solidarity like that is not something that can be recruited, but only forged in fire. It is because of them (and their patient, tolerant, and supportive families) that I can write this postmortem.

WHAT WENT RIGHT

1 STRONG GAME VISION AND UNCOMPROMISING QUALITY.

Since the company's inception, Tim had an idea to make a psychic action/adventure game whose levels were located in a character's mind, locales where the surreal visuals would immerse players in the mental state and back story of that character.

In his inimitable style, Tim crafted a storyline that weaved together the relationships between a collection of psychic children and their camp leaders with the minds of the misfits, monsters, and madmen that held the clues to saving the world from total annihilation. The environments were fantastical, the characters were memorable, the gameplay was inspired. The use of psychic powers as the tools by which the player progressed in the world was an innovative and uncontrived scheme in these settings. Each piece of the high concept fit together to make a cohesive whole, which survived intact through the project's entire development.

An oft-uttered mantra at Double Fine is "God is in the details." *PSYCHONAUTS* is a shining example of a game that got the details



[Other competitive solutions require multiple supporting packages to handle our assets. With **Alienbrain**, everything we need is in one package.]

Christopher Bretz Art Director, Secret Level

**Starts
at
\$690**



Brainpower

Avid Alienbrain Studio 7.1 has the intelligence to help your team work faster and smarter.

Its intuitive interface tells you more than the what, where and who of an asset. It tells you what an asset references, what it's referenced by, and whether it's buildable, it's optimized, or it's redundant. It knows what platform an asset is for and what should happen to it next. It even knows an asset's location in a texture map, its resolution in level three, and how much damage it inflicts when eaten. Alienbrain knows. You know. Your pipeline knows.

ASSET MANAGEMENT FOR CREATIVE TEAMS

Witness the power of **Avid Alienbrain Studio 7.1** at SIGGRAPH 05:
Avid Computer Graphics Booth #1401.

Free eval copies available at www.alienbrain.com



www.alienbrain.com

Avid
computer graphics

right. So many design ideas that at first seemed like they'd be insignificant to the player—or elements that would be easy to cut if time ran out, or other things not worth the performance or memory hit—turned out to be the features that make *PSYCHONAUTS* so appealing and memorable. Each detail presents itself as a beautiful little discovery, and collectively, they make the game much deeper. Details are one of the hallmarks of a Tim Schafer title, and *PSYCHONAUTS* is no exception.

2 RASM. Early in development, a strike team called RASM was formed. RASM stands for Raz Action Status Meeting, but eventually it meant something more concrete: a concentrated collection of team members tasked with ensuring that the main character's core movements and actions felt exactly right.

RASM was successful because of the composition of the strike team and the frequency with which the meetings were held. The cross-functional group included at least one participant from each discipline on the team. At RASM, the designer described how he would use the action element in the level. The animator discussed how to exaggerate the character's movement. The programmer demonstrated new functionality and tweaked the implementation in response to group feedback at the meeting. A test level containing each of the action elements was created to assess the look and feel of each movement. Each bi-weekly meeting was dedicated to one action element, with some movements requiring multiple discussions. Over the course of development, Raz's full complement of core movements emerged. The feel of the main character is important in any game genre, but is especially important in a platformer. The RASM team did character movement in *PSYCHONAUTS* especially right.

3 TOOLS, TOOLS, TOOLS. Three noteworthy tools had profound effect on the development of *PSYCHONAUTS*.

Dougie and the Debug Interface. We chose to use an off-the-shelf scripting environment to write much of the high-level game code, wanting fast feedback without having to compile the game and an easy debug interface where we could enter commands and inspect in-game object states. We selected Lua for its small memory footprint, fast performance, and flexible environment that allowed us to add features such as class inheritance and cooperative multi-threading without digging deep into the language runtime. We wrote a remote native debugger, Dougie (named after a neighborhood homeless conspiracy theorist we befriended), to be able to inspect and use the features we wrote on top of the Lua language. In addition to traditional debugging features (e.g. break points, single stepping, and stepping over functions), we added object watch windows, profiling tools, hot script reloading, custom scripting buttons, and a command line console interface to the game. We wrote the platform-agnostic Debug Interface to standardize the user interface and facilitate the extraction of debug information, allowing automated control of the game by other proprietary tools and the flexibility to embed connections to it in other third-party tools, such as Maya, Python, and Emacs. It was fortuitous that we developed such powerful tools even before we knew just how much of the game we would be custom crafting in Lua.

Automated Build Process. Our automated build process (ABP) made a build of all SKUs of the game along with some of our tools. The ABP released automated builds at least daily, freeing the programmers from having to build and post versions of the



The Double Fine Productions team.

game for the team regularly. It ran a test build in a clean environment based only on code that was checked into source control, provided immediate feedback to the programmers about changes included in the build, and reported any compile errors that resulted from the build (which were fixed immediately). The ABP saved countless hours of programmer time, especially as the team size expanded.

Cutscene Editor. *PSYCHONAUTS* has a colossal number of in-game cutscenes. Prior to the creation of this tool, the Gameplay Programmers (GPPs) would hand-craft each of the cutscenes. Needing a less cumbersome approach, we created the Cutscene Editor, which made scene creation much easier. The Cutscene Editor displayed a visual timeline, divided into columns of dialog lines and setup/cleanup sections, as well as rows of "actors" for each scene. In each actor, you could place an action at a specific time to play animations, place and orient a camera or actor, set actor properties, or even call Lua functions. Placement information could be read directly from the game running on the Xbox, and the cutscene could be previewed at any time. The Cutscene Editor rightfully put control of the scripting in the hands of the programmers and put control of the cinematography in the hands of the animators—and it saved countless hours of work.

4 ART DIRECTION AND HUMOR. *PSYCHONAUTS* reviews consistently praise the art direction and the humor, which is a gratifying reflection of the priorities and strengths of the company. Tim recruited artist Scott Campbell after seeing his work at an art show. Scott's drawings had a subtle and understated cartoony look, something never seen before in games. Scott drew hundreds of loose, 2D sketches for *PSYCHONAUTS*, which our modelers turned into beautiful 3D geometry.

Many factors contribute to the success of the humor in *PSYCHONAUTS*—the art, the animations, the voice acting and, of course, the dialog. Tim wrote most of the dialog in the game



himself, but enlisted the help of Erik Wolpaw (www.OldManMurray.com) for much of the script. The collaboration between the two was so successful that it's nearly impossible to tell which of them wrote any one of the more than 8,000 lines of dialogue.

The script was brought to life by dozens of extremely talented voice actors. Animators hand-crafted scores of animations for the characters, conveying the humor of the dialog in the expressiveness of their movements. It's rewarding to read players' reactions to the characters and dialog in online forums,

some even using game characters as avatars and game quotes as signature files. To us, that means we did the art and humor right.

5 HIRING SMART: GAMEPLAY PROGRAMMERS AND THE TEST DEPARTMENT.

Originally, we staffed a team of level designers to script the game events, believing the scripting burden was simply a matter of placing some triggers. We soon learned that to make the unique cinematic experiences come to life, much more scripting was required. It wasn't long before the level designers spent entire days bogged down in Lua. Though tech-savvy, they were not programmers. The code they generated was complex, buggy, and ultimately unusable. So, eight gameplay programmers were hired to rewrite the entire Lua side of the game from scratch. Staffed with experienced industry programmers and fresh college graduates from Computer Science departments, this group was one of our most critical (and overworked) ones, contributing profoundly to the design and quality of the game.

We founded an internal test department to shift the burden of stabilization away from the development team. Due to the highly dynamic codebase and limited reuse of scripted elements, continual regression was necessary to ensure that new features did not break existing code.

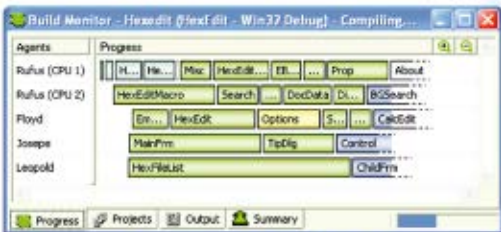
The ever-increasing game size, however, made it unreasonable for us to play through the entire game to test



Slow C/C++ Builds?

IncrediBuild accelerates C/C++ builds by distributing compilation tasks across the network, cutting down build time by 90% and more!

Download
FREE, Fully Functional
30-Day Trial!



- Simple setup, requires no changes in code or settings
- Fully integrated with Visual Studio 6.0 and .NET IDE
- Supports Win32 and Xbox builds

XOREAX

www.xoreax.com

LIFE STUDIO: HEAD® 2.7 SDK

MULTIPLATFORM SDK SOLUTION NOW AVAILABLE!

LEADING FACIAL ANIMATION TECHNOLOGY
AT THE CUTTING EDGE!

To find out more about our licensing policy, please contact sales@lifemi.com.

- Run-time control using API
- Automated lip-synch
- SDK supports PC PlayStation® 2 and Xbox®

Visit
Lifemode Interactive
at booth 2263
at Siggraph!



Lifemode Interactive
www.facialanimation.com
www.lifemi.com



"PlayStation" is a registered trademark of Sony Computer Entertainment Inc.
"Xbox" is a registered trademark of Microsoft Corp.
"LifeStudio:HEAD" is a registered trademark of Lifemode Interactive Ltd

each change prior to check-in. We needed ongoing testing with up-to-the-minute assets but had no funds with which to hire a test team in light of our cancellation.

Undeterred, we put out a call for volunteers on our web site. And they came, friends and strangers, all willing to commit unpaid hours to the PSYCHONAUTS testing cause. The immediate and lasting benefits of an internal volunteer test team were so positive that we hired a full-time test department as soon as we signed with Majesco. Though few developers house an internal test team, we never felt having one was an indulgence. The test efforts led to consistently stable builds and greatly influenced our shipping product.

WHAT WENT WRONG

1 WHO OWNS THE LEVELS? Since no one wanted to compromise gameplay or visuals, we developed a level-sharing system. A level designer would design the world, lay down action paths, and script game events, and an artist would build additional world mesh around that design. This process failed miserably.

Over time, we decided that only artists would create visual geometry, causing resentment among the level designers. Complicating matters and heightening tensions, the level designers and artists both worked in the same software and tools, causing work to be overwritten and leaving levels in unworkable states. Consequently, the levels created were not quite up to par, and they could never exceed a first-pass implementation state.

With no producer on staff, Tim's demanding corporate responsibilities left him little time to handle the emerging problem effectively. Working independently, the level designers produced concepts that Tim ultimately rejected, causing a rift between them. In time, they stopped communicating effectively.

In 2003, on the bloodiest day in company history, the level design department was put to rest, and all but one of the designers were let go. Unaware that the situation between the groups had gotten to this point, the abrupt departure of the designers left the team shaken. The sole remaining designer, Erik Robson, was made lead and was put in charge of the artists in a newly created World Builder department responsible for both the design and the visuals, leaving the scripting to the gameplay programmers. Regardless of the upside that eventually resulted from the reorganization, the level ownership issue was not handled gracefully as it unfolded, and the messy departure of the level designers remains one of the biggest blemishes in Double Fine's history.

