



United Business Media

TOP DECK The Industry's Most Influential Players

NOVEMBER 2008

game developer

THE LEADING GAME INDUSTRY MAGAZINE

» BUILDING TOOLS

GOOD DESIGN FOR
INTERNAL SYSTEMS

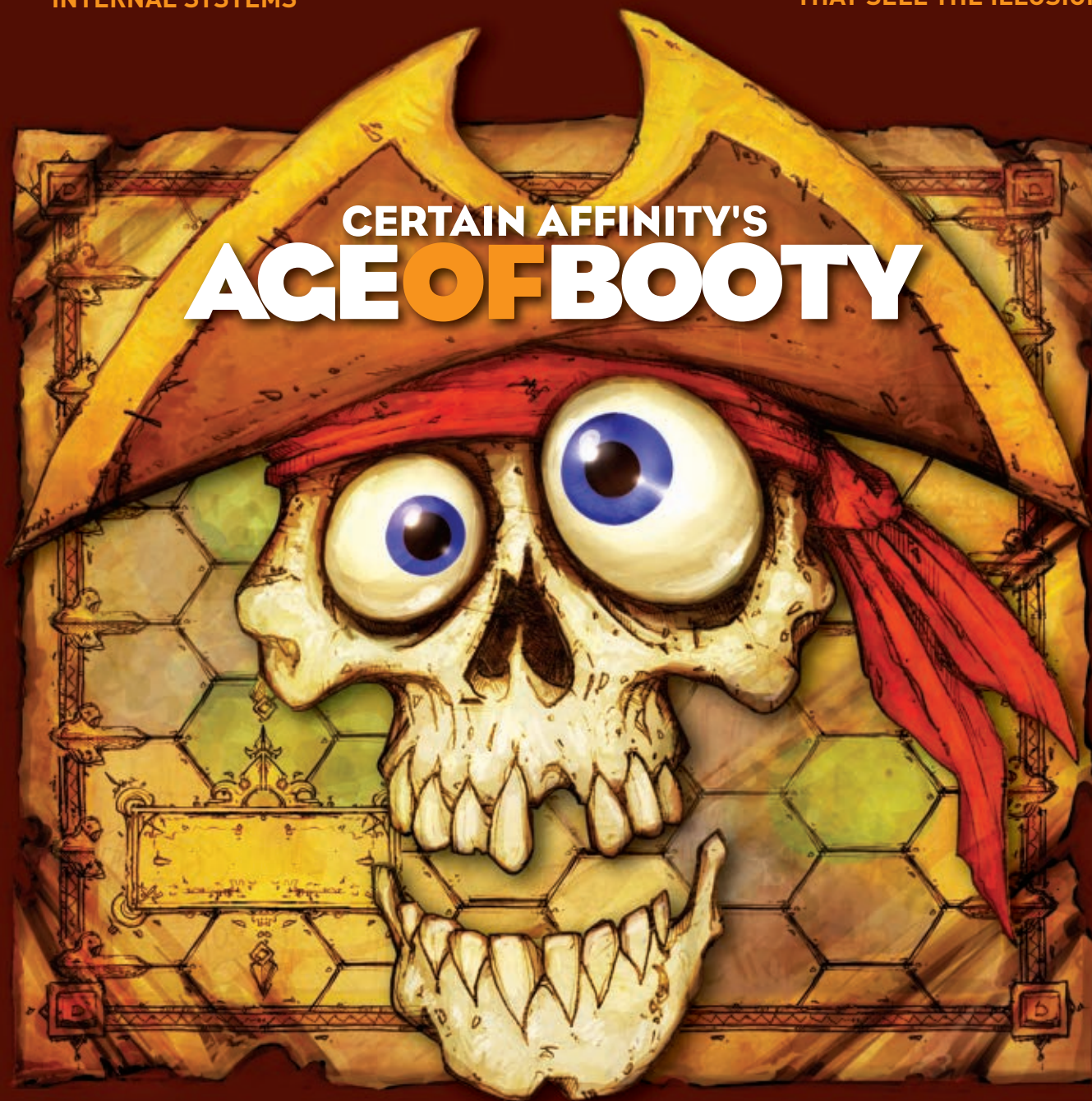
» PRODUCT REVIEW

NVIDIA'S PERFHUD 6

» LITTLE TOUCHES

ARTISTIC FLOURISHES
THAT SELL THE ILLUSION

CERTAIN AFFINITY'S ACE OF BOOTY



gamedeveloper
FRONTLINE
FINALIST '07

"ReplayDIRECTOR rocks. I doubt we'd have found it otherwise. It turned out to be an occasional array overwrite that would cause random memory corruption..."

Meilin Wong, Developer, Crystal Dynamics



BUGS. PETRIFIED.

RECORD. REPLAY. FIXED.

ReplayDIRECTOR™ gives you **Deep Recording**. This is much more than just video capture. Replay records every line of code that you execute and makes certain that it will Replay with the same path of execution through your code. Every time. Instantly Replay any bug you can find. Seriously.

DEEP RECORDING. NO SOURCE MODS.

download today at www.replaysolutions.com

email us at info@replaysolutions.com



1600 Seaport Blvd., Suite 310, Redwood City, CA, 94063 - Tel: 650-472-2208 Fax: 650-240-0403





22

POSTMORTEM

22 CERTAIN AFFINITY'S AGE OF BOOTY

Beginning with a paper prototype and then creating an original engine, Certain Affinity brought AGE OF BOOTY to completion in just over a year, all while juggling multiple projects and shuffling developers on and off the project. Even an unexpected last minute dispute over the name couldn't scuttle the game and Certain Affinity took home the real treasure: full control over its own IP.

By Max Hoberman

DEPARTMENTS

2 GAME PLAN *By Brandon Sheffield*
Social Responsibility

4 HEADS UP DISPLAY
Notes from the Tokyo Game Show, game developers enter the new space race, and more.

31 TOOL BOX *By Amir Ebrahimi*
NVIDIA's PerfHUD 6

COVER ART: CHRIS WOOD

FEATURES

7 GAME DEVELOPER'S TOP DECK

Not all game developers are cards, but many of them are unique in their way—in *Game Developer's* first Top Deck feature, we name the top creatives, money makers, and innovators, highlighting both individual and company achievements. 52 developers, organized by suit, with two jokers to round out the bunch.

By Staff

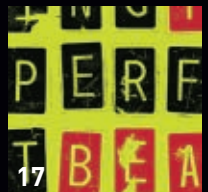
17 BUILDING THE PERFECT BEAST

In-house tools and editors can present clever solutions for asset handling. However, as projects grow over long development cycles, what was once a handy fix can become a costly burden to maintain. A little planning at the onset of a project can make for tools that are easily extendable over the long-term and offer flexible error reporting with more useful real-time feedback, keeping your game on track and your asset creators happy.

By Oliver Franzke



7



17

COLUMNS

36 PIXEL PUSHER *By Steve Theodore*
Gems or Expectations?

[ART]

40 INNER PRODUCT *By Noel Llapis*
Secrets of Multiplatform Data Baking

[PROGRAMMING]

45 DESIGN OF THE TIMES *By Soren Johnson*
Nary a Dime

[DESIGN]

47 AURAL FIXATION *By Jesse Harlin*
GANG Signs

[SOUND]

56 ARRESTED DEVELOPMENT *By Matthew Wasteland*
Famous Emails From Game Development History

[HUMOR]



SOCIAL RESPONSIBILITY

GAMES NEED TO GROW UP. AS THIS MEDIUM follows that inevitable path toward mainstream social acceptance, the limitations as an art form become more apparent. The focus of our work is still far too narrow, or more correctly, our narrow focus is pointed in far too few directions. As the medium that will lead youths into the next generation, I feel that we have a social responsibility to represent a diversity of views in terms of our content.

Every other mainstream media, from books to theatre to movies to comics, has major genre or thematic derivations. Narrow focus isn't a bad thing in itself—I think it can actually help people identify strongly with a given subject. But we greatly need to diversify the themes and subjects we tackle.

In video games, the vast majority of content is still combat and competition-based. This isn't a problem, after all most games of all types (not just electronic) are about good-natured competition. The trouble is that due to the common theme, the message is often quite simple—there are good guys, there are bad guys, and probably the bad guys aren't who you thought they were at the start, but really you don't care as a player—you just want to keep shooting, smacking, or otherwise subjugating whatever's in front of you.

SOCIAL CONTEXT

Games have been dealing with social issues for as long as they've had narratives. Unfortunately they usually have very shallow messages to impart. War is bad because it killed your family. People should understand each other, because your character used to be poor. The intentions are good, but generally the message is told to the player, rather than shown to him. If you want to be told that war is horrible, play METAL GEAR SOLID 4. If you want to be shown, play CALL OF DUTY 4. If you want to be told about the dangers of capitalist extremism and its dystopian results, play FINAL FANTASY VII. If you want to be shown, play BIOSHOCK.

The examples are a bit trite, as these are the games everyone trots out when they want to praise the future of narrative. But my point is only further validated by the fact that better examples are still very difficult to find.

ENTERTAIN TO INFORM

If games are going to be tackling social issues, which most narrative games seem to strive toward, there needs to be more outside influence. I don't mean outside the industry—I mean game developers need to draw more from their daily

lives and other media for inspiration. Right now, games are too influenced by other games. People know games, and they're safe—we need to move outside the comfort zone if we're to make any impact. BIOSHOCK's ode to Ayn Rand is a good start, and games like CIVILIZATION do a good job of simulating real-world economies and warfare—but we need more examples to point to. A Dickens or Fitzgerald-inspired game, properly handled, would yield amazing results. Or a largely original work like BRAID. This sort of thing is usually relegated to the Experimental Gameplay Sessions panel at GDC—but these sorts of games should actually be made.

WHO NEEDS IT?

Really, most games don't need complex narratives or themes. We insert them because we want our games to have an "awesome story," but most games fall horrifically flat here, and would be better off with simple objective screens. In a game like HALO 2 or 3 for instance, where you can't even understand what the characters are saying, and the plot is needlessly convoluted—wouldn't the experience be better with no story at all?

One of the large problems is the lack of a true director or auteur. The compartmentalization of leadership in Western game companies has its serious advantages in terms of workflow, but one thing Japan still has over us is singular vision. One person truly directs the project and has the final say. This yields both astounding successes and spectacular failures, but if nothing else, helps to point a game in a specific direction.

BROADLY NARROW

I think most current games, even the hardcore ones, appeal too much to the mainstream in terms of their themes. Just as the movie 300 appeals to frat boys, so too does GOD OF WAR or SOCOM. Our blockbuster games are designed to be mainstream, even if they do only appeal to the hardcore by and large [see my previous editorial "The Hardcore Niche," June/July 2008].

I want to believe that game developers care about more interesting things. We should be showing more of this in our games. We need to present a diversity of viewpoints, themes, and gameplay styles to the people who are absorbing and internalizing our content (when we do it right). 2007 proved that games with vision can actually be popular. So now, the only limiting factor is our own creativity.

—Brandon Sheffield

Think Services, 600 Harrison St., 6th Fl.,
San Francisco, CA 94107
t: 415.947.6000 f: 415.947.6090

SUBSCRIPTION SERVICES
FOR INFORMATION, ORDER QUESTIONS, AND ADDRESS CHANGES
t: 800.250.2429 f: 847.763.9606 e: gamedeveloper@halldata.com

EDITORIAL

PUBLISHER
Simon Carless scarless@gdmag.com
EDITOR-IN-CHIEF
Brandon Sheffield bsheffield@gdmag.com
PRODUCTION EDITOR
Jeffrey Fleming jfleming@gdmag.com
ART DIRECTOR

Joseph Mitch jmitch@gdmag.com

SENIOR CONTRIBUTING EDITOR
Jill Duffy jduffy@gdmag.com

CONTRIBUTING EDITORS
Jesse Harlin jharlin@gdmag.com
Steve Theodore stheodore@gdmag.com
Noel Llopis nllopis@gdmag.com
Soren Johnson sjohnson@gdmag.com
Damion Schubert dschubert@gdmag.com

ADVISORY BOARD

Hal Barwood Designer-at-Large
Ellen Guon Beaman Microsoft
Brad Bulkeley Neversoft
Clinton Keith High Moon Studios
Ryan Lesser Harmonix
Mark DeLoura GreenScreen

ADVERTISING SALES

MEDIA ACCOUNT MANAGER
John Malik Watson jmwatson@think-services.com t: 415.947.6224

GLOBAL SALES MANAGER, RECRUITMENT & EDUCATION
Aaron Murawski amurawski@think-services.com t: 415.947.6227

ACCOUNT MANAGER, EDUCATION AND RECRUITMENT
Gina Gross ggross@think-services.com t: 415.947.6241

SR. EVENTS ACCOUNT MANAGER, SOUTHWEST
Jasmin Davé

ACCOUNT MANAGER, WESTERN CANADA, INDIA, AUSTRALIA, & ASIA
Amanda Mae Miller

ADVERTISING PRODUCTION

PRODUCTION MANAGER Pete C. Scibilia pscibili@ubm-us.com

REPRINTS

PARS INTERNATIONAL
Joe Nunziata t: 212.221.9595 e: reprints@parsintl.com

THINK SERVICES

CEO THINK SERVICES Philip Chapnick
GROUP DIRECTOR Kathy Schoback
CREATIVE DIRECTOR Cliff Scorso

AUDIENCE DEVELOPMENT

GROUP DIRECTOR Kathy Henry khenry@techinsights.com
DIRECTOR Kristi Cunningham
kcunningham@techinsights.com

LIST RENTAL Merit Direct LLC t: 914.368.1000

MARKETING

SERVICES MARKETING DIRECTOR Karen Tom ktom@think-services.com
SERVICES MARKETING COORDINATOR Laura Robison lrobison@think-services.com

UBM TECHNOLOGY MANAGEMENT

CHIEF EXECUTIVE OFFICER David Levin
CHIEF OPERATING OFFICER Scott Mozarsky
CHIEF FINANCIAL OFFICER David Wein
CHIEF INFORMATION OFFICER Kevin Prinz
CORPORATE SENIOR VP SALES Anne Marie Miller
SENIOR VP, STRATEGIC DEV. AND BUSINESS ADMIN. Pat Nohilly
SENIOR VP, MANUFACTURING Marie Myers
SENIOR VP, COMMUNICATIONS NORTH AMERICA Alexandra Raine



United Business Media

morpheme™

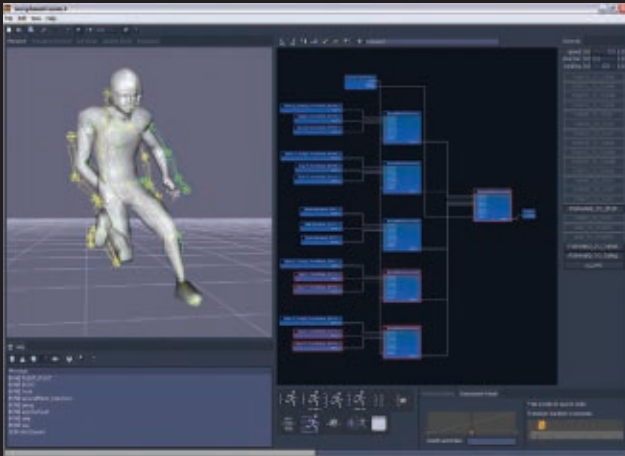
advanced animation system

morpheme is the industry's first graphically authorable animation engine. morpheme consists of morpheme:runtime: an advanced runtime animation engine for PLAYSTATION®3, Xbox 360™, Wii™ and PC. morpheme:connect: a highly-customizable 3D authoring application.

morpheme gives animators and developers unprecedented control over the look and feel of their animations in-game: blends, transitions, compression, etc. can all be previewed and modified graphically in morpheme:connect and live on the target platform.

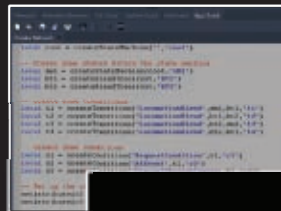
morpheme:runtime ships with full source code and integrates seamlessly with euphoria, NaturalMotion's Dynamic Motion Synthesis technology.

For more information, visit www.naturalmotion.com



scripting

Full Lua scripting for automating tasks, adding AI logic or polling game pads for real-time input



timeline

Graphical mark up of animation data to add one-shot and duration events, for highlighting footfalls, sound effects, etc.



node palette

Advanced blend notes for dragging and dropping into transition network. Fully customizable node types through C++ and scripting



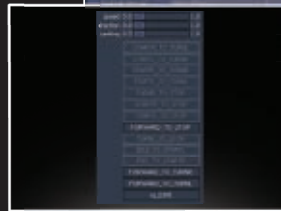
animation browser

Easy browsing and selection (drag & drop) of source animation. Animation list is automatically updated to reflect changed source files



transition requests

Exposure of custom transition messages. In-tool emulation of interaction between morpheme:runtime and game AI system



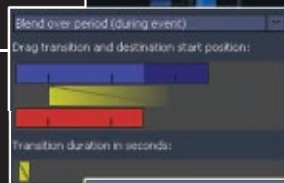
blend tree

Advanced graphical tools for building complex blend trees. Real-time visualization of animation source contribution through node highlighting



blending

Graphical control of transition blending between states in the transition graph



multiple characters

Visualization of multiple runtime characters in morpheme:connect for easy authoring and analysis of character interaction



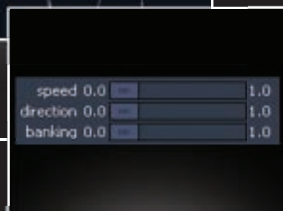
network

Advanced graphical tools for creating and visualizing transition networks through drag-and-drop



control parameters

Exposure of custom high-level controls for entire animation system. Real-time manipulation through sliders or game pad controller



TOKYO SENSE OF WONDER NIGHT

AT TOKYO GAME SHOW ON FRIDAY THE 10TH OF OCTOBER, multiple entities, including industry association CEDEC, and TGS organizers Nikkei BP and IGDA Japan combined to present the unique Sense Of Wonder Night, an evening showcasing 11 independent games from all over the world—with the creators on hand to give presentations, and translation between Japanese and English making for a truly international flavor. The showcased games were picked from a set of game submissions made earlier this year for the contest.

IGDA Japan head Kiyoshi Shin acted as effusive MC for the evening, which was inspired by the GDC's Experimental Gameplay Workshop in format, with just 10 minutes (including questions) for each set of creators to show and explain their games.

The loose, informal format was very much appreciated by the packed crowd of around two to three hundred, a significant majority Japanese, who packed one section of the Restaurant NOA at Makuhari Messe, adjacent to the Tokyo Game Show.

A particular theme of the SoWN selections were new forms of expression and emerging genres in games, from titles using physics and flocking, through titles that were prototypes or sketches designed to make you think about the nature

of video games. The light-hearted, wry approach was quite a contrast from the relatively formal strictures of TGS' Business Day and expo floor.

The order of the presentations for Sense Of Wonder Night 2008 was as follows:

- **CAMERA** (Yareyare, Japan) (not available online)
- **DEPICT** (Jesus Cuahquemoc, Moreno Ramos, Mexico) (<http://depict.villavanilla.net/?q=node/3>)
- **THE UNFINISHED SWAN** (Ian Dallas, USC, United States) (<http://iandallas.com/games/swan>)

- **WORLDICELANSISTA** (Ambition, Japan) (<http://wil.tv/pc>)
- **TWIN TOWER** (OMEGA, Japan) (http://nagoya.cool.ne.jp/o_mega/product/tower.html)
- **PIXELJUNK EDEN** (Q-Games, Japan) (<http://pixeljunk.jp/library/Eden>, out now on PS3)
- **GOMIBAKO** (Trash Box/PlayStation C.A.M.P.I., Japan) (www.indiegames.com/blog/2008/10/preview_gomibako_trash_can.html, PS3 version due out soon)
- **MOON STORIES** (Daniel Benmergui, Argentina) (www.ludomancy.com/blog/2008/09/03/i-wish-i-were-the-moon)
- **THE MISADVENTURES OF P.B. WINTERBOTTOM** (The Odd Gentlemen, United States) (www.winterbottomgame.com/game)
- **GENOCIDE AUTOMATION** (Naoya Sasaki, Japan) (www.11.plala.or.jp/normal)
- **NANOSMILES** (Yu Iwai, Japan) (<http://engrishgames.blogspot.com/2008/02/nanosmiles.html>)

Following the end of the presentations, the judging committee, which included KATAMARI DAMACY and NOBY NOBY BOY creator Keita Takahashi, Enterbrain's MAKER series director Kenji Sugiuchi, and Vector.co.jp executive Takashi Katayama, as well as the author of this article, were asked to give their impressions on what we had seen.

Particularly notable were the clipped comments from an ever enigmatic Keita Takahashi, who claimed that he hadn't been to Tokyo Game Show for 8 years—but that this event had drawn him to attend.

The evening ended with assurances from Shin that Sense Of Wonder Night will occur again next year—a good thing for uniting the indie game scene in Japan with the rest of the world.

—Simon Carless



Overheard at the Tokyo Game Show

Treasure president Masato Maegawa on the line between making money and doing what you love:

"We won't grow the company, no matter how many people want us to make games. I don't want a big company. It becomes too complicated. This company was created for us to make the kinds of games we want to make, and if I think more than 20 or 30 people are trying to agree on what kind of game they want to make, that's probably too much. I think 20 or 30 people is the best size for a game company. I don't think there's much

merit in making a company larger. Everyone here is basically a game maniac. Too many different views would clash.

