CMP
United Business Media

>> **PRODUCT REVIEWS** NVIDIA'S NV PERF HUD 4 * SLICKEDIT'S SLICKEDIT 11

AUGUST 2006

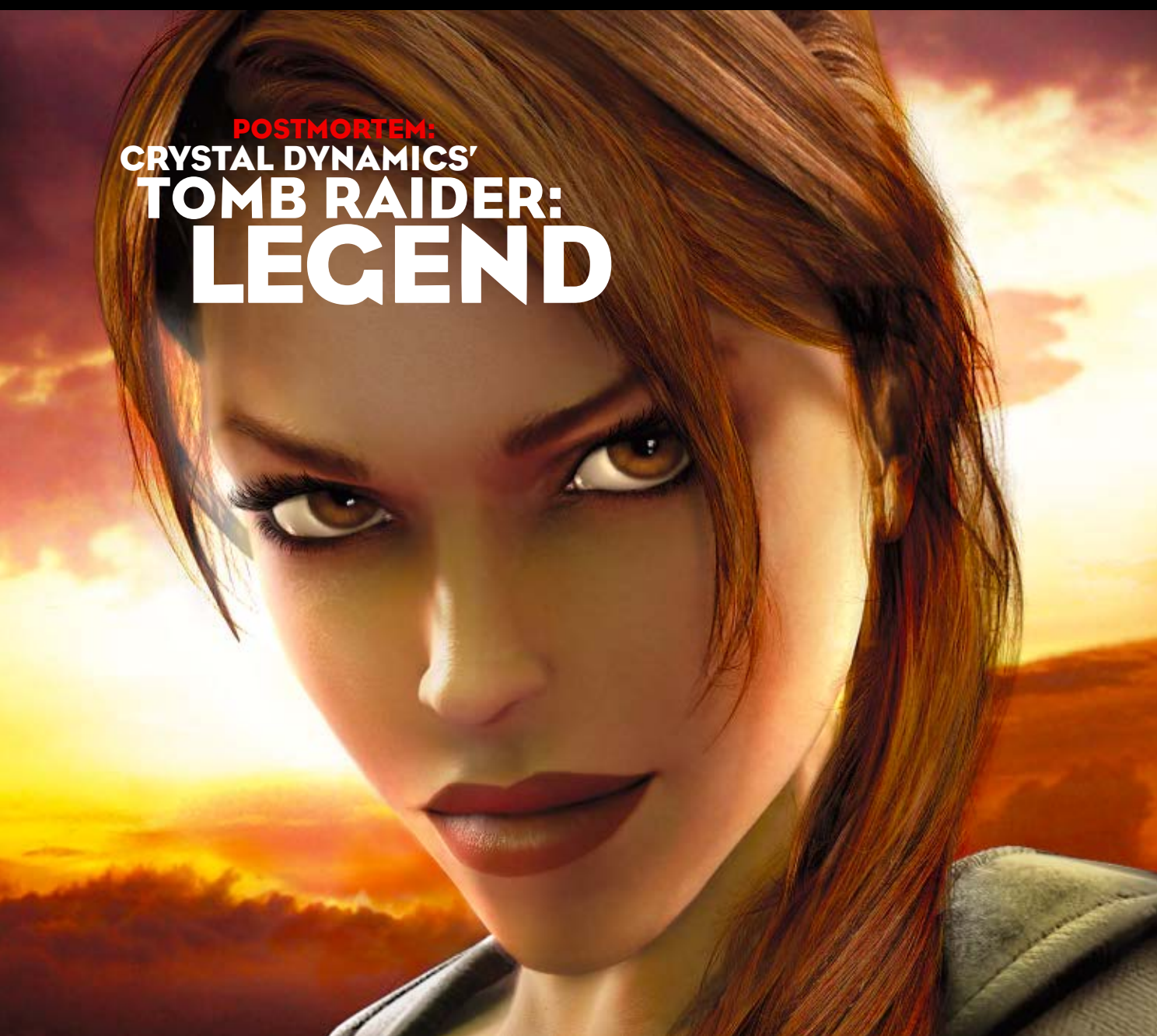
gamedeveloper

THE LEADING GAME INDUSTRY MAGAZINE

>> **DELIVERING THE GOODS**
HOOK YOUR AUDIENCE, THEN REEL THEM IN

>> **I GOTTA BELIEVE**
MASAYA MATSUURA OUTSIDE THE MOLD

>> **SLERPING MADE SIMPLE**
FREE AT LAST FROM TRIGONOMETRIC FUNCTIONS



POSTMORTEM: CRYSTAL DYNAMICS' TOMB RAIDER: LEGEND

VICIOUS ENGINE

making games. easier. faster. better.

PSP™(Playstation®Portable) · PC · Playstation®2 · XBOX · GCN

A powerful and flexible development environment offering everything you need to start building games today.

Engine support for

Nintendo Wii

and

XBOX 360™

Coming Fall '06!

"PSP" is a trademark of Sony Computer Entertainment Inc. "Playstation" is a registered trademark of Sony Computer Entertainment Inc.

Contact us about our fully functional 90 day evaluation and recently reduced licensing fees.

www.viciousengine.com

*in-game model and environment.

gamedeveloper

AUGUST 2006
VOLUME 13, NUMBER 7

CONTENTS

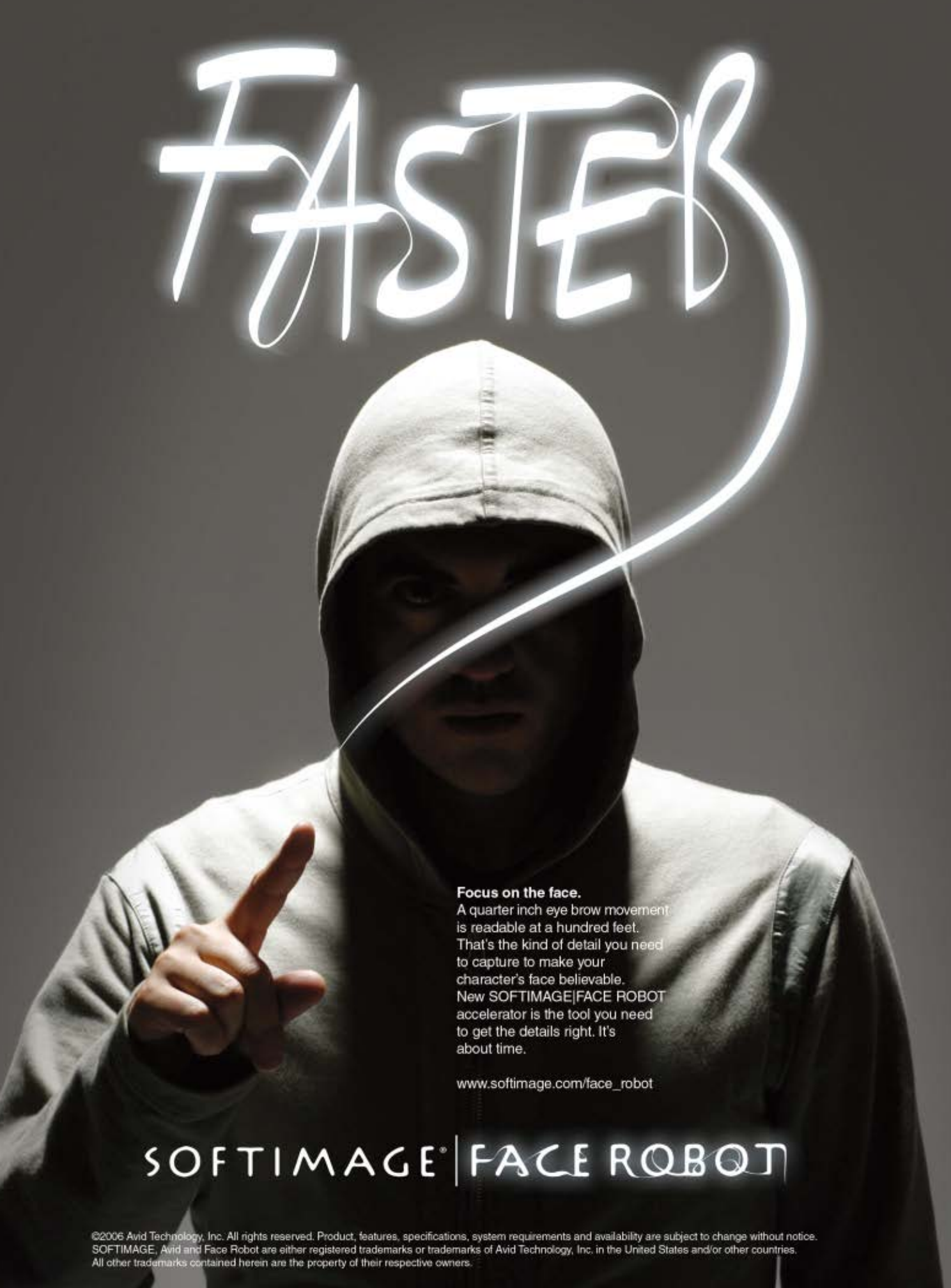
FEATURES

POSTMORTEM

DEPARTMENTS

COLUMNS

COVER ART: JOEL BOUCQUEMONT



Keep it simple.
Character pipelines are complex, and assets can easily fall apart. With XSI's proprietary GATOR technology, you can modify your assets with confidence and be sure they'll never break. Spend less time fixing characters and more time bringing them to life.

www.softimage.com/xsi

SOFTIMAGE® | XSI

CONNECTING PEOPLE

YOU KNOW WHAT I'M MOST EXCITED ABOUT WITH regard to the next generation of game consoles? It's not flashy graphics, or super-advanced physics (with or without a PPU), or even hard drives for streaming data, or next-gen disc formats for HD playback. It's the ability to interact with friends and acquaintances while playing—and for me, on the Xbox 360, this has revolutionized my playing experience.

WHY PLAY IF YOU CAN'T SHOW?

I'm aware that the above statement has been made plenty of times about a multitude of things, from HD to motion controllers. But what's surprising is that this sentiment is genuinely felt by me with regard to consoles and online—I care more about games again because of my experiences on the 360. And most interestingly, there are some obvious deathmatch-happy elements to why permanent online connectivity is so cool for consoles, but I think it's actually non-hardcore players who are served the best by Xbox Live.

The reason I say this is because I (XBL tag simoniker) and a lot of the people on my friends list—co-workers and people in the game journalism or development industries—don't necessarily have the time or inclination to slap down on a raucous multiplayer game of CALL OF DUTY 2 every night for six hours. We have families and lives and needy web sites to update.

What we can do is check out what the others have been playing recently on Xbox 360 and Xbox Live Arcade via our Gamer Profiles. We can compare scores, chat to one another about them, set little goals to beat one another's GEOMETRY WARS score, and build up our Achievement scores to semi-respectable levels along the way ... hopefully.

This whole concept of being able to cross-reference at any time makes single-player games feel a whole lot less lonely. Even playing through the relatively neglected AMPED 3 recently, I was reading regular updates on how I ranked worldwide in individual challenges as I completed parts of the game.

APE SHALL NEVER KILL APE

On Xbox 360, at least, it's both humanizing and reassuring. People have been here before. You can see how well they've done, gauge yourself to improve your performance, and empathize with fellow players. Perhaps this isn't revolutionary or even very complex, but I now feel motivated to boot up my machine just to see what other people are up to, and I feel like it's worth completing Xbox 360 games just to get Achievements, especially since my Gamerscore is embedded as a widget on my weblog.

More than that, it makes me feel like playing games isn't a waste of time. Somehow, interacting with other people and having my best fruits of play available over the entire net instead of just locked onto a memory card, helps to legitimize the whole gaming experience for me, and doubtless a lot of other people. The difference is between sharing my time with a controller and sharing it with the entire world.

BEYOND THE SOLITARY

Of course, that's just talking about passively interacting with friends and other Xbox Live denizens. With all the other benefits of being able to play co-op and versus gameplay on full retail and Live Arcade games alike, as well as being able to download updates, maps, and trailers easily, there are reasons to return to your console regularly even if you haven't bought any new games.

Perhaps the ultimate example of this connected era is an Xbox 360 Achievement planned by Gastronaut Studios for its Live Arcade game called SMALL ARMS. According to a recent MTV News report, the achievement, called Six Degrees of SMALL ARMS, "will start with the four people at his small game development company... and spread to anyone who plays them online. It will continue to be contagious for anyone who has it." The implementation logistics for the Achievement were still being worked out as of press time, but this concept is exactly why the next generation of consoles is the Connected Era, not the HD Era. ⁘

—Simon Carless
editor-in-chief

gamedeveloper
www.gdmag.com
CMP Media, 600 Harrison St., 6th Fl., San Francisco, CA 94107 t: 415.947.6000 f: 415.947.6090

EDITORIAL
EDITOR-IN-CHIEF
Simon Carless *scarless@gdmag.com*
MANAGING EDITOR
Jill Duffy *jduffy@gdmag.com*
FEATURES EDITOR
Brandon Sheffield *bsheffield@gdmag.com*
ART DIRECTOR
Cliff Scorso *cscorso@gdmag.com*
CONTRIBUTING EDITORS
Jesse Harlin *jharlin@gdmag.com*
Noah Falstein *nfalstein@gdmag.com*
Steve Theodore *stheodore@gdmag.com*
Mick West *mwest@gdmag.com*
ADVISORY BOARD
Hal Barwood Designer-at-Large
Ellen Guon Beeman Microsoft
Andy Gavin Naughty Dog
Joby Otero Luxoflux
Dave Pottinger Ensemble Studios
George Sanger Big Fat Inc.
Harvey Smith Midway
Paul Steed Microsoft

ADVERTISING SALES
DIRECTOR OF SALES
Afton Thatcher e: *athatcher@cmp.com* t: 415.947.6217
GLOBAL SALES MANAGER, RECRUITMENT & EDUCATION
Aaron Murawski e: *amurawski@cmp.com* t: 415.947.6227
SENIOR ACCOUNT MANAGER, EAST COAST, EUROPE & EASTERN CANADA
Ayrien Machiran e: *amachiran@cmp.com* t: 415.947.6224
ACCOUNT MANAGER, NO. CALIF., NORTHWEST, ASIA & WESTERN CANADA
Nick Geist e: *ngeist@cmp.com* t: 415.947.6223
ACCOUNT MANAGER, SO. CALIF., SOUTH WEST, CONTRACTORS, & MARKETPLACE
Jasmin Davé e: *jdave@cmp.com* t: 415.947.6226

ADVERTISING PRODUCTION
ADVERTISING PRODUCTION COORDINATOR Kevin Chanel
REPRINTS Julie Rapp e: *jarapp@cmp.com* t: 510.834.4752

CMP GAME GROUP
VP, GROUP PUBLISHER APPLIED TECHNOLOGIES Philip Chapnick
VP, STRATEGIC MARKETING Michele Maguire
GROUP DIRECTOR Valerie Pippin
DIRECTOR OF MARKETING Tara C. Gibb
CONFERENCE DIRECTOR, GDC Jamil Moledina
SENIOR CONFERENCE MANAGER, GDC Meggan Scavio
EXECUTIVE WEB PRODUCER Peter Leahy
EDITOR-IN-CHIEF, GAMASUTRA.COM Simon Carless
FEATURES EDITOR, GAMASUTRA.COM Quang Hong
ASSISTANT EDITOR, GAMASUTRA.COM Frank Cifaldi

CIRCULATION
CIRCULATION ASSISTANT MANAGER Jessica Ward e: *jward@cmp.com*
CIRCULATION COORDINATOR Miguel Mendiolaza e: *mmendiolaza@cmp.com*
CIRCULATION ASSISTANT Michael Campbell e: *mcampbell@cmp.com*
CIRCULATION ASSISTANT Andrea Abidor e: *aabidor@cmp.com*

SUBSCRIPTION SERVICES
FOR INFORMATION, ORDER QUESTIONS, AND ADDRESS CHANGES
t: 800.250.2429 f: 847.763.9606 e: *gamedeveloper@halldata.com*
INTERNATIONAL LICENSING INFORMATION
Mario Salinas
t: 650.513.4234 f: 650.513.4482 e: *msalinas@cmp.com*

CMP MEDIA MANAGEMENT
PRESIDENT & CEO Steve Weitzner
EXECUTIVE VP & CFO John Day
EXECUTIVE VP, GLOBAL SALES & BRAND DEVELOPMENT Jeff Patterson
SENIOR VP, BUSINESS OPERATIONS Adam Marder
SENIOR VP, AUDIENCE MARKETING & DEVELOPMENT Bill Amstutz
SENIOR VP, INTERNET BUSINESS Mike Azzara
SENIOR VP, CMP INTEGRATED MARKETING SOLUTIONS Joseph Braue
SENIOR VP & GENERAL COUNSEL Sandra Grayson
SENIOR VP, CORPORATE SALES Anne Marie Miller
SENIOR VP, MANUFACTURING Marie Meyers
SENIOR VP, COMMUNICATIONS Alexandra Raine
SENIOR VP, CORPORATE MARKETING Kate Spellman
VP, AUDIENCE DEVELOPMENT Michael Zane
PRESIDENT, CHANNEL GROUP Robert Faletra
PRESIDENT, CMP ENTERTAINMENT MEDIA Tony Keefe
PRESIDENT, CMP HEALTHCARE MEDIA Vicki Masseria
SENIOR VP, GROUP DIRECTOR, ELECTRONICS & SOFTWARE GROUPS Paul Miller
SENIOR VP, ENTERPRISE GROUP, GROUP DIRECTOR Fritz Nelson
SENIOR VP, GROUP DIRECTOR, COMMUNICATIONS GROUP Stephen Saunders

Game Developer is BPA approved
CMP United Business Media

TRIPLE EVENT WRAP

CONFERENCES FOCUS TIGHTLY ON SUB-SECTORS OF INDUSTRY

WITH INDUSTRY EXPANSION COMES CONFERENCE FRENZY. DEVELOPER-oriented events have grown increasingly numerous, with smaller and more specific events cropping up across the U.S., particularly during the week of June 26, when three were held on the west coast: Hollywood and Games in Los Angeles, Casuality in Seattle, and MI6 in San Francisco.

Hollywood and Games (operated by CMP, which also owns *Game Developer*), held June 27, sought to identify the increasing synergies and intellectual property crossover potential between film and video games.

Paul W. S. Anderson, director of the *Mortal Kombat* and *Resident Evil* movies, delivered a keynote in which he outlined some ways to make a successful movie adaptation. Other speakers included Shiny founder Dave Perry, EALA's Neil Young, Kevin Feige of Marvel Studios, and New Line Cinema's Gordon Paddison.

On dealing with consumer expectation, Anderson said, "Clearly, it's a minefield. And it's a minefield as a filmmaker that you'd better learn to navigate," pointing out the fact that fans of a game can sour toward the film version before it's even released. "If you stray too far from the source material, you're doomed. But if you stay entirely true, you're equally doomed." The best way to make a game out of a movie or vice versa, says Anderson, is to create an IP with both media in mind, allowing one to enhance the other.

Casuality Seattle, a three-day event in late June, offered no less than four keynote addresses on the conference topic, casual games. The event's aim was to not only inform, but also maintain the fun and light mood that marks most casual games.

Real Networks' CEO Rob Glaser, Bing Gordon from EA's Pogo.com, Liberty Media's senior vice president Michael Zeisser, and PopCap co-founder John Vechey all had choice words at the show.

A particularly notable session concerned getting the most out of your contract, as casual games are still a growing field with many kinks to be worked out. "The contract is pretty much worth the paper it's printed on," said a skeptical Josh Welber, Large Animal Games' co-founder.

Mumbo-Jumbo president Paul Jensen added, "It's very rare for portal promotion to be guaranteed in a publisher contract," and further suggested that developers partner with different publishers for different methods of distribution, from mobile to PC. Welber mentioned that contracts could have fringe benefits though, stating that "[the contract] at least forces you to sit down and figure out the spirit of a project with a publisher."

MI6, or Marketing Interactive '06, was held over two days in San Francisco. The conference was put on by the non-profit interactive marketing awareness group Promax/BDA, and focused on marketing games. MI6 largely trod familiar territory, with keynote speaker Chris Di Cesare (director of marketing for Xbox) discussing the launches of HALO 2 and the Xbox 360.

"What it all comes down to is control," said Di Cesare. "There's unprecedented power to decide what, when, and how consumers view things," he said, adding that internet and Tivo-savvy consumers of today know how to avoid ads that they don't like.

MI6 was also host to the peer-voted Senet Awards, which acknowledge excellence in game marketing. Will Wright received the Hall of Fame award; Activision and its contractors won awards for Campaign of Distinction, Best PR Piece, and Best Promotional Item; Sony Computer Entertainment took home the Landmark Award for Excellence in Consumer Branding; Ubisoft was granted the award for Creative Excellence; Nvidia accepted the Technical Driver Award; and the Xbox 360 marketers were awarded Marketing Team of the Year.