2 DIFFERING DEVELOPMENT STRATEGIES LEADING TO SCHEDULE OVERRUNS. PSYCHONAUTS' scheduling problems began early in development. During pre-production, our publisher requested volumes of documentation and a fun, polished unit of gameplay that would provide immediate payoff. However, the documents and demos were primarily created for the purposes of shepherding the project through the green-light



process at the publisher. Neither one facilitated our creative process, nor did either help us understand the game at a deeper level ourselves. Pre-production should have been able to accomplish both goals: solidify the design and technology, and mitigate any outstanding risks through the creation of a vertical slice of gameplay.

Years later, after our cancellation and in our last few months of development, we finally created a rough, full game walkthrough. From it, we learned that we needed to rework fundamental areas of the game. Our fragmented understanding of the global game design and miscommunication with our publisher about what the game really needed led to schedule overruns by wide margins. The game slipped twice—in mid 2002 by six months, and in early 2003 by 16 months. It wasn't until we prioritized the interactive walkthrough and global feature design that a realistic schedule could be created, though it remained heavily desire-driven.

3 A CRUSHING CRUNCH. A game design document was initially created, but the design continued to grow over time. Those evolved specs were memorialized only in email threads, loosely collected documents, notebooks, napkins, and



whiteboards. As company business took up more of Tim's time, and without a producer to ensure task reallocation, the game design document quickly became stale. Regardless, launching level construction was essential, so we devised a schedule wherein we would design and build somewhere between two and five levels to a playable state every eight weeks.

Rework was usually required to bring earlier levels into compliance with global design elements that were developed



INDEPENDENT
GAMES FESTIVAL

REWARDING INNOVATION IN INDEPENDENT GAMES

THE 8TH ANNUAL

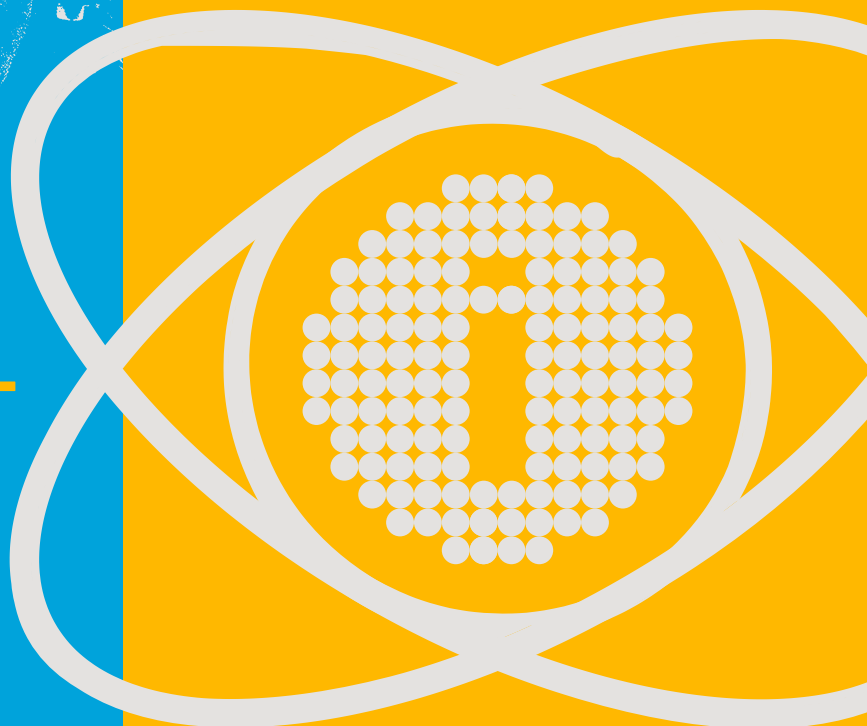
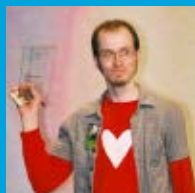
INDEPENDENT GAMES FESTIVAL

MARCH 20-24, 2006

SAN JOSE, CA

WWW.IGF.COM

The stage is being set for the eighth annual Independent Games Festival—the most respected event celebrating the indie game developer. Here's your chance to see if you have what it takes to bring home an award in 2006!



OPEN CALL FOR ENTRIES

IGF COMPETITION

DEADLINE FOR SUBMISSIONS: SEPTEMBER 6, 2005

Open to all independent game development teams, the IGF Competition offers over \$35,000 in cash and prizes.

IGF MODDING COMPETITION

DEADLINE FOR SUBMISSIONS: OCTOBER, 2005

New Competition! Non-commercial mod teams compete in one of four categories for a chance to take home a \$2500 cash prize.

IGF STUDENT COMPETITION

DEADLINE FOR SUBMISSIONS: NOVEMBER 15, 2005

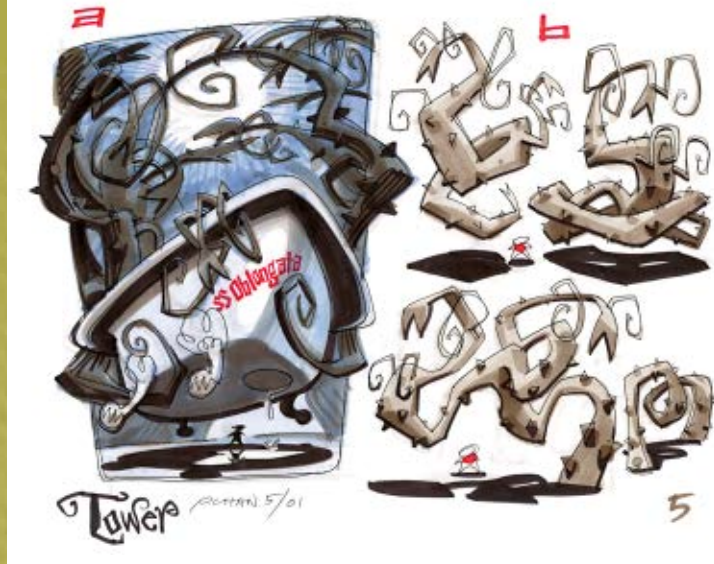
A competition for games developed by full-time students.

FOR INFORMATION ON CATEGORY DESCRIPTIONS,
ENTRY FORMS, AND ELIGIBILITY RULES GO TO

WWW.IGF.COM



CMP
United Business Media



later. Additionally, I opted to maintain previously developed levels alongside the creation of new levels, increasing our per milestone load. It wasn't long before milestone Fridays lasted all night and into the weekend. Developing a PC version ourselves and supporting the development of a PlayStation 2 version on the same schedule [expertly

handled by our talented friends at Budcat Creations] complicated the already impossibly tight schedule.

The floating design, the failure to cut content, schedule underestimation, additional SKUs, and an immovable final deadline caused build requirements to pile up faster than team members were able to service them, resulting in massive overtime. The team was forced into a multi-month crunch to complete the game by the ship date, on the heels of over a year of aggressive development. Though the game was technically in development for four-and-a-half years, it was actually developed in less than two. We learned some painful lessons as a result, but now, we place a high value on process, constraints, rapid iteration, and agile development practices to home in on the essential fun factor of the high concept as early as possible.

4 CREATIVE DIRECTOR BOTTLENECK. For several years, in addition to designing and writing, Tim was president, producer, office manager, human resources, CFO, COO, and webmaster. He was slow to staff these positions because he felt the people in those roles would be inventing the corporate culture, and he wanted that culture to be something special. Viewing those responsibilities as too important to delegate, he tried to do it all himself, at a great cost to the game's early development.

This single team member over-tasking created a tight bottleneck for multiple disciplines. Tim's contributions were required for continued progress in the game design, art and animation approvals, and programming feature specs, yet he would start to miss meetings to deal with one emergency or another. Eventually, an associate producer handled the HR and office management responsibilities, but Tim was still doing too much: recruiting, budgeting, scheduling, managing the publisher, in addition to directing game development. He was simply stretched too thin without a producer. After three desperate years in this schizophrenic role, Tim hired me to manage the project and the business operations. As soon as he did, *PSYCHONAUTS* was back on track.

5 LARGE TEAM MANAGEMENT. While our leads were very senior, few had significant hands-on experience managing and growing a large group of people. As is a common but often detrimental practice, we made our most senior team members the leads on the project. As a result, their valuable direct contributions to the game were diminished by the time they spent managing ever-growing teams.

To add to their leadership challenges, some leads managed largely green teams, resulting in critical workflow chokepoints. This learning curve took its toll in the project schedule. The leads had difficulty breaking down the loosely-defined scope of work into constituent parts and ascribing reasonable time estimates to those tasks. As the development pace quickened near the end of the project, the leads collectively took on more tasks, at the expense of their management. Because of the lean budget at the end of the project, additional production resources could not be hired to alleviate the scheduling and tracking burden on the leads.

The trial-by-fire ultimately made the leads stronger, but the stress took its toll, and many of those individuals elected to take scaled back or non-leadership positions moving forward.

PSYCHO, YES, BUT STILL DOUBLE FINE

Shipping *Double Fine's* inaugural game was an exercise in fierce determination, passion, and perseverance. By a purely Machiavellian standard, we were resoundingly successful. The result is a beautiful and fun interactive experience published on multiple platforms to a unanimously appreciative reception by the press and fans.

By any other metric, we had a rough time of it. We learned much from our experience on *PSYCHONAUTS*—most importantly, to never give up. Even when we lost our publisher, or when we ran years longer than expected, or when we had to navigate around sewage to get to our desks, we never gave up.

Double Fine had its share of growing pains, but we got the chance to express ourselves creatively in ways we never had before and develop a new company culture of our own. We have become more mature, cohesive, and smarter as a company—one that can't wait to apply its hard-learned lessons to its next project. ::





STEVE THEODORE

PIXEL PUSHER

DRY EYES

Building eyes for animation

THE EYES, POETS TELL US, ARE WINDOWS to the soul. You'd have a hard time telling that from computer games, though. It's no accident that the best introduction to facial animation is the title *Stop Staring* [by Jason Osipa, Sybex, 2003]. Up until now, audiences have tolerated the wall-eyed gaze of game characters, but with high quality CG animation on display everywhere and rapid advances in rendering quality, the age of the 1000 yard stare is coming to an end. So this month we're going to consider the construction and behavior of the human eye. This might be a slightly dry topic, but it's a critical foundation for bringing characters to life in animation.

UNDERSTANDING THE SHAPE OF THE EYE

Eyes are complex forms. They're much harder to reconstruct from photo reference than noses, chins, or eyebrows. Eyes are also the most mobile part of the face, so you can't model the eye properly without thinking in advance about how the eyelids will animate. This sounds daunting, but eyes can actually be handled pretty easily if you think about them as a working system rather than as static shapes.

Understanding how eyelids work is simple if you consider them in just two dimensions before worrying about the 3D reality. Everyone knows the classic almond shape that represents the eye in 2D. But that shape isn't static. It's created as the muscles in the eyelids work to keep the lids clear of the pupil as the eye roves around. Fear or surprise will widen the eyes, allowing white to show above or below the iris. Anger and concentration narrow the eyes, pushing the lids right up to the margin of the pupil itself. At all

times, though, the curve of the eyes is the simplest curve that will keep the pupil from being obscured by the lids [see Figure 1].