And no, I don't want to become rich. If I decided to make games for money, I really don't think I could make the kinds of games I want to make, to speak plainly! So I've got to kind of bear with it and make what I want. It's not like I don't think about what games will sell, but I can't spend too much time worrying about what will appeal to the widest audience if I want to have fun myself."

Masaya Matsuura, president of NanaOn-Sha on what it means to be a musician in games:

"A musician has to be a musician. So if a musician doesn't make their own music, that's not a musician! I really want to say musicians shouldn't belong to the game company. So quit anyway! And making your own performance, or your own CDR or something, and appealing to the audience ... that's a very heavy mission, but a very important thing to have.

Finding your own mission by yourself is very difficult. It's hard to get motivated, and I understand

that. I'm writing the soundtrack for our new game ... but I'll keep doing something else, writing on the side. I really like to do performances sometimes. I don't have many chances to do it though. If I play in the street, maybe you can cross the street by accident, so that requires no intent, in terms of 'music interaction.' But these kinds of encounters are a very basic function that music has. I'm thinking about that a lot lately, and how to turn that into games, but it's still very experimental."



TOP LEFT: Armadillo Aerospace's 'Pixel' in flight. ABOVE: Garriott on board the International Space Station.

ROCKET MEN

GAME DEVELOPERS JOIN THE NEW SPACE RACE

ON OCTOBER 12TH RICHARD GARRIOTT BECAME the sixth space tourist to visit the International Space Station. Taking off from the Baikonur Cosmodrome in Kazakhstan in a Soyuz TM spacecraft, Garriott traveled with Russian cosmonaut Yuri Lonchakov and American astronaut Michael Fincke to the station where he will spend almost two weeks in orbit. At the time this issue goes to press he is set to return to Earth on October 24. As a "Spaceflight Participant" Garriott will use his time in orbit performing a number of small-scale biology experiments and educational outreach to students.

The ride reportedly cost \$30 million in a deal brokered with the Russian Federal Space Agency (RKA) by the space tourism company Space Adventures. The U.S.-based company has arranged flights for private citizens to the ISS since 2001. Garriott is a member of Space Adventure's board of directors and his father, Owen Garriott, who was an astronaut on NASA's Skylab mission, also serves on the company's Astronaut Advisory Board.

A long time supporter of private space development, Garriott joins fellow game developer Will Wright on the board of trustees for the X Prize Foundation. The foundation was established in 1995 by the co-founder of Space Adventures, Peter Diamandis, and sponsored the \$10 million Ansari X Prize for the first private company to attain manned sub-orbital flight (won in 2004 by

Scaled Composites' SpaceShipOne). Since then, the foundation has branched out, establishing prizes in the fields of genomics, fuel-efficient vehicles, and lunar exploration. As part of a suite of prizes that are intended to incentivise space exploration, the foundation is facilitating the Google Lunar X Prize and the Northrop Grumman Lunar Lander Challenge.

Among the teams competing in the Lunar Lander Challenge is John Carmack's Armadillo Aerospace. Founded in 2000, the small start-up has focused its efforts on Vertical Take Off/Vertical Landing vehicles (VTOL). Using a software engineering-like approach, Armadillo has kept costs low by rapidly building and testing incremental engine designs that emphasize simplicity and reusability. So far the company has successfully built an unmanned rocket that can take off and within 90 seconds travel 100 meters down range to make a controlled landing at a second launch pad. Competing in the 2007 Lunar Lander Challenge, Carmack's team came close to victory but engine trouble and a hard landing prevented it from making a successful return flight to the original launching point in order to win the prize. The 2008 competition was set to take place this October at Holloman Air Force Base but has since been postponed to a yet to be determined date.

In addition to its VTOL efforts, Armadillo Aerospace's liquid oxygen/alcohol fueled

engines are being used to power rocket planes for the Rocket Racing League. The league was also formed by Peter Diamandis (in 2005), and hopes to popularize high-performance, rocket powered sky races.

—Jeffrey Fleming

CALENDAR

Independent Game Conference

Austin, Texas
November 20–21
Price: \$250

www.independentgameconference.com

DIG London Game Conference

London, Ontario, Canada
November 27–28
Price: \$150

<http://game.diglondon.ca>

Asia Game Show

Hong Kong, China
December 19–22, 2008
Price: See Web Site

www.asiagameshow.com



Tools for Creation

creation 3D
editing
audio/video
motion capture
tracking
2D-3D



Understanding your workflow
Enhancing your pipeline



Autodesk

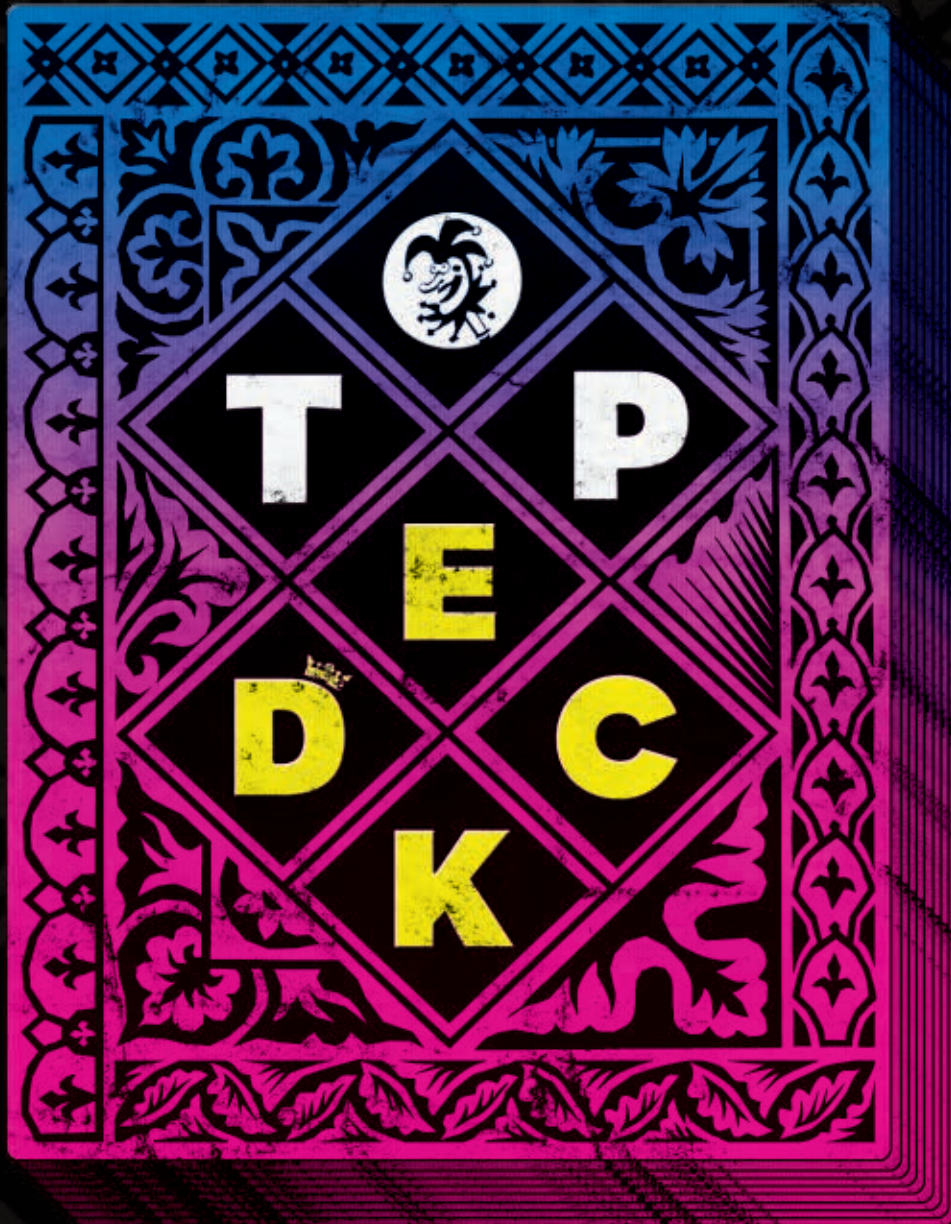
Media and Entertainment
Authorized Reseller

LET US HELP YOU, CONTACT US

gamedeveloper@3vis.com / 1.866.WWW.3VIS (999.3847)



GAME DEVELOPER'S



2008

✦ YOU'VE SEEN LOTS OF TOP LISTS BEFORE, BUT WE HERE AT *GAME DEVELOPER* magazine decided it was time for a distinctly focused but slightly alternative take on the important personalities in the game business.

Thus, *Game Developer's* Top Deck was created to recognize those members of the game development community who have—either individually, or as part of their company—made particularly outstanding achievements in the past year. The picks were made by *Game Developer's* editors.

Each “suit” of the Top Deck represents a group of game creators and businesspeople who distinguished themselves particularly well in a specific area of the game industry—specifically, Trailblazers,

Progressives, Ambassadors, and Capitalists. The suits themselves are not ranked, nor are the persons within them.

This does not suggest that any given person on our list didn't contribute in multiple different arenas—but this is where we felt they shone particularly bright this year. In addition, we are aware that the vast majority of games and product lines are not made by a single person.

So, while one individual is generally mentioned, we would like to acknowledge here that none of the people on this list would be here without the support of those who work with them. Nonetheless, individuals have to spearhead, mastermind, and create—and we're delighted to be honoring them in the first ever Top Deck. Onward:

TRAILBLAZERS

The folks in the Trailblazers group have made the world easier for their fellow developers by going where no one has gone before—or at least, not to the extent that these individuals did, or with as much obvious success.

From standing up against piracy to simply making effective systems, these folks have evolved the business in ways that will continue to be emulated.



ROB PARDO

BLIZZARD ENTERTAINMENT

Not only has WORLD OF WARCRAFT shown the world that there are at least 10 million dedicated PC gamers out there, it has undisputedly proved the mass appeal of MMORPGs. Pardo was instrumental in creating this phenomenon, and with the WRATH OF THE LICH KING expansion and a massively successful BlizzCon, 2008 continue to be a banner year for the company. And that's notwithstanding the upcoming dual hammer of STARCRAFT II and DIABLO III, plus the company's next MMO, of course.



MASAHIRO SAKURAI

SORA

Sakurai, once a designer at HAL Laboratory, is best known for his creative influence over both the KIRBY and SUPER SMASH BROS. series. While he has since started his own company, Sora, he has continued to work on the SMASH BROS. series, and the latest iteration is what gets him on this list. Not only a palpable game design mash-up success, Sakurai assembled over 40 different sound composers to create music for the game, making the project almost a jam band-style get-together. In an age of licensed soundtracks, this is to be applauded.



JASON KAPALKA

POPCAP

The only major casual game developer to both enchant the everyday gamer, while impressing the hardcore, BEJWELED and PEGGLE creator PopCap has got the balance just right, and chief creative officer Kapalka has been there since the company's genesis. Not only excellent at brand maintenance, PopCap seems to have mastered brand creation and extension, with BOOKWORM ADVENTURES and PEGGLE NIGHTS just two of the titles that continue the company's focus on broad entertainment.



TIM SWEENEY

EPIC

No other game engine out there has had such an impact as Unreal Engine 3. It is more ubiquitous than even Renderware was in its heyday, and lawsuits and quirks aside, there's got to be a reason nearly everybody uses it. Tim Sweeney, as the main architect of this beast, has opened up the market for developers looking to cut costs and prototype early, while also supporting the company's own original software. Sure, it may not be cheap, but this little "side business" has turned into Epic's largest contribution to game culture thus far, and given Sweeney's history in game tools [see: ZTT], it's only to be expected.



ALEX EVANS

MEDIA MOLECULE

PS3 standout title LITTLEBIGPLANET is blazing new frontiers for user-generated content on consoles, and for co-operative content generation too. Evans, one of the top creators in the PC demo-scene

in the 1990s, and subsequently at key British talent nurturer Lionhead, is one of the main architects of the LBP experience. And what's most notable about the Media Molecule success story is that it's the team's first title together—a significant achievement.



CLINT HOCKING

UBISOFT MONTREAL

One of the best industry stories of recent years surrounds FARCRY 2, when it was discovered that this perpetual living world, led in its construction by creative director Clint Hocking, could burn to the ground. The story goes that a tester set a fire in the game's forest, and then went for lunch. When he came back, he learned that the fire had spread through the whole world, killing the final "boss" in under two hours. Though the game has since been tweaked, that's what a living world is all about—and Hocking, previously instrumental to the SPLINTER CELL series, is leading that particular charge.



MARK JACOBS

EA MYTHIC

Jacobs has presided over an already-impressive 750,000 subscriber haul for his studio's newest MMO title, WARHAMMER ONLINE, not distracted by the DARK AGE OF CAMELOT creators' acquisition by Electronic Arts. Not only that, he's opinionated and passionate in public—unusual for an executive in today's PR-removed market. Following the removal of 400 people farming for gold on WARHAMMER's servers, he posted on his blog: "I hate goldfarmers with every fiber of my being." Whether right or wrong (and probably right), this kind of public passion is often lacking in today's game biz.



DYLAN CUTHBERT

Q GAMES

Relatively obscure in the West until the launch of the PIXELJUNK games on the PS3's PlayStation Network, Cuthbert's Kyoto-based studio has close relationships with both Sony and Nintendo, working on elements of the PS3 operating system and on STARFOX DS. But by spearheading his Japanese studio to work on a largely Western medium (downloadable console games) and make it work, most recently with the transcendent PIXELJUNK EDEN, Cuthbert deserves a place on this list.



KEN LEVINE

2K BOSTON


After BIOSHOCK debuted in late 2007, the reverberations of its unconventional original IP success impacted the industry more than you might guess. The game has one of the more sophisticated narrative structures of any released so far, in its integration of visual, oral, and textual storytelling into the actual gameplay. With 2K Marin working on the sequel, and the movie version fast-tracked, plus creator Levine now working on a mysterious new project, his influence has not diminished.




MARK BEAUMONT

CAPCOM


Of all the Japanese-headquartered publishers operating in the West, Capcom perhaps has the most invigorating autonomy from its bosses back East. Beaumont is taking great advantage of this, thanks to a major digital download initiative spawning titles like AGE OF BOOTY, and is funding Western-developed titles such as BIONIC COMMANDO and DARK VOID. Most impressively, a robust blog, community, and fanbase are letting Capcom's classic Japanese franchises breathe and flourish, even as the West grows ever stronger.

4
 **ALEX WARD**
CRITERION GAMES

BURNOUT PARADISE is blazing new trails for an extended life of downloadable content, thanks to the "Year Of BURNOUT" promotion. Sure, there's been plenty of DLC before, but so comprehensive, well-planned, and advertised in advance? Not so much, and thanks to Ward and his team, many players have been convinced to buy and hold onto his title, awaiting the extra vehicles, motorbike additions, new levels, modes, and more. Its status as one of the first full-price titles to be downloadable on PlayStation Network further cements BURNOUT PARADISE'S—and Ward's—status as a trailblazer.

3
 **JONATHAN BLOW**
NUMBER-NONE

Jon Blow, the creator of breakthrough XBLA title BRAID, is a wonderfully unvarnished game creator—one who is willing to voice his opinion, no matter what the reaction may be. But it turns out he's got the chops to back it up. His latest title, BRAID, tackles emotions and relationships in a very circuitous but interesting way—appropriate for a rather complicated fellow like Blow. The lesson here? When you are beholden to no-one, and make your games yourself, you can speak your mind without repercussions, and truly advance them as entertainment.

2
 **SARAH CHUDLEY**
BIZARRE CREATIONS

Not only one of the UK's top developers, Bizarre Creations' acquisition by Activision has cemented its place as an important game creator at multiple levels, headed by Chudley and colleagues. From the landmark downloadable GEOMETRY WARS series, through the PROJECT GOTHAM RACING franchise and whatever new racing titles the company may develop under Activision's wing, even partial miss-steps such as THE CLUB haven't dampened the cutting edge at Bizarre Creations. Long may it continue.

AMBASSADORS

The personalities in the Ambassadors group have expanded the market in ways that nobody would have expected 10 years ago. Games are not only increasing in users, but also in media mindshare—the audience is broadening tremendously.

From new delivery methods to new platforms, without the recent contributions of these people, games would not be in the position they're in today—which is far more mainstream and influential one than it was ten years ago.

A
 **GABE NEWELL**
VALVE


Under Gabe Newell, Valve's PC digital download service Steam has gathered over 15 million users. That's a lot of people, especially for a platform that some cynics are continually discounting as dead—perhaps due to a spreading-out of revenue rather than an actual decrease. Steam releases developers from the tethers of retail, and gives consumers a much better user experience, while providing a piracy-free alternative to boxed copies. Not only that, since it's developer-run, the service is certainly different from a traditional publisher arrangement. In 2008 this sounds like a re-statement of the obvious—but can you imagine the current game industry without it?

K
 **SATORU IWATA**
NINTENDO

The Wii and DS both came out some time ago, but this year, the platforms continued to deliver on several of the Kyoto-based company's rather bold promises. Nintendo has opened the idea of games up to new users to a degree that the company itself didn't even anticipate—and can now claim the two most purchased consoles on the world market. Even though it might be Nintendo first-party titles dominating the top of the charts—scant consolation for third parties—progressions like WiiWare, the DSi and MotionPlus continue to move things forward.

Q
 **WILL WRIGHT**
EA MAXIS


SPORE has been lauded as the next big thing for several years now, and this year, it's finally released—which should make it this year's big thing! What the game—one of the most technically advanced and innovative so far—does very well is introduce new users to sophisticated, evolutionary gameplay. The Maxis masterpiece is simple and accessible on the surface, but beyond that, Wright's latest is a world inside your computer, and for scientific ambassadorship alone, gives the SIMCITY and THE SIMS creator a place on this Deck.

J
 **STEVE JOBS**
APPLE

For ages, mobile game companies have been touting the numbers—billions of handsets, billions of potential customers. Soon thereafter, another North American cell phone game firm closes, consolidates, or otherwise downshifts. Now, with Apple's iPhone, the field becomes a tad more even. Though not yet a gigantic market, with an actual store to purchase games from, a pleasant (and somewhat new) interface, and the ability to for developers to circumvent carriers and third party publishers, the iPhone can potentially truly bring games to a whole new group of people. Jobs and Apple have truly created a gaming platform here, for the first time since the Apple II.

10
 **MIN KIM**
NEXON

MAPLESTORY creator Nexon, "big in Korea" since practically the year dot, has promoted free-to-play PC online games from both the consumer and developer sides for several years now. Among its ranks, the firm's Min Kim has been the most vocal evangelist of this business model in recent years, speaking at conferences worldwide, and lately it seems people have been starting to listen. You can make hundreds of thousands or even millions of dollars a month just from microtransactions and other alternative forms of revenue—and even though too many people may be chasing that dream, it's Nexon and Kim that blazed the way.

9
 **HIDEO KOJIMA**
KONAMI

There are few game creators with the name recognition of Hideo Kojima, and even fewer still that can move significant amounts of console hardware with the release of a single title. Kojima did just that, giving the PlayStation 3 an extra 200,000 unit sales boost in North America this past June, when METAL GEAR SOLID 4 was released. Beyond his importance to Sony's bottom line, however, is Kojima's unwavering insistence that video games are a story-telling medium on par with literature and film. Even if opinions are mixed on whether it succeeded as a narrative, the game remains a study on the power of story to engage players.

8
♣

ALEX RICOPULOS

HARMONIX

It's probably the cultural and stylistic forces at work at ROCK BAND (and original GUITAR HERO) co-creator Harmonix that enchant us the most, and lead to Rigopolos' appearance in the Deck for Ambassadors. By which, we mean—Konami has been making music games for years that have cultural resonance in their home land, but Harmonix has always understood how to infuse the passion of music into games. The downloadable content around ROCK BAND and ROCK BAND 2 is some of the most careful broadening of the market in some time—sure, we get Metallica, but The Grateful Dead? Jimmy Buffett? Delightful.

7
♣

LANE MERRIFIELD

CLUB PENGUIN

Subscription-based MMOs are often described as the exclusive realm of the hardcore, but Lane Merrifield's bright, surprisingly soulful CLUB PENGUIN proves otherwise. In his keynote speech at Austin GDC this year, Merrifield explained why this company runs the virtual world, with its grin-inducing stories and charming demeanor. Basically, it's not because they were waiting for Disney to buy them—rather, that they love the kids who play the game, and want to serve them above all. That's pretty ambassadorial, selfless, and above all, genuine, which is why this grassroots creator has gone on to so much success.

6
♣

PAULINE JACQUEY

UBISOFT

No third party has understood Nintendo's hardware and target demographic as well as the Paris-headquartered Ubisoft, a key to its recent success. From the more whimsical RABBIDS through the tremendously popular IMAGINE and PETZ series, all of which have sold multiple millions, Ubisoft's casual titles—while perhaps not of interest to many of the core readers of *Game Developer*—have distinguished themselves by being well-made non-shovelware, and productions that don't try to take advantage of the innocence of the target market. Jacquy, who heads up the casual division at the firm, should take pride in the sales, the marketing broadening, and most of all, the way the company has gone about it.

