



United Business Media

» PRODUCT REVIEWS TORQUE GAME ENGINE » NUENDO 2.1

SEPTEMBER 2004

game developer

THE LEADING GAME INDUSTRY MAGAZINE

» TOP NEWS

EA BUYS CRITERION,
DEVELOPER REACTIONS

» MANHUNT TO MORTAL KOMBAT

WHY VIOLENCE PREVAILS AND
HOW IT HOOKS PLAYERS

» IMPROVING AI

CUSTOM DEBUGGERS
TO THE RESCUE

POSTMORTEM: THE SWINGING SYSTEM OF **TREYARCH'S SPIDER-MAN 2 GAME**



RT/shader: TMGINZA



As an artist, creating great-looking real-time shaders isn't easy. RT/shader 1.0 changed that all with the first real-time visual shader editor for artists. Now, RTzen is proud to announce the next major milestone release, code-named Ginza.

Ginza continues the revolution by bringing even more innovation, more functionality, and more extensibility than ever before. Download a free preview of Ginza now at www.rtzen.com and see what the excitement is all about.



www.rtzen.com



26

POSTMORTEM

26 THE SWINGING SYSTEM OF TREYARCH'S SPIDER-MAN 2

Before Spidey can rescue distressed citizens or sweep Mary Jane off her feet, he needs to learn to swing without colliding into walls or crashing through windows. Treyarch's programming team had to experiment with the superhero's pendulum physics, anchor-point algorithms, and IK animation to see what stuck and what didn't.

By Jamie Frstrom

DEPARTMENTS

- 2 GAME PLAN** *By Jamil Moledina*
Going to Plan B
- 4 HEADS UP DISPLAY**
EA buys Criterion, Internet cafe curfews, and more
- 8 SKUNK WORKS** *By Justin Lloyd and Alexander Brandon*
GarageGames' Torque Game Engine and Steinberg's Nuendo 2.1
- 56 A THOUSAND WORDS**
Sammy Studios' DARKWATCH

COVER ART: William Lykke, animator, Treyarch.

FEATURES

12 MANHUNT TO MORTAL KOMBAT: THE USE AND FUTURE USE OF VIOLENCE IN GAMES

From GRAND THEFT AUTO's bullet-riddled Vice City to MEDAL OF HONOR's heroic battlefields, varying degrees of violence have been used to simulate an unavoidable part of human interaction—conflict. Steven Kent invites game makers, the ESRB, and the National Institute on Media and the Family to address the issue [without resorting to violence].

By Steven L. Kent



12

18 GROOVY GRAVY: TRICKING OUT YOUR CUSTOM GAME DEBUGGER

If your characters don't return fire in combat, veer off course despite clear objectives, or voluntarily walk into danger zones, you may need to debug your AI. Red Storm uses its debugging tool Gravy to keep the AI of GHOST RECON 2 from misbehaving.

By David Hamm



18

COLUMNS

- 36 THE INNER PRODUCT** *By Sean Barrett* [PROGRAMMING]
Opening Doors
- 40 PIXEL PUSHER** *By Steve Theodore* [ART]
Stitch, Cut, and Relax
- 44 NECESSARY EVIL** *By Paul Hyman* [BUSINESS]
Advergame Aspirations
- 46 AURAL FIXATION** *By Alexander Brandon* [SOUND]
What Buzzes the Masses
- 47 GAME SHUI** *By Noah Falstein* [DESIGN]
Brainstorming

YOU CAN REACH JAMIL MOLEDINA AT JMOLEDINA@GDMAG.COM



GOING TO PLAN B

SO LET'S CUT TO THE CHASE: EA BOUGHT

Criterion. According to both sides (the one side?), Criterion's middleware products, including the fabulous RenderWare 4 suite they dazzled developers with at their GDC castle-booth, will continue to be available to all. In my communications with David Lau-Kee, president and CEO of Criterion (the former?), he's been passionate and dedicated to developers' needs and concerns, so I'm inclined to believe that he sincerely intends for RenderWare to continue to be available. However, that hasn't stopped the rest of the entire developer and publisher community from going to orange alert. While scant few are willing to go on record about it (naturally), developers are suddenly afraid that the drawbridge to their darling middleware package has suddenly been yanked up. After all, having your middleware owned by a camera company doesn't quite have the same impact as having it owned by one of the biggest publishers in the industry. While there isn't a shred of proof (yet) to validate that concern, it's not stopping competing tool vendors from hastily updating their PowerPoint presentations to include a suggestion that developers go to Plan B for their middleware.

PLAN B

Gabe Newell also wants you to go to Plan B, not just with Source engine, the technology behind HALF-LIFE 2's facial animation/real-world physics/shader-based rendering/AI goodness, but with its online alternative to retail distribution: Steam. After all, as he said at Meltdown, "the next battle isn't polygons, it's services." While Microsoft is developing its next major version of Windows, codenamed Longhorn, to include the ability to automatically download patches for games, Steam already does that today and delivers game server connections, instant messaging, and the games themselves. While many publishers are wary to comment on online distribution for fear of upsetting their retail relationships, there comes a time when inflating budgets and dropping MSRPs make it difficult to stay in business, and you have to consider alternatives.

And if you're still not sure there's a real business model there, take a look at Jamdat's Form S-1 registration statement for an initial public offering,

filed with the SEC. This leading mobile game publisher distributes virtually all its games over the Internet, and showed for the first time an operating profit—of a respectable \$738,000 in Q1 2004 (compared to a loss of \$1.7 million for the same quarter in 2002). As more deals are struck in which digital entertainment is legitimately exchanged over the Internet in this and other industries (witness the agreement between Apple and Motorola to link iTunes to mobile phones), it seems to be only a matter of time before that logic takes hold in the game industry—particularly in the Internet-friendly PC game business.

NAVIGATION TIPS

Yet, if all that seems a bit too Plan C compared to where you started, there's a more conservative alternative available. As we continue to teeter on the brink of the next generation of consoles, with varying levels of disclosure reflecting the pecking order of the nobility of elements in the great game developer table, another way for developers to stay in the game is to develop for the PSP and DS. While the SDKs are still pretty pricey relative to previous handheld platform SDKs, overall development costs are supposed to be lower than console title budgets, and it's a great way for developers to retain strong ties to the key platform providers without the gamble of starting a two-year current generation console title project.

When things aren't going as planned, you have to make course corrections. While that scenario may be frequent enough on the creative side of game development, facing it on the business side can be more than a bit disconcerting. But then many of the challenges ahead have a creative workaround to them; the puzzle can be solved. You just have to be willing to accept that your Plan A may not pan out, and then try something new. ✨

Jamil Moledina
Editor-in-Chief

CMP Media, 600 Harrison St., 6th FL., San Francisco, CA 94107
t: 415.947.6000 f: 415.947.6090

EDITORIAL

EDITOR-IN-CHIEF

Jamil Moledina jmoledina@gdmag.com

DEPARTMENTS EDITOR

Kenneth Wong kwong@gdmag.com

ASSISTANT EDITOR

Jill Duffy jduffy@gdmag.com

ART DIRECTOR

Cliff Scorsio cscorsio@gdmag.com

CONTRIBUTING EDITORS

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

ADVISORY BOARD

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

ADVERTISING SALES

GROUP ASSOCIATE PUBLISHER

Michele Sweeney msweeney@cmp.com t: 415.947.6217

SENIOR ACCOUNT MANAGER, EASTERN REGION & EUROPE

Alfon Thatcher athatcher@cmp.com t: 404.658-1415

ACCOUNT MANAGER, NORTHERN CALIFORNIA & MIDWEST

Susan Kirby skirby@cmp.com t: 415.947.6226

ACCOUNT MANAGER, WESTERN REGION & ASIA

TBA t: 415.947.6223

ACCOUNT MANAGER, TARGET PAVILION, EDUCATION & RECRUITMENT

Aaron Murawski amurawski@cmp.com t: 415.947.6227

ADVERTISING PRODUCTION

KEVIN CHANEL
DIRECTOR OF PRODUCTION COORDINATOR Kevin Chanel

REPRINTS Julie Rapp jarapp@cmp.com t: 510.985.1954

GAME GROUP MARKETING

DIRECTOR OF BUSINESS STRATEGY Michele Maguire

DIRECTOR OF MARKETING Jennifer McLean

MARKETING SPECIALIST Scott Lyon

MARKETING COORDINATOR Yukiko Grové

CMP GAME GROUP

VP, GROUP PUBLISHER APPLIED TECHNOLOGIES Philip Chapnick

EXECUTIVE PRODUCER, GAMASUTRA.COM Alex Dunne

ASSOCIATE CONFERENCE DIRECTOR, GDC Susan Marshall

CIRCULATION

CIRCULATION DIRECTOR Kevin Regan kregan@cmp.com

CIRCULATION MANAGER Peter Birmingham pbirmingham@cmp.com

ASST. CIRCULATION MANAGER Lisa Oddo loddod@cmp.com

CIRCULATION COORDINATOR Jessica Ward jward@cmp.com

SUBSCRIPTION SERVICES

FOR INFORMATION, ORDER QUESTIONS, AND ADDRESS CHANGES

t: 800.250.2429 or 847.763.59581 f: 847.763.9606

gamedeveloper@halldata.com

INTERNATIONAL LICENSING INFORMATION

Mario Salinas

msalinas@cmp.com t: 650.513.4234 f: 650.513.4482

EDITORIAL FEEDBACK

editors@gdmag.com

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

CHIEF INFORMATION OFFICER Mike Mikos

SENIOR VICE PRESIDENT, OPERATIONS Bill Amstutz

SENIOR VICE PRESIDENT, HUMAN RESOURCES Leah Landro

VP/GROUP DIRECTOR INTERNET BUSINESS Mike Azzara

VP & GENERAL COUNSEL Sandra Grayson

VP, COMMUNICATIONS Alexandra Raine

PRESIDENT, CHANNEL GROUP Robert Faletra

PRESIDENT, CMP HEALTHCARE MEDIA Vicki Masseria

VP, GROUP PUBLISHER INFORMATIONWEEK MEDIA NETWORK Michael Friedenberg

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



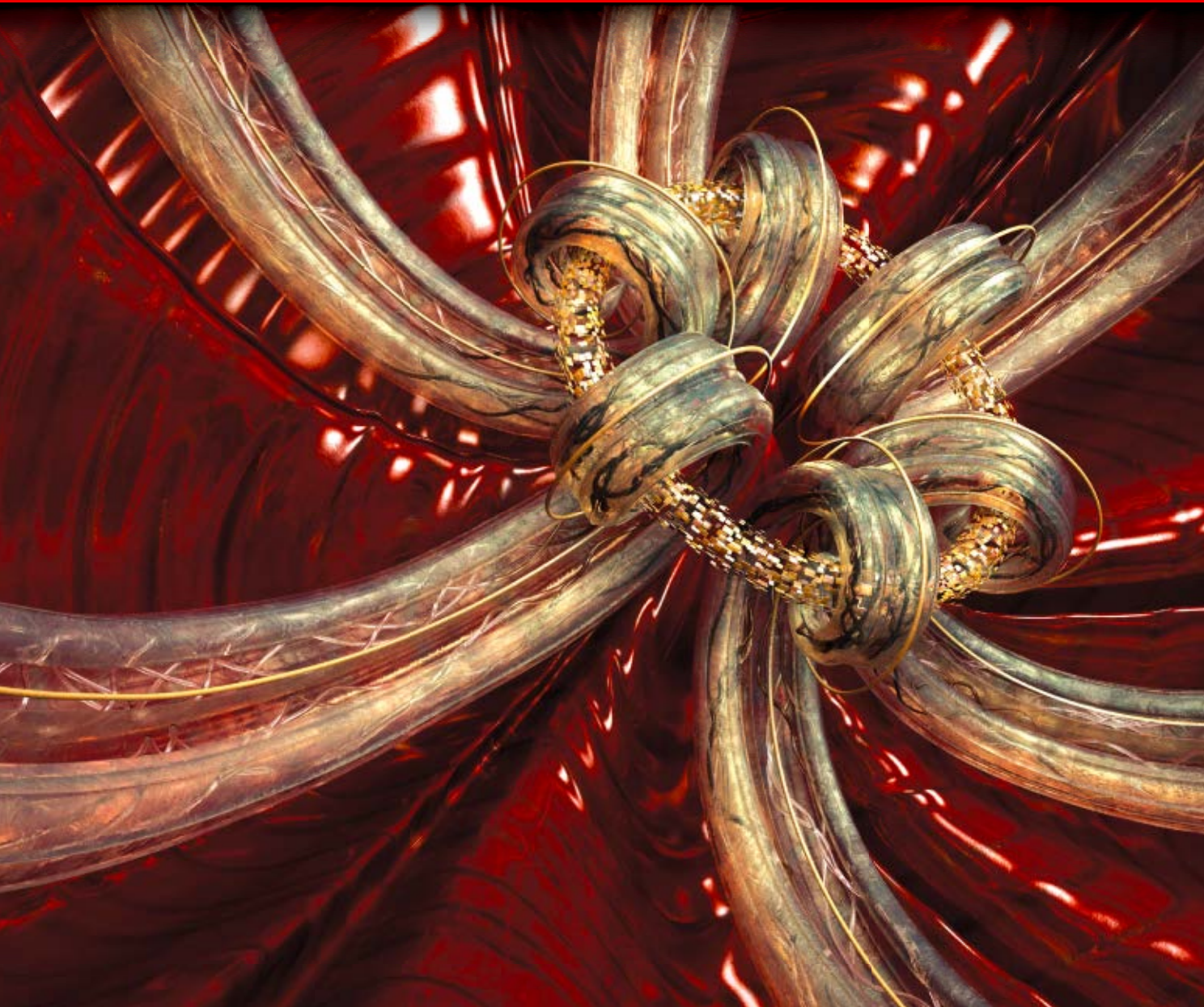
CMP
United Business Media



BPA
Game Developer
is BPA approved

the evolution of 3D

Experience Maya® 6,
a customer driven release packed with literally
hundreds of new features & enhancements.



User requests have led to improvements in character animation, integration with industry standard tools used on every job and the scalability to meet next generation production challenges. Add to this significant advances in modeling, continued refinement of rendering technologies and tools that allow the creation of unparalleled effects and you get the evolution of 3D. Experience the power of the artist and application coming together.



© Copyright 2004 Alias Systems, a division of Silicon Graphics Limited. All rights reserved. Alias and Can You Imagine are registered trademarks and the swirl logo and the Maya logo are trademarks of Alias Systems, a division of Silicon Graphics Limited in the United States and/or other countries worldwide. Maya is a registered trademark of Silicon Graphics, Inc., in the United States and/or other countries worldwide, exclusively used by Alias Systems, a division of Silicon Graphics Limited. Image created by Kenneth A. Huff (www.itgoesboing.com), © Alias Systems, a division of Silicon Graphics Limited.

For more information: www.alias.com/maya

EA Renders the Next-Gen Future

» **WHEN ELECTRONIC ARTS ANNOUNCED** in late July that it would acquire Criterion Software Group, industry heads buzzed at the news and possible outcomes. The purchase, for about \$48 million, gave EA control over Criterion's Guildford, UK, studio; intellectual properties, including *BURNOUT* and a first-person shooter in development called *BLACK*; and the middleware panacea for 20 to 25 percent of developers: RenderWare.

"We did this acquisition because we wanted the studio talent, we wanted the intellectual properties *BURNOUT* and *BLACK*, and we wanted the middleware for internal use to position [EA] for the next-generation technology," says Jeff Brown, vice president of corporate communications.

"As quickly as possible, in the next two to three years, I think you're going to see all EA games using some element of the middleware, but not necessarily the same engine," Brown

says. As for whether the new RenderWare owner will definitely continue to license the middleware, Brown says, "It's our plan."

The middleware's importance lies in facilitating industrywide growth, says Bruce McMillan, EA's executive vice president, which can only occur if developers have effective tools for creating better games. However, anonymous comments from developers suggest an overall feeling of

skepticism. What will stop EA from keeping RenderWare all for itself?

Whether EA will succeed in creating a more robust industry won't affect the company's future position in the market. With RenderWare in its studios, EA undoubtedly has secured a piece of the next-gen pie, possibly profiting from the licensing fees while definitely avoiding the high cost of R&D to create new titles for new consoles.

—Jill Duffy

» first person



as NDL's Gamebryo graphics engine, a second look.

*John Austin,
President and CEO of NDL*

WE HAVE HEARD QUITE A BIT of skepticism about how the industry's largest publisher will license its internal technology to its competitors. What motivation do they have to help other publishers get to market faster with a better game? Which customers will drive features? Who gets the best support at "crunch time?" Given this new development, we believe that developers and publishers should give alternative solutions, such



ZOMBIE IS A LONG TIME licensee of RenderWare—going all the way back to our first two titles in '94. We were planning to license it again for PSP development. I hope EA allows Criterion to continue to license and support RenderWare. It's a great API for cross-platform development.

*Mark Long,
CEO, Zombie Studios*



IT IS PROBABLY TOO EARLY to know what will be the exact outcome of this acquisition. Middle-size and small-size game developers who are using RenderWare are probably very concerned about this acquisition. But, in all prospects, it is a serious sign that the gaming industry has reached a new level of maturity regarding middleware, which is very exciting, as well as a very smart move for EA. It is also very positive for Virtools. Besides validating the interest of the market for

middleware, this latest development comforts our current developments for Virtools's future platform for next-gen consoles.

*Bertrand Duplat
President and CTO, Virtools*



I THINK THE REASON EA acquired RenderWare was to simply get full access to their technology to speed up the development of their next-generation console titles. EA is all about quick and efficient turnaround these days, and has proven time and again that they'll happily spend

whatever money or acquire whatever company that allows them to get a product on the shelf more efficiently. I've heard some say that they think EA is trying to screw the competition, and while I'm sure they enjoy making everybody else nervous, I doubt this is their primary goal. Product on the shelf is the best and most efficient way to beat the competition, after all. Still, I wouldn't be surprised if RenderWare did not remain a viable option three to four years down the road. Developing middleware is not EA's business, so it's unlikely that they'll support its future.

*Jay Wilson, Lead Designer,
Relic Entertainment*

Cyber Curfew Counter-Strikes Turf Wars

A LOS ANGELES ORDINANCE WILL soon restrict what hours minors can patronize Internet cafes that house five computers or more. Any one under 18 will not be permitted at the cafes on school days between 8:30 a.m. and 1:30 p.m., or after 10 p.m. Additionally, owners

of the cafes will be required to install security cameras and obtain a police permit to operate.

The L.A. City Council approved the ordinance in early July. It will take affect as soon as the 30 or so cafes that will be affected are properly informed of how to comply with the regulations.

The move follows repeated outbreaks of violence at the venues—more than 300, including one that left a teen dead—as well as a January 2003 police investigation of the matter, prompted by

CONTINUED ON PG 55



Leland Yee



Dennis P. Zine

Acclaim Encounters Fourth Financial Letdown

WHEN ACCLAIM'S FISCAL YEAR ENDED MARCH 31, THE COMPANY'S NET revenue totaled \$142.4 million, an upsetting number compared to the previous year's \$210.1 million. Officials released the news in July in compliance with NASDAQ Marketplace rules.

The filing included a paragraph from Acclaim's independent auditors explaining causes for the sore financial breakdown, a cascading problem for the last four consecutive years of filings. "The Company's working capital and stockholders' deficits as of March 31, 2004 and recurring use of cash in operating activities raise substantial doubt about its ability to continue as a going concern," a portion of the report in the Securities and Exchange Commission states.

"This explanatory paragraph is not a new qualification, as [the auditor's] reports have included a going concern qualification relating to [Acclaim's] financial statements, for each of the Company's past four fiscal years," company officials stated in a press release.

Furthermore, at least three organizations (Major League Baseball Players Association, Battleborne Entertainment Inc., and Classic Media Inc.) have called off agreements they made with the Glen Cove, N.Y.-based company, saying Acclaim did not meet all the terms of the agreement, such as making royalty payments. Acclaim officials maintain that interpretations of the details are merely in dispute and that the company will seek amicable resolutions.

—Jill Duffy

Splinter Cancer Cells

BEN'S GAME, DOWNLOADABLE AT www.makewish.org/ben, is the brainchild of Ben Duskin, a 9-year-old Leukemia patient, and Eric Johnston, a developer from LucasArts. Following his physician's advice to visualize the body's healing process during treatment, Duskin imagined his medicine as Pac-Man mowing down cancer cells. His disease in remission, Duskin told Make A Wish Foundation he wanted to create a videogame to help other patients cope with the pain and stress of cancer treatment. Make A Wish first approached game studios for help, but to no avail. So the foundation's executive director Patricia Wilson decided to appeal directly to the developer community instead, through game sites and forums. That's where Johnston, the lead programmer behind ESCAPE FROM MONKEY ISLAND and STAR WARS: STARFIGHTER, stepped in.