Ironically, this is one place computer graphics work almost like real life. If you imagine the upper and lower lids of the eyes as NURBS curves, you can quickly see how easily the different shapes of the eyelids can be created by simply pushing the NURBS CVs up or down as the eye moves about (see Figure 2).

Since the shape of the eye is so fluid, a real-time modeler should begin building the eye geometry with animation in mind at all times, rather than trying to capture a single expression perfectly. If you're working with a bone-based setup, it's usually best to model the eyes closed and rely on the animation to establish the distinctive shape of the almond—you'll have more room for textures on the closed lids, and you won't have to worry about the animation causing the underlying eyeballs to poke through. On the other hand, if you're using vertex animation you may want to model in a half-opened position to minimize vertex interpolation problems. In either case, clean geometry with a regular tessellation is essential for controlling the animation of the lids, so plan accordingly and don't just start throwing verts and edges around.

DOMINO MASK METHOD

Some modelers get control of the tessellation by starting with a flat 2D pattern of quads and wrapping it around the eyeball as if it were an eyemask being laid across the face.

One advantage of the 2D method is that it demonstrates how simple the anatomy of the eye really is. There are only two important features to account for. At the inner corner of the eye is the tear trough, a small extension of the eye area which shows the reddish tissue of the tear sac, which helps drain tears from the eye. At the outer corner of the eye, you may want the upper lid to slightly overlap the lower. While not everyone's eyes actually



FIGURE 1 As the human expression changes, and as the eyes rove around, various amounts of white and iris are visible, but the lids always keep the pupil unobstructed.

STEVE THEODORE started animating on a text-only mainframe renderer and then moved on to work on games such as *HALF-LIFE* and *COUNTER-STRIKE*. He can be reached at stheodore@gdmag.com.

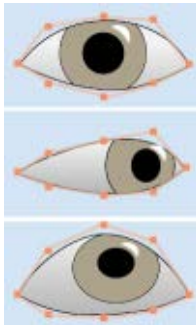


FIGURE 2 Simulating the changing shape of the eye with simple NURBS curves makes the basic mechanics much easier to see. Note how all these shapes are achieved with only vertical CV movements. In a real 3D eye rig, the same effect could be achieved with one-axis bones.

show this folding effect, adding it does help prevent “peek through” at the corner when the eye closes.

Naturally, you’ll first need to model the eyeball itself before you can wrap the geometry over it. The eyeball is mostly hidden by the orbital ridge of the skull, so it can be hard to gauge its size. The rule of thumb is that the diameter of the entire eyeball is a bit more than twice that of the iris. Purists will note that eyeballs are not, in fact, spheres. The cornea produces a distinct bulge, which is plainly visible when viewed from the side. The iris underneath the cornea is actually cut back into the sphere of the eyeball, as you can see from Figure 3. For most game applications, though, a spherical piece of geometry is good enough—and it simplifies animation enormously. A handy compromise is to use a normal map with a flattened iris for the diffuse lighting and a second normal map with a bulging

cornea for the specular lighting, which will give almost the right effect without the inconvenient geometry issues.

When wrapping the skin to the eyeballs, it’s important to capture the thickness and plasticity of the lids themselves. The facial feature of many ethnic groups from cold climates (where the fat acts as a useful insulator) generally don’t express much of the eyeball sphere, presenting eye sockets that seem very shallow and smooth. This is sometimes thought of as a north-Asian feature, but it’s common among Eskimos and in other cold regions as well. [Björk (from Iceland) and Renée Zellweger (part Laplander) are said to have Asian-seeming eyes.] Older or heavier faces will tend to have drooping skin above the eyelids, making the brows seem lower and thicker. And some people have exaggerated tear glands above the other corner of the eye or fat bulges under the lower lid.

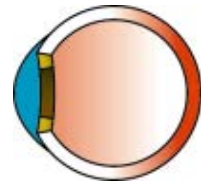
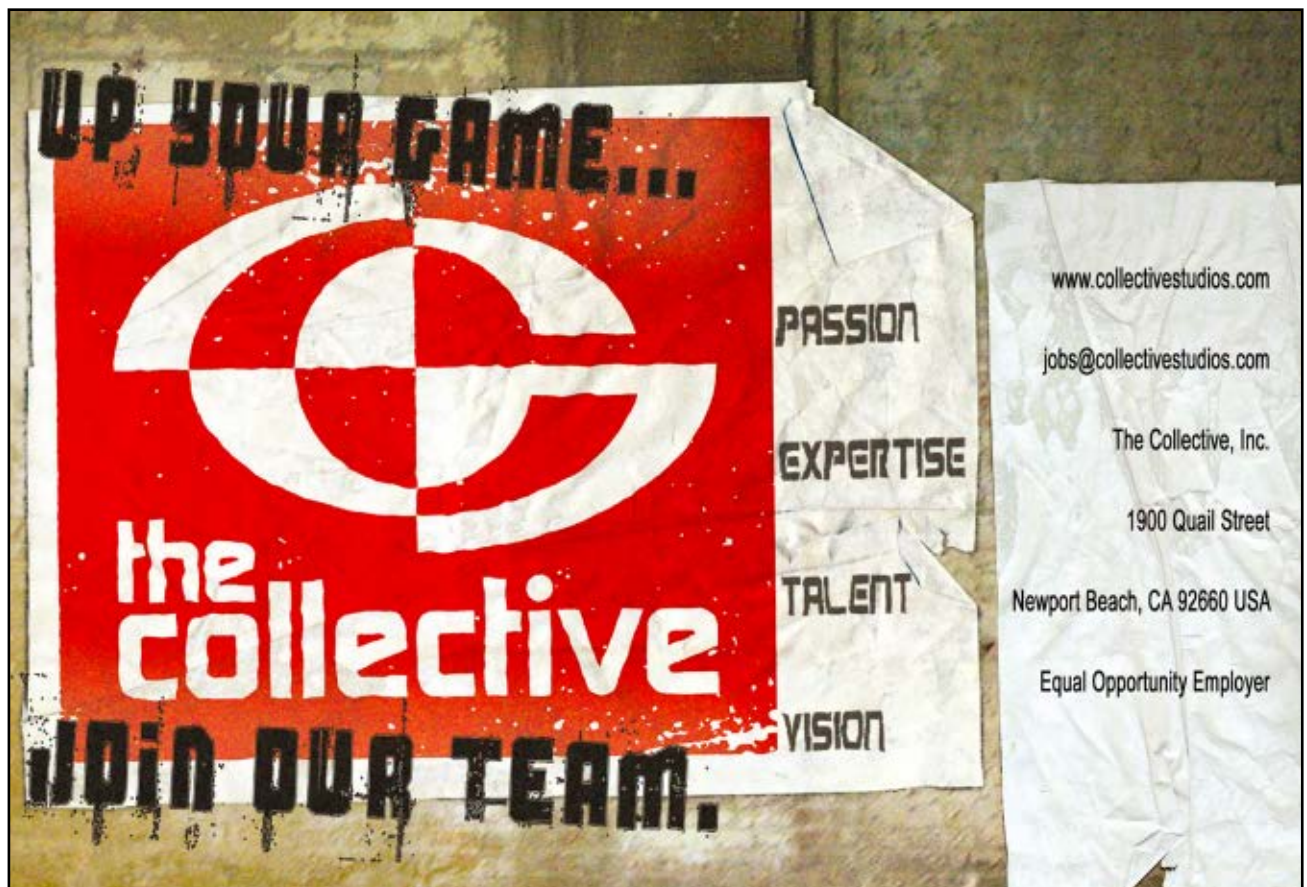


FIGURE 3 The eye “ball” actually shows a distinct bulge over the cornea. In closeups, you can also notice that the iris is slightly flattened.

ANIMATING THE LIDS

Now that the geometry is in place, we need to create an animating structure that will create the shape of the eye in motion. We’ll also need to capture the movement of the lids as the eye glances around to reflect emotion, such as eyes widened in terror or narrowed in suspicion. Finally, we’ll need to be able to key blinks and closures, either by hand or programmatically. To set up these



-  EDUCATION
-  GOVERNMENT
-  HEALTH
-  MILITARY
-  CORPORATE
-  FIRST RESPONDERS
-  SCIENCE

SERIOUS GAMES SUMMIT

WASHINGTON D.C.
OCT 31-NOV 1, 2005 **D.C.**

HAZMAT: HOTZONE - Entertainment Technology
Center at Carnegie Mellon University.

REGISTER BY
SEPTEMBER 21, 2005 AT
WWW.SERIOUSGAMESSUMMIT.COM
AND SAVE \$200 ON YOUR PASS

Use Priority Code PEMAXW to activate early rate pricing.

THE NUMBER OF NON-ENTERTAINMENT games under development is rapidly increasing and demand for the ideas, skills and techniques used in commercial entertainment games is at an all time high. As a result, an entirely new market has emerged.

Serious Games cover the intersection of games, learning, policy and management. Major corporations, government and military institutions, foundations, educators, and non-profits are turning to games and

commercial entertainment technologies as a new approach to simulations, training, education, and other practical applications.

The Serious Games Summit D.C. is the only conference 100% dedicated to this emerging field of non-entertainment games and gives customers, developers, and evangelizers of serious games the opportunity to learn from successful serious games applications and forge links with the traditional videogame industry.



These lids are deformed with a simple six-bone setup and animated with driven keys. Note also the fake corneal bulge, simulated with a normal map on simple spherical eye geometry.

actions, there are two distinct options: vertex animation (“morph targets”) and bone animation.

VERTEX ANIMATION

As always with vertex morphs, it's important to know the technical limitations of your system. Movements like the opening of an eye, which are basically rotational in nature, aren't ideally suited for many types of vertex animation.

Make sure you set things up in such a way that you won't ever have to worry about an eyeball potentially poking right through a half-closed lid—whether by code wizardry, extra morph targets, or simply thicker eyelids.

The first job of eye animation is simply to bring the movement of the eyes to life. You'll need to create morph targets for looking left, right, up, down, and straight ahead, reproducing the eyelid behavior we spelled out above, too. If resources permit, add the corners (up-left, down-right, and so on) because they will work to produce better overall results. Once you've got the morph targets for look-directions, you'll need to drive them using the eyeball's rotation; hand-keying the directional morph targets for every shift in gaze is a nightmare. You'll get best results if you can track the direction of the eyes and drive morph targets for the left and right eyes independently. Otherwise, with only simple targets for “eyes right” and so on, you'll find that your character can't change his or her focal distance.

Look-direction is basically an automatic feature, but you'll also want to hand-key some basic eye behaviors. You'll need a “closed” target for blinks, of course, and “wide-surprised” and “narrow-angry” targets to capture other emotional expressions. A “scrunched” target, when

the eyes are closed and compressed by raised cheeks, might be useful if you need to convey intense pain or powerful laughter.