5
♣

JIM GREER

KONGREGATE

Ex-Pogo staffer Greer formed Kongregate with a simple idea—to take the Pogo "stickiness" and bring it to bear on free, ad-supported Flash-based web games. For many in the regular game industry, it might be a little scary how competent many of these titles are, and the layers of Web 2.0 chat, rating, and achievements make the site even more intriguing. Of course, the monetization is relatively unproven for end users, at least in terms of making a living easily, but Kongregate is a key site in the democratization of gaming, and that's a wonderful thing.

4
♣

RUSTY BUCHERT

SONY SANTA MONICA

Some of the most interesting creative endeavors out of Sony recently have been birthed from Sony's Santa Monica studio, and veteran Buchert has helped facilitate a lot of these more esoteric first-party wonders. From indie console breakthrough EVERYDAY SHOOTER through ThatGameCompany's upcoming FLOWER to the unique interactive demo-scene project LINGER IN SHADOWS, Buchert is facilitating some of the titles that are truly giving the PlayStation Network its personality—and showcasing how a first-party should juice up its line-up with truly different productions.

3
♣

CLIFF BLESZINSKI

EPIC

Most critics and developers agree that the game industry needs "faces" in order to be accepted by the mainstream in the way movies are. GEARS OF WAR's Cliff Bleszinski is such a face—personable, perceptive, and with a successful enough game series to wind up on television, but also with the intelligence and care for the industry to actually say something interesting once he gets there. Bleszinski may be a little smoother than the average INTJ game developer, but for the future profile of the game industry, isn't that a good thing?

2
♣

TETSUYA MIZUCUCHI

Q ENTERTAINMENT

Very few Japanese creators have a distinctive style and a cross-media bent—perhaps only Masaya Matsuura has a similar profile—and Q's Mizuguchi is notable for his ability and his firm's ability to take abstract puzzle and action concepts (see: LUMINES, REZ) and make them resonate worldwide. His devotion to synaesthesia has, in some ways, prefigured the music game boom, and most of all, his ability to skip from games to elsewhere—whether it be the holographic AI Gore he created for Live Earth or his virtual Genki Rockets pop stars—makes him a cultural figure beyond the obvious game creator-geek.

ENTREPRENEURS

These individuals making it into the Entrepreneurs list are businessmen, sure, but you have to be a little more than a penny-pusher to be on our listing.

To make it into this group, one must not simply make money—one must do so in a way that reinvents the company, advances the industry, or flies in the face of convention. From massive franchises through small indies, all of the below honorees have done just that.

A
♥

JOHN RICCIETIELLO

ELECTRONIC ARTS

Riccitiello's return to EA has marked a turning point for the company. It's rare to see a CEO make such smooth, relatively contiguous, but still effective changes to a line-up. It's perhaps even rarer for a CEO that originally came from outside the industry to know so much about the games its company makes. But the firm's maintenance of its top franchises and staff and simultaneous nurturing of well-made potential new ones, from MIRROR'S EDGE through DEAD SPACE, has meant that the company is becoming, surprisingly, less "The Man" and more "The Man You Want To Work For."

K
♥

THE HOUSERS

ROCKSTAR

Sam and Dan Houser understand what very few others in the game business have managed to perfect—that a combination of controversy and well-executed, stylish games add up to sales gold. Sure, one might say that Take-Two division Rockstar Games overeggs the "rebel" card, but GRAND THEFT AUTO IV's massive initial sales—and a robust slate of other franchises, including BULLY, MIDNIGHT CLUB, and MAX PAYNE—mean that the brothers continue to power much of their parent company's buzz and profits.



Canadian-born Mark Rein is vice president and co-founder of Epic Games based in Cary, North Carolina.

Epic's Unreal Engine 3 has won Game Developer Magazine's Best Engine Front Line Award for the past three years, and "Gears of War," the 2006 Game of the Year, sold over 5 million units on Xbox 360 and PC.

Epic recently shipped "Unreal Tournament 3" for PC, PlayStation 3 and Xbox 360. "Gears of War 2" for Xbox 360 is scheduled for release on November 7, 2008.

Upcoming Epic Attended Events:

Game Connection Europe
Lyon, France
November 5-7, 2008

IGDA Leadership Forum
San Francisco, CA
November 13-14, 2008

KGC/Gstar
Seoul, Korea
November 13-16, 2008

D.I.C.E. Summit
Las Vegas, NV
February 18-20, 2009

Please email:
mrein@epicgames.com
for appointments.



Unreal Technology News

by Mark Rein, Epic Games, Inc.

UNREAL ENGINE 3 (SUPER) POWERS DC UNIVERSE ONLINE FOR PC AND PLAYSTATION 3

The following is an excerpt of a story written by John Gaudiosi for www.unrealtechnology.com.

Comic-Con International: San Diego 2008 saw the debut of *DC Universe Online*, the first massively multiplayer online (MMO) game based on a licensed comic book.

Sony Online Entertainment LLC is using Unreal Engine 3 to bring the DC Universe to life as a persistent world for PC and PlayStation 3, and its creative team recently opened up about its experience with the engine.



Jim Lee, comic book legend and *DC Universe Online* executive creative director

"Early on, we did testing with different engines to see what would be best for developing this game, and Unreal Engine 3 was awesome because it gave us the tools to build this world," said comic book legend Jim Lee, executive creative director for *DC Universe Online* and artist for DC Comics.

"Unreal wasn't originally designed to build MMOs, so there was tweaking that had to be done early on, but the results have been terrific."

Lee said the team of 70 at SOE Austin has been able to do remarkable things with lighting, specular effects and the building of iconic DC Universe characters.

"The Unreal Engine has been great because it's allowed us to change things very quickly and make modifications like adding darkness to the shadows and other

things like that," explained Lee.

John Blakely, vice president of product development for SOE Austin, concurred.

"Unreal Engine 3 allowed us to start playing the game from the start," said Blakely. "It gave us a tool base to get art into the game quickly, allowing designers to test out and prototype what kind of gameplay we were going to be developing.

"It's been really useful because we're taking an action game mechanic and blending that with some of the online persistent world elements. To be able to prototype this from day one, learn what's important, and then build the infrastructure of the game knowing full well what to expect has been invaluable."

Blakely said the Unreal Engine's toolset gave his artists all the pieces they needed from the start, which allowed SOE Austin to staff up quickly on this project.

"Usually with an MMO game, building the content – the environment and characters and all the things you need to create – is the long-lead item," added Blakely.

"We got out of the gate quickly without a lot of disruption, which enabled us to focus on the other fundamental pieces we needed to build, while leaving the artists to work in their world undisturbed."

Comic book fans know that Lee has a very unique style to his art. Chris Cao, studio creative director for *DC Universe Online*, said Unreal Engine 3 gave the team a jump forward on a lot of the graphics and the ability to render these characters well yet cost effectively.

"Jim's characters have a very cut feel," said Cao. "Our DC characters have a distinct quality, and we didn't want them to look soft or cartoony. Unreal Engine helped us do that."

DC Universe Online features approximately 150 characters from the DC Comics mythology, ranging from well-known figures like Batman and The Flash to lesser known characters like Hawkman.

The entire world of the DC Universe will be there for gamers to discover over time, and with *DC Universe Online* open to millions of gamers, there will be many ways to explore.



For UE3 licensing inquiries email:
licensing@epicgames.com

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

WWW.EPICGAMES.COM

Q **ROD HUMBLE**
THE SIMS STUDIO/EA

Recently appointed the head of the Sims Label at EA, Humble has notable street cred with developers, having created his own art-games such as THE MARRIAGE in his spare time in recent years. But it's the diversification of THE SIMS line that he's now masterminding, and quite apart from the surprisingly sophisticated THE SIMS 3, extensions such as MYSIMS (and the return of SIMCITY to greater console prominence) are showing why the original franchise of "play" is coming full circle in these casual times.

J **RANDY PITCHFORD**
GEARBOX

Make no bones about it, running a successful independent developer is tremendously difficult in today's rapidly stratifying market—and FPS veteran Gearbox, headed by Pitchford, is doing an amazing job of growing and expanding its company. Starting with conversions or new versions of other companies' titles, Gearbox has created the developer-owned BROTHERS IN ARMS franchise, and is now diversifying further, thanks to games such as BORDERLANDS and even a cheeky SAMBA DE AMIGO Wii version. Well-respected by peers and creating games that do well in stores, the company's entrepreneurial spirit seems to be swelling over time.

10 **TED PRICE**
INSOMNIAC

A key second-party Sony developer, what Price and his Southern California staff continue to do, perhaps more so than any other system-exclusive developer, is to iterate and create high-quality experiences on a yearly timeline. The original RESISTANCE was an impressive diversification, and with a much-awaited RESISTANCE 2 out almost as you read this article, and the PSN-exclusive RATCHET & CLANK FUTURE: QUEST FOR BOOTY being one of the first intentionally bite-sized AAA downloadable titles, and a North Carolina studio expansion planned for next year, the company seems in rude health.

9 **JONTY BARNES**
BUNGIE

Splitting from the mothership is a gutsy move, but if you have one of the biggest selling current console generation titles, as Bungie does with HALO 3, then setting up separately from Microsoft isn't such a stretch. Helping them do so is Lionhead veteran and production head Barnes, and with the newly announced HALO 3: RECON being practically a mini-team side project for the now multi-project developer, we eagerly await the firm's continued evocation of its independent spirit.

8 **MAKOTO IWAI**
NAMCO BANDAI GAMES AMERICA

The Japanese-headquartered Namco Bandai is reinventing itself in the U.S., with internal studios, externally developed Western-produced titles, and the whole nine yards. Issues with the crumbling HELLGATE: LONDON notwithstanding, the company has been making some interesting moves recently, and much of it has to do with EVP and COO Makoto Iwai (previously development director), and his so-called "samurai mentality." Iwai has been shaking up development teams, and reforming the company from the inside—an impressive thing to see.

7 **BRAD WARDELL**
STARDOCK

Is the hardcore PC game scene the new face of independent games? Some would say so, and Stardock's Wardell is one of the up-and-comers, thanks to a rich history with the GALACTIC

CIVILIZATIONS series, and the Stardock-backed SINS OF A SOLAR EMPIRE reaching a super-impressive 500,000 units. Add to that the Gamer's Bill Of Rights and his firm's Impulse digital distribution system, and the rise of the independents continues, even beyond the obvious.

6 **SATOSHI TAJIRI**
GAME FREAK

POKEMON is a financial powerhouse. A new proper title in the series is guaranteed to sell at least a million within a few weeks, and the game has essentially refined, if not started, a complete game genre—one that has brought success to even its imitators, in lesser degrees. Game Freak's Satoshi Tajiri makes the list because his company has managed to deliver time and time again what the customers are looking for, expanding the dynasty to astronomical heights—this year is no exception, with POKEMON PLATINUM a Japanese smash, and POKEMON DIAMOND/PEARL having sold around 15 million units. POKEMON is the giant hit that no one ever thinks about—and that makes it all the more powerful.

5 **JOHN BAEZ**
THE BEHEMOTH

John Baez has guided tiny San Diego-based indie and ALIEN HOMINID and CASTLE CRASHERS creator The Behemoth as a company through thick and thin, using distinctly unconventional business tactics. How so? By making original action figures, selling t-shirts, going to expos, and basically hustling all day long to promote the company. It is through strength of will that the company battled to release CASTLE CRASHERS to huge success—over 350,000 units on Xbox Live Arcade in a tremendously short period of time—and deliver a massive lesson on what it takes for independent developers to be heard in today's market.

4 **REGGIE FILS-AIME**
NINTENDO OF AMERICA

While Satoru Iwata appears elsewhere in the Top Deck, Nintendo of America head Fils-Aime appears in the Entrepreneur section for one simple reason—he's helped make the very Japanese company successful in the West through smart marketing and intelligent use of the amazing concepts created out of Nintendo HQ. Using Nicole Kidman to advertise the Nintendo DS in gossip magazines is hardly a conventional tactic for your average game hardware firm, but it's been all-encompassing moves like this that have helped Reggie convert the masses to Nintendo.

3 **SHINICHI SUZUKI**
ATLUS

As the game market expands, we're seeing increasing amounts of smart entrepreneurship within those niches—and import gaming is one of the more beloved of those. Atlus, a relatively small Japanese firm, has been rapidly expanding its Western translation of Eastern titles, with some significant success. Quite apart from its own PERSONA series, which is increasingly critically acclaimed in the West, Suzuki and the Atlus U.S. team are licensing from small Japanese developers, bringing valid forms such as the strategy RPG and the surgery simulator to wider audiences, and uniting the world along the way.

2 **CHRIS SATCHELL**
MICROSOFT XNA

One of the signs of entrepreneurship is opening up new avenues of creativity and revenue creation, and Satchell's endgame—using the Microsoft XNA Studio tools to have "bedroom programmers" create XNA Community Games across Xbox 360, PC, and even Zune—is a

What tools are you using?



The Ultimate MMO Toolset



HeroEngine™ is a workflow revolution. Did World of Warcraft alter the way you think about gameplay? HeroEngine will change the way you think about development.

Imagine building your game where your work is live in the game world the instant you create it. Where you work online with your teammates around the world. Where the toolset enables teamwork and collaboration in real-time. Where you can jump into the game at any time while it's being built.

Working on your game is almost as fun as playing it.

This is HeroEngine. A complete client and server solution for MMO games with the best tools for every member of your team. HeroEngine optimizes your development – it's a faster, better system that will save you time and money and let you focus on what you do best.

See it for yourself at heroengine.com or email bizdev@heroengine.com and ask for a demo. See why the leading studios in the world choose HeroEngine.

Isn't it time your tools grew up?

HeroEngine is a trademark of Simutronics Corp., all rights reserved.
Other trademarks property of their respective holders.



www.heroengine.com

massive step forward for user-created content on consoles. The fact you can make money off your Community Games releases, too, makes it even closer to some of the more dynamic game ecosystems out there right now—such as Apple's App Store. Also, with XNA's professional development on the Xbox incredibly robustly supported—that's down to Satchell and team, too.

PROGRESSIVES

Folks making it into the Progressives list are largely game designers who stood out from the crowd of traditional creators. Obvious? Sure.

But how did they differentiate? Well, to make it into this section of the list, you have to implement brand new ideas in game development, perhaps branching in unexpected yet compelling directions. Alternatively, you can simply make what already works, work a whole lot better.

A DR. RAY MUZYKA & DR. GREG ZESCHUK EA BIOWARE

Over the past decade, Ray Muzyka's and Greg Zeschuk's BioWare has turned the previously niche genre of Western computer RPGs into a mass-market phenomena. From the groundbreaking *BALDUR'S GATE* to *STAR WARS: KNIGHTS OF THE OLD REPUBLIC* and beyond, BioWare gets it. And now that *MASS EFFECT* is enjoying impressive sales on the Xbox 360, *SONIC CHRONICLES* has re-interpreted an icon, and "that MMO" has been announced—well, the multi-hundred million-dollar purchase of BioWare Pandemic by EA is vindicated, right?

K JASON WEST INFINITY WARD/ACTIVISION

When Activision decided that it needed a game to compete with EA's *MEDAL OF HONOR*, it turned to the developers at Infinity Ward, many of whom had already cut their teeth on the acclaimed series of WWII shooters. The result was *CALL OF DUTY*, and within an impressively short period of time it has become a cornerstone franchise for the publisher. With *CALL OF DUTY 4*, one of the highlights of last holiday season, Infinity Ward stepped away from the World War II setting and crafted a shooter that was as sophisticated in its narrative as it was in its software engineering. In an already overcrowded genre, the game has sold more than 10 million copies—and that's why CTO West and his compatriots deserve honoring.

Q CHRIS CHUNG NCSOFT

The MMO market is a hotbed of exuberant investment and wildly optimistic publicity. The undeniable success of *WORLD OF WARCRAFT* has convinced many companies that a new gold rush is on with nearly limitless opportunities to print money. The reality is a little more complex, and it's been interesting to see how Korean powerhouse NCSoft has dealt with it. Arguably its greatest Western success has been with Seattle-based *ArenaNet* and *GUILD WARS*, and that's why, after some miss-steps with *TABULA RASA*, Chris Chung is essentially running Western business for the firm. So with a sequel coming, and a pledge to flourish in the subscription MMO space, Chung seems perfectly set up to execute on a portfolio of competitive MMOs—something not many others can say.

J COICHI SUDA GRASSHOPPER MANUFACTURE

2008 will go down as the year punk broke, at least as far as upstart Japanese developer Grasshopper Manufacture is concerned. When it was announced in August that Electronic Arts was publishing a new game from Suda, produced by *RESIDENT EVIL* supremo Shinji Mikami and co-developed with Mizuguchi's Q Entertainment, we couldn't help but imagine the sound of Grasshopper's buzzing, three chord rave-ups transformed into the chromium roar of a new supergroup. And with the critically beloved *NO MORE HEROES* getting a Wii sequel, the bold, amusing rebel stylistic strokes are apparently coming to a much larger worldwide audience—all for the good.

10 EMIL PAGLIARULO BETHESDA SOFTWORKS

The *FALLOUT* series has a long history of dealing with the weight of fan expectations. Now that the franchise has transitioned to a new developer in Bethesda, lead writer and designer Emil Pagliarulo has to walk a fine line between staying true to *FALLOUT*'s post-apocalyptic roots, and making its *OBLIVION*-esque open-world RPG evolution accessible to a console audience. By what we've seen so far, the Looking Glass school of game design graduate has the chops to do it, making it one of holiday 2008's key games.

9 JONATHAN SMITH TRAVELLER'S TALES

Working with licensed properties has long been a fact of life within the video game industry. In fact, it could be argued that licensed properties provide the daily working capital needed to keep the business running. Still, in this sometimes unglamorous sector, there are developers like *Traveller's Tales* which does outstanding work that far exceeds audience expectations. Titles like *LEGO STAR WARS* (and *INDIANA JONES* and *BATMAN* and ...) are witty and delightful. Smith's work may seem like an unholy promotional mélange, but it takes deep talent and craftsmanship to bring such irresistibly fun games to the whole family.

8 ATSUSHI INABA PLATINUMGAMES

Realizing that it's better to be The Man than to work for him, Atsushi Inaba and the *OKAMI* creators at Clover Studios left the relative security of Capcom for the wilds of independence. Of course, the resulting PlatinumGames studio, which includes Hideki Kamiya, Shinji Mikami, Shigenori Nishikawa, and Hifumi Kouno, is a powerhouse of Japanese development talent so its success is fairly assured—but it's an important statement in the relatively staid Japanese market that major creators can strike out on their own. And the reaction to PlatinumGames' signing of a four game deal with Sega seems to indicate that talent is dictating the terms of the deal, even in Japan, symbolically vital for that territory.

7 RON CARMEL AND KYLE GABLER 2D BOY

The development of the tremendous *WORLD OF GOO*—just-debuted on PC and WiiWare, and at one point the highest ever Metacritic rated game on Nintendo's console—is an inspiration for game designers who believe in the DIY culture that video games are founded on, but find themselves making rote titles in cubicles. Created by ex-Electronic Arts employees Kyle Gabler and Ron Carmel, *WORLD OF GOO* was built on the go with a tiny, essentially two-person team, and a "better, faster, cheaper" ethos that utilized many open source solutions and was free from the money-draining overhead of a physical office. This, ladies and gentlemen, may be the future of short-form gaming.

6 SID MEIER
FIRAXIS

Sid Meier is part of a generation of designers who came of age in a world without video games and whose primary inspirations come from the complexity of board and wargames in the style of SPI or Avalon Hill. And that's no bad thing, coming as it does from a time when games were specifically made for smart people. But Meier has an unerring instinct for fun, and his work is marked by a lovely intersection of whimsy and rigor. Thus, with the creation of CIVILIZATION REVOLUTION, Meier brought that experience to the widest possible audience by designing a game that played to the accessibility of consoles without diluting the central intellectual challenge—easily earning him a place on the Deck this year.

5 TETSUYA NOMURA
SQUARE ENIX

Tetsuya Nomura's character designs—rendered in the spindly style of manga, but imbued with the jumping rhythms of street culture—have increasingly become the visual signature of Square Enix. As skilled as he is at creating art that is cross-cultural in its appeal, it is Nomura's efforts behind the scenes at Square Enix that puts him in the *Game Developer* Top Deck. His work on the KINGDOM HEARTS franchise, the ongoing Compilation of FINAL FANTASY VII project, and the upcoming Fabula Nova Crystallis FINAL FANTASY XIII project, all of which feature interconnected titles across a wide variety of platforms, is an instructive lesson in brand management. It's a fresh approach that upends the standard industry practice of providing linear sequels and instead gives fans multiple points of entry to their favorite game worlds.

4 MICHAEL BOOTH
VALVE SOUTH

Perhaps many are predisposed to liking Valve's LEFT 4 DEAD just on the basis of its zombie apocalypse setting. But more discerning players will appreciate what's going on under the hood. The game's scalable AI technology promises to generate a dynamic experience that adjusts to player performance on the fly. Taking on many of the tasks that would traditionally be hard-scripted by designers, LEFT 4 DEAD's AI stage manages the player experience from moment to moment, providing lulls and crescendos to the action that are unique to each instance of the game. As the creator of LEFT 4 DEAD's "AI Director," Booth can justly be proud of something both sophisticated and truly next-gen in terms of visceral, cooperative experience.