—Brandon Sheffield

product news

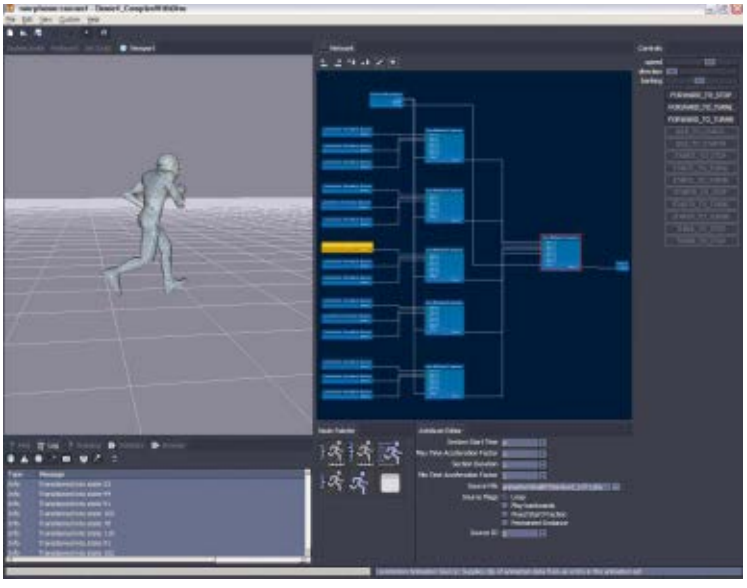
NATURALMOTION REVEALS NEW ENGINE

A NEW ANIMATION ENGINE FOR GAME DEVELOPMENT WAS unveiled late last month from Oxford-based NaturalMotion. Dubbed morpheme, the product is the third major offering from the company, which also makes endorphin and euphoria, and will support development of Xbox 360, PlayStation 3, and PC games.

Morpheme consists of two components: a run-time engine, which ships with full source code, and a 3D authoring application, called morpheme:connect. Morpheme:connect lets developers play with animation in real time using a system of sliders, viewports, and other user interface elements, and is compatible with NaturalMotion's endorphin.

According to NaturalMotion, morpheme is expected to be available in October. Additional information is available at www.naturalmotion.com.

—Jill Duffy



JAPANESE GAME MARKET UP 33%

Nintendo leads pack with DS, DS Lite, smarty-pants games

THE OVERALL JAPANESE video game market surged 33 percent in the first half of 2006 compared to the previous year, reaching 277 billion yen ($2.40 billion) for combined hardware and software sales, according to a survey released by Japanese firm Enterbrain.

Enterbrain's survey, which collected information from 35,000 Japanese game stores, revealed particularly beefy results for Nintendo, which sold 2.6 million units of the DS Lite and 1.3 million units of the DS in the first six months of this year. Software sales for Nintendo also proved strong.

The news, twinned with Wii anticipation, sent Nintendo's shares up 3.9 percent to hit a four-year closing high of 20,390 yen ($179) in early July, according to Reuters, bucking the Nikkei's downward trend for the day and reinforcing the company's current domination of the Japanese handheld market.

In comparison, Sony sold slightly less than 1 million units of its PSP in the first six months of 2006, effectively a quarter of the sales of DS-related hardware, exposing the difficulty the company faces in selling its more complex 3D titles for the handheld console to a market endeared with cheery 2D simplicity. Sony's LocoRoco, a game more in line with current Japanese buyer preferences shows Sony's effort to win back some of the market share.

Overall Japanese game software sales, dominated by DS-related games in the Brain Training and Touch Generations series, was up 30.4 percent to 175 billion yen ($1.53 billion), with Brain Training for Adults the overall top-selling title for the period, and games such as New Super Mario Bros. quickly rising to rival it.

—Simon Carless

WOMEN DRIVING MOBILE MARKET

WOMEN AND PUZZLE GAMES STEERED THE mobile game market to strong results in the first quarter of 2006. Puzzle and strategy games generated one-third of the total revenue in Q1 for the U.S. mobile game market, with a full 65 percent of all mobile game revenue generated by female wireless subscribers, according to research from Telephia's Mobile Game Report. According to the report, female wireless gamers contributed 72 percent of the revenue generated by puzzle and strategy games.

The most popular games among females were mobile titles categorized as trivia and word games, such as Cosmic Infinity's Who Wants To Be A Millionaire 2005. While this category represented only 11.4 percent of overall mobile revenue for the quarter, 74 percent was attributed to the female audience.

"The casual nature of mobile games provides tremendous appeal to women who are not traditionally hardcore gamers by console or online gaming standards," says Kanishka Agarwal, vice president of new products at Telephia.

The report also found that four of the top five revenue-generating titles were puzzle or strategy-based games, with Tetris, Tetris Deluxe, and Bejeweled securing the top spots.

Tetris and Tetris Deluxe held 5.2 and 3.6 percent of the total mobile game revenue for the quarter, respectively, while Bejeweled held 2.6 percent. The fourth most revenue-generating title in the report was JAMDAT Mahjong with 2.2 percent. All four top titles were published by EA Mobile.

"The wireless industry has presented a hungry target audience willing to pay premium for access to games through their cell phones," Agarwal says. "Mobile game purchases continue to grow with more than 8.6 million games bought in April, increasing 60 percent since the beginning of the year."

—Jason Dobson

CALENDAR

GC Developer Conference (GCDC)
Leipzig Fairground
Leipzig, Germany
August 21–23
Price: 195–395 euros
www.gcdc-germany.com

Toronto Independent Games Conference
Casa Loma campus of George Brown College
Toronto
August 31–September 2
Price: $50–$75
www.torontoigc.com

Nordic Game 2006
MalmöMässan
Malmö, Sweden
September 19 and 20
Price: $195–$830
http://nordicgame.com

OUR RATING SYSTEM :

EXCEPTIONAL | GREAT | FAIR | POOR | UNFORTUNATE

NVIDIA'S NVPERFHUD 4

MATT SHAW

NVIDIA

STATS
Nvidia Corporation
2701 San Tomas Expy.
Santa Clara, CA 95050
(877) 7-NVIDIA
www.nvidia.com

PRICE
Free for download

SYSTEM REQUIREMENTS
Windows XP
DirectX 9.0c
Any NvidiaGPU (GeForce 3 or better), GeForce 7 Series, GeForce 6 Series, G70-based or NV4X-based Quadro FX, or better recommended. Reduced functionality with older than GeForce 6 Series.

PROS
1. Easy to install and set up.
2. Provides insight on GPU performance issues.
3. Useful for artists and programmers.

CONS
1. Only works with Nvidia GPUs.
2. Tied to the driver included with the package.
3. Annoying to have to download OpenGL debugging tools if all you're interested in is NVPerfHUD for DirectX. The download and install is large enough already.

NVPERFHUD 4 STANDS FOR NVIDIA

Performance Heads Up Display, a performance analysis tool that lets developers see how their application uses the GPU, both globally and down to the level of individual draw calls.

NVPerfHUD 4 is free to download from the Nvidia Developer web site (http://developer.nvidia.com). It is a component of NVPerfKit 2, which was released in June 2006. It requires DirectX 9.0c, Windows XP, a modern Nvidia video card, and some minor software support from the developer. Included with the approximately 85MB download is some excellent documentation: a 59-page user manual, a two-page quick start guide, and an instrumented video driver.

You launch NVPerfHUD from the command line or by simply dragging and dropping your game's icon onto the NVPerfHUD icon. The tool then executes your game and shows a series of performance-related overlays on top of the game screen, displayed in either full screen or windowed views, which is often quite handy.

The appearance of elapsed game time is regulated, and it communicates directly with the Nvidia display driver. This allows you to see a number of interesting facts about how your game renders and gives you insight about performance bottlenecks. You can run around in your game for as long as you like and use a pre-defined hotkey to engage or disengage NVPerfHUD.

NO INDUSTRIAL ESPIONAGE

Due to NVPerfHUD's investigative abilities, it could easily be used for finding asset handles or competitive analysis, so Nvidia implemented some clever and unobtrusive safeguards. For example, you can't analyze a game unless you specifically allow it by having the code look for a video device named NVPerfHUD; and the application must turn on software rendering.

The NVPerfHUD device will then use the hardware to render. This prevents unauthorized analysis of a developer's application and keeps a developer from accidentally leaving this set, as it will not run on any machine without the DirectX SDK installed—and even if so, it will run at 1fps or so due to the software rendering setting.

NVPerfHUD has many diagnostic features, but here I'll only highlight the items on each of the four main NVPerfHUD modules that I've found the most useful.

FOUR MODES

There are four main modes in NVPerfHUD 4: Performance Dashboard, Debug Console, Frame Debugger, and new to v4, the Frame Profiler.

Performance Dashboard. This module displays a series of useful graphs showing number of draw calls, time spent in the GPU, and how much texture memory and AGP memory is being used. I use this module primarily for auditing how much texture memory is in use (for example, to figure out if the game will run on a 64MB or 128MB video card) and if I'm hitting my draw call budgets and batch sizes.

One handy use of the performance dashboard is that it lets you set options in the game that reduce texture usage or draw calls, or improve batching and verify that it works. This module also provides graphics pipeline experiments. These experiments allow you to change how things are drawn on a temporary basis via a set of hotkeys. These can be used to quickly identify any number of bottlenecks in performance.

The performance dashboard also shows a resource creation monitor that indicates when items like textures or vertex/index buffers are created. A few times the tool indicated that we were accidentally destroying and recreating mesh data or render targets every frame when it was unnecessary.

New for NVPerfHUD 4 is a nice ability to slow down the time passing in the game so you can catch suspect bad frames before switching to Frame Debugger mode. This feature has helped us a few times in catching intermittent issues.

NVPerfHUD 4's Frame Profiler can identify the relative cost of each draw call in a frame and reports where it is bottlenecked.

Debug Console. The Debug Console screen puts up a dialog that shows any debug runtime messages from DirectX, including performance warnings and custom messages along with when it was generated in the frame. Any message piped to OutputDebugString() shows on this screen. Yes, programmers can already see this in Visual Studio, but it's handy to be able output messages during a frame that both programmers and artists can see.

Frame Debugger. A build of a game goes through a lot of DirectX code to draw one frame. NVPerfHUD can help you see what is happening in the video card and driver for each frame. The frame debugger mode lets you scrub though a single frame, one draw call at a time, and inspect what's happening at an incredibly fine granularity. This is in my opinion, the most day-to-day useful function of NVPerfHUD.

Once you enter the Frame Debugger mode, you see a scrollbar on the bottom of the screen that lets you scrub back and forth and see the frame being built. By itself, this can be useful to see if any strategies for drawing front to back or occlusions are working.

Entering the advanced mode on this module allows you to examine the mesh used on each draw call, the pixel vertex shader in use, and any textures bound to

this mesh. This module was useful for me mostly for auditing textures and shaders. However, it's also extremely useful for shader/effect development and debugging.

As you scroll through the frame, you can see all textures in use. With NVPerfHUD, you can check that they aren't too large for their current use, that textures which should have mipmaps, do, and that the most appropriate texture format is used.

Examining the shaders can help you determine if the wrong shader was bound without the right number of lights, or was missing the specular math, for example.

During shader development, it's very handy to see the constants being fed to the shader, and that you aren't accidentally feeding in out of range or NAN numbers. New for NVPerfHUD 4, one can examine the entire matrix palette for hardware skinned meshes, which is useful if your figures are rendering screwy and you want to ensure that the matrix info that the shader is receiving is correct.

When programming for render targets, such as with a bloom-like effect, it is not uncommon for the effect to do several renders to textures that are in turn used as input into the next render-to-texture effect. NVPerfHUD allows you to see the effect rendering step by step, so you can see how you got to the final render.

Frame Profiler. Also new to NVPerfHUD 4 is the Performance Profiling module. This feature only works on GeForce 6 Series or later GPUs that can report the information necessary for this special module. The module presents information broken down in a way that I never expected.

Essentially, Frame Profiler repeatedly runs a frame from your games a few times and identifies the relative cost of each draw call and reports where it is bottlenecked. This is done using the GPU Hardware Performance Counters. On the surface, this may sound simple, but it's incredibly hard to isolate the relative cost of each draw call without a tool that automates the process for you.

NVPerfHUD produces a series of graphs that shows the cost in milliseconds for each draw call, and how many pixels it drew to the screen. You can group the results of various methods in buckets.

During one profiling session, I found that that rendering of the terrain was taking up most of the frame drawing time. Depending on view angle, terrain often covers a large area of the screen, and often, close patches of terrain draw on top of patches in the distance. The Performance Profiling module indicated that the pixel shader cost for terrain was excessive. I was able to improve performance by drawing the terrain patches front to back and by making the pixel shader run faster.

FUTURE OF NVPERFHUD

There are a few features I'd like to see Nvidia add to the next release of NVPerfHUD. For example, resource monitors for shader creation, cache misses (to show when the video card's cache is exceeded), and texture density usage. The NvidiaFXComposer has an interesting UVInspector shader that basically shows by color if one is using texture sizes that are appropriately sized for the number of pixels rendered to the screen, but I'd still like to see the functionality incorporated into the graphics pipeline.

Improving framerate is tricky. It's an ever-changing situation that is dependent on CPU and GPU. Once a bottleneck is eliminated or mitigated, a new one rises up.

NVPerfHUD is an excellent free tool to gain insight on current bottlenecks on the video card and helping plan solutions.

MATT SHAW *is chief technology officer at EA Mythic in Fairfax, Virginia. He has worked in the game industry for 15 years. Email him at* ***mshaw@gdmag.com****.*

SLICKEDIT'S SLICKEDIT 11
By Tom Plunket

SLICKEDIT 11 IS A POWERFUL AND HIGHLY configurable text editor. This newest version comes with a number of enhancements over the previous release, plus one particularly interesting new feature: Code Templates. Earlier releases already boasted a large feature set and this addition fills a useful slot in that list. This extensive functionality comes at no small cost, however, as the demands that SlickEdit makes of its users are considerable.

MEAT AND POTATOES

On the one hand, SlickEdit is amazing. The list of promises it makes to a potential user is huge. It can load Visual Studio's solution and project files and do a bit of C++ refactoring. It also has a context tagging system that allows trivial navigation of large codebases while supporting 42 different languages

SLICKEDIT

STATS
SlickEdit
3000 Aerial Center Pkwy., Suite 120
Morrisville, NC 27560
919.473.0070 or 1.800.934.EDIT
www.slickedit.com

PRICE
$139 upgrade from v9 or v10
$284 electronic download
$299 shrink-wrapped
$60 additional support fee, optional

SYSTEMS SUPPORTED & DISK SPACE REQUIRED.
- Microsoft Windows XP, 2000, NT, Me, 98: 170 MB.
- Linux Kernel 2.4 and later: 200 MB.
- AIX 5 and later: 250 MB.
- HP-UX 11 and later: 270 MB.
- IRIX 6.5 and later: 280 MB.
- Solaris SPARC 7 and later: 240 MB.
- Mac OS X v10.3 and later: 190 MB.

SUPPORTED LANGUAGES AND FILE TYPES
Ada; ANTLR; AppleScript; AWK; Bourne shell scripts; C, C++; C Shell; C#; CFScript; Ch; CICS; COBOL; DB2; DTD; Fortran; High Level Assembler, HTML, CFML; IDL; InstallScript; INI, config files; J#; Java; JavaScript; JCL; JSP; Lex; Makefile; Modula-2; Objective-C; Pascal; Perl; PHP; PL/I; PL/SQL; PowerNP Assembler; Progress 4GL; PV-Wave; Python; REXX; Ruby; SAS; Slick-C; Tcl; Transact SQL; Visual Basic, Visual Basic .NET; VBScript; Verilog; VHDL; Windows batch files; x86 Assembly; XML, XSD; Yacc.

PROS
1. Browsing code is really easy.
2. Integrates readily into existing work environments.
3. High level of configurability.

CONS
1. Difficult to configure optimally.
2. Many features are difficult to use or don't work as expected.
3. Support channels not highly responsive.

for context coloring (and adding new ones is straightforward).

It can emulate all of the major editors' keyboard handling, from Brief to Emacs, vi to Developer Studio, and on top of all this, SlickEdit is highly configurable by the user.

Many aspects of the program are written in its own scripting language, allowing the user to do anything from writing extensions to changing core functionality. The largest new feature, Code Templates, allows the user to create common structures for files and use these templates for creating new files. For example, you can make a template for "create a new game class" which could create default layouts for header, implementation, and test files.

The templates can also prompt for input, such as for the class name to be built, which reduces the amount of time that a developer spends tediously keying in boilerplate.

TOO MUCH SAUCE

On the other hand, the price of all of this power is that SlickEdit ends up being a very complex piece of software. Some of its most promising features are hard to use, and configuring the program to work to any one developer's tastes can be tedious.

On top of it all, I experienced a number of issues with fundamental functionality that went unresolved. The font rendering had numerous graphical bugs; dot crawl on my LCD displays was pretty terrible just from moving the cursor around with the arrow keys on my keyboard. At one point, SlickEdit started rendering all document text as black-on-black, which I found I could work around by simply not having the window maximized. SlickEdit has since noted that my monitor setup may have caused this problem.

The "thorough" C++ refactoring is heralded as one of SlickEdit's unique and market-leading features. However, it never worked on any of the codebases I tried. Attempting to do these operations lead to a report of "unexpected EOF found," which is not the most helpful message in a project with a thousand files.

In addition, the UI often behaved inconsistently, for example, by disallowing operations that should be available for no apparent reason, but only sporadically.

PROOF IN THE PUDDING

Altogether, SlickEdit shows promise and has many useful features. However, its power can only be harnessed by people who are ready to spend considerable time learning how to truly use it.

For users who want a text editor that works right out of the box, SlickEdit will not provide a comfortable work environment. Users who are happy to write their own modifications to the editor (using a C-like scripting language) and who are willing to spend the time to modify their work habits to suit a highly-powerful text editor would find a lot to like in this tool.

SlickEdit offers a 15-day demo on the company's web site. This is enough time for an introduction to the tool, but not really enough to do much more than kick the tires.

Developers who are working with makefiles and standalone debuggers would find SlickEdit a great primary editor. For developers who frequently jump from new project to new project, the context tagging and browsing systems can help you understand and navigate the codebases quickly.

People already using an integrated development environment, however, may find the benefits of SlickEdit less obvious. Additionally, while the editor can context-color many languages, overall support for these languages is sketchy at best.

My current project has a number of tools written in Python and relies on test-driven development, but the options to redirect output into the editor's output pane resulted in not being able to even start the scripts from within the environment. Given that I already had a fair number of support requests unanswered, I didn't even send a query about this issue.

SlickEdit set up a web-based community forum to help support the program as this article was going to press. It's still too early to tell what value that forum will provide, but it is certainly a step in the right direction.

CHECK, PLEASE!

It's a mixed bag, then, for SlickEdit. It is a powerful editor and it has a number of interesting and useful features. Unfortunately the package is marred by a few issues that don't have obvious solutions. Since a demo is available from the developer, the best suggestion is to try it before buying the software. Since it's a pretty complicated package, be sure you have the time (and interest) to learn and configure it effectively.

TOM PLUNKET *has been making games for ten years and splits his time between writing code and looking for ways to make the development process easier. Email him at* ***tplunket@gdmag.com****.*

Havok Behavior SDK and tool.

1. Export character assets

2. Load in Tool

3. Bring character to life

4. Put in your game

Saved: Countless hours.
Sneak out early.

Unlimited character development possibilities.

havok
www.havok.com
Stay Ahead of the Game!™

» david sirlin

THE POWER OF PACING

Designing games for lasting impact

» **IS IT BETTER TO SAVE YOUR BEST CONTENT UNTIL THE END** of a game so you have a strong finish, or is it better to make the first few minutes of gameplay as good as they can possibly be? If your best stuff only shows up after the player has invested 20 hours, some reviewers and players might not even know it's there. But if you "give away the farm" on the first level, the game has nowhere to go but down.

The general trend I see in critically and financially successful games is that they tend to show a great deal of their coolness (but not all of it) in the first few minutes to half hour of gameplay. Let's look at some case studies.

GIVE UP THE GOODS

Metroid Prime. As of this writing, Metroid Prime is the fourth highest rated game of all time on gamerankings.com, having received a 9.7 from GameSpot, a 9.8 from IGN, and a perfect 10 from EGM. The game sold 1.3 million units on GameCube in North America, according to TRST data.

The first few minutes of Metroid Prime show off an amazing amount of the game. We learn basic movement (R button shifts to freelook, L button shifts to strafe/lock-on). A button shoots and B button jumps. If you hold the A button, you get a fancy charge-up shot, while the Y button fires missiles. The X button turns you into a ball (with third-person camera) equipped with bombs (that make you bounce), and everyone loves rolling around as that ball. Your visor lets you scan the world to get info and tips and even open some doors. Most doors you just shoot to open, and your charged shot can be used to clear rubble. You also learn how to operate elevators and use the save stations. After only a few minutes, you fight a boss where you learn how to circle strafe while locked on and dash sideways during a lock-on. A few seconds after that, you get to use your grapple gun. (It was probably a mistake that Retro Studios had you use the grapple gun for the first time during a timed sequence, but that was their choice.) You also get a taste of Metroid Prime's map and mini-map, which are probably the best in-game maps of 3D levels the industry has seen yet.

That's an incredible number of cool features revealed in the first few minutes of the game. It makes you realize right away that Metroid Prime is a class act that deserves your time. After the intro sequence, your character gets

DAVID SIRLIN *is a producer and game designer at Backbone Entertainment. He's a multiple-time national Street Fighter tournament champion, author of* Playing to Win, *and co-organizer of the Evolution Fighting Game Championships national tournament series. You can reach him at* ***dsirlin@gdmag.com****.*

damaged and loses access to the morph ball, charged shot, missiles, and grapple gun. The game gives you back these items slowly over time to reward you, but the designers wanted to make sure your first few minutes were packed with coolness, so they gave you a great taste of what was to come.

