



United Business Media

PROFESSIONAL CAREER ADVANCEMENT ISSUE

MAY 2008

game developer

THE LEADING GAME INDUSTRY MAGAZINE

» **NEW JOB, NEW LIFE**

WHO'S HIRING WHO IN THE GAME INDUSTRY?

» **ENGINES RUNNING**

A SIDE-BY-SIDE LOOK AT TODAY'S GAME FOUNDATIONS

» **INTERVIEW**

BUNGIE'S MIKE ZAK TALKS ART AND ENVIRONMENTS

POSTMORTEM:
HARMONIX'S
ROCK BAND



CREATE

In *Assassin's Creed*, Ubisoft used Autodesk® 3ds Max® software to create a hero character so real you can almost feel the coarseness of his tunic.

ANIMATE
Autodesk®

MotionBuilder™ software enabled the assassin to fluidly jump from rooftops to cobblestone streets with ease.

INTEGRATE

Using Autodesk® HumanIK® middleware, Ubisoft grounded the assassin in his 12th century boots and his run-time environment.

**HOW UBISOFT GAVE
AN ASSASSIN HIS SOUL.**

autodesk.com/Games

Image courtesy of Ubisoft

Autodesk, MotionBuilder, HumanIK and 3ds Max are registered trademarks of Autodesk, Inc., in the USA and/or other countries. All other brand names, product names, or trademarks belong to their respective holders. © 2007 Autodesk, Inc. All rights reserved.





PROFESSIONAL CAREER GUIDE

17 FEATURE The Game Developer Market

The job market is hot right now, especially for developers with experience. This state of the industry report aims to discern what companies are really looking for.

By Paul Hyman

25 DESIGN OF THE TIMES Idea Synthesis

Ruminations on collaborative design and humility as ways to get ahead as a game designer.

By Damian Schubert

28 AURAL FIXATION The Three Rs of Audio Leadership

An audio lead and a sound designer necessarily have different skillsets—three of which are outlined here.

By Jesse Harlin

29 ARRESTED DEVELOPMENT It's a Trap!

Tips for surviving in a manager-driven world.

By Matthew Wasteland



FEATURES

36 POSTMORTEM: HARMONIX'S ROCK BAND

Witness Harmonix's transformation from a game developer to a peripheral manufacture hybrid, as the company undertakes its most ambitious game to date—one which comes close to fulfilling the studio's ultimate vision.

From controller management to over-stretched leads, this postmortem chronicles the trials and tribulations of this innovative game.

By Rob Kay

7 ENGINES OF CREATION

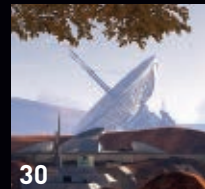
Third-party licensed game engines are increasingly used in the development of high-profile titles. But which is right for you? Here, we lay the major entries' features side-by-side.

By Jon Jordan

30 INTERVIEW: MIKE ZAK

HALO 3 environment artist Mike Zak talks designing art for curious players, HALO's unique look, and Bungie's new found independence.

By Christian Nutt



DEPARTMENTS

2 GAME PLAN *By Brandon Sheffield*
Share and Share Alike

4 HEADS UP DISPLAY
Engine talk, Nocturnal Initiative, used games research, and more.

43 TOOL BOX *By Sharan Volin*
Audiokinetic's Wwise

COLUMNS

46 THE INNER PRODUCT *By Noel Llopis*
Implementing Deterministic Playback Systems: Part 1

49 PIXEL PUSHER *By Steve Theodore*
Raw Crude

[PROGRAMMING]

[ART]



SHARE AND SHARE ALIKE

INSOMNIAC ANNOUNCED ITS NOCTURNAL

Initiative back at GDC, and we just recently got around to talking about it in *Game Developer* (See HUD, page 5), but it's really quite important. Sharing techniques and technology can only make the industry as a whole stronger, especially now that graphics are coming to a bit of a plateau—or are at least aren't as much of a talking point for the marketing folks.

I've long maintained that a lack of sharing is one of the big elements that has held the Japanese game industry back technologically, and the more we do share, the better the development experience becomes. We'll probably never get to a place where game development tools are truly standardized, but a common knowledge base is a good start in terms of getting everyone on the same page.

RAISON D'ÊTRE

This idea of sharing is the main reason why *Game Developer* and Gamasutra exist. They are both meant to help working developers share insights so that they may do their jobs better. But every so often, I get the occasional off-the-cuff comment from someone regarding a feeling that the publications cater much more to the student and aspirational crowd, versus the working professional.

It can be difficult for us to tell—we're not working in the trenches. But if this is indeed the case, the failure is shared. We rely on submissions from industry folks, and if everything we're getting is indeed "too kiddy," then that will certainly be reflected in the content. It's very interesting to see what happens when I turn the tables on that statement and ask the developer in question to write something good and appropriate for us, as they often feel as though they have nothing to contribute.

What's worse though, is when nobody speaks up. Quite often in our industry there are certain perceptions, unspoken rules, and standards that are followed. Most developers accept crunch time as an inevitability. When CEOs talk, their words are often shrouded in rhetoric. This is true of most industries, but ours is supposed to be comparatively straightforward and personable. A particularly painful trend is that most developers won't fire someone who's doing a bad job. Mediocre developers are allowed to languish in their positions, mismanage teams, and drift from company to company because they worked on a successful game, and because nobody wants to

speak ill of anyone else in this often-incestuous industry. Speaking up and telling the truth is an incredibly crucial kind of sharing.

This is also why what Insomniac is doing is so important. The company is sharing good, pertinent information [granted it all concerns the PlayStation 3, but that's a major pain point for lots of folks out there], and no matter what ulterior motive they may possibly have, they're doing a lot of work for everyone's benefit. That's what we need more of. If Ubisoft, Naughty Dog, 2K Boston, Blizzard, or Bungie were also to open up this directly in a public space, imagine how much learning could be gleaned!

Game Developer and Gamasutra are meant to be vehicles for this sort of thing. A blog, a company website, or developer forums are also appropriate venues, but a single hub of information would be the most efficient as a resource. Gamasutra may not be there yet, but it's one of the closest things to a database of collective game development knowledge out there. It will only function that way if people are willing to share their experiences, their technology, their business practices, and their ideas. We need to stop all solving the same problems. Each game has specific solutions, but with a springboard of what's worked in the past, couldn't you get going a lot quicker?

So with that in mind, I'd encourage anyone reading this to think about what you might have to contribute to the community. Something that would actually help other developers who might be facing problems you've faced—or even problems you're facing and working out currently. If you send them to me, I'll make sure they get seen. We'll all be better for it.

NEW KIDS ON THE BLOCK

I also want to welcome our new columnists this month. Taking over The Inner Product for Mick West is Noel Llopis, previously of Day 1 Studios and High Moon, and currently going indie with his own studio Power of Two. We've expanded the design column from one page to two, and now have two authors who will alternate by month. One is Soren Johnson, currently working on *SPORE*—you may have seen his column last month. The other is Damien Schubert, who is working on BioWare Austin's unannounced MMO. We've also added a humor column called Arrested Development, which we hope will bring some levity to our rather serious pages.

—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

SENIOR EDITOR

Brandon Sheffield bsheffield@gdmag.com

SENIOR CONTRIBUTING EDITOR

Jill Duffy jduffy@gdmag.com

ART DIRECTOR

Cliff Scorso cscorso@gdmag.com

PRODUCTION EDITOR

Jeffrey Fleming jfleming@gdmag.com

CONTRIBUTING EDITORS

Jesse Harlin jharlin@gdmag.com

Steve Theodore stheodore@gdmag.com

Mick West mwest@gdmag.com

ADVISORY BOARD

Hal Barwood Designer-at-Large

Ellen Guon Beeman Microsoft

Brad Bulkley Neversoft

Clinton Keith High Moon Studios

Ryan Lesser Harmonix

Mark DeLoura Ubisoft

ADVERTISING SALES

MEDIA ACCOUNT MANAGER

John Watson ejmwatson@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

SR. ACCOUNT MANAGER, EAST COAST, EUROPE & EASTERN CANADA

Cecily Herbst

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

AUDIENCE DEVELOPMENT

GROUP DIRECTOR Kathy Henry e: khenry@think-services.com

DIRECTOR Kristi Cunningham

e: kcunningham@think-services.com

LIST RENTAL Merit Direct LLC t: 914.368.1000

INTERNATIONAL LICENSING INFORMATION

Mario Salinas t: 650.513.4234 f: 650.513.4482

e: msalinas@think-services.com

UBM TECHNOLOGY MANAGEMENT

CHIEF EXECUTIVE OFFICER David Levin

CHIEF OPERATING OFFICER Scott Mozarsky

CORPORATE SENIOR VP SALES Anne Marie Miller

SENIOR VP, MANUFACTURING Marie Myers

SENIOR VP, COMMUNICATIONS Alexandra Raine



United Business Media

morpheme™

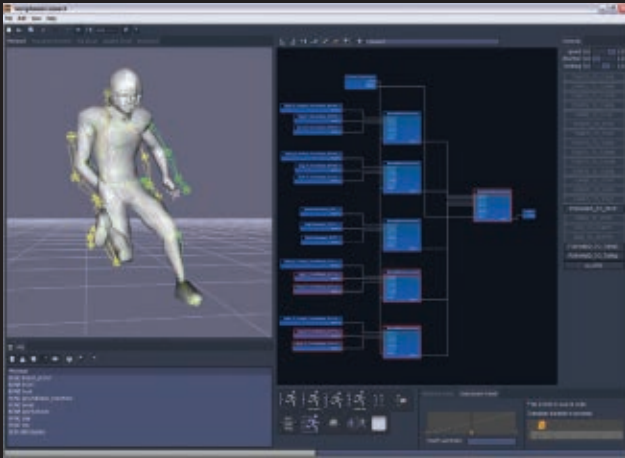
advanced animation system

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.