3 DYLAN FITTERER
INVISIBLE HANDLEBAR

AUDIOSURF is a game with a simple concept and an elegant execution. Combine a music visualizer with a puzzle game, mix in a deep scoring system and the player's personal music library, and the result is a game with infinite replayability. Almost the perfect software toy, AUDIOSURF provides a deep synesthetic rush as its visual action synchronizes to your favorite music with an uncanny precision, and emailed high-scores and smart online synchronization make it even more tempting. That AUDIOSURF is largely the result of Dylan Fitterer's singular efforts makes its triumph all the more sweet—another indie success story.

2 JAMES NORTH-HEARN
SUMO DIGITAL

North-Hearn is now running much of Foundation9's development, but we particularly wanted to highlight and recognize the

beautifully curatorial spirit of the UK's Sumo Digital, his original development home. Taking products from Sega (SUPERSTARS TENNIS) through Namco (NEW INTERNATIONAL TRACK & FIELD), the level of smartly executed fan-service in the games seems to outdo even what the original companies might have intended. Often, remakes or updates are less, well, caring—and it's beautiful to see a European company taking great care of Japanese franchises from some of the all-time greats, in a relatively under the radar manner, too.

JOKERS

As with any deck of cards, we need a couple of jokers in the pack. We've picked a couple of creators whom we adore, and are significant creative forces in their own right.

But they're both folks who make us grin in different ways, and play the fool—largely intentionally—while advancing the biz at the same time. This isn't entirely the booby prize!



DENIS DYACK
SILICON KNIGHTS

Denis seems to have gotten a bit of a reputation as class clown of late, thanks to his forthright views on subjects as wide as the one console future, marketing games, and, of course, Unreal Engine. But what gets him on this list is his unfortunate trolling of famously the firey game forum NeoGaf. Dyack stated his opinion that the forum is hurting both society and the game industry, and challenged forum members who had yet to play his then-upcoming TOO HUMAN to voice their hatred, to stand and be counted. Then when the game came out, there would be egg on their face. The game's middling critical reception, combined with an angered mass of forum flammers, didn't help to prove him right. What's unfortunate is that some of his more outlandish statements have masked the largely excellent points he has on a variety of subjects, from journalism to the nature of flow.

PETER MOLYNEUX
LIONHEAD

In general, there's the world-changing game Peter Molyneux talks up prior to a release, and the eventual game you get. While the result is always satisfying, there's generally a rather amusing disconnect there. The designers at Molyneux's studio have set out a host of ambitious goals for FABLE 2 (even as Molyneux himself is more careful to manage our expectations this time)—and the entire experience will hinge on getting the AI right. From what we've seen, the underlying logic that drives contextual choices in FABLE 2's dynamic does indeed open up some new modes of expression in game design. But can any Molyneux-developed game ever match up to the expectations laid out for it by its creator, prior to its release? Perhaps a happy medium behind the hype and reality is what makes it work, but with Peter already starting to hint at revealing another project, even before this one is out, the delightfully charismatic Molyneux circus continues. ✨

ACCESS IT IN REAL TIME

game developer

DIGITAL EDITION



Game Developer magazine brings you postmortems, product reviews, and strategies for creating innovative, successful games from industry leaders and game development experts.

www.GDMAG.com

THINK
SERVICES
A DIVISION OF UNITED BUSINESS MEDIA LLC



Best Practices for Error Reporting and Value Editing Systems for In-House Tools

I'M STILL BAFFLED AT THE AMOUNT OF ASSETS NEEDED to make a next-gen game. It seems like only a couple of years have passed since a game studio only had one artist who was using a simple paint application to draw all the background images and sprites required for the project. Back then the only tool you had to worry about was the compiler for the code. Comparatively, the current situation looks quite different. We certainly do need a lot more of everything. We have more computational power and memory at our disposal, but

consumers expect even more than what we have to work with. They want a consistent game world, awesome and very detailed graphics, gorgeous sound, a compelling story, and interesting characters to interact with. All this basically means a huge amount of assets.

Even though some of the data can be generated automatically, like light maps, visibility sets and so on, it still comes down to artists and designers to create most of the assets by hand. This means that in-house tools and editors are much more important these days, because if they break or if they are

OLIVER FRANZKE has been coding in games and graphics research for a number of years in various places all over the world. Currently he is working as an engineer in San Francisco. Email him at ofranzke@gdmag.com.



inefficient, they will slow down the whole project, which will waste a lot of time and money.

Therefore it is important for us to improve the quality of our tools and editors. This doesn't necessarily mean you have to rewrite a lot of code, because even some simple changes can improve your applications quite a bit. In this article I want to share a few ideas about how to make game development tools better with very little effort. For that I'm going to concentrate on some possible improvements in the error reporting and value editing systems. The hints presented here are by no means revolutionary, but when implemented with care they will help reduce the turnaround time for asset creation, which means developers can have more iterations in the same period of time.

WATCHING TOOLS GROW

I'm pretty sure that every tools programmer knows the situation: You want to overcome a slight inefficiency in the pipeline, like a manual data conversion, by quickly writing a small tool that will do the task automatically. It is meant to only do this one thing and nothing else. The application is implemented in no time, and the interface is going to be both clean and easy to use. Job done!

You forget all about it until the format of the source data gets updated, which makes the problem slightly more complicated (for example, there are now two ways to convert the data). Well no big deal, you think. All you have to do is to extend your tool and the day is saved again.

After a while, you realize your neat little tool has evolved into this huge application with dozens of modules and even more different modes and therefore is probably very difficult to extend and debug. What makes the situation even worse

is the fact that by this time, the tool is going to be an essential part of the content pipeline, so refactoring the code must be done very carefully in order to keep the functionality as it is without breaking the entire pipeline. Unfortunately, very often there is just no time to do the refactoring at all, and so the tool will haunt you until the end of the project—and sometimes even beyond.

What happened? The answer is quite simple really. When you originally implemented the tool, there was no way to properly plan the architecture of the application so that it would incorporate all the future changes as well. Even when you try to anticipate all possible usages, there are always going to be

adjustments and unforeseeable problems along the way. That is just the way it is, especially for the three-year or longer development cycle of a next-gen game.

It's time to think about what can be done to at least improve the situation a little bit.

INFORMATION OVERKILL

One specific problem I see over and over again is that tools don't handle error reporting and logging very well. For example consider the pseudo code snippet in Listing 1 that processes all materials of a given input mesh (such as a translation into a platform-specific format). If the conversion causes a problem, then a message is written to the console and sent into a log file.

Originally this might be sufficient, because the process probably won't fail very often due to the code being written properly to handle all materials known at that time. But what will happen if the situation changes? For example, let's assume that your engine gets optimized, and in order to reach maximum performance the materials have to be tagged by the artists with certain flags (static or dynamic lighting and transparency). The old materials will still work, but they won't render as fast as they could. Initially, this issue probably isn't critical, but since it should be addressed at some point, we could add a warning to the log file if the tool can't find the newly defined flags.

Of course, it's good to add diagnostics like that, but on the other hand it will generate a lot of output since every material in every mesh could potentially add this kind of message to the report. If other parts of the conversion process generate a similar amount of output, then you'll soon end up with log files that are thousands of lines long. These files are simply incomprehensible and nobody will ever really look at them until something breaks—and that's where things go horribly wrong.

Artists and designers usually don't enjoy data mining very much (to be honest, I don't like it either), but now they are confronted with a massive log file that contains this one bit of information they need to do their job, in between thousands of lines that are not relevant to them at all. You can't blame them if they send you the report via email with a request to help them figure out what went wrong.

While the log file contains a lot of important information (at least that is how it should be) some messages are more urgent than others. Even though the materials might need the new tags at some point, it's distracting and annoying to read about it when the conversion failed because of something completely unrelated.

LEVELING UP

It's quite easy to improve the information overkill problem significantly by adding error level support to the report system, which makes it possible to prioritize the messages. Obviously, an error is more important than a warning and therefore it should have a higher priority.

Listing 2 shows our simple material conversion example with error level support; as you can see, not much has changed. The only difference is that the status of the conversion is now directly used by the report system to set the importance of the log entries instead of being defined implicitly through the text of the message.

LISTING 1

Converting All Materials of a Mesh

```
foreach( Material mat in Mesh ) {
    if( ConvertMaterial(mat) != OK )
        Report("Problems while converting " +
            mat.name);
}
```

LISTING 2

Material Conversion With Prioritized Messages

```
foreach( Material mat in Mesh ) {
    Result = ConvertMaterial(mat);
    Report(Result, mat.name);
}
```

Prioritized messages can then be used to format the final error report so that it will be much easier and faster to spot the important issues. For example, it makes sense to group the different priorities together as well as to sort them by their importance. This way, the errors will always be at the top of the log file followed by the messages with lower priority.

To see the advantage of this approach, compare the two conversion reports in Figures 1A and 1B. The former represents an imaginary (and extremely simplified) log file without error levels, while the latter shows the same report with prioritized messages. Spotting the error in Figure 1B is much easier. Now imagine the difference in a real log file!

Adding error levels to tools isn't much work at all and is already going to have a big effect on the efficiency of your pipeline. The funny (or should I say ironic) thing is that error levels have been supported by IDEs like Visual Studio for quite some time now. I certainly don't want to miss this feature anymore, so why not help our users in the same way? But before you run off to change all your tools, let's have another quick thought about the report system to see whether we can improve it even further.

BE SELECTIVE

While your tool is evolving over time, it probably has to deal with more and more different things. New features and modes are added, and before you know it, your little baby is all grown up and quite big.

To illustrate this transition, imagine the material conversion example again. Let's say the application was so popular (because of the awesome log files) that it got "promoted" and is now responsible for processing entire levels. Now it has to manage a lot of new things—spawn points, particle systems, sounds, and so on.

Since all the different modules are adding messages to the error report, the generated log file is soon going to be very clunky again. Something has to be done to avoid a new information overkill problem. The following simple observation will help us to get back on track: When concentrating on one specific problem artists are probably not really interested in the output of unrelated systems. In fact in most cases, they will only need the information of the module they are currently working with.

Fortunately, it's quite easy to translate this observation directly into code by adding a report threshold that can be defined individually for each module. This simple trick will make it possible for users to be selective about what kind of information the error report should contain and which log entries can be ignored. For example, while being interested in all the messages regarding the material conversion, an artist can now turn off warnings of other modules, which will make it much easier for her to concentrate on her actual task. Figure 1C shows our simple example log file using error thresholds to ignore all messages that are unrelated to materials. The report is significantly more concise and therefore easier to comprehend.

It's remarkable that although a good error report system isn't difficult to write, hardly anyone ever does it. This is especially true since it has no dependencies on other systems and can easily be implemented as a library and

```
Material "Bricks" is old
Degenerated triangle found in mesh "Door"
Material "Wood" is old
Degenerated triangle found in mesh "Window"
Material "Window" is invalid
Material "Carpet" is old
```

FIGURE 1A Simple imaginary log file.

```
*Errors:
Material "Window" is invalid

*Warnings:
Material "Bricks" is old
Degenerated triangle found in mesh "Door"
Material "Wood" is old
Degenerated triangle found in mesh "Window"
Material "Carpet" is old
```

FIGURE 1B Log file with error levels.

```
*Errors [Material]:
Material "Window" is invalid

*Warnings [Material]:
Material "Bricks" is old
Material "Wood" is old
Material "Carpet" is old
```

FIGURE 1C Log file with error levels per module.

therefore reused in other projects, too. (Even your game engine could use it!)

One thing to keep in mind while implementing a system like this is to make sure the interface is as simple as possible, because if it's more difficult than using `printf`, then people might soon fall back to old habits, and the whole cycle starts from the beginning.

"ERROR: PROCESS SUCCESSFUL!"

Hopefully I showed you by now that even some simple changes to the code can have a positive influence on the quality of your tools and therefore will make the users of your applications happier. Following this strategy, there is more that can be done to improve the workflow.

Another general problem with error reports is that the individual messages are often either too short and ambiguous (like "Error while <doing whatever>") or are not written in a way that allows users to immediately understand and react. I don't know how often I've heard things like, "The message says the material is invalid, which usually means that the converter couldn't find the shader or one of the textures."

It's true that less is often more, but this mantra definitely does not apply to problem descriptions. Leaving the users in the dark about what actually went wrong is going to cost your



project a lot of time, because instead of being able to fix the problem immediately, the users are now blocked until they can find a solution.

Apart from having clearer descriptions, the messages can be improved significantly by adding an additional wiki link (or something similar) to the log entry. The target page can then be used to describe the problem in more depth. Make sure to also mention possible causes and their solutions as well as contact information of the programmer who wrote the code just in case the proposed fix didn't work.

WHAT ELSE CAN BE DONE?

With the information overkill problem solved, we can start thinking about whether and how other parts of the content creation pipeline can be enhanced with similar tricks. Just like optimizing code, it makes a lot of sense to spend more time improving functions that are frequently called, because if you can make them better, then the whole program will benefit from it. The same strategy can (and should) be applied in the context of content creation. It's a good idea to identify and improve those low-level inefficiencies.

To get started, we should have a closer look at what artists and designers are doing most of time while working with editors and tools. When analyzing common workflows, it's not hard to figure out that the setup and tweaking of values is a very important part of every asset creation application. This might seem like a trivial fact at first, but if you take a closer look at most editors and tools,

you'll notice there is a lot of potential for improvement.

A good way to reduce the turnaround time for asset creation is to increase the responsiveness of editors by providing real-time feedback for all value changes. Traditionally, tools are developed in a dialog-based fashion, which means that if the user wants to tweak the value of a property, he or she usually has to click a button, which will then open an editor dialog window for the specific property type. After specifying the new value, the user can accept the changes by pressing "OK," at which point the change will be applied and the tweak becomes visible. If the user isn't quite satisfied with the result, the whole process has to be repeated over and over again.

The best example for this kind of modal editing is the color picking dialog, which is still used in a lot of applications—level editors for example. To see how the color

you've just selected actually looks in the context of your work, you have to click "OK," which then makes the dialog disappear so that another edit is required for you to reopen it. That's annoying!

Dialog-based editing is usually implemented as shown in Listing 3: a simplified example of an event handler method that is called when the user wants to change a color value. The editor dialog is created and launched using the blocking call ShowModal. Once the user has changed and accepted the new value, the color is updated.

NON-MODAL EDITING

In an ideal world where every change magically leads to the perfect result, dialog-based editing wouldn't be a problem. In reality, though, there will always be a lot of iterations to find the best value for a property.

Consequently, it's much better to use a non-modal editing approach, which basically means that the user is no longer required to manually accept a tweak, but that the selected value is applied immediately. The change happens in real time, making the responsiveness of your editors much better.

The nice thing about non-modal editing is that there's no reason to limit it to simple types like numbers, vectors, and colors, but that it can easily be used for all other kinds of properties like textures, models—you name it! For example, instead of using the default explorer-style file selection dialog, it's possible to show a simple list, which will immediately update the scene every time the user picks a new file.

Fortunately, it's quite simple to support this method of value editing by replacing all standard dialogs with an equivalent non-modal version. The only difference is that the new dialog needs to know which value it is attached to so it can directly apply all changes made by the user. Listing 4 shows the same color editing example as before with support for non-modal editing. The code hasn't changed much apart from the fact that the editor is created with a reference to the color and that the dialog is launched with a call to the non-blocking method Show.

The only caveat with the non-modal approach is that you have to be careful when deleting objects that still have editors connected to them. Make sure to close all editor dialog boxes so that a user can't change values that don't exist anymore.

CLEANING THE GEARBOX

In the last few years, we've had to learn that game development is a difficult balancing act between long-term planning and the ability to react to certain inevitable issues in an agile way. Similarly, it's very important to think about the content pipeline and the overall workflow on a big scale, but without losing track of all those little problems. They might not seem like a big deal at first, but small inefficiencies accumulate over time and eventually could slow you down.

In the end it doesn't really matter if your gearbox isn't working well because of a cog that won't fit or because of dirt that's creating friction. Everyone who works with motorized engines will probably confirm that cleaning an engine sometimes does wonders, even though it's conceptually such a simple thing to do. I think the same is true for software development: Removing the dirt between the virtual gears can lead to big improvements. ❖

LISTING 3 Dialog Based Color Editing

```
void OnChangeColor( ColorVal col ) {
    ColorEditor edit = new ColorEditor();
    // Wait for the user to click on "Ok"
    if( edit.ShowModal() == OK )
        col = edit.color;
}
```

LISTING 4 Non-Modal Color Editing

```
void OnChangeColor( ColorVal col ) {
    ColorEditor edit;
    edit = new ColorEditor(col);
    // Launch the editor without
    // blocking the main thread
    edit.Show();
}
```

JOIN THE REVOLUTION!



FIRAXIS WANTS YOU!

SENIOR MODELERS - ANIMATORS - ENVIRONMENT ARTISTS
CONCEPT ARTISTS - UNREAL LEVEL DESIGNERS - SENIOR GRAPHICS ENGINEERS
SENIOR NETWORK ENGINEERS - INTERFACE ENGINEER - UNREAL PROGRAMMER
PS3 ENGINEERS - PRODUCERS - ASSOCIATE PRODUCERS - AND MORE

WWW.FIRAXIS.COM



© 2005-2006 Take-Two Interactive Software, Inc., and its subsidiaries. Sid Meier's Civilization Revolution, Civ. Civilization, 2K Games, the 2K Games logo, Firaxis Games, the Firaxis Games logo and Take-Two Interactive Software are all trademarks and/or registered trademarks of Take-Two Interactive Software, Inc. All rights reserved.



CERTAIN AFFINITY'S

Age of BOOBY™

MAX HOBERMAN
is the president and
creative director and
holder of all important-
sounding titles at
Certain Affinity, Inc., an
independent developer
based in Austin, Texas.
In a previous life he
was the multiplayer
and online lead at
Bungie. Email him at
mhoberman@gdmag.com.

"IT WAS THE BEST OF TIMES, IT WAS THE WORST OF TIMES, IT WAS THE AGE OF BOOBY ..."

I admit, I've just been waiting to use that line somewhere. But it's a truly appropriate introduction to a postmortem for our casual downloadable RTS, AGE OF BOOBY.

AGE OF BOOBY started life as a paper design for a real-time strategy game based loosely off of *Settlers of Catan*, one of my favorite board games. The inspiration for the pirate theme was, truth be told, the long-running April Fools joke at Bungie (from which many Certain Affinity developers splintered) about a game in development called "Pimps at Sea." I admit I had a hand in that joke, but it wasn't until years later when the bug hit me to imagine *Catan* in real time that the idea of a pirate theme was finally explored. One weekend a friend—Stefan Sinclair (currently a programmer at Certain Affinity)—and I put quill to paper, and the idea for the game was born.

Fast forward a bit to when Certain Affinity was just finishing up our first project, a multiplayer map pack for HALO 2. While we scrambled to figure out what to work on next, we were also evaluating engines for our own purposes. A friend at Microsoft recommended we try out XNA. Two of our programmers, Paul Isaac and Peter Carter, dove right in and started working on a side-scroller for the PC as a means of checking it out. They had it up and running so quickly that I got inspired and suggested we prototype an old game idea. I dug up the design doc for the game I affectionately called "Pirate's Booty" and gave it, and a quick lesson in *Settlers of Catan*, to David Bowman, one of our design leads. The concept revolved around two key elements: accessibility and depth. I wanted a game that I could play with my wife, who doesn't play many games (but adores *Settlers*);

he wanted one he could play with his children. But we also wanted a game we could play together.

David took this foundation and worked it into a playable board game, using cut out bits of paper. This took some imagination but it helped him get his head around the idea, work out a few kinks, and then explain it to others. In the meantime our programmers began creating the foundation for the world: land and water, moving ships, etc. As the artists wrapped up their HALO map work they started concepting and modeling, and in six weeks we had a playable, networked prototype that looked and played remarkably like the final game.

Once we had sold ourselves on the concept we put together a paper pitch document to help shop the game around. We settled (somewhat arbitrarily) on an 8-player cap and determined that we'd support 4-way split screen. We wanted the host of a game to have tons of configurable options and for players to have the ability to create and share their own maps. We also wanted to create an enjoyable online social experience, and so settled on features like a party system and intelligent matchmaking-features that most full-size games don't have. We whipped up plans for these features like nobody's business, and then, design doc, demo, and a certain amount of trepidation in hand, we set out to find someone willing to fund our little pirate adventure.

I'll save the details of the actual publisher pitch for another story, but suffice it to say that two representatives from Capcom eventually flew to Austin, played one multiplayer game, and were hooked. We went to California and got a game going with the rest of their team and left that same day with the high-level points of an agreement worked out. AGE OF BOOBY

GAME DATA

DEVELOPER:

Certain Affinity

PUBLISHER:

Capcom Entertainment

RELEASE DATE:

October, 2008

NUMBER OF DEVELOPERS

(XBOX 360 LEAD SKU):

5 full time, 9 shared, 3 part-time contractors

LENGTH OF DEVELOPMENT:

13 months: 2 months preproduction,
11 months production

PLATFORM:

Xbox 360 (separate developers for PlayStation 3 and Windows versions)

DEVELOPMENT SOFTWARE USED:

Proprietary engine and development tools, 3D Studio Max, Photoshop, Perforce, Illustrator, XUI, Visual Studio

BUGS:

2,147

SOURCE FILES:

457

LINES OF CODE:

117,000



Age of Booty



Portraits of the unbeatable pirate duo, Steve "Fuzz" Pietzsch and Dean "Spork" McCall.

would be coming to Xbox Live Arcade, PlayStation Network, and Windows. We were responsible for developing the Xbox 360 version and planned on contracting out work on the Windows version, while Capcom would handle the PS3 version.