Johnston, who created his first game PIPE DREAM in 10 weeks, judged that it was possible to make BEN'S GAME without the kind of extensive funding and large crews usually required by commercial titles. He got permission from his employer LucasArts to use its facilities. Over the next six months, he and several developer friends used their spare evenings and weekends to create the game the boy had envisioned. Along the way, Johnston rediscovered the fun of making games without commercial considerations. "No design committee, no economic constraint, just building exactly the kind of game we want to build," he says.

Johnston is planning to deliver a presentation on this project at the upcoming Serious Games Summit in Washington, D.C., October 18–19, 2004.

—Kenneth Wong



Ragnarok manwha creator Myung-Jin Lee

Anime MMORPG 25 Million Strong

ACCORDING TO GRAVITY INTERACTIVE REPRESENTATIVE DAVID KIM, RAGNAROK ONLINE, an anime-styled MMORPG, recently topped 25 million registered users worldwide with over 780,000 concurrent users globally at any given time. Of those 25 million, nearly a million hail from the U.S. RAGNAROK ONLINE's rapid growth can be attributed to its distribution model where, in most cases, Gravity partners with an existing company in the target country that then hosts the game after the initial setup by Gravity.

While most popular MMORPGs feature an immersive world based on 3D realism, RO sports 2D sprite-based anime characters reminiscent

CONTINUED ON PG 55

More Hourly? More Salary!

A FORMER APPLICATION PROGRAMMER FOR VIVENDI UNIVERSAL GAMES, Neil Aitken, is attempting to sue his ex-employer for unpaid time worked since June 2000. Aitken's complaint (Neil Aitken vs. Vivendi Universal Games) claims management at Vivendi instructed nonexempt programmers to falsify records indicating weekend hours worked and "to enter eight hours for each workday regardless of the actual number of hours worked." He alleges that Vivendi issued paychecks for 40-hour work weeks only.

Allen Graves, the plaintiff's attorney, says a law protects full-time computer programmers from getting burned by having nonexempt status. Programmers earning less than \$44.63 per hour by law must be paid time-and-a-half for overtime. Developers whose employers refuse to pay them overtime should keep personal, secure logs of actual hours worked and should contact their state's Department of Labor to file a complaint. Additional information and advice can be found in an IGDA white paper, "Quality of Life," online at www.igda.org/qol.

—Jill Duffy

The Future is Smaller in C#

ARENA WARS, AN ACTION-BASED 3D RTS game set in the year 2137, was developed by exDream Entertainment and distributed jointly by Ascaron Entertainment and Tri Synergy. This, according to its creators, may well be one of the earliest commercial games written entirely in C#. Benjamin Nitschke, the lead developer from exDream, is

credited with single-handedly coding the game. He says writing in C# allows him to harness all the powerful classes in the .NET framework: "It's way easier to write a new application in C#. Unlike in C++, where it takes a long time from coding the first line to executing the program for the first time, this can be done in minutes in .NET."

C# programming allows Nitschke to design a 3D engine that's a mere 3MB in size—compact but nimble enough to support three different game types, 60 single-player missions, and multiplayer mode for up to eight participants (voice- and web cam-enabled). "The basic engine in ARENA WARS," Nitschke explains, "is only 550KB. Another

part of the engine uses some native-code from C++ and Assembler, because .NET is not fit for certain tasks, like OpenGL, voice features, and Webcam features. So all parts came to about 2–3MB." But designing this engine is not a small process; the whole project takes several million lines of code.

—Kenneth Wong

The Glow of Meltdown



IN LATE JULY, MICROSOFT HELD ITS ANNUAL developer-only game conference, showcasing the latest developments in DirectX, XNA, and other gaming initiatives. Microsoft chose the venue to announce that its DirectX 9.0 SDK Update (Summer 2004) is now available from MSDN, tangibly demonstrating the XNA vision of creating a common game dev tool environment. It contains updated versions of the D3DX library, graphics

samples, sample framework, and Managed DirectX documentation. Most notably, the update includes HLSL support for Pixel Shader and Vertex Shader 3.0, the previously Xbox-only PIX tool for debugging Direct3D apps, and a Preview Pipeline Library that offers real-time viewing of shaders on objects, among other goodies.

Amidst all the hoopla, the Windows group made an appeal to developers to apply for the "Games for Windows" logo, as part of Windows' strategy to lure back and expand the PC game market. The program offers free marketing and PR support to PC-debuting titles and an extra boost of credibility to developers pitching their games to publishers. Recognizing that PC users can spend up to an hour trying to install a game only to find that their system is too out-dated to run it, the Windows group also offers a web site geared toward the mass market, featuring a tool that tells if a particular game will run on your PC (www.gameadvisor.com). Also, the latest build of Longhorn showed a dedicated game management structure, which includes a Windows Update-styled automatic patch download system.

—Jamil Moledina

Every developer is a middleware provider, creating tools for future games.

Valve's Gabe Newell discussing the games business at Meltdown

heads up display

WORLD CYBER GAMES 2004 Game Conference

Bill Graham Civic Auditorium
San Francisco, Calif.

October 7, 2004

Cost: \$85–\$180

www.worldcybergames.com/wcg2004

XTREME GAME DEVELOPER'S XPO

Computer History Museum
Mountain View, Calif.

October 9–10, 2004

Cost: \$99–\$249

www.xgdx.com

THE SERIOUS GAMES SUMMIT D.C.

Loews L'Enfant Plaza Hotel
Washington, D.C.

October 18–19, 2004

Cost: \$495–\$695

www.seriousgamesummit.com

CGAIDE 2004 (INTERNATIONAL CONFERENCE ON COMPUTER GAMES: ARTIFICIAL INTELLIGENCE, DESIGN, AND EDUCATION)

Microsoft Campus, Thames Valley Park, UK
Reading, UK

November 8–10, 2004

Cost: EU100–EU450

www.scit.wlv.ac.uk/ffc1822/cgaide.htm

Fall in love with
DevTrack.



DevTrack 5.6

The market leading defect- and project-tracking tool

Powerful

Sophisticated workflow engine
Auto routing and escalation
Beta customer Web portal

Affordable

Fast deployment
Built-in integration with SCM systems
Low cost of ownership

Easy to use

Intuitive user interface
Point-n-click administration
Windows and Web client

More than 1000 companies worldwide, including many of the top game development companies, have chosen DevTrack for its power and ease-of-use. Visit www.techexcel.com to download a free, fully-functional 30-day trial.

TORQUE GAME ENGINE

JUSTIN LLOYD

TORQUE GAME ENGINE

PRETTY SLICK

STATS

GarageGames

392 East 3rd Ave., No. 200
Eugene, OR 97401
(541) 345-3040
www.garagegames.com

PRICE

TGE Indie License (games with less than \$250K annual revenue): \$100;
TGE Commercial License (simulations or over \$250K): \$495 per programmer seat; Torque Shader Engine: (must own TGE) \$295 indie, \$995 commercial;
Torque Network Library: \$295 indie, \$995 commercial.

SYSTEM REQUIREMENTS

For PC. Windows 98/SE/ME/2000/XP, Pentium II 500, 128MB RAM, OpenGL or DirectX-compatible 3D graphics accelerator, DirectX-compatible soundcard, Microsoft VC++ 6.0 and above.

For Mac. OS X, G3+, 64MB RAM, OpenGL-compatible 3D graphics accelerator, Project Builder (OS X only).

For Linux. Pentium 500, 128MB RAM, NVIDIA TNT2 or better 3D graphics accelerator, Linux-supported sound card, XFree86 4.0 or newer with NVIDIA OpenGL drivers, glibc 2.2 or newer (e.g.: Redhat 7.x+, Mandrake 8.x+, Debian 3.0+), SDL version 1.2 or newer (1.2.3 or later is recommended), OpenAL Runtime or SDK Installation Mesa3D version 3.4 or newer (3.4.2 or later recommended), GNU make and g++ (version 2 or 3).

For TSE. Windows 2000/XP, Pentium 1GHz, 256MB RAM, DirectX 9-compatible 3D graphics accelerator with latest drivers.

PROS

1. Responsive on the community forum and deal with support directly.
2. Battle-tested, fully cross-platform, AAA game engine with all the bells and whistles.
3. Full C++ source code to the engine.

CONS

1. Lack of an easy starting point.
2. Documentation somewhat limited.
3. Size and scope of the engine can be overwhelming.

WHEN DYNAMIX, DEVELOPER OF THE TRIBES franchise, imploded a few years ago, several employees carried on the vision of improving the technology that they had developed over the years. Unlike most displaced developers, they did not simply wander off to yet another game company. Instead, they formed a new middleware company, GarageGames, licensing the source code to the TRIBES 2 game engine from the now-defunct Dynamix, marketing it under the name Torque Game Engine (TGE): a viable—and incredibly affordable—alternative for small, independent development houses.



Though Torque Game Engine was initially a FPS engine, GarageGames is beginning to introduce features for RTS and puzzle games, such as MARBLE BLAST, shown here.

WHAT YOU GET. TGE is a true game engine in the same market as the QUAKE and UNREAL engines. TGE contains several individual components: graphics, audio, networking, input, scripting, and tools. There is minimal AI support and the second weakest area is physics. As a TGE licensee, you receive access to the source code repository via the Open Source concurrent version system (CVS). The latest release is always available along with previous versions, allowing licensees to review recent changes that may affect their development schedule.

TGE graphics makes use of a fixed function pipeline that is capable of executing on 32-bit Microsoft Windows, Mac OS X, and certain Linux distributions. TGE makes use of OpenGL with a user-selectable fallback to DirectX 7 for Win32 platforms if a compliant Win32 OpenGL implementation is unavailable for the particular brand of graphics card.

The audio component is a vanilla implementation, enabled through the OpenAL API, offering ordinary stereo and 3D positional audio capability. Both Microsoft's .WAV files and the Ogg format are supported for audio files. Other than OpenGL and OpenAL, all other components of the engine are proprietary to GarageGames.

TGE was developed to handle first-person shooters, so there's quite a bit of that FPS legacy still apparent (in the definition of default cameras and player

input, for instance). GarageGames has been working hard to show that TGE can create more than just FPS games by creating and co-publishing several new titles: THINK TANKS, ORBZ, MARBLE BLAST, plus various puzzle games. In development are role-playing games, puzzlers, real-time strategy, and several MMORPGs. GarageGames is also keen to promote TGE add-ons by third-party developers, some of which include an RTS module and Torque 2D Gamemaker for creating 2D games.

SCRIPTING. TorqueScript, TGE's scripting language, resembles a form of type-less C++ with most of what you would expect from a modern object-oriented scripting language; local variables, functions, inheritance, and function overloading. TorqueScript is interpreted; as each file is loaded, a sort of compile-on-demand reads in a raw text file and tokenizes it, generating a new compiled version, which the game then executes.

The script language is fast enough for most game requirements, but bear in mind that most of what takes place in script is manipulation and testing of a few variables before placing calls in to a powerful C++ game engine. The script is extensible via C++, so a project requiring extra features or a speed boost can make use of a C++ solution.

PHYSICS/AI/MODELING. TGE currently includes little AI function. According to

GarageGames, a third-party AI package is in development. Another area of weakness is physics. GarageGames has put out a capable engine but the physics are rudimentary, with vehicles being the most developed and everything else taken care of with basic collisions, gravity, and friction. Anything beyond that would require a dedicated physics engine such as Havok or Open Dynamics Engine.

NETWORKING. TGE's networking component is the biggest prize in the package, capable of working through firewalls. It's a robust, proven technology designed for high-speed action games in a high-latency, low-bandwidth environment. The behind-the-scenes technology handles the problems of interpolation and prediction to smooth out the gameplay and animation.

GRAPHICS. The graphics core, based on TRIBES 2 (2001), is a capable beast even for a four-year-old design. Feature highlights that really make the graphics engine a powerful deciding factor against rolling your own are: multiple rendering paths for different hardware platforms; cross-platform capability without altering your game assets; a capable, native OS-looking windowing GUI; and automated level-of-detail on almost all 3D game objects, including the terrain and water. Add to all that the built-in terrain, world, and mission-editor tools, and you've got a comprehensive set.

TGE offers large, open environments and a competent portal system for transitioning to building interiors and underground areas. A single area can be several miles across, the terrain repeating indefinitely so there is no edge to the world, and dozens of large, complex buildings and structures can be placed across this area.

One of the issues I encountered with the repeating terrain is that permanent objects, such as buildings placed on the map, cast shadows across the landscape. Travel far enough in any direction and the map will repeat and you can detect the repetition by locating the shadows that are baked in to the terrain's shadow map,

sans objects that are casting the shadows. According to GarageGames, the Torque Shader Engine (TSE) will soon have a completely rewritten terrain engine for improved performance.

For 3D model assets TGE supports most of the major industry-standard packages (3DS Max and Maya, for starters) with limited support for others (such as Softimage). Models are created and then exported via a plug-in to the appropriate game engine format, and the rendering system is capable of handling QUAKE and HALF-LIFE BSP level-data for buildings and interiors.

WHERE IT NEEDS IMPROVEMENT. With a product of this size and complexity, there is going to be a very steep learning curve, similar to learning Maya or Max. Through initial evaluation and development, I had a number of serious issues that left me muddling through with low productivity for several weeks.

My biggest concern is the lack of clear documentation with no easy starting point where you can just jump right in. GarageGames' web site, overflowing with valuable information, often felt like a library after a hooligan had thrown all the books on the floor. The information you need is there; you just have no idea where to look. The signal-to-noise ratio of information is very good; it's just that there's too much signal.

In GarageGames' defense, the documentation problem is being addressed and what's available now is a vast improvement over what was around 12 months ago when I first looked at this engine. The company has been working feverishly the past few months on bringing its documentation up to par with the rest of its product. Proper organization of the web site, the TGE documentation project, and a separate, purchasable resource—Kevin Finney's new book *3D Game Programming All In One*—should prove helpful (see Resources).

FUTURE DIRECTIONS. GarageGames has lined up as a technology partner with Microsoft, porting TGE to the Xbox console ready for Xbox Live Arcade. The

price for licensing the Xbox version of TGE is still to be determined, but from what I gather, the final price wouldn't cover the cost of a typical mid-range laptop.

GarageGames makes no bones about the age of its engine. Three years in game development is a long time, and the engine was designed to run on DX7-class hardware even when DX8 had been kicking around for a while. TSE is GarageGames' answer to many people who point at the latest UNREAL, QUAKE, and HALF-LIFE engines as the geek equivalent of technological muscle-flexing. TSE adds a lot of the features developers have been asking for, namely, a move away from the fixed function pipeline to a more flexible shader-driven format. Rather than patching an already long-in-the-tooth graphics engine or bolting on a simple shading component, GarageGames decided to do it properly. Currently, TSE is available as an early-adopter product for developers keen to exercise the new features and work out the kinks before general release.

I was pleasantly surprised to discover that GarageGames is working with some of the larger handheld device manufacturers to port TGE to platforms that utilize the soon-to-be-released 3D graphic chipsets. Some of these devices were on show this year at GDC with prototypes utilizing chipsets from ATI, Intel, and NVIDIA, which approximate DX7-level device capabilities.

YOU GET WHAT YOU PAY FOR. The general perception in the industry is that the more you pay, the better the product you get. The extra cost buys you, among other things, higher quality, responsive technical support, and proper documentation. It's difficult to decide whether this holds true for TGE. I've worked with other middleware companies—Havok, Renderware, and id, to name but a few—and found them all to be immensely helpful when you are a licensee paying the big bucks. TGE, available for as low as \$100 per programmer, is in a different price bracket, offering a limited level of support. So, if you can live without the mothering the other middleware providers offer, TGE really does measure up in almost every area.

Justin Lloyd has more than 18 years of commercial game programming experience on almost every released platform.

RESOURCES

Finney, Kenneth. *3D Game Programming All In One*. Premier Press, 2004.

GarageGames forums and resources
www.garagegames.com

Torque Game Engine SDK documentation
www.garagegames.com/docs/torque/

Beyond TGE

GarageGames has been wise enough to see beyond TGE, spinning off individual components from its main product to interested developers. At GDC 2004, GarageGames announced the Torque Network Library to be available to developers looking for a robust, proven, award-winning networking middleware component at a reasonable price. I'm unsure what else can be broken out from TGE as separate products, but it will be interesting to see what they try next.

NUENDO 2.1
Alexander Brandon

THERE ARE MANY AUDIO TOOLS

available for general-purpose audio production. When the game industry was relatively young, such tools were out of the financial and practical reach of average developers. Now, game developers are using the same tools used in the record, film, and TV industries. But before we start thumbing our noses at those who once sneered at game audio, let's remember that our audio budgets are still low compared to what other media spend. However, many tools on the market are now comparable with the higher-priced options. One such tool is Steinberg's Nuendo.

Nuendo is a multi-track recording, editing, and postproduction system that covers nearly every audio need. It records multiple audio tracks; has full mixing capabilities compatible with just about any control surface available; features a complete suite of MIDI recording, playback, and notation tools; and offers compatibility with the largest library of plug-ins available: Virtual Studio Technology (VST). A standard developed by Steinberg, VST makes the product not only a recording/editing/post environment, but also a platform for numerous synthesizers and samplers.

Nuendo is a good choice for game developers for two major reasons. First, it is PC- as well as Mac-based and is designed to be the first audio tool to provide the processing-power equivalent to Pro Tools, with native PC processing power. IT departments for internationally dispersed game studios prefer one platform to deal with for troubleshooting and repair, and that platform is PC. (Bear in mind that Pro Tools comes on a fixed hardware platform, whereas Nuendo is hosted by the user's own hardware. Therefore, Nuendo's performance may be affected by your system's horsepower. I used a Pentium 4 2.0GHz machine with 512MB RAM. For maximum performance, Steinberg recommends Intel Dual Xeon processors with hyperthreading, AMD Dual Opteron 64-bit processors, or Appel Dual G5 processors, with 1-2GB RAM, 7,200-10,000RPM HD, and a 800MHz system bus.)

Second, it is a viable alternative to Sony's Sound Forge for sound effects production and editing. Sound Forge has been a staple for game developers worldwide as an audio-editing tool, but the multi-track editing

power that Nuendo offers also gives great control and freedom. Having said that, I use both of these tools regularly. Many of my colleagues still use Sound Forge exclusively for sound effects creation, but let's take a look at how Nuendo handles it.

I had experimented with multi-track sound editing around four years ago, but only saw it put to use masterfully by Todd Simmons, the sound engineer at Ion Storm, on *DEUS EX: INVISIBLE WAR*. When working in Sound Forge, one has a lot of plug-ins available, but currently the only plug-ins natively supported are DirectX-based, which operate with less speed than VST plug-ins for processing needs (applying a reverb to a file will take a few more seconds on a system with the same hardware). In addition, Sound Forge only supports editing two-channel at a time; when working with multiple files, you need to switch back and forth. In Nuendo, you can have as many sound effects as you like in a track-based view. You can also apply effects through channels, rather than applying them to the files directly. Finally, in Nuendo there is a history of every action you make, so you don't need to repeat Control+Z to go back to a specific stage. How often have you edited a file numerous times and wished you could change what you did five edits ago—without going back and making all five edits again? With Nuendo, just as in Photoshop, this is possible. Specifically, suppose you want to create a car engine sound and wish to load six to seven engine recordings simultaneously, then mute any one of them or groups of them at a time. In Sound Forge, this would be a big hassle with copying and pasting, but in Nuendo the process can happen in a fraction of the time using a multi-track editing method.

There are only a few issues with Nuendo. It doesn't have batch processing capability; its video importing codecs vary in quality



Nuendo's multitrack layout with video display.

and speed; and its MIDI functionality isn't quite as honed as the MIDI found in such tools as Cakewalk's SONAR. In the case of batch processing, this is an area where Sound Forge takes up the slack with its Batch Converter 5.0. As for MIDI, Nuendo falls short in the quantization department, putting notes where they don't belong in some cases. An example can easily be demonstrated if you select a section of MIDI notes and set your quantizing parameters, then hit "quantize." You'll notice that if you want your notes to line up perfectly on the beats, it doesn't happen. This works easily on a Korg Triton that I own, so this odd behavior in a major software app is disappointing. For video needs, Nuendo is good but requires small AVIs to work well. It has displays that use DirectX, Quicktime, or ActiveX. Since ActiveX is not only out of use in mainstream development circles, but slow as hell, I recommend using DirectX. But even with this, make sure your video is no larger than 200-300MB, or Nuendo will choke. Aside from these shortcomings, Nuendo is the best tool for the money for all your audio needs and is poised to become a serious competitor for Pro Tools.