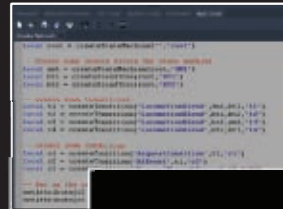
morpheme:runtime ships with full source code and integrates seamlessly with euphoria, NaturalMotion's Dynamic Motion Synthesis technology.

For more information, visit www.naturalmotion.com



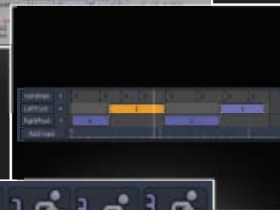
scripting

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



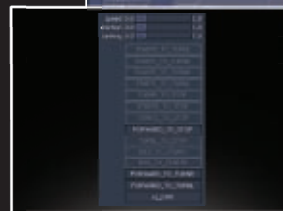
animation browser

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



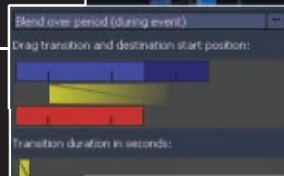
blend tree

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



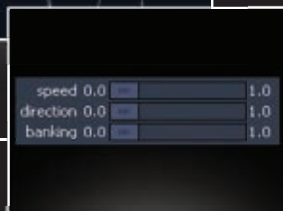
network

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



ENGINE TALK

AT THE 2008 GAME DEVELOPERS CONFERENCE WE SPOKE WITH A NUMBER OF GAME ENGINE COMPANIES. THEIR EXPERIENCE AND APPROACH TO THE business was diverse. Some license their engines as a simple way to recoup R&D investments while others have ambitious plans to create enterprise-level software. Everyone we talked to was eager to share their company's unique solutions to the complexities of game production. Here are a few excerpts from that larger conversation.

—Jeffrey Fleming



HARALD SEELEY
ENGINE BUSINESS MANAGER
AT CRYTEK

ON LICENSING VERSUS BUILDING YOUR OWN

"I don't think people want technology for technology's sake. People want the ability first to be very productive. It used to be that a one-year development cycle was common for a franchise and then maybe two years for something new. Now three years seems to be more and more the amount of time it takes to build products at this level, with this complexity, and this amount of assets. To the degree that they can spend all their time and effort building content and not worrying about

engineering problems is to the degree that we can also help them get a better product out."

ON BUILDING THE ENGINE BUSINESS

"The first version of the engine wasn't really optimized as middleware, we didn't think of it as middleware at the time we were building it. We were building it specifically to make *FARCRY*. But we learned a lot from that experience and from the experience of licensing it out to know that if we wanted to make something that was middleware we had to engineer it a little differently to make easier, a little bit more modular, a little bit more carefully structured, and a lot more documentation. So that's what we've done from the very beginning on *CryEngine 2*."



MARK REIN
VICE PRESIDENT OF EPIC GAMES

ON MIDDLEWARE

"Engine-related games are still a minority, not a majority. More games use their own tech than use our tech or somebody else's tech. But everybody uses middleware to some degree. Middleware is pretty proven. But the game engine middleware business is still a relatively small percentage of the overall game business."

GUARD, one of the very first games that licensed the engine, is a great example. In the original *UNREAL* we had this huge, giant, rock-throwing beast and in *KLINGON HONOR GUARD* there's this huge, giant, rock-throwing beast! It was clearly just our beast because you could see from *Unreal Script* that it was our beast code in a new body. Great! We have no problem with that whatsoever."

ON WHAT'S NEXT

"We have a research project going on for the next generation engine but that will closely follow the next generation of game consoles. There's going to be a huge paradigm shift in technology before that's going to be viable. So for many years to come we're very committed to *Unreal Engine 3*, to keep improving it and to keep making games for it, hopefully right up to the end of this generation. We don't know when that will be. It's a pure speculation. *UE4* is really just a research project right now. One guy on it. Tim Sweeney. But we have our best man on it."

ON INTELLECTUAL PROPERTY

"We encourage people, take our engine and make it your engine. In addition, we also give them access to our game code once we've shipped it as well. They're welcome to use our code. We don't want them using our characters and our textures and our content. But code wise—I mean, *KLINGON HONOR*



DOUG LOMBARDI
DIRECTOR OF MARKETING AT VALVE

ON THE ENGINE BUSINESS

"It's different than *Steam* and our content business. It's available and if people want to talk to us about it, we have people that are there and we manage the SDK for the Mod community and for the licensees and stuff like that. But it's not something that we're making a focal point. I mean, we're always revving the engine, always upgrading the engine and adding new features for our content. Then we add that to the SDK and there are people that are available to talk to for licensing and support. But it's just one of those things that we've sort of had as a complement to the rest of the business, it's not *the* business."

ON USING SOURCE TO DEVELOP GAMES IN OTHER GENRES

"I think *THE ORANGE BOX* shows that it's a pretty flexible tool. *TEAM FORTRESS* is wildly different from *PORTAL*, which is wildly different from *EPISODE TWO*. And we've seen some mods out there like *GARRY'S MOD*—that's kind of a game, I guess. It's more like this application tool thing or something. I don't know how to describe it. Those four alone show some of the diversity of it."

ON THE SOURCE DESIGN

"The whole idea and the goal was that *Source* was made to be this pool of technologies or systems that could be used to make a bunch of different things. That was the idea from the onset. The other basic philosophy that has been applied to it was that it could always be extended, modified, and upgraded. A lot of people say that *Source* is getting old but actually it has probably the fastest growing code base. It's getting new features in real-time rather than waiting for the big sequel release kind of thing."

SURVEYING THE USED GAME MARKET

NEWLY-GATHERED DATA ABOUT THE BUYING habits of the gamers who drive the \$1.3 billion used game market in the United States has been released, commissioned by consulting firm OTX and presented at this year's M16 Game Marketing Conference in San Francisco.

According to OTX, a third of the country's 75 million game consumers sell their games back to retailers. A significant majority of gamers, 60 percent, buy both new and used games—only 28 percent reserve all their dollars for new games, and a mere 3 percent confine themselves to used titles. Two thirds of used game purchasers buy their used software at retail chain GameStop, which depends heavily on the used game market.

There is a correlation between platform and used game sales—Wii gamers purchase only 8 percent of games used, while PlayStation 3 gamers purchase 12 percent used and Xbox 360 gamers purchase 16 percent used. It may not be a coincidence that these percentages roughly correspond with the length of time each console has been on the market.

Early adopters, both of hardware and software, are unsurprisingly more likely to invest in new games. About half of new games sold are purchased within a month of release, and half of those are purchased within a week. Those who own current-generation systems generally

have the fewest used games in their collections, and are the most likely to pre-order games.

The used game market is one heavily dominated by brick-and-mortar stores. According to OTX, 81 percent of used game buyers completed their transactions in a retail store, more than double the 35 percent in-store ratio for those who bought games new or used.

Of the 26 million game sellers, the great majority reinvest into more games. OTX found 19 million of that base to be "category re-investors," those who subsidize further purchases with used game sales; 16 million of those are "new game gluttons" who speed

through titles to retain the highest resale value and buy more new games.

What reasons do gamers give for selling particular games back to retailers? The biggest group, 62 percent, cited poor quality; almost as many, 59 percent, said they finished the games in question; and 43 percent said they were looking to scrounge up money for new games.

Despite the size of the used game market, OTX claims it is helping spur new game sales, driving \$415 million in new game revenue in 2007—five percent of the total \$8.6 billion U.S. game market.

—Chris Remo

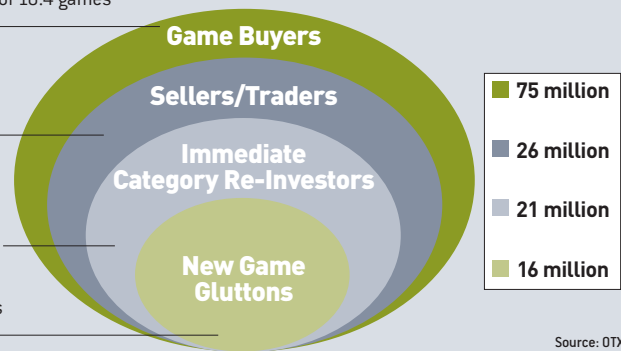
Most Sellers Put Their Profits Back Into The Industry

Purchased an average of 10.4 games each in the past year

Sold back or traded in an average of 6.7 games each in the past year

Sellers who sell their games to save towards **any** video game related products

Sellers who sell their games to save towards **brand new** games



Source: OTX

THE NOCTURNAL INITIATIVE

INSOMNIAC IS GIVING ITS CODE AWAY. THE DEVELOPER HAS LAUNCHED WHAT IT CALLS THE NOCTURNAL Initiative, which aims to ease the pain of development on the PlayStation 3 for all developers, as the console remains the most difficult platform on which to develop for most studios. The initiative was announced at GDC 2008, and has since become rather robust.

The Nocturnal Initiative includes code libraries, debugging helpers, and other releases, all of which are open-source and available for developers to use. To support the initiative, Insomniac has created plenty of technical how-to guides, and essentially opened up to other developers some elements of R&D for RESISTANCE 2, often with weekly updates.

As Insomniac says on the official site: "Nocturnal Initiative is not a game engine. The libraries provided here are potentially very useful for developing a game engine, but we want to avoid the 'all things to all people' that so often results in overly complex and/or under-performing monolithic engines. Instead, we want to provide a useful toolbox for the professional game developer."

Visit <http://nocturnal.insomniacgames.com> and www.insomniacgames.com/tech for more information.