BONE-BASED ANIMATION

Bone-based systems can also create convincing eye animation, but the organizational hassles can be slightly daunting. Creating the variable curve of the eyelids will take at least three bones per lid (six per eye) and careful vertex weighting. Larger numbers of bones will produce better results as well as more management overhead. Luckily, the techniques driving the bones are pretty simple, and once they're set up, they won't require much personal attention. Each bone should rotate only around the horizontal axis of the eye. They don't need to move or scale. Be careful to name them clearly because you'll be selecting them from an outline view. Getting at the overlapped bones in the viewport is nearly impossible.

Weighting the vertices is pretty simple. You may find it useful to feather the vertex weights horizontally, allowing each bone to influence the verts of the next bone over just a little. This setup helps create a rubbery effect that makes it easier to maintain the curve of the eyelid. The vertices along the edge of the lids should be weighted heavily to their corresponding bones, until the effect fades out about two-thirds of the way to the solid structures of the brows, cheeks, and nose. For eyes with epicanthic folds, the vertices below the fold should be completely weighted to their respective bones. For fatter lids or those without folds, get as little movement as you can while preserving flexibility of the eye shape. Usually the eyes with fatter lids should have fewer vertices surrounding the eye to facilitate this simplified movement.

One potential problem with bone-based eye animation is the shading on the opened upper lids. We don't mind if the geometry of the raised lid disappears into the head a bit (after all, it's being folded up), but this can cause shading artifacts on visible areas as the polygons bend double. If no amount of fiddling with weights gives an acceptable compromise, it may be necessary to add even more bones. You can try adding rotational fix-up

bones to lessen the rotation of the middle vertices while allowing the verts on the edge of the lid to move as freely as needed.

Actually driving the bones is the easiest part of the setup. The most common method is to use a set of driven keys (“reactor controllers” in Max) powered by the movement of the eye. You'll also want to use sliders or animate-able parameters to represent the wide-afraid, narrow-angry, and closed-blink poses, using them to drive the bone positions just as you might blend morph targets. Scrunch poses, though, are harder to achieve with bone systems. Bones can't easily simulate the complex wrinkling behavior that makes a satisfying scrunch. Raising the cheeks and lowering the brows over closed eyes will still convey a good deal of emotion, however.

EYE CANDY

The result of all these considerations ought to be a set of animating eyes that works on the reflex level and can be keyed to express a few simple expressions. As the eyes glance around the scene, the lids should behave naturally with little or no hand keying. If you need to constantly adjust the eye shape by hand, the system isn't working. It will become an intolerable burden to the animator if it isn't basically automatic, so be sure to stress test it before putting it to use.

Even with just these basic features, you can achieve a great deal. It's amazing how much punch you can add to simple animations with just a slight widening or narrowing of the eyes at the right moment. Compare a very basic shot with moving eyes, flexible lids, and some simple punctuation in the form of blinks and squints to an old-school shot with fixed eyes or mechanical, shutter-like eyelids, and you'll see that all your work has been rewarded.

Of course, all this is merely laying the foundation for the really interesting parts of eye animation: focus, subliminal emotions, and expressive acting, which we'll discuss in more detail in the near future. Until then, keep your eyes open! ❖



THE INNER PRODUCT

SCANNING LARGE DIRECTORIES

IN THE PAST YEARS, HARD DRIVES HAVE gotten phenomenally large, but without proportional increases in speed. That additional size allows for both larger files and more files. Unfortunately, the performance of each of these relies on different properties of the hard drive. Reading large files efficiently requires high throughput. Reading many smaller files, as we needed to for my February 2005 article on late-binding data for games, requires fast seek time. Of these two, it seems discussion focuses more on large files; for example, the most common performance metric for drive speed is RPM. Although the disk rotation speed also has a secondary impact on random seeking, it primarily affects throughput. As a result, modern, large hard drives seem to cope better with increased file sizes than with increased directory counts.

At the same time, I tend to be a curmudgeonly stick-in-the-mud when it comes to using software; I prefer to use the software I've already learned how to use rather than switch to newer software with more bells and whistles but a different user interface, and, often, slower performance. For example, in Windows, until a couple of weeks ago I'd always used the simple "My Computer" version of the explore window, which only shows a single directory. Lately I've decided to join the 21st century by using the full-blown "explore" version of the window, which includes a separate pane with a complete, expandable directory tree (see Figure 1).

As soon as I started using it, though, I discovered some unfortunate performance issues. Each plus-marked box next to a folder in the tree is expandable because it has subdirectories. Figure 2 shows the results of expanding two of the nodes in turn. What it doesn't show is that expanding the second node, the one labeled "1-40 dirs," took more than two

seconds, during which time the window did not respond to input. I repeated the test on two other drives, getting the results shown in Table 1 (pg. 44).

Why is this operation so slow? To display the tree in the folder pane, Windows has to determine whether each newly-visible directory (folder) should have a plus sign or not. This requires knowing whether each of these folders has any subdirectories. Apparently, doing this requires Windows to iterate through all the files in each directory and check whether any are subdirectories, since the file system doesn't maintain this fact directly.

In this test case, the 40 directories contain about 7,500 files, all of which must be scanned.

What's puzzling is not just that the newer drives aren't proportionately faster than the old drive, but that they're slower for the same sized data. I haven't controlled all the variables in this scenario; the drives have different degrees of fragmentation, are different ages, and different vendors. I don't know if NTFS is at fault, or if the drives are screwy or something else; whatever the truth is, these are the hard drives I have, and I have to live with and use them. I do know that a version of this on a non-synthetic test takes more than five seconds to expand a folder which recursively contains about 10,000 files total.

When dealing with large numbers of files, this performance problem manifests itself in many other places besides Windows Explorer. Trying to recursively find a file or compute the recursive total size of a folder can be excruciatingly slow, since they don't just look in the next subdirectory as in the above situation, but visit every file. Finding all duplicate files on a drive takes impossibly long since it must not only scan every directory recursively but compare files of the same length—and as the number of files increase without similar increases in file length, the number of non-identical files that happen to have the same length also increases.

Of course, Windows will cache file "metadata," and successive search attempts may be faster. However, the data doesn't tend to stay cached for long if you're actually doing much else, like running a game.

In February's column ["Late-Binding Data"] I

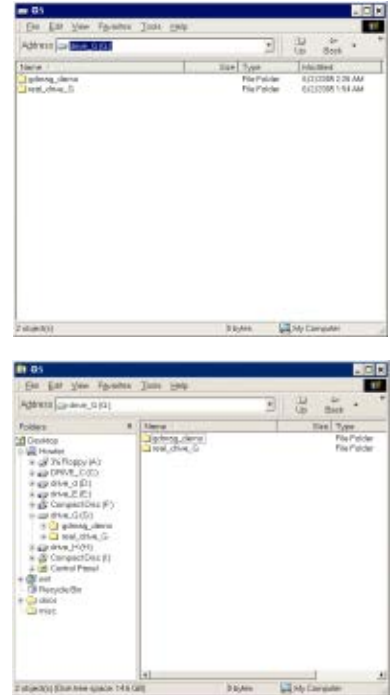


FIGURE 1 Two methods of file browsing in Windows are shown.

proposed a scheme for handling both raw game data files and data files packed into WAD-like files, by fully scanning all the relevant raw data file directories during startup. Although this was reasonable with a moderate number of files on my C: drive, I've discovered it's entirely unacceptable on my other hard drives with a larger number of files, at least by my personal standards for start-up time.

An easy fix to the problem is to not scan the directories at start up. Instead, when a piece of data needs to be loaded and isn't cached, you can attempt to find it in all the directories where it might be. This isn't unreasonable for loading a large piece of data, but for a small file, the multiple seeks required for checking the directories can be more expensive than loading the data itself. In my case, I also use my data management system to access procedurally-generated content, which requires first trying to load the data off of disk before falling back to the procedural generator. Adding extra disk accesses

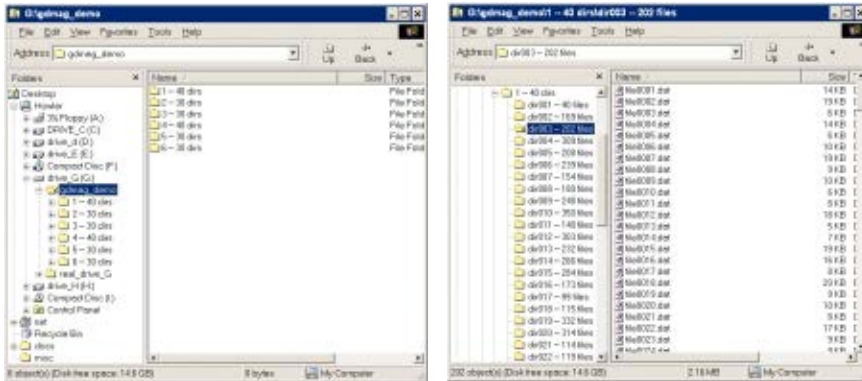


FIGURE 2 Clicking on the “expansion” nodes (marked with plus signs) expands the tree. Sometimes this requires a surprising amount of work behind the scenes; see text.

is a significant overhead in that case.

Instead, I decided to confront the problem head-on and see if I could speed up directory searching. If so, it might prove useful for other tasks, like searching or computing recursive sizes.

RECURSIVE SCANNING

To start with, I created the synthetic directory tree shown in the previous figures. It consists of a top level directory with six subdirectories. Each subdirectory has 30 or 40 subdirectories with 40 to 400 files, for a total of 40,000 files. My synthetic test program recursively traverses the directory and counts how many files and directories it encounters, as well as the total size of all those files. After reporting the time spent, it then simulates a real application’s workload (so as to flush the disk cache) by reading a 120MB file from disk and then allocating and writing to 350MB of memory (on a machine with 512MB of physical memory). The lines marked “naive” in Table 2 show the performance of the recursive scan phase of this program.

I tried several things to improve performance. Although only one of them worked, I’m going to describe them all to save you the trouble of trying them yourself. My first step was to make sure that my basic recursive program wasn’t being pessimal. A naively-written recursive scan appears in pseudo-code in Listing 1.

The potential problem with this code is that it starts scanning a directory, then recurses into any subdirectories before finishing scanning that directory. Depending on how the underlying find-next-file operation works, this might mean the operating system doesn’t bother loading the entire directory information into memory before the recursion; later, when it returns, it will have to seek

back to the directory to finish reading it. This could happen repeatedly during a single scan as the search goes down and back from a given directory.

Instead, you can completely scan each directory, then recurse into each of the subdirectories you found on the first pass. This is not the same as breadth-first versus depth-first traversal; both implementations are depth-first in how they explore the tree. To use terminology normally used for binary tree traversals, it’s “pre-order” rather than “in-order.”

In my experiments, this didn’t seem to make much difference; however, my synthetic test bed doesn’t have any directories with many files and some subdirectories, so most likely the directories that did trigger recursion were small enough to fit in a single cluster. I also didn’t see any difference from using the portable `_findfirst/_findnext` API versus the Windows `FindFirstFile/FindNextFile` API. Another technique I experimented with was using Windows’ `FindFirstFileEx` API with the `FindExSearchLimitToDirectories` search option to see if the file system supported finding all the directories without scanning the files.