Alexander Brandon is the audio manager at Midway in San Diego and the Aural Fixation columnist for Game Developer.

Cubase

It should be noted that many similarities exist between Nuendo and Cubase SX, another Steinberg product. In fact, the two are nearly identical. (Weird, huh?) Nuendo was introduced originally as a postproduction tool and had no MIDI capability, but it did have surround-sound mixing, which Cubase didn't. It was also based on an entirely different engine and was much more robust than the aging Cubase VST. Cubase switched to the Nuendo engine in Cubase SX 2.0. Now, the two are like multi-track powerhouse brothers. Cubase can mix up to 6.1 surround channels. Nuendo can handle up to 8.0 and 10.2! So why not abandon Cubase? Well, Cubase has a large following that remains loyal.

NUENDO 2.1



STATS

Steinberg
280 N. Bernardo Ave.
Mountain View, CA
94043
(818) 973-2724
www.steinberg.net

PRICE

(Nuendo media
Production Systems)
\$1,499.99

SYSTEM REQUIREMENTS

For PC. Pentium/Athlon 800MHz, 384MB RAM, Windows 2000 or XP, ASIO-compatible sound card, USB port.
For Mac. PowerMac G4 867MHz, 384MB RAM, OS X 10.2.5, ASIO- and OS X-compatible sound card, USB port.

PROS

1. Excellent interface.
2. Stable, affordable price for most developers.
3. Most plug-ins available.

CONS

1. No batch processing capabilities.
2. Finicky MIDI functionality.
3. Specific video codec requirements.

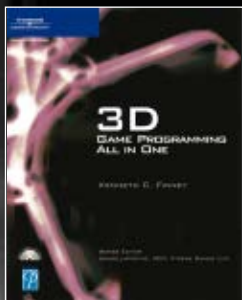
PLAY BY YOUR OWN RULES!

But first, play by ours and win! Join Thomson Course Technology, the leading publisher in game programming, and enter the **Play by Your Own Rules*** sweepstakes at any participating Borders store September 1 – October 31, 2004.

One Grand Prize winner will receive a \$1,000 online tuition package from The Game Institute, a premium membership to GameDev.net, and a one-year subscription to *Game Developer* magazine.

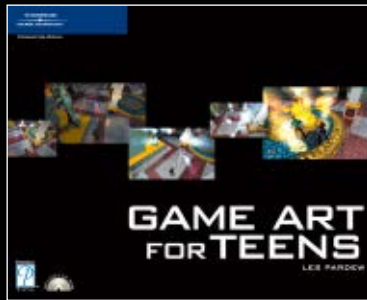
Three Second Prize winners will receive a \$100 online tuition gift certificate from The Game Institute, a premium membership to GameDev.net, and a one-year subscription to *Game Developer* magazine.

Ten Third Prize winners will receive a premium membership to GameDev.net and a \$20 Borders Gift Card.



**3D Game Programming
All in One**

ISBN: 1-59200-136-X
\$49.99



Game Art for Teens

ISBN: 1-59200-307-9
\$29.99



**Game Programming
All in One, 2nd Edition**

ISBN: 1-59200-383-4
\$49.99



**3D Game Engine
Programming**

ISBN: 1-59200-351-6
\$59.99

Save your spot at the Xtreme Game Developer's Xpo.
Sign up today at www.xgdx.com!

ADDITIONAL SPONSORS

THOMSON
COURSE TECHNOLOGY™
Professional ■ Trade ■ Reference



*No purchase necessary. Open to US residents ages 18 or older by 11/30/04. Void in Alaska, Hawaii, Puerto Rico, and otherwise where prohibited by law. Please read the official rules before entering the Play by Your Own Rules Sweepstakes. The complete rules are available at www.courseptr.com/sweepstakes.

MANHUNT TO MORTAL KOMBAT

The use
and future use
of violence in games

>> **BEYOND THE PSYCHOLOGICAL STUDIES, THE** moralizing, and the sales charts, there is a basic truth about storytelling: There is no story without conflict. In interactive games, that conflict is predominantly played out in violence. Just how prevalent is violence in games? Look at the five games nominated for best original game at E3 2004: *DESTROY ALL HUMANS!*, *DONKEY KONG JUNGLE BEAT*, *GOD OF WAR*, *JADE EMPIRE*, and *ODAMA*. Not only does this list demonstrate how prosaic violent games have become, it also shows how the degree of violence can vary from one game to the next.

GOD OF WAR and *DESTROY ALL HUMANS!* have violence in mass quantities, no question about that. But *ODAMA*, a game that combines pinball controls and real-time strategy, recreates the wars of feudal Japan, letting players crush enemies with a giant cannonball. Violent? And what about *DONKEY KONG JUNGLE BEAT*? Here's a game that the Entertainment Software Rating Board (ESRB) will likely rate E for Everyone, in which the main character punches enemies to clear them out of the way. According to the Webster's Dictionary definition of violence, "exertion of physical force so as to injure or abuse," *DONKEY KONG JUNGLE BEAT* is indeed violent. What about *JOHN MADDEN NFL*? If football is described as a violent sport, wouldn't an accurate simulation of that sport be violent?

Perhaps past versions of *MADDEN* may not have been violent enough. The latest iteration includes a

hit-stick feature that lets players add more authority to tackles. There are limits: The NFL will not allow on-field decapitations; but watching the in-game replays of these hard-hitting tackles, you would be hard pressed to say they are not violent.

"For people to get into the games, they need to be aroused," says Dr. David Walsh, president of the National Institute on Media and the Family. "People might not get aroused watching a boring basketball game; but if the game is back-and-forth, seesawing into the last minute, then there is all kinds of interest in that game. I think that arousal and engagement go together."

Walsh, whose organization creates an annual videogame report card monitoring the progress and enforcement of the ESRB rating system, sees violence as one of the most potent ways to immerse players in games. "I believe that is why there is so much of it. I think that the thing that is lacking is the creativity that is needed to engage the player without resorting to the tried-and-true recipe of violence."

BY DEGREES

Mario and Sonic jump on enemies to make them disappear—a nebulous fate that may or may not involve death. Even though this is done with guns, knives, and explosives in the *MEDAL OF HONOR* games, it's no bloodier than death in Mario's Mushroom Kingdom. Then there are games like *MANHUNT* and *KINGPIN*, where the shooting and stabbing produce blood. According to the ESRB, the combination of violence and gore is more offensive than straight violence.

"There are a number of factors that kept both

STEVEN L. KENT is the author of *The Ultimate History of Video Games*. He writes about games for USA Today, MSNBC, Delta Sky, and the Japan Times. His latest book, *The Making of Doom 3*, is scheduled for release in September.





VIOLENCE IN GAMES



MEDAL OF HONOR and CALL OF DUTY in the Teen category,” says Pat Vance, head of the ESRB. They are straight, historical simulations for one thing. They are non-gratuitous in terms of the types of injuries they depict. The amount of blood in these games is minimal. There’s no friendly fire. “These are straight World War II simulations, and the developers made a concerted decision not to include the more gratuitous injuries and other things that you might find depicted in an M-rated game.”

According to David Jaffe, director of Sony’s TWISTED METAL franchise and the upcoming GOD OF WAR, stripping the gore out of games can diminish their impact. “I think you might be able to [separate the gore from the violence], but it’s not as simple as shooting someone and simply not having any blood. The MEDAL OF HONOR games do this. I love those games, but without the blood, they just don’t have visceral impact. They feel watered down.

“I think the idea of creating impactful



“Just like when we saw a glut of cute, furry mascot games when Sonic was ruling the roost, now we’ve got a bunch of GTA/MORTAL KOMBAT/DOOM wannabes. Everyone wants a hit, so they mix their talent for making great games with the flavor-of-the-month vibe [which, for the console these days, can be extreme violence].”

David Jaffe



“I think games are an exciting medium that has tremendous possibilities for the future both for education and for entertainment. My hope for the gaming industry is that it continues to advance and reaches the maturity where it can rely less on violence as a way to engage players.”

Dr. David Walsh, PhD.

violence without gore is very interesting. I have not really thought about it because up until now, my games have been arcade-like, fast-paced titles. I think it would be really hard to create violence without gore in that genre.”

If there is a genre in which violence and gore have been successfully extricated, it’s fighting games. The first one-on-one combat games, such as the Vectorbeam game WARRIOR, were bloodless because of the limitations of the hardware. Even when STREET FIGHTER II suddenly made fighting games arcade headliners, fighters remained unblemished for the most part. Then came MORTAL KOMBAT.

“When Midway released NARC, it was the first digitized videogame—I think it was a little bit before PIT FIGHTER,” says MORTAL KOMBAT co-creator Ed Boon. “All of a sudden, that opened the door for all kinds of stuff, and we thought, ‘Let’s put blood on the screen to shock people.’ It was not something that we set out from the beginning to do. It was more something we could do suddenly with the technology that became available.”

Capcom, and later Namco and Sega, did not follow Midway’s lead. “TEKKEN is the equivalent of a PG-13 movie,” says Boon. “MORTAL KOMBAT is the equivalent of an R-rated movie: an M-rated game. It just presents it in a more hyper-realistic way. The intended audience is different. We never make our games with the intention [of attracting] younger players.

“It’s kind of like saying, ‘Why was *Goodfellas* an R-rated movie? And why would *The Sopranos* not be R-rated in theaters?’ Well, maybe it would. Okay, but another kind of movie of that type. It’s just a different way of presenting a game. Since we did it first in our games, it has become one of the things that

people like about MORTAL KOMBAT. [They like] the extreme presentation, so we keep it; but we don’t think, ‘Oh, this is a necessary ingredient in order for the game to be fun.’”

Boon admits that MORTAL KOMBAT did benefit from a certain amount of shock value back in the early 1990s, but states that despite the head start they got from the shock value, good play mechanics were even more important. “I don’t think that the violence was the main contributing factor. I think what the violence did was that it got people’s attention. People who might not have played the game played the game, and then they got hooked on the secret moves and all the hidden features and all the fun of playing the game. Today, we don’t think that the violence is going to carry us just because so many other games have it.”

In fact, when asked if MORTAL KOMBAT has kept up with the violence in games over the years, Boon’s answer is vehement. “Oh, no. No. When it first came out it was around the top of the heap in violence, but there are games that have long since surpassed it. I think violence has been less and less of an ingredient in every MORTAL KOMBAT game. I’m not saying that the violence has decreased in the games. It’s just that violence is so common in games today that it’s not going to make you stand out.”

As of last Christmas, the poster child for over-the-top was Rock Star Games’ MANHUNT. According to the ESRB’s Pat Vance, MANHUNT isn’t alone. “I think that MANHUNT is very, very high-end. I think that there are other games that are as high. I think that POSTAL 2 was equally high. I think THE SUFFERING in some scenes is equally high. I think that there are other titles [in that category as well].”

CONTINUED ON PG 16



SOFTIMAGE® | XSI®

Jun Takeuchi on XSI

“We continued to work with Softimage on **Onimusha 3** because we knew the project would require reliable support as well as leading edge technology. With its flexible character creation workflow, profound tool integration and unique Animation Mixer, **SOFTIMAGE | XSI** has significantly improved our workflow. And, of course, Softimage’s support and responsiveness continue to be assets as we push the boundaries of game design and development.”

Jun Takeuchi, Producer
Onimusha 3

Be a part of the 3-Democracy – order **SOFTIMAGE | XSI now for only **\$495**^{USMSRP} and receive the 5 DVD Softimage Production Series training set for free (a \$200 value)! Go to store.softimage.com for your copy now.**

Download a free, fully-functional 30-day trial version of **SOFTIMAGE | XSI** version 4.0 from softimage.com/XSI

i am 4. the future of animation.

Character Samanosuke by iFu Long Production, ©CAPCOM CO., LTD. 2004, ©CAPCOM U.S.A., INC. 2004 All rights reserved.

©2004 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 for prices outside the U.S. and Canada. SOFTIMAGE, Avid, XSI, and make manage move | media 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.



VIOLENCE IN GAMES

CONTINUED FROM PG 14

VIOLENCE AND STORY

If there was a watershed moment in which videogame violence went from shock to awe, it was the launch of *GRAND THEFT AUTO III*. Before *GTA3*, people talked about *DEATH RACE*, *CUSTER'S REVENGE*, *CHILLER*, *MORTAL KOMBAT*, and *DOOM*. *DEATH RACE*, *CUSTER'S REVENGE*, and *CHILLER* are historical footnotes—games that defined the boundaries of their time and little else.

MORTAL KOMBAT is another story. *MORTAL KOMBAT* was a major best seller, but it sold into a decidedly non-mainstream market of pre- and young-teen boys. *DOOM*, on the other hand, may be said to have broken into the mainstream. But *DOOM* launched nearly concurrently with *MYST*, the uniquely mainstream title that helped launch multimedia as a technology for the masses.

More than any other company, Sony Computer Entertainment has succeeded in making games a mainstream form of entertainment. And with the mainstreaming of videogames, the stage was set for *GTA3* to become a truly mainstream game.

"I clearly remember when the first two *GRAND THEFT AUTOS* came out,"



"I do not think violence is a way of immersing players as much as a way of getting their attention. It's almost like slapping somebody in the face; and now that you have their attention, you need to keep their attention."
Ed Boon

anything other than a sense of vicious fun that plays into the gameplay well," says Jaffe. "It serves as one of the core elements in *TWISTED METAL*, now *GOD OF WAR*. I like violence in games if it's done in a creative, interesting way."

According to the Motion Picture Association of America, 81 percent of the movies submitted for ratings last year received an R rating. Twelve percent received PG-13, 6 percent received a PG rating, and only 1 percent received a G. (Only one movie



"When people think about videogame violence they tend to think about the extreme forms which are found in the M-rated category. The Teen games have violence in them, but it comes in all forms, some of it is relatively cartoon-like."

Pat Vance

says Boon. "There was a very small uproar: 'I can't believe they are letting people carjack as part of the gameplay. Oh my god!' But that was just a PC game. It was not really mass market at the time and you weren't right there for the action. Suddenly, they presented it on PlayStation 2 and they had made great advancements in the gameplay. That's when everybody started playing it. And when everybody started playing it, that was when it got the attention."

Unlike the fighting, shooting, and FPS games that preceded it, *GTA3* had a sophisticated storyline. It had its own dynamic world. "With certain games, violence is one of the tools that allows me to direct the feel/vibe of the game," says Jaffe. "In the case of *GOD OF WAR*, I wanted there to be this vibe of letting your inner beast out to run free; letting the player just cut loose and run wild. That was my barometer. It was like: 'Is this element making the player feel strong and brutal?' If so, in it goes. And more often than not, violence was one of the tools that allowed us to give the player this feeling."

If the E3 demo was any indication, *GOD OF WAR* is an excellent example of how violence can be integrated into a highly mature game. In some ways, *GOD OF WAR* feels like a bad acid trip version of Nintendo's *LEGEND OF ZELDA* games. It has the same responsive controls and similar combat mechanics; only, instead of happy elves and a sparkling TriForce, *GOD OF WAR* has a suicidal Spartan and the hordes of Hades.

"For *GOD OF WAR*, we use violence to complement the storyline and make the player feel strong, brutal, and unhindered by

received an NC-17, making up less than one percent.) "Last year 10 percent of the games were M-rated, and that was up from 8 percent in 2002," says Vance. "It's gradually increased over the years from 6 or 7 percent to 10 percent. There has probably been more fluctuation in other categories such as T than there is in M in terms of growth."

Not surprisingly, Dr. Walsh is concerned about a rise in violence across several media. "I would say that [violence] is more prevalent in games. I mean, there are a lot of violent movies, but there is a wider palette of themes in the movie industry than in the game industry. That could have a lot to do with the maturity of the industry. The movie industry is nearly one century old, whereas the game industry is relatively new."

When discussing violence in games, terms like "comic" and "cartoon" come up often. According to Vance, the violence in many of the T-rated games is cartoon-like. "I think of it as being like punctuation, like an exclamation point," says Boon. "It's not necessary for getting your point across, but it heightens things."

"For me personally, I don't like it when violence gets too real from a subject matter standpoint," says Jaffe. "All of my stuff is fantasy, comic book-style stuff. Even *GTA3*, set in the real world, has a comic book/action movie vibe applied to it. Same with a game like *MAX PAYNE*."

"That's where I am comfortable with violence. That's where I think most players are comfortable: when the violence is presented in a way that it clearly is done for fun and visceral impact. It's when you start getting to the real dark stuff, and the ripped-from-the-headline scenarios, that people start either tuning out or getting upset." ❖

Battlefield Brought to Life.



"Havok raised our game to a whole new level. The dynamic gameplay scenarios resulting from Havok physics are outstanding and have brought MCHPA to life."

Brady Bell, Project Lead, EA LA

"Havok enables our development teams to focus on the gameplay and not the technology. The world of MCHPA is a dynamic place. Simulating it with Havok makes it easy."

Mark Dochterman, Technical Director, EA LA

Check out Havok on the web at:

www.havok.com

Went Havok in your next game? Send email to: eval.gdm@havok.com

MEDAL OF HONOR
PACIFIC ASSAULT

powered by
havok®



Havok and Havok 2 are registered trademarks © 2004, of Havok and Telekinesis Research LTD.
Medal of Honor Pacific Assault Materials © 2004 Electronic Arts Inc. Electronic Arts, EA, EA GAMES and the EA GAMES logo are trademarks or registered trademarks of Electronic Arts Inc. in the U.S. and/or other countries. All Rights Reserved. Medal of Honor is a trademark or registered trademark of Electronic Arts Inc. in the U.S. and/or other countries for computer and video game products. All other trademarks are the property of their respective owners. EA GAMES™ is an Electronic Arts™ brand.

GROOVY GRAVY

TRICKING OUT YOUR CUSTOM GAME DEBUGGER

» JUDGING THE EFFECTIVENESS OF GAME AI IS OFTEN DIFFICULT.

Even when problems are identified, debugging the problems can be very time consuming and tedious. A runtime tool for tracking game state can be a great help in improving AI and reducing frustration in many areas of development. AI coding on *GHOST RECON 2* benefited greatly from the creation of a remote debugging tool affectionately known as Gravy. The term loosely translates to *GHOST RECON* Analysis and Verification utility, but the name actually predates the acronym.

While developing an organized assault behavior for enemy AI, many groups of enemies were unexpectedly gathering in the same area of the test map before proceeding to the target. Pathfinder results displayed in Gravy quickly revealed that an incorrectly specified danger zone was being factored into the pathing costs. A small AI coding change was made and the intended result was verified in Gravy.

In another debugging session, a group of characters seemed unable to return fire on the player. Gravy clearly indicated that the troops had a combat behavior, valid line of sight, sufficient aim time, and a number of other prerequisites for firing. A look at the code revealed that only the desired rate of fire was unverified in the tool. The rate of fire request had indeed been ignored. This was remedied in the simulation code and the rate of fire tracking was added to Gravy.

A *GHOST RECON 2* designer observed two teams of enemies closely following their desired paths when suddenly, they veered off course to take up defensive positions elsewhere. The behavior stacks in Gravy showed that platoon-level orders had overridden the team-level plans, a problem easily fixed with scripting adjustments once it was understood.

This evolving tool has been in daily use for nearly a year, initially assisting engineers with AI and simulation development. Mission scripters have also embraced the tool as a means of accelerating their design iterations. The advantages of a custom game debugger, for AI and beyond, can greatly outweigh the perhaps surprisingly lean development investment in building such a tool.

DAVID HAMM is a seven-year veteran of Red Storm Entertainment, where he works on all the game features that can't be easily seen, heard, or explained to his parents. He can be contacted at dhamm@gdmag.com.

BUILDING BLOCKS

The core components of a custom game debugger are very basic—a method for messaging between the game and tool is the main prerequisite. Data structures are needed to store and provide access to the debug information, and UI elements should display the data in a useful format.

Gravy uses the Xbox debug channel for communication with the *GHOST RECON 2* Xbox engine and a basic Winsock system for the PC version. Most networking sample apps do more work than is needed to get debugger messaging running. Care should be taken not to interfere with net traffic for any multiplayer component or other development tool. Also, an automatic reconnection scheme between game runs will save the hassle of manually reconnecting the debugger.