Grand Theft Auto 3. This infamous breakthrough title conveys its core ideas in the first few minutes. The game starts with a very short series of three missions: First, get in the car and drive your buddy to Point A. Now drive to Point B. Finally, pick up a certain passenger at the hospital and take her to Point C. This sequence teaches you how to get in and out of cars, basic driving (gas, brake, turning), how to change the radio station in the car, how to pick up passengers, how to get a new car if your current car gets too damaged, and how to use the mini-map to find mission objectives.

After those first three missions, the game turns you loose into the world to do whatever you want. Doing whatever you want is the core concept of Grand Theft Auto 3, and the player realizes it right away. You can drive anywhere. You can fight people on the street and take their money. You can crash cars, wreck environments, and steal newer cars. You can totally ignore the story and mission structure and make up your own story and missions. In doing so, you quickly learn about the police "star" system where committing worse and worse crimes increases the force of police who are sent after you. You have to hide out or find secret police stars hidden in the world to reduce your infamy rating and get the cops off your back.

Early on in the game, Metroid Prime gives players a taste of their future abilities.

I've watched several people play GTA3 for the first time, and all of them abandoned the game's mission structure within five minutes to explore the world and create their own goals. No wonder it sold over 5.6 million units on PlayStation 2 alone (and quite a bit more than that once you factor in other platforms and expansions).

WITHHOLDING THE BAG OF TRICKS

Castlevania and God of War. Castlevania: Symphony of the Night (PlayStation) and God of War (PlayStation 2) are both examples of a near ideal distribution of "good stuff." Both games start by showing you a large portion of the game mechanics.

Castlevania uses the same trick as Metroid Prime where the players get to start with a bunch of cool moves and weapons that they won't get to use again until much later. God of War introduces basic fighting, ground throws, air throws, opening hatches, walking tightropes, a boss fight, special finishing moves, and use of magic all within the first few minutes. Note that the best boss is the first one (the Hydra) and the most fun and effective magic power is the first one you get, (Poseidon's Rage, the 360 degree lightning attack). Each of these games is putting its best foot forward to get your attention from the start.

Grand Theft Auto 3 teaches the basics, then unleashes players into the city.

The interesting thing is that these games feel great right off the bat, but they don't immediately feel like the 9/10 or 10/10 games that they are. In each case, something later in the game takes the quality from 7 or 8 up to 9 or 10. In God of War's case, it's the emotional content of the excellent story that builds to a very satisfying conclusion. Even though it's a "fight a bunch of guys" game, the story and presentation elevate it to the status of a memorable experience, rather than just a brawler.

Castlevania: Symphony of the Night has one of the biggest surprises in games, and if you haven't played it, I'm about to ruin it. The game leads you to believe you've reached the end when you find Dracula and kill him. There's a map that keeps track of what percentage of the game you've visited, and it approaches 100 percent by the time you reach the big boss.

Castlevania's game world was designed to support inverted play—shown here is an almost complete map of the regular castle.

PSYCHONAUTS started out slow on the gameplay front, saving its best moments for later in the game.

The surprise is that you're only at the halfway point. The boss's defeat causes the castle (the entire game world) to flip upside down, and you must now play through it all over again, this time walking on what used to be the roof. Chandeliers stick up from the ground, you walk on the undersides of stair cases, and you begin to realize that the entire game was planned from the start to support an entirely different upside down game. All the enemies are replaced with harder ones, and the various keys are hidden in new places. The awesome design of the upside down world elevates CASTLEVANIA to a very memorable experience.

So if starting out strong, but ending stronger is the key to victory, then what are some examples of games that break this trend? Games that start out weak and end up weak aren't very informative here, but games that start weak and end strong would be great examples. We'd expect those games not to sell very well.

COULD'VE BEEN A CONTENDER

PSYCHONAUTS. I hate to pick on PSYCHONAUTS because Tim Schafer's great writing is one of the reasons I joined the game industry in the first place. That said, the first 12 minutes of PSYCHONAUTS are, from a gameplay perspective, a very poor experience. The only interactive things I did in those 12 minutes were enter my name, move the camera to the right and then up one time each in a tutorial, and walk two steps to an NPC that triggered even more movies. The rest of the 12 minutes was all movies. I just wanted to play the game. What's just as bad is that after two hours, I didn't get even a single Psi-power. Only after about three hours did I get to see anything that set this platformer apart from any other platformer, and the real interesting stuff isn't until much later in the game. Even though

many people told me that the game has wonderful ideas and cool gameplay as you get into it more, the first time I played it, I never made it past the first 12 minutes.

Traditional RPGs. Most traditional (FINAL FANTASY-like) RPGs also suffer from this phenomenon. These games tend to start you out with a wooden sword and no spells, and have you fight a few rats or something. Over time, your arsenal of spells and attacks increase and you usually get the ability to do combos of spells (or use your party members together in combos) that are pretty interesting and fun. This fun tends to come later in the game though, at hour five rather than minute five. This is perhaps why the single-player RPG genre isn't selling as well as it used to, except for games called FINAL FANTASY or that have the *Star Wars* license.

PACE FOR IMPACT

It should be no surprise that you need to start out strong, or at least strong enough to grab the player's attention. Burying the best content at the end is generally not a good idea, but it's a question of degree. If your final boss is a 9/10 experience, but your first level is a 2/10, you have a major problem because no one is going to see that final boss.

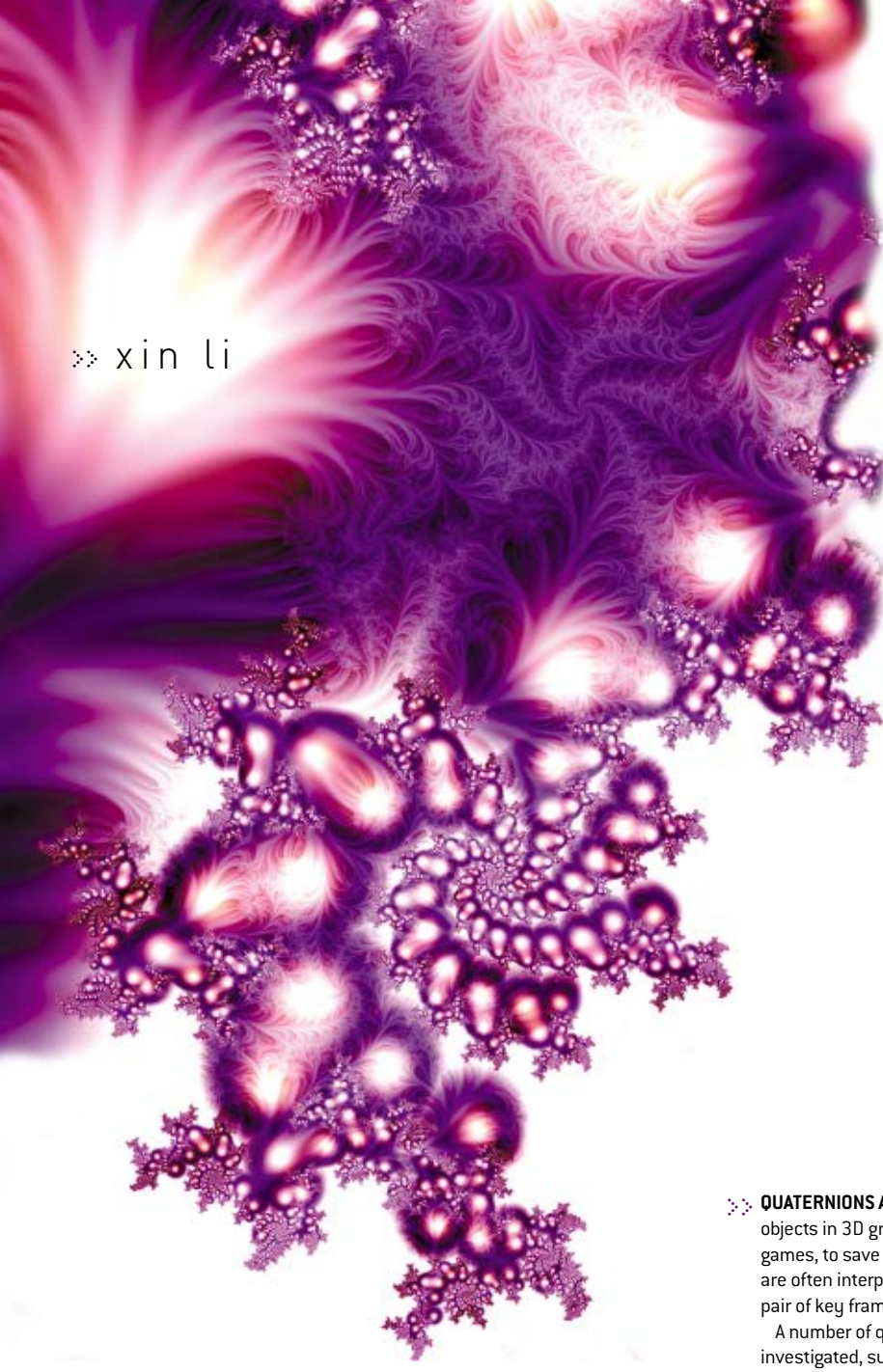
On the other hand, if you can get that first level up to a respectable 8/10 experience, then ending on a 9/10 boss is great, and perhaps nearly the ideal scenario. In any case, don't be tempted to save all the fun until after the 20 hour mark because your first level is going to be your most played and most judged one. Go the extra mile to make it stand out, even if it means giving the player a preview of a few fun mechanics you planned on saving for the end.

If you really want to start with a bang, then get rid of all that junk that players are tired of waiting through when they turn on a game. Get rid of the intro movie with the publisher's logo, the intro movie with the developer's logo, the legal screens (put them as an option on the main menu), and any other non-content fluff you can find. It's getting totally out of hand how many screens of garbage games start out with before getting to the main menu. You're much better off building brand awareness by actually making a good game than forcing everyone to see your logo every time the game boots. Put your logo in the main menu if it matters so much. When I pay $50 for a game, I expect to be exempt from even five seconds of this stuff.

I predict that no games will follow this advice in the years 2006 or 2007. Yours could be the first. Oops, I saved my most interesting idea for the end of the article where no one will really see it.

GOD OF WAR has good pacing, but the emotional impact is what makes it great.

Traditional RPGs such as FINAL FANTASY VII start players off slowly, then ramp up to more complex gameplay later.



xin li

TO SLERP, OR NOT TO SLERP?

QUATERNIONS ARE WIDELY EMPLOYED TO REPRESENT ROTATIONS OF objects in 3D graphics. In real-time simulation software and video games, to save the space occupied by the animation data, quaternions are often interpolated to generate intermediate orientations between a pair of key frames.

A number of quaternion interpolation algorithms have been investigated, such as Lerp, Slerp, nLerp, log-Lerp, polySlerp, cubic interpolation, and so forth (see References) to meet different efficiency and accuracy requirements. Among those algorithms, the linear interpolation (Lerp) is the simplest and fastest method. However, it changes the angles with a non-constant velocity and does not always produce unit quaternions. At the high end of the spectrum, spherical linear interpolation (Slerp) interpolates a pair of quaternions on the surface of a unit sphere. The intermediate quaternions follow a geodesic path with constant velocity.

Unfortunately, the application of Slerp in video games and real-time simulation software is limited due to its high computational cost, thanks to two trigonometric functions that are a hundred times more expensive than simple floating-point number additions and multiplications (see Intel 2006 in References). To eliminate those trigonometric functions, Jonathan Blow did a thorough investigation on other approximation alternatives in 2004 (see References). He further concluded that everyone should understand Slerp, then not use it.

Currently, Slerp is considered by most game programmers as the

DR. XIN LI *teaches animation and real-time rendering techniques at Digipen Institute of Technology. Prior to the faculty position, he had seven years experience with Lockheed Martin and three years with Nintendo Software Technology, making real-time simulation software and video games. Email him at* ***xli@gdmag.com****.*

authoritative method for rotation interpolation since it serves us well with its superior geometrical characteristics. But the quality services are not surprisingly associated with high cost. For this reason, we are now told to stop Slerping.

Are we doomed?

INCREMENTAL APPROACHES

This article will introduce a few so-called incremental approaches of quaternion interpolation algorithms. With the assumption of a constant angle increase between a pair of given quaternions, the cost of Slerp can be significantly reduced. Specifically, the trigonometric calculations will be replaced by a few inexpensive arithmetical operations.

The algorithms presented in this article should be interesting to game developers who work with dynamic object simulations, such as character animation, camera control, or inverse kinematics. Although the content might seem a bit "mathematically involved," the equations are mostly just trigonometry, complex numbers, and some simple vector calculus—nothing really beyond high school tricks.

Readers should be familiar with the basic concept of the quaternion and its operations (normalization, inverse, multiplication). Covering quaternion preliminaries is beyond the scope of this discussion. If you feel rusty in your quaternion skills, try David Eberly's "Quaternion Algebra and Calculus" (see References) as a refresher.

I will try to keep the "facts" separated from the "causes." In other words, I will encapsulate proofs of the important formulae in appendices, which can be found on www.gdmag.com, and simply skip the trivial ones.

After reading this article, you should understand the incremental approaches of Slerp and feel comfortable implementing one.

HOW TO INCREMENT AND WHY

Often, an augmentative method can be applied to simplify the complexity of a problem and reduce the cost. This approach takes advantage of the gradual and progressive procedure. It often benefits from the knowledge obtained in the previous steps.

Some examples include incremental sorting, incremental searching, incremental linking, and incremental rendering. To avoid confusion, let me use the most relevant example—the linear interpolation algorithm—to illustrate the concept of incremental approaches.

The original Lerp equation is given as Equation 1.1.

$$q_t=(1-t)q_0+t\,q_n, \quad t\in[0{:}1] \qquad \textbf{EQUATION 1.1}$$

It creates a linear and geodesic path between two quaternions, q_0 and q_n. The approach is simple and efficient. However, we can demonstrate how Lerp will be implemented in an incremental approach to further reduce cost.

$$\begin{aligned} q_{t=0} &= q_0 \\ q_{t+\Delta t} &= (1-(t+\Delta t))q_0+(t+\Delta t)q_n \\ &= (1-t)q_0+t\,q_n+\Delta t(q_n-q_0) \\ &= q_t+\Delta q \end{aligned}$$

where $\Delta q=\Delta t\,(q_n-q_0)$ is a constant quaternion if q_0 and q_n are given and interval parameter t is fixed. You see, the computation of Lerp is reduced to a single quaternion addtion, based on the quaternion generated in the previous step. See Figure 1 for its geometric intuition.

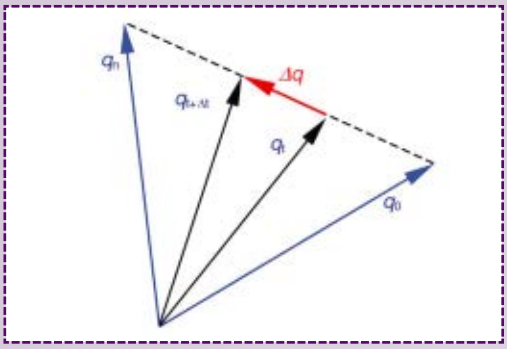


FIGURE 1 **Incremental Lerp: Computing next quaternion q_{t+1} by adding q_t and Δq.**

SOME PRELIMINARIES

To understand incremental Slerp approaches, we need to take a look at different variations of the spherical linear interpolation formulas. We start with the original Slerp equation:

$$q_t=(1/\sin(\alpha))(\sin(\alpha-t\alpha)q_0+\sin(t\alpha)\,q_n), \quad t\in[0{:}1] \qquad \textbf{EQUATION 2.1}$$

where

1. $q_0=[s_0, v_0]$ and $q_n=[s_n, v_n]$ are given unit quaternions, and
2. α is the angle between q_0 and q_n. That is, $q_0\cdot q_n=\cos(\alpha)$.

The equation interpolates two quaternions when parameter t changes from 0 to 1. If we assume that the angle between q_0 and q_n will be interpolated incrementally through n steps with a fixed interval, then the angle between any intermediate quaternions q_k and q_{k+1} is a constant $\beta=\alpha/n$. Under this assumption, when t varies from 0 to 1, we always have $t\alpha=k\beta$ with $k=0, 1, \ldots, n$.

This allows us to replace $t\alpha$ with $k\beta$ in Equation 2.1 and obtain

$$q_k=(1/\sin(\alpha))(\sin(\alpha-k\beta)q_0+\sin(k\beta)q_n), \quad k=0, 1, \ldots, n \qquad \textbf{EQUATION 2.2}$$

Figure 2A depicts the idea.

We now expand Equation 2.2 using trigonometric formulas in the following way:

$$\begin{aligned} q_k &= (1/\sin(\alpha))(\sin(\alpha-k\beta)q_0+\sin(k\beta)q_n) \\ &= (1/\sin(\alpha))(\sin(\alpha)\cos(k\beta)q_0-\sin(k\beta)\cos(\alpha)q_0 \\ &\quad +\sin(k\beta)q_n) \\ &= \cos(k\beta)q_0+(\sin(k\beta)/\sin(\alpha))(q_n-\cos(\alpha)q_0) \end{aligned} \qquad \textbf{EQUATION 2.3}$$

If we assume

$$\hat{q}_0=(1/\sin(\alpha))(q_n-\cos(\alpha)\,q_0) \qquad \textbf{EQUATION 2.4}$$

and replace the right-hand side of Equation 2.4 with $\hat{q}_0$ in Equation 2.3, it becomes

$$q_k=\cos(k\beta)q_0+\sin(k\beta)\,\hat{q}_0, \quad k=0, 1, \ldots, n \qquad \textbf{EQUATION 2.5}$$

www.gstar.or.kr

G 2006
Game Show & Trade, All-Round

The entire game industry all under one roof!
Your Success in Focus!

G★2006, Asia's premier game expo!
Looking to make it in the game market? Act now.

November 9~12, 2006 Korea International Exhibition Center (KINTEX)

Enhance Your Brand

G★2006 has become one of the world's top 3 game expos in terms of size and number of visitors. By offering business networking opportunities and various programs, G★2006 will make your business a more popular, competitive and streamlined operation.
Just enjoy your business opportunities here!

Big Names and Big Issues

At dozens of seminars and conferences you'll find great ideas using the latest technology for more profitable business opportunities. Take advantage of the experiences and insights from industry leaders! You will be able to put it all to work for you right away.

Expose Your Product and Make Amazing Business Connections

Last year, more than $200 million in sales can be attributed to purchase plans made from 750 business meetings during the G★2005. G★2006 is expected to have 5,000 worldwide game publishers from 15 different countries and will offer a variety of business support programs such as business match-making, IR seminars and more. You will return to your company with of thousands of business results with qualified buyers in your hands.

Overview

- Dates : November 9~12, 2006
- Venue : Korea International Exhibition Center (KINTEX)
- Scale : 53,541sq m, 180 exhibitors, 2,000 booths, 160,000 attendees
- Organizer : G-star Organizing Committee
- Supported by: Ministry of Culture & Tourism
  Ministry of Information & Communication
  Gyeonggi Provincial Government
  Korea Game Development & Promotion Institute
  Korea IT Industry Promotion Agency
  Korea Trade & Promotion Corporation
  Korea Association of Game Industry, etc.

Exhibits

- PC/On-line games
- Wireless/Mobile/PDA games
- Video games
- Arcade games
- Peripherals
- Game software & technologies
- Entertainment/Education games & software
- Multimedia content
- Others

Participation Method

- Application : Fax, E-mail, Post delivery
- Submission : Application form,30% Deposit
- Payment : USD Only

※ Full payment by Aug. 31, 2006

G-star Organizing Committee : 31-6, Technomart, Guui-dong, Gwangjin-gu, Seoul, Korea Tel : +82 (0)2 3424 4091~5 Fax : +82 (0)2 3424 5049 e-mail : info@gstar.or.kr

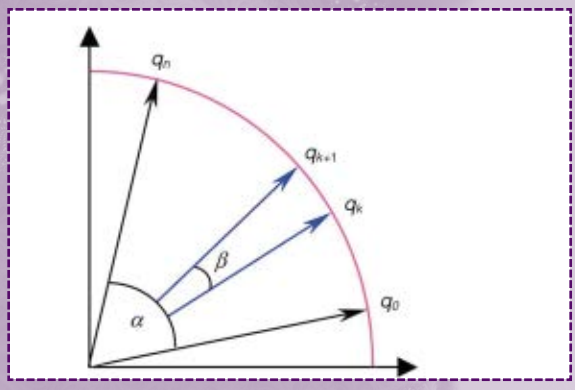


FIGURE 2A Incremental angle β between two interpolated quaternions is always a constant.

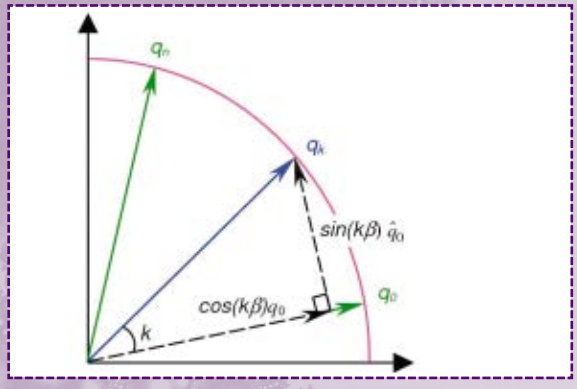


FIGURE 2B Interpolated quaternion q_k is expressed as a combination of scaled q_0 and its tangent $\hat{q}_0$.