Unfortunately, this seems to never have been implemented, and newer Windows documentation doesn’t even mention the option.

CACHING DIRECTORY SCANS

Since the problem is that the operating system is doing a poor job of caching the metadata I’m interested in, the only real solution left would be to cache it myself. One approach to this would be to cache it in memory: leave a process running which keeps track of the current directory structure, updating it by using `ReadDirectoryChangesW`. This might be the best possible solution, but it’s complicated and introduces strange interdependencies.

The only other alternative strategy seems to be caching the metadata on disk so it can be shared between multiple runs of a program. Each time a program needs to scan a directory tree, it can load the cached directory-tree meta-information off of disk. This might seem entirely redundant to the file system’s storage of the meta-data, and in fact it is. The difference is that the file system is designed to satisfying many different functions and clients, and hasn’t been tuned to the particular use to which I’m putting it. A cache of the directories and filenames for my synthetic test tree is only 500K, which is roughly just the sum of the string storage required to store the 40,000 filenames—and Windows file systems can load 500K from a single disk file just fine.

The tricky part of using this kind of cache is that the contents of the directories may have changed since the last time the program was run. Obviously a cache of wrong information doesn’t solve the problem at all, so we need to determine which information is wrong and adjust it—and we need to do this without scanning the entire directory tree looking for changes.

Fortunately, the file system stores a “last write time” or “last modified time” for each file and directory. In the case of directories, this value is updated whenever the contents of the directory changes, in the sense of a new file being created or an existing file being deleted or renamed. The

Table 1

DRIVE	SIZE (GB)	FILESYSTEM	TIME TO EXPAND FOLDER (S)
C:	40	FAT32	0.1
G:	120	NTFS	2.0
H:	200	NTFS	2.5

Time to expand the second folder shown in Figure 2.

Table 2

DRIVE	SCAN METHOD	TIME (S)
C:	naive	1.6
	post-order	1.5
	caching	0.14 ± 0.01
G:	naive	14.8 ± 1.0
	post-order	14.9 ± 0.1
	caching	0.16 ± 0.03

Time to scan a tree with 40,000 files.

LISTING 1 A simple recursive directory scanner

```
count-files dir:
  count = 0
  info = find-first-file(dir)
  while not-done(info)
    if is-directory(info)
      count += count-files(info.fullpath)
    else
      count += 1
    endif
    find-next-file(info)
  endwhile
  return count
end
```

directory's time is not updated if file contents change, e.g if a file is extended or altered.

One odd behavior is that the directory's time isn't always immediately updated. Sometimes Windows waits up to 10 seconds before changing the directory last-modified time. This is unfortunate and can cause a program using a cache to run with slightly stale information, but the performance gain is sufficient, so I don't mind this. I've been unable to track down the exact circumstances; for a while I thought it was just when you create a new subdirectory, but the behavior does not actually seem consistent. Fortunately, it is not the common case.

By storing the previous last-modified time for each directory in the cached metadata, we can then validate the cache by comparing all the last-modified times. The `_stat()` function call lets us query each directory's last-modified time directly, without scanning all the directories. We then rescan each changed directory, ignoring any cached information about it. Of course, we can only `_stat()` the directories we already know about because they're in the cache, but any new directories that aren't in the cache will cause their parent's last-modified time to have updated, so we'll find them when we scan the parent.

RESULTS AND LIMITATIONS

The caching line of Table 2 shows the performance of this algorithm. As you can see, the cache gives a huge speedup to my slow hard drive, nearly 100 times, and a smaller but still significant speedup to the slow hard drive, about 10 times. In both cases, the time has been reduced from an amount which is noticeable even for application startup

time to an amount which would be tolerable for interaction, e.g. in response to a mouse click.

However, the table only shows the performance for the case where the cache is an exact match for the actual tree. The first time this scanner runs, there is no cache, and it takes as long as the others. But otherwise it is quite effective. If new files are added, the caching algorithm tends to take roughly the time required to scan only those new files—typically a small fraction of the total. However, what actually matters is the number of changed directories; if you add just one file to every directory in a tree, the time will be identical to the full-scanners. And, of course, if all the files are in a single directory, there is no savings if just one file is added.

Because of the rules for when a directory's last-modified time changes, the only thing this cache can guarantee is that it has the names of all the files in the tree. It can't keep track of all the files' sizes or last-modified times; it appears the only way to do that is to fully scan the directory. This means this technique isn't useful for things like measuring the size of a directory's recursive contents.

It's also possible for directory last-modified times to be incorrect. For example, programs can set them directly; a program to restore a backup might do this. Generally, it's not a problem. Even if you had a strange unzip program that preserved the original file-modification times for directories, as long as you were creating a new top-

level directory, the caching system would still detect it and scan. Only if you did this over an existing directory such that it added new files to a subdirectory would this go wrong. Moreover, I'm not aware of any unzip-style program that does this. But it is something to be aware of.

I've created a 500-line C library that does this caching. You pass it a directory path to scan and the name of a cache file (which needn't exist yet). The library returns a complete list of all the files with the directory path as a prefix, and writes out an updated cache to the specified file. Since it doesn't provide file sizes or last-modified times for files, it may not be that useful, but if you're interested my web site <http://silverspace-ship.com/inner/> has the complete package. ❄





NOAH FALSTEIN

» GAME SHUI

THE RIGHT TO BARE ARMS

SOME TIME AROUND 2003, I REMEMBER

an unnamed Electronic Arts executive was rumored to have said, "The latest JOHN MADDEN FOOTBALL has the best graphics yet! It's the first in the series where you can actually see John Madden's arm hair!" ... which reminds me of one of the

perennial arguments in the game industry: Which matters more: good graphics or good gameplay?

The glib answer is "both," but in the spirit of The 400 Project, I think it's worth exploring some of the tradeoffs between the two. Certainly,

pursuing both high-end graphics and innovative gameplay is a great goal, but often it's not feasible. Sometimes it's not even desirable. I doubt there were many MADDEN fans hoping for an arm-hair close-up, and it seems likely that the resources devoted to that facet of the game would have been better utilized elsewhere. But where?

LICENSE LOYALTY

Let's look specifically at the role of graphics and gameplay regarding licensed properties. If a company has paid a lot of money to license a character, movie, book, or sports franchise, then it should logically support that license. In fact, economics has shown that a company paying for an expensive license is not in a good position to experiment with radically new gameplay that can

alienate casual players drawn to the game because of the licensed IP. Paying for a hot, exclusive license and supporting it with great production values while staying conservative with the gameplay has been the formula behind many top 10 hits.

But supporting a license doesn't necessarily mean spending a fortune on graphics. The key principles to creating a good licensed title are 1) be faithful to the essence of what makes the license popular, and 2) translate that essence into the domain of the target medium. Sometimes the essence has a strong visual component, particularly if the license is for a hit special-effects film. But, for example, if it's a *South Park* license, staying true to the visual component would mean using a limited, 2D approach.

SMART PLAY, SIMPLE DESIGN

Sometimes graphics are the right aspect of a game to invest in. From a financial viewpoint, great graphics sell, particularly when attached to a popular license so that there's already an interest in the subject and built-in marketing advantages.

Historically, there are many examples of games that looked great but were only moderately fun at best and which somehow managed to sell millions of units. On the other hand, games with just decent graphics that spurn a license in favor of fresh concepts and great gameplay often have languished, or are favored mostly by developers. I'm thinking specifically of *ICO* and *KATAMARI DAMACY* here. But the very ground-breaking nature of these games helps provide the innovation and fresh ideas that are an essential part of the lifeblood of the industry. If all games simply depended on licensing content and using derivative gameplay, we would bore and alienate our audience.

EARL WEAVER

Sometimes the essence of a license has more to do with character. Even before the MADDEN franchise at EA, the company

had a successful game called *EARL WEAVER BASEBALL*, which licensed the likeness of the famous Orioles manager. I was at a press conference and got to see Weaver play a demo of the game. The graphics were Commodore 64-era 2D animations—you couldn't make out even the numbers on the players' jerseys, and there was no question of animating arm hair, for you could hardly see the player's arms.

When the in-game umpire called one of Weaver's players out, I was amused to see how much Weaver had gotten into the game, despite the puny graphics. He started yelling at it right there in the press room. Then, on the screen, a tiny Earl Weaver walked onto the field and started yelling at the umpire, kicking dirt on him. The developers had clearly captured the essence of his personality in the game, which was a big hit.

For a counter-example of a fun game with minimalist graphics that sold millions of units, we need to look to the casual game downloadable market for games like *SNOOD* or *BEJEWELLED*, or all the way back to the uber-counter-example game *TETRIS*.

Oddly enough, the exception proves the rule. Apparently, only in the fairly limited subset of games that are designed to stimulate our joy of discovering abstract patterns can we get away with simple graphics and still do well. But, the simple purity of these games makes it hard to come up with lots of compelling variations.

GRAPHICS SHOWDOWN

It's counterproductive to think in terms of a fight between gameplay and graphics. Rather, developers should strive to work on both gameplay and graphics to find the best available balance for the game's subject, genre, platform, and possible license that delivers the best experience for the player.

And if you found the title of this column entertaining, perhaps in a future column I'll discuss the fabled magical ritual, *Incantum Ursus Belli*, known colloquially as the Rite to Bear Armies. ❖



Even with graphics, sometimes less is more. (JOHN MADDEN FOOTBALL '93.)

NOAH FALSTEIN is a 25-year veteran of the game industry. His web site, www.theinspiracy.com, has a description of The 400 Project, the basis for these columns. Also at that site is a list of the game design rules collected so far, and tips on how to use them. Email him at nfalstein@gdmag.com.



ALEXANDER BRANDON

✦ AURAL FIXATION

FUN AUDIO DESIGN

SOUND TO PICTURE. THAT'S WHAT GAME audio is, right? I'll respond to that by laughing maniacally.

All too often, sound in a game waits until something visual is created and follows the visuals as closely as possible. In most genres, this is a perfectly reasonable thing to do, but sometimes you have to think outside the box and doing so can create some truly enjoyable audio design experiences. Here are just a few ways to bring more abstract techniques into the mix for generating unique and effective sounds, voiceover, and music for a variety of projects.

AUDIO+DESIGN

That audio can influence game design is something not many designers think about, unless the overall game design itself warrants it (such as *REZ*, *PARAPPA THE RAPPER*, *DANCE DANCE REVOLUTION*, *GITAROO-MAN*, *SPACE CHANNEL 5*, *SAMBA DE AMIGO*, *DONKEY KONGA*, and *TAIKO DRUM MASTER*). Still, here are some excellent examples of games in which audio is becoming a more integral part of gameplay, such as *THIEF* and *ASHERON'S CALL 2: FALLEN KINGS*.



While not a looker, *THIEF* didn't just set the bar for stealth-based sound. It broke new ground all over for audio use in games.