Messages can be sent to the debugger as specific events occur, such as an AI spotting a player for the first time. Other data is more appropriately tracked each frame and sent to the debugger when changes are detected. An overview of game data tracked in Gravy is listed in Table 1.

In the case of rapidly changing information, such as object position, it may be necessary to only register the change when a threshold delta has been exceeded. For example, Gravy updates

- Static 2D world bounds and obstacles
- Object creation, damage, and death events
- Object movement and facing
- Object awareness lists and active threats
- AI behaviors and target locations
- Human state, including stance, alertness, and suppression levels
- Team state, including rules of engagement and threat levels
- Vehicle state, including steering parameters
- Detailed gunshot and explosive data
- Pathfinder queries and results
- Standard debug text statements
- Object-filtered debug text statements

TABLE 1 Gravy tracking data for *GHOST RECON 2*



GROOVY GRAVY

LISTING 1 Overview of Gravy message construction for game debugger data.

```
/* Gravy data message types */
enum GravyDataType
{
    gdtDebugOut,          // Standard debug text
    gdtAIScriptChange,    // Addition and removal of behaviors
    gdtSimEvent,          // Spawning, shooting, and wounding
    gdtObjectMovement,    // Position updates
    ...
    kNumGravyDataTypes
};

/* Base class for Gravy data messages */
struct GravyData
{
    float    mTimeStamp;    // Game time in seconds
    GravyDataType mDataType; // Class of data for filtering
    UInt32   mObjectId;     // ID of human, team, etc for filtering

    GravyData(GravyDataType type);
    ...
};

/* Human and vehicle positional update message */
struct GravyObjectMovement : public GravyData
{
    float    mXPos;        // World X in meters
    float    mYPos;        // World Y in meters
    float    mFacing;      // Angle in radians

    GravyObjectMovement() : GravyData(gdtObjectMovement)
    ...
};
```

human positions with two meters of precision and facing within 60 degrees. It's important to find a balance that provides an accurate view of the game state without flooding the messaging system.

Listing 1 provides an overview of the message construction scheme used with Gravy. A base class common to all debug messages holds a timestamp for the data and two properties to help with information filtering. A data type identifies the class of message and an object ID associates the message with a game entity such as a human, vehicle, or team. Derived message classes hold the actual game debug data. This system allows for relatively efficient message handling without the overhead of a fully generalized messaging system.

The rest of the debugger coding effort falls into data management and display. A standard UI library, MFC in the case of Gravy, can facilitate this work. The needs here are game-specific, but a general log dialog of all messages received will help not only with game debugging; it can also be invaluable in verifying the debugger itself. This log should be easily searchable and preferably filterable. Gravy also includes dialogs with information about the selected object, including AI behavior stacks, and playback controls. The main pane uses basic GDI calls to draw the overall state of the game.

WINDOW ON THE WORLD

The original motivation for Gravy was to evaluate the existing AI code of the GHOST RECON series. This was a big task as the primary author of

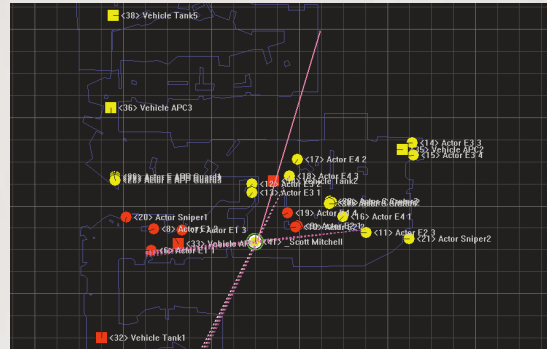


FIGURE 1 Gravy receives the navigation mesh edges from the game and displays a basic top-down world view. Gridlines are used to convey a sense of scale.

the code had moved on, leaving a great deal of quality work but limited documentation. Gravy proved ideally suited to this task, providing information more reliably and efficiently than code inspection. When additional programmers joined the project well into development they too could turn to Gravy as a way to get up to speed on the state of the AI.

Although many games are played in a 3D world, it's hard to beat the efficiency of a 2D, top-down view in relaying basic situational data. Figure 1 demonstrates a typical world rendering seen in Gravy. The in-game maps in the GHOST RECON series are heavily dependent on assets that cannot be generated until the levels are nearly complete. By sending Gravy a representation of the static world at the start of each game, the debugger can render a simple but very helpful view of the play space.

Since this representation is based on messages straight from the game, there is no problem finding the right data files to match the map in use. In fact, Gravy reads no game files at any point. This single, direct line of debugger input adds credibility, as any displayed results must be purely reflective of the actual game data in use.

Using a custom game debugger is a great way to improve the visibility of the otherwise hidden inner workings of a game engine. AI decision-making, from strategic choices to weapon aiming, is a large class of behavior that's often difficult to debug with only the player's viewpoint in the game. Figure 2 (page 22) illustrates some of the AI state information that can be displayed for a character in Gravy.

Instrumenting code by pushing state changes and events to the debugger as a feature is being written can turn up bugs quickly and help verify the desired behavior. When a feature is finished or put on hold, the debugger can continue monitoring its execution as other parts of the engine are developed. This can give another layer of protection against future breakage beyond that of sprinkled assertions and regression testing.

Most games benefit from a simple text output of debug spew. In big projects, however, key debug statements regarding specific objects can quickly get lost in a long list of irrelevant spam. This can lead to a lot of careful searching or removing worthwhile messages that aren't related to features in active development. Using a debug output filtering scheme can improve the efficiency of this traditional debugging aid.

Figure 3 (page 22) illustrates the Gravy message log with filtering options. Once the debugger has a display of all the

CONTINUED ON PG 22

THE WORLD'S **FIRST**

**IT'S
HERE!**



4

MEGAPIXEL

MOTION CAMERA
CAPTURE

4 MegaPixel CMOS Sensor
2352 x 1752 resolution
10 bit greyscale processing
Multiple processor
Unlimited Scalability
1:117,000 accuracy



is...

state of the art,
breakthrough
technology that
allows use of
the very highest resolution
digital cameras for 3D motion
capture.

MX accuracy is measured in
hundredths of a millimeter -
smaller than the period at
the end of this sentence.



VICON
MOTION SYSTEMS

VICON USA 949.472.9140
VICON UK +44 1865 261 800
www.vicon.com

A subsidiary of **OMG**

GROOVY GRAVY

CONTINUED FROM PG 20

important game objects, a selection can be tracked. If debug statements are tagged with an associated object identifier, and the debugger is already receiving the debug output as messages, a filtered debug trace is just one more small step with a big payoff.

A common buzzword in software development today is test-first methodology. There's a lot of truth to the philosophy that a feature is not complete unless it's fully verifiable. Unfortunately, games frequently include complex systems that challenge traditional testing processes. Although raw data output to a debugger can't fully prove the correctness of game systems, it is a relatively effective alternative to writing endless test-case code. If the impact of a feature still cannot be judged clearly with a custom debugger, it's a dangerous feature that could break silently later.

INSTANT REPLAY

Perhaps the easiest way to dramatically increase the utility of a game debugger is adding playback functionality. With the tool already receiving and logging messages, piping the messages back through

the system on command is a natural extension. Playback controls for Gravy are shown in Figure 4.

A visual replay of events can be very helpful in identifying the context in which an error or strange behavior occurred. Full rewind functionality may be more effort than necessary, but simply clearing the world state and replaying the message log at variable speeds provides the key benefit.

Gravy provides a run-to option that will process messages up to the selected log entry, wrapping around to the start of the game session if necessary. A message-by-message step-through command is also available. Finally, additions and deletions from AI behavior stacks of the selected object can be stepped through to show a clear trace of the decisions controlling its actions. Note that it should be possible to continue to receive real-time game data while reviewing an earlier portion of a game log.

With a continuous stream of game data available for later review, a custom game debugger becomes a suitable test bed for stress testing with long-running, standalone game sessions. Gravy can track a battle of eternally respawning forces that fight one another throughout the night. In the morning, if the steady-state behavior has broken, the log can be replayed and quickly advanced until the problem is isolated.

If playback is available for the current game session, the message log can also be saved to disk for future review. Like crash dumps and standard text logs, a game debugger log can be passed to relevant programmers and attached to bug reports to aid debugging. In the case of AI behavioral bugs, sometimes a Gravy log is a great help in just describing the problem, let alone actually fixing it. As with saved games and other file formats subject to change during development, a versioning scheme should be considered to avoid log file-read failures.

Replay-capable debug logs have benefits beyond debugging individual

problems. Logs of extended game sessions can serve as a snapshot of the state of the game flow for a particular build. Saving such logs can be particularly useful before diving into major system rewrites, if they are available for comparison later. A visual debugger can also provide a highly crash-resistant demo environment for showcasing technology that can be appreciated in such a tool, such as coordinated movement of forces.

A debugger with playback functionality merits some comparison to traditional replay systems. In practice, Gravy has been much more reliable throughout development than an in-game replay system. A full replay system can be easily broken with a missed message that throws game events out of synch. With a dedicated debugging tool, developers can pick and choose which data to track and record during execution. Although Gravy tracks some delta-based data, there have been no instances of logs missing data required for accurate playback.

FEATURE CREEP

A custom game debugger supplements traditional code-based tools and analysis, opening up new approaches to investigating and solving development problems. A simple debugger framework can begin to produce these advantages, allowing new data-tracking features to be added incrementally as a project grows.

A standalone debugger avoids some of the pitfalls associated with piggybacking tools on the actual game engine code. It is common to have a game editor share as much code with the game as possible to allow accurate visualization of levels using the game renderer. Although this may be a big win for an editor, sharing code in this way can result in the tool breaking the game or the game breaking the tool. If the debugger hooks in the game engine are solely responsible for collecting and sending data, these dependencies carry minimal risk.

GHOST RECON 2 mission scripters have found Gravy helpful in finding behavioral bugs, which can often be quickly isolated as scripting problems or game engine problems. An alternative approach might involve building a simplified simulation engine into the scripting editor to provide feedback to the scripters. Any such tool may be better than nothing, but the game debugger is again more credible as the scripts are executed on the actual game build.

Once a game debugger is receiving a rich stream of game state information, additional analysis can be performed. For GHOST RECON 2, Gravy receives



FIGURE 2 Basic AI state information can be viewed in the main pane. The selected object dialog provides detailed behavior stacks.



FIGURE 3 The Gravy message log can be filtered by information type and the selected object. The log can be dumped to a text file for search operations.



FIGURE 4 Playback controls allow for offline analysis of Gravy logs.

details about each gunshot or explosive fired. The creation and completion of all AI behavior objects are also noted. In addition to representing these changes and events as they occur, Gravy is a natural place to tally statistical tables of weapon usage and AI behaviors.

In this way, Gravy has been useful in tuning the combat model and in finding appropriate sizes for the AI behavior memory pools. Sample Gravy statistical output is shown in Figure 5. Monitoring statistical results is also a way to track overall changes during development and can turn up unexpected side effects and problem areas.

Although the process of adding new tracking data to a custom debugger can be streamlined, pushing all state information in a complex game is impractical. However, an object selection mechanism in the tool can be leveraged to specify targeted execution breaks at runtime. For example, when attached to a game running under the code debugger, Gravy can force a break in the selected human, vehicle, or AI behavior source code. This provides quick access to the current state of any class members in these objects that are not already explicitly exposed in Gravy.

As with other debuggers, a custom game debugger can benefit from two-way communication with the game. Such a tool is a natural place to include a console interface for sending debug commands to the game. In addition to standard text entry, Gravy provides shortcuts for commonly used cheat codes, in-game debug display toggles, and other utility commands.

The 2D map display can also be used to specify a destination for a "teleport player" command, triggered by a double-click on any valid ground. This command interface could be further extended to simulate events such as destruction of the selected object or an explosion at a specified location.

CHOOSE YOUR OWN ADVENTURE

The idea of a custom game debugger is not new, but using such tools isn't currently a standard practice in the industry. The long list of benefits is perhaps held in check by the budgetary concern of allocating development resources to a tool that is not explicitly required in the

- Enemy and Allied AI -			
Aimed shots: 439 (5.53%)	Rolled hit: 179 (40.77%)	Actual hit: 95 (21.64%)	
Suppress shots: 7306 (92.02%)	Rolled hit: 1068 (14.62%)	Actual hit: 1041 (14.25%)	
Moving shots: 159 (2.00%)	Rolled hit: 5 (3.14%)	Actual hit: 4 (2.52%)	
FixedGun shots: 36 (0.45%)	Rolled hit: 6 (16.67%)	Actual hit: 0 (0.00%)	
Head hits - Rolled: 0 (0.00%)	Actual: 99 (8.63%)		
LeftLeg hits - Rolled: 319 (25.36%)	Actual: 256 (23.75%)		
RightLeg hits - Rolled: 319 (25.36%)	Actual: 266 (24.68%)		
LeftArm hits - Rolled: 250 (19.87%)	Actual: 84 (7.79%)		
RightArm hits - Rolled: 300 (23.85%)	Actual: 54 (5.01%)		
Chest hits - Rolled: 70 (5.56%)	Actual: 325 (30.15%)		
Hit wrong target: 62 (5.73%)	Hit wrong part: 416 (36.59%)		
Target switches: 52 (152.69 shots per switch)			
Grenades thrown: 1	launched: 0	Rockets: 0	
			<ul style="list-style-type: none"> <0> Actor Allied Defender1 Crouch Alert NotSup <2> Actor Allied Defender3 Crouch Alert NotSup

FIGURE 5 Gravy tabulates some recorded game events into statistical reports, useful for gameplay tuning.

Need
GAME AI?



Fast & Visual!
Reusable!
Collaborative!

A.I. Middleware for Games

www.simbionic.com

From your friends at
Stottler Henke

GROOVY GRAVY

production chain. Focus on tools development seems to have steadily increased in many studios, but much of this effort has been needed to meet increasing demands on integrating art, sound, and other assets. Less attention has been given to tools that help solve the growing problems of code complexity in modern game engines.

The resource demands of deploying such tools aside, they do have limits. A game debugger will likely be best suited to debug builds. The messaging hooks in the engine and the extra network traffic can throw off profiling data and could potentially skew the performance too much for a release build. Gravy is configured to connect with debug builds only, but support can be quickly enabled in release builds for special-case debugging needs.

Any standalone tool development may face issues of access to existing game code, such as enumerated types and helper routines. The same architecture of separation that avoids conflicts and upkeep between games and debuggers can also be a drawback when attempting to access structures that are readily available in the main game code. Gravy links with some core utility libraries shared with the



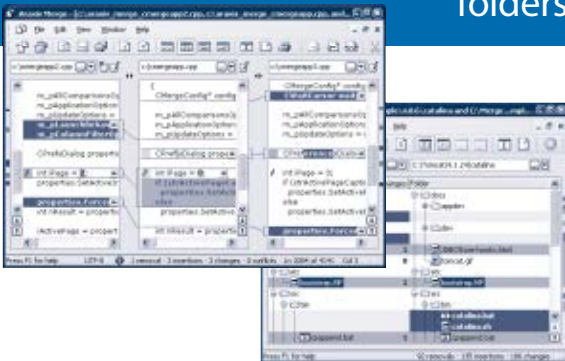
game that change infrequently to provide some basic common functionality.

Another useful practice to reduce frustration involves passing enumerations through the debug messaging system at runtime, rather than statically redefining them in the tool. For instance, when new AI behaviors are added to GHOST RECON 2, Gravy code once had to be updated to recognize the new type. Now, less maintenance

is required because AI messages to Gravy define the behavior types dynamically.

Gravy continues to grow, improving our ability to add new features safely to a massive game. The nature of AI development on GHOST RECON 2 made it evident that Gravy would be a clear win after just a few days of design and experimentation. Other projects may require a larger leap of faith to invest in a custom debugger. However, even with the rise of licensed engines and middleware, games are only getting bigger. When adding that final hour must-have feature, a clear view inside the many layers of game logic is invaluable. ❖

Take control with Araxis Merge. Compare and merge your files and folders



- See differences at a glance
- Two and three-way compare
- Create HTML reports
- Point-and-click merging
- In-place editing with undo
- Automatic merging
- Integrated folder hierarchy comparison and merging
- Unicode/MBCS support
- Integrates with other apps

Araxis Merge v6.5 for Windows is designed to help you work quickly and accurately with multiple revisions of text files. Use it for team software/web development, code review, code audit and release management. Download a **free 30-day trial** from www.araxis.com.

Standard Ed \$129 • Professional \$219

 **Araxis**[™]
www.araxis.com

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
- Compatible with any Visual C++ Win32 project
- Fully integrated with MSVC's IDE

XOREAX

www.xoreax.com



Getting your game to market, you have two choices:

1. Use Perforce
2. Be a runner-up

Whoever said Games were all about winning and losing... was right - so get Perforce's Fast Software Configuration Management on your team.

Perforce's Software Configuration Management System is the choice of winners, because no other tool can match the speed, control and reliability that it brings to the management of digital assets (binary and text files).

Maintain your top speed no matter how many users or files. At the core of Perforce lies a relational database with well-keyed tables, so simple operations can be accomplished in near-zero time. Larger operations (like labeling a release and branching) are translated into keyed data access, giving Perforce the scalability that big projects require.

Install it fast, learn it fast, execute operations fast. With other SCM systems, developers face an unpleasant choice: do it the right way or do it the fast way. Perforce's speed and reliability mean the fast way is always the right way.

Work anywhere. Perforce is efficient over high-latency networks such as WANs, the Internet and even low-speed dial-up connections. Requiring only TCP/IP, Perforce makes use of a well-tuned streaming message protocol for synchronising client workspace with server repository contents.

Develop and maintain multiple codelines. Perforce Inter-File Branching™ lets you merge new features and apply fixes between codelines. Smart metadata keeps track of your evolving projects even while they develop in parallel.

Supports all development platforms. Perforce runs on more than 50 operating systems, including all flavours of Windows, UNIX, Linux, Mac OS X and more.

Integrate with leading IDEs and defect trackers: Visual Studio.NET®, Visual C++®, Visual Basic®, JBuilder®, CodeWarrior®, TeamTrack®, Bugzilla™, ControlCenter®, DevTrack® packages, and more.



Fast Software Configuration Management www.perforce.com

Download your free 2-user, non-expiring, full-featured copy now from www.perforce.com
Free (and friendly) technical support is on hand to answer any and all evaluation questions.

All trademarks used herein are either the trademarks or registered trademarks of their respective owners.

THE SWINGING SYSTEM OF TREYARCH'S SPIDER-MAN 2 GAME



AS KIDS, WE ALL WATCHED THE 1970S SPIDER-MAN TV SHOW AND COLLECTIVELY

wondered “what exactly does Spidey’s web attach to?” While nobody batted an eye when Spider-Man seemed to cast his web into thin air and swing through the sky, we couldn’t help thinking that there had to be a better way to represent swinging. When the movie came out, the issue was put to rest: Spider-Man doesn’t have to swing from the sky; it’s much more dramatic to swing from buildings. And the rest, as they say, is history.

CONTINUED ON PG 28

WRITTEN BY:



JAMIE FRISTROM is a programmer, manager, and designer who has worked in the game industry since 1991. Titles he’s worked on include the *MAGIC CANDLE* series, *DIE BY THE SWORD*, *TONY HAWK*, and *SPIDER-MAN* and *SPIDER-MAN 2*. He’s contributed to *Gamasutra.com* as well as Game Developer, and maintains a blog on game development at www.gamedevblog.com. He can be reached at jfristrom@gdmag.com.



GAME DATA



PUBLISHER:
Activision

NUMBER OF DEVELOPERS:
Six at inception swelling to approximately 60 at peak

LENGTH OF DEVELOPMENT:
2 years

RELEASE DATE:
July 2004

PLATFORMS:
Xbox, PS2, GameCube

DEVELOPMENT SOFTWARE USED:
Perforce, 3DS Max, Photoshop, Visual Studio.NET, ProDG



CONTINUED FROM PG 26

Neversoft, the developers of the first SPIDER-MAN game had already done some work figuring out how to make the web swinging more realistic by testing out a system that worked something like the old grappling-hook mod from QUAKE; you picked a point to attach your web to and then you would swing from there, using pendulum physics. Neversoft programmers discovered that it was difficult to pick the right point and that Spider-Man almost never went where the player wanted him to go. Knowing about that dead-end gave us the idea to utilize pendulum physics to create the sensation of swinging, but we wanted the game to pick your anchor points for you.