It is important to point out that $\hat{q}_0$ defined in Equation 2.4 is a unit quaternion as long as q_0 and q_n are unit. Furthermore it is always perpendicular to q_0. (I trust the reader to prove these properties.) Thus we refer to $\hat{q}_0$ as the tangent quaternion of q_0. With this notation, Equation 2.5 simply states that an interpolated quaternion by the Slerp function can be represented by the combination of a scaled quaternion and its tangent quaternion (see Anders Hast in References). Figure 2B illustrates the geometric intuition.

Equations 2.1, 2.2, and 2.5 produce the same sequence of interpolated quaternions once given the same q_0, q_n, and n (the desired number of intermediate steps). They also take roughly in magnitude the same amount of time to compute since each contains two trigonometric functions. No savings in that regard.

POWER IT UP

Now we are ready to look at the first incremental approaches. We start with a method that utilizes the power operation defined on quaternions. Let q_0 and q_n be two unit quaternions. Let $q_0^{-1}=[s_0,-v_0]$ be q_0's inverse quaternion. Since $q_0^{-1}q_0=1$, we have

$$q_n=q_n(q_0^{-1}q_0)=(q_n\,q_0^{-1})\,q_0 \qquad \textbf{EQUATION 3.1}$$

Remember, by the definition of the quaternion multiplication,

$$q_n\,q_0^{-1}=[s_0s_n+v_0\cdot v_n,\ s_0v_n-s_nv_0+v_0\times v_n]$$

Note that the scalar part of the multiplication is equal to $\cos(\alpha)$. If we assume some unit vector $u=(s_0v_n-s_nv_0+v_0\times v_n)/\sin(\alpha)$, then the vector part of $q_n\,q_0^{-1}$ can be rewritten as

$$\sin(\alpha)u=(s_0v_n-s_nv_0+v_0\times v_n)$$

and thus

$$q_n\,q_0^{-1}=[\cos(\alpha),\quad \sin(\alpha)u]$$

How do we know that u is a unit vector? One can prove (Appendix 3.1) that $|s_0v_n-s_nv_0+v_0\times v_n|=\sin(\alpha)$ and therefore $|u|=1$.

Now, replacing in Equation 3.1, we have

$$q_n=(q_n\,q_0^{-1})\,q_0=[\cos(\alpha),\ \sin(\alpha)u]\,q_0 \qquad \textbf{EQUATION 3.2}$$

Plug it into the second form of the Slerp equation (Equation 2.2) and obtain

$$\begin{aligned} q_k &=(1/\sin(\alpha))\,(\sin(\alpha-k\beta)\,q_0+\\ &\quad \sin(k\beta)[\cos(\alpha),\sin(\alpha)u]q_0), \qquad k=0,1,\ldots,n\\ &=\ \ldots\ldots\\ &=[\cos(k\beta),\sin(k\beta)u]\,q_0 \end{aligned} \qquad \textbf{EQUATION 3.3}$$

You might have noticed that I skipped quite a few steps in Equation 3.3. I did not intend to drown anyone in the muddy pond of math derivations. If you are really interested in testing the water, just remember that $[\cos(k\beta),\sin(k\beta)u]$ is a quaternion. All you need to do is some trigonometric manipulations and follow the definition of the quaternion multiplication. Work from both ends and you will eventually meet in the middle. (See Appendix 3.2.)

Are we there yet? Not quite. We need one more formula

$$[\cos(k\beta),\sin(k\beta)u]=[\cos(\beta),\sin(\beta)u]^k,\quad k=1,2,\ldots,n$$

It simply states that the quaternion defined by trigonometric functions of multiples of an angle, namely $k\beta$ (on the left hand side), can be replaced by the multiplications of a constant quaternion (on the right side) if β is fixed. The equation can be easily verified by mathematical induction. Now, Equation 3.3 becomes

$$q_k=[\cos(k\beta),\sin(k\beta)u]q_0=[\cos(\beta),\sin(\beta)u]^k q_0,$$

or

$$q_k=q_c^{\,k}q_0,\qquad \text{where } q_c=[\cos(\beta),\sin(\beta)u]$$

The good news is q_c is a constant quaternion for given q_0, q_n, and n. In other words, this quaternion power function permits the Slerp to be performed incrementally:

$$\begin{aligned} &q_1=q_c\,q_0;\\ &q_2=q_c^{\,2}q_0=q_c(q_c\,q_0)=q_c\,q_1\\ &\quad\ldots\ldots\\ &q_{k+1}=q_c^{\,k+1}q_0=q_c(q_c^{\,k}\,q_0)=q_c\,q_k \end{aligned}$$

The idea of interpolating quaternions by a power function is pretty simple. Remember that the quaternion multiplication represents the concatenation of rotations. It is as if we are turning a clock backward. From q_0 to q_n, we rotate the minute hand each time by a fixed angle β counterclockwise from its previous position. This "incremental rotation" is expressed by a constant quaternion q_c and concatenated with q_k to obtain q_{k+1}. See Figure 3 for an illustration and Listing 1 for the psuedo code, available online at www.gdmag.com.

The cost to compute the next quaternion is only one quaternion

multiplication. Hurray, no trigonometric functions! (q_c can be pre-computed once and stored with the animation data.)

iSLERP (NOT AN iPOD ACCESSORY)

Can we do better than a quaternion multiplication? Yes. Let me introduce another incremental approach to Slerp-ing (see Li in References). In this method, with each interpolated quaternion q_k ($k=1, 2, ..., n-1$) between q_0 and q_n, an additional quaternion $\hat{q}_k$ is also computed. With this "auxiliary," q_{k+1} can be very easily expressed as an addition of two scaled quaternions from the previous slerping step.

Let's first present the following equations:

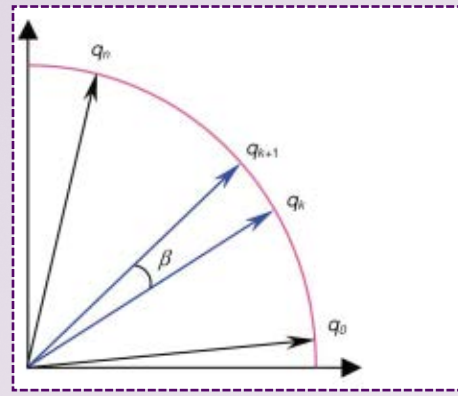


FIGURE 3 **Rotating quaternion q_k by β degree obtains q_{k+1}.**

$$q_k=(1/\sin(\alpha))(\sin(k\beta)q_n+\sin(\alpha-k\beta)q_0)$$
$$\hat{q}_k=(1/\sin(\alpha))(\cos(k\beta)q_n-\cos(\alpha-k\beta)q_0),\quad k=0, 1, ..., n \qquad \textbf{EQUATION 4.1}$$

The first equation recites the Slerp definition (see Equation 2.2). It computes the interpolated quaternion q_k between q_0 and q_1 for a given angle. The second equation specifies another quaternion $\hat{q}_k$. Since $\hat{q}_k$ is always perpendicular to q_k and has a unit length, we refer to it as q_k's tangent quaternion. See Appendix 4.1 for discussion.

You might have noticed that the terms of q_0 and q_n are swapped in the above equations. It is just a cosmetic concern.

With the above definitions, when $k=0$, we have

$$q_0=(1/\sin(\alpha))(\sin(0)q_n+\sin(\alpha)q_0)=q_0$$
$$\hat{q}_0=(1/\sin(\alpha))(\cos(0)q_n-\cos(\alpha)q_0)$$
$$=(1/\sin(\alpha))(q_n-\cos(\alpha)q_0) \qquad \textbf{EQUATION 4.2}$$

And they are extended to the following forms for $k+1$:

$$q_{k+1}=(1/\sin(\alpha))(\sin((k+1)\beta)q_n+\sin(\alpha-(k+1)\beta)q_0)$$
$$=\cos(\beta)q_k+\sin(\beta)\hat{q}_k$$
$$\hat{q}_{k+1}=(1/\sin(\alpha))(\cos((k+1)\beta)q_n-\cos(\alpha-(k+1)\beta)q_0)$$
$$=\cos(\beta)\hat{q}_k-\sin(\beta)q_k \qquad \textbf{EQUATION 4.3}$$

Equation 4.2 confirms Equation 2.4. Equation 4.3 shows the recursive forms of Equation 4.1. See Appendix 4.2 for its proof. It indicates that, along the geodesic path of interpolated orientations, the next quaternion q_{k+1} (and its counterpart $\hat{q}_{k+1}$) can be calculated based on two scalars ($\cos(\beta)$ and $\sin(\beta)$), the current quaternion q_k and its tangent quaternion $\hat{q}_k$. Figure 4 illustrates the geometric intuitions.

I encourage the reader to prove that, if q_k and $\hat{q}_k$ are orthogonal and unit quaternions, then q_{k+1} and $\hat{q}_{k+1}$ are also orthogonal and unit quaternions. Equation 4.3 enables us to process the Slerp procedure incrementally. See Listing 2, the pseudo code, on www.gdmag.com.

The cost to compute each interpolated quaternion (and its tangent) is reduced to four scalar-quaternion products, one addition and one subtraction. (C and S are pre-computed once and stored with the animation data.)

Comparing with the original Slerp algorithm, we already made significant improvements so far. But, we can do even better. Just read on.

IMAGINARY NUMBERS

Remember the study of the complex number from high school? A complex number c is defined by $c=a+ib$, where a and b are any rational numbers and $i=\sqrt{-1}$. a and ib are also referred as the real part and the imaginary part. They can be treated as two numbers on the orthogonal axes in a 2D space.

Now, let's assume that $a=\cos(\beta)$ and $b=\sin(\beta)$ in a complex number. By mathematical induction, one can easily verify that, for any natural number k,

$$(\cos(\beta)+i\sin(\beta))^k=\cos(k\beta)+i\sin(k\beta)$$

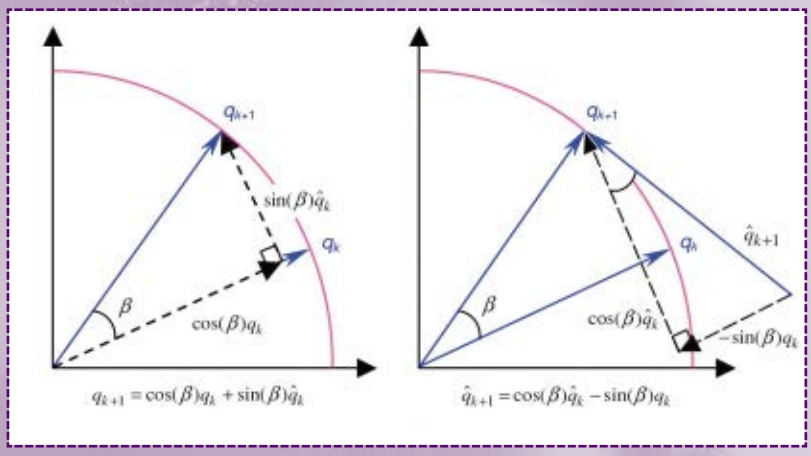


FIGURE 4 Quaternion q_{k+1} is expressed as a combination of scaled q_k and its tangent $\hat{q}_k$ (left). Tangent quaternion $\hat{q}_{k+1}$ is expressed as a combination of scaled q_k and its tangent $\hat{q}_k$ (right).

It is called De Moivre formula, useful in deriving trigonometric identities (see Saff and Snider in References). Can we count on this friend to reduce the cost of the Slerp interpolation?

Well, let's first define a new operation between a complex number and an addition of a pair of quaternions. We assume that $a+ib$ is a complex number and q_a and q_b are two quaternions. The dot product defines a linear combination between them and yields another quaternion q:

$$q=[a+ib]\cdot[q_a+q_b]=aq_a+bq_b$$

After we plug in $\cos(k\beta)$, $\sin(k\beta)$, q_0 and $\hat{q}_0$ it becomes the Slerp equation Equation 2.5). We now apply the De Moivre formula and obtain

$$\begin{aligned} q_k &= \cos(k\beta)q_0+\sin(k\beta)\hat{q}_0 \\ &= (\cos(k\beta)+i\sin(\beta))\cdot(q_0+\hat{q}_0) \\ &= (\cos(\beta)+i\sin(\beta))^k\cdot(q_0+\hat{q}_0) \end{aligned}$$

See Listing 3 on www.gdmag.com.

The cost for the next quaternion is one complex quaternion dot product and one complex number multiplication. α, β, complex number $[\cos(\beta)+i\sin(\beta)]$, and $\hat{q}_0$ can be pre-computed and stored.

CHEBYSHEV, I HARDLY KNOW HIM

We're almost done. This will be the last alternative of Slerp (see Barrera in References).

A Chebyshev recurrence relation represents a function that can be expressed by itself recursively in the following form:

$$T_{k+1}(u)=2uT_k(u)-T_{k-1}(u) \qquad \textbf{EQUATION 6.1}$$

The name might not be familiar and the formula might look too abstract, but it is not really that bad. In plain words, $T_k(u)$ defines a sequence of $T_0(u)$, $T_1(u)$, ... , $T_k(u)$, $T_{k+1}(u)$, ... , where u is the parameter and k is any natural number. Along that sequence, any term can always be represented based on previous two terms.

A simple example of the Chebyshev sequence is the list of odd numbers. Let $u=1$, $T_0(1)=1$, and $T_1(1)=3$, we have

$T_0(1)=1$;
$T_1(1)=3$;
$T_2(1)=2T_1(1)-T_0(1)=2\times3-1=5$;
$T_3(1)=2T_2(1)-T_1(1)=2\times5-3=7$;
.....

In our case of the quaternion interpolation, we can assume $u=\cos(\beta)$. Since $\cos(k\beta)$ and $\sin(k\beta)$ can be mathematically treated as functions of $\cos(\beta)$, for given q_0, q_n, and n, we can rewrite the Slerp equation (Equation 2.5) as the function of $\cos(\beta)$:

$$q_k(\cos(\beta))=\cos(k\beta)q_0+\sin(k\beta)\,\hat{q}_0, \quad k=1,2,...,n-1 \qquad \textbf{EQUATION 6.2}$$

But how does it help? Well, let's try to answer the following question first: Can we represent $q_1, q_2, ..., q_{k-1}, q_k, q_{k+1}, ..., q_{n-1}$, interpolated incrementally between q_0 and q_n by Equation 2.5, in a Chebyshev recurrence relation? More specifically, if we begin with

$$q_0(\cos(\beta))=\cos(0)q_0+\sin(0)\,\hat{q}_0=q_0$$

$$q_1(\cos(\beta))=\cos(\beta)q_0+\sin(\beta)\,\hat{q}_0$$

can we prove the following Chebyshev sequence?

$$q_{k+1}(\cos(\beta))=2\cos(\beta)q_k(\cos(\beta))-q_{k-1}(\cos(\beta)), \quad k=1,2,...,n-1 \qquad \textbf{EQUATION 6.3}$$

Luckily, we can. The proof is available in Appendix 6.1. If you still have a hard time comprehending the Chebyshev form of Slerp, Figure 5 gives the geometric explanation of why it works.

With Equation 6.3 in our bag of tricks, the interpolated quaternions can always be computed based on the previous two

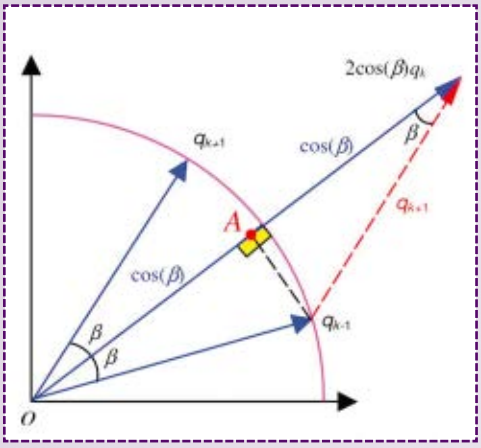


FIGURE 5

1. The dashed line (from q_{k-1} to point A) is orthogonal to q_k.
2.The length from O to A is $\cos(\beta)$.
3. The red-dashed arrow is parrallel and equal to q_{k+1}.
4. Thus $2\cos(\beta)q_k=q_{k-1}+q_{k+1}$.
5. That is: $q_{k+1}=2\cos(\beta)q_k-q_{k-1}$.

quaternions in the sequence. See Listing 4 on www.gdmag.com for the pseudo code.

The cost of this approach is cut down to only one scalar-quaternion product and one quaternion subtraction. Again, the rest can be pre-calculated once and stored with the interpolation animation data structure. Can it be any cheaper? Hardly!

TABLE 1: Comparison of different incremental Slerps.

METHOD	COST	BREAK-DOWN
By Power Function	1 quaternion multiplication	16 floating* 12 floating +/-
By Tangent Quaternion	4 scalar-quaternion products, 2 quaternion additions/subtractions	16 floating* 8 floating +/-
By Complex Number	2 scalar-quaternion products, 1 quaternion addition, 1 complex number multiplication	12 floating* 6 floating +/-
By Chebyshev Sequence	1 scalar-quaternion product, 1 quaternion addition/subtraction.	4 floating* 4 floating +/-

Efficiencies are broken down and compared by numbers of floating point multiplications, additions, and subtractions.

TO SLERP, OR NOT TO SLERP

It is important to understand that all four alternative Slerp methods that we discussed in this article are not approximations of the original Slerp function. They generate exactly the same sequence of interpolated quaternions as Slerp does. Therefore all intermediate quaternions possess the same geometric characteristics, i.e. unit length, constant velocity, and geodesic path. Besides, they are much more efficient than most quaternion interpolation algorithms, such as Slerp, nLerp, log-Lerp, polySlerp, cubic interpolation, just to name a few.

On the practical side, all these algorithms are very easy to implement. No evaluation of logarithmic, polynomial or spline functions involved. No parameter-tuning needed. Since the incremental interpolation only uses additions and multiplications, the calculation is immune to numerical instability.

Table 1 compares the cost of these incremental Slerp algorithms in numbers of floating-point number multiplications and additions/subtractions. Notice that even the linear interpolation, Equation 1.1, would take two scalar-quaternion products, one quaternion addition and one floating point number subtraction.

My conclusion is that since most games, if not all, use the fixed and pre-determined intervals when interpolating orientations during animations, since the incremental approaches are numerically stable, since they reduce the cost of Slerp equal to Lerp in magnitude, since some of them are very easy to understand and implement, and since all the superior geometric characteristics (constant velocity, geodesic path and unit length) are inherited, I would suggest that it is time for us to Slerp, well, incrementally.

I hope you are convinced.

Acknowledgment. The author acknowledges Dr. Matt Klassen at DigiPen for his review and valuable suggestions to this article.

REFERENCES

Barrera, Tony, et al. "Incremental Spherical Linear Interpolation," Proceedings of SIGRAD, vol. 13, 2004.

Blow, Jonathan. "Hacking Quaternions," *Game Developer*, March 2002.

Blow, Jonathan. "Understanding Slerp, Then Not Using It," *Game Developer*, April 2004.

Busser, Thomas. "PolySlerp, A Fast and Accurate Polynomial Approximation of Slerp," *Game Developer*, February 2004.

Eberly, David. "Quaternion Algebra and Calculus." **www.geometrictools.com/Documentation/Quaternions.pdf**

Grassia, F. Sebastian. "Practical parameterization of rotations using the exponential map," *Journal of Graphics Tools*, vol. 3.3, 1998.

Hast, Anders, et al. "Shading by Spherical Linear Interpolation using De Moivre's Formula," Conference Proceedings, WSCG, 2003.

Intel, April 2006. "IA-32 Intel Architecture Optimization Reference Manual." **http://download.intel.com/design/Pentium4/manuals/24896613.pdf**

Li, Xin. "iSlerp, an Incremental Approach of Slerp," ACM Journal of Graphics Tools (submitted March, 2006).

Saff, E.B. and Snider, A.D. *Fundamentals of Complex Analysis*, Prentice-Hall, 1976.

Shoemake, Ken. "Animating Rotation with Quaternion Curves," *Computer Graphics*, vol. 19, no. 3, 1985.

Walsh, Genevieve. "Great Circle Links in the Three-Sphere." **www.ma.utexas.edu/users/gwalsh/dissertationfinal.pdf**

POSTMORTEM

REESTABLISHING

The Peaks and Pitfalls of Tomb Raider: Legend

WHEN WE AT CRYSTAL DYNAMICS FIRST HEARD THAT WE were getting to make the next TOMB RAIDER (the seventh) our emotions were mixed. Yes, we had great memories of playing through the first two; yes, it was and is one of the biggest licenses in video games, but boy had it been cut down by its last four sub-par outings.

One day a passerby who knew about Crystal Dynamics asked, "What are you guys working on?"

"TOMB RAIDER," I responded. His reply: "You guys are still making those?" This was one of the more positive reactions we encountered. Many other former fans that we spoke with were angry. They literally hated the franchise. We found that like most of us, they had been increasingly let down after the first two TOMB RAIDERS and really wanted to relive those first inspiring experiences. This desire defined our goal to recreate the original TOMB RAIDER experience with current play mechanics.