In the "Horn of Quintus" level in the original *THIEF*, the player had to find an artifact by listening to it. An ancient horn would regularly make a deep, whale-like sound, and the player would follow the sound in order to reach it, rather than work from visual or textual clues. Game audio designer Eric Brosius guided the player through a level by placing a 3D sound at the end of a

level and giving it a huge radius as well as a decent falloff curve, so that it could ramp up slowly as the player got nearer, giving the proper amount of location based feedback.

In *ASHERON'S CALL 2: FALLEN KINGS*, Jason Booth, the creative director for this long-standing MMORPG, took audio into his own hands and created a design mechanic that allowed players to play musical instruments in the game world, which allowed for the creation of bands for the first time in a persistent fantasy setting.

EXPERIMENTATION

When you think of the theme music from *Star Wars*, you don't picture a panoramic vista of the old west. However, the theme music could actually fit such a visual. We associate certain types of music with visual themes often out of habit, but at times it can work to break this mold. Sometimes it's successful, sometimes not, but it's something developers can take more chances with than outrageous graphics.

VASTEEL. This Human Entertainment-developed, Working Designs-published turn-based strategy game from 1990 brings to mind lofty orchestral scores or heavy electronica, as it is set in a futuristic, robotic universe. However, the developer chose to go with ... jazz? Not just jazz, but jazz that's one step away from muzak? Granted, jazz isn't exactly what comes to mind, but when dealing with turn based strategy action can certainly grate on the ears. In this case, the developers chose a different but pleasantly acceptable alternative, though I'm not sure everyone would agree.

TIGER WOODS PGA TOUR 2005. I had blasted the 2004 version of this game for its motley collection of inappropriate licensed songs, but along with reviewers and players alike, I stand very satisfied with the current version of the game. Legend has it that EA's Steve Schnur approached electronic music composer BT for the soundtrack of TW2005 and said "I want you to make the sound of



Licensed music wasn't the answer for *TIGER WOODS PGA TOUR 2005*; BT scored a soundtrack for it that fits like a glove.

golf." And he certainly did, but not with a string quartet—with ambient techno. Granted, this might not be what we usually picture for golf, but it puts a great new spin on the game that fits mightily appropriately.

VOICEOVER MAGIC

Voiceover is the most underutilized element in games nowadays. With union difficulties, limited stream capabilities in the now aging generation of consoles, and a history of awfully managed game dialogue, who can blame people for being timid about voiceover? However, using resources well (and having enough of them), you can direct player action in a way that is more to the point than any other game asset—except, of course, those huge button combo overlays you see in *GOD OF WAR*.

Player feedback. Next gen is the answer. Even more streams for Xbox 360 and cell-based processing on PlayStation 3 (with *Revolution* specs forthcoming), voices can potentially spew tons of information to the player in any number of genres. Imagine sports announcers having a unique line for every player action? Imagine the player character muttering or shouting unique lines when under attack to lend personality to each battle?

More drama. A real-time strategy battle where zooming in reveals hordes of unique shouts, commands, grunts, and death vocalizations. No longer just little ants you have at your command, are they?

THE FUTURE IS NOW!

There are many places to go for audio, but game audio is now in a position to break out of simply attaching itself to each animation or each object where appropriate. There are many techniques to differentiate yourself from the pack, and you'll probably enjoy brainstorming how to do it! ✦

ALEXANDER BRANDON has been involved with game audio since 1994 and is currently the audio manager at Midway in San Diego, Calif. You can email him at abrandon@gdmag.com.

ACTIVISION.
activision.com

THIS IS YOUR GAME

- Programmers
- Game Designers
- Scripters
- Producers
- Artists
- Art Directors
- Animators
- Sound Engineers
- Marketing
- Creative Services
- Sales
- IT

Please visit us at the Siggraph Job Fair at Booth 45

ACTIVISION.
VALUUS PUBLISHING, INC.

NEVERSOFT

Infinity Ward

Vicarious
Visions®



Luxoflux



SHABA



ZAXIS



FOR A COMPLETE LISTING OF OPENINGS AND TO APPLY,
VISIT WWW.ACTIVISION.COM (GO TO "CAREERS").



GET IN THE GAME™

ELECTRONIC ARTS IS LOOKING FOR CREATIVE TALENT TO JOIN OUR AWARD-WINNING TEAM!

THE WORLD'S NUMBER ONE INTERACTIVE ENTERTAINMENT COMPANY is hiring for a wide range of positions. Electronic Arts (EA) develops, publishes, and distributes software worldwide for video game systems, personal computers, and the Internet. Our world-class studios create massively popular products that we market under the EA SPORTS, EA GAMES, and EA SPORTS BIG brands. EA remains committed to hiring and retaining the industry's best talent. Get in the game, and help us shape the future of entertainment.

POSITIONS AVAILABLE IN:

- Art
- Audio
- Distribution
- Engineering
- Finance
- Game Design
- Game Testing
- Human Resources
- IT
- Marketing
- Production
- Sales

LOCATIONS:

- Asia
- Chicago
- Japan
- Los Angeles
- Louisville
- Orlando
- San Francisco Bay Area
- Montreal
- UK
- Vancouver

Great People Make Great Games!

VISIT US AT:

jobs.ea.com



Developing Multiple Titles
On Next-Gen Consoles



NOW HIRING
the Best of the Best!

Apply at: www.infinityward.com

and see us at SIGGRAPH'05
Job Fair Booth #45

Programmers Designers Artists Producers

Headquartered in Chicago with studios in San Diego, CA, Seattle, WA, Austin, TX, Los Angeles, CA & Chicago, IL and offices in Munich, Germany and London, UK, Midway Games Inc. is a leading developer and publisher of interactive entertainment software for all major video game systems.

Midway has released some of the most popular games in the video game industry for play on all major video game platforms including PS2, Xbox and GameCube. These games have entertained millions of players around the world.

We are on the lookout for the best and brightest. Our goal is to create the most compelling, high-quality games around. We're always searching for world-class talent with energy, commitment to excellence and unrestricted imaginations to join our exceptional teams.



VISIT OUR BOOTH ON THE EXHIBITION FLOOR AT SIGGRAPH AUGUST 2nd-4th TO LEARN MORE ABOUT OPPORTUNITIES AT MIDWAY.

WE HAVE IMMEDIATE OPENINGS FOR:

- Software Engineers
- Programmers
- Lead Programmers
- Technical Directors
- Animators
- Lead Animators
- Lead Modelers
- Environment Artists
- Concept Artists
- User Interface Artists
- Level Designers
- Game Designers
- Producers
- QA Supervisors
- Art Directors
- Studio Creative Directors

We have openings on many different game teams and genres including RPG, Action Adventure, Racing, Horror, Sports, etc.

For more detailed information on available positions at Midway, please visit our website:

<http://jobs.midway.com>

Artists: Providing samples of your work is essential. Please first apply with your resume online, then send sample art/demo reel with a breakdown sheet detailing your contributions to the Midway location you applied to.

Midway Locations:

- Chicago, IL
- San Diego, CA
- Los Angeles, CA
- Seattle, WA
- Austin, TX
- Dallas, TX
- London, UK
- Munich, Germany

Benefit Highlights

- Competitive Salaries
- Competitive Bonus/Incentive Programs
- Medical/Dental/Vision/LTD/Life
- 401(k)
- Vacation/Holidays
- Referral Bonus Program



Midway Games Inc. encourages diversity and is an Equal Opportunity Employer.

MYTHIC ENTERTAINMENT®

BUILDING THE FUTURE OF ONLINE GAMING



NOW HIRING

STUDIO:

3D CHARACTER ANIMATORS
CHARACTER ARTISTS
CONCEPT ARTISTS
TERRAIN ARTISTS
UI/2D ARTISTS
EFFECTS ARTISTS
ENVIRONMENT ARTISTS
CLIENT PROGRAMMERS
GRAPHICS PROGRAMMERS
SERVER PROGRAMMERS
CONTENT DESIGNERS
CONTENT DEVELOPERS
CONTENT DEVELOPER - WRITER
IN-GAME SUPPORT REPRESENTATIVES

PUBLISHING:

PRODUCT MARKETING MANAGER
PUBLIC RELATIONS MANAGER

WWW.MYTHICENTERTAINMENT.COM/JOBS

© 2005 Mythic Entertainment, Inc. All rights reserved. Mythic Entertainment, the Mythic Entertainment logo, "Imperator," the Imperator logo, the phoenix symbol, "Dark Age of Camelot," the Dark Age of Camelot logo and the stylized Celtic knot are trademarks of Mythic Entertainment, Inc., in the U.S. and/or other countries. "Warhammer," the Warhammer logo, and the hammer and shield symbol are trademarks or registered trademarks of Games Workshop, Inc. All rights reserved.



Rockstar Games is actively seeking experts in the following fields:



■ **Rockstar North**

Animators
Character Artists
Environment Artists
Vehicle Artists
Game Programmers
Level Designers

■ **Rockstar San Diego**

Graphics Programmer
Networking Programmer
AI Programmer
Tools Programmer
Concept Artist

■ **Rockstar Vienna**

Graphics Programmer
Animation Programmer
AI Programmer

■ **Rockstar Vancouver**

Art Director

■ **Rockstar NYC**

Motion Capture Specialists

■ **Rockstar Toronto**

AI Programmer

■ **Rockstar Leeds**

no current openings

Rockstar Games

622 Broadway, New York, NY 10012

Att: Recruiting

Email: resumes@rockstargames.com

Fax: 212-334-6644

www.rockstargames.com

Please indicate the studio and position you are applying for in the subject heading.



CAROL

» NOW HIRING PEOPLE WITH TECHNICAL AND ARTISTIC VISION

rainbowstudios.com

relic.com

volition-inc.com

crankypantsgames.com

heavy-iron.com

helixe.com

locomotivegames.com

concretegames.com

thq.com

bluetongue.com



THQ, Rainbow Studios, Relic Entertainment, Volition, Inc., Cranky Pants Games, Heavy Iron Studios, Helixe, Locomotive Games, Concrete Games, THQ Studio Australia, Blue Tongue and their respective logos are trademarks and/or registered trademarks of THQ Inc. and/or their respective owners. All rights reserved.



THQ Studios... there's a place for you.

THE SECRET IS OUT!

Our first title, Titan Quest, was recently announced at E3 to glowing reviews! But we still have a few select opportunities left for talented individuals to get in on the action.

Our company, Iron Lore Entertainment, was founded by Brian Sullivan (co-creator of the award winning and best selling Age of Empires® series of games), to develop blockbuster games that expertly blend the best of action, RPG and adventure genres.

We have a commitment to excellence in all aspects of development. To achieve this high standard requires the focused efforts of top talent in a fun, productive environment.

We have openings for **Artists, Designers** and **Programmers** who can really make things happen - people with the desire and talent to create fun, top-quality blockbuster games. If this describes you, contact us at:

jobs@ironlore.com

IRONLORE™

WWW.IRONLORE.COM

Age of Empires is a registered trademark of Microsoft Corporation © 1999 – 2002



CHANGE THE WORLD...