It may seem obvious that this is the direction we should have taken the franchise. After all, Treyarch had tried to capture physical simulation in previous games such as DIE BY THE SWORD, a game that featured simulated sword fighting—VSIM, we called it. While the game had a nice cult following, it appealed to a very small number of people because learning the system was too difficult. Furthermore, we had experimented with physical simulation with our surfing and snowboarding games and had to scuttle the experiments. We used to joke that this was VSIM Spidey and felt very optimistic about the direction we were moving in. While we did have some concerns, we were committed to ensuring that history would not repeat itself.

WHAT WENT RIGHT

1 WAITING FOR THE RIGHT MOMENT. We actually experimented with our first proof-of-concept for a new swinging system midway through the development of the first SPIDER-MAN game a few years back. It sucked because you kept sticking to walls you didn't want to stick to, and controlling it was difficult. You could learn to become somewhat proficient with it eventually, but the learning curve was incredibly steep. When we showed it to the project lead, he took the controls from us cold and started trying to use them, but he couldn't figure them out, so he red-lighted the idea right there. While this was frustrating for us, he was right to red-light it. We had already developed a handful of levels at this point, and introducing a whole new way of movement would mean redoing many of them, which was something we did not have time for.

But after our first SPIDER-MAN game shipped, we were in a perfect position to try dynamic swinging again. It would be two years before the next movie came out, and we didn't have a screenplay yet, so it was an ideal time to prototype new gameplay ideas. It was time to try salvaging the swinging experiment.

2 PROTOTYPING. The Cerny method espouses an iterative succession of prototypes, the last of which is a "shippable prototype" and the model for which the rest of your game is made. This is exactly what we did in the early days of our SPIDER-MAN 2 game as we tried to make a game with a dynamic swinging system where the webs didn't anchor to the sky.

We had a fallback position; if we couldn't make the system work in a few months, we'd go back to our old tried-and-true system from our first SPIDER-MAN game.

Bink takes the grind out of game video!

The video codec for games

Bink has shipped in 2,000 games for a reason. It is the standard for video in games - it is faster (up to 4 times faster), it uses less memory (up to 18 MB less), it has a built-in audio codec, it supports every platform, multiple audio tracks, data interleaving, alpha and RLA files (for Z-depth, normal and uv per-pixel data), and much more!

It's like having your own codec

Using Bink is like using a video codec you wrote yourself. You can, for example, play to the screen, to a 3D texture, to a back buffer, to an overlay, to a plain old chunk of memory, or whatever. Bink works the way you want it to.

RAD knows games

We have been selling middleware solutions since 1991. We can solve the problems you face, because we (or our 2,000 previous customers) have had to solve them too!



**Now shipping for
PlayStation®2!
New Japanese Manual!**

**For more information:
425.893.4300
www.radgametools.com**



©

THE BEST IN GAME DEVELOPMENT TECHNOLOGY



Pixomatic
Software
Renderer



Granny 3D
Animation
System



Miles
Sound
System

"PlayStation" is a registered trademark of Sony Computer Entertainment Inc.



At first, it was simple: We manually placed a bunch of what we thought were desirable anchor points in our level on building corners and edges, and depending on where you were pressing the joystick, when you shot out a web line, it would choose the anchor point it thought was the best. That first prototype had no animation—a stiff-armed Spider-Man would swing unconvincingly from the web. We made it so he no longer automatically stuck to walls, and that was a big help. And while he bumped into walls a lot, it felt a little more like being Spider-Man.

I'd like to define a couple of terms more rigorously. (Who knows? Maybe they'll stick and become part of game developer vocabulary.) A proof-of-concept (POC) is a videogame demo that

everyone who worked on the game in those early months was a critical part of the feedback that directed the coders to implement features and tools to make the swinging fun and controllable.

Programmers took turns owning it as we advanced through a succession of prototypes. One coder got it up and running, different coders tried different experiments with controls, another got it to blend between animation poses, another got it



exists only to prove a concept; it can have placeholder art, it can be missing animations and sounds, but as long as it shows enough promise to justify making a better POC, it's a success. A prototype, on the other hand, is a videogame demo, most of which will probably go into the final game with only minor changes.

Our POCs were promising and the development team really enjoyed playing with them. In fact, we could sit there for hours, sometimes, just swinging around our small prototype level, getting a feel for the swinging. But how would others react, if they were introduced to the new system cold?

3 ROTATING TALENT ONTO THE SYSTEM. The swinging system was very much a collaborative effort. Although sometimes one programmer would work on it alone, fairly often a group of people would be in the office trying it and making suggestions. Different aspects of the system were worked on by different people. Almost

working indoors, another added the IK animation on the arms, and so on. The anchor-point searching algorithms for which it found anchor points went through many hands and stages of revision, and the coders provided interfaces so the designers could tweak and tune the system for maximum effect.

And the system didn't exist in a vacuum either. The motion of the camera is key to making it feel dramatic, and the city itself was built to make the swinging fun, as we discovered certain widths of streets and heights of buildings that worked and others that didn't. The end result was much better than it would have been if a single person had owned the system.

4 THE ADVANTAGES OF CONSENSUS. There was one decision in particular I'd like to highlight—the choice to be able to swing from two webs. Some people on the team wanted to be able to web-swing with just one button, holding down the button would shoot out a web line and swing; and releasing the button would let go. Others on the team wanted to be able to swing from two webs, shooting out the second web before you had let go of the first, and to hang from the two webs. But we faced a challenge in

Fun-a go-go!

Mobile data is fun.

Mobile data is colliding with the Entertainment industry to bring mobile gaming, video, music and camera applications to a diverse audience of wireless users. Given that every mobile device must perform certain basic functions, consumers are increasingly choosing their products based on the fun extras that are becoming the industry standard.

Back by Popular Demand!

CTIA's Mobile Entertainment Reception

at



Tuesday, October 26, 2004
San Francisco, CA
5:30 PM - 7:00 PM

Keynotes



Day One
Ed Colligan
President
palmOne, Inc.



Day Two
Dr. Paul Jacobs
President,
DUALCOMM Wireless
& Internet Group.



Day Three
Dan Rosenzweig
COO
Yahoo! Inc.

Stay tuned for the announcement of more exciting Keynote Speakers.

CTIA WIRELESS I.T. & ENTERTAINMENT 2004 is the one industry event that effectively joins the wireless industry with the excitement of the entertainment world.

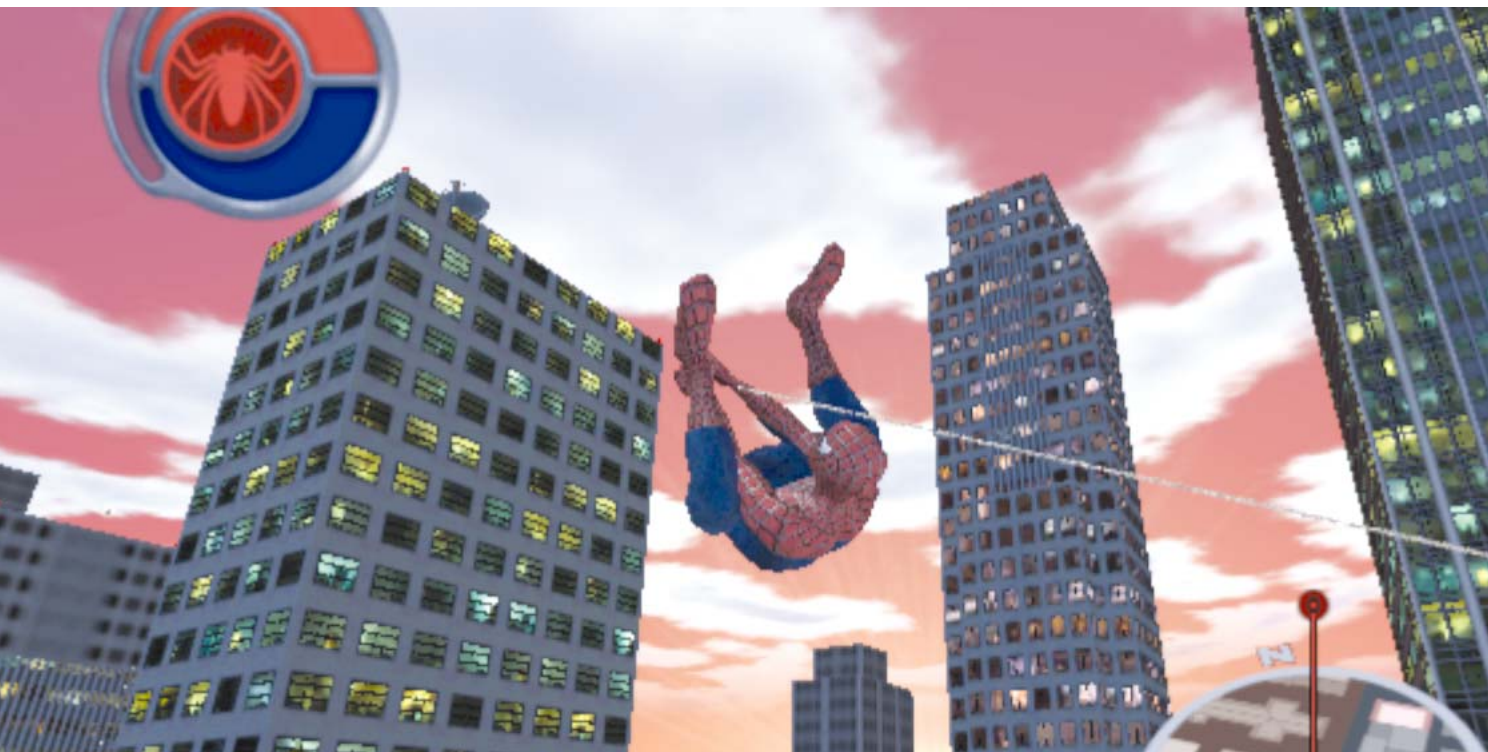
- 3 days of keynote addresses featuring a diverse and influential line-up of the industry's most sought-after speakers
- A vast show floor jam-packed with the latest in wireless products, services and entertainment applications
- 3 Tracks, 2 Workshops and 12 Breakout Sessions covering mobile entertainment technologies
- 7 Special Interest Seminars, including the return of the full-day Mobile Entertainment Summit
- 2 key networking opportunities allow you to mix and mingle with the industry leaders – Exhibit Hall Preview Reception and the Mobile Entertainment Reception.
- Mobile Entertainment Summit – This all day event brings together leading entertainment and wireless companies to discuss mobile gaming, music, sports, video, graphics, ring tones and much more.
- RFID World – the Smart Pass Executive Conference will explore how this new technology is set to transform many sectors of our economy

Register Today – and discover how the wireless industry is keeping entertainment at your fingertips.

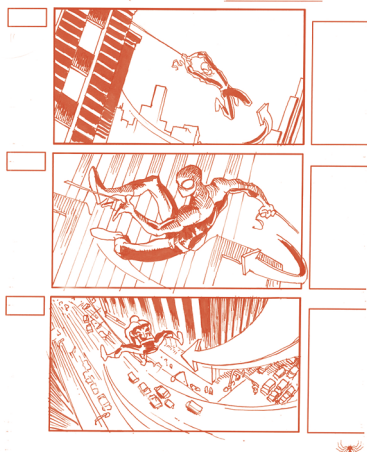
www.ctia.org

CTIA
The Wireless Association™

October 25-27, 2004 Moscone West San Francisco, CA
CTIA WIRELESS I.T. & Entertainment 2004 for everything that is Mobile Data.



SPIDER-MAN



implementing that effect without using up one of our precious buttons for the second web. So the final idea we went with was to press the button once for one web and twice for the second web. Thus, you could swing with just one button, and you could still swing from two webs.

This gave us the two webs we wanted and the one-button control. The two webs turned out to be a nice point to make in PR and marketing—showing screenshots of Spidey hanging from two webs helps show that the web-swinging is something dramatically new this time around. But it had disadvantages, which I'll hit in the What Went Wrong section.

5 HALLWAY GAMEPLAY TESTING. You

don't always need to put ads in the paper to get focus testers. If your company is big enough, you can find people "down the hallway" working on other teams, or in IT, or as administrative assistants, who will test the gameplay for you. We would poach other teams for testers, but our preference was always to find testing virgins, or "Kleenex testers," as Maxis calls them—people who had never seen the game before, who would test it just once.

Because this was a whole new kind of gameplay, we were pretty sure people would have to go through a tutorial and a learning curve before they got it. We were consciously violating the rule of keeping our core gameplay brain-dead simple because we thought the result was dramatic enough to justify it. Still, just in case a tutorial wasn't necessary, we tested on a couple of people without one in a simple POC where Spidey had

to chase a floating disc through nondescript city streets and simple block buildings. As we should have expected, they didn't get it. They tried playing the game like the previous SPIDER-MAN game, climbing up to the top of a building, jumping off, trying to swing, and ending up back on the ground again.

So our next POC had two parts—a tutorial level and the chase-the-disc level. The tutorial level gradually took you through swinging step by step: first swing from this pole, then from two in a row, then jump and swing, then swing down this street. Later we realized this was overkill, but when we started gameplay testing, we saw some good results. People were now swinging like Spider-Man. They were clumsy at first, but they got better. We got to the point where after about 10 minutes of play people were competent swingers. Ten minutes sounds good on paper—but if we were making any other game it would have probably been nine minutes too long. People don't have much patience with their videogames. We were counting on the draw of SPIDER-MAN to make people give it a chance. In other words, kids, don't try this at home. Still, we were getting to the point where we were starting to feel ready to show it to upper management.

WHAT WENT WRONG

1 EXECUTIVES TOO LENIENT? The day we decided we were ready to show our POCs to upper management, I was nearly sick with stress. All the stories I'd heard about game publishers fearing change and innovation were going around in my head, those rumors about how *THE SIMS* was nearly canceled five times and so on and so forth. Would we have to flush months of work down the drain?

It actually went incredibly well—like expecting to have to punch through a rock and encountering tissue paper. Our producer loved it and brought in his boss. And he brought in his boss. And he brought in his boss. Everyone loved it.

I can't believe I'm saying this, but maybe they were too

GAME FACE

Kulturelles Computer- und Videospiele Entwicklungsmagazin

JULI/AUGUST 2004 | NO.7

NEU!

Jetzt am
Kiosk!



the
DOUBLECROSS

© 2004, ATI Technologies, RhineFX

HAMBURG SPECIAL

HENRY JENKINS (MIT) INTERVIEW

ATI DEMO MAKING OFF

NEBULA TUTORIAL

GAME FONDS

ROCKING THE
MAKING OFF!



lenient. The problem was that it was just us demoing the system to executives—they didn't actually get hands-on experience with it—and one thing about that early swinging system was that in the hands of an expert, it looked really cool, but in the hands of a newbie it looked like a failure.

2 NOT ENOUGH RIGOROUS FOCUS TESTING IN EARLY PHASES.

Hallway gameplay testing is great, but it isn't enough. Most of the people who work at a developer are fairly hardcore gamers, and you're not going to get enough of a feel for how casual gamers are going to receive the game by only testing on people at your company. Once we started focus testing with external people, we learned a lot. Their reactions were very mixed—some of them got it, but some of them were clearly struggling, fighting the controls, bumping into walls. Some of them were so uncomfortable with the swinging they'd start running through the streets instead of propelling themselves through the city.

After all our hard work, this was hard to take, so we went into

mode, but the press-to-hold the web, release-to-let-go mechanism was self-explanatory enough that the next round of testers we brought in seemed to get it. It felt very risky to be changing our core gameplay at that late hour, but it worked, and it probably will make a lot of our players happier.

4 THE DISADVANTAGES OF CONSENSUS. On the other hand, a big, ugly dialog box comes up when you start the game, asking if you want Normal or Easy swinging. The team is split on this issue. Maybe we should never have gone for the two-web swinging, and it should have been Easy Swinging for everybody from day one. After all, you don't actually need the two-web swinging to play the game. However, it does make for a nice screenshot and it gives hardcore players something to talk about, but how useful is it?



denial. The conclusion we drew was that the tutorial wasn't good enough. (In fact, our first focus test had no tutorial at all. That was a big mistake. We postponed testing for a month so we could get a tutorial in before the next focus testing session.) We would have to work on it and keep testing and repeat until we captured as many people as we could.

3 IT'S NOT THE TUTORIAL, IT'S THE SYSTEM. So that's what we did—continued working on the tutorial and continued to bring in more focus testers and people started to get it and have fun. But still, even with all that, there was at least one person in each test session that seemed to be struggling. One focus tester even said, "I liked it better the old way; you could just cruise along without really paying attention." Even though the other focus testers disagreed vehemently with him, it was still worrisome.

Some of us thought we should just write off those dissatisfied customers—you can't please everybody all the time—but common sense prevailed and we eventually agreed we had to do something. The problem was not that the tutorial wasn't good enough. The problem was that the system was too hard. This was the birth of our Easy Swinging mode, which worked just like the one button swinging described in the Advantages of Consensus section. We were getting close to our submission date and there was no time to rewrite the tutorial to handle the easy swinging

THE LESSONS OF PROTOTYPING

It may be hard to apply the lessons we learned here to your game. If your game represents an evolution on a previous title rather than a revolution, you probably don't need to spend the kind of time and effort on prototyping that we did. If your game doesn't have the power of a strong license behind it, you may not have as much leeway to invent something with a steep learning curve. And if you don't have the kind of resources that a blockbuster title like a SPIDER-MAN game has, you may not be able to pursue this type of extensive prototyping.

Our SPIDER-MAN 2 game is just one more example in a series of games that had a prototyping phase and were successful. I'm referring to games like JAK AND DAXTER, RATCHET & CLANK, DEUS EX, PRINCE OF PERSIA, THIEF, and HALF-LIFE. (Although those last two examples didn't know they were prototyping when they were.) Also, to use corporate speak, the opportunity justifies the risk. If you can create a new kind of core gameplay you can create a new franchise that will have life beyond the current game. And if your game has unique gameplay, you will have no competitors—at least for a while. ❖

SERIOUS GAMES SUMMIT

WASHINGTON D.C.
OCTOBER 18-19, 2004

D.C.

WWW.SERIOUSGAMESSUMMIT.COM

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 are applications of interactive technology that extend far beyond the traditional videogame market, including: training, policy exploration, analytics, visualization, simulation, education and health and therapy.

The Serious Games Summit D.C. gives game developers the opportunity to tap into this new revenue stream, learn from successful serious games applications, and forge links between the traditional videogame industry and program managers for homeland security, state and local governments, military agencies, and educational institutions.

REGISTER BY SEPTEMBER 15, 2004 AT WWW.SERIOUSGAMESSUMMIT.COM AND SAVE \$200 ON YOUR PASS



SEAN BARRETT

THE INNER PRODUCT

OPENING DOORS

THE MONTY HALL SO-CALLED PARADOX

in 50 words: Pick one of three doors. One door opens on a valuable prize; the other two hide ravenous hordes of trilobites. After you choose, one door—not yours—is opened, revealing half of the trilobites. Now you have a new choice: keep your door, or switch to the other unopened one.

Most people feel like it comes down to a 50/50 chance for the final question; since they went with their gut instinct the first time, why change their minds? But switching doors actually has a two-in-three chance of gaining the prize, since there was only a one-in-three chance the initial guess was right. This may be surprising, but it's not really a paradox. The real paradox is how hard it is to find a convincing explanation of this fact.

I didn't introduce the Monty Hall paradox to explore the psychology of probability, but rather because we programmers tend to stick to the old, familiar doors, even if standing by them is actually more likely to introduce bugs into our code than switching to a "better" door. And given all the bugs we've seen behind those doors at other times, no wonder we're scared.

Bugs in software generally arise from complexity in control flow and complexity of state. A design that requires more "if" statements is likely to have more bugs from incorrect conditionals; designs with more variables offer more opportunities to fail to maintain invariants between them. Programs with little flow control or state, e.g. scripts, offer few chances for bugs. Not much can go wrong if your game script says `playsound blorp;`

create 3 UltraWraiths in `inner_sanctum`; destroy `central_forcefield`, except maybe an extreme typo that the compiler can't catch, leading to three unintended UltraWaifs.

ENEMY OF THE STATE

There are ways to avoid bugs caused by complexity of state. Functional programming tries to do away with bugs involving state by doing away with side-effects. Object-oriented programming uses encapsulation to ease maintenance of invariants.

Or, state can be embraced. One project I worked on made as much state as possible explicit in the form of state machines. Everything from GUI sliders to animating sprites to major game modes inherited from a common state machine abstraction. A Turing machine is essentially a state machine, so clearly state machines can do anything. But they're not always the right tool for the job. When state machines have many states, it can be difficult to understand what they do unless you draw a flow diagram. This might be appropriate if a problem is really that complex, but sometimes a state machine unnecessarily complicates straightforward control flow.