—Brandon Sheffield

CALENDAR

Interfaces Conference

Troy Community Center

Troy, MI

June 14

Price: \$25

www.iadtdetroit.com/events.asp

NLGD Festival of Games

Utrecht, Netherlands

June 14–22

Price: 50–495 EUR plus VAT

www.nlgd.nl

Paris GDC 08

Cœur Défense

Paris

June 23–24

Price: 475–750 EUR

www.parisgdc.com

BAD GAME TOOLS NEED KIPPING APART



George Jeganathan, Studio Director,
Deep Red Studios Ltd;
formerly Lead Programmer, Codemasters

"My most hectic time was developing 3 projects on 5 different consoles at the same time: PS2, GameCube, Xbox, PC and PS1. The shortest development time was 4 months for a PS1 racing game using an engine and tool chain designed by me.

"I like developing with Q; it makes good sense. The main crossover point between artists and programmers is the content and Q provides me with an easy to understand framework to manage it the way I want.

"The example build processes are a great shortcut to a solid build process, which developers really need and often undervalue. After all, without a good build process, how can you deliver reliably to your publishers and other stakeholders?

"Q allows us to concentrate on the IP, not the nuts-and-bolts technology. But, more than that, Q also illuminates various development paths that are even more advantageous to our company. Put as simply as I can, I can write with a notepad and pencil on a bumpy train, and on a comfy seat in the garden with a word processor; when you present your work, would you rather it be a printed manual or a pencil filled notebook?

"I think it's wise to invest in technology that is flexible, not just says it's flexible. That is designed as middleware and not as a spin off from a successful title. That is designed to be used by others for any purpose they wish. I know that Qube have done a great job of delivering that product."

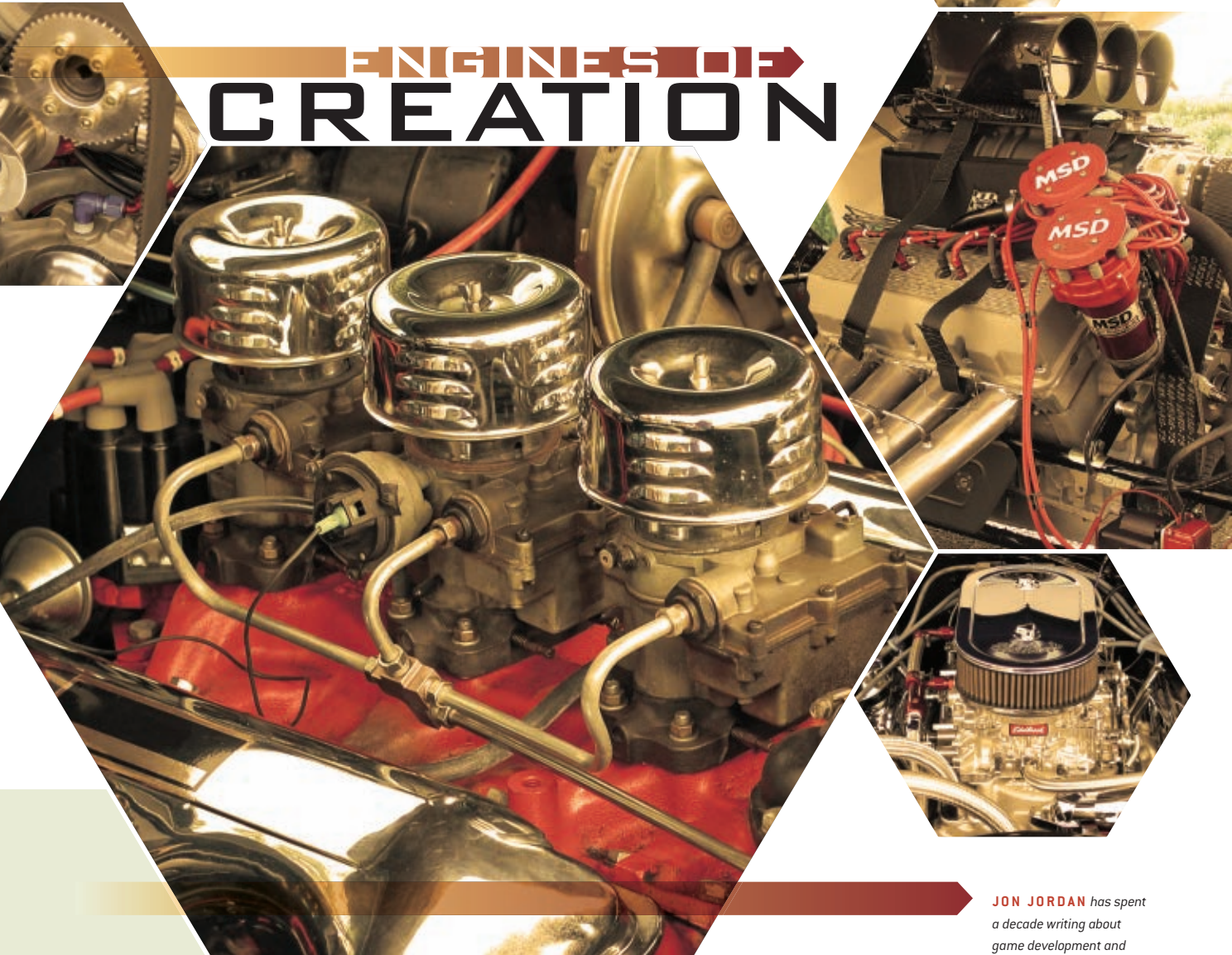


Q: future generation game framework and tools from the creators of Direct3D.

Try Q free today. For more information, contact Jamie Fowlston on +44 (0) 20 7431 9995, email info@qubesoft.com, or visit www.qubesoft.com.

- High-end runtime and tool features
- Extensible plug-in framework: cleanly customize anything
- For all platforms and any genre
- Share progress across games and license on to other Q developers

ENGINES OF CREATION



PERHAPS WHAT'S MOST REMARKABLE ABOUT TALKING TO the people behind the range of game engines listed in this article is their attitude toward each other. Despite there being clear competition between some of the technologies—try Unreal Engine 3, idTech 5, CryENGINE 2, and possibly Source—the main threat is still viewed as being the internal engine. Yet, it's the industry's growing acceptance of not-invented-here solutions that underpins the variety of middleware companies and wares now available to developers and publishers.

The success of such middleware has been aided by several different trends. Rising game budgets, the need to mitigate risk factors where possible and the robustness of most of these engines means that even at the top-end, paying a couple of million dollars to be able to get something up and running in weeks not months (or years) has become what could be described as the lesser of two evils.

Of course, some bastions of homegrown technology remain, certain in the knowledge they can come up with something more specific and suited to their actual project. Issues such as vendors being bought—need we remind people of the disaster that was EA's acquisition of Criterion [disastrous in terms of RenderWare licensees at least]—still linger. But more generally, the rise of middleware should be seen as a good thing. Less focus on technology and more time spent on gameplay was always the sales pitch, and maybe games such as *BIO SHOCK* and *MASS EFFECT* are proving it finally works.

Here we look at commercial game engines (listed alphabetically by company name) and give developers an overview of the off-the-shelf solutions available. Each company's approach to engine design is different and developers who are looking to build their next game on outside technology will find a wide variety from which to choose.

JON JORDAN has spent a decade writing about game development and middleware for publications such as *Edge*, *Develop*, *Gamasutra*, and *3DWorld*. Despite this he remains a fresh-faced innocent when it comes to C++. In fact, he'd probably get a D—. Email him at jjordan@gdmag.com.

CONTINUED ON PG 8



ENGINEERING CREATION

CONTINUED FROM PG 7

BIGWORLD

ONE OF THE FIRST COMPLETE ENGINES DESIGNED SPECIFICALLY FOR THE massively multiplayer online market, BigWorld developed out of research at Australia's longest-established game studio, Micro Forté. It has since spun-out into its own company and eight years on continues to prove popular both with Eastern publishers such as Japan-based GungHo Online and Chinese online giant Netease as well as US outfits like Cheyenne Mountain and John Romero's Slippgate Ironworks. Technology-wise, it's split into four basic components: a dynamically load-balanced server infrastructure that can supply a large, contiguous world; live server deployment and maintenance tools; a DirectX 9-class game client including integrated physics and AI; and a collaborative development environment including world, model and particle editors. The Python scripting language is an important element, enabling you to customise in-game elements without direct code. Future client features being worked on include new lighting and terrain engines as well as optimization for low-end hardware—something that is particularly important in terms of ensuring the widest possible audience for online games. More general planned improvements such as language support for Mandarin, Japanese and Russian, plus better terrain importers and exporters, are expected during 2008.



BIGWORLD TECHNOLOGY SUITE V1.9

FEATURES: Dynamic load-balancing server with web console tools such as cluster control and log viewer; client including terrain, animation, special effects, and path-finding engines; world, model and particle editors
PLATFORMS: Linux (server), Windows (client) plus support for Xbox 360, PlayStation 3 and mobile devices
INTEGRATION WITH OTHER TECHNOLOGIES: Diamondware, Scaleform, Spatial Voice, SpeedTree, Umbra, and Vivox, plus plug-ins for 3ds Max and Maya
COST: Includes up-front license, ongoing royalty and annual support fee
RELEASED GAMES INCLUDE: DARK AND LIGHT (Farlan), KWARI (Kwari)
GAMES IN DEVELOPMENT INCLUDE: HOUSE OF FLYING DAGGERS (T2CN), STARGATE WORLDS (Cheyenne Mountain), TBA (Slippgate Ironworks), TBA (VUG/Sierra Online), TBA (38 Studios), TIAN XIA 2 (Netease)
www.bigworldtech.com

CRYTEK

CRYENGINE 2 WAS HERALDED BY THE RELEASE OF THE PC SHOOTER CRYISIS in 2007 and one issue for German-based studio Crytek has been filtering enquiries from interested parties in order to ensure it's working with clients who can make the most of the engine's capabilities. Of course, the headline feature remains the engine's graphical quality, with the DirectX 10-class engine offering support for features such as real-time lighting and shadows that do not require pre-baked textures to create time of day and dynamic weather conditions. Also integrated is a flexible physics engine, which enables

the creation of fully destructible environments, and a realistic animation system that can combine motion capture and hand animation. But in terms of getting good games made quickly, CryENGINE 2 production tools are just as important. The Sandbox game editor offers a collaborative, real-time working environment for game designers and level editors. Tools include terrain editing, visual programming of features such as AI, special effects creation, facial animation, sound design, and asset management. This is important as versions of your game can be generated and tested on target platforms without the need for assets to be compiled. Indeed, the only obvious lack is support for Xbox 360 and PlayStation 3, something that's currently in development.