Sony Computer Entertainment Europe is the leading developer and publisher of PlayStation®2 and PSP™ (PlayStation®Portable) delivering hits today whilst preparing for the launch of PlayStation®3

There are currently a number of opportunities to join one of their BAFTA award winning studios in London, Cambridge or Liverpool, developing some of the most technically and creatively advanced games in the business while breaking sales records on the leading platform.

A number of representatives from SCEE will be attending SIGGRAPH and would like to meet with individuals who are interested in a career at our internal studios.



LONDON STUDIO
△ ○ × □



We are currently looking for

- Lead Artists
- VFX Artist/Animators
- Technical Artists
- Senior Environment Artists
- Senior Animators
- Animation Directors
- Animation Programmers
- Tools Programmers
- 3D/Graphics Programmers



datascope
RECRUITMENT

109/110 Bolsover Street,
London W1W 5NT
Tel: 020 7580 6018
mailto: sceedev@datascope.co.uk
www.datascope.co.uk/scee

Datascope, the games industry specialists, have been solely appointed to manage all of SCEE's development recruitment.

SIGGRAPH2005 

To find out more please send your resume to paul@datascope.co.uk or visit us at our stand in the SIGGRAPH Job Fair (booth 24)

™ "PS" and "Playstation" are registered trademarks of Sony Computer Entertainment Inc.

® is a registered trademark of Sony Corporation. All Rights Reserved. All Images and game logos are copyright of SCEE

CREATIVE CAREERS

BUILD THE NEXT ELDER SCROLLS® AND FALLOUT® GAMES

Game Producer

Quest Designer

Character Artist

Special FX Artist

Level Designer

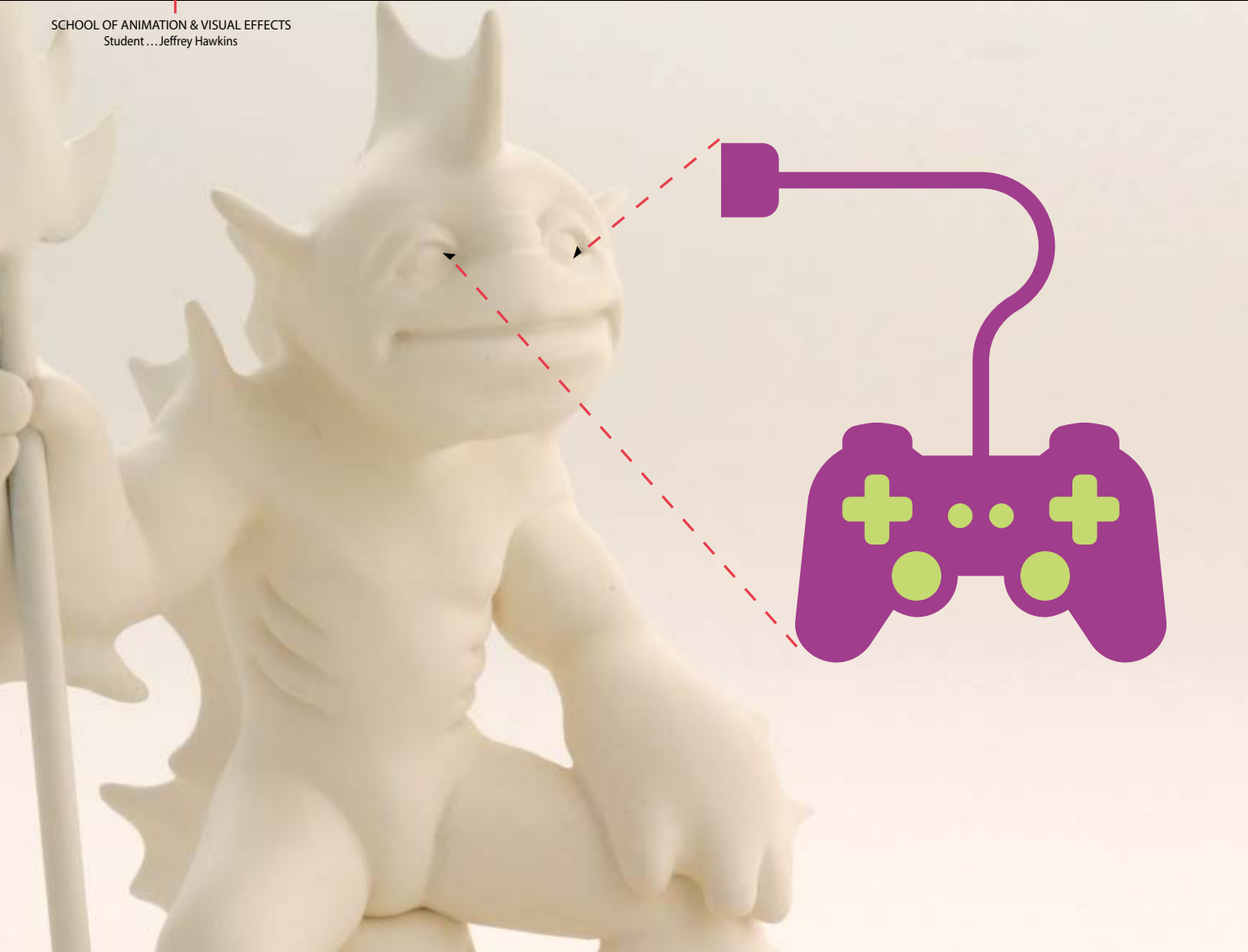
Game Programmer

For more info or to apply online
visit www.bethsoft.com or e-mail your
resume to advjobs@bethsoft.com





SCHOOL OF ANIMATION & VISUAL EFFECTS
Student...Jeffrey Hawkins



want game? we teach it!

80% of our graduates are working in the art and design industry

SCHOOL OF ANIMATION & VISUAL EFFECTS

Games, 3D Animation,
2D Animation, 3D Modeling,
Character Animation,
Background Painting,
Storyboard & VFX/Compositing

ACADEMY *of* ART UNIVERSITY

FOUNDED IN SAN FRANCISCO 1929
BY ARTISTS FOR ARTISTS



1.800.544.ARTS | www.academyart.edu FOR FALL-CLASSES START SEPTEMBER 1

79 New Montgomery St., San Francisco, CA 94105 • Nationally Accredited by ACICS, NASAD, FIDER (BFA-IAD), NAAB - Candidate Status (M-ARCH)

VFS™

VANCOUVER FILM SCHOOL
WHERE RESULTS MATTER



GAME DESIGN

Work Hard, Play Harder

Siggraph

Los Angeles Convention Center

August 2 - 4, 2005

Visit us at **Booth 2166**

Come and see award-winning student work and meet with a VFS representative. Join us for **TechTalk** Wednesday, August 3 at 10am on the show floor.

For detailed program information call **1.800.661.4101** or visit **vfs.com**.

THE GUILDHALL AT SMU

Quality Education. Quality Graduates.
Quality Companies.



Art Creation



Level Design

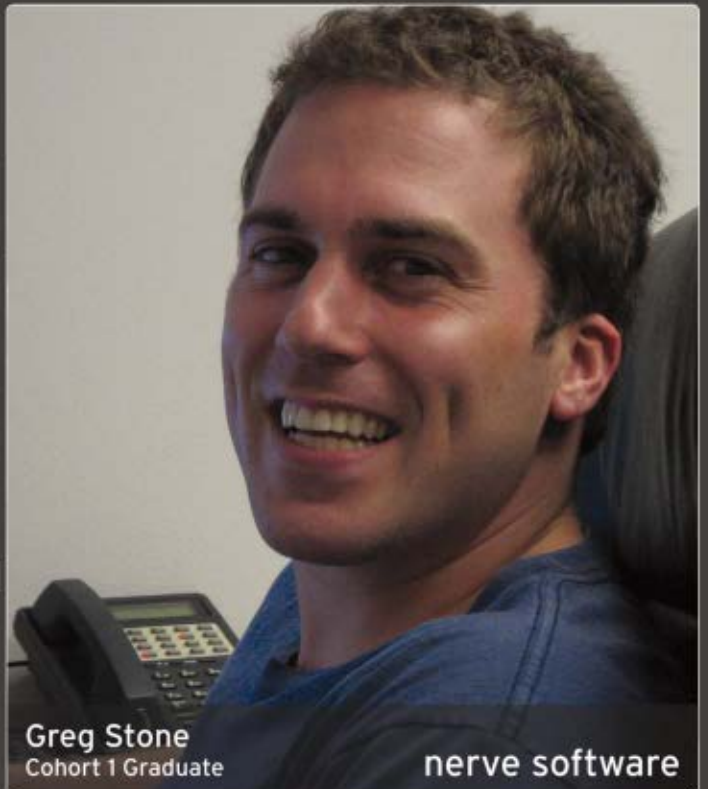


Software Development



Team Games

- ACTIVISION
- AGEIA TECHNOLOGIES
- ARCHIMAGE
- BUDCAT CREATIONS
- FACTOR 5
- GEARBOX SOFTWARE
- ID SOFTWARE
- NCISOFT
- NERVE SOFTWARE**
- NEVERSOFT ENTERTAINMENT
- PANDEMIC STUDIOS
- RITUAL ENTERTAINMENT
- STREAMLINE STUDIOS
- TERMINAL REALITY



Greg Stone
Cohort 1 Graduate

nerve software

COGSWELL

The Fusion of ART & ENGINEERING

Bachelor of Art & Science Programs:

- * Digital Art & Animation
- * Digital Motion Picture
- * Digital Audio Technology
- * Digital Arts Engineering
- * Electrical Engineering
- * Software Engineering

WASC Accredited
Financial Aid Available
Small Class Size
Tours Available

ACCEPTING APPLICATIONS!
CALL NOW!



Yes! Tell me more about the Programs at Cogswell College.

Name _____

Address _____

City _____ State _____ Zip _____

Email _____

Phone _____

School Now Attending _____

Present Grade Level 9 10 11 12 College

Digital Art & Animation

Digital Audio Technology

Digital Motion Picture

Digital Arts Engineering

Electrical Engineering

Software Engineering

Fire Science

COGSWELL

Tel: (800) 264-1995 Fax: (408) 747-0784 www.cogswell.edu

COGSWELL
POLYTECHNICAL COLLEGE

1175 Bordeaux Drive
Sunnyvale, California 94089
(408) 541-0100

www.cogswell.edu

3D Model By Ex'pression Student Paul Nguyen



SOME PEOPLE WERE BORN FOR THIS.

Ex'pression College For Digital Arts



Emeryville, CA
(San Francisco Bay)
877.833.8800

Accredited College, ACCSCT

THE DIGITAL VISUAL MEDIA DEGREE FROM EX'PRESSION

Some are slaves to tradition. But you? You see things a little, well, differently. Maybe you ought to check out the Digital Visual Media degree program from Ex'pression College for Digital Arts. You'll have 24/7 access to professional grade equipment. You'll learn 3D animation and visual effects for TV and movies, film and video games. And you can earn a Bachelor's degree in about 2 1/2 years. Curious? Visit expression.edu for more details.