Recently, I designed a state machine to try to get an AI through a closed door. Opening and closing the door was easy: A simulation swung it open once the AI

triggered the door's opening/closing interface. The complexity arose from getting the AI to be near enough to the door to trigger it, to be out of its way while it opened, to go through the doorway, to close it afterward, and to handle failures. On the surface, this is still easy; the overall state machine diagram is shown in Figure 1. However, there's a lot of complexity hidden inside the edges and the nodes—enough so that I was having a lot of trouble understanding and maintaining the state machine code. Frustrated with the bugs, I decided there had to be a better way. After some experimentation, I decided to try cooperative multithreading.

COOPERATIVE MULTITHREADING

Traditional multithreading is pre-emptive: one thread runs for a while, then the OS stops it and switches to another thread. This can happen at almost any time. Two threads from a process can interleave their control flow in arbitrary ways—a dangerous scenario to start with—and if the two threads access the same state, things can go very, very wrong. Traditional multi-threaded programming requires a lot of attention to synchronization techniques to make sure threads and state interact properly, and doing this efficiently can be hard: The double-checked locking pattern was determined not to be thread-safe in Java [see For More Information, page 38]. Except in limited cases, such as access to hardware resources or multiple processors, pre-emptive multithreading is more complexity than it's worth.

But just because the traditional multithreading door is dangerous to open doesn't mean we shouldn't

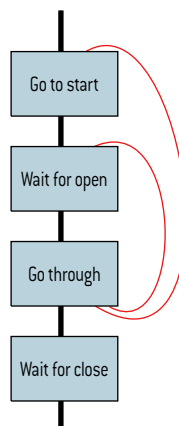


FIGURE 1 (door_states.png): State machine for opening a door. Red lines indicate failure conditions which retry earlier steps.

SEAN BARRETT is a text-adventure enthusiast and co-organizer of the Indie Game Jam. He lives in Oakland, Calif. Reach him at sbarrett@gdmag.com.

consider a third door: cooperative multithreading. Cooperative threads explicitly yield control to other threads, rather than become spontaneously pre-empted by the OS. The concerns about pre-emptive multithreading mostly don't apply, as there's no unexpected control flow. Threads must be at some (usually small) number of explicit synchronization points when there's a thread switch.

To demonstrate cooperative threading, Listing 1 shows the threaded code into which I re-coded my door-opening state machine. It's a fair bit of code because this conversion is only valuable when the state machine implementation is large.

To begin, in line 3 the AI records whether the door was open or not at start. If it was closed, the AI will close it once it's through. In line 6, the AI picks a location clear of the door as its target. Then the AI calls a routine, which makes it head straight toward its current target. This routine runs for multiple frames, returning when the target is reached. Note that this routine's name ends with `_YIELDS`, which makes it easy to check that all loops contain a yield statement somewhere.

Once the target is reached, the AI starts opening the door (line 11) and waits for the length of time it expects the door to take to open. The implementation of waiting starts at line 48. If neither of two failure cases (lines 15 and 17) occurs, then the AI is ready to move on. If things go wrong, the failure system is updated. The failure system is a self-monitoring system in the AI that allows it to detect if it's not making progress toward goals, allowing it to abandon them. The thread code simply updates failure state. The main game thread, which is responsible for driving the main AI process, will kill the thread and start a new task if too many failures occur. In this case, if the door is closing (presumably because someone else started closing it, most likely the player trying to mess up the AI), then the AI clears any failures that may have occurred so far. The fact that the AI is failing to make progress toward this goal doesn't mean it should give up. In either case, the AI makes sure the door is opening, and tries again.

With that incomplete description, let's compare the state machine approach. Each call to the function `yield` corresponds to a place in the state machine where the code would return from the state machine and come back to it later. In this case, the main function has no explicit calls to `yield`; they're all buried in functions,

which we couldn't easily do in a state machine. Instead, the `Goto Start` state would look something like this:

```
case STATE_goto_start:
    if (m_ai->driveToTarget(m_target)) {
        // arrived at target!
        m_ai->openDoor(door);
        m_state = STATE_wait_for_open;
        m_end_time = getCurTime()
            + door->openTimeEstimate();
    }
    break;
```

The code assumes there's a unique class implementing the opening-door state machine. All the local variables in the original implementation, like `started_open` and `end_time`, have become member variables.

This code is less clearly structured than the cooperatively-threaded version. `STATE_goto_start` begins with a call to `driveToTarget`, but it's not obvious what that target was. The call to `doorRelativeTarget` was moved onto the edge leading to the state, whereas in the cooperatively threaded version, the two lines (6 and 7) are adjacent. (Where does that incoming edge originate? There are actually two: one from `Go Through` and one from an implicit start "state" that would run when the state machine was constructed.)

Another drawback of the state machine code shown above is that instead of limited, clearly scoped variables, the use of instance variables (such as `m_end_time`) removes the obviousness of when and where the values are meaningful. The code has a potential bug: You could set the future state to `STATE_wait_for_open` without setting `m_end_time`. That's a bug I actually had for the case corresponding to line 23 but didn't discover until I converted the code to use threads. You could get around this risk by making pseudo-constructors for each of the states, so you would call `switchToWaitForOpen(time)` or

Listing 1

```
01 void AI::goThroughDoor(Door *door)
02 {
03     bool started_open = door->isOpen();
04     restart:
05     // Goto Start
06     doorRelativeTarget(door->length+m_obj->radius);
07     driveToTarget_YIELDS();
08
09     reopen:
10     // Wait For Open
11     openDoor(door); // start door opening
12     waitFor_YIELDS(door->openTimeEstimate());
13
14     for(;;) {
15         if (door->isClosing())
16             resetFailureCount();
17         else if (door->open_amount < INWARD_OPEN_MIN)
18             incrementFailureCount();
19         else
20             break; // success
21
22         openDoor(door);
23         waitFor_YIELDS(0.5);
24     }
25
26     // Go Through
27     doorRelativeTarget(-(m_obj->radius + 0.05));
28     driveToTarget_YIELDS();
29
30     while (distanceFromTarget() > 0.1) {
31         if (door->open_amount < INWARD_OPEN_MIN)
32             if (door->blockedBy(m_obj))
33                 goto restart;
34             else
35                 goto reopen;
36
37         if (door->isClosing()) {
38             resetFailureCount();
39             openDoor(door);
40         }
41         driveToTarget_YIELDS();
42     }
43
44     if (!started_open) {
45         closeDoor(door);
46         // Wait For Close
47         waitFor_YIELDS(DOOR_CLOSE_TIME);
48     }
49
50 void AI::waitFor_YIELDS(float wait_time)
51 { // "Sleep(time)"
52     float end_time = getCurrentTime() + wait_time;
53     while (getCurrentTime() < end_time)
54         yield();
55 }
```


introduce a new subclass for each possible state with appropriately limited instance variables. This requires constructing a lot of scaffolding, especially wasted on states with only one incoming edge. It wouldn't allow code-sharing either. Many different state machines might want to have timed waits in different states, so even though you can modularize `switchToWaitForOpen` for this state machine, it's hard to abstract out the waiting math and logic in a reusable way, unless you start nesting state machines (with one state machine invoking another which does the waiting). While this is possible, it becomes even harder to follow what's going on, especially compared to the straightforward code in the cooperatively-threaded version.

As bad as that is, the above example only describes a simple state, not a complex one. The more complicated states will be messier in the state machine version, although perhaps not proportionally; a certain amount of this is fixed overhead per state and per outgoing edge. Complex computation within a state wouldn't become more complicated in the state machine implementation. The in-state logic, however, would be abstracted from loop control flow, which would be returning and re-entering instead of yielding. As a result, the consequences and interactions would be harder to track.

State machines also exhibit plain old code bloat. The code for `goThroughDoor` is long, but will fit on a single screen, whereas a state machine version probably wouldn't. I expect state machines will often be two or three times larger than equivalent cooperatively threaded code.

IMPLEMENTATION ISSUES

There are two ways to use cooperative multithreading: in native code or in a scripting language. The latter is probably the only likely way of shipping an AAA title using cooperative multithreading, so let's consider the problems and see why native code is more difficult.

I described how, after repeated failures,

my AI system will reject the current task, freeing the thread representing it. Typically, this means freeing the stack used by that thread. However, if the thread has allocated any memory or other resources, those need to be freed, and unfortunately it's clumsy to do that with native code. For example, Microsoft Windows provides cooperative threads with the Fibers API, but deleting a fiber does not call destructors on outstanding objects on the stack. To free the thread cleanly, you'd need to set a global flag and run the fiber. You would wrap `yield` to check that flag and throw an exception to unwind the stack cleanly.

More difficult is saving the game. A state machine has completely explicit state, which can be serialized easily. A thread's state is embedded on its stack, and, again, might include arbitrary pointers to objects or allocated memory. This state would need to be saved and restored, and execution continued with that stack would need to be restored and active. Serializing the stack is unlikely. Even if natively compiled code is disciplined and uses handles to objects to facilitate it, the compiler can still move raw pointers into the registers saved on the stack. Restoring a thread with a live stack is possible if you roll your own cooperative threads, but unlikely to be supported by existing APIs.

The only plausible native-code approach is not saving threads at all; instead, systems that use threads must restart after a game restore. For example, my (native code) door-opening system uses some additional omitted startup code to skip unneeded steps, allowing AIs to continue from the middle after a restore. However, an AI navigating a door during a save will end up forgetting to close that door after a restore because the `started_open` flag isn't saved.

An interpreted scripting language makes this task much simpler. The interpreter and run-time environment can be designed so that threads are easily freed and are fully serializable. On the other hand, an interpreted language may be harder to

debug than native code. It also means your programmers are using your scripting language for low-level path-following, which may put unfortunate pressures on a system primarily intended for high-level designer scripting. Still, it's more viable than native code.

Once you've opened the cooperative multithreading door, you may discover new possibilities. For a game design that featured individual, distinctive characters driving vehicles, I broke down the vehicle AI task into four separate roles—captain, pilot, navigator, and gunner—each handled with its own cooperative thread. This may have been overkill—some of them could have been reactive instead of threaded—but it gave me a lot of freedom to experiment and include degenerate (stateful) strategies for individual roles.

Game software development is unlike most software development in many ways. We have real-time-like demands (but not like embedded software). We have severe serialization requirements (save games), but we don't expect to maintain our games two years after they ship, unless they're online. These factors tend to cause a certain level of mismatch between our needs and the tools available commercially, from inappropriate profiling tools to the inability to serialize threads. However, that also means there are probably techniques out there, like cooperative multithreading, that are largely abandoned but would be useful to us if only we knew about them. The only way we're going to find them is to open more doors. ❌

FOR MORE INFORMATION

Pugh, Bill, et al. *The "Double-Checked Locking Is Broken" Declaration*.
<http://www.cs.umd.edu/~pugh/java/memoryModel/DoubleCheckedLocking.html>

Knuth, Donald E. *The Art of Computer Programming, Volume 1*. Section 1.4.2: "Coroutines." Addison Wesley Longman, 1997.

game**developer** presents

FALL 2004

GAME CAREER GUIDE

>> **THE M.O. OF H.R.**
HOW THE HIRING
PROCESS WORKS

>> **THE TEACHING DILEMMA**
THE BIGGEST PROBLEM
FACING SCHOOLS

>> **100 GAME DEV. PROGRAMS**
DIRECTORY OF
HIGHER EDUCATION

IS A GAME DEVELOPMENT EDUCATION WORTH IT?

REFLECTIONS BY RECENT GRADUATES
WHO MADE IT



DISPLAY UNTIL NOVEMBER 11, 2004



PIXEL PUSHER

STITCH, CUT, AND RELAX

GETTING THE MOST OUT OF AUTOMATIC TEXTURE MAPS

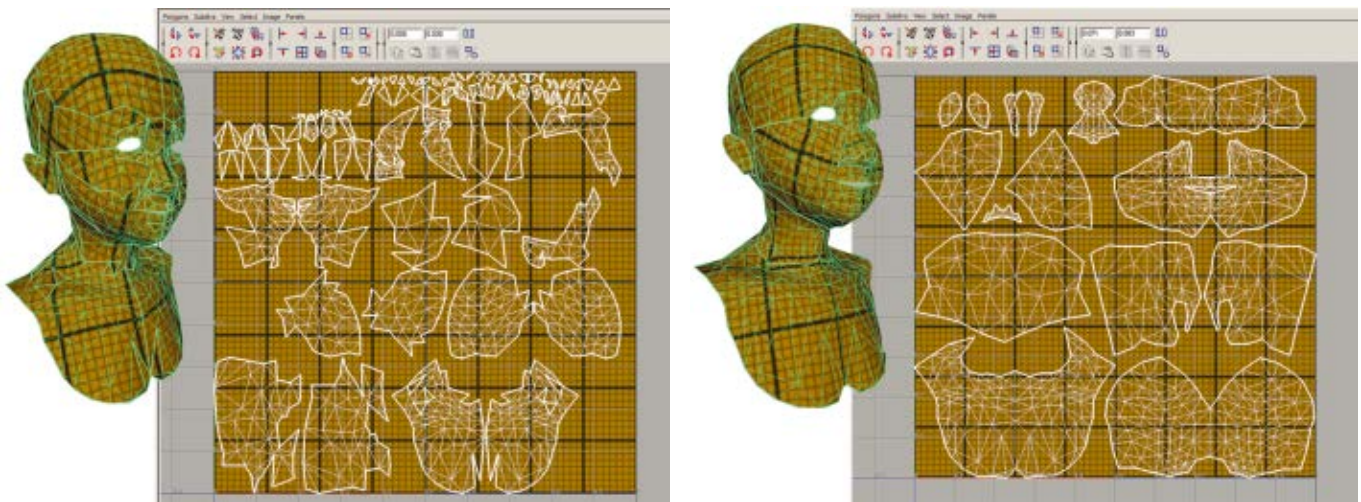


FIGURE 1 Automatic mapping systems generally produce very fragmented UVs (left). To turn them into usable UVs, you need to clean them up by hand (right).

GOOD UV MAPPING ISN'T PRIMARILY about technology or technique—at least if you buy the argument I laid out before in “Maps and Legends” (June/July 2004). Instead, UV mapping strategy is about making good technical tradeoffs. For the last few product cycles, the texturing features that get all the bullet points in new releases of Max and Maya have all been centered on increasing autilism—trying to make sure the texel density on a model is the same everywhere, which pretty much eliminates streaks and smears. However, as we saw in the June/July column, highly autilic maps have some real drawbacks as well:

They're often impossible to read; they can have awkward seams; and perhaps worst of all for real-time games, they're inherently inefficient. So this month, we're going to look at some tricks for using automatically-generated, stretch-free maps in real time applications.

First a quick recap. “Autilism” means every part of a model gets the same amount of texture space—in essence, every texel on the finished model is the same size in world space as every other. If all you care about is a high quality render, autilic is the most important value in mapping. Typically, though, autilic maps don't make efficient use of texture memory. Moreover, in game production, every texel on a model comes with a cost both in memory use and texture upload time. Wasting texture memory on unimportant parts of a model

is a bad idea. For example, a character's face is more important than the soles of his shoes; or the mechanical doodads on the back of a tank matter more than the blank plates on its underside. If we're going to get good use out of highly autilic maps in real time, we're going to need techniques for maximizing efficient packing and for allocating texels where they're needed while preserving the overall quality of the mapping.

AUTOMATIC FOR THE PEOPLE

The easiest way to start is with an automatic mapping system like Maya's Automatic Mapping or Max's Flatten mapping. “Automatic” implies that you just push a button and—voila! A finished UV map appears like magic. In reality, the

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.

automatic mapping process is just the beginning. Automapping always produces a lot of oddly shaped UV shells, and frequently, a lot of stray UV triangles as well. Since you need to leave a border around every UV shell to account for MIP-map bleeding, individual triangles are a bad use of texture space. It's worth cleaning them up even if it introduces a few stretch marks into your texture. Therefore, the first task is to clear up as many of the strays by using the UV "stitch" or "sew" in your package (see Figure 1).

Once you've gotten rid of the random triangles, you'll need to adjust the seams created by the automap so they go into unobtrusive locations. Cutting the UV edge within a shell will introduce new seams. But for reasons that will become clear in a moment, they're fairly unobtrusive. You can cut up the shells generated by your automapper and reassemble them into more logical ones, both for better packing (think nice square chunks instead of crazy Los Angeles-shaped sprawls) and to keep the seams in innocuous places.

At this point, the textures are still the same density across the whole model, the product of the original automapping. To get more texels into the important parts of the image, we'll need to scale up the UV shells in the high-resolution areas. This means that you'll need to cut seams that define the boundary between your high- and low-resolution areas—for example, the line of edges at the top of the neck might mark the border between the high-density texture of a face and the lower resolution of the body. The scaling itself is simple. Grab the UV shells that make up the high-resolution areas and scale them up in your UV editor. The really difficult part is not the execution; it's deciding how much to scale up the high-resolution UV areas. It's often a good idea to tweak this relationship iteratively, by repeatedly scaling the UV shells up appropriately and then using your package's automatic

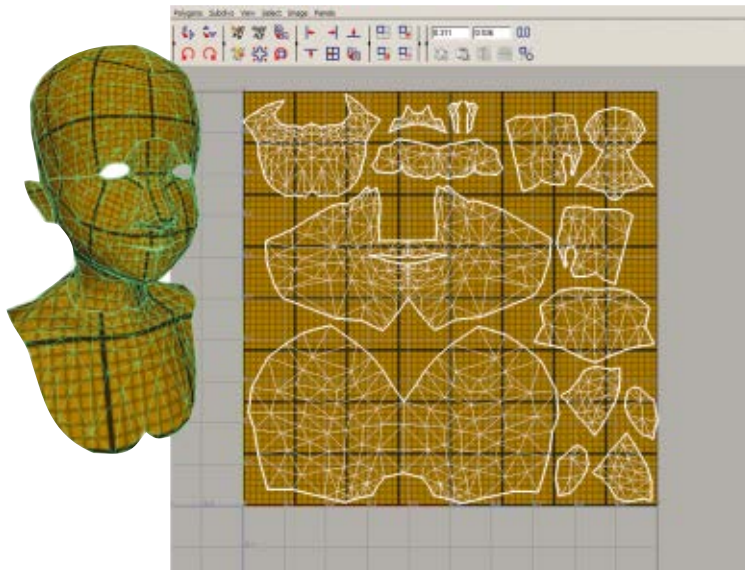


FIGURE 2 The original automap has been cleaned up for better seam placement, and the UVs on the face have been scaled up to add more resolution there.

UV layout function. This will be somewhat misleading because automatic UV layouts are never very good, but you can use them as a quick aid to finding the right balance between your high- and low-resolution UV shells.

Once you've got a relative distribution that seems to make sense, go back and hand-pack all the UV shells as tightly as you can (remembering, of course, to leave enough room between pieces to avoid bleed-through at lower mip levels). Typically, hand packing can save as much as 12–15 percent over automatic packing, so it's very worthwhile, even though it's also pretty tedious. As you can see in Figure 2, this method results in a reasonably smear-free map that still puts texels where they're needed the most. The map is ready to be painted in a 3D paint app or a projection-paint method. It is unlikely that you'd be able to paint it using a



FIGURE 3 Seams are less noticeable when the underlying pixel grain is aligned. The seam between the gray and orange areas is aligned in UV space, the rest are not.

traditional texture-sheet method, since automatic mappings rarely produce maps that can be hand-painted in 2D. There are some tricks for getting around that without a 3D paint package, but we'll have to save them for another column.

ON THE SEAMY SIDE

In the bad old days, the transition from the high- to low-density areas wasn't particularly glaring amid all the other artifacts that we used to live with. When entire characters had to be squeezed into 128KB of texture memory, ruthlessness was usually pretty easy. Alas for the poor artist of today, higher texture resolutions overall mean that artifacts we used to be able to get away with are now much more noticeable. Moreover, the latest advances in texturing techniques seem to punish minor imperfections very severely—the obvious seam running across your beautifully normal-mapped, procedurally-textured, 3-pass shaded surface just feels a lot worse than a similar seam in an old

fashioned 256 pixel color map.