CRYENGINE 2 (v1.2.1)

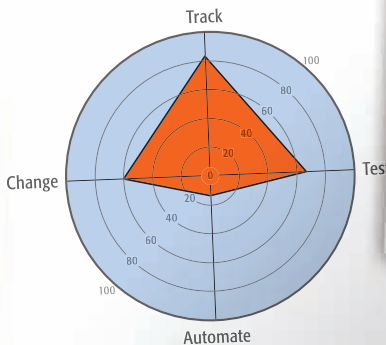
FEATURES: DirectX 10-class renderer with dynamic soft shadows; script-based shader system; asset streaming system, animation system; audio engine; integrated Sandbox 2 game editor including Lua-driven AI system and smart objects; low bandwidth networking
PLATFORMS: PC (Xbox 360 and PS3 in development)
INTEGRATION WITH OTHER TECHNOLOGIES: Alienbrain, CRI, FMOD, Perforce, and Scaleform, plus plug-ins for 3ds Max, Photoshop and XSI
COST: pre-visualization, SDK, and SDK plus source licenses available. Price on request
RELEASED GAMES: CRYISIS (Crytek)
GAMES IN DEVELOPMENT INCLUDE: ENTROPIA UNIVERSE (MindArk), MERCHANTS OF BROOKLYN (Paleo), THE DAY (Reloaded), BLUE MARS (Avatar Reality)
www.cryengine2.com, www.crytek.com, www.crymod.com





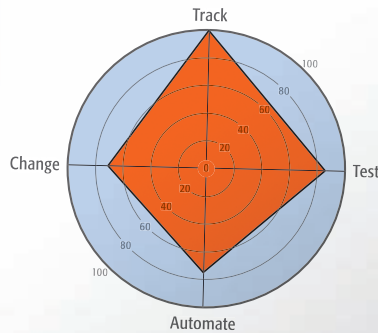
Seapine Software

Seapine Software Quality-Ready Assessment



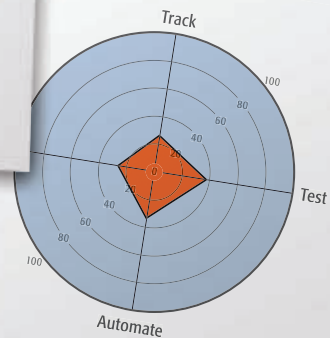
- Not prepared for an audit
- Little or no automation

Seapine Software Quality-Ready Assessment



- Audit-ready development artifacts
- Strong testing methodology and automation

Seapine Software Quality-Ready Assessment



- Lacking in most areas of quality
- Probably an "It's good enough" organization

What's Your Quality-Ready Score?

You need the right tools to build a quality-ready advantage. Take the Seapine Software Quality-Ready Assessment to measure your development and QA teams' ability to create a sustainable competitive advantage through integrated ALM and learn how to make your organization's productivity soar.

Find out your score at www.seapine.com/qra





ENGINEERING CREATION

EMERGENT

TRACING THE HISTORY OF ITS GAMEBRYO TECHNOLOGY, U.S. OUTFIT

Emergent can claim to offer the longest-lived piece of game graphics middleware. Back in the 1990s, when the company was called NDL, it first released its NetImmerse engine before this was eventually rebranded in 2003—something that happened to NDL itself in 2005. Of course, there's no lineage in terms of legacy code but NetImmerse's reputation as a flexible piece of technology that offered good integration with third-party tools continues with Gamebryo. Interestingly, both have also proved to be popular clients for MMOG developers with the likes of Mythic (now EA Mythic) and NCsoft as current licensees. This is perhaps due to the suite of other options that Emergent offers, which also include a server solution for online games. Another significant piece of technology is Floodgate, a multi-core streaming solution with particular focus for PlayStation 3 developers. As with other engine companies, Emergent has used its position in the industry to encourage smaller third parties to support Gamebryo by integrating their products into its Tech Connection Program. Finally, features for the summer

release of v2.5 include a re-architected geometry pipeline, a new terrain system, integrated GPU instancing and support for SoftImage XSI.

GAMEBRYO V2.3 (V2.5 JUNE)

FEATURES: High-end renderer including customizable shaders and object culling; Floodgate stream engine; particle system; scene designer and terrain editor; animation tool; asset viewer

PLATFORMS: PC, PlayStation 3, Wii, Xbox 360

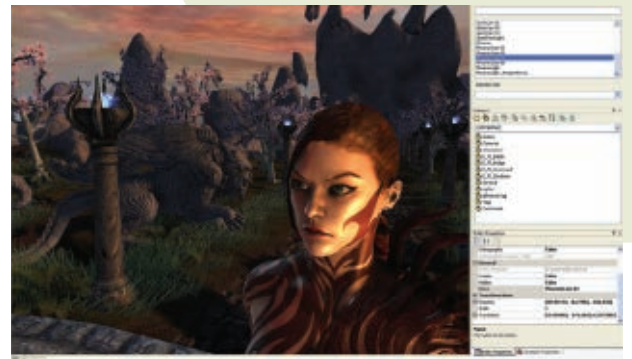
INTEGRATION WITH OTHER TECHNOLOGIES: AiLive, Anark, Bink Video, CRI, Kynapse, morpheme, Miles, PhysX, ProFX, Scaleform, Speedtree, Umbra and Wwise, plus plug-ins for 3ds Max and Maya

COST: Options include per SKU, per platform or site license. There is no royalty option.

RELEASED GAMES INCLUDE: CIVILIZATION IV (Firaxis), DARK AGES OF CAMELOT (EA Mythic), THE ELDER SCROLLS IV: OBLIVION (Bethesda), ZOO TYCOON 2 (Blue Fang)

GAMES IN DEVELOPMENT INCLUDE: FALLOUT 3 (Bethesda), TBA (NCsoft), TBA (Rockstar), TBA (Shanda), TBA (Sony Online), TBA (The9), WARHAMMER ONLINE (EA Mythic)

www.emergent.net



EPIC

IT'S SAFE TO SAY EPIC'S UNREAL ENGINE 3 IS THE CURRENT, DE FACTO

industry standard middleware when it comes commercial game development. Not only is it used on some level by most major publishers and developers, it's increasingly the favored client solution by MMOG developers while architects, film companies and educators are licensing it as well. There are two main reasons for this status. The first flows from Epic's historic reputation as one of the prime first person shooter studios thanks to games such as UNREAL and UNREAL TOURNAMENT. The second, which is more pertinent to Unreal Engine 3, is the engine's transformation from its original PC focus into a robust cross-platform technology, with the company claiming to have

the most popular engine across PC, PlayStation 3 and Xbox 360. It certainly comes with plenty of bells and whistles. As well as the DirectX 10-class game engine, which includes the 64-bit HDR rendering system Gemini, there's a flexible animation system which links into the engine's integrated physics, and the Cascade particle effects system. Development tools include the UnrealEd content creation suite, and the Kismet visual gameplay language and Matinee cinematic system, while the Java-like UnrealScript simplifies in-game programming.

UNREAL ENGINE 3 (BUILD 37XX)

FEATURES: HDR renderer including advanced shadowing; texture streaming system; modular material framework; integrated animation, physics and audio; Cascade particle system; Kismet gameplay scripting; Matinee cinematic editor, UnrealEd world editor; UnrealScript

PLATFORMS: PC, PlayStation 3, Xbox 360

INTEGRATION WITH OTHER TECHNOLOGIES: Beast, Bink Video, Digimask, DivX, Enlighten, FaceFX, GameSpy, Kynapse, morpheme, PhysX, ProFX, Rendez-Vous, Spark, SpeedTree, Voiceln

COST: Available on request

RELEASED GAMES INCLUDE: ARMY OF TWO (EA), BIOSHOCK (2K Boston), GEARS OF WARS (Epic), LINEAGE II (NCsoft), MASS EFFECT (BioWare), UNREAL TOURNAMENT 3 (Epic)

GAMES IN DEVELOPMENT INCLUDE: ALL POINTS BULLETIN (RealTime Worlds), BROTHERS IN ARMS 3 (Gearbox), HALO WARS (Ensemble), Various (2K), Various (Activision), Various (EA), Various (NCsoft), Various (Square Enix), Various (THQ)



GARAGEGAMES

IF ANY EXAMPLE WERE REQUIRED TO ILLUSTRATE THE EXPLOSION OF INDIE development combined with the rise of casual, downloadable games, Oregon-based GarageGames' Torque engine and development suite would be it. Initially created from the skeleton of the PC-based TRIBES 2 engine by ex-members of its development team at Dynamix in 2000, the Torque engine has since blossomed to offer various technological options for game makers ranging from hobbyists through to professional studios. This is reflected in the pricing options that are split into indie and commercial, depending on the size of the licensee. The simplest option is Torque X 2.0, which is designed for use as part of Microsoft's XNA Game Studio 2.0 for Xbox 360 and PC games, while a similar option for general PC and Mac games is the 2D drag and drop engine, Torque Game Builder. The 3D version of this is the Torque Game Engine, while Torque Game Engine Advanced includes features such as dynamic lighting, programmable shaders, a terrain rendering system and special effects such as water. Other flavors of these technologies are specifically available for Xbox 360 and Wii development in the form of Torque 360 and Torque for Wii.