Closing this deal was one of the high points of the project. We had a highly original game, we were writing our own engine, we had a high-profile publisher that cared about quality, and we had an experienced team. Oh, and we had funding, of course. Now we just needed to deliver the final product, and it was just a downloadable game. Easy, right? Of course not. But we had no idea what was in store.

WHAT WENT RIGHT

1 PROTOTYPING. One of the things I most appreciate about *Settlers of Catan* is that a player can learn all the rules and mechanics during a single game. This was the very heart of what we were trying to accomplish with AGE OF BOOTY—nailing a simple, intuitive control scheme was our highest priority. We knew a simple control scheme would be easier if we had a simple world, something flat to minimize the complexity of camera control. A pirate game where the playable area was all water gave us a nice flat playing field, while also allowing us to go 3D and add verticality to the non-playable land. And so partially for this reason, and partially as an homage to board games and pen and paper games, we went with a hex-based 3D environment.

Once we'd nailed the basic controls it was possible to manage resources and sail a single ship around and fight, but there really wasn't much depth to combat, partially due to the simplified control scheme. We wanted combat to happen automatically when ships got near each other, without

requiring micro-management, and this added to the feeling that there wasn't enough to do. Rather than adding complexity to the controls we decided to get team play working. The programmers were skeptical, and their confidence that the game was fun dropped daily, but they got networking and support for two teams with two

players each working. This was the genesis for the next "aha!" moment and the high point for our prototype.

Before this there was a little bit of randomness in combat, but when two ships were fighting, the player with the bigger guns almost always won. The addition of teams allowed you to work together against a single enemy and manipulate the odds. We didn't fully appreciate the depth that this added until we played a few team games, and two of our artists—guys that barely even play games—skunked us designers. We'd been beating them



Concept for the pirate ship with cannon and speed upgrades.

mercilessly on our HALO maps for months, so at first we thought this was a fluke. We tried it again. And again. And again. They were unbeatable. We actually thought that they were cheating until we watched them play. They stuck together like glue, while we got cocky and went off and each did our own thing, and this worked to their advantage time and time again. We tried to emulate this strategy but inevitably bickered and lacked coordination and always ended up losing. We were consumed with a burning passion to beat the artists at our own game, and at that point we knew that we'd discovered something wonderful.

Team play added a ton of depth to the game without compromising the simple controls. This was probably the best possible start for a game. We had created a playable prototype that looked good, that was both easy and fun to play, and that demonstrated the game's core mechanics. The value of this prototype can't be underestimated—not just in helping us pitch the game, but in guiding us later on.

2 TRUE TO FUNDAMENTALS. Once we had this foundation in place we did something else right: we stayed true to it—to ideas such as the simple control scheme and simple mechanics. On many occasions temptation almost got the better of us, but in the end we never compromised simplicity for depth. And so while our prototype had five resource types, the final game has only three (gold, wood, and rum), and towns automatically generate resources instead of players having to put them into production.

We ended up adding depth to the game in other ways that didn't compromise the simple controls. For example, fairly late in development we added what we call "curses", essentially powerups that you acquire by attacking ambient merchant ships. Which curse you picked up was randomly determined, with half of them being fairly useful to turn the tide of battle. Yet you can only carry a single curse at a time, and you use it by targeting a hex and pressing a single dedicated button, so the controls remained very simple.

We did constant user-testing throughout development, usually with friends and anyone else we could find. Capcom



The paper prototype we used to figure out the basic mechanics.

Ship it right
the **first time**



DevSuite

Design it. Plan it. Implement it. Test it. Ship it.

- Integrated ALM solution offering requirements management, project planning, implementation tracking and QA testing
- Provides out-of-the-box implementations for Agile and other standard methodologies
- Facilitates close collaboration of globally distributed team
- Scalable from small teams to thousands of users

TechExcel

www.techexcel.com | 1-800-439-7782

THREE WAYS TO
GET STARTED WITH
ALM

- Free hosted evaluation
- Download a free evaluation copy
- Purchase the team edition, full suite at single product price



A concept for the pirate lair. Note the intentional "board game come to life" feel.

also did its own testing, both internal and external, and would regularly send us feedback. Whenever we ran into a problem with complexity we tackled it head on. Capcom was also a big help here, not only calling out issues that they saw with usability and discoverability, but also suggesting ways to address them. It was a great experience because we'd defined and communicated clear objectives up front and everyone stuck to them throughout.

3 IT'S BETTER TO BE "THE MAN." One of the very best things we did was negotiate a deal that left us entirely in control of the game and the intellectual property. We went in to discussions with this as our highest priority, but to be completely honest I was a bit surprised when we walked out with full IP ownership.

What this did for us in a development sense was provide awesome motivation. There was no way the publisher could dictate our game design, and regardless of how much the company made off of this first title we knew that we were building something that had long-term value. This was a great motivator both for the development team and for business decisions. Simply put, this gave us a reason to invest our own money in the game, above and beyond what we got from the publisher. We couldn't have crammed all of the juicy features we wanted in otherwise; the business model for a single title, and especially for a risky new IP (a downloadable game, at that) simply didn't justify the investment.

4 CUT EARLY, CUT OFTEN. Despite our ambition and our own contribution to the development costs there were still limits on what we could accomplish given the small team, short timeline, still relatively tight budget, and the fact that we were creating our own engine from the ground up. We were constantly aware of this and so were disciplined about making big feature cuts early. Many of these were features that we badly wanted, such as host migration, a centralized map server, and a pirate flag editor. This was always painful, but our experience told us that it was necessary, and so we grit our teeth and forced ourselves to make the cuts.

This discipline paid off in spades further down the road as we started shifting resources off of the project and buckled down to finish the game. Even then we came in later than anticipated due to a number of other factors (more on that later), but without these early cuts we would not have been able to finish the game in a reasonable amount of time or within a reasonable budget.

5 KEEP ON CLOSING. Once we were in the home stretch we had a fairly rigorous closing process, driven first and foremost by a strong push for stability and quality. Paul Isaac, our engineering lead, took the reins here and made the final calls on what was and wasn't acceptable code risk, while designer



A hand-drawn concept for a palm tree hex.

Age of BOOTS



A concept for how to visually represent town upgrade state.

David Bowman and I pushed back on anything that posed a risk to the things that we'd settled on that were fun about the game.

Despite our good intentions this was actually a tricky balancing act, especially because at times we had to fight Capcom, which was pushing for feature additions and improvements. In the end this was resolved by the company giving us an extra (paid) six weeks for polish, something we hadn't asked for and certainly weren't anticipating. In all we went through three major pushes to close the project down, which was certainly more than we anticipated, but the discipline that we exercised by beginning the first of these early and subsequently only taking on tasks that we deemed low risk made a big difference.

WHAT WENT WRONG

1 JUGGLING PROJECTS. Despite our strong start, we made some serious mistakes. Much of this was due to the fact that we split our focus across too many projects and shifted staff onto and off of the game on numerous occasions. I'm guessing this isn't uncommon with a new company, and we certainly understood the risks, but our initially self-funded, pay-our-own-way business model necessitated this; without much working capital it was imperative that we generate as much cash flow as possible, and this made it hard to turn down good opportunities.

On the programming side, for instance, we pulled the original two programmers off of the project while we negotiated the deal, which took several months to finally get signed. We eventually got these two back on it by hiring two replacements to handle LEFT 4 DEAD, our other big project at the time. We hired a third programmer

to assist these two while also contracting out some programming work. We then hired another full-timer and ended our work with the contract programmer. We handed the work the contractor was doing over to one of our LEFT 4 DEAD programmers, who volunteered to complete it in his spare time.

This was the most programmers we had on the project at one time, but then we took on a third project and so pulled two of the full timers and our

"volunteer" off of it, ending up with the same two original programmers to finish the project out. This is just what the programmers went through, but similar staffing changes plagued all of the other disciplines due to us having too many projects on our plate and too few employees.

When we originally spoke with Capcom they asked us to do all three platforms, Xbox 360, PlayStation 3, and Windows. This made us nervous and so they ended up taking on the PS3 version and allowing us to contract out the Windows version. There's no question that this had a big impact on the quality of the final game; given all of our juggling of projects I can't imagine what kind of mess we would have been in if we'd also taken these on.

2 COMMUNICATION IS THE FIRST CASUALTY. The changing assignments and splitting of focus challenged everyone and presented a prioritization nightmare that left no area untouched. One serious consequence was a breakdown in cross-disciplinary communication, something we take a lot of pride in, as evidenced by our completely open pit-style office. Over the course of the project there were numerous disconnects between the perceived state of the game and the actual state of the game, especially between programmers and designers, but sometimes also between artists and programmers. Animation support was one huge casualty of this disconnect, ultimately causing us to cut characters and character animation entirely, after we'd already done the modeling work and re-done the animation work several times.

The hardest hit were the designers, who continued fine-tuning plans for sophisticated

Looking for the best payment service to monetize your virtual world?

Choose Paynova - you'll be up and running in a week.

paynova.com/gamingoffer

paynova

features like matchmaking and party support long after the programmers had already made huge simplifications (and often cuts) to these systems. A combination of lack of attention to the project, poor communication, and wishful thinking led to the design team believing that several features were far more advanced than we were actually able to implement, and they did not find out the reality until very late in the project. These misconceptions inevitably trickled down to Capcom and caused some confusion and even a bit of friction toward the end of the project when they learned about big cuts and changes that had been made months earlier.

3 SINGLE PLAYER IN FLUX. AGE OF BOOTY was prototyped as a multiplayer game but we were always committed to having a strong single player mode. The problem was we couldn't quite agree on what it would be. One camp held firm to the belief that since the game is downloadable and an "arcade" game, we should treat single player like a challenge mode, with lots of rewards and flashy unlockables. The other camp wanted extensive storytelling, persistent characters, linear story progression, etc.

About half way through the project I drove all of the decision makers to lunch, sat them down, and said that no one was leaving the table until we all agreed on our single player direction. About two hours later everyone was finally in agreement. I was shocked. We had to put in some after-hours work to catch up, but we ended up with a single player mode that turned out to be a lot of fun. Unfortunately the direction we settled on came with its own needs for engineering support, level design, and UI, so coming in late with this design created a lot of extra work for the team.



Filthy McScupper, one of the baddies from the single player game.

4 UI IS HALF OF A SMALL GAME. Given many of our backgrounds, one of the silliest-seeming mistakes we made was waiting too long to do the first pass implementation of the game shell interface. All of the factors mentioned in this section added to this: my own attention being pulled from the project, staffing



The concept for our animated rum worker, hard at work.

changes and late additions to (and subtractions from) the team, lack of design for single player, changing social system designs, and more. But all that said, at the end of the day our specs and art ultimately dragged behind and sometimes changed too often, so by the time we implemented a piece of UI it was typically the only pass we had time for.

This had major consequences later on in the project, as we attempted to finish the game and discovered that many systems required complete reworking, sometimes simply because the system as implemented varied so much from the original system design. We broke one of our cardinal rules by not implementing first pass UI early on, and it's not a mistake we'll make again.

5 TOO MANY RULES. Having gone through all of this turmoil we entered the closing process with a solid game and a team pared back to its original members—two programmers, a design lead, a single artist, a production manager who doubled as audio director, and me—to help close things down. Thus began an arduous process where our designers and programmers realized that they'd grossly underestimated the time and energy required to make our engine compliant

what's in a name?

Capcom announced the game as PLUNDER in 2007. At E3 in 2008 they announced that they were changing the name to AGE OF BOOTY. Not a lot of explanation was given, but behind the scenes we'd learned a few things

about naming a game. One thing we learned is that the pirate space is crowded; it's difficult to come up with something original that uses standard pirate terminology. We wanted to avoid a subtitle and so settled on PLUNDER. This

was a calculated risk, but we were confident that we could handle everything that had come up in our initial trademark search. Later on we received a cease and desist order from a web-based game company that had a

game they never had trademarked, but had previously released, called PLUNDER. We're the IP owner, and so it was our decision how to move forward, but we ultimately didn't feel we had the financial resources to fight

a legal battle over both this and the original issues that had come up with the trademark filing, and so we changed the name to AGE OF BOOTY. And yes, we did consider CALL OF BOOTY!

GDC

09

learn
network
inspire

www.GDConf.com



Game Developers Conference

March 23-27, 2009 | Moscone Center, San Francisco

Visit www.GDConf.com for more information and GDC08 proceedings.

Age of Booty

POSTMORTEM



A screenshot from the finished game.

with the extensive Xbox 360 rules and guidelines, or Technical Certification Requirements (TCRs).

One of the reasons we ran afoul of so many TCRs was because we supported 4-way split screen, file-management for user-created maps, and complex online systems such as file sharing. That's the hidden price of these systems, and one of the reasons that we'd made some hard cuts early on. We also ran into problems when we realized that the process we'd established with Capcom for localization of text was completely borked, and so the finishing of the game dragged on. We ended up working through about ten release candidates with our test team before we finally submitted to Xbox Certification. This work paid off though, and our trip through cert was relatively painless despite the complexity of our feature set.

These delays could have had serious consequences if Capcom hadn't provided two extensions to the schedule specifically to improve gameplay and interface. We went into the publisher relationship very guarded but were pleasantly surprised with how things turned out. We got pretty lucky here, we ended up with a game that we love, and all of these problems pale when we look at them from that perspective.

BOOTY CALL

We just hired one more programmer, who brought his 10 year old son to the office while he did a little prep work before his first day. He sat him down with AGE OF BOOTY and promptly forgot about him while he dove into other things. Several hours later, when it was time to pack up for the day, I realized that his

son was still sitting there, quietly playing through the single player challenges. I asked him what he thought of the game and he said that he loved it and didn't want to stop playing. Then he showed me a custom map that he'd created. That was a great moment.

We did something a bit unusual for this game—we brought a highly experienced team of developers together; a team that, at least in recent memory, has worked on nothing but big, multi-million dollar, multi-year projects. Based on the strengths of the individuals we set out to build something very big in a very small package. Had we been able to focus solely on this project I have no doubt that things would have gone more smoothly, but a big takeaway was also our need to build a true production management discipline at Certain Affinity. Fortunately Capcom stepped in and helped us with this early on, providing a ton of assistance, and since then we've hired a strong production management lead and another production manager that used to be lead production manager on the Xbox Live team.

Despite all the challenges we ended up with a game that we're very proud of, and along the way we gained a ton of experience working together and working with Capcom. And of equal importance, many of our people gained their first experience actually shipping an Xbox 360 title. When all is said and done, AGE OF BOOTY is a testament to the strength of our individuals and a monument to the difficult process of building a new team. ❌



NVIDIA's PerfHUD 6

Amir Ebrahimi

THE LAST TIME GAME DEVELOPER

reviewed PerfHUD was during the summer of 2006 when version 4 was released. Since then, PerfHUD has completed two major revisions with the current release landing at version 6.1. At the time of this review I have been using the beta of version 6.1, however, a final version should be available by the time you are thumbing through these pages. Although version 5 never received a Tool Box review, it was a finalist in the 2007 *Game Developer* Front Line Awards in the Programming Tools category.

PerfHUD is a free tool that according to NVIDIA allows you to analyze your application like an NVIDIA engineer. If you are fortunate enough to be a developer working with the NVIDIA Developer Relations (DevRel) group, then most likely you receive plenty of feedback already on how to correct errors and improve the performance of your game. Unfortunately, there is a plenitude of games in development at any one time and a limited number of DevRel engineers, so maybe you do not fall into this category. In that case, PerfHUD is your best bet for

fixing graphical errors and fine-tuning the performance of your game when it is running on an NVIDIA GPU.

FOR THE NEW USER

If you have used PerfHUD before, then you'll want to skip to the next section to see what to expect from this latest version. If you have never used PerfHUD, then I can think of a few reasons why:

- You did not know it existed, which could mean that this is your first time working in the fun world of graphics programming.
- You are so busy with adding features to your engine that you haven't had time to look at performance issues.
- You are painfully attempting to profile your graphics performance using standard sampling or instrumenting profiling methods.
- You are a technical artist responsible for the performance of the assets the whole art department produces.
- Your game is already running at 60 fps, has loads of graphical features, and you are getting plenty of sleep at night.

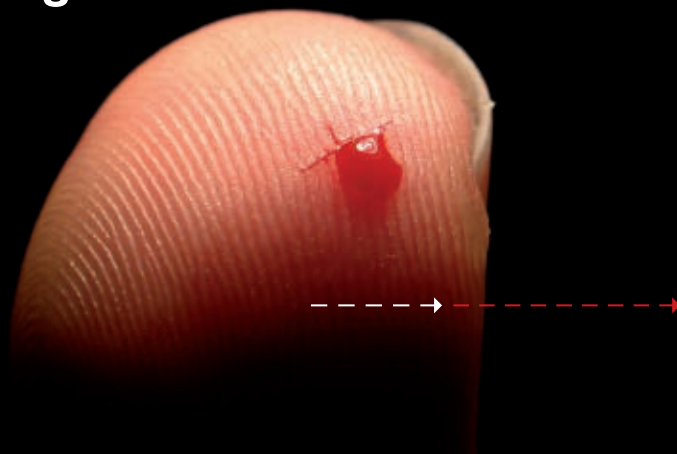
If you happen to be the last guy or gal, then please by all means share your wisdom in a future *Game Developer* article. However, even if you are this person you can count on a time when you are approached and asked:

"How come the game is running so slow?!"

Now begins the fun part of tracking down just what happened. You might start by simply profiling the different parts of your engine to see where time is being spent. Be careful if you take this route, as you'll soon encounter the inaccuracies of measuring API calls. Microsoft has an excellent article about this topic called *Accurately Profiling Direct3D API Calls* in the DirectX SDK documentation. As you will soon see, not having access to the internals of Direct3D and the display driver make it very hard to resolve your graphics performance issues. Enter PerfHUD, which works specifically because NVIDIA does have direct access to the internals that are so coveted.

If you have never used a GPU performance tool before, then you may

A minor issue in the beginning.





PerfHUD's Performance Dashboard running on top of AVERT FATE.

be overwhelmed at the wealth of data you will receive with PerfHUD. PerfHUD has four major modes that serve as your heads-up display: Performance Dashboard, Debug Console, Frame Debugger, and Frame Profiler. Which of these modes you are working in the most will depend on what problem you are trying to solve.

Performance Dashboard is where you get a high level view of your graphics performance with a nice set of graphs that can be customized. You can see such data as the number of draw calls, average batch size, and memory requests for textures, render targets

and buffers. Most importantly, you get graphs of the time spent in the driver, idle GPU time, and total frame time. Using this graph alone you should be able to figure out whether your application is CPU-bound, meaning that you are doing too much work per frame in your game code, or GPU-bound, meaning that either you are sending too much down the pipe or your shader code may be too expensive. If you are CPU bound, then you can suspend work with PerfHUD for now and hop over to profiling your game code.

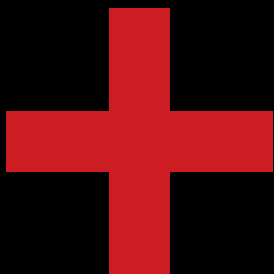
If you are operating on a GPU that is prior to the GeForce 6, you will only be able to perform some simple experiments

in Performance Dashboard. With the following key combos you can target specific bottleneck areas:

- CTRL+T** Isolate the texture unit, which will show if you are limited by your texture usage.
- CTRL+Shift+T** Swap textures with a color mip-map texture, which will display visually if you are making the best use of your mip levels.
- CTRL+S** Scissor all pixels, which shows you whether rasterization and shading is your culprit.
- CTRL+N** Simulate an infinitely fast GPU, which will show you if you are CPU-bound.
- CTRL+M** Eliminate geometry, which will reduce the amount of vertex unit work that the GPU is doing and subsequently geometry and pixel shading work that is done.
- CTRL + 1** Color fixed function pixels with red or disable fixed function work.
- CTRL + 2** Color ps_1_1 in light green or disable.
- CTRL + 3** Color ps_1_3 in green or disable.
- CTRL + 4** Color ps_1_4 in yellow or disable.
- CTRL + 5** Color ps_2_0 in light blue or disable.
- CTRL + 6** Color ps_2_a in blue or disable.
- CTRL + 7** Color ps_3_0 in orange or disable.
- CTRL + 8** Color ps_4_0 in red or disable.

Additionally, in this mode, you can adjust the game's speed and the window used for calculating your frame rate average.

Debug Console is where you can see any output from your application, DirectX, or PerfHUD. For a programmer



Launching an MMO is a complex process. It requires skill and great fortitude, sprinkled with optimism. Each decision is dependent on the next. Before your MMO hits the crowd, the work begins to make sure your big idea is both well designed and well coded. It also has to be technically capable of coping with several thousands of concurrent players – if not, something that appears to be a minor technical issue in the beginning, can lead to a catastrophe once your game is launched.

Get Help at

www.MMOhospital.org

Because you only get one chance.

this may not be a big deal, however, for a technical artist who is not running the game through an IDE it can be extremely helpful for seeing messages from the application that would normally go unnoticed. You can filter messages in this mode, clear the log, and stop further messages from being printed to the log.