If we want to minimize the downside of seams, it's worth asking what actually makes a seam visible in the finished model. Three factors contribute to the visibility of a seam. The most important factor is texture density, which we'll return to in a moment. However, a second noticeable factor is orientation, which is less critical but worth a quick digression. As you can see in Figure 3 (page 41), it's not hard to spot the seam when adjacent triangles have grains that run in different directions. The final factor is one we can't do much about: the fact that you can't "snap" every UV edge to the pixels of the texture, so partial texels will peep through along seams. Luckily, this artifact scales down directly as texture resolution gets better, so the best way to fix it is to make sure your textures are efficiently packed.

Both Max and Maya offer tools that allow you to stitch UV edges together,

automatically moving, rotating, and even distorting the original UV shells to make the connection. These tools are an excellent aid for fixing the grain and density mismatches, which lead to visible seams. If you have UV shells that can't be connected in the final map (say, because of an overlap that would turn the output map into gibberish), you can use the UV-stitching tools to connect your shells and then immediately cut the seam you just made. You can then move the two shells apart and the edges will remain aligned in UV space, unless you distort or rotate one of the shells. If you do need to rotate one or both of the shells in the course of packing, try to avoid juxtaposing shells at 15-, 30-, or 45-degree angles (remember that the human eye is peculiarly sensitive to multiples of 15 degrees).

Since stitch-and-cut is so handy at dealing with seam artifacts, can we use it to mask the transition from high- to low-resolution areas, like going from the head to the neck? Unfortunately, this isn't likely



FIGURE 4 [1] A harsh transition between a high-resolution area: the face. [2] A low-resolution area: the seam is stitched, but the join is distorted by the resolution change [note the shape of the "6" under the ear]. [3] The distortion is spread throughout both areas with a relax operation, hiding the transition.



to work because of the way most graphics cards work. Older cards and mid-range hardware today use tri-linear filtering to slap textures onto triangles. The math is complicated, but the resulting resolution of the texture inside a triangle is only about as good as the resolution along the triangle's lowest edge (high-end cards that support anisotropic filtering will come off better, although even there, one really under-strengthened UV edge can still make the texture inside a triangle look pretty muddy). Consequently, stitch-and-cut is best used when the texel densities around the stitch are very close. Since at least one edge of every triangle in the border between a high- and low-resolution shell will have a very different resolution, stitching and cutting will produce triangles that look muddy or smeary at best. At worst, the border triangles may be saw-toothed due to bad filtering. In either case, it's unlikely to be less noticeable than a simple seam.

FRANKIE SAY RELAX

Now, texture painters know that the best way to hide an unwanted transition is to blur or feather it, making the transition less sharp by spreading it out over space. To get a similar effect with UV maps, we can combine the stitch-and-sew technique with UV relaxation. As I mentioned in the June/July issue, relaxation (a strangely out-of-place word for a topic like this) treats the texture space within a given UV shell more or less like a slightly stretchy rubber sheet. The relax operation attempts to smooth out the "wrinkles" that come from wrapping a 2D texture onto a 3D surface. In mathematical terms, UV relaxation tends to average out errors throughout a UV shell—in essence, feathering. To hide a resolution change, try stitching the seams and then running a UV relax on the combined shell. If your package offers the option, remember that you want the edge weighting in the relax operation to be based on the world space length of the edges. This will distort all the UVs within the shell, but if your shells are large enough, the individual errors will be hard to spot (see Figure 4). Run the relax through enough iterations that the UVs inside the shell are stable (the algorithm actually works just like a physics spring simulation; it

takes a few iterations to spread out the tension to a stable configuration).

After the relax operation, none of the triangles in either shell will be perfectly authentic anymore. Ideally, though, you'll be spreading the "errors" in texture density across enough vertices that the variations in density will be hard to spot. Probably the biggest problem comes from weirdly shaped UV shells—it's possible for the UVs inside a highly concave UV shell to actually bleed out of the shell itself during a relax operation. If this happens, you'll need to subdivide the shell before running a relax. There will be some models that just offer shells which are big enough to hide the errors—generally, if you don't have at least three ranks of UV vertices to either side of the seam, it's not worth trying to hide the resolution change with relax and you should just accept the seam instead.

That's a quick sketch of how to leverage automatic mapping for real-time textures. Automatic mapping is a very powerful tool for generating good UVs. However, it has several drawbacks when you try to use it in real-time applications. By carefully managing seams with stitch-and-cut techniques, you can retain the high quality of the automatic maps and still pack them efficiently for real time. Adding UV relaxation to stitch-and-cut lets you feather out the transition between triangles with different texel densities. Of course, the other major drawback of automatic mapping remains: It creates texture sheets that are almost impossible to paint with a simple 2D paint package. We'll return to dealing with that problem in the near future. In the meantime, just relax! ❖

SECUREPLAY

Stops Cheating Before It Starts

Works with any type of game
Available via CD or Download

Platform and Network Independent
C++, Java, J2ME, Flash

Source Code provided with product purchase. Fully documented SDK/API with pluggable communications.

Build game security with a simple API for standard game transactions. SecurePlay delivers a secure foundation for everything from FPS and RTS to simple card and board games. Supports both P2P and client-server.



HAVE ANY HOLES



IN YOUR GAMES?

Security Services

SecurePlay engineers can help you stop predators, so you can focus on building a great game.

- Solutions for DRM, Licensing Security & Revenue Protection
- Game-Specific Cheating Solutions
- Traditional Computer & Network Security, High Availability, Disaster Recovery & Contingency Planning
- System Architecture Design & Project Management



www.secureplay.com
1.202.332.5070
Email: gaming@secureplay.com

STOPS CHEATS



PAUL HYMAN

NECESSARY EVIL

ADVERGAME ASPIRATIONS

THERE WAS A TIME WHEN ADVERTISERS could buy into any kind of advergame they chose, as long as it was of the hard-coded, product placement variety.

For example, when Sony Ericsson wanted to tout its new smartphone, it paid Ubisoft to have Sam Fisher, the hero of Tom Clancy's *SPLINTER CELL PANDORA TOMORROW*, flip open a P900 mid-game.

But games that typically take 12 to 18 months to develop are not necessarily the best vehicles for advertisers who develop campaigns around, say, the new *Spider-Man* movie or the latest CD. Unless, of course, the in-game advertising can be changed dynamically to run for just a month or even a week or two.

In October, New York-based Massive Inc. will launch its new network, designed to serve ads onto in-game billboards for whatever length of time or frequency the advertiser wants. In effect, it's like buying commercial time on TV.

"Advertisers want to buy videogames in the same way they buy other media," explains Massive's CEO Mitch Davis.

"They want to be able to change the creative, run campaigns, and measure reach and frequency."

Massive sells ad time based on 15-second units. "If you're whizzing around a track in a racing game," says Davis, "you might see an ad on a poster for one second, 15 times. Or, if you stop in front of the poster and it's on the screen for 15 seconds, that's one ad unit." Davis admits the rates are high—between \$20 and \$30 CPMs (cost per 1,000 impressions). The first game to employ Massive's technology will be *SPLINTER CELL III* for PC, scheduled for year-end release; other Ubisoft games are "in discussion."

That's fine for advertisers who are satisfied with having their message on a billboard, says Dave Madden, executive vice president of sales and marketing at Wild Tangent. "But, for many brands, that's not the level of integration and depth of customer relationship they're looking for."

Instead, the Redmond, Wash.-based developer custom publishes branded advergames to advertisers' goals and specifications.

For example, Wild Tangent built a game called *RACE THE PROS* in which co-producer Fox Sports pulls the times from each Friday's NASCAR qualifying races and pumps them into the game, which can be downloaded from the Fox Sports web site and is promoted during every NASCAR race. Throughout the week, gamers can race against those qualifying times in one of two vehicles, both of which are replicas of actual Dodge-built NASCAR cars.

"What you get is a living, breathing advertising campaign for Dodge, which has done both a broadcast and a game-buy with Fox," says Madden. "But the consumer isn't aware of any of that. All the gamer knows is that he's getting a really cool, free, console-quality racing game."

Coincidentally, *RACE THE PROS* has an element of dynamic product placement of its own, similar to that offered by Massive. Before gamers can play, they need to key in their zip codes. Then, as the player's car speeds around the NASCAR track, Dodge updates the ads on

the billboards, but not with just any promotion. Based on the zip code, Dodge inputs local advertising for the dealer nearest the player's home.

"Let's not kid ourselves," says Madden. "The experience of playing that sort of Dodge racing game vs. glancing at a billboard in somebody's football game is a totally different experience."

In general, that experience costs an advertiser seeking to build a custom game between \$250,000 and \$500,000 for game development that takes three to five months.

"That's what it might cost for one 30-second spot on a prime-time TV show, not including the cost of filming the commercial," says Michael Goodman, senior analyst at The Yankee Group. "We're talking about a terrific return on your investment."

According to Goodman, the growth in advergaming results from, among other things, advertisers not feeling they're getting their money's worth on TV.

"TV ratings are declining and digital video recorders enable you to skip the commercials entirely," he says. "Advertisers just aren't getting the bang for their buck that they used to. Meanwhile, there are 108 million people in the U.S. playing videogames. It's a huge market."

In a recently released report, Goodman tallies the advergaming market at about \$79 million: \$10 million for in-game product placement plus \$69 million for branded advertising, for example, Wild Tangent's. He anticipates that by 2008, in-game product placement will grow to \$92 million while branded ads will leap to \$259 million.

"The ad money will provide publishers with an additional revenue stream," he notes, "which is just the sort of thing publishers need at the moment. The cost of game development has skyrocketed, and this is still very much a hit-driven business; only 20 percent of games ever break even." ❖

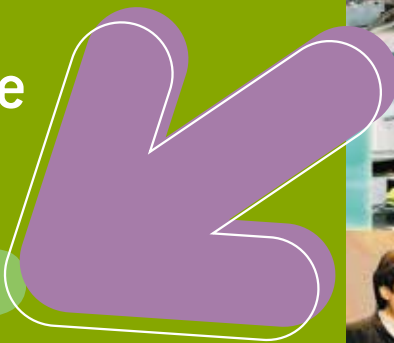


The artfully applied slogan, "See your local Dodge dealer," on the back of the car is reinforced on the racetrack with a billboard, which advertisers can customize based on their local businesses and your zip code.

PAUL "THE GAME MASTER" HYMAN was editor-in-chief of CMP Media's *GamePower*. He's covered the game industry for more than a dozen years. Reach him at phyman@gdmag.com.

Don't miss out on the

Best of GDC



Available now at:

► www.gamasutra.com/store

GDC 2004 post-show coverage:

NEW ► "Best of GDC" DVD Includes 1997–2003 - \$59

NEW ► GDC 2004 Audio proceedings
\$99 per track \$399 for a set of six

Get it all at www.gamasutra.com/store



Gamasutra.com

THE ART & SCIENCE OF MAKING GAMES

Sponsored by:





ALEXANDER BRANDON

✦ AURAL FIXATION

WHAT BUZZES THE MASSES



PARAPPA THE RAPPER relies on adaptive audio, an inherent feature of its gameplay

OTHER THAN THE marketing staff of a game publisher, the managers on a team need to understand their target audience more than anyone else. Some managers will be interested in the game's genre, and some couldn't care less. Music and sound are changing how the public at large looks at games, however, and it's time to take a long, hard look at the

cool audio features we're fighting for to see whether they really have marketable value.

ADAPTIVE AUDIO

One of the biggest buzz terms in game audio is "adaptive audio," also known as "interactive audio." If you are unfamiliar with these terms, you can check out an article I wrote on the subjects at www.iasig.org/pubs/features/adaptaudio/adaptiveaudio.shtml.

Without a doubt, a soundtrack that adapts and responds to player actions is a great idea for a more unique (and potentially more dramatically effective) experience, but how many people notice such a thing? More important, how many people will buy a game because of this feature?

For games that rely on adaptive soundtracks as the foundation for the game's design (such as *MTV Music Generator 3*, *Rez*, and *Parappa the Rapper*), no one will buy these games without a good soundtrack because the games revolve around the soundtrack. For other games, the answer is more difficult, but one recent discovery I made is that game music in reviews is judged on two areas

of quality: the quality of the music production itself (a soundtrack that uses a live orchestra will almost always score high marks even if the music isn't adaptive), and the adaptive nature of the soundtrack. An example can be found in *Cinescape's* review of *Spider-Man 2*. "The music was composed for the game and is dynamic based on the situation." *Gaming Age* adds to this by saying, "Even the movie score is mixed in well with all new music for the game." *Cinescape* is a magazine mostly devoted to films, so the example demonstrates more mass media paying attention to adaptive audio.

Reviews are one thing, public response is another. To figure out whether the average player is affected by adaptive soundtracks, we would need a questionnaire of some sort. (Now that I've mentioned that, expect to see a poll released through *GameSpy* in the next couple of months.)

Regardless of whether it brings in more sales because of a solid marketing report, adaptive soundtracks are here to stay. My suggestion: Combine a rock solid adaptive design with a killer orchestral soundtrack. Get the best of both worlds. Take a revamped version of *Wing Commander* and you'll see some very impressed game players.

SURROUND SOUND

Surround sound is another area of audio with a lot of buzz. DTS and Dolby are locked in a battle not unlike the one between Agent Smith and Neo in *The Matrix: Revolutions*, neither truly gaining the upper hand, but both releasing new technologies every few years to expand their market penetration.

Dolby was the first company with surround sound to approach games. The problem was the company didn't have a real-time solution. For example, surround sound might be in full-motion videos, but not in gameplay, so few publishers took an interest. This has changed to the point where nearly every title for Xbox and most for GameCube and PlayStation 2 have either Dolby Digital, Dolby Pro Logic II, or DTS in real time.

But do reviewers notice it? Let's look at

a review of Namco's *Tales of Symphonia* by Planet GameCube: "Unfortunately, *Tales* doesn't support Dolby Pro Logic II, but sound quality is quite high." *GameSpot* does not comment on surround sound. *GameSpy* mentions this in a preview for the upcoming console Xbox 2: "Sound plays a huge part in many of today's games, as hearing an enemy sneaking up behind you can be the difference between life and death. The original Xbox had the most impressive audio capabilities of the three consoles, utilizing Dolby's powerful Dolby Digital 5.1 technology to create immersive, multichannel soundscapes. Gamers have loved the Xbox's audio qualities since day one, so nearly everyone expects Microsoft to continue [its] partnership with Dolby. While there hasn't been anything more than speculation to date, you can rest assured that the audio capabilities will be even more impressive than they were in the previous console."

Now that we have established that reviewers for the most part value surround sound, the readership must be consulted as well (another poll to follow in *GameSpy* and I'll report back on the results). It has been stated that surround sound is most effective in 3D immersive game situations (*Thief: Deadly Shadows* comes to mind), but also that a surround sound setup in an average living room is often more unwieldy to arrange ergonomically than a stereo setup. Nevertheless, surround systems are becoming more popular because audio/video chains are stocking more of them.

When looking at "the next big thing," think carefully about whether it is being promoted as a developmental specialty, or something that is appealing to the masses. It doesn't mean you should orient your brain entirely toward the public, because the public is often a large, hulking, stupid, giant sheep that follows anything that's repeated. However, this sheep will provide your company with the golden fleece of profit if your idea appeals to it, so make sure to keep it in mind. ✦

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



» GAME SHUI

BRAINSTORMING

IF PEOPLE
HAVE THE
POWER TO GIVE
YOU A RAISE
OR FIRE YOU,
IT'S HARD TO
BE OBJECTIVE
ABOUT THEIR
IDEAS ...

I LOVE MY JOB. AND WHY NOT? I'M MY OWN boss, get paid to work on a wide variety of games that entertain and educate, travel around the world, and work with some of the best people in game development. Perhaps most important for this industry, my success is not inextricably linked to the survival prospects of any given game company. I even get to write a column for the best darn game magazine in the world, and only have to brown-nose about it once every six months (check, that's it for the rest of 2004).

But of all the fun and exciting aspects of work as a freelance game designer, there has always been one aspect that I enjoy the most—brainstorming.

DEFINING THE PROCESS

Brainstorm (brān 'stōrm) n.

1. A sudden clever plan or idea; 2. A sudden, violent disturbance of the mind.

That first definition is the standard meaning in most game design sessions in the sense of coming up with clever plans or ideas quickly, usually with a group of people. But the second definition captures the essence of what I love best about brainstorming—the critical moment when one key idea or twist suddenly jump-starts the process, and new ideas and variations start coming so fast, it's tough to write them down.

STARTING

Brainstorming is a process with some basic guidelines or (in the parlance of *The 400 Project*) rules. One standard rule is “No judgments,” or as I saw it on one web site, “Every person and every idea has equal worth.” The idea is that during

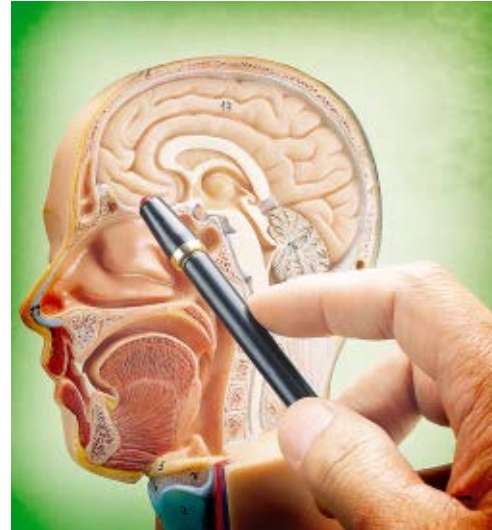
a brainstorming session no one is allowed to say anything judgmental, which helps avoid crushing the ego of less vocal participants. In practice, I find this is a lot like training wheels on a bicycle—a good way to help beginners get started, but likely to slow you down once you've mastered the basics. Having brainstormed with some very brilliant designers and writers, I can also authoritatively state that some people are just better at it than others, and their ideas should carry more weight. A rule I prefer is “Critique ideas, not people.” I've found that some judgment is a good way to make faster progress, but only when the focus is on the ideas themselves, avoiding personal slurs or blanket comments, such as, “That's dumb.” This is just common sense and basic maturity, but in the games industry those qualities are sometimes lacking.

One good rule for beginning brainstorming is “No bosses allowed.” If people have the power to give you a raise or fire you, it's hard to be objective about their ideas or to propose something that may differ from their preferences. Having an experienced leader or facilitator for the session is a good thing, but I recommend restricting most brainstorming sessions to the core team, not including upper management.

ADVANCING

Once a group has more experience brainstorming with each other, some more advanced brainstorming rules are helpful. One of my favorites is “Challenge assumptions—your own and others'.” It's amazing how often our ideas are shackled by hidden assumptions that come from nothing more than historical precedence, or the “it's always been done that way” fallacy. Some of the most exciting innovations in games came from people who questioned the status quo and tried something new.

Another helpful technique is to “Alternate your approach,” moving back



and forth between brainstorming about the theme or story of a game and the actual game mechanics. It's possible to get stuck in the story and forget to think about what the player will be doing, or to get so lost in minor details, such as how to deal with people accidentally setting fire to a crucial building, that they forget a story-related change, like making that building from stone to neatly sidestep the issue.

DELVING

I learned a useful brainstorming technique from Orson Scott Card, a science fiction writer. He suggests that we think of our creative memories as a kind of mental bookshelf of all our experiences and ideas. When we try to come up with an idea, we often go to the first book on the shelf, which has all the stuff we've seen repeatedly, the cliches and standard approaches. If you go farther down the bookshelf to find ideas that are surprising or offbeat, and yet not so far down that they lose all relevance, you can make your game fresh and exciting and avoid the obvious.

In the end, remember that brainstorming is about fun, and if you're not enjoying the process, you're missing out on one of the best parts of design. ❖

PHOTO COURTESY OF PHOTOS.COM

NOAH FALSTEIN is a 24-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. You can e-mail Noah at nfalstein@gdmag.com.



**SWINGIN' APE
STUDIOS™**
ORANGE COUNTY, CA

WE'RE MAKING



WANNA HELP?

POSITIONS NOW AVAILABLE:

SENIOR ARTIST	LIGHTING ARTIST
TEXTURE ARTIST	PROGRAMMERS

VISIT WWW.SWINGINAPE.COM FOR MORE DETAILS,
OR E-MAIL WORK@SWINGINAPE.COM



datascope RECRUITMENT

DATASCOPE, THE UK'S LEADING GAMES RECRUITER, IS NOW IN THE U.S.A. Following recent expansion Datascope has opened an office in Chicago. Dealing with development jobs throughout the U.S.A. and Canada, we are proud to have some of the largest, well respected developers, as well as new exciting companies at the forefront of games technology, on our client list.