WHAT WENT RIGHT

1 **GOOD EARLY AND MAINTAINED GOALS.** Two components we felt to be critical to recreating the glory of TOMB RAIDER's past were "fluid movement"—our name for pick-up and play controls, graceful animation and three steps ahead camera view—and integration of physics into actual gameplay rather than just as an aesthetic flourish of destruction. The animation systems for fluid movement were a tremendous undertaking. The complexity they presented and the number of coders we had working on them simultaneously made them an exceptional challenge, but we had defined fluid movement as the single most important aspect of the experience so we stayed the course, only very tactically dropping functionality on the road to their completion. Integrating physics into gameplay was equally challenging because of the few existing examples we had to go on. (HALF-LIFE 2 has physics based puzzles, but theirs were designed for a first person camera, a large difference when it comes to physics-based puzzle design.) Some of our early efforts were unconvincing, but we kept at it and physics ended up being one of the most innovative setups in our game.

2 **TRUE PARTNERSHIP WITH MARKETING.** Yes, I said a partnership with marketing. Those of you who see marketing as pure evil, hear me out! Marketing has something we as game developers don't: the experience of working with all the people who must buy into our game concept for it to be a success. The most important of these people are those you hope will buy your game, as well as the folks who work at game magazines and web sites with their covers, previews and reviews, and lastly those who work in the retail channel where shelf space, posters, and standees are placed. If any of those groups don't get on board, starting with your own publisher's marketing group, your five-page preview might go down to one page or none, and that poster and other presentation media you were hoping would be put up at Electronics Boutique might not be. Sure each game has different expectations for what presence it might achieve, but for every person involved that doesn't buy in, your game's publicity is diminished. Nothing is worse than your potential audience not even knowing your game is out there. Given the amount of apathy and even animosity from TOMB RAIDER fans, game magazine editors, and retailers that has built up over the course of the previous four TOMB

RILEY COOPER *has spent his entire career at Crystal Dynamics, where he was the lead designer on TOMB RAIDER: LEGEND, as well as SOUL REAVER II and LEGACY OF KAIN: DEFIANCE. Before that he was a designer on SOUL REAVER, BLOOD OMEN: LEGACY OF KAIN, and SOLAR ECLIPSE. Email him at rcooper@gdmag.com.*

AN ICON

GAME DATA

DEVELOPER
Crystal Dynamics

PUBLISHER
Eidos

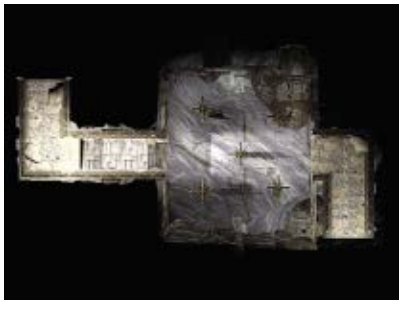
RELEASE DATES
APRIL 7, 2006 (E.U.)
APRIL 11, 2006 (U.S.)

PLATFORMS
PlayStation 2
Xbox
Xbox 360
PSP
PC
GameCube

DEVELOPMENT TIME
2 years

TEAM SIZE
Approximately 80

RAIDERS, winning over hearts and minds was critical for LEGEND to have even a chance, and this would not have occurred without the steady and constant dialogue that occurred between Eidos' public relations, brand management, and marketing teams, and the development team at Crystal Dynamics.



A section of TOMB RAIDER: LEGEND's Bolivia stage, from level layout to final assets.

3 **FOCUS TESTING.** Crystal undertook intense focus testing in closing out TOMB RAIDER: LEGEND. This isn't like movie testing where the execs find out if the audience likes the ending—we'd have had no time to change anything as large as that even if they didn't. No, our focus testing is primarily the painful process of watching where people get stuck. There's a fine line between removing the potholes in layout where this occurs and removing challenge, but the difference between these two is evident when you see it happen first hand. A player will try the same incorrect tactic over and over again, and there's usually more than one player doing this exact same thing—and not one of them is smiling.

There are two factors that affect how much fun people have playing your game: the frequency with which they have exciting memorable moments and the infrequency with which they are genuinely stuck and have to decide whether to keep trying (read: whether to throw the controller at the television).

These two types of events create the highest highs and lowest lows of your game's experience. Focus testing can't really help you create memorable events, but it did allow us to iron out roughly 90 percent of the places where players tended to get stuck. Because of this intense focus testing, LEGEND flows well for the vast majority of its players.

4 **COMMITMENT TO STORY.** By now the importance of quality storytelling in games seems to be accepted, particularly in the action adventure genre, but how often does its importance translate into a day-to-day commitment to its execution? On TOMB RAIDER: LEGEND, we had two and a half people dedicated to story throughout the course of the game, one and a half on-site, one off, and they were involved in all major decisions where story would be impacted. While gameplay is and always will be king, a good story provides motivation and thus it is foolish to separate the two. Our highest goals were to tell the story in-game wherever we could, and to not allow Lara to do things in cinematics that players would rather control themselves. Story can always be (and in our case, was) improved over the course

of development, so long as the changes that come with them don't hinder asset creation or subvert established design. Through these rules and resources and the talent of the individuals involved, we were able to create a story we genuinely liked that was integrated with gameplay.

5 **EXPERIENCED AND VOCAL TEAM.** By now, most of us know that it takes a team to make the games we do. No genius can pull a great game out of a hat. It takes a group of people who are passionate enough about the quality of the entire experience to speak up when a decision made for one part of the game is leading to an unhappy compromise in another. Many of the people on TOMB RAIDER: LEGEND had worked on previous action adventures at Crystal in the SOUL REAVER / LEGACY OF KAIN series of games. The knowledge gained from working on these titles was a boon to the project. No, Lara doesn't suck blood or souls, but just like Kain and Raziel, the main characters of those games, she needs a third-person camera, her world needs visual language, and she needs her story told. We also had talented new team members who injected fresh and invaluable perspectives into the mix.

WHAT WENT WRONG

1 **DEVELOPING CONTENT AND ENGINE IN PARALLEL.** Developing the content and the engine simultaneously is an issue that most game developers deal with to a varying extent with every single game they make. On TOMB RAIDER: LEGEND it presented one of the largest challenges we faced.

Our aggressive schedule required us to start and complete our level designs by specific dates, and our lofty technology goals were more or less mandated by our ambition of reviving one of the largest franchises in video games. This meant we had to design levels and put art in them before we could really even play them. We prototyped a lot of mechanics to approximate the finished mechanics at a very low level, but nothing, nothing, nothing beats hands on playing to find out if something is actually fun or not.

The team showed great resilience in making changes to otherwise final layouts as we learned more about the game we

Unreal® Technology News

by Mark Rein, Epic Games, Inc.

Canadian-born Mark Rein is Vice President of Epic Games based in Raleigh, North Carolina. Their Unreal series of games is reported to have sold over 7 million copies world-wide. Epic's Unreal Engine 3 has won Game Developer Magazine's Frontline Award for Best Game Engine for the past two years. Since 1992 Mark has worked on Epic's licensing & publishing deals, business development, public relations, academic relations, marketing and business operations Currently in development at Epic: Gears of War for Microsoft and Unreal Tournament 2007 for Midway.

Upcoming Epic Attended Events:

Microsoft's Gamefest
Seattle, WA
August 14-15

Games Convention
Leipzig, Germany
August 23-26

Tokyo Game Show
Makuhari Messe,
September 22-24

GDC London & London Games Summit
London, UK October 3-5

Serious Games Summit
Washington, DC
October 30-31

Please email: mrein@epicgames.com for appointments.

POWERED BY UNREAL TECHNOLOGY

Welcome! Some of the things you might see in this space include profiles of our licensees' games, news about events we've attended or plan to attend, details of new engine features and obviously some blatant attempts to sell Unreal Engine 3 (UE3) licenses. I also hope to have other Epic people chime in from time to time and I encourage you e-mail your feedback or suggestions.

E3 2006 IS NOW BEHIND US

Epic Games is both a game developer and an engine licensor. This years' E3 was a huge success for us in both capacities but I think it is the latter that excites me the most. Gears of War picked up several major E3 awards including Best Console Game from the Game Critics Awards: Best of E3 2006. Four UE3 games; Bioshock (Take2), Gears of War (us), Huxley (Webzen), and Mass Effect (Bioware/Microsoft) combined to pick up an amazing 12 nominations in these, the most prestigious of E3 awards. What is so amazing about this number? Nintendo was the only company to have more nominations than this!

Monster Madness from Artificial Studios

Some of the other award winning and nominated Unreal Engine titles from E3 include; Brothers In Arms: Hells Highway (Gearbox/Ubisoft), Frontlines: Fuel of War (THQ), John Woo's Stranglehold (Midway), Rainbow Six Vegas (Ubisoft), Splinter Cell Double Agent (Ubisoft), Too Human (Silicon Knights/Microsoft), and our own Unreal Tournament 2007. Congratulations to these Unreal Engine licensees and developers!

In addition to showing off our games during the day we also hosted a hotel suite nearby where we were able to give demos and answer technical questions for existing and prospective licensees.

MONSTERS IN THE HOUSE

If you liked games like Zombies Ate my Neighbors and Smash TV and if you enjoy facing down hordes of goblins, ghouls, zombies, and demons, while having a laugh doing it, then you're going to love Monster Madness a new game from rising star American developer Artificial Studios.

GameSpot calls Monster Madness *"an attractive alternative to the traditional first-person shooter."* What makes Monster Madness special for me is that this isn't the kind of game you'd expect to see made with UE3. What you notice immediately is its isometric view co-op mode where up to four players control characters on the same screen to slash, shoot, and drive through a menagerie of monsters. The less-common viewpoint combined with a humourous pop-culture horror setting and unique art-style really helps this game stand out from the crowd.

Artificial Studios President and lead developer, Jeremy Stieglitz says UE3 has *"allowed our artists & programmers to work much more efficiently together. All of our AI behavior is now animation driven... this allows our artists to tweak or reconfigure animations without messing up game behavior... using [Unreal's] Cascade visual effect systems in combination with character 'sockets' gives artists the ability to bind visual effects to all sorts of components without the need for any programming. The end result is that programmers can focus solely on writing code for behavior, while letting the artists themselves handle the complete setup of their visual assets in-game."*

Jeremy also praises their decision to use UE3; *"The bottom line is that for Monster Madness we have a short development cycle and I would never have considered it possible if we weren't using Unreal Engine 3. But with this technology, in particular the many ways in which Unreal Engine 3's tools and pipelines ensure increased productivity, we're ahead of schedule and I'm entirely confident we're going to beat our deadlines. It's really been a pleasure from the moment we started working with it."*

For more information about this very fun game please visit: www.monster-madness.com.

See you back here next month!

EPIC GAMES

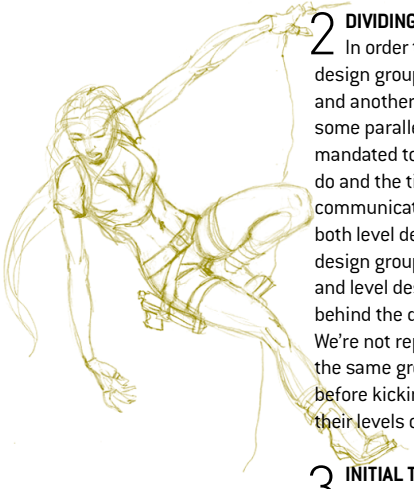
For UE3 licensing inquiries email:
licensing@epicgames.com

For Epic job information visit:
www.epicgames.com/epic_jobs.html

w w w . e p i c g a m e s . c o m

actually had, but it's a large drain for anyone to have to revisit work that was previously meant to be final. Unfortunately in a rapidly evolving industry, this challenge is almost impossible to avoid. If you're making an ambitious game and you don't have more than the standard one to two years, you're going to be stuck juggling what is with what might eventually be.

In the future we intend to better anticipate the degree to which this will impact the project, scheduling for the inevitable changes that we will feel compelled to make, and setting expectations within the team so it doesn't hit like a ton of bricks.



2 **DIVIDING DESIGNERS BETWEEN SYSTEM AND LEVEL CREATION.** In order to hasten development, we decided to break our design group into two camps, one focusing on game systems and another focusing on level creation. While this did create some parallelism in fleshing out the complete design, and was mandated to a certain degree by the amount of work we had to do and the time we had to do it in, it created a responsibility and communication gap within the design department that made both level design and system design more difficult. The system design group lacked the grounding that level creation brings, and level design lacked the intimate knowledge and reasoning behind the different game systems that they had to work within. We're not repeating this mistake. The extra time it takes to have the same group of designers work through systems design before kicking off levels is saved in the efficiency with which their levels can be created and maintained.

3 **INITIAL TEAM SIZE.** When development of the game began, Crystal was understandably anxious to get the TOMB RAIDER ball rolling, and there were talented people who were not assigned to other projects. This combination led to an initial team size that was simply too big. From a distance this might seem to be a good problem, but particularly for such a high profile title where there is a lot of passion for what it could become, it is exceedingly difficult to get 45 or so people to all march in the same direction when large details about the title, such as story locations and core play mechanics, have yet to be hammered out. This led to a lot of wasted energy and stress, by many if not all of those involved. We've seen the error of our ways in this regard and are being much more careful with how many bodies are initially assigned to projects.

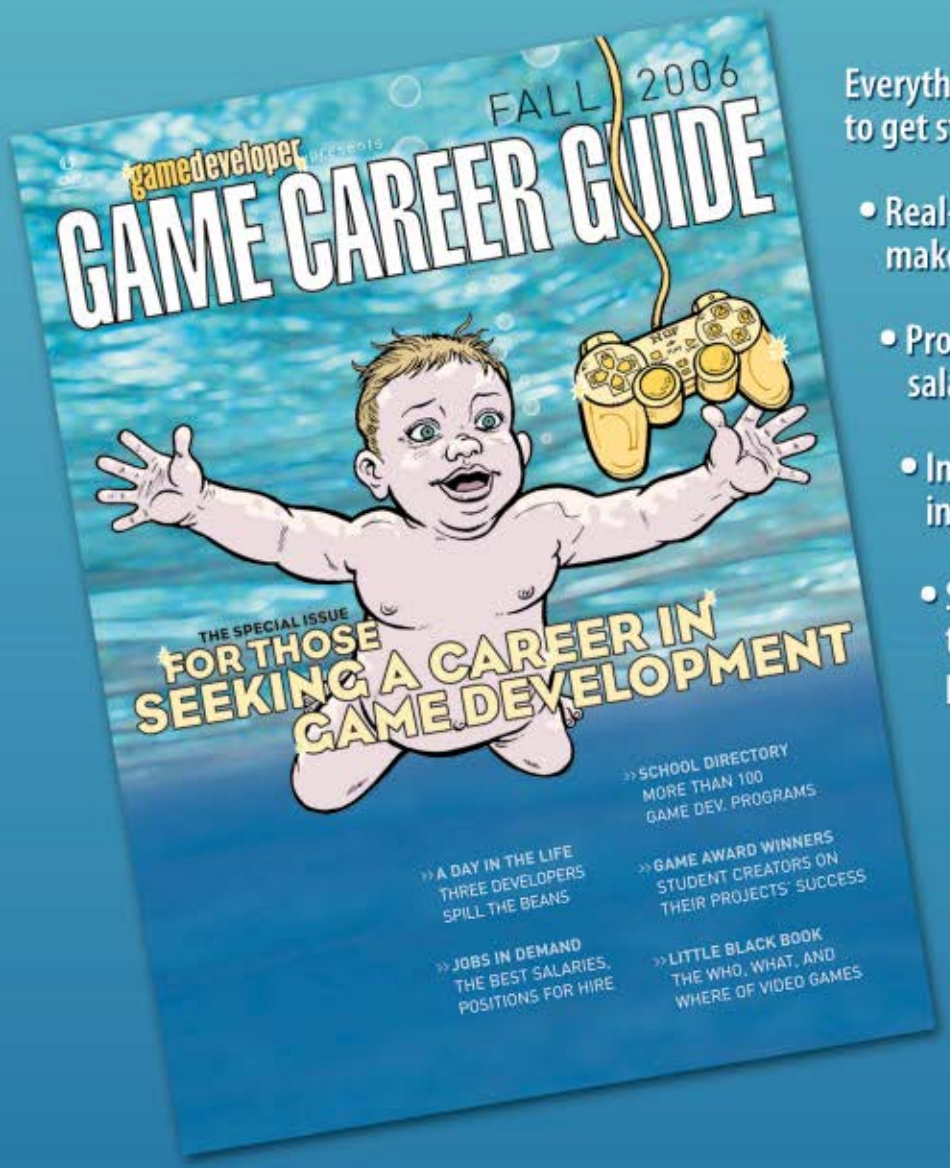
4 **SCHEDULE EXTENSIONS.** As some of you may know, Eidos was sold more or less halfway through the development of TOMB RAIDER: LEGEND. The impact of this on the team was negligible. In fact we were given more time to complete the project, a testament to the support and confidence we received from our new managers at SCI. This extra time was crucial. I don't even want to think about the game we would have shipped without it. That said, time in game development is everything, and making the game to one set of expectations and then revamping goals to fit a new set of expectations creates a sizable amount of stress as you come up to the short deadline, and rework as you make adjustments for the long one. Hindsight being 20/20, we feel we should have been more vocal about the problems we saw with the shorter schedule, as the situation might have been completely avoided.

5 **LATE AND UNDER-RESOURCED FEATURES.** There were a few features that we started too late or waffled on for too long. Invariably these were the features which we feel could have been better. In retrospect, it's hard to say whether we should have cut those features or could have supported them more than we did. This clearly serves to remind us that if you don't want any features in your game to under-perform, you need to do them 100 percent or not at all.

THE LEGEND CONTINUES

TOMB RAIDER: LEGEND provided the most challenging game development cycle many of us have been a part of. We had times when we didn't know if we were going to be able to pull it off, and like any development cycle there were as many lessons learned as triumphs. And, like everyone in this industry, we're still learning our craft while the canvas grows bigger and our paintbrushes are refined. That we were able to make a game that we are all proud of and that has been well received by critics and both those who loved TOMB RAIDER already and those wanted to love it again, is all the reward we could ask for.

Don't play games with your career.



Everything you need to know to get started in the game industry:

- Real-world advice on how to make it in the business
- Professional resources and salary survey
- Insider information from industry veterans
- School directory with more than 100 game development programs

Purchase a digital edition online at

www.gamecareerguide.com

THE 2006
gamedeveloper
GAME CAREER GUIDE

Be sure to check-out these events - **all part of the London Games Festival, October 2-7**

LONDON GAMES SUMMIT

PRESENTED BY eLSPA Tiga

4-5 October, 2006

195 Piccadilly, London

BAFTA Headquarters

A leadership seminar addressing next-generation business issues and exploring strategies for maximizing market opportunities. The definitive event for game industry executives and business managers in Europe.

www.londongamessummit.co.uk

GameDevelopers Conference London

2-4 October, 2006

195 Piccadilly, London

BAFTA Headquarters

Focusing on advanced techniques in game development and key next-generation console and PC development issues, with a special emphasis on the UK.

www.gdclondon.com

LONDON GAME CAREER FAIR

3-4 October, 2006

68 Regent Street, London

Cafe Royal

An opportunity for industry professionals and recent graduates to discuss career opportunities with leading gaming companies.

www.gamecareerfair.com

There is limited seating and both of these events are guaranteed to sell-out. To ensure your place, REGISTER NOW!

london games festival 2006

PARAPPA'S PAPA

INDUSTRY INSIGHT FROM MASAYA MATSUURA

brandon sheffield

NANAON-SHA'S MASAYA MATSUURA, CO-CREATOR OF PARAPPA THE RAPPER, is a unique individual with a small but prolific company. He gained recognition through several years in the music business, then entered the game industry in order to more fully express himself creatively. While his games vary in quality and scope, they all bear his unique signature. Matsuura is particularly notable for helping form music and rhythm games as genres and being one of the 32-bit era's first game auteurs. *Game Developer* spoke with Matsuura to unearth his influences, his aspirations, and his thoughts about the industry, from the unusual perspective of a non-game-playing game creator.

Brandon Sheffield: *What made you decide to make a rhythm game from the onset?*

Masaya Matsuura: I started my career as a musician in the early 1980s. I had a band at the time and released 10 albums through Sony Music Entertainment. It was kind of electronic pop music, and I did everything from composition to production, with a female singer doing the vocal track.

The reason I started to make games is that I was sometimes frustrated with the style of music I was making. Computer manipulated music makes very virtual images in the mind of the audience. A lot of listeners don't care if a rhythm track is played by a human or a computer.

On top of that, the record company always wants to make more mass-market appealing music. It was getting frustrating to continue along those lines. As a result, I became interested in computer manipulated music, instead of computer programmed music. I wanted to control the music dynamically and directly from the computer program, instead of just using [the computer] as a ready-made application for composition. That was the start.

BS: *Do you find that the game industry is more receptive to your ideas?*

MM: Yeah, so far the game industry is much more friendly and open to those who aim to create something new.

BS: *Do you feel like you can really communicate your vision in games as much as you'd like?*

MM: Difficult question. Recently, as you know, the game industry has grown very conservative. So it's very hard to find a way to keep the production fresh and creative. I'm always fighting against this sort of situation.

BS: *How important is audience reaction for you?*

MM: Also a difficult question. At [a lecture I saw at GDC], everyone talked about the niche market. To a lot of people, niche market just means a minority market. But I don't think that way. The game industry is too focused on the hardcore gamers. Fortunately, we have various types of people who aren't interested in games—like my wife. The niche market right now is considered those who are already interested in the games market—those very close to games and who have been playing for a long time. But my image is that we have to appeal to people outside the traditional game freak. It's a pretty big issue for us as an industry.