Tired of your day job?

The Centre for Distance Education offers diploma programs in New Media, entirely online.

- Get a Diploma in:
- 3D Animation
 - Character Animation
 - Video Game Art & Illustration
 - Digital Imaging
 - Digital Publishing
 - Web Page Design

Get with the program. Get in the game.



CENTRE FOR
Distance Education **www.cd-ed.com**
1.866.446.5898

The Centre for Distance Education is a member of the Private Career Colleges Association of Nova Scotia and of the National Association of Career Colleges. Financial assistance may be available to those who qualify. International applicants welcome. Campus is entirely virtual.

Collins College >> Your passion. Our profession.

Add a new dimension
to **YOUR** life.



Collins College offers a Bachelor of Arts degree in Game Design and a Bachelor of Arts degree in Visual Arts with a major in Game Art.

STOP BY OUR BOOTH #2204 AT SIGGRAPH.

CALL NOW FOR INFORMATION **1-888-356-7777**
www.CollinsCollege.edu

Main Campus - Tempe, AZ >> West Campus - Phoenix, AZ

Collins College is accredited by the Accrediting Commission of Career Schools and Colleges of Technology (ACCST). Financial Aid is available for those who qualify. Career services assistance. Collins College West is a branch of Collins College. Not all programs available at all locations.

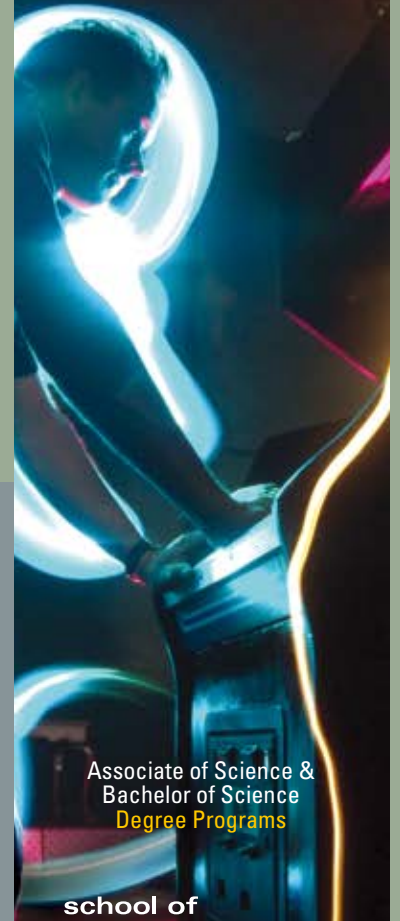


by Rick O'Connor

MARKETPLACE



Game Design & Development
Bachelor of Science
Degree Program



Associate of Science &
Bachelor of Science
Degree Programs

school of

- ▶ Computer Animation
- ▶ Digital Media
- ▶ Entertainment Business
- Film
- ▶ Game Design & Development
- Recording Arts
- Show Production & Touring

800.226.7625
fullsail.com

3300 University Boulevard
Winter Park, FL 32792

Financial aid available to those who qualify
Job placement assistance
Accredited College, ACCST

© 2004 Full Sail, Inc. All rights reserved. The terms "Full Sail," "Full Sail Real World Education," and the Full Sail logo are either registered service marks or service marks of Full Sail, Inc.

EXPERIENCE

THE NEXT GENERATION OF MODELING

FROM 3DS 7

FROM NPOWER SOFTWARE

POWER BOOLEANS

THE MOST ADVANCED BOOLEAN AND TESSELLATION CODE IN THE INDUSTRY.

ADVANCED POLY REDUCTION

CLEAN, LOW POLY RESULTS

MULTI/GROUP BOOLEANS

ADVANCED QUAD TESSELLATION

POWER SOLIDS

UNLEASH THE POWER OF PARAMETRIC 3-D GEOMETRY MODELING.

RAPID COMPLEX MODEL BUILDING

VARIABLE EDGE FILLETING

DYNAMIC MESH TESSELLATION

IMPORT IEGS, RHINO, STEP, SAT

THANKS TO ALL OUR FANATICAL FANS!!!

EMPOWERING CREATIVITY WITH ADVANCED TECHNOLOGY

WWW.NPOWERSOFTWARE.COM

**“I turn
dot.coms
into money.”**

—Justan Brandt
B.A. in Technology Commerce
Class of 2004



www.uat.edu or 800.658.5744

Available online or on-campus.

- | | |
|-------------------|------------------------------|
| GAME DESIGN | VIDEO PRODUCTION |
| DIGITAL ANIMATION | NETWORK SECURITY |
| ARTIFICIAL LIFE | SOFTWARE ENGINEERING |
| MULTIMEDIA | TECHNOLOGY MANAGEMENT |
| WEB DESIGN | GAME PROGRAMMING |

“Let’s just go drop a car...”



Pushing game audio forward, one sound at a time.
www.chakrasound.com

The right way.



NEW PENCIL

www.newpencil.com

LIBERTY LATUS™



Let the Games Begin!

The **TOTALLY WIRELESS SOLUTION** for
ALL-MOTION CAPTURE over **LARGE AREAS**

Self-Contained Wireless Markers • No Body Packs Needed • Full 6DOF • Completely Flexible • Scalable
Over 35 years of excellence

POLHEMUS

First in the third dimension®

40 Hercules Drive | PO Box 560 | Colchester, VT 05446-0560

US and Canada: 800.357.4777 | 802.655.3159 | fax 802.655.1439

www.polhemus.com/libertylatus.htm



Copyright ©2005 Polhemus. LIBERTY and LATUS are trademarks of Polhemus

ADVERTISER INDEX

COMPANY NAME	PAGE	COMPANY NAME	PAGE	COMPANY NAME	PAGE
3D Connexion	2	Datascope Recruitment Ltd.	56	Polhemus	63
Academy of Art University	57	Electronic Arts	49	RAD Game Tools	C4
Activision	48	Expression College	60	Rockstar Games	53
Ageia/Novodex	8	Full Sail Real World Education	61	Seapine Software Inc.	20
Akamai Technologies Inc.	11	Infinity Ward	50	Softimage Co.	C3, 33
Alias Systems	19	Integrity Ware Inc.	61	Sony Ericsson Mobile	23
Anark	13	Iron Lore Entertainment	55	The Collective Inc.	40
ATI Technologies Inc.	5	LifeMode Interactive	35	The Hartecenter at SMU Guildhall	58
Autodesk	C2-1	LucasArts	45	THQ	54
Centre for Distance Education	60	Midway Amusement Games	51	University of Advancing Technology	62
Chakrasound	62	Morgan Kaufmann Publishers	14	Vancouver Film School	58
Cogswell Polytechnical College	59	Mythic Entertainment	52	Xoreax Software Ltd.	35
Collins College	61	New Pencil	62	Zenimax Media	56

Game Developer (ISSN 1073-922X) is published monthly by CMP Media LLC, 600 Harrison St., 6th Fl., San Francisco, CA 94107, (415) 947-6000. Please direct advertising and editorial inquiries to this address. Canadian Registered for GST as CMP Media LLC, GST No. R13288078, Customer No. 2116057, Agreement No. 40011901. **SUBSCRIPTION RATES:** Subscription rate for the U.S. is \$49.95 for twelve issues. Countries outside the U.S. must be prepaid in U.S. funds drawn on a U.S. bank or via credit card. Canada/Mexico: \$69.95; all other countries: \$99.95 (issues shipped via air delivery). Periodical postage paid at San Francisco, CA and additional mailing offices. **POSTMASTER:** Send address changes to Game Developer, P.O. Box 1274, Skokie, IL 60076-8274. **CUSTOMER SERVICE:** For subscription orders and changes of address, call toll-free in the U.S. (800) 250-2429 or fax (847) 647-5972. All other countries call (1) (847) 647-5928 or fax (1) (847) 647-5972. Send payments to *Game Developer*, P.O. Box 1274, Skokie, IL 60076-8274. For back issues write to *Game Developer*, 1601 W. 23rd St. Ste. 200, Lawrence, KS 66046-2703. Call toll-free in the U.S./Canada (800) 444-4881 or fax (785) 841-2624. All other countries call (1) (785) 841-1631 or fax (1) (785) 841-2624. Please remember to indicate *Game Developer* on any correspondence. All content, copyright *Game Developer* magazine/CMP Media, LLC, unless otherwise indicated. Don't steal any of it.

» A THOUSAND WORDS

SONY'S SHADOW OF THE COLOSSUS

Artists: SCEJ art team





TDs: Take Five.

Announcing

SOFTIMAGE® | XSI® v.5.0

- Work with ten times the detail
- Non-destructive everything
- Easy migration from *Maya*
- Native 64-bit

Introducing **SOFTIMAGE | XSI v.5.0**, the digital character and effects software with features that are carefully designed to work for you, and not against you.

Whether you're a technical director, an artist or a developer, with **XSI** software you're free to do what you're best at.
It's really that easy.

Pre-order XSI v.5.0 now at softimage.com/five starting at **\$495***.

Experience XSI v.5.0 at **Siggraph 2005** at the Avid Computer Graphics Booth #1401.

© 2005 Avid Technology, Inc. All rights reserved. *All prices are USMSRP for the U.S. and Canada only and are subject to change without notice. Contact your local Avid office or reseller outside U.S. and Canada. Product features, specifications, system requirements and availability are subject to change without notice. SOFTIMAGE, Avid and XSI are either registered trademarks or trademarks of Avid Technology, Inc. in the United States and/or other countries. All other trademarks contained herein are the property of their respective owners.

Avid.
computer graphics

Pixomatic brings you the whole world

A Global Solution

Games with Pixomatic can run on any Windows or Linux machine that supports MMX. Sell to the largest possible market, save on tech support, and say goodbye to games that get returned because they don't run.

Just Author For Hardware

Pixomatic offers robust DX7-class functionality that your game can fall back to just as it would on an older 3D card.

Easy To Add

Adding Pixomatic to your games is easy, since we designed it with porting in mind. We can even integrate Pixomatic for you—we've ported Unreal® Tournament 2003 and Dungeon Siege, among others.

The Planet's Best Software Rasterizer

True 32-bit pixel processing • per-triangle MIP mapping • bilinear filtering • alpha blending • alpha test • 2 textures and 3 full multitexture stages • independent alpha & RGB channels • 32-bit textures • fog • specular • stencil • 16- & 24-bit z-buffering • indexed primitives • vertex streams • high-speed blitting • and much more!

Built For Performance

Pixomatic automatically generates optimized code optimized for MMX, SSE and 3DNow! on the fly, delivering up to 77 million multitextured, Gouraud pixels/second and 4.5 million textured, Gouraud triangles/second — that's enough to run Quake II at more than 100 frames per second. (Benchmarked on a P4 3.06 GHz)

Source Code Included

Full rasterizer source code in C and ASM included.

For more information:

425.893.4300

www.radgametools.com

Unreal is a registered trademark of Epic Games Inc.
Dungeon Siege is a registered trademark of Cas
Powered Games Inc.



THE BEST IN GAME DEVELOPMENT TECHNOLOGY