Frame Debugger allows you to see your draw calls one at a time in the form of textures used for the call, render targets after the draw call, and a wireframe of the current call. New to version 6, you can override specific textures with a variety of utility textures. You might use these overrides when a shader is misbehaving and you are trying to track down if it is related to diffuse or specular term calculations or possibly a rogue noise texture. Finally, in this Simple view, you can toggle between logs of draw call dependencies, all the individual D3D API calls, the current calls for your draw call, and D3D Perf Events that you added specifically to your game code.

In the Advanced view, you can really get to work debugging your frame by seeing your geometry data as it works its way down the pipe. You can start by verifying your input with the Vertex Assembly tab. Next you can take a look at your vertex shader, sampler states, constants, defines, and input textures. Here is one of the killer features of PerfHUD: you can edit your shader, compile it and see results immediately. It's even easy to toggle between your edited shader and your original by pressing F9. You get this same

functionality with the Geometry Shader and Pixel Shader tabs. Finally, you can play with your final rasterizer operations by changing the rasterizer, depth/stencil, and blend states directly. You can also toggle between your modified and original states with F10.

Frame Profiler is the final mode of PerfHUD, which lets you really peek under the hood at all the details of rendering a single frame of your game. You get precise CPU/GPU time in milliseconds, instruction count ratios, and much more detailed graphs including shaded pixel count/coverage, unit utilization, and unit bottlenecks. This mode has helped me countless times to catch expensive draw calls that resulted in a small number of shaded pixels.

FOR PREVIOUS USERS OF PERFHU

Since PerfHUD 3, versions have improved greatly with each revision. With PerfHUD 3 we saw the introduction of the Frame Debugger, and PerfHUD 4 introduced the Frame Profiler mode. The transition from version 4 to version 5 of PerfHUD, however, was a bumpy ride for those who absolutely needed the new functionality of version 5 and could not wait until the final release. For that reason, I readily joined the beta program.

While developing HELLGATE: LONDON, we were targeting 32-bit and 64-bit platforms for both Windows XP and Windows Vista. Additionally, for Windows Vista we were supporting DX10. Among PerfHUD 5's new features was support for

Windows Vista and DX10. I started using PerfHUD 5 with the first beta because Microsoft's PIX was still maturing as a useful tool for solving DX10 issues and only recently has become much more solid. I also needed to profile the 64-bit version of HELLGATE on XP and Vista, which PerfHUD provided before PIX added 64-bit support.

The fall of 2006 was a difficult time for game developers, Microsoft, and GPU vendors. DX10 appeared to be behind schedule when compared to Vista, GPU vendors were still revising the hardware, and drivers were terrible. Many developers who were supporting DX10 and Vista know that the DXSDK, drivers, Vista revisions, and developer tools were all being developed concurrently. One of the most frustrating problems was having newer drivers being pushed for DX10 hardware that fixed critical issues and then having to wait for those same fixes to be pushed to the instrumented drivers. This meant that there was a blackout period where you couldn't profile your graphics performance if it depended on the driver fixes. By adding 64-bit to the mix, you had to wait even longer for those changes.

Fast-forward to now and those growing pains are long forgotten. One of the best new features of PerfHUD 6 is that on Windows Vista there is no longer a distinction between the instrumented and release driver. Now every driver past version 173 has instrumentation support included. However, if you are developing for XP you still have to install

A minor issue can turn into a serious nightmare in the end.

#623342627872
ELECTRIC STUDIOS, SFO
O OF INVESTIGATIONS
STRONGLY PROHIBITED.

an instrumented driver. Additionally, you can now use PerfHUD to profile 32-bit applications on a 64-bit operating system, which was not possible with PerfHUD 5. The other features worth noting are:

- Per-texture overrides.
- A new full-screen texture/render target visualizer.
- A full API call list.
- Support for D3D Perf Events that even allows you to set debug breakpoints that PerfHUD can catch.
- SLI support with analysis tools to catch common SLI bottlenecks.
- Draw call dependency view.

ROAD TEST

I started testing PerfHUD with the demos that are included in the full PerfKit install. The first thing I noticed is that mouse and keyboard handling has not improved much since version 5, despite what is said on the PerfHUD home page. With the included SPARKLES demo, I proceeded to hit the hotkey over and over and could not get the menus to come up. I closed the demo, launched it again and then it worked. Intermittently, it seems to regress to the point where I can't use the Fn keys to move between the different modes. I've seen this behavior before when I used previous versions

of PerfHUD with HELLGATE: LONDON. One thing that does seem to be fixed is that the mouse pointer is no longer offset from the actual position of the cursor. In the past, I had to turn on the Show Software Cursor option in the PerfHUD configuration window to see where PerfHUD thought the mouse was.

Next, I tried out the new 32-bit support on a 64-bit host platform by testing the AVERT FATE demo from Unity Technologies. At first I noticed that PerfHUD wasn't working even though the executable had compiled in support. Upon quick inspection with a hex dump I noticed that the driver string that was being searched for was "NVPerfHUD" and not simply "PerfHUD". I believe that this driver string changed with version 5 of PerfHUD. This was simple enough to fix by editing the executable directly.

After that change, I was off and running the demo with a new PerfHUD 6 overlay. I encountered more issues with the mouse support once I activated PerfHUD. While PerfHUD is supposed to intercept mouse input it seems that mouse clicks were still passing through to the game. It is possible that the engine is gathering input in a way other than what PerfHUD expects, but in exactly what way eludes me. Also, keyboard response only seems to work some of the time, which may be because it is also gathered in a non-PerfHUD-friendly way. I searched the developer forums and it seems that others are having similar problems.

I found a spot in the AVERT FATE demo where frame rate dropped by 25 percent. I performed a few simple tests and found that with the scissoring test (CTRL+S) the frame rate improved quite a bit. So, I proceeded to analyze this further with the Frame Profiler, as it appeared to be a post-vertex unit issue. With the support to easily sort on any column, I first started by sorting state buckets on GPU time.

Next, I looked at the draw calls for that bucket, also sorted by GPU time. The biggest offender was an inexpensive piece of geometry that took 1.4 ms on the GPU and only shaded 342 pixels and was apparently bottlenecked by the frame buffer. The sad part about this is that the same piece of geometry was completely overwritten about 20 draw calls later.

There are many ways to solve this problem such as using a PVS (Potentially Visible Set) system, making two passes with color-writes turned off in the first pass to perform an early fill of the Z-buffer, simple depth culling, or possibly sorting opaque geometry in front to back order. From past experience, I have found that improvements you make to fix specific performance problems usually improve the rest of a game's performance. However, you have to be careful because some performance improvements may only help with a certain generation of GPUs and penalize others.

FINAL THOUGHTS

The features that have been added to PerfHUD over the last two years make it an indispensable tool for performance optimization. The amount of data you can get from the tool for the small amount of time it takes to add the code to enable PerfHUD support in your game is remarkable. Kudos to NVIDIA for unifying their Vista instrumented and release driver.

Now that there are a wealth of features and support in PerfHUD, I'd like to see NVIDIA focus more on the issues that cause developers so much frustration. The company should fix issues such as the mouse/keyboard interception weirdness as well as add a unified driver for XP if that is possible. Documentation should be corrected and could be improved by adding more information about what some of the graphs mean (e.g. Bottlenecks Per Unit). Also, I could see having a tutorial app that helps a new PerfHUD user determine what problems can affect GPU performance, rather than including two demos that already appear to be running at reasonable frame rates.

NVIDIA informs us that it is working to address the mouse and keyboard issues and it expects to deliver a new version of PerfHUD by the time this issue goes to print.

AMIR EBRAHIMI has been working in the games industry for five years. He has been eating his fill of danishes and pølse while working with the Unity guys out in Copenhagen, Denmark before returning Stateside. When back in San Francisco, he also teaches part-time at the Art Institute of California. Email him at aebrahimi@gdmag.com.

PERFHUD 6

★★★★

STATS

NVIDIA
2701 San Tomas Expressway
Santa Clara, CA 95050
http://developer.nvidia.com/object/nvperfhud_home.html

REQUIREMENTS

An NVIDIA GPU: GeForce 6 Series and later GPUs are supported, as well as G80, G70, and NV4X-based Quadro FX GPUs. Older GPUs are supported with reduced functionality. Microsoft DirectX 9.0c or Microsoft DirectX 10. Windows XP or Windows Vista.

PRICE

Free

PROS

- 1 Windows Vista no longer requires a separate, instrumented driver.
- 2 You can now profile 32-bit applications on a 64-bit host platform.
- 3 Texture overrides can be set on a per-texture basis.

CONS

- 1 Input interception for mouse and keyboard still seems flaky.
- 2 The documentation still has errors in it from previous versions. Use the PerfHUD help (F1) to see which key presses actually do something.
- 3 Multiple installers for 32-bit and 64-bit versions.



GAME CAREERGUIDE.COM

THINK SERVICES

GET YOUR GAME ON!



DOWNLOAD AND PLAY THE LATEST STUDENT GAMES!

EVERYTHING YOU NEED TO KNOW TO GET INTO THE GAME INDUSTRY!

- News and features for students and educators
- Getting Started section – an invaluable how-to guide
- Message Boards

NEW!

Download your FREE digital edition of the 2008 *gamedeveloper* Game Career Guide online at: www.gamecareerguide.com



I'LL MATCH YOUR INTEREST AND GOALS WITH THE RIGHT GAME RELATED PROGRAMS AND SCHOOLS FROM AROUND THE WORLD.

www.gamecareerguide.com

Sponsored by **Ai** The Art Institute of Pittsburgh®
Online Division



STEVE THEODORE

PIXEL PUSHER

GEMS OR EXPECTATIONS?

How artistic 'gotchas' hook the player

WORKING IN GAMES HAS ITS PERKS, BUT it also has one very important drawback. Becoming more knowledgeable and analytical about the business can also make you more cynical and hyper-critical. When you make games 9 to 5 (or, more likely, 10 to 8) you can lose touch with the little bits of magic that made us fall for games in the first place.

Having a kid in the house is a great antidote for cynicism. I've been following the technical and business stories about *SPORE* for more than two years. I've been to sessions at both GDC and SIGGRAPH devoted to the game's unconventional animation system and development challenges. But it didn't take long to see that what made my son go nuts for the game has nothing to do with the technical challenges, production hassles, and business brinkmanship that we've all read about. An eclectic selection of snouts, the swaying skirt for your tribal chieftain, or the way a hapless alien flails around as he's sucked up into a probing UFO may require all sorts of sophisticated technologies, but they aren't products of computer science. They are the result of many decisions, great and small, by people like us, who found some time in their schedules for little touches that bring the game to life.



Dynamix's RED BARON.

Watching a kid play, you start to see that some of the decisions that might seem least important to a cynical old dev are the ones that really sell a game to its audience.

THAT MAGIC MOMENT

Making hard trade-offs and carefully picking your battles is a constant refrain for pixel pushers, but even the surliest vets have to admit that sometimes it's the grace-notes that really matter.

Like a lot of game people, I have a very clear memory of the moment I turned from a casual consumer of games into a wild-eyed convert and would-be game artist. As a child of the 70s and 80s I'd certainly pumped a lot of quarters into arcade machines, but for me the moment when amusement was transformed into amazement was a rainy afternoon in 1992. I was ducking my grad school coursework on Vulgar Latin Palaeography with a fresh copy of the classic Damon Slye dogfight sim, *RED BARON* (Sierra, Dynamix). I was duly impressed by the state-of-the-art 8-bit sound, the 320x240 16-color graphics, and the 3D planes in all their 50 polyгон glory.

In those days, of course, that was heady stuff. But the magical moment had almost nothing to do with tech.

Limping home from my first tutorial foray across the Western Front, in a badly shot-up Fokker Eindekker, I was jumped by a pair of marauding DH-VIIs. As I dove under the nearest cloudbank for cover, the little 16-color bitmap that represented my instrument panel darkened. I didn't really process it as a game feature for a few seconds—I'm in the shadow of the cloud, so it's dark, right?—but I was hooked. I didn't care how it was done. I just knew that for a few seconds I was there in the cold, wet mists of 1915 and somebody, somebody who really loved what he was doing, had sent me that elegant little invitation to suspend my disbelief.

It was the first time I wondered what it would be like to be that guy. That's pretty good for what must have been just a simple EGA palette swap.

GOTCHA!

Ever since then I've always had a soft spot for those beautiful little "gotcha!"

STEVE THEODORE has been pushing pixels for more than a dozen years. His credits include *MECH COMMANDER*, *HALF-LIFE*, *TEAM FORTRESS*, and *COUNTER-STRIKE*. He's been a modeler, animator, and technical artist, as well as a frequent speaker at industry conferences. He's currently content-side technical director at Bungie Studios. Email him at stheodore@gdmag.com.

moments, the little gems that don't do anything to directly advance the story or enhance the gameplay, but which set the tone for the entire game by rewarding the player's investment in the game's illusions.

The most common gotchas are the sorts of details that players usually call "realistic." Fans of the original 1998 *HALF-LIFE*, for example, always recall fondly the working soda machines and microwaves in the Black Mesa Research Lab. But those immersive moments aren't about reality. They're about interactivity, bringing life to parts of the game that might otherwise be sterile. The phenomenal success of *LEGO STAR WARS* and its successors owes a huge debt to the myriad clever little interactions players encounter as they button-mash their way across the galaxy.

Whatever the genre, these gotchas are an overlooked part of gaming greatness. For us pros it's easy to fixate on good UV maps or precise transition animations. But for our audience, it's often these little moments that forge the emotional tie to the game. Try Googling "GTA" and the phrase "little touches." It turns up about 10,000 reviews.

These little content flourishes—the kinds of details and extras you have so much trouble sneaking past your producer—matter to making great game art.

GREAT EXPECTATIONS

Gotchas may be "hidden" content, but they're not the same as Easter Eggs. Part of the definition is that gotchas don't require any detective work from the audience. They hide in plain sight. In fact, accessibility is a key to success. There's no point staying late to add your favorite little surprise if you know it'll only be seen by forum trolls and script kids with debuggers.

The essence of a good gotcha is a mixture of surprise and recognition.



Valve's *HALF-LIFE*.

The surprise comes when a game offers more than you expect; the recognition, when you realize that the game correctly anticipated your expectations.

"Hey look! When I stand under the torches, the plumes on my helmet catch fire!" That byplay of expectations is really the key to why we love gotchas.

It's a sad fact of life in games that all our fancy simulations and painstaking content constantly disappoint our players. Whether it's that six-inch high ledge that we won't let them climb, or the plate glass store window that miraculously withstands hits from an RPG, we're always setting up situations in which the player's subliminal expectations are frustrated. Most of us feel that frustration as acutely as our players. Many gotcha moments come from content that tries to live up to players' expectations better than pure graphics horsepower can. Your physics engine can't deform the handrails in the skate park in real time for every grind, but if you can steal the time from your schedule (or your family life) you could hack in that flex as a content fix and delight your players.

It's a mistake to think that realism is the important goal here. If reality were really so great, our audience would be outside getting some exercise instead of paying for pizza and Diet Cokes. Our players (bless their little hearts) are as forgiving about the limitations we place on them as they are out of shape. The pleasure of a gotcha isn't the somber intellectual appreciation of a solidly tuned physic simulator. Instead, it's more like participating in improv theater. It's the moment when we break the fourth wall and flash the players a wink about the illusion we're helping them experience.

If you were feeling particularly postmodern, you could toss in a reference to Derrida and the play of symbols here, though the interplay of dimensional imagery and the funky deconstruction of raster graphics in *SUPER PAPER MARIO* is both more fun and more sophisticated. But, fancy terminology aside, players love to be included in the act. Like the audience at a magic show, they're willing to be hypnotized if it's done with skill, economy, and a little acknowledgement of their part in the process. It's no

surprise that the companies that do a consistently good job of including players in the joke—Blizzard, Rockstar, and Nintendo come to mind—have fanatically devoted followers.

So gotcha moments aren't just a simple matter of realism. They're about acknowledging all sorts of player expectations. Sometimes we add little



LucasArts' STAR WARS: FORCE UNLEASHED.

bits of content to paper over the limits of a physical or graphical system. But just as often, we're playing to cultural or dramatic expectations rather than Newtonian mechanics. When you infiltrate the Tie Fighter factory in STAR WARS: FORCE UNLEASHED and are immediately confronted by squeaky little shoebox-bots, you are pleasantly surprised by the rightness of it all, no matter how little influence it has on your level grind. Over the years the radio and television programming in the GRAND THEFT AUTO games has almost transcended the narrow concept of a gotcha to become a new genre of its own, the free-floating media satire game.

HOW MUCH IS TOO MUCH?

Of course, the success of the GTA brand of media satire suggests an interesting problem. The definition of "gotcha" we've been working on is narrow. We're talking

about very small, very personal bits of content, the sort of mini-feature that gets into a game because of the passion and enthusiasm of a few individuals, rather than a proper feature with all the attendant scheduling and support. The classic gotcha is the brainchild of the lone artist who stays up late making breakable versions of all the office

furniture because bulletproof Aeron chairs leave the climactic shootout scene unsatisfying.

But what happens when everybody loves the busted up furniture in the first version of the game and they expect more in the sequel? All-new Fully Destructible Office Furniture™ technology, with its attendant tech specs and design docs and a team of outsource artists

cranking away on upholstery gibs, is a very different beast.

The little itch that drives us to sneak our gotchas into the game is the urge to satisfy our player's expectations. ("It's just an office chair, and I have an AK47!") But once we change the player's expectations, we can end up on the hook for satisfying them and suddenly the "gotcha" factor goes in the other direction.

Raising expectations you can't satisfy can be as negative as exceeding players' expectations can be positive. When you start sprinkling your work with gotchas, you run into a problem very much like the Uncanny Valley. [See "Uncanny Valley," Pixel Pusher, *Game Developer*, December 2004.] For the artist who wants to add a little extra special something in the way of satisfying little details or extra interactivity, it's important to stay in

the pleasant position of exceeding the audience's expectations.

PASSION PROJECTS

The appeal of the half-hidden gem and the beautiful little surprise is undeniable, but unfortunately, it's hard to institutionalize. In a big-money business like ours, careful calculation of means and ends is a virtue. Every production artist learns (usually the hard way) that the cold realities of schedules and deliverables have a way of edging out the happy little inspirations and pet projects. Even for the best teams, the delicate dance with player expectations is an art rather than a science, and it's hardly surprising if leads and producers would prefer to keep their artists focused on sure-thing content and features.

Sobriety and forethought are helpful. But we can't ever forget that we make games, not airliners or medical equipment. Having fun is part of what we do (even if we work on ponderous, self important epics way too often). Regardless of genre, our work lives or dies by pulling players into the worlds we make. Sometimes we do that by awing them with our graphic prowess, but sometimes we cajole them in with a nod and a wink and a little token of our enthusiasm.

The gotcha moment is a very human element in a business that sometimes can be too businesslike for its own good. We should all keep trying to sneak in those little extras whenever we can muster the time. If you can't institutionalize the creativity and playfulness of the most memorable gotchas, you can at least make sure to schedule enough polish time and build tools that encourage artists to experiment and push boundaries. You can certainly recognize and reward the folks on your team who do the best at anticipating and exceeding the expectations of the players. But most of all, you can try not to forget the unnecessary—but absolutely vital—little moments that got you into this crazy business in the first place. ❖



Jobs on GAMASUTRA

The Art & Business of Making Games

The Leading Professional Game Source for Job Seekers and Employers

The screenshot shows the Gamasutra website interface. At the top, there's a navigation bar with links for 'The Art & Business of Making Games', 'Jobs on GAMASUTRA', 'The World's Largest Game Industry Resume Database and Job Board', and 'Log onto Gamasutra.com/jobs'. Below this is a search bar and a 'Search' button. The main content area features a 'The Game Industry Leader in Career Resources' banner with 'JOB SEEKERS' and 'EMPLOYERS' buttons. A 'Get the Most Career News' section includes an email sign-up form. The 'Gamasutra Job Board' section is active, displaying a list of job categories with their respective counts: Accounting / Finance (0), Audio / Music / Sound (4), Castleton / Services / Tech Support (2), Eschelon / Management (9), Harmon / Services (1), Producer / Director / Product Manager (3), Network / System Admin / IT (2), Programmer / Engineer (121), Retail (1), Art / Animation (109), Business Development / Sales / Leads (5), Educator / Instructor (6), Game / Level Designer / Creative Director (57), Writer / Submitter (4), Marketing / PR / Communications (12), Other (8), QA / Tester (4), and Sales (2). There are also logos for 'GAMASUTRA', 'GAMESDECK', 'GAMES SOURCE', 'GAMEDUTIES', 'INDIE GAMES', and 'WORLD OF GAMES'.

NEW FEATURES:

- Job Seeker Center
- Employer Job Manager
- GDC Events Job Section

Subscribe to the **CAREER NEWSLETTER** – a weekly update with detailed listings of all recent jobs posted on Gamasutra.com.

The Web's Largest Game Industry Resume Database and Job Board

Log onto **Gamasutra.com/jobs**



» THE INNER PRODUCT

SECRETS OF MULTIPLATFORM
DATA BAKING**BAKING IS OFTEN DESCRIBED AS**

precision cooking. Unlike boiling spaghetti or simmering a stew, baking usually requires a very particular ratio of ingredients, precise directions, and an exact time and temperature in the oven. If any of those parts changes, the overall result will be affected, often in not very pleasant ways.

The same is the case with data baking. In last month's column, I covered the basics of data baking, from the time data is exported from the content-creation tool, to the time it makes it into the game. What I didn't discuss was that different target platforms often have different in-memory formats. If the memory image created by the baking process is off even by a single bit somewhere, the result is usually completely unusable data.