BS: *It seems like in movies, the niche markets are there to cater to a very wide audience in specific ways, not just the hardcore movie buff.*

MM: It's kind of hard to compare games and movies. My friend who's a big fan of movies told me that in the early 1990s, movies got really boring. Everything was a blockbuster with Arnold Schwarzenegger or Sylvester Stallone. The game industry is very close to that kind of situation now.

BS: *What do you want people to come away feeling after they play your games, maybe even aside from just having fun?*

MM: I think that making the audience have fun is enough, really. And maybe if they have fun, they'll better remember what was happening in the game, whether it's sound, or characters, or story. And that makes some sort of statement.

CONTINUED ON PG 32

BRANDON SHEFFIELD *is the features editor at Game Developer, and his favorite Nanaon-Sha game is* VIB-RIBBON. *Vector art forever! Email him at* ***bsheffield@gdmag.com****.*

PHOTO BY BRANDON SHEFFIELD

PARAPPA THE RAPPER was Nanaon-Sha's first big hit.

CONTINUED FROM PG 31

BS: *How did your experience in music help you when you started to design games?*

MM: To make better [music] tracks, we have to learn a variety of things outside of music, like marketing, advancements in and uses of digital technology—these kinds of outside experiences. They were very important for me at the time. The knowledge of the technical workings of CDs especially helped me to think about working on games.

BS: *When VIB-RIBBON came out [a rhythm action game in which the game data would load into RAM, then your own CDs would become the soundtrack and basis for the action], that was one of the first games that allowed you to use your own music. What made you decide to do that?*

MM: After PARAPPA, I found out that a lot of music fans hate hip hop. So a lot of my friends, especially from the music industry, asked me, "Why do you make hip hop games? You didn't play hip hop when you made music!"

It was a bit weird for me, but I wanted to answer their requests. Of course I did UMJAMMER LAMMY after that, and it was a guitar-based game. But at the same time I don't want to make something like GUITAR HERO COUNTRY, GUITAR HERO HEAVY METAL, things like that. I wanted to make a music-based game that would be open to all kinds of music fans.

BS: *How large is your staff? It feels like many of your games take a unique low-fi art style, which might keep staff low.*

MM: Every time I make a game with a unique art style, it's assisted by very talented graphics programmers. But they're not direct employees of my company. So I actually just contract out that sort of work. Right now, it's just 10 people, including contractors. It's a very small team.

BS: *How long does it take you to make a game on average?*

MM: It's hard to say. A few years ago it was a much different story. But recently we spent three months in development. Of course, we spent a few months in planning, and another few months in production. Six to nine months, usually [for the whole process]. Anyway, less than a year.

BS: *Do you feel it's more rewarding being in games than in music now? Did you find what you were looking for?*

MM: I think so, yes. But still I'm watching the music industry. The music industry is a good model for the game industry's future. [The markets are] very similar. Of course the game industry is also similar to the movie industry, but the big difference between them is that the movie industry is kind of based on the subscription model—you need repeat customers. The music industry is based on a more direct selling model. The game industry has both. Game software sells in physical packages, so this kind of retail culture is very close to the music industry I think.

BS: *Would you ever consider moving back into just music?*

MM: No, just because to make good music requires a very good influence from someone else. Of course good musicians and good songs make me think about better music sometimes. But sometimes good music actually makes me feel like making a game. Still, sometimes I want to do jam sessions with great musicians that I already know.

BS: *Do you still create music on your own just for yourself?*

MM: Oh yeah. But if I had a chance to do jam sessions with other people I'd actually want to make a game for them.

BS: *Do you ever think there could be such a thing as a game designer jam session? Would you ever want to do something like that?*

MM: That'd be great.

VIB-RIBBON **allowed players to use their own CDs in-game.**

BS: *It seems like everyone in Japan is still a bit too secretive about their own techniques. Hopefully that will change soon.*

MM: I think that kind of thing is already happening. The game industry in Japan is in a tough spot right now. Companies can't keep their talent in-house, because of a lack of ... something. I don't know what it is. But these kinds of people who have been leaving their companies recently are making new trends for the industry. ⁘

MICK WEST

A SHATTERED REALITY

Why presenting 'realism' is unrealistic

PROGRAMMERS AND GAME DESIGNERS sometimes start out a project with the noble intention of making the most realistic game possible. But as they progress, they discover that their initial dreams of realism are somewhat difficult to implement in a practical manner.

First, in the area of player control, the most realistic physics is often not the most enjoyable physics. Second, in the area of non-interactive effects, such as explosions or smoke, the programmer quickly finds that trying to accurately simulate the underlying physics is computationally not feasible.

This article discusses the problems of simulating reality, with particular reference to shattering glass.

INVESTIGATING REALITY

Shattering glass is used in many games. The most obvious example is any game—but particularly first-person shooters—that allows the player to shoot at a window, either leaving a bullet hole in the glass or shattering the whole pane of glass.

But shattering glass is also used in other types of games that aim for increased realism. In auto racing games, the cars' windows and headlights shatter in a crash. In basketball, the backboards sometimes shatter. Even in wrestling, a realistic-looking fluorescent glass tube being smashed over your opponent's head can add to the feeling of immersion.

Suppose you, the programmer, have been tasked with making the glass shattering effect, and your producer has told you to make it "as real as possible." How should you proceed? Perhaps the first thing to do is read some physics books, search the internet, and try to find some physical models, some equations, that describe how glass shatters in the real world.

The first problem that you'll run across for a lot of these effects is that nobody actually knows how they work in the real world.

Take another common effect, fire. Nobody knows how the underlying physics of fire really works. Something as simple as a candle burning is a complex interplay of molecules, gravity, chemical reactions, and the heating, motion, and radiation of multiple gasses, liquids, solids, and plasma.

Since physics has yet to explain exactly what's going on when something burns, rendering an accurate image of the candle is an inexact task, involving light emitted from the burning gasses, reflected off the wick and the wax (both liquid and solid), transmitted and scattered through the solid wax, refracted through the pooled liquid wax, refracted through waves of rising hot air and vaporized paraffin, absorbed and reflected off the smoke particles and interacting with the rest of the environment. And that's just for one candle. Imagine a whole cathedral full of them!

Similarly, scientists simply do not know how glass shatters. There are competing models of what happens when a piece of glass breaks in two. The debate is whether the fractures happen via the breaking of subatomic bonds, one after the other in the direction of the fracture, or if the fracture follows the formation of microscopic cavities that form ahead of the fracture tip. (See Mills in References.)

Unfortunately, these distinctions, while interesting, are entirely academic to the game programmer. If you've gone as far as discovering this in your research, then you've probably gone too far. Some things can never be simulated.

LIMITS TO COMPUTATION

The real world operates at a much finer grained level than is possible to simulate on a computer. The "real world" operates at molecular, atomic, and subatomic levels. The so-called "rigid bodies" that modern physics engines simulate are in reality composed of septillions (1 septillion = 1 billion billion billion billion) of molecules; it is the interactions between these molecules that create the apparent motion of the rigid body. In addition, the time-step (or "main loop") of the real world operates in an essentially infinitely small step of time, compared with the 1/60th of a second that many game physics systems run at.

In last month's article, I described how to simulate a blob using a mass-spring system. In the real world, solid matter actually works a little like my blob, except that there are vastly more masses (molecules), an additional order of magnitude more springs (inter-atomic and inter-molecular forces), and a significant number of the springs keep breaking and re-forming.

So, if we want to simulate something as straightforward as a bullet going through a glass window, with the aim of it looking realistic, then what we can't do is simulate the interactions of the 10^{28} silicon molecules, with quintillions of simultaneous micro fractures resolving into the macro fracture pattern we want.

Glass contains a lot of molecules. The most accurate simulation would simulate the state of each molecule, and the interactions between molecules. Even forgetting that we don't actually know what's happening at the molecular level, the sheer number of molecules in matter is unfeasibly huge. In a single gram of a material like glass, there are approximately 10^{25} molecules. Even ignoring the physical limits of computing,

MICK WEST *was a co-founder of Neversoft Entertainment. He's been in the game industry for 17 years and currently works as a technical consultant. Email him at* ***mwest@gdmag.com****.*

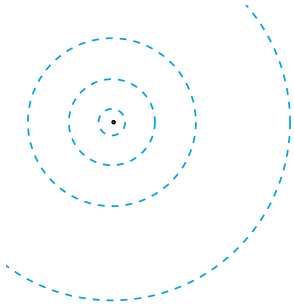


FIGURE 1 **When simulating breaking glass, circles act as guides for the placement of cracks.**

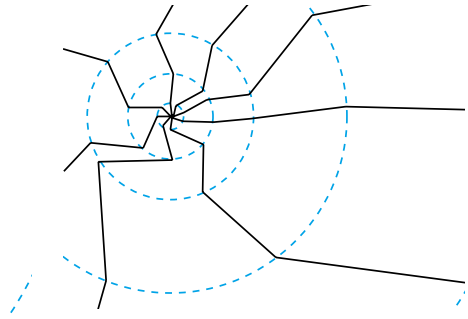


FIGURE 2 **Joining roughly equi-spaced points on the circles lets you create randomly jagged radial cracks.**

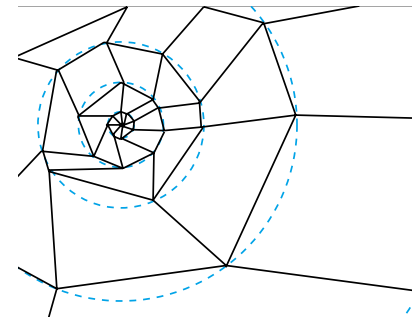


FIGURE 3 **Joining sequential points on the circles makes the radial cracks look like a spider's web.**

Moore's Law still requires about 100 years before we'll have enough computing power to even store the state of the simulation.

Still, much research has been done into precisely that type of simulation, albeit on a greatly reduced scale. It is still possible to simulate what is happening within a material using a molecular model simply by making the molecules a lot bigger so you don't have to use as many of them.

Results seem to be very realistic, but are still rather expensive. In 2002, the ASCII White, 12 teraflop, $110 million super-computer ran simulations of fractures in a small cube of material with one billion molecules, taking nearly two seconds per frame (see Abraham in References). By comparison, the Sony PlayStation 3 has a theoretical performance of two teraflops, and it needs to do a lot more than simulate one crack.

In short, a hyper-realistic simulation is not yet feasible in video games—not even close. We game programmers must move on to looking at models of the interplay of forces within the object that are much cheaper to implement. Realism must begin to take a back seat to another type of realism—the reality of our limited resources.

PRACTICAL SHATTERING

To create the effect of shattering glass in a game, we first need to ask how quick the shattering code actually needs to be. Unfortunately, there are two main performance problems with shattering.

First, shattering happens very fast. Cracks propagate in a material at about the speed of sound in that material. For glass, that's around 5,000 meters per second. If simulating at 60fps, then any shattering of glass is effectively instantaneous, which in turn means that the entire calculation has to happen in a single frame.

Second, the shattering of an object turns what was a single piece of the environment, or a single rigid object, into hundreds of individual rigid objects, each requiring memory and CPU resources. If players are given the freedom to shatter everything in sight, then how do we stop them from exhausting system resources?

Even in older games where the fragments vanished after a period of time, a common trick game testers used to crash the game was to shatter multiple objects at the same time, especially in split-screen mode.

What to do about the fragments generated is a problem with multiple solutions. You can devote a very large amount of your resources to these chunks (assuming it's beneficial to the game somehow). You can make the chunks vanish after a period of time. Or, you can remove old chunks as new ones are generated, perhaps with some priority system, but that's more of a game design issue than a programming one.

The problem of the time used generating these chunks or fragments is another matter. If the shattering takes a long time, it can cause the game to glitch perceptibly. Players will be familiar with many games that slow down when many objects are blown up. This can be due to an excessive amount of graphical effects on screen, but in more recent games can also be caused by increased physics complexity in the arena. Adding accurate shattering to the mix has the potential to greatly increase the amount of slowdown in such situations.

In order for the game not to slow down perceptibly, the shattering code must be able to shatter several objects per frame. Let's say we can share the shattering over a few frames if we happen to have a large number of objects shatter simultaneously. Then a reasonable number would be perhaps 10 objects shattered per frame, which must incur no additional overhead to the frame's processing load. This essentially means that we have to have some processing power kept permanently in reserve for this kind of thing. You'd probably not want to budget more than 10 percent of your processing for such frivolity as shattering objects, so that means each individual shatter must happen in less than 1 percent of a frame, or about 0.16 microseconds.

Still, we can look at this issue another way. Assuming that the shattering effect is going to create a large number of rigid bodies that were not there before, then the system must be able to simulate and render those bodies without dropping speed. All we have to do is to make it so that the shattering code for an object does not take longer than the simulation and rendering code for the rigid bodies that are generated by that shattering. However, this might not be true if the shattering of the object and simulation of resultant rigid bodies use different resources—such as when the shattering happens on the main CPU, and the simulation is on another processor such as the SPU, PPU, or GPU.

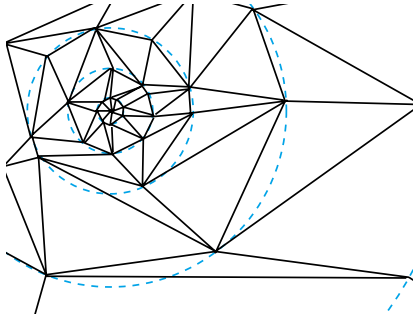


FIGURE 4 **Triangulating the quads, randomly along either diagonal, sets up the possible paths for the cracks.**

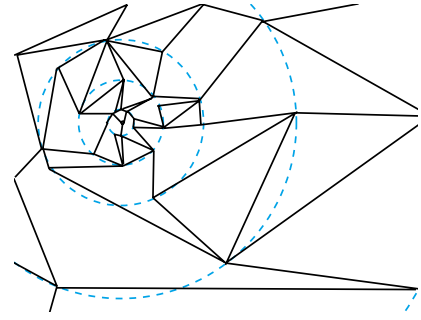


FIGURE 5 **Randomly joining the triangles, this image looks similar to patterns observed in actual shattered glass.**

BEYOND MOLECULES

The next step is to try to model the physical forces acting on an object at an even higher level, using something like a mass-spring system. We can devolve the object into a system of connected points as a regular mesh of triangles (or tetrahedrons), and then model the propagation of forces through this mesh, and allow the object to split along lines or planes when the forces at that junction surpass a certain level.

Even a highly abstract simulation of shattering can be very time consuming. At the Game Developers Conference in 2002, O'Brian and Hodgins (see References) presented a method of 3D shattering using this kind of decomposition into tetrahedrons. Their example of a single small wall took an average of 399 minutes to calculate a simulation of one second of shattering, based on 1999 hardware. Updated by a factor of 20 to 2006 hardware, this still only gives us about 1/1,000th the speed of a real-time simulation.

BEYOND PHYSICS

The problem with simulating physics is that, while we know what result we want to get, the physics model does not always supply this result, and takes a long time to not supply it.

Instead, we can try to shatter glass based on aesthetics. We can observe how glass breaks to the naked eye and then try to duplicate it with simple heuristics.

Kadono and Arakawa used high-speed cameras to study crack formation in a sheet of glass (see References). Others have noted observations—which are consistent—about the way glass breaks when impacted by small objects.

1. Radial cracks propagate from the impact point like spokes on a wheel.
2. Other cracks propagate between radial cracks, like a spider's web.
3. Cracks stop when they hit another crack.
4. The size of different glass fragments is a power function of the distance from the impact point.

These observations suggest a number of simple algorithms we could try. Since we are essentially generating a visual pattern, no physics need be involved, and we can use a number of cheats to get the result we want. In this case, we can essentially generate a spider's web.

1. Imagine a framework of concentric circles, centered on the impact point, with the distance between them increasing exponentially (see Figure 1).
2. Create jagged radial cracks by joining roughly equally spaced points on these circles in sequence from the center to the edge (see Figure 2).
3. Create traverse cracks by joining sequential points on the circles (see Figure 3).
4. Turn the resultant quads into triangles by randomly splitting them along either diagonal (see Figure 4).

Now that's a very simple algorithm and it's very fast. It will even give you a reasonable result for different types of glass and different types of impacts.

There's a lot you can do to extend this algorithm. For one, you can add various types of random perturbation to make it look less regular. You can randomly join adjacent triangles to create jagged irregular pieces, as in Figure 5. You can let the radial cracks bifurcate. You could either arrange the line generation so there isn't a possibility of the cracks crossing (by limiting them to an angular segment), or you could allow your cracks total freedom and have an additional step to detect and resolve crack-crack collision.

CRYSTAL CLEAR

Simulating the underlying physics of something that is essentially a cosmetic effect is often inefficient and does not always give satisfactory results. Simulating at any kind of molecular level is infeasible. Simulating as a system of joins and forces can give reasonable results, and much work is being done in this direction.

However, you can still get a perfectly usable result by simply observing what is going on, describing it, and then simulating your description, without any need to understand the underlying physics. Since the model is based purely on visual results, it has the potential to look more aesthetically pleasing than a physics-based solution.

REFERENCES

Abraham, F. "Simulating materials failure by using up to one billion atoms and the world's fastest computer: Work-hardening," Proceedings of the National Academy of Sciences of the United States of America. (April 30, 2002. Vol. 99, No. 9), 5,783–5,787.
www.pnas.org/cgi/content/full/99/9/5783

Kadono, T., and Arakawa, M. "Crack propagation in thin glass plates caused by high velocity impact," *APS Physics Review* (March 2002).

Mills, W. "What Makes Glass Break?" *Research Penn State Magazine* (October 31, 2005).
www.rps.psu.edu/probing/glass.html

O'Brien, J. F., and Hodgins, J. K. (1999). "Graphical Modeling and Animation of Brittle Fracture," Computer Graphics Proceedings, Annual Conference Series. Los Angeles, California: ACM, 1999, 137–146.
www.cs.berkeley.edu/b-cam/Papers/obrien-1999-GMA.pdf

TOKYO GAME SHOW 2006

New Excitement. New Sensations. A New Generation.

tgs.cesa.or.jp/english

TOKYO GAME SHOW 2006

9/22 [FRI] Business Day
23 [SAT] 24 [SUN] Open to the Public

Venue : MAKUHARI MESSE tgs-info@publicity-bur.co.jp

Organizer : Computer Entertainment Supplier's Association(CESA)
Co-organizer : Nikkei Business Publications, Inc.(Nikkei BP)

CESA

Nikkei BP
Nikkei Business Publications, Inc.

REGISTER TODAY AND SAVE $125!

Where Entertainment Meets Mobile

Make the Right Connections at this year's Mobile Entertainment, Content, Commerce and Applications Conference!

SEPTEMBER 11, 2006

THE WESTIN BONAVENTURE HOTEL
LOS ANGELES

THE OFFICIAL ENTERTAINMENT CONFERENCE FOR CTIA

The Wireless Association™

Billboard CTIA mecca

MECCA FALL 2006 will shed new light on the evolving mobile entertainment industry. Network with over 800 key influencers and gain development, marketing and consumer insights from leading analysts and end users from all corners of the converging mobile and entertainment fields.

Learn the latest trends, innovations and insights through MECCA's compelling programming, guided by Billboard's expert editorial team and an advisory board panel representing carriers, content providers, developers and more!

NEW! MECCA FOR GAMES **What's Next for Mobile Games?** All MECCA attendees get access to this definitive mobile gaming track, consisting of three additional sessions. Get connected with industry professionals as they present their knowledge of the exploding mobile gaming sector.

Powered by GameDevelopers Conference

Register Today!

$375 Earlybird Rate REGISTER BY AUGUST 3

For more info including sponsorship opportunities and how to register visit: **www.BillboardEvents.com**

TITLE SPONSOR: sms.ac always connected

iRADIO CAR DISPLAY

PRESS ROOM SPONSOR: tira wireless

BASH SPONSOR: AOL

MEDIA SPONSORS: mma mobile marketing association; Wireless Week; mpeg Industry Forum; MobileWirelessJobs content driven.

CORPORATE SPONSORS: JULY SYSTEMS; m:metrics; Mobile 365; mBlox; WapOneline; bango; mobiTV; CORE MEDIA; GREYSTRIPE; nellymoser

CTIA WIRELESS I.T. & Entertainment 2006
A Department of CTIA–The Wireless Association™

One Show. Two Personalities. The largest wireless data event in the industry that truly embodies the ever-changing, dynamic and innovative world of wireless data in Enterprise and Entertainment.

September 12-14, 2006 | Los Angeles Convention Center

MARK LITVACK

DEALING WITH PIRACY

DEVELOPERS OFTEN SPEND SIGNIFICANT dollars and countless hours of ingenuity and sweat equity bringing a game to market. Embarking on an anti-piracy program to protect that ingenuity may be an important weapon when seeking to protect intellectual property, but such a program may also be ill-advised. In this article, I attempt to answer a number of common questions about piracy and intellectual property as they relate to the world of game development.

PIRACY PREVENTION

There's one thing that anti-piracy is not going to do, namely, stop piracy. Piracy is an economic fact of the marketplace. In some places it is rampant and widespread; in some areas it is basically irrelevant.

A successful anti-piracy program from the standpoint of a developer is not one that simply punishes people for pirating. Developers should be honest and realize that there are times when piracy simply does not hurt the owner of the intellectual property. For example, assume there is no legitimate distribution (and no plans to develop legitimate distribution) for your intellectual property in a country. Then it makes no sense to spend limited time and assets in that country fighting piracy.