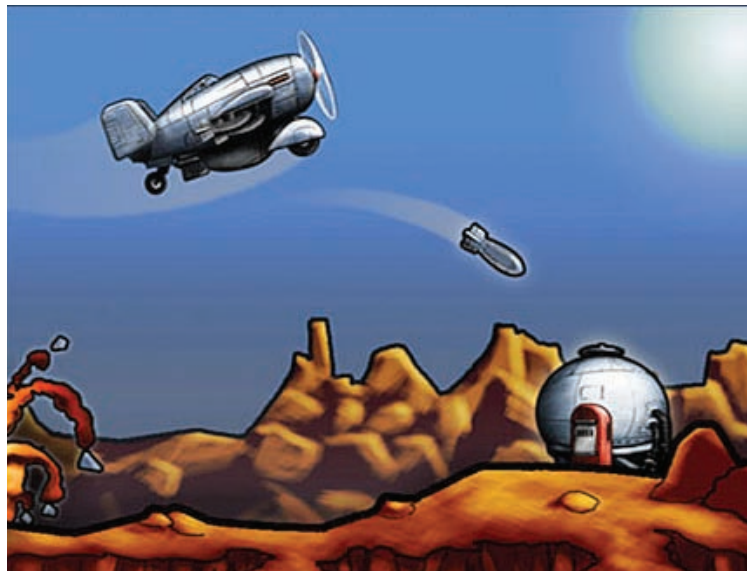
TORQUE / GAME / BUILDER / ENGINE / ADVANCED / X / 360

FEATURES (TGEA): DirectX 9-class renderer including procedural generated shaders and custom materials; Puppeteer Mesh animation system, terrain generation engine, TorqueNet networking, TorqueScripting language; Mission Builder editor

PLATFORMS: Linux, Mac, PC, Wii, Xbox 360

INTEGRATION WITH OTHER TECHNOLOGIES: Bullet Physics, OGRE, Havok, PhysX, Scaleform and Umbra, plus plug-ins for 3ds Max, Maya, Houdini and XSI

COST: ranges from \$100 per seat to \$1,495 commercial license depending on technology. No royalties.



RELEASED GAMES INCLUDE: CURSE OF THE PHAROAH (Ph03nix), CYCLOMITE (Widload), MARBLE BLAST ULTRA (GarageGames), THINK TANKS (BraveTree), WILDLIFE TYCOON (Pocket Watch)

GAMES IN DEVELOPMENT INCLUDE: PENNY ARCADE ADVENTURES (Hothead), LEGIONS (GarageGames), METAL DRIFT (Black Jacket), TBA (EA), TBA (NCsoft), TBA (Ubisoft), TBA (Vivendi)

www.garagegames.com

ID

IDTECH 5 WAS PUBLICLY UNVEILED BY JOHN CARMACK DURING THE APPLE Worldwide Developers Conference 2007 as the next-generation game engine from Texas-based id Software. Of all the engines featured in this roundup, it's the one about which least is known as no games have yet shipped using the technology. The first to do so will be id's RAGE, a vehicle combat game involving large, exterior areas. In this respect, one of the most important features of idTech 5 is the MegaTexture system—a high-quality streaming technology which treats environments as one very large texture rather

than breaking it down into tiled components. Another element highlighted is the collision system which id claims prevents the typical geometric interpenetrations and collision errors currently seen in games. Linked into the game engine is the real-time idStudio, which interfaces with the system's development tools and editors, ensuring data consistency and integration with source control solutions. IdStudio also allows you to run game content practically instantaneous on all supported target platforms. In keeping with other id engines from QUAKE onwards, IdTech5 supports the OpenGL graphics standard, ensuring core cross-platform support across Mac, PC as well as Xbox 360 and PlayStation 3.

IDTECH 4, IDTECH 5 AVAILABLE ON A SELECT BASIS

FEATURES: HDR rendering; soft shadows; integrated physics, animation, AI and audio engines; MegaTexture system; idStudio development suite

PLATFORMS: Mac, PC, PlayStation 3, Xbox 360

INTEGRATION WITH OTHER TECHNOLOGIES: interfaces for Alienbrain and DevTrack, plus plug-ins for 3ds Max, LightWave 3D and Maya

COST: idTech 4, \$250,000 guarantee against a 5 percent royalty; idTech 5, available on request
RELEASED GAMES INCLUDE: idTech 4: DOOM 3 (id), QUAKE 4 (Raven), PREY (Human Head), ENEMY TERRITORY: QUAKE WARS (Splash Damage)
idTech 5: none

GAMES IN DEVELOPMENT INCLUDE: idTech 5: RAGE (id)

www.idsoftware.com/business





ENGINEERING CREATION

QUBESOFT

Q ENGINE IS THE RESULT OF A DECADE-LONG DEVELOPMENT

process carried out by DirectX founders Servan Keondjian and Doug Rabson following their departure from Microsoft. Setting up Qubesoft in North London, they decided to create a from-the-ground-up piece of middleware with particular attention placed on the flexibility of the technology. To that extent, Q is built as a cross-platform extensible plug-in framework that enables you to customize and add game-specific technologies as you see fit, without modifying existing source code. Features include a texture manager capable of handling large amounts of data, n-dimensional animation blending and data streaming. The engine also links into the real-time QStudio editing system that lets you run custom plug-ins live while you are developing and

testing your game. It's still an early stage for Q, officially launched at GDC 2008, although it is being used internally by Qubesoft as well as by a number of smaller game developers and virtual world builders. Ongoing work includes broader integration with other middleware providers, improved debugging for massively multi-threaded script applications, enhanced animation tools and an expanded library of ready-to-use shaders.

Q 2.0

FEATURES: Arbitrary scene rendering algorithms, programmable shaders; background data streaming, texture manager; cross-platform data format; n-dimensional animation blending; background work queue; integrated QStudio

development environment

PLATFORMS: Linux, Mac, PC, PlayStation 3, Wii

INTEGRATION WITH OTHER

TECHNOLOGIES: Visual Studio 2008

COST: An annual support fee plus shipping fee per SKU

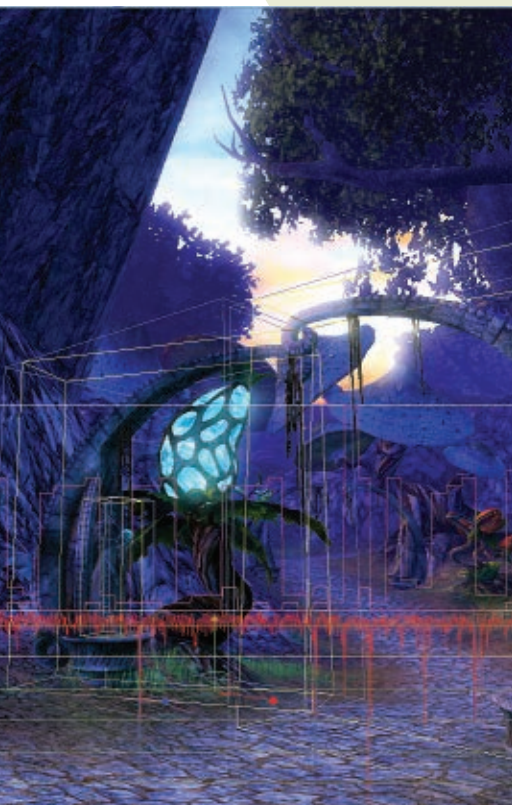
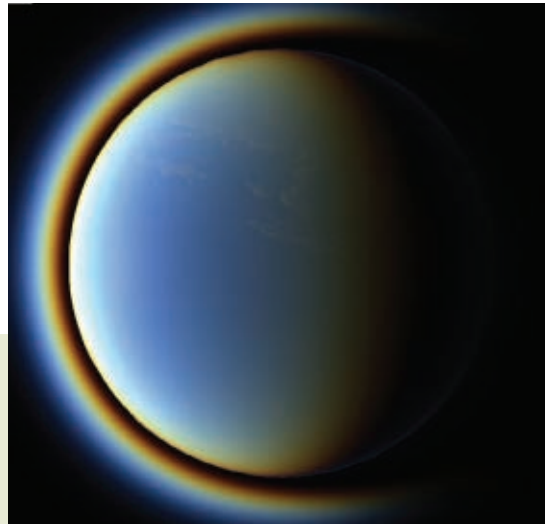
RELEASED GAMES

INCLUDE: EARTHSIM (Earthsim), LEGO DIGITAL DESIGNER (Qube)

GAMES IN DEVELOPMENT

INCLUDE: NEAR (Near)

www.qubesoft.com



SIMUTRONICS

BUILT ON THE BACK OF EXPERIENCE STRETCHING INTO THE DAYS OF TEXT-BASED ONLINE GAMES, U.S.-

based Simutronics' HeroEngine takes a different approach when it comes to the creation and deployment of massively multiplayer online games. Hero Engine was launched in 2006, and after five years of work, it provides an integrated server-client engine and development system, both of which support a dynamic plug-in architecture that enables a collaborative and 'always live' environment for the fast prototyping, building, and testing of games. This toolset, called HeroBlade, features components such as a world builder, particle and special effects editors, a character and animation system, an audio engine, and an internal scripting language. This is linked into a live asset repository for intelligent objects and the DreamManager project management system and quality assurance tools. Further improving the ability for collaboration is a real-time whiteboard tool so you can make notes directly onto in-game levels. Finally, the server architecture itself is designed to handle seamless environments, instanced worlds, or variants between the two. Performance and player behavior metrics tools are also provided. Indeed, the only thing missing is a finished game as the length of time to make MMOGs means that no commercial products have yet been shipped using HeroEngine.