BYPASSING THE PROBLEM

It is possible to bypass the problem completely by baking your data in the target platform. Then all you need to do is load up your data structure and save it to disk. End of story.

That approach might work well for PC games, even when you're doing cross-platform development. If you're developing in Windows, it's pretty easy to

involve a Mac or a Linux machine in the baking process. Although in the case of PC games, the amount of data baking you can do is limited because you hardly ever know the exact hardware your game will run on, so you might not need to involve target machines at all.

If you're developing for game consoles or other fixed platforms, where you definitely want to create the exact memory image of your data, you could consider involving the target platform in the data baking process. While it is possible to use a game console to build your data, and some games have done that in the past, it's a route fraught with peril and potential for disaster. You're likely to encounter difficulties in automating data builds, poor handling of errors and crashes, slow builds because of underpowered machines, or even just a lack of hardware to build data on.

Whenever possible, stick to using development PCs for data baking. That will give you the fastest and most reliable builds, but you'll need to invest a bit of work to create the exact memory image for your data.

A MATTER OF SIZE

Imagine you're about to bake a structure like this:

```
struct WaypointInfo
{
    int m_id;
    bool m_active;
    float m_position[3];
    char m_letter;
};
```

It looks like an innocent enough data type, right? Surprisingly, it can have wildly varying sizes and layouts in different platforms (see Figure 1 Pg. 41). There are three things that will affect its size and layout:

1. SIZE OF BASIC DATA TYPES

How big is an int exactly? Or a bool? The C standard doesn't specify a size for any fundamental data type. In some platforms ints and floats will be 32 bits, and sometimes they will be 64 bits. You're likely to encounter the inconsistency that a Boolean can sometimes be 32 bits and sometimes 8 bits. That's going to change things!

2. MEMBER PADDING

You may think that all members of the structure will be laid out sequentially in memory, and that's almost true. Again, referring to our trusty, if slightly verbose, C standard, we find that members have to be laid out in memory in the same order they were declared in the structure (only if they're within the same public/private/protected block though), but they could have gaps between them. How big are those gaps? It depends on the platform, compiler, and specific compilation flags. Most often compilers will add padding between member variables to improve performance access to each member variable. In 32-bit processors, it's faster to load a 32-bit value if it's aligned on a 32-bit boundary, so most compilers will insert enough padding to align it that way.

3. STRUCT PADDING

Figuring out the size of the data types and the amount of padding in between each of them is not enough. Consider this array `WaypointInfo m_waypoints[10]`; Compilers will often pad structures for performance purposes so that arrays of them will be aligned correctly. In the example above, if we assume 32-bit integers, 8-bit Booleans, a 3-byte padding, 32-bit floats, and 8-bit characters, we might think the structure is 21 bytes. In reality, it will probably be padded to 24 bytes so subsequent structures in the array will be aligned on

NOEL LLOPIS recently gave up a steady paycheck, and decided to follow his lifelong dream of being an indie game developer. He keeps busy by pretending to do everything from programming and design to business and IT at *Power of Two Games*. When he was still getting paid, he worked on *THE BOURNE CONSPIRACY*, *DARKWATCH*, and the *MechAssault* series. Email him at nllopis@gdmag.com.

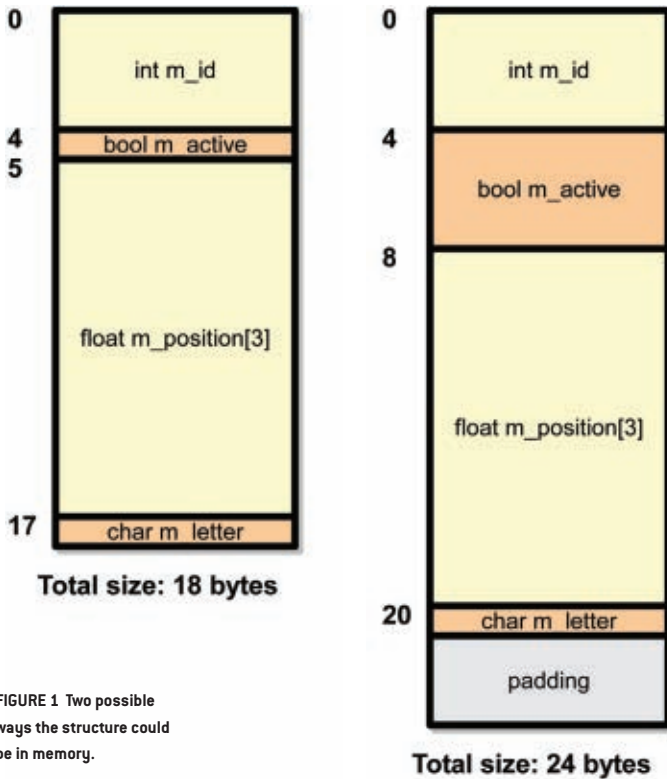


FIGURE 1 Two possible ways the structure could be in memory.

a 32-bit boundary. Some compilers might go as far as to pad it to 32 bytes.

What's a poor data baker to do with all those rules?

The varying size of basic data types is often dealt by creating user-defined data types that have a well-defined size. See Listing 1 for an example. In each target platform, you can provide definitions

for those data types so their sizes are the same.

Doesn't it seem wasteful that we're all redefining our own data types just so we can know their exact sizes? It gets even worse when middleware providers do the same thing. We end up with many different "basic" data types of varying sizes and properties all over the same code base.

Fortunately, C99 introduced a new header file `stdint.h`, which among other things, declares integer data types with exact sizes, such as `int8_t`, `uint8_t`, `int16_t`, `uint16_t`, and so forth. Do yourself a favor and start using those data types whenever exact size is important. If your compiler isn't yet C99 compliant (tsk, tsk, Visual Studio 2005!), you can get a third-party header file that adds those defines. (See Resources.)

The rules for member and struct padding aren't defined in the C++ standard, so it's completely up to each compiler implementation to decide how to do it. Fortunately, a lot of common

compilers (most notably Visual Studio and gcc) support the `#pragma pack` directive, which allows you to specify the byte alignment desired in your structures. You can either use `#pragma pack` everywhere that matters, or you can learn the padding rules for your compiler by implementing those structures and seeing what the compiler creates.

Another common source of problems are bitfields. Using the C language bitfields is very handy to pack flags into a small amount of space:

```
struct EntityState
{
    bool m_active : 1;
    bool m_invisible : 1;
    bool m_invulnerable : 1;
    bool m_playerControlled : 1;
    bool m_inVehicle : 1;
    // ....
};
```

The C++ standard guarantees that all those flags will fit in one bit each plus some padding. What it doesn't make any promises about is exactly how those bits will be laid out and or how they will be padded. You either need to find out how the compiler in your target platform does it, or replace those flags with something you have control over, such as explicit bit masks on a 32-bit unsigned integer.

```
struct EntityState
{
    uint32_t m_flags;
    // .....
};

#define ENTITYSTATE_ACTIVE 0x00000001
#define ENTITYSTATE_INVISIBLE 0x00000002
#define ENTITYSTATE_INVULNERABLE 0x00000004
#define ENTITYSTATE_PLAYERCONTROLLED 0x00000008
#define ENTITYSTATE_INVEHICLE 0x00000010
```

In general, you need to watch out for anything that the standard doesn't explicitly dictate, and that's left up to each implementation. For each of those

LISTING 1
Example Data Definition for One Platform

```
typedef __int64 int64;
typedef signed int int32;
typedef unsigned int uint32;
typedef unsigned short uint16;
typedef unsigned short int16;
typedef unsigned char byte;
```

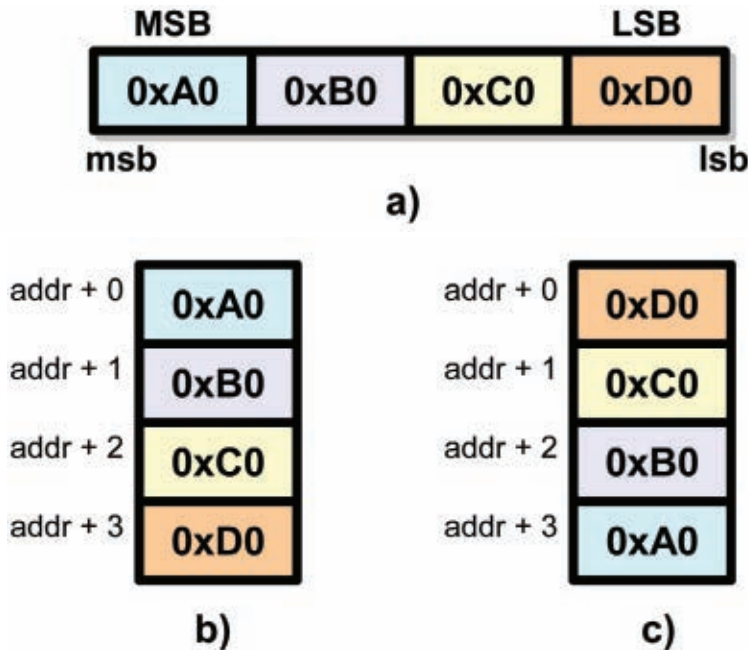


FIGURE 2 a) 32-bit data in a register. b) Memory layout in big-endian format. c) Memory layout in little-endian format.

cases, you should either substitute it with something that is well defined and consistent, or learn how each implementation defines it and make it part of the rules of your data baking.

LITTLE END OR BIG END?

Once you've figured out the size of your data types and their offsets in the structure, you still need to know how exactly they're stored in memory. You might know that an integer is 32 bits, but what bit pattern describes a particular number?

There are two parts to that answer. The first one relates to how data types are represented in different hardware. And here, there's good news: Most modern platforms use the same method to represent basic data types. Signed integers are represented with two's complement, and floating point numbers use the IEEE 754 standard for both 32- and 64-bit numbers (sign, mantissa, and exponent). A few platforms might not support floating point numbers, in which case we'll need to translate the data to fixed point or some other format. But in

most cases, this is not something we have to worry about.

That's not the end of the story, though. The second part of the answer relates to how that number is stored in memory. In all modern platforms, a byte (8 bits) is the smallest addressable memory unit. Data types that are just a byte long (like a char) are simply stored at a particular memory address in a single byte, with nothing more to it.

The problem comes with data types that are larger than a single byte. Integers and floats are often 32-bits long, which is 4 bytes. How are those bytes arranged in memory? This is such a fundamental issue that you would hope there were one standard everybody followed. Unfortunately, because of historical reasons, there are two standard ways to do it.

Figure 2A shows the 32-bit integer 0xA0B0C0D0 broken down into four bytes. The bit on the far left has the highest potential value (2³¹) and is called the most significant bit (msb). Conversely, the bit on the right has the lowest potential value (2⁰) and is called the

least significant bit (lsb). Extending this to bytes, the byte containing the lsb is called the least significant byte (LSB) and the one containing the msb is the most significant byte (MSB).

One approach, known as the big-endian format, stores bytes in memory starting with the MSB (see Figure 2B). The other approach, little-endian, stores bytes in memory starting with the LSB (see Figure 2C). The names big-endian and little-endian come from Jonathan Swift's *Gulliver's Travels*—tensions are high between two rival nations because one cracks its eggs on the big end, and the other cracks them on the little end, and each is convinced that its way is the correct way.

Just like in *Gulliver's Travels*, both formats make perfect sense depending how you visualize memory and data, and there isn't any advantage to using one over the other. Some architectures use one and some use the other one; just be aware of which format is used in each of your target platforms and format your data accordingly.

As an example, Intel and AMD-based CPUs use little-endian format, whereas PowerPC CPUs (which include the Microsoft Xbox360, Sony PlayStation 3, and Nintendo Wii) use the big-endian format. Some platforms go as far as being able to switch between the two memory formats.

It's not just the CPU that needs to be little-endian or big-endian. Any hardware that fetches multi-byte data types from memory needs to be aware of the format of that data. Most GPUs can work in either mode for that reason and are customarily set to match the CPU format to keep programmers from going insane.

Data endianness is something that programmers only have to be aware of when sharing binary data between different platforms. You might never have to think about data endianness if you're only developing for a single platform. You really don't care in what order those bytes are stored in memory; you just load them into a register and the CPU takes care of fetching them in the correct order.

THE INNER PRODUCT

An example of a common situation in which data endianness is crucial is network communication. Binary data is transmitted over network packets and might be received by very different platforms. Fortunately, to allow different machines to communicate with each other and interpret the data in the same way, everybody agreed on a standard network format for binary data—big-endian. The network sockets API provides a set of standard functions to convert long and short data types between the host format and the network format (`htonl`, `htons`, and `ntohl`, `ntohs`), which do nothing in hosts with native big-endian format, and swap bytes around in little-endian platforms.

SAVING DATA

As game developers, the most common situation in which we have to deal with byte-endianness is saving and loading data across multiple platforms. Whether it's because we're baking data on a little-endian PC and loading it on a big-endian console, or because we want save games to work across a variety of platforms, we need to be very careful how we arrange those bytes.

We could take the same approach as network data and just pick one

format and transform the data into that format before saving it. Then, if the target platform uses a different byte-endianness, we could swap the bytes around at load time. That approach would work, but it would add an extra operation at load time that we could have done ahead of time. So we fold that operation into the data baking process.

When we create the memory image for the data we're baking, we need to compare the byte-endianness of the target platform and the building platform. If they're both the same, we don't need to do anything extra, and we continue baking as usual. If they're different, we need to rearrange the bytes of every data type larger than one byte. Listing 2 shows a function that swaps the endianness of a piece of data for any data type.

It's useful to note that data endianness is a completely orthogonal concept to the way the data is represented. Both a 32-bit integer and a 64-bit floating point number are going to be stored MSB-first in a big-endian format. This will make our job a lot easier when converting data for specific platforms because we

can first convert the data to the correct representation, then convert them to the right data endianness, and finally apply any padding rules.

With these new tools in hand, we can now deal with different data sizes, padding, and byte-endianness and create perfect data memory images for just about any platform. Happy baking! 🍪

RESOURCES

MSINTTYPES

<http://code.google.com/p/msinttypes>

C++ STANDARD

Required reading for low-level C++ issues

www.open-std.org/jtc1/sc22/wg21/docs/projects

www.rtpatch.com

RTPatch and Pocket Soft are registered trademarks of Pocket Soft, Inc.

LISTING 2 Function for Swapping Endianness

```
template< typename T >
T SwapEndianness(const T in)
{
    T out;

    const unsigned char* src = reinterpret_cast< const unsigned char* >(&in);
    unsigned char* dst = reinterpret_cast< unsigned char* >(&out);

    for (int i = 0; i < (int)sizeof(T); ++i)
        dst[i] = src[sizeof(T) - 1 - i];

    return out;
}
```



THINK SERVICES
GAME GROUP

Calendar of Events

March 23-27, 2009

Game Developers[®] Conference

Game Developers Conference
Moscone Center, San Francisco, CA
www.gdconf.com

March 27, 2009

GAME CAREER SEMINAR

Game Career Seminar at GDC09
Moscone Center, San Francisco, CA
www.gamecareerseminar.com

May 12-13, 2009

GDC Canada

Game Developers Conference Canada
Vancouver Convention & Exposition Centre, Vancouver, BC
www.gdc-canada.com

September 14-18, 2009

Austin Game Developers[®] Conference

Austin Game Developers Conference
Austin Convention Center, Austin, TX
www.austingdc.net

For updates and more information on our events visit:
www.tsgamegroup.com



SOREN JOHNSON

» DESIGN OF THE TIMES

NARY A DIME

How the free-to-play business model affects design decisions

IN CHINA, A NEW MMORPG WITH A VERY aggressive business model, ZT ONLINE, has gained significant popularity. With more than 10 million users and an average revenue per user of \$40 per month, the game has made its publisher, Giant Interactive, one of the most profitable online entertainment companies in China [see References]. Like many Asian games, ZT is free to play (F2P) and focuses primarily on player-vs.-player gameplay. Not only can players steal from their defeated foes, but weaker characters can even be kidnapped and held for ransom, locking their owners out of the game.

Access to equipment in ZT is very limited. First of all, there are no loot drops from killing monsters or completing quests. Further, all items in the game are completely bound to the owner, so there is no way to trade for better weapons with other players. Instead, the primary way to gain equipment to empower one's character is by paying real money directly to the publisher to open "treasure chests." Essentially in-game slot machines, these chests have only a small chance of producing something useful, and finding the best equipment often requires opening thousands of chests. In fact, each day, the game confers a special bonus to the player who has opened the most chests, meaning the player who has spent the most real-world money to obtain better items.

ZT ONLINE's complete embrace, at every level of the game, of real-money

transactions (RMT) may be appalling to some in the West, but the game is in many ways at the vanguard of a trend to develop games that take advantage of the players' appetites for spending money to gain in-game advantages.

Ironically, the F2P-with-RMT model traces its origins to the challenge of getting Asian gamers to buy boxed, retail games, most of whom preferred the free ride of easy and widespread piracy. In



Giant Interactive's ZT ONLINE.

response, Korean companies like Nexon and NCsoft built server-based online games that could not be pirated and would require alternate business models.

Starting with subscriptions (including the world's first million-subscriber MMO, NCsoft's LINEAGE), the Korean industry eventually shifted to free-to-play games that made money from micro-transactions, such as Nexon's KART RIDER and MAPLESTORY. With many of these online games serving tens of millions of players, the Korean model has begun attracting the attention of major Western publishers, who have chartered their own free-to-play games in Asia: EA's FIFA ONLINE, Valve's COUNTER-STRIKE ONLINE, and THQ's COMPANY OF HEROES ONLINE.

The promise of free-to-play games is that players will get hooked on a free game and then eventually spend money on their new passion. However, designing these games is not a simple endeavor. The challenges of free-to-play design can make developers

appreciate how fortunate they were when they could design for a fixed-cost product, either a shrink-wrapped retail game or a standard subscription-based MMO. In a fixed-cost world, the designer can focus on just one thing: making the player's experience as engaging and interesting and fun as possible.

For a F2P game, however, designers have to balance making free content fun enough to engage first-time players but not so much fun that they would not yearn for something more, something that could be turned into a transaction sometime in the future. Every design decision must be made with a mind toward how it affects the balance between free and paid content. Thus, the true cost of piracy is that the line between game business and game design has become very blurry.

As games move from boxed products to ongoing services, business decisions will become increasingly indistinguishable from design decisions. Of course, the industry has seen game designers play businessmen before—a fundamental part of arcade game design was understanding how to suck the most quarters out of players. Thus, understanding how successful F2P game have navigated these waters is instructive.

BUSINESS OR DESIGN?

The aforementioned 2D MMORPG MAPLESTORY has an in-game RMT store where players can purchase items for their characters. These purchases can range from purely cosmetic items, such as funny shades or blue-colored hair, to consumables which give actual in-game bonuses. These consumables include tickets for earning double experience points over 24 hours, avatar warps for triggering instant travel, and ability resets for realigning character traits.

In a nod to in-game fairness, these bonuses only save the purchaser

SOREN JOHNSON is a designer/programmer at EA Maxis, working on an unannounced project. He was the lead designer of CIVILIZATION IV and the co-designer of CIVILIZATION III. Read more of his thoughts on game design at www.designer-notes.com. Email him at sjohnson@gdmag.com.

time instead of directly increasing the power of his character. This distinction is important as RMT can still have in-game meaning without needing to be tied to the game's best weapons and equipment, as with ZT ONLINE.

Another popular free-to-play game with a different business model is the web-based MMORPG RUNESCAPE, which uses optional subscriptions instead of optional micro-transactions. Subscribers gain access to more quests, new areas, player housing, and extra skills. Again, the designers have to decide where to draw the line between free content to grow the game and paid content to drive revenue. In RUNESCAPE's case, one in every six active players currently chooses to subscribe, striking a good balance.

TRAVIAN, a successful web-based MMO strategy game, allows players to purchase temporary in-game bonuses, such as +10 percent attack strength or +25 percent wood production for a week. These bonuses have been controversial among the community as many players feel obligated to buy them in order to compete at the highest level. Gamers can also purchase TRAVIAN PLUS, which unlocks an improved interface to make playing the game more efficient. The PLUS mode includes a larger map display, a combat simulator, empire management tools, graphical info screens, and queued construction orders.

As a comparison, all of these features would be expected in a similar boxed, retail strategy game, such as CIVILIZATION 4. However, by withholding their best, the designers are walking a dangerous line here as players could be turned off by the purposely crippled interface. For example, in TRAVIAN, each of the player's towns can construct only one upgrade at a time. Thus, players are encouraged to visit their towns every time an upgrade is finished, and as each upgrade might take half an hour, players may need to check the site many, many times each day just to keep pace with their competitors. A simple order queue would fix this problem, but the designers purposely decided to offer this feature only to players willing to pay for the PLUS version.

Whether this decision was right or wrong remains an open question, but perhaps a more important one to ask is who made this decision: game designers or businesspeople? Does it even make sense to think of these two groups as being different in a world where every element of a game can be given a price? Without a good balance of the needs of profit and fun, free-to-play games will feel either like a con job designed to suck away all of player's money (as with ZT ONLINE) or a charitable endeavor that never acquires the resources needed to develop and grow. However, when facing a difficult decision, one should always err on the side of providing the best free content possible. Greedy developers looking to maximize profits in the short-term risk losing their evangelizers willing to spread the word about a great game which is genuinely free-to-play.