A successful anti-piracy effort for the owner of the property is one that enables that owner to make more sales and generate more revenue. Thus, often I advise clients to actually spend more money fighting piracy in the markets where they are making money than in those markets where they are losing it.

Often the best goal of an anti-piracy program is simply to protect the legitimate distribution one already has. Thus, a crucial obtainable goal for many anti-piracy programs is "keep honest people honest."

Remember that your goal is to sell your product to as many people as possible. To do so, you must make the product user-friendly and give the consumer a positive buying experience. Anti-piracy solutions often get in the way of this. For example, if every time consumers use a piece of software they must go through an extensive verification process, although it may make the software far more difficult to pirate, it will most certainly lead to a less enjoyable experience.

PIRACY ACROSS THE SEVEN SEAS

Certain countries do a far better job of protecting intellectual property than others. For example, traditionally developed economies, such as Canada, England, and Japan for the most part look at intellectual property in a way that's similar to how we view it in the U.S. As creators of their own intellectual property, they tend to value and protect it. Also, given the health and strength of their economies, their populations tend to be able to afford the price of legitimate products. Anti-piracy efforts in these countries tend to deliver greater value for the dollar in that developers are protecting the legitimate market that they may already possess. Thus stopping a pirate sale may actually lead to it being replaced with a legitimate one.

Russia, much of Eastern Europe, and China are examples of countries with developing markets whose anti-piracy efforts tend to be far more problematic. First of all, the legal system is often not as developed and has additional strains on it, and police are often far more concerned with maintaining peace on the street rather than protecting anyone's intellectual property. Second, the government may not be anxious to assist you in your anti-piracy efforts. Piracy tends to deliver products to end-users at a cheaper cost, the governments of these countries may be quite willing to allow the practice to continue.

These markets also tend to export pirated products. Thus one can find many pirated movies, software, and games stemming from Southeast Asia and Russia, which generate revenue for those nations. These pirated products are shipped to developed markets. To the extent that these areas are exporting pirated products they become a far greater problem.

However, as the markets are developing, they provide fertile ground for future legitimate distribution. One may want to spend anti-piracy resources in these countries so that the markets may develop. Unfortunately, it tends to be a "one step forward, two steps back" process, where one spends tremendous time and resources, feeling the entire way is an uphill climb.

There is another batch of countries in which at this point it's simply futile to attempt any anti-piracy efforts. Countries such as Iran, Afghanistan, and Sudan fit into this category. Regardless of what is going on in these countries, as a matter of economic and geo-political reality, it simply is a waste of time and money to address the piracy issue there.

SCURVY DOGS

Each developer will have to strike the balance between anti-piracy protection and customer service that it is comfortable with. However, one should never underestimate the importance of giving customers what they want. Over-emphasizing anti-piracy and creating a hassle for one's legitimate customers to use a product is almost certainly a mistake. Ideally, one's anti-piracy efforts should be invisible to the legitimate consumer.

MARK LITVACK *is a partner in the litigation and intellectual property departments of Mitchell Silberberg & Knupp. Prior to joining the firm, he was vice president and worldwide legal director for the Motion Picture Association of America's anti-piracy department. Contact him at mlitvack@gdmag.com.*

NOAH FALSTEIN

FAIR PLAY

THERE ARE SOME GENERAL RULES OF game design that encompass entire families of more specific rules. One such family concerns fairness. Recently, I received several emails from readers suggesting different rules that fit into this group. It's a rich and interesting area of discussion, with lots of intriguing questions and few clear answers.

But first, a note about the issue of fairness itself. The general rule I refer to is "Make the game appear to be fair to the player." There are two key phrases there: "appear to be fair" and "fair to the player." The first phrase indicates that this rule is a psychological one, where the illusion or appearance of fairness trumps an absolute measure of fairness. "Fair to the player" points out the asymmetrical nature of the rule. A game doesn't need to be fair to a non-player character. In fact, few games are, or we wouldn't have those hordes of minions to defeat, and bosses would learn to cover up that one critical point of vulnerability.

MEANINGFULNESS

Several rules or partial rules have been suggested that describe more precisely how a game should be made to appear fair. Amir Ebrahimi, Neil Druckman, and Elan Ruskin of Naughty Dog suggest, "Provide meaningful choices, but not too meaningful," elaborating that a player should have meaningful and interesting choices to make in a game, but not so meaningful that they impact the rest of the game with little or no chance of reversal. This rule ultimately concerns fairness and asks designers to define exactly how to keep a player from getting stuck—unfairly.

Q Entertainment's NINETY-NINE NIGHTS doesn't give most enemies a fighting chance—is that fair?

Coming at a similar problem of how players recover from failure from a different angle, Bob Spink of Pivotal Games suggests, "Blame yourself, not the game." Spink's description: Players should attribute their failure to their own poor performance, rather than blame the game. They should be armed to the teeth with knowledge regarding not only how not to fail, but also how to succeed and, most crucially, why their last attempt did not succeed.

ALL'S FAIR IN LOVE AND WAR, BUT NOT GAMES

I have yet to enter these two new rules and some others into the official 400 Project because they touch upon some really interesting yet unanswered questions. Consider for example these questions about fairness in games:

Is it acceptable to let players get stuck in a dead end because of choices they have made if they have no way of knowing the consequences before making those choices?

What if the consequence is merely the loss of 10 seconds of play time?

What if providing more information about the consequences damages dramatic tension, and makes the story or emotional payoff less enjoyable?

Is it fair for the game to "cheat" if the cheating is hidden from the player and achieves a higher purpose of entertaining the player, perhaps by making more challenging or realistic opponents?

Is it considered cheating for an AI player to use information not available to a human in the same position? Is it considered cheating if an AI player takes advantage of the computer's calculation abilities?

Is it fair for the game to stack the deck in the player's favor? Dynamic difficulty adjustment might fall into this category. Certainly I've noticed that players almost never complain about the game handling a situation inequitably, if it is biased in favor of the player.

How should the issue of fairness be handled in multiplayer games where there is a zero-sum game and someone has to lose? The fairness of player-vs.-player mechanisms in MMOs has been hotly debated. It also raises the question of whether fairness is a cultural phenomenon. Would a Korean player find a situation of a powerful player character preying on weaker ones to be fair where an American would not?

Is it fair to allow some players to buy advantages (weapons, armor, upgrades) in a game with money, while other players have to earn those through investment of time and skill? Does it affect the answer if the skill-only player is allowed to create (or steal or win) those same items and turn them in for real-world cash?

CRUSADES FOR THE GAME

I have some strong opinions about some of these questions, but I like the fact that the topic of fairness has ignited a level of interest among designers that I haven't seen since the discussion of "Save the game anywhere"—which now that I think about it, might also have been a fairness rule, too! What do you think?

NOAH FALSTEIN *has been a professional game developer since 1980. His web site, www.theinspiracy.com, has a description of The 400 Project, the basis for these columns. Also at that site is a list of the game design rules collected so far and tips on how to use them. Email him at* ***nfalstein@gdmag.com****.*

RYAN ELLIS

AMBIENT OCCLUSION AND YOU

ASIDE FROM BEING A REALLY ANNOYING buzzword, does the term "next-gen" have any significance? To me it does, because I think it actually means something tangible.

"Next-gen" indicates not only higher quality game assets with more attention to detail, but also the dedication of resources to those ends. It's a mindset. It's the desire of gamers everywhere to be more deeply immersed in their playing experiences. The need to focus on small details is greater than ever.

Using ambient occlusion to add more artistic depth to a video game asset was not really possible in the past—not because we didn't have the technology or the horsepower, but because it just wasn't that important. You could make the argument that our texture sizes where too small to show off these kinds of details, and it would have been hard to justify spending the necessary development dollars to make content using ambient occlusion. But really, the expectation level of the consumer didn't warrant spending time on such features. That has changed with the "next generation." People and press want to be fully consumed in the details of the worlds in which they play. Ambient occlusion works toward this end.

Yes, it's subtle. Yes, it's time consuming. And yes, it looks totally cool.

GAME NOIR

Ambient occlusion is the buildup of shadow in corners. Although that description is not completely accurate, it's a simple way to think about the technique and how or why you can use it.

Figure 1A shows an extruded cube with standard directional lighting—nothing special. Figure 1B, on the other hand, shows the same model with ambient occlusion applied to the ambient term of the material. It looks pretty cool. The image takes on a globally illuminated look.

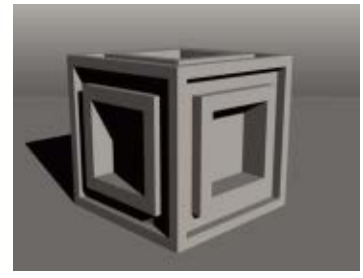


FIGURE 1A **This extruded cube is shown with standard directional lighting.**

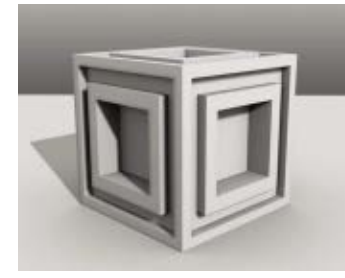


FIGURE 1B **The same cube has been enhanced using ambient occlusion, which was applied to the ambient term of the material.**

Now that we've compared images with and without ambient occlusion, let's look at how to calculate the effect. Rather than pump out a bunch of artistically unfriendly equations, I'll predominately use images to explain the procedures. And although ambient occlusion can be calculated using a few different methods, this column will only explain one in particular, describing what is happening in simple terms.

OCCLUDING AT ITS SIMPLEST

From each sample point on the surface of an object, rays are cast, and the amount of times those rays hit other surfaces dictates how occluded or dark that point will be. Generally, these sample points are evenly distributed on the surface regardless of localized detail, but this can vary from solution to solution.

In Figure 2, you can see that when the rays are cast from the surface, they hit other surfaces. This action thus calculates the ambient occlusion for that point. The more times the ray hits another surface, the darker the point should be.

As the sample points approach the corner, as you can see on the right side of Figure 2, the more rays hit those points, and the darker the samples appear.

Most algorithms have at least some tweakable values for changing the look of the ambient occlusion. Some values are sample, ray length, occlusion spread, and occlusion color.

Samples. The number of samples refers to the accuracy of the sampled points. Generally, fewer samples will result in a less accurate solution, but that's not always a bad thing. For one, it's much faster to use fewer samples. In addition, the renders typically have a grainy look, which may be just what you're looking for if, for example, the ambient occlusion represents a layer of dirt or some other texture that has a grainy feel.

Ray length. Ray length is another variable that can greatly affect the look of the ambient occlusion. It refers to the length of the rays that are cast from each surface point. The longer the rays, the greater the chance they will hit a surface, resulting in more ambient occlusion being applied to the surface.

Occlusion spread. Occlusion spread refers to how far from a corner the ambient occlusion effect appears. Basically, it changes the ratio of ray hits required to darken the surface. Fewer required ray hits will result in a greater spread of the occlusion effect.

Occlusion color. Some ambient occlusion solvers will let you tweak the light and dark area values with a color wheel, which allows you to artificially lighten or darken the effect as well as change the color.

WHY USE IT?

Why use ambient occlusion instead of another technique? First, ambient occlusion is non-directional. This means it can be calculated independent of your main

RYAN ELLIS *is a seven-year veteran of the game industry, who started his career at Oddworld where he became the lead real-time artist on* STRANGER'S WRATH. *He has since moved back home to the Pacific Northwest and is currently an art lead at Zipper Interactive. Email him at* **rellis@gdmag.com**.

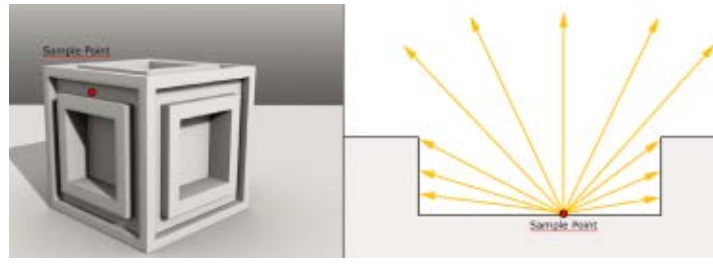


FIGURE 2 **When rays are cast from the surface of an object, they hit other surfaces, and the number of times this happens determines how dark the spot should be.**

lighting solution. It also means the ambient occlusion effect will work with any lighting situation you eventually use. No matter which direction the sun is coming from, the ambient occlusion will look the same.

Another reason to opt for ambient occlusion is that it's typically faster to calculate than a global illumination solution. Speeding up the calculation is usually as simple as dialing down the number of samples. Because of the way ambient occlusion is generated, it is capable of creating much richer detail than a global illumination solution.

Third, since ambient occlusion is based on surface topology, it is much easier to control than other methods and that may have quicker turnaround times.

DIRT, YOUR BEST FRIEND

Aside from producing light and shadow, ambient occlusion can be used to create other visual tricks as well. For example, it can map out the placement of dirt or rust buildup on an object because the visual effects of ambient occlusion collect in the corners and edges, reflecting the behavior of weathering or the pooling of water.

The quickest way to get a basic dirt look is to dial down the samples and or the sample quality until the ambient occlusion begins to get noisy or dithered. Also, depending on your program, you can sometimes change the color of the ambient occlusion effect so that it takes on a dusty hue. If your software package does not support this, you can always post-filter the ambient occlusion in the photo editing suit of your choice.

This procedure works well if you're creating an even, overall coat of dust. But of course, nothing in the natural world is quite that perfect or evenly distributed. To combat the ambient occlusion's even and computer generated feel, you can use a fractal cloud or hand painted mask to modify the effect and create a more random distribution.

If you have the time, a hand painted mask will always produce more realistic results. Just spend a few minutes thinking about how the object behaves in the world and you should be able to come up with a reasonable mapping of dust.

CALCULATIONS

There are a few ways to achieve the actual calculation of the ambient occlusion. Most current 3D software has some sort of render node or baking option that will do it for you. Different packages use various techniques and will result in slightly different looks. Typically, just searching the help docs of your favorite software package will reveal whether it has the built-in capability.

However, if your software does not support it natively, there's still another way. You can actually create a similar look using a ball of directional lights. Depending on how many lights and the size of the shadow maps you use, this procedure can be reasonably fast, or it can be quite slow.

First, make a ball of directional lights. They should be evenly spaced in all directions, as shown in Figure 3. It's important that they are directional lights because you do not want any distortion in the shadowed areas.

Make sure they are all shadow-casting and that the shadows are fully black. Normalize the intensity of all the lights together, meaning that all together the value of the intensity should equal the desired intensity. Typically, this value is around 1.0. So simply divide 1 by the number of lights used, and that is the value of a single light's intensity. For a 60-light dome, each light would be set to a value of 0.0166. You may need to tweak this value later, so it's a good idea to script a node to control each light from a single attribute.

Next, give the shadows a very soft edge. This is done so that there are no shadow edges that are visible. If you skip this step, your image will turn out looking like Figure 4. You can see that the lighting edges create a stadium light effect. Since you're blending each light's shadow together, the softer the shadow edge, the more blending—but there's usually a point when more softening has an unwanted effect on the render. If you think you may have reached this point, do a test render (see Figure 5).

Using as few lights as possible will help keep render times down. Also, using smaller shadow maps will speed up the calculation, but be careful here as poor shadow resolution will create jagged edges. Once you have a nice render of ambient occlusion, you should transfer or bake the light maps out to a texture. As mentioned, each software package handles this differently, but most high-end tools have a reliable feature for this.

IT'S NOT ALL ROSES

At this point you are probably thinking, "What's the catch?" Well unfortunately, there are a few catches, which vary depending on how you use ambient occlusion.

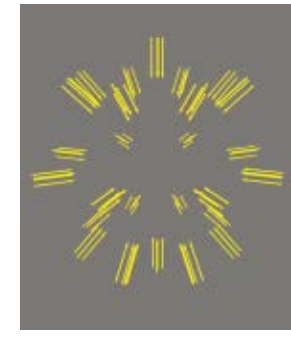


FIGURE 3 **A ball of directional lights is created with each light equidistant from the others. Directional lights prevent distortion in the shadowed areas.**

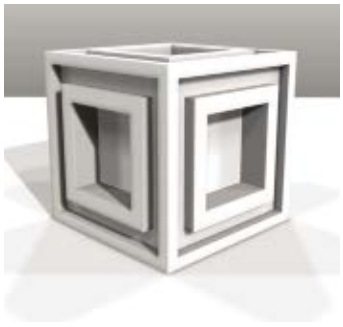


FIGURE 4 **If shadow edges are visible, the ambient occlusion will not generate a clean image.**

FIGURE 5 **Testing is the only way to determine when too much softening has affected the ambient occlusion.**

Let's assume there are a few main techniques: color-map multiply, light-map calculation, and ambient occlusion generation. There are probably more than that, but let's keep it manageable for now. If you choose to use the first method (color-map multiply) which simply means that the ambient occlusion is calculated and used as a multiply layer in your color-map, you can expect a few pitfalls. First and most troubling is that your UV layout needs to be uniquely mapped, meaning that you can't have any overlapping UV faces. Each face must have its own space in your UV page, which is a problem because it makes it so you can't get tricky and share UVs like people are used to. In other words, you can't mirror an object.

There are tricks to get around this problem, such as cutting your model down the center, calculating the ambient occlusion for half, then mirroring it back together. This is time-consuming and doesn't always work.

Along with the UV issues, just getting ambient occlusion values that work correctly is time-consuming. I know that's the name of the game with next-gen content, but it's still something to consider.

If you're lucky enough to have a light-map generator which can also calculate the ambient occlusion, you're ready for the second technique, light-map calculation. With light-map calculation, your light mapper can use the unique mapped UV channel it generated for the standard lightmaps for calculating the ambient occlusion maps, which will obviate the problem from the first technique.

Why not use light-map calculation all the time? Well for one, not all objects will be light mapped. You will still need to calculate the ambient occlusion for real-time objects. Of course they could use the same algorithm as your light-mapped objects, but it's an extra complication.

Another issue with calculating ambient occlusion in this way is that it requires a bunch of programmer time to implement the system. Also, depending on your lighting setup and calculations, the ambient occlusions might not look correct.

However, using light-map calculations for ambient occlusion will take an incredible load off your artists' plates, which will elicit cheers and hugs from your art staff. We all know how much programmers love artist hugs.

The last technique is a hybrid of the two previous methods. How it works is that you have shaders that support an extra texture channel along with a supporting UV channel. Now you could use your lightmap UVs but that would require them to be 1) generated on your source art and 2) accessible to the artists.

That means that you have unique UVs (from the extra channel) and an extra texture map that is multiplied into the surface calculation. The problem with this hybrid method is that you use more texture RAM for the separate occlusion map, your shader becomes more complex and more expensive, and that your geometry gets more expensive from having to store the extra UV channel. Suddenly, it sounds pretty bad, but sometimes the hybrid technique works well for certain objects.

PIPELINE INTEGRATION

Using ambient occlusion will give your game art a certain feel. Run some tests before you fully commit, as it may generate a look that is undesirable for your project. There's also a strange side effect of using ambient occlusion that may not be immediately evident. Depending on what settings you use and how fine the details are that you're getting out of your calculations, ambient occlusion can change the perceived scale of your objects.

If you have decided that you want to use ambient occlusion in your game, the next issue to consider is how to integrate it into your pipeline.

First, talk to your tools team about what they are willing to support. Depending on what they're willing to do and what they have time to do will clarify how much responsibility falls on the individual artists. Typically, the ambient occlusion will be generated from high-resolution models then transferred to low-resolution ones.

Depending whether the shader you use is capable, you will either load the ambient occlusion map as is, or load it into Photoshop so you can modulate the ambient occlusion map over the color map.

It's important to have a standard practice in place when generating the ambient occlusion so that you have a consistent feel across all content.

TRUE COLORS

What does the future hold for this technique? As it becomes commonplace, the tools and methods associated with ambient occlusion will mature and it will be easy for all games to make use of it.

I've seen a few demos and heard rumblings about doing using the technique in real time; as graphics processors become more powerful, this will become more possible. But for the next few years most people will have to settle for pre-baked solutions such as the techniques described in this column.

It's an exciting time for all of us. Our task as an industry will be to find ways to include techniques like this in a financially reasonable way. Happy occluding!

JESSE HARLIN

AURAL FIXATION

THE ART OF CONVERSATION

WHEN RECORDED DIALOGUE FIRST CAME to the big screen, the film industry changed overnight. By contrast, the advent of recorded dialogue for games has been a slow trickle over the last three generations of gaming technology. Only now is voice finally approaching "must-have" status for developers and consumers. As such, voice is a new challenge for many developers. Even for seasoned audio professionals, recent advances in technology mean that game dialogue is still a rapidly expanding area of game design.

PHOTO BY BRUCE HARLICK

LucasArts voice director Darragh O'Farrell directs David W. Collins in a voice acting session for the upcoming THRILLVILLE.

Voice comes in three distinct flavors. Conversational dialogue is the bulk of any game's voice system and covers story, tutorials, cutscenes, sports play-by-plays, and any other linear in-game voice. Utterances are voice files that include hit impacts, taunts, and other short, non-linear pieces of dialogue. Localization is the process of translating your domestic voice set into foreign language sets for European and Asian markets. Each area of voice represents its own unique set of challenges. This month we're going to focus on the process of producing convincing conversational dialogue.