HEROENGINE

FEATURES: HeroBlade development tool; 'always live' client-server; world editor; intelligent objects; character animation system; HeroScript; DreamManager process management system

PLATFORMS: Linux (server), PC (client)

INTEGRATION WITH OTHER TECHNOLOGIES: Alseek, FMOD, PhysX, RTView, Scaleform, SpeedTree, StreamBase, Wwise and Vivox, plus plug-ins for 3ds Max and Maya

COST: Evaluation, prototype, basic and source licenses available, price available on request

RELEASED GAMES INCLUDE: none to date

GAMES IN DEVELOPMENT INCLUDE: TBA (BioWare), TBA (IT Territory), TBA (ZeniMax)

www.heroengine.com



TRINIGY

GERMAN MIDDLEWARE COMPANY TRINIGY MAY NOT BE A HOUSEHOLD NAME WHEN IT COMES TO GAME technology, but over the years it has steadily built up its presence from its central European heartland to pick up clients in Asia, the U.K. and the U.S. The engine features advanced lighting and shadowing features, with particular attention recently being paid to stream processing and multi-threading. Other components include integrated physics, animation, particle effects and audio. The next major release of the Vision Engine (version 7) will see significant improvement in areas such as visibility and occlusion processing, instance scripting and support for third-party tools. In term of game development, the Vision Engine links into the vForge editing framework; something that's designed to be extended and customized using the C# programming language so gameplay features can be exposed for artists and level editors. An important part of the system is the vLux light-mapping tool that enables you to pre-process and bake static lighting information into your textures. Interactive shader editing, the addition of dynamic effects, volumetric effects and camera paths can be carried out within vForge too, while future releases of the tech will include enhancements between the art pipeline and tools such as 3ds Max and Maya.

VISION ENGINE V6.4 (V7 DUE 2008)

FEATURES: HDR renderer supporting radiosity-based, normal-mapped illumination; integrated physics, particle, animation, cinematic playback and audio components, vForge development environment including vLux light mapping tool

PLATFORMS: PC, PlayStation 3, Xbox 360 (Wii in development)

INTEGRATION WITH OTHER TECHNOLOGIES: Bullet Physics, FMOD, Kynapse, ODE Physics, PhysX, ProFX, ScaleForm and SpeedTree, plus plug-ins for 3ds Max and Maya

COST: per-title/per-platform fee without royalties. Details available on request

RELEASED GAMES INCLUDE: ALARM FOR COBRA 11 (Exozet), BACK TO GAYA (4Head), HELLDORADO (Spellbound), THE SHOW (16Tons), UNDERCOVER (Sproing)

GAMES IN DEVELOPMENT INCLUDE: DUNGEON HERO (Firefly), GOTHIC 4 (Spellbound), WARLORD (Neowiz)
www.trinigy.net

learn
network
inspire

Paris Game Developers
Conference

08

The key meeting place for
European and international developers
focusing on building great games:

CORE & CASUAL

CONNECTION
EVENTS

THINK
SERVICES
A DIVISION OF UNITED BUSINESS MEDIA LLC

June 23-24, 2008

Coeur Défense, Paris, France

www.parisgdc.com

REGISTER EARLY AND SAVE UP TO 35% ON REGISTRATION BY MAY 14!

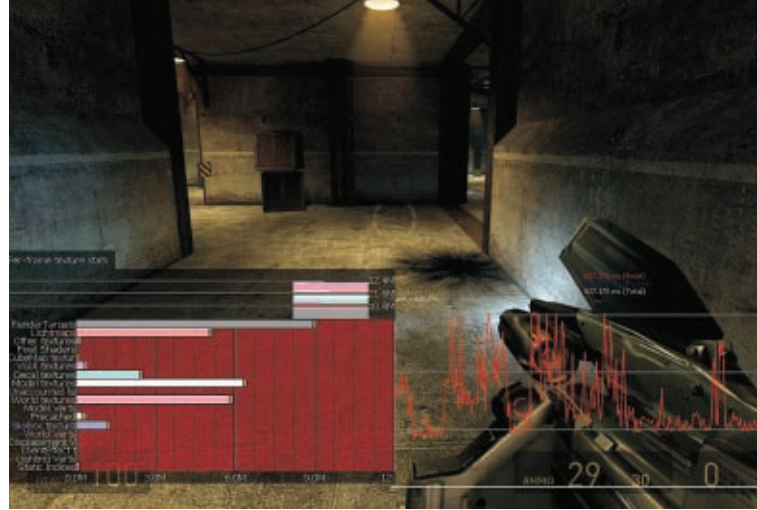


ENGINEERING CREATION

VALVE

THE SOURCE ENGINE WAS DEVELOPED IN THE YEARS FOLLOWING VALVE'S

initial licensing of the Quake engine for HALF-LIFE in the mid-1990s. The strengths of the Source engine reflect the focus the Seattle developer has placed on its own games: high-end graphical effects, facial and character animation, the integration of in-game physics, effective networking code and support for a wide range of PC hardware. Unlike other middleware companies, Valve seems to have taken a measured approach since it started licensing the engine in 2004, working with a small group of developers rather than the large publishers. This may be because of Source's PC-centric nature. Wii and PlayStation 3 are supported but licensees are warned such ports will require additional custom work. Yet, for PC and Xbox 360 development, Source offers a fully featured solution. The renderer supports dynamic shadows and HDR, the latter even on DirectX 9-class hardware, while indirect radiosity is supported using a distributed solver and then baked into textures. Other engine components include a unified material system covering textures, physical behavior and fracture properties; network-enabled physics; and a squad-based AI system. Development tools include the Hammer level editor, the Faceposer character animator and an audio suite.



SOURCE (BUILD 3412)

FEATURES: DirectX 10 HDR renderer with advanced shaders; material systems; particle engine; networked physics engine; pathfinding and navigation system; Hammer level editor; facial and character animation tool; Faceposer

PLATFORMS: PC, Xbox 360 (PlayStation 3 and Wii available but not officially supported)

INTEGRATION WITH OTHER TECHNOLOGIES: plug-ins for 3ds Max, Blender, Cinema 4D, FragMOTION, LightWave 3D, Maya, Milkshape 3D and Softimage XSI

COST: Available on request

RELEASED GAMES INCLUDE: COUNTER-STRIKE: SOURCE (Valve), DARK MESSIAH OF MIGHT AND MAGIC (Arkane), HALF LIFE 2 (Valve), PORTAL (Valve), TEAM FORTRESS 2 (Valve)

GAMES IN DEVELOPMENT INCLUDE: LEFT 4 DEAD (Turtle Rock Studios), MABINOGI HEROES (Nexon), POSTAL III (Akella), SALVATION (Black Wing), THE CROSSING (Arkane)

<http://source.valvesoftware.com>

VICIOUS CYCLE

CHAPEL HILL-BASED VICIOUS CYCLE LAUNCHED ITS VICIOUS ENGINE IN

2005 as the first technology to offer dedicated support for PSP. In part this arose from its own work on the PSP title, DEAD HEAD FRED, but since then, the first version of the engine has been extended for other platforms such as PC and Wii and is one of the newcomers to the middleware arena. The forthcoming Vicious Engine 2 will further expand support to include Xbox 360 and PlayStation 3. In order to handle these, new features have been added

such as a dynamic lighting system, including the ability to bake lightmaps, an animation blending system, a shader-based material editor, occlusion culling, and better pathfinding and AI functionality. More generally, one of the main features of the Vicious Engine is its data-driven point-and-click scripting, which the company claims makes it much easier for non-programmers to work with—something that is particularly important in terms of the fast turnaround projects for which it's typically used. Working in conjunction with this is a script editor, which enables you to debug gameplay using breakpoints and analyzing object variables and agent AI states. A built-in user interface development tool and an intelligent memory management system are also included.

VICIOUS ENGINE (VER. 2 DUE "SOON")

FEATURES: Dynamic lighting system; animation system; material editor; AI engine, scripting system and debugger

PLATFORMS: PC, PlayStation 3, PSP, Wii, Xbox 360

INTEGRATION WITH OTHER TECHNOLOGIES: none

COST: A combination of license fee and support fee on a per-platform basis. Price available on request

RELEASED GAMES INCLUDE: 300: MARCH TO GLORY (Collision), ALIEN SYNDROME (Totally Games), DEAD HEAD FRED (Vicious Cycle)

GAMES IN DEVELOPMENT INCLUDE: ELEMENTS OF DESTRUCTION (Frozen Codebase)

www.viciousengine.com





Unreal Technology News

by Mark Rein, Epic Games, Inc.

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 Front Line Award for Best Engine for the past three years, and Epic was awarded Best Studio at the 2006 Spike TV Video Game Awards. Epic's "Gears of War" won overall Game of the Year in 2006, and sold over 4.5 million units on Xbox 360. Epic recently shipped the PC version of "Gears of War" for publisher Microsoft Game Studios, as well as "Unreal Tournament 3" for PC and PlayStation 3 for publisher Midway.

Upcoming Epic Attended Events:

Nordic Game Conference
Malmö, Sweden
May 14-15, 2008

Sony DevStation 2008
London, UK
June 10-11, 2008

GameHorizon Conference
Newcastle, UK
June 18-19, 2008

E3 2008
Los Angeles, CA
July 15-17, 2008

Microsoft Gamefest
Seattle, WA
July 22-23, 2008

Casual Games Connect
Seattle, WA
July 24, 2008

Please email:
mrein@epicgames.com
for appointments.



INTEL AND EPIC LAUNCH '\$1 MILLION INTEL MAKE SOMETHING UNREAL CONTEST'

Epic has teamed up with Intel to launch the "\$1 Million Intel Make Something Unreal Contest".

This is a great way for aspiring game makers and mod teams to hone development skills, gain recognition and possibly even land a job. Many Epic employees have been recruited directly from the mod community. We always have openings, and this is a great way to get noticed!