- 13 years' industry experience
- Expert career advice
- Exclusive clients
- Transparency & confidentiality
- Daily job bulletins




www.gatewaytogames.com
datascope - recruitment specialists for interactive entertainment

datascope
Chicago +1 312 587 3020 info@gatewaytogames.com
London +44 (0) 20 7580 6018 info@datascope.co.uk

KONAMI
COMPUTER
ENTERTAINMENT
HAWAII

Konami Hawaii is looking for highly motivated and talented individuals to bring the next generation of quality AAA games to the market.

The following positions are available:

- Senior Programmers
- 3D Graphics Programmers
- Mobile Programmers
- Audio Programmers
- Artists
- Animators
- Game Designers

Please contact us at HawaiiHR@konami.com or visit our website at: www.konamihwl.com/jobs.php



Work in Paradise!



**TIMEGATE
STUDIOS**

A New World
Awaits You...

Positions Available:

- Sr. Software Engineers
- 3D Artists/Animators
- Graphics Engineers
- Sr. Texture Artists
- Game Designers
- Level Designers

Apply Today!

jobs@timegate.com
www.timegate.com
14140 Southwest Frwy
Suite 400
Sugar Land, TX 77478

WELCOME TO THE JUNGLE



The studio that created **FARCRY** is searching for talented and passionate professionals to join it's multinational team. We are located in Bavaria, Germany

The following positions are available:

Producers (Senior to Assistant).

Programmers (Lead to Senior).

Designers (Lead, Senior, Technical and Level).

Artists (Lead, Senior, Conceptual, Technical and Environmental).

Animators (Lead, Senior)

Apply to:

codejobs@crytek.de (tech, programming)

artjobs@crytek.de (art, animation)

designjobs@crytek.de (design)

jobs@crytek.de (production)

By mail:

Crytek GmbH

Attn: Human Resources

Rosenauer Str. 16 96450

Coburg Germany

www.crytek.de

www.farcry-thegame.com

© Crytek. All Rights Reserved. Published by UbiSoft Entertainment.

Far Cry, UbiSoft, ubi.com, and the UbiSoft logo are trademarks of UbiSoft Entertainment in the U.S. and/or other countries.



ACADEMY *of* ART UNIVERSITY

FOUNDED IN SAN FRANCISCO 1929



**Register Now for Fall, Spring
& Summer Semesters**

Courses offered in:

2D & 3D Animation

3D Modeling

Computer Graphics

Filmmaking

Game Design

New Media

Visual Effects

Web Design

Classes taught by working professionals from companies such as Pixar and Industrial Light + Magic

AA | BFA | MFA Degrees
Personal Enrichment | Online Courses
State-of-the-Art Facilities

**Over 80% Job Placement
upon Graduation**

1.800.544.ARTS
www.academyart.edu



Art by recent graduate Jung-seung Hong, modeler at Industrial Light + Magic

**A GREAT COLLEGE IS NOW A GREAT UNIVERSITY.
FORMERLY ACADEMY OF ART COLLEGE**

79 New Montgomery St., San Francisco, CA 94105
Nationally Accredited by ACICS, NASAD & FIDER (BFA-IAD)



Your Future is Here...

New Age

New Dreams

New Games

edia

The Center for Digital Imaging Arts
at Boston University

282 Moody Street
Waltham, MA 02453
800.808.2342

www.digitalimagingarts.com

**BOSTON
UNIVERSITY**

**Certificate Programs
in Game Art & Design,
Digital Filmmaking, Digital Photography,
3-D Animation, Visual Effects
and Digital Media & Web Design**

Intensive nine-month programs providing
the tools needed to launch your imaging arts career.
Apply now for fall and spring terms.



**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, ACCSCT

© 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.




image by: Eric Hurley

BRING YOUR IDEAS TO LIFE @

COLLINS COLLEGE

COLLINS MAIN TEMPE, ARIZONA
COLLINS WEST PHOENIX, ARIZONA

Career Services assistance -
Approved for VA benefits -
Financial Aid available to those who qualify -




image by: Jonathan Schulz

you have a vision, you know what you want, a bachelor's degree in **game design or animation** lets you show the rest of the world what

ad design by: Danielle Knotts

Not all programs available at all locations -
Collins West is a satellite campus of Collins College

YOU're made of. you BELONG here @ COLLINS COLLEGE

call **ANYTIME** for more information:
1.888.356.7777





image by: Ryan Heuett



Walt Disney • Don Bluth Animation • Siemens • Paramount Pictures • Sony Entertainment • Lilo & Stitch • Brother Bear • Silverwing • Barde

Want to find out about video game schools?

Do a credit check.



Joseph Gilland
Head of Department
2D Animation & Maya



Robert Grieve
Head of Department
Sound Design for Visual Media



Larry Bafia
Head of Department
3D Animation & Visual Effects



Stephen Webster
Head of Department
Interactive Media

Game Stream Programs at VFS

The game industry now leads the film industry in market share of the worldwide entertainment dollar. Growth is expected to be exponential. In Vancouver alone, thousands of jobs will be created over the next few years.

VFS has four programs designated and viewed by the industry as Game Stream Programs (GSPs): **2D Animation, Sound Design for Visual Media, 3D Animation & Visual Effects and Interactive Media for Communication & Game Design.**

To determine the appropriate game stream for you, call us to speak with an admissions advisor or visit us online at www.vfs.com.

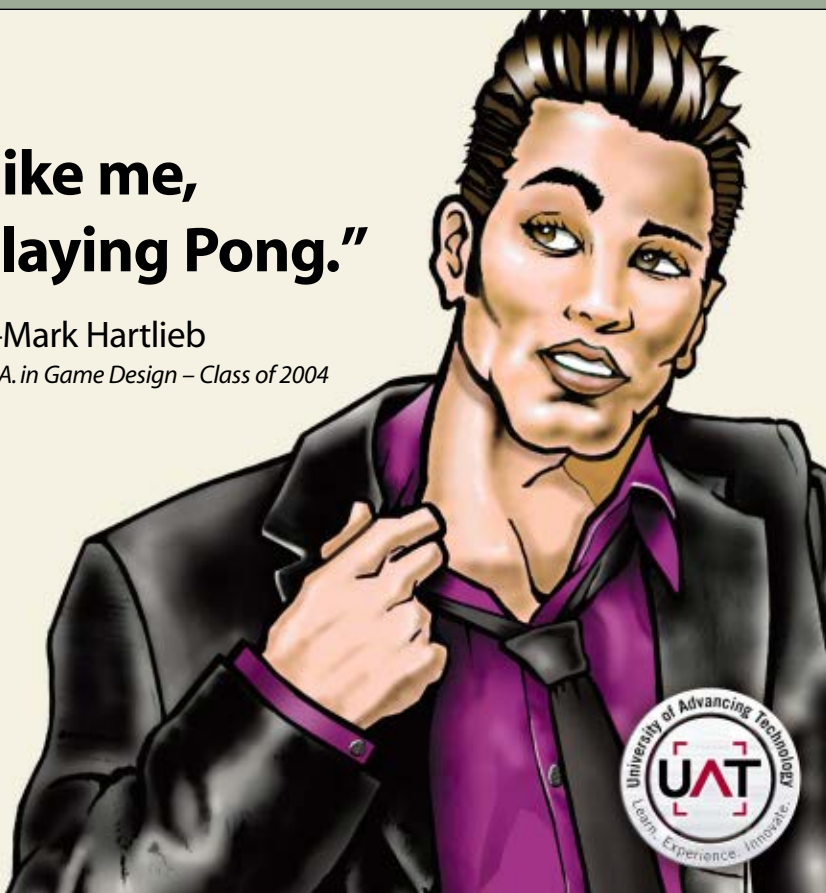
Come visit us at **Booth #445** during the Game Developers Conference, March 24 to 26. Pick up your own copy of our DVD of great student work.



For more information, call Vancouver Film School
1-800-661-4101
or visit us online at www.vfs.com/credits

**“Without guys like me,
you’d still be playing Pong.”**

—Mark Hartlieb
B.A. in Game Design – 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

Rewarding Innovation in Independent Games

The 7th Annual Independent Games Festival

2005 Call for
Entries

IGF Competition

Submissions due: September 1, 2004

Open to all independent game development teams, the IGF Competition offers over \$40,000 in cash prizes.

IGF Student Showcase

Submissions due: November 12, 2004

A non-competitive exhibition of games developed by full-time students.

March 9th-11th, 2005 • San Francisco, California

Entry forms and eligibility rules available at:

www.igf.com



INDEPENDENT
GAMES FESTIVAL

The right way.



NEW PENCIL

www.newpencil.com

Cyber Curfew Counter-Strikes Turf Wars

CONTINUED FROM PG 4

councilman and former police officer supervisor Dennis P. Zine.

"We're not tampering with the content," Zine says about the ordinance, adding that the industry and businesses affected were consulted before the law was established. "This is a solution to an expanding, growing problem. With the industry's assistance, we were able to find a solution that has teeth, but without it being oppressive to the businesses."

Zine says the fights result from gang turf wars, not exposure to violence in games like COUNTERSTRIKE, as some residents suggest. His is a rare gesture from a legislator, and not one shared by California State Assemblymember and child psychologist Leland Yee.

Earlier this year, Yee proposed two bills to prohibit minors from purchasing mature- and adult only-rated videogames on the grounds that the violence in them should be classified as a harmful

substance—like alcohol, tobacco, and firearms. "This is about protecting children," says Yee.

"No kid should be out late at night in these videogame places," he says. "They should be at home doing their homework or with their families. We have curfew laws now, but they are rarely implemented and cumbersome to enact," he adds.

Yee insists that game content induces adolescents to lash out, whereas the Southern California councilman maintains that the problem stems from kids—often gang members—congregating at the gaming venues, which are open late and subject to gang territorial claims. Zine's recognition that the violence tends to be regional and associated with gang activity dismisses the notion that the law panders to parents' concerns about violent videogames. "I was very pleased to see this ordinance pass," Zine says, adding he would be even more impressed if other cities modeled their laws after L.A.'s.

—Jill Duffy

Anime MMOG 25 Million Strong

CONTINUED FROM PG 5

of pre-Playstation FINAL FANTASY that can only be described as cute, animated onto a 3D world. The player characters resemble closely the game's concept art drawn by Myung-Jin Lee, the creator of the manwha (Korean comic) of the same name on which the game is based. As such, they have enormous eyes and sometimes gravity-defying hair. The reason was to make the

game as accessible as possible to all ages and demographics from "8 to 10 years old all the way up to 60." According to Lee, great care went into making the game "least offensive and most appealing" to everyone so that the game is not too violent or too vulgar. This general approach has netted the game a high ratio of female gamers (28 percent as opposed to the MMORPG average of approximately 8 percent).

—Quang Hong, Gamasutra.com

Interplay's Bottom Fallout

You work late into the night and nosh on cold pizza, provided by your employer, if you're lucky. And every other week or so when the direct deposit statement falls into your lap, you balance your checkbook. Another day, another dollar, another cycle.

Not so for the 79 employees of Interplay Entertainment Corp. who were barred from working for several days this summer when the California Department of Industrial Relations, Division of Labor Standards Enforcement put its foot down on their delinquent employer.

In late May, two employees filed complaints that their paychecks bounced. Soon after, 20 employees had filed complaints with the state agency alleging that Interplay, headed by CEO Herve Caen, missed payroll for three consecutive periods. Interplay also failed to carry workers' compensation insurance for its staff. Three months of overdue rent resulted in the property owner filing a lawsuit, too.

"On April 16, 2004, our lessor filed an unlawful detainer action against us alleging unpaid rent of approximately \$432,000," Interplay officials stated in a report to the Securities and Exchange Commission. "Since that filing, we also failed to pay the May and June 2004 rent and vacated the office space during the month of June 2004."

Adding to the financial sinkhole, the IRS notified the company that it owes about \$70,000 in payroll tax penalties. "We estimate that we owe an additional \$30,000, which we have accrued in penalties for nonpayment of approximately \$100,000, \$102,000, and \$99,000 in Federal and State payroll taxes," the SEC filing stated. As of press time, company officials would not comment on the financial situation. To heave itself out of this near-bankrupt muck, Interplay licensed FALLOUT 3—a planned sequel to the profitable RPG—to Bethesda Softworks.

—Jill Duffy

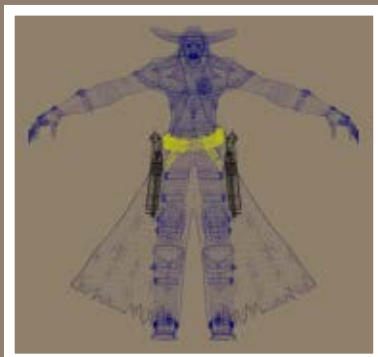
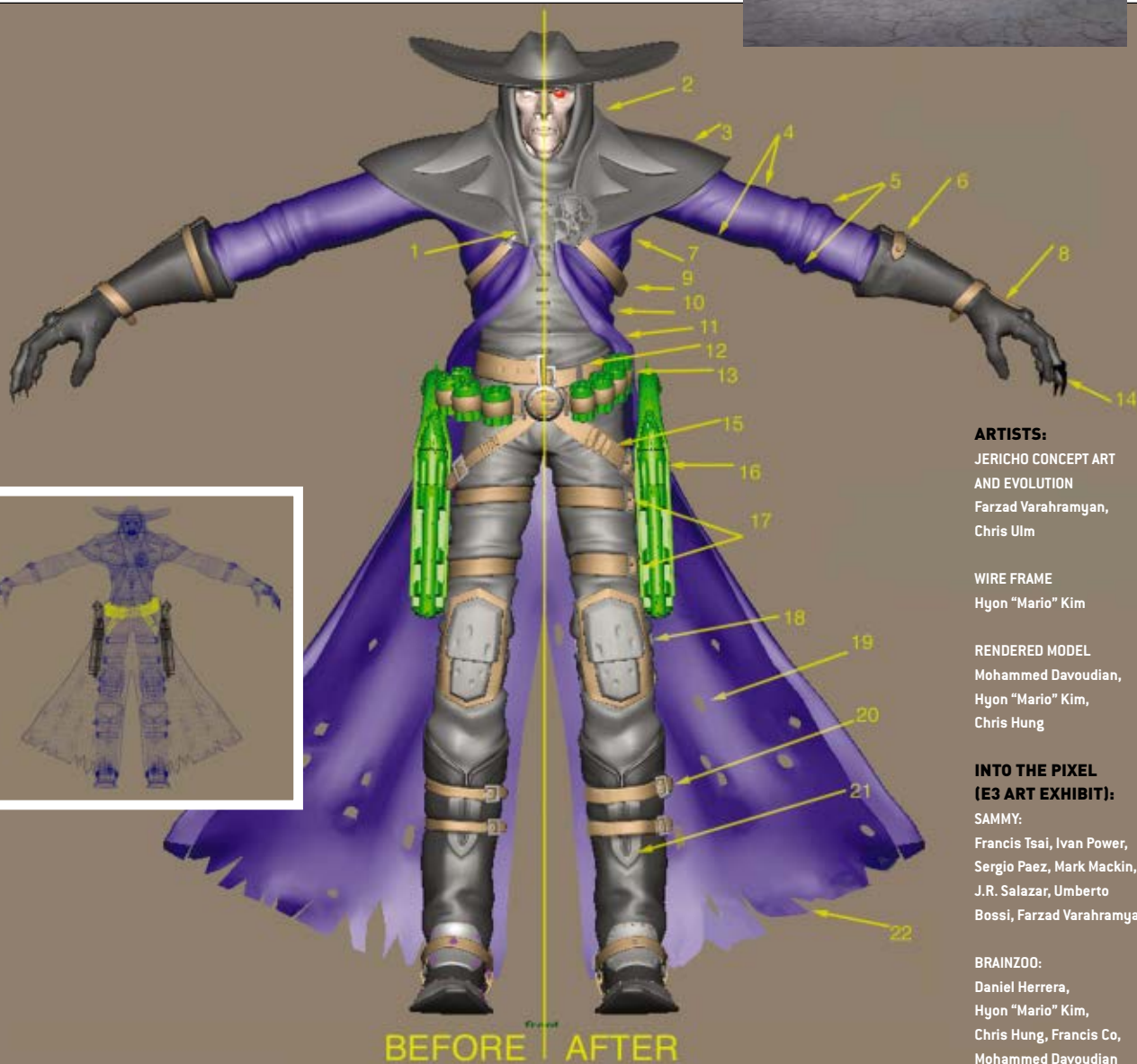
ADVERTISER INDEX

COMPANY NAME	PAGE	COMPANY NAME	PAGE	COMPANY NAME	PAGE
Academy of Art University	50	Havok	17	Stottler Henke Assoc.	23
Alias	3	IT Global Secure	43	Swingin' Ape Studios	48
Araxis	24	Konami	48	Techexcel	7
Center for Digital Imaging	51	New Pencil	54	Time Gate Studios	48
Collins College	52	Numerical Design	C3	University of Advancing Technology	53
Course Technology/Premier Press	11	Perforce	25	Vancouver Film School	52
Crytek GMBH	49	RAD Game Tools	29, C4	Vicon Motion Systems	21
Datascope Recruitment	48	RTzen	C2	Xoreax	24
Full Sail Real World Education	52	Softimage	15		

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



ARTISTS:

JERICOH CONCEPT ART AND EVOLUTION
Farzad Varahramyan,
Chris Ulm

WIRE FRAME

Hyon "Mario" Kim

RENDERED MODEL

Mohammed Davoudian,
Hyon "Mario" Kim,
Chris Hung

INTO THE PIXEL (E3 ART EXHIBIT):

SAMMY:

Francis Tsai, Ivan Power,
Sergio Paez, Mark Mackin,
J.R. Salazar, Umberto
Bossi, Farzad Varahramyan

BRAINZOO:

Daniel Herrera,
Hyon "Mario" Kim,
Chris Hung, Francis Co,
Mohammed Davoudian

BEFORE | AFTER

THE EVOLUTION OF JERICOH CROSS FROM SAMMY STUDIOS' DARKWATCH

1. Corner of coat cleaned up; the connecting silver button and buckle should be bigger than shown.
2. Adjusted collar a bit higher.
3. Adjusted silhouette of shoulder flap.
4. Adjusted arm silhouette to be more muscular, less sausage-like.
5. Swelled out elbow creases a bit more.
6. Widened strap and adjusted details, glove also now flares out slightly less.
7. Adjusted silhouette of armpit and chest connections.
8. Set knuckle strap on top of glove.
9. Made side straps thicker.
10. Adjusted silhouette of coat side to be a bit more in and crisp.
11. Adjusted coat flap as shown and pulled in tighter to the body.
12. Lowered belt and added pant loops and a bit of pant edge.
13. Scaled down ammo cartridges and placed as close as possible to body.
14. Adjusted the scale and length of hand slightly (smaller and shorter).
15. Made belt thicker and bullets bigger.
16. Scaled gun down slightly in width only and placed as close to body as possible. Has the mounting bracket for the pistol been built and placed where the pistol attaches to the thigh?
17. Made belts thicker and added buckles.
18. Enlarged knee pads slightly. Were the changes to boots and spurs made on this model?
19. Added some new holes and deleted some holes.
20. Placed buckles on outside of boots.
21. Slightly lowered boot slit and pointy border.
22. Fine tuned fray, where possible, to match tear pattern in coat flaps.

Why NOT to buy a commercial graphics engine

REASON #2

**It's made
by your
biggest
competitor.**

(Enough said)

You've got a better choice.

To get technical specs on the leading independent
3D graphics engine, visit us at gamebryo.com



Just Drop Us In!



The Miles Sound System provides the most sophisticated sound for the PC, Mac, Xbox, and Linux. It supports 2D, 3D, streaming, MIDI with DLS, the world's fastest MP3 decoder (with patent rights), internet voice chat, software reverb, and much more!



Pixomatic is our powerful software renderer for PCs. Essentially, it is a high-end DX7 video card in software form! MMX, 3DNow, and SSE optimized assembly is generated at run-time. Don't let low-end hardware or bad video drivers hurt your game's sales!



Granny is a powerful toolkit for building all kinds of interactive 3D applications. It features the most efficient and flexible content exporters, data manipulation, normal mapping and run-time dynamic 3D animation system you'll find anywhere.



Bink is the finest video codec for games! It provides better than DVD quality at half the data rate along with a perceptually lossless 10:1 audio codec in a simple and clean API. It is available for PC, Mac, Xbox, GameCube, Linux, and now Sony PlayStation®2!



GAME TOOLS

THE BEST IN GAME DEVELOPMENT TECHNOLOGY

www.radgametools.com **425.893.4300**

*PlayStation® is a registered trademark of Sony Computer Entertainment Inc.