PREP TIME

METAL GEAR SOLID, RATCHET & CLANK, and the MADDEN series all share one thing in common. At the heart of each is a conversational voice system that delivers critical information to the player about story, setting, characterization, and progression through the game. With games continually growing in size and scope, even a small game these days will contain thousands of voice lines. These thousands of lines are recorded one character at a time with fragmented scripts that are often performed out of context and without the benefit of any rehearsal time for the actors.

Though most practical and efficient, this approach is counterintuitive to the goal of cohesive conversational dialogue. Far too often, the result is a voice system riddled with dialogue that sounds glaringly unnatural. Wrong words are stressed for the actual meaning of the sentence. Emotion levels don't follow normal human speech patterns. At its best, badly done dialogue is simply an awkward annoyance to the player. At its worst, it becomes fodder for endless internet mockery.

TOOLS OF THE TRADE

There are two invaluable tools that are a script's best allies. The first is a robust dialogue database using database management software such as FileMaker Pro or Microsoft Access. Each record in the voice database represents an individual line of dialogue and contains fields to denote critical information for each file such as character, level, a unique filename, and the line of dialogue itself. Additionally, each conversation in the game should be numbered with a unique conversation ID number. Each line of dialogue within a conversation is then given a unique dialogue ID number. When the writer delivers the script, it should be delivered as a Microsoft Excel document within which a separate column represents each of these distinct database fields. This spreadsheet is then imported directly into the voice database.

Once imported, the database can be used via macros to search for and sort by conversation IDs and dialogue IDs, organize these lines by character, and output them as a theatrically formatted script where conversational voice is preceded and followed by lines that help to give context and meaning to otherwise disjointed sentences.

Additionally, don't discount the benefits of simply formatting a script's text so as to portray intention. Take this line, for example, when the character Bastila Shan says: "The Force fights with me!"

This line of dialogue from STAR WARS: KNIGHTS OF THE OLD REPUBLIC represents the inherent ambiguity of a game script. Taken completely out of context, the intent of the sentence is unclear. Is "Force," "fights," or "me" the main point of the sentence? Each one radically changes the meaning and delivery of the line. The standard arsenal of punctuation—in addition to bold, italic, and underlined text—will go a long way toward communicating performance intent.

THE RIGHT DIRECTION

The second secret weapon in your arsenal should be a dedicated voice director. A voice director is tasked with running the recording session and represents the last bastion of quality control over dialogue before it's edited and implemented. As such, it's the voice director's responsibility to ensure that lines are recorded as written, all performances are convincing, and critical issues such as pronunciation of character, planet, or weapon names are consistent and correct.

It's important to note that your voice director should have previous game experience. Game voice is a very different beast from traditional animation. A single line of dialogue might be heard thousands of times over the course of a game. A game voice director should be savvy enough to know which is the right take for a game, and which is going to drive players crazy or not serve the game well.

In the end, every step you take to ensure clarity of intent will bring the script closer to being cohesive and convincing. By taking care to meticulously prepare the script and place it into the hands of a professional voice director, you give the script its best chance of fully engaging both actors and audience.

JESSE HARLIN *has been composing music for games since 1999. He is currently the staff composer for LucasArts. You can email him at jharlin@gdmag.com.*

Jade Empire · Dragon Age · Mass Effect · BioWare Austin

artists • programmers • animators • designers • writers • producers • operations

One of the world's largest and best-funded independent game developers, BioWare/Pandemic aims to attract the top industry talent by combining experienced management with a record of innovative, award-winning releases for original and licensed content.

BioWare Corp., with studios based in Edmonton, Alberta and Austin, Texas, is an award-winning game developer of AAA titles.

BioWare Corp. believes that quality is the foundation of our core values:

Quality in our Products
All 14 of BioWare's games are in the top 10% highest rated games of all time at Gamerankings.com
Quality in our Workplace
Winner of Canada's Top 100 Employers 3 years in a row
in a context of humility and integrity.

Visit BioWare Corp. at SIGGRAPH 2006!

HOT JOBS:

BioWare Edmonton
Director of Technology
Project Managers
Lead Visual Effects Artists
Lead Cinematic Animator
Writers
Programmers
Senior Operations
& many more!

BioWare Austin
Senior Environmental Artist
Senior Character Artist
Senior Artist GUI/Special Effects
Senior Project Manager
Writers
Platform Lead Engineer
& many more!

jobs.bioware.com

BioWare | Pandemic Studios

BioWare Corp


ACTIVISION.
activision.com
THIS IS YOUR GAME
• Programmers
• Game Designers
• Scripters
• Producers
• Artists
• Art Directors
• Animators
• Sound Engineers
• Marketing
• Creative Services
• Sales
• IT
radoctane
ACTIVISION.
VALUE PUBLISHING, INC.
NEVERSOFT
infinity ward
Vicarious Visions
RAVEN
Luxoflux
BEENOX
SHABA
TREYARCH
ZAXIS
Toys For BOB
FOR A COMPLETE LISTING OF OPENINGS AND TO APPLY,
VISIT WWW.ACTIVISION.COM (GO TO "CAREERS").

SUPERVILLAIN STUDIOS™

Our legions of Monkey Henchmen and Labor Automatons are feverishly working towards the completion of our current secret weapon! As we approach the zero hour we are actively formulating devious plans for our company's growth! If you would like to be considered for a position, check out our current openings and visit our website for more information! You will submit!

PRODUCTION
- Senior Producer

DESIGN
- Level Designer
- Technical Designer

ART
- Senior Environment Artist
- Senior Character Artist

PROGRAMMING
- Console Graphics Specialist
- Lead Programmer
- Online Programmer
- Physics Programmer
- Gameplay Programmer
- Tools Programmer

www.SuperVillainStudios.com | jobs@svsgames.com

DERIVCO

ARE YOU GAME?

We are hiring only the best

Game Designers - 3D Animators
Flash Designers - Flash developers - C++ Developers

Our home base is in Umhlanga Rocks, South Africa

WWW.DERIVCO.COM


BLUE FANG
GAMES
Now Hiring...
Blue Fang Games, creators of the enormously successful Zoo Tycoon series, is now hiring for these positions:
3D ANIMATORS
3D ARTISTS
SOFTWARE ENGINEERS
What we are really looking for is monster talent, willing to engage, share creative risk and provide some amazing creative leadership. If this sounds interesting and you think you have the chops, get in touch.
artjobs@bluefang.com
engjobs@bluefang.com

CREATE IN NEXT GEN

Imagery from Midway's next-gen console game *John Woo Presents Stranglehold*

jobs.midway.com

> PROGRAMMERS
> GAME DESIGNERS
> ARTISTS
> PRODUCERS
> ANIMATORS
> Q.A. ANALYSTS
> MANAGEMENT

RECENT RELEASES:
AREA 51
BLITZ: THE LEAGUE
MORTAL KOMBAT: SHAOLIN MONKS
NBA BALLERS: PHENOM
THE SUFFERING: TIES THAT BIND

ANNOUNCED NEXT GEN TITLES:
JOHN WOO PRESENTS STRANGLEHOLD
TNA WRESTLING
THE WHEELMAN

Visit us at the Siggraph Career Fair.

MIDWAY

AUSTIN * CHICAGO * LOS ANGELES * NEWCASTLE UK * SAN DIEGO * SEATTLE

We're looking for you.

If you're an individual with technical and artistic vision, we've got a place for you.

rainbow STUDIOS

relic ENTERTAINMENT

volition inc

crankypants games A THQ COMPANY

XDG

HEAVY IRON STUDIOS

helixe

locomotive games

VIGIL GAMES

concrete games

THQ studio australia

bluetongue

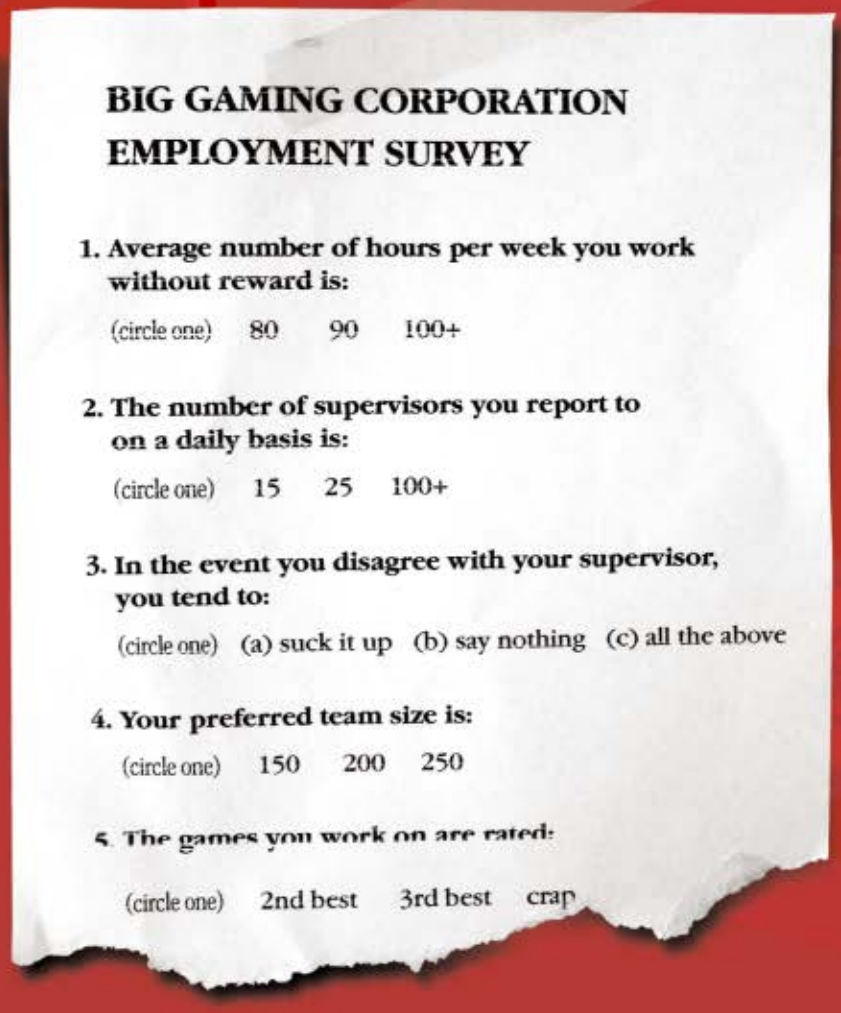
THQ WIRELESS

THQ

www.thq.com

IF LAME WORK AND MEDIOCRITY ARE NO LONGER PART OF YOUR GAME PLAN, CALL JOHNNY.

Are big corporation tactics stifling your talent? Want to feel good about the games you are developing, the technology you are leveraging, and the people you are working with? Call Johnny now at 415-507-7695 or write jprewitt@2ksports.com and tell him how ready you are to enter the gaming elite.



BIG GAMING CORPORATION
EMPLOYMENT SURVEY

1. **Average number of hours per week you work without reward is:**
   (circle one) 80 90 100+

2. **The number of supervisors you report to on a daily basis is:**
   (circle one) 15 25 100+

3. **In the event you disagree with your supervisor, you tend to:**
   (circle one) (a) suck it up (b) say nothing (c) all the above

4. **Your preferred team size is:**
   (circle one) 150 200 250

5. **The games you work on are rated:**
   (circle one) 2nd best 3rd best crap

WE KNOW YOU'RE OVER IT.

Work with world class engineers, designers and artists.
2KSPORTS.COM/JOBS

2K SPORTS · VISUAL CONCEPTS · Kush Games

BEST SPORTS GAMES	BEST BASKETBALL GAME	BEST HOCKEY GAME	BEST GAMES FOR GAMERS
2K : 63% EA : 37%	2K : 80% EA : 20%	2K : 71% EA: 29%	2K : 75% EA: 25%

IGN.COM SURVEY OF 200K GAMERS

FOR IGN SURVEY RESULTS OF THE BEST SPORTS GAMES (EA VS. 2K) : 2KSPORTS.COM/GAMERSCHOICE

NaturalMotion is looking for passionate individuals to work on next-gen projects with major game publishers. Using our company's technology IP, you will create game experiences that were previously thought impossible. We offer a friendly working environment in our Oxford studio and encourage a healthy work / life balance.

CAREER OPPORTUNITIES

Next Gen Game Programmers <

Network Programmers <

Senior Animation Programmers <

C++ Application Programmers <

3D Tools Programmers <

Behaviour Engineers <

Game Physics Programmers <

Animator / TD <

Product Specialist (US) <

naturalmotion

Send your CV to
careers@naturalmotion.com
or vist
www.naturalmotion.com

CORRECTION

In the May 2006 issue, a news story reported console development kits to be reaching universities for the first time. A reader kindly noted that the University of Advancing Technology in Tempe, Ariz., already has GameCube and Game Boy Advance development kits in place for its students. We regret the error.

ADVERTISER INDEX

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.

FASHION PHOTOGRAPHY FINE ART MOTION PICTURES & TELEVISION ADVERTISING INTERIOR ARCHITECTURE & DESIGN

Build your Character

Florida Interactive Entertainment Academy

What is it?

The Florida Interactive Entertainment Academy (FIEA) is a graduate game-development program that teaches you how to succeed in interactive entertainment industries like video games and films. It formed from a unique partnership between the video game industry, the state of Florida and the University of Central Florida.

What will I do?

You'll become part of a team of fellow students who work with you as producers, programmers and artists on real-world projects with milestones and tight deadlines. All overseen by some of the best gaming faculty in the world.

What degree will I earn?

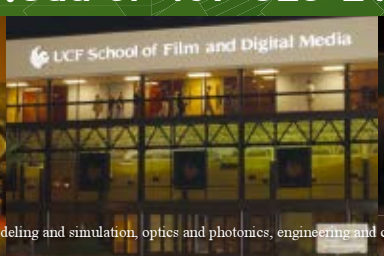
In 16 short months, you'll earn a fully accredited master's degree in interactive entertainment. Specializations include tracks for producers, programmers and artists.

How will FIEA help me get into industry?

Besides putting you through a rigorous, industry-based curriculum, FIEA has a full-time job placement director with contacts in many of the nation's best gaming and film studios. In industry, he placed and recruited hundreds of the top executive producers, designers, development directors, programmers, concept artists and art directors.

Learn more at www.fiea.ucf.edu or 407-823-2121

FIEA

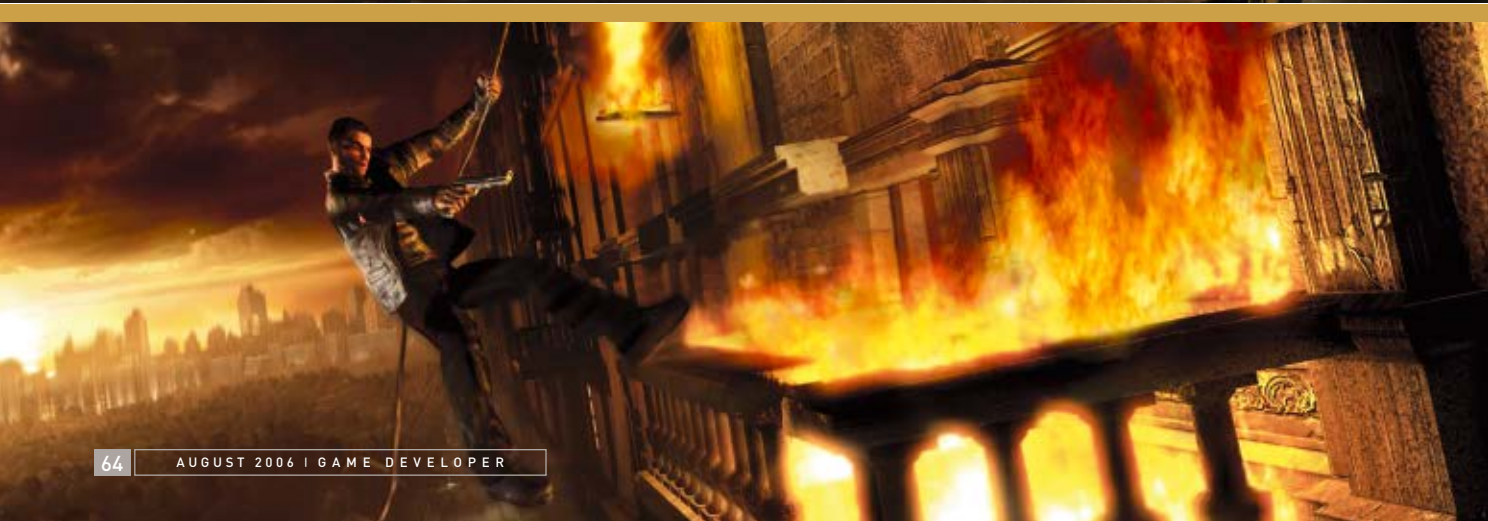


FIEA is a part of the University of Central Florida, the nation's seventh-largest university and a leader in modeling and simulation, optics and photonics, engineering and computer science, education and other disciplines.

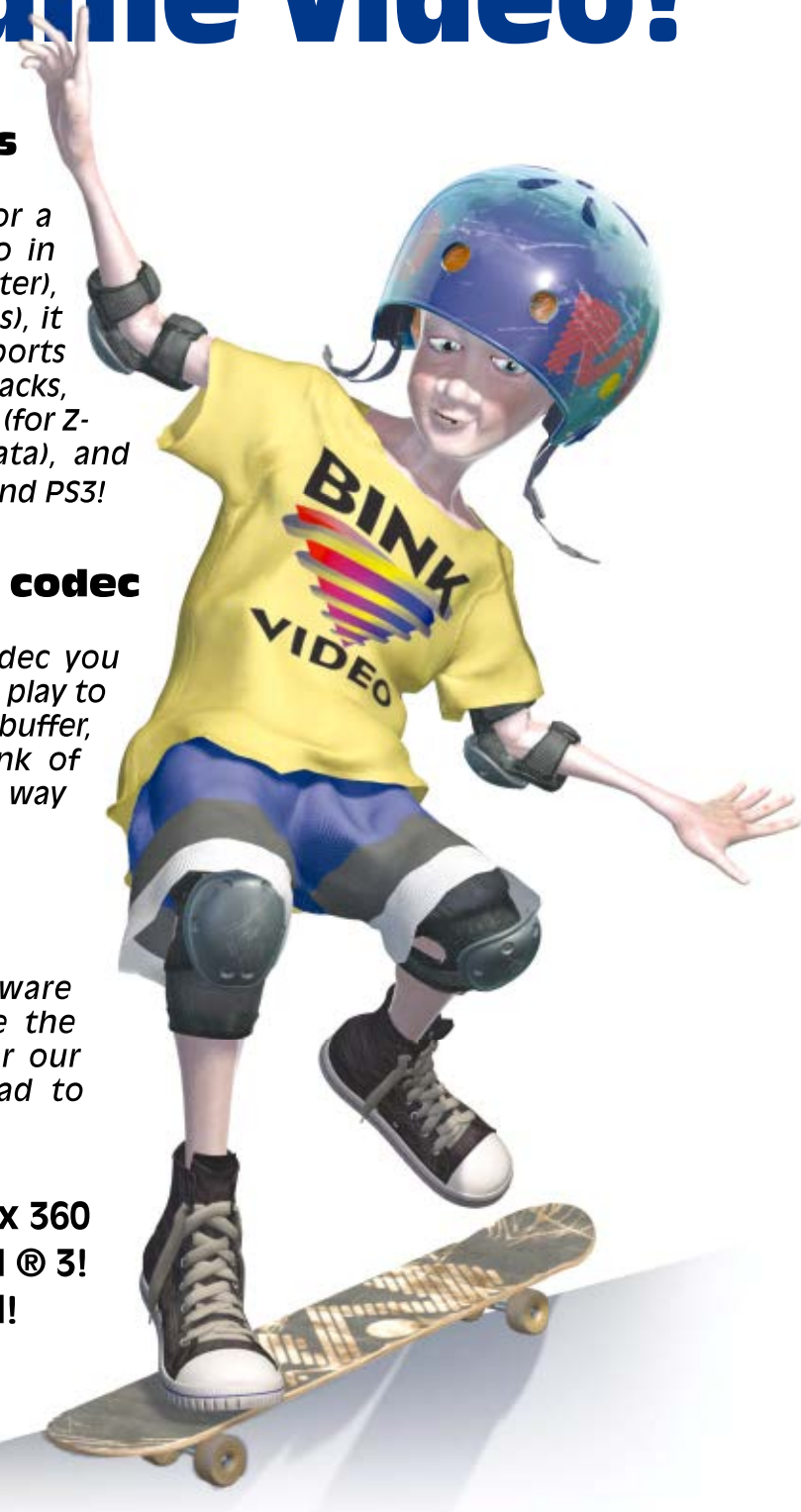
MARKETPLACE

A THOUSAND WORDS

Eden Studios' ALONE IN THE DARK (working title), published by Atari, is the latest in the pioneering survival horror series, and is planned for next-gen consoles. The game hopes to bring emotion to new levels with its gritty realism, under the art direction of Gilles Benois.



Bink takes the grind out of game video!



The video codec for games

Bink has shipped in 2,900 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! Use HD video on Xbox and PS3!

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!

New!

Now shipping for Xbox 360 and Sony PLAYSTATION ® 3!
New Japanese Manual!

For more information:

425.893.4300

www.radgametools.com

© GAME TOOLS

THE BEST IN GAME DEVELOPMENT TECHNOLOGY

Granny 3D Animation System

Miles Sound System

Pixomatic Software Renderer