Winnings include cash awards, high-end PCs and, for the grand prize winner, an Unreal Engine 3 license in recognition of *Unreal Tournament 3* mods that stand out in a range of categories, including level design, graphics, physics, characters, vehicles, weapons and more.

The previous "Make Something Unreal Contest" grand prize winner, Tripwire Interactive, earned over \$80,000 in cash and hardware prizes throughout the contest, as well as an Unreal Engine 3 license. Tripwire's World War II shooter *Red Orchestra* was later distributed on Steam and released to retail outlets worldwide.

The competition will entail four preliminary phases and a grand final, with judging starting in June 2008 and running through fall 2009.

Contestants may enter one phase or multiple phases, and the same mod may be submitted in multiple categories and phases.

For more information on the new competition, check out www.makesomethingunreal.com

ELECTRONIC ARTS BOLDLY INCREASES ITS COMMITMENT TO UNREAL ENGINE 3

EA has deepened its relationship with Epic by signing a deal that gives it the right to incorporate Unreal Engine 3 into "more than five" of its upcoming titles, in addition to games already released or under development per its initial Unreal Engine 3 license agreement.

Under its original contract, EA has utilized Unreal Engine 3 to create several cross-platform games, including *Medal of Honor: Airborne* and *Army of Two*.

This new agreement represents EA's confidence in Unreal Engine 3 by providing EA development teams with industry-leading tools and technologies to best serve the needs of each game.

"With the largest and most talented studio operation in the world, it's critical for us to give our studio teams the best tools they need to make great games," said EA President Frank Gibeau. "This agreement reflects our commitment to Epic's technology which, in combination with our own cutting-edge systems, allows us to create ground-breaking hits."

NCsoft, 38 Studios and Bluehole License Unreal Engine 3 for Upcoming MMO Games

NCsoft has licensed Unreal Engine 3 for two top-tier, massively multiplayer online role-playing games (MMORPGs), a move that signifies its commitment to Unreal Engine technology. NCsoft has used Unreal Engine 2 to develop *Lineage II* and *Exteel*.

"Unreal Engine 3 has a well-structured rendering pipeline, and its graphical quality is superb thanks to advanced lighting and shadowing systems," said Young-muk Choi, lead programmer, Development Unit, NCsoft.

"Tools within the Unreal Editor empower us to instantly produce and optimize our outputs, and we especially love how the engine enables designers to easily prototype concepts without the need for programming."

38 Studios, founded by legendary Major League Baseball pitcher

Curt Schilling, is using Unreal Engine 3 to develop its upcoming PC MMO game, which is being art directed by Todd McFarlane and written by R. A. Salvatore.

"Epic's Unreal Engine 3 is best-of-breed, empowering developers with superior content creation tools, extensive middleware integration, and exceptional visual quality and rendering," said 38 Studios President and CEO Brett Close.

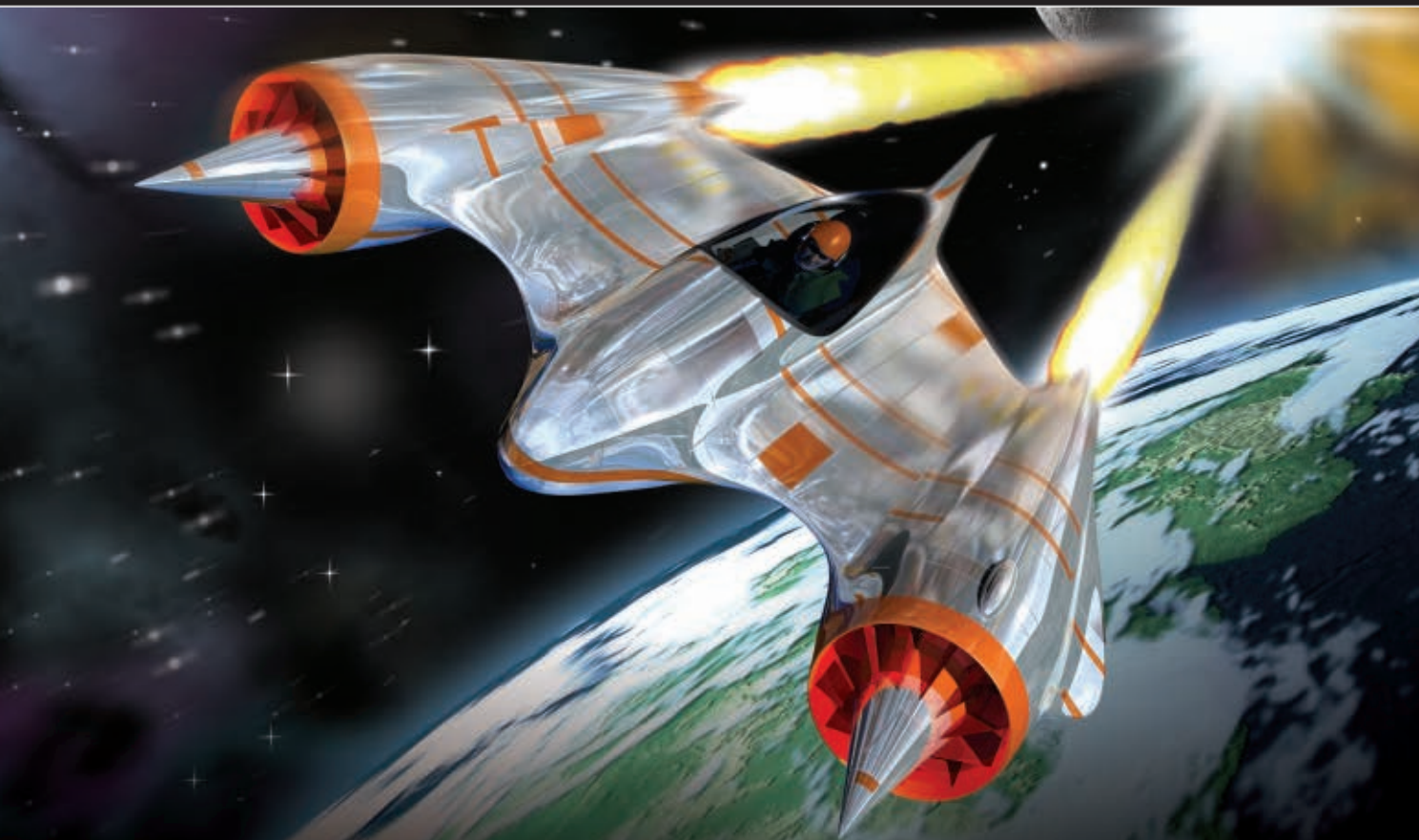
Epic would also like to welcome Bluehole Studio to the Unreal Engine 3 family. Bluehole is an online game company comprised of key members from some of the foremost development and management teams in Korea, and it has licensed Unreal Engine 3 for its next-generation, flagship MMORPG for PC, currently codenamed *Project S1*.



For UE3 licensing inquiries email:
licensing@epicgames.com

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

WWW.EPICGAMES.COM

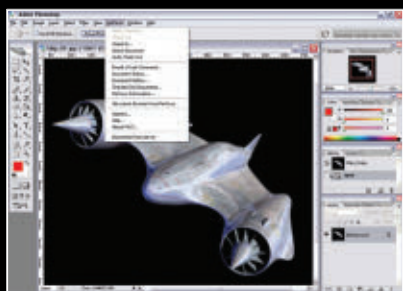


Introducing P4GT, a productivity feature of Perforce SCM.

The Perforce Plug-in for Graphical Tools, P4GT, makes version control painless by seamlessly integrating Perforce with leading graphical tools. Drop-down menus allow access to Perforce from within 3ds Max, Maya, Softimage XSI, and Adobe Photoshop.

Art and development teams can standardize on Perforce to version and manage both source code and digital assets. Enhanced collaboration during the design process helps teams to work together in real time to release small patches or create whole new worlds.

P4GT is just one of the many productivity tools that comes with the Perforce SCM System.



P4GT

PERFORCE
SOFTWARE

Download a free copy of Perforce, no questions asked, from www.perforce.com. Free technical support is available throughout your evaluation.



THE GAME DEVELOPER MARKET

WHO'S HIRING WHO IN THE GAME INDUSTRY, AND WHY?

FROM THE SMALLEST DEVELOPER TO THE LARGEST PUBLISHER, there is more hiring going on now in game development—and more searching for worthy job candidates—than at almost any time anyone can recall.

An informal survey of hiring managers reveals that the video game industry, now in heavy growth mode, is in particular need of skilled programmers, but also artists and designers. And while experience counts, sometimes just the right attitude and the ability to be a team player is enough to convince HR people to hire someone out of left field.

It's a job candidate's market to be sure, and this roundup of hiring trends gets into the nitty-gritty of how game builders can take advantage of these good times.

WHERE HAVE ALL THE ENGINEERS GONE?

At Electronic Arts, Activision, Blizzard, Insomniac, and Wideload, the need for qualified programmers is unanimous and at the top

of everyone's search list. That's mainly because video games are becoming more complicated and, at the same time, fewer people seem to be interested in learning how to create them, say hiring managers.

"All developers are looking for engineers in all spectrums of the experience range," observes Alex Seropian, founder and CEO of Chicago-based Wideload Games. "Unfortunately, finding people who have years of experience making next-gen console games is nearly impossible because the Xbox 360, PlayStation 3, and Wii haven't been around for years. A big pool of experienced applicants doesn't exist."

Blizzard Entertainment too needs engineers—and more. The developer has a number of jobs open, with more to come through the remainder of this year, says Lenny Grossi, senior manager of global staffing. They include art, animation, IT, game and level design, community management, and creative development.

PAUL HYMAN was the editor-in-chief of CMP Media's GamePower and currently writes a weekly column on the videogames industry for The Hollywood Reporter. He's covered gaming for over a dozen years. Email him at phyman@gdmag.com.