A FREE MARKET SOLUTION

One interesting way to solve this problem—pioneered by Korean companies like Nexon—is the dual currency system, which lets the free market manage the balance. As an example, the Java-based MMO PUZZLE PIRATES employs such a system to meet the needs of both players who are time-rich and players who are cash-rich.

One type of currency, pieces of eight (PoE), is earned by spending time playing puzzle games, while the other type of currency, doubloons, is bought directly with real money. A wide variety of items are available for purchase, with effects ranging from aesthetic changes to in-game upgrades. However, as items often cost both types of currency, players who cannot afford to buy doubloons can trade for some by giving their PoE to cash-rich players. These latter players may need the PoE because they don't have the time to spend earning it by playing puzzles for hours. By allowing players to freely trade the two currencies, the designers have created multiple paths to earning any single purchasable item.

Thus, the designers avoid the balance issues faced in TRAVIAN by making sure all content and features are available to all players, whether they are willing

to spend money or not. In fact, when a time-rich player trades for doubloons, the cash-rich player is essentially "sponsoring" her peer—every doubloon spent in PUZZLE PIRATES earns the developer money, whether the doubloon is spent by the original purchaser or not. A natural free market dynamic keeps the two sides balanced. If too many time-rich players flood the game, the value of PoE will plummet, tempting players on the bubble to spend a little cash to take advantage of the low prices. Thus, with the help of the auto-balancing market forces of the dual currency system, the designer's goal simply becomes creating a compelling experience that keeps people playing the game.

Even Giant Interactive is beginning to understand the limitations of the soak-the-rich design of ZT ONLINE. The publisher is developing a subscription-based version of ZT (without the casino-style treasure chests) that is being launched for the low-income market not happy about playing a game full of rich players who have bought their way to the top. Another game they are publishing, Giant Online, aims for the middle-income segment by allowing RMT but adding spending caps to prevent a monetary arms race.

These developments are welcome because the free-to-play format holds great promise. Free-to-play games have a much larger potential audience than their fixed-cost counterparts because of the former's ability to satisfy different levels of player commitment, in terms of both time and money. Further, the potential for innovation is greater because consumers are no longer required to make a leap of faith when making a large, up-front retail purchase. However, the challenge of developing free-to-play game is that being "just" a game designer is no longer sufficient. Success, both in terms of profit and popularity, will be determined by how well the game design matches the business model. ❖

REFERENCES

TOP 10 FREE TO PLAY PUBLISHERS
by Adam Martin
www.freetoplay.biz/2008/09/08/top-10-free-to-play-publishers/



JESSE HARLIN

❖ AURAL FIXATION

GANG SIGNS

The Audio Club Climbs on the Networking Wagon

OF ALL THE IDEAS THAT HAVE SPRUNG from George Sanger's annual barbeque/ think tank, perhaps none has achieved as much notoriety as the Game Audio Network Guild, more commonly referred to by its acronym GANG. Originally designed as a means for game audio professionals to network between trade shows in a manner similar to the Film Music Network, GANG quickly expanded into educational partnerships, gear discounts, and an annual awards show for all things interactive audio.

Seven years on, however, the grand vision of GANG had started to atrophy. The web site was sparsely updated, the forums stagnated into a handful of devoted regulars engaged in little more than demo critiques, and networking get-togethers dwindled to annual events that coincided with GDC. With member retention declining, something had to change.

VERSION 2.0

GANG president Paul Lipson, in partnership with colleague and Sun Microsystems veteran Steve Years, labored in secrecy for months on a massive redesign of the entire GANG site that would better reflect his vision for the organization. At its core, Lipson wanted the site to "support the needs of the global community in terms of communication, social interaction, and resource sharing."

When www.audiogang.org relaunched on June 6 this year, GANG had been transformed into a social networking site along the lines of Facebook or LinkedIn. While the structure and function of the organization remain the same, the new GANG site now offers each member his or her own profile page complete with space for credit lists, their own blog, and 60MB of storage space for a 10-song, Flash-based demo player.

It's essentially MySpace for game audio professionals. Users can now browse the contacts of other members and send connection requests, personal messages, and comments to members. There are also communal areas for posting photos and video, as well as an arcade of open source games. Originally added simply as a nod to gaming past, the site recently held a gaming tournament to award all-access passes to Austin GDC to high scorers. The site's forums have been revamped as well.

So far, the reaction to Lipson's "labor of love" has been very positive from the GANG membership. Since the reboot, users have also been making use of the site's improved communication tools to organize and hold GANG Hangs, smaller regional get-togethers across the globe for the purposes of networking. The profile-based audio player has been a hit with many members commenting on how they simply surf through contacts listening to demos. Others have mentioned improved access to resources such as contract and invoice templates.

The biggest criticism of GANG is still the same one that has been levied against the organization for years, namely that while the organization is a great networking tool for those looking to break into the industry, it offers little to GANG's more experienced members. This in turn leads to a lack of involvement from high-end audio professionals and therefore

dissatisfied lower-tier members who hope to network with the bigger name members. In this regard, GANG still fights a battle for attention with forum-only sites like GameAudioForum.com or the gameaudiopro Yahoo! group.

VERSION 3.0 AND BEYOND

For Lipson, though, the new version of GANG's homepage is a starting point rather than an end result. Organizationally, GANG is expanding with what are being called "professional branches." The first such branch is the Interactive Entertainment Sound Developers branch, or IESD, an offshoot of GANG geared specifically towards Sound Designers and Sound Implementers that came about to combat the charge that GANG had become too composer-centric. Further plans exist to create a branch specific to game voice talent issues, as well.

Additionally, plans are afoot to overhaul the GANG Advisory Board structure. Lipson's goal is to "engage and energize the global Guild community." He intends to create Advisory Boards for North America, Europe, and Asia, which will help insure that the needs of the regions are respected, understood, and translated accurately online for the benefit of the entire membership. Regional directors are being established, such as the U.K.'s Richard Jacques, to create content and events specific to GANG members in those various global markets.

As for the site itself, Lipson isn't content to simply rest on recent additions and let GANG atrophy again. He envisions a whole host of improvements over the coming years, such as podcasting capabilities, "geo-targeted email blasts and newsletters," and functionality that will allow users to share their profile pages with the public as content-rich calling cards. ❖

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.

LEVEL UP

FIRST ACT INTERACTIVE
DESIGNS MUSICAL VIDEO GAMES
FOR PLAYERS OF ALL AGES.

BE PART OF THE FUTURE
OF MUSIC GAMING
AND TAKE YOUR CAREER
TO THE NEXT LEVEL.

Art Director
Leads/Artists

Game Designers
Programmers
Hardware Engineers

For more information:
firstact.com

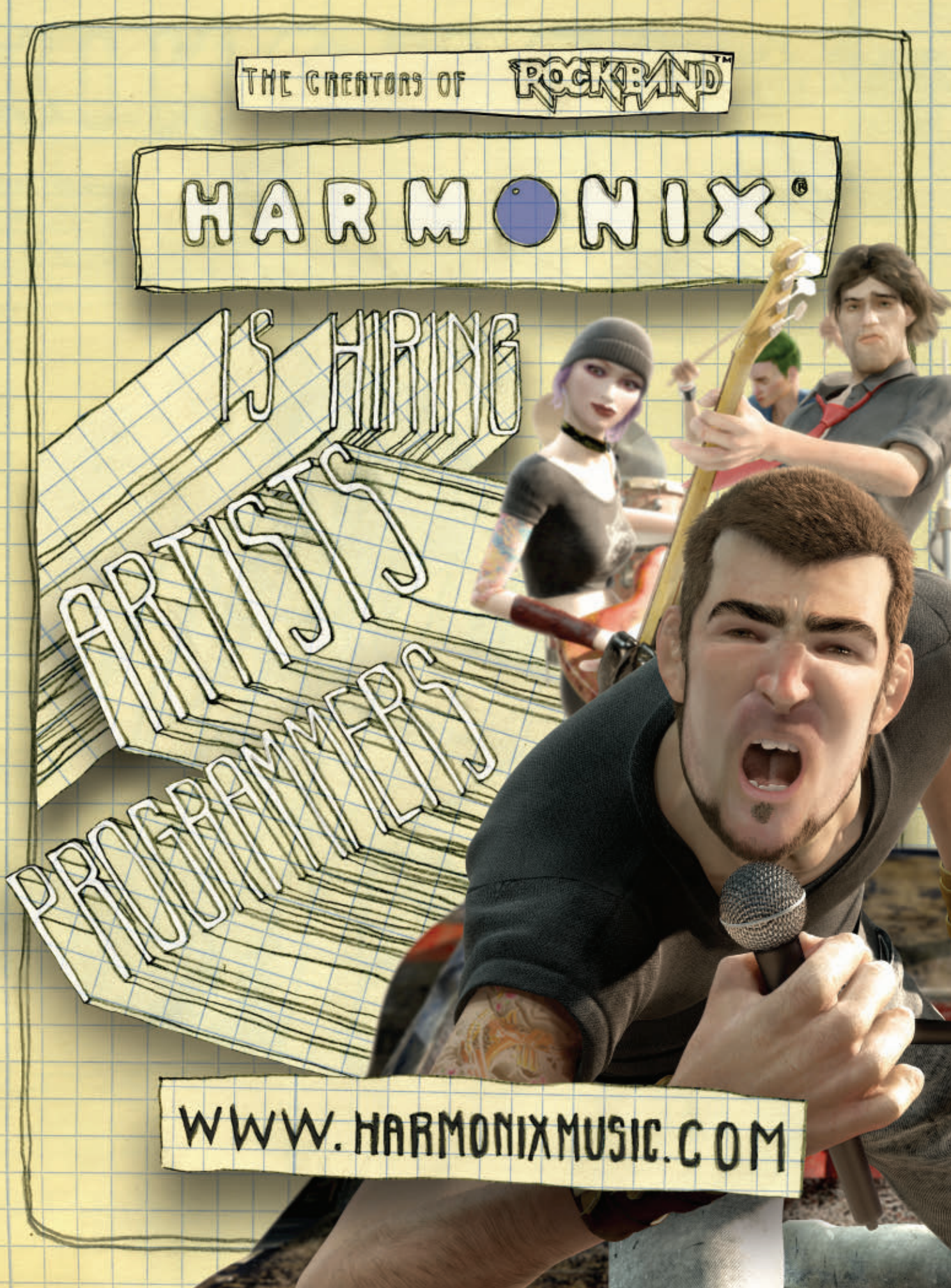
FIRST ACT
INTERACTIVE

THE CREATORS OF **ROCKBAND™**

HARMONIX®

IS HIRING
ARTISTS
PROGRAMMERS

WWW.HARMONIXMUSIC.COM



ACTIVISION®

GREAT GAMES

START WITH

GREAT PEOPLE



ARE YOU READY TO EXPLORE THE MOST SOUGHT AFTER JOBS IN ENTERTAINMENT?



Visit our site: activision.com



THE MAKERS OF EVE ONLINE ARE ACTIVELY RECRUITING LEADING TALENT

» ATLANTA - REYKJAVIK - SHANGHAI «

QA MANAGERS
ARTISTS
ANIMATORS
SOFTWARE DIRECTOR
SENIOR PROGRAMMERS

**VISIT US AT BOOTH #600 AT THE AUSTIN GDC.
MEET WITH CCP DEVELOPERS AND LEARN MORE ABOUT THE COMPANY!**

VISIT US ONLINE AT WWW.CCPGAMES.COM/JOBS TO LEARN MORE AND APPLY TODAY



Studios located in:
 Austin, TX
 Boston, MA
 Emeryville, CA
 Eugene, OR
 Irvine, CA
 Kirkland, WA
 Pune, India
 Sheffield, UK

Superheroes Wanted.

Cape fittings already under way. To join the team, go to:

www.f9e.com/jobs



NOW HIRING

- Producer
- Project Manager
- Lead Game Designer
- Lead Level Designer
- Project Art Director/Lead Artist
- Studio Art Director
- Special Effects Artist
- Senior Engine Programmer

More information can be found at:
www.edgeofreality.com

taking games
to the edge

Champlain College, located in Burlington, Vermont, seeks engaged professionals passionate about games to teach in an exciting, ground-breaking major. The program is recognized for its outstanding labs, state-of-the art software, team oriented faculty, and strong, innovative project-based curriculum patterned after and designed with game development industry input and support. Responsibilities for both positions include curriculum and course development, student advising, instruction and enrichment, industry outreach and development of program presence. A terminal degree in a creative field and evidence of effective teaching is preferred, but not required. Rank is contingent upon the qualifications of the successful candidates. We are currently accepting preliminary applications for both faculty positions.

Assistant/Associate Professor of Game Design

The successful candidate will have industry experience in the field of game design and production, specifically level design. Skills should include working in a game engine such as Unreal, Torque, Virtools or similar; level design, construction and scripting; working in a 3D pipeline; and some experience using Maya or other similar 3D package.

Assistant/Associate Professor of Game Art & Animation

The ideal candidate has professional experience in the field of 3D art and animation for games, with an emphasis and knowledge of the game production process. Must be an expert in Maya or 3ds Max; have experience in creating the visual design of level environments, and modeling, texturing and lighting those environments; have a solid understanding of color theory, lighting, surface textures, and color palettes; and will also have a highly developed aesthetic sense and ability to critique. Candidates who have experience in digital sculpting packages, use of shaders, and normal maps; a 3D pipeline for Torque and/or Unreal are preferred.

To apply please submit a cover letter, vita or resume to www.champlain.edu/hr. If you are sending a portfolio or reel for the Game Art & Animation Search, please send to: Champlain College, Human Resources Attn: Art & Animation Search, PO Box 670, Burlington, VT 05402-0670



Champlain College values, supports and encourages diversity of backgrounds, cultures and perspectives of students, faculty and staff.

Create the Game

Gaming Degree Programs for the Next Generation

Game Art

Associate's Degree Program

Game Development

Bachelor's Degree Program

Game Design

Master's Degree Program



ANIMATION | DESIGN | ENTERTAINMENT BUSINESS | FILM | RECORDING ARTS | SHOW PRODUCTION | VIDEO GAMES | WEB

Master's | Bachelor's | Associate's Degrees

800.226.7625 • 3300 University Boulevard • Winter Park, FL 32792
 Financial aid available to those who qualify • Career development assistance • Accredited University, ACCSCT

fullsail.edu

Online Programs Available



FULL SAIL
UNIVERSITY

MAKE MORE ENEMIES

Game Design at Vancouver Film School shows students how to make more enemies, better heroes, cooler levels, and tighter connections to the industry.

In just one year, you'll learn every aspect of game design. Your portfolio project is a playable video game.

VFS grads get snapped up by top game companies like BioWare, Radical, Relic, and EA Mythic.

VFS™

vfs.com/enemies



VFS student work by Jeff Plamondon

Serious Game Design Institute

SANTA BARBARA CITY COLLEGE

Do you want to work on a Serious MMO?
Online Production Internship Program

Visit: sgdi.sbccc.edu for current opportunities

CREATE your WORLD



**CERTIFICATE PROGRAMS | 3D ANIMATION+INTERACTIVE MEDIA
GAME ART+CHARACTER ANIMATION**

Intensive full- and part-time programs for the skills and tools you need to turn your ideas into reality. Financial assistance and career services available. *Apply today!*

CALL 800-808-CDIA EMAIL INFO@CDIABU.COM WEB WWW.CDIABU.COM



(newbie)



(alpha geek)



You can talk the talk.
Can you walk the walk?
Here's a chance to prove it.
Please geek responsibly.

LEARN:

- | | |
|-----------------------------|-----------------------------|
| DIGITAL ANIMATION | GAME PROGRAMMING |
| DIGITAL ART AND DESIGN | NETWORK ENGINEERING |
| DIGITAL VIDEO | NETWORK SECURITY |
| GAME DESIGN | SOFTWARE ENGINEERING |
| ARTIFICIAL LIFE PROGRAMMING | WEB ARCHITECTURE |
| COMPUTER FORENSICS | ROBOTICS & EMBEDDED SYSTEMS |

www.uat.edu > 877.UAT.GEEK
877.828.4335

ADVERTISER INDEX

COMPANY NAME	PAGE	COMPANY NAME	PAGE
3Vis	6	NaturalMotion	3
Activision	50	Paynova	27
CCP North America	51	Pocket Soft	43
Center for Digital Imaging	55	Rad Game Tools	CV4
Champlain College	53	Replay Solutions	CV2
Edge of Reality	52	Santa Barbara City College	54
Epic Games	11	Simutronics	13
First Act	48	Take 2 Interactive	21
Foundation 9 Entertainment	52	Techexcel	25
Full Sail Real World Education	53	University of Advancing Technology	55
Harmonix Music Systems	49	Vancouver Film School	54
Havok	CV3	Versant Corporation	31, 32 & 33

Game Developer (ISSN 1073-922X) is published monthly by United Business 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 United Business 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*, 4601 W. 6th St. Suite B, Lawrence, KS 66049. Call toll-free in the U.S./Canada (800) 444-4881 or fax (785) 838-7566. All other countries call (1) (785) 841-1631 or fax (1) (785) 841-2624. Please remember to indicate *Game Developer* on any correspondence. All content, copyright *Game Developer* magazine/United Business Media LLC, unless otherwise indicated. Don't steal any of it.



ARRESTED DEVELOPMENT

FAMOUS E-MAILS FROM GAME DEVELOPMENT HISTORY

Subject: Re: The front of the console has three red lights on it
Date: 04/05/05 16:41 GMT -8

Hmm, that's strange. I have a lot on my plate right now, but I'll look into it just as soon as I get a free moment.

> I am seeing this problem on some of the prototype units where three red lights come on in the front ring. Seems like some kind of hardware issue? Just a FYI ...

Subject: Hi there!
Date: 12/03/07 03:30 GMT -8

Well, I have to say the news was a bit of a shock at the office here, but that's life—always throwing surprises at you!

I figured some introductions were in order. I'm the guy in charge of everything here at our publishing group that's not Blizzard, Spyro, or Crash Bandicoot—in other words, all the interesting stuff! ;)

I'm really excited about what the resources of our combined companies can do for each of the titles in my portfolio. Let me know when you want to get together to discuss them.

Looking forward to working with you!

Subject: Re: NeoGAF ...
Date: 06/24/08 09:22 GMT -5

No, the die is cast. These jerks will soon realize the error of their ways ...

> Just between friends here, and if I may speak frankly ... why don't you just give it a rest? The game is out soon [finally!!], and it will speak for itself. There's no good to come of trolling the message boards, surely?

Subject: Re: Revolution production is go!
Date: 05/10/06 16:41 GMT -8

Just don't make too many... the LAST thing we want is a bunch of unsold inventory sitting around!!

> Looks like the manufacturers are ready to start! I need initial target volumes for each territory, please.

Subject: Re: Controls
Date: 02/10/07 16:41 GMT -8

We need to show that Sixaxis motion controls are "the" next-gen feature. Just focus and do your job, it'll be fine.

> Hey, so we don't think the dragon controls are working very well. People are having a lot of trouble getting it to go where they want. We've all been working really hard on this game and we just think the controls are the one thing we can't afford to mess up here. What about adding an option to not use the motion controls? We don't think it'd be that bad to at least give players the choice ...

Subject: Re: Blue laser diodes
Date: 08/10/06 16:41 GMT +9

Well, this is awesome. Why don't you tell me how I'm supposed to release this console now? Who is going out there to explain to the families all over the world that have been working overtime to afford one that they can't have it because of a tiny little part that YOU were supposed to be making?

There was a time once when Japanese engineers possessed COMPETENCE and this never would have happened. I hope you fully appreciate how your pathetic bungling has shamed this whole company. Japan's electronics leadership isn't long for the world. How depressing.

Ugh, I need a vacation.

P.S. Never e-mail me again.

> Sir, sorry about this, but it seems like we will not be able to produce the number of blue laser diodes you have asked for. It's a delicate process, compounded by the complexity of the new materials needed for the Blu-ray laser. I'm really sorry. I know it's important.

Subject: Re: Back from Dallas
Date: 08/11/08 20:15 GMT -8

Haha, yeah, I was wondering the same thing. I guess it was a PC game series that was really highly regarded and sold really well. Whatever.

> What was the "Age of" something or other those devs kept going on about when we were arguing with them? I've never heard of it. ❄

MATTHEW WASTELAND is a pseudonymous game developer who has a fairly common first name. Email him at mwasteland@gdmag.com.

Let us take care of your clothes.



cloth.havok.com



Bink is the best video codec for games! It provides amazing quality, super fast performance, and a great audio codec all in a simple, clean API. It is available for all platforms, including Windows, Xbox 360, Sony PSP and PS3, and Nintendo Wii.



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



The **Miles Sound System** is the most sophisticated and robust sound API available. It includes 2D and 3D sound, streaming, multichannel audio (5.1 and up), Internet voice chat, Ogg, ADPCM, and the world's fastest MP3 decoder (patent rights included). It is also available for all platforms, including next-gen consoles.



Pixomatic is a powerful software renderer for PCs. Think of it as a high-end DX7 video card in software form! MMX, 3DNow, and SSE optimized assembly are generated at run-time, so you don't have to let low-end hardware or bad video drivers hurt your game's sales!



WARNING: These products can make you more efficient.
425-893-4300 www.radgametools.com