CONTINUED ON PG 18

THE GAME DEVELOPER MARKET

CONTINUED FROM PG 17

Specifically, the top three positions Blizzard is looking to fill in the area of art are animators (key-frame, not motion-captured), environment artists (hand-painted textures, no normal maps or procedural textures), and character artists (hand-painted, stylized). In the area of design, Blizzard needs level designers, game designers (quest, class, encounter, and systems), and game balance designers (competitive gaming or professional gaming experience is a huge plus). And, in the area of engineering, Blizzard is looking for tools programmers (art tools, plug-ins, level editing tools, etc.), graphics programmers, and database programmers.

"Across all disciplines, we look for passion and enthusiasm," Grossi adds. "Our ideal candidates are individuals who find themselves coding, drawing,

or designing levels in their spare time and out of sheer passion. Possessing applicable skills and experience, while also being up-to-date on the current trends and technology, is also important."

Because it is the industry's largest developer, EA's size means both good and bad news when it comes to hiring talent—it attracts more candidates than anyone else, but it suffers the most from "today's dry engineering pipelines," according to Mary Welsh, director of talent acquisition. "The complexity of the gaming platforms has increased exponentially these last few years," she explains, "which makes finding new, top-performing engineers critical to our development effort. We're a pretty big draw with the variety of genres we can offer an engineer, as well as the variety of locations and different team cultures. But finding talent is still a growing challenge."

development effort. We're a pretty big draw with the variety of genres we can offer an engineer, as well as the variety of locations and different team cultures. But finding talent is still a growing challenge."

THAT'LL EARN YOU POINTS!

At Activision, the HR people are looking for not only the number of development launches a candidate has accrued, but the reputation they've generated.

"If a talented person can't get along with people or they are difficult or inflexible, they won't survive in our organization," says Activision's Kate Ruddon, director of talent acquisition. "Are they a collaborator or do they act like a bull in a china shop? In our company, it's all about retaining top talent. A producer might be able to get one title out the door by running roughshod over



Alex Seropian, CEO of Wideload Games.



Mary Welsh, director of talent acquisition at Electronic Arts.

"When we have an open position, we prefer to reach out to people in the industry who we know, whose reputation precedes them, to urge them to come join our team."

—Alex Seropian, Wideload Games

people but nobody will want to work with them again to get the second title out. We don't want such people."

While Blizzard specializes in creating MMOs, Grossi says a candidate doesn't lose points because his or her experience is in, say, sports games or shooters and not MMOs.

"What makes a developer attractive to us is their passion and desire to make great games," he says. "If someone would love to create beautiful hand-painted environments for WORLD OF WARCRAFT but has only worked on racing games, we're still interested in speaking to them."

On the other hand, Insomniac likes candidates to have experience in the sort of game they'd be working on.

"We look to see whether someone has done, say, first-person shooters or worked on certain platforms," says Carrie Oliff-Dieterle, HR director at Los Angeles-based Insomniac Games. "That says something about what they like to work on and what they are passionate about. Looking at their body of work is more important than counting how many titles they've shipped."

And EA, which says it is exploring ways to "capture a greater diversity of population to impact the development of its games," chooses to give points to further that effort.

"For instance, our THE SIMS games have a 50%-plus female playership," explains EA's Welsh, "and so, in order to continue to draw in that female playership, we want to make sure that our development teams are more heavily female. That is a trend here at EA," she says. "We are actively pursuing ways to attract a more diverse population into our organization and, I must say, I think that will be a growing trend in our industry."

YOU CAN'T TELL A JOB BY ITS TITLE

Despite the fact that the video game industry has yet to fully standardize job titles—a technical artist at one developer may require different skills than a technical artist at another—HR pros say they've been able to work around that particular hurdle. In most cases, they tend to evaluate the



skill sets of job candidates rather than the job titles they've held.

EA, for example, has gone through the process of mapping out its particular skill sets, and its HR people use that tool to evaluate talent coming through the door.

"For us, titles don't matter," says Insomniac's Oliff-Dieterle. "It doesn't matter whether you are a gameplay programmer or a software engineer or a programmer level 1, 2, or 3. We are very cognizant that each studio will have its own title set. And so we try and identify people by their skill set. Our job postings on our Web site and externally reflect that. We've gotten a lot of help understanding the skill set of each job from the professional associations I work with, like ESHRA [Entertainment Software Human Resources Association]."

Some developers, like Wideload which is smaller in terms of internal staff size, keep to a minimum the advertising it does for job openings. Instead, it relies on networking.

"When we have an open position, we prefer to reach out to people in the industry who we know, whose reputation precedes them, to urge them to come join our team," says Wideload's Seropian. "We find that a lot easier than posting job titles which change with every cycle, which become more specialized every time we do a game."

SPECIALISTS OR GENERALISTS?

These days, there's no rule-of-thumb for whether a specialist or a generalist has a better chance of securing a job opening; in most cases, say hiring managers, it depends on the size of the team involved in the project.

At Irvine, CA-based Blizzard Entertainment, for instance, smaller teams call for more generalists who can take on multiple roles, while larger teams lean towards having more specialists.

"Considering that game development is a collaborative process, it benefits everyone to have some level of general skills even if they consider themselves to be a specialist," says Blizzard's Grossi. "Because each of our various departments and technology teams has specific and general needs, we want to speak with everyone."

Similarly, at Wideload, there is a need for both generalists and specialists on each team.



"The complexity of the gaming platforms has increased exponentially these last few years, which makes finding new, top-performing engineers critical to our development effort."

— Mary Welsh, EA

CONTINUED ON PG 20

POINT OF VIEW

- SPECIALIZING IN ADVANCED TECHNOLOGY**
- OVER 12 YEARS DEVELOPMENT EXPERIENCE**
- PROPRIETARY MULTI-PLATFORM GAME ENGINE**
- PARTNERED WITH THE INDUSTRY'S TOP PUBLISHERS**

UNREAL® ENGINE 3 DEVELOPER

LIFE IS SHORT, MAKE GAMES!

VISIT **POINTOFVIEW.COM**

THE GAME DEVELOPER MARKET

CONTINUED FROM PG 19

"For example," says Seropian, "our art teams report to an art director who is the generalist and who directs the vision for the entire game. At the same time, he has the specialists reporting to him—the animation director, the character modeling director, and the environment director, for instance, who are responsible for their specific areas. To some degree," he adds, "there's some expectation that everybody should be able to be a generalist."

Activision prefers hiring generalists which gives them the flexibility and adaptability that the company's development teams require. And, from a career perspective, it gives them more choices.

But, in recent years, Insomniac has chosen to eschew generalists in favor of specialists.

"When the company opened in 1994, when we had fewer than 40 people in our studio the first 8–10 years of our history, we required jacks-of-all trades who could pitch in wherever needed," recalls Insomniac's Oliff-Dieterle. "But game development has become more complex, we now have

almost 160 employees, and our need for specialists in particular areas has risen. At the moment, we would not want to hire, say, a modeler or a texture artist. That's too specific. But a person with two or three strong skills—say, a character artist with texturing skills or painting skills—that's what we would consider a strong job candidate."

"While some companies believe they can save money by outsourcing, there is just as much of a cost to integrate the work that comes from outsourcers."

- Carrie Oliff-Dieterle, Insomniac Games

HOORAY FOR HOLLYWOOD!

Suddenly it seems that Hollywood has become excited about video games. Movies and games are being released day-and-date, and it has become more commonplace for artists and writers, who previously might never have been caught dead in a game studio, to make that transition. And HR people at developers love the opportunity they now have to hire from other entertainment industries.

Activision's Ruddon says she's hearing from writers who were affected by the writers' strike, from animators, from actors seeking voiceover work, and especially from the music community.



Carrie Oliff-Dieterle, HR director at Insomniac Games.

"Records certainly aren't making money," she explains, "and so there are musicians approaching us because we bring them an entirely new audience. Traditional entertainers are starting to see gaming as a new distribution channel and we are benefiting from that."

Blizzard, too, has been hiring from various disciplines, including Web programmers from Internet service providers for its online technologies department; film and effects artists to work in their cinematics department; and novel, film, and TV writers to contribute to their creative development department. Insomniac Games has been hiring fine artists and programmers from a number of other industries, including film. But Oliff-Dieterle cautions that there is still a difference among industries, especially in production time lines.

"Our biggest challenge is to make sure that, say, an animator who is used to feature films understands the kind of animation and deadlines we face in games," she says. "They may be used to having a cleanup artist following behind them or they may be used to having a certain kind of visual development staff. So the issue is whether they can adapt to our needs."

Because writing has become an important part of game creation at Wideload, Seropian says he's taking advantage of the fact that "games seem to be the place to be." He's hired a writing



TRINIGY

VISION GAME ENGINE

**BOOST
YOUR
GAME**

**PROVEN TECHNOLOGY
POWERING MORE THAN 70 TITLES ALL OVER THE WORLD**

WWW.TRINIGY.NET



PLAYSTATION 3



TRINIGY, INC., 701 BRAZOS STREET, SUITE 500, AUSTIN, TEXAS 78701

THE GAME DEVELOPER MARKET

director, and now employs professional writers from various media, including TV and film. And he's using voice actors who previously did voice work in movies.

"Suddenly people seem to be migrating to our industry," he observes. "Suddenly we've become sexy."

ANYTHING TO FEAR FROM OUTSOURCING?

While outsourcing has become the de facto method of containing costs at most studios, it hasn't had any significant negative affect on hiring, especially at the handful of developers that are trying to keep their outsourcing to a minimum.

Blizzard, for example, chooses not to outsource its development, preferring instead to fill all of its open jobs with regular, full-time employees. And Insomniac doesn't outsource anything other than music.

"While some companies believe they can save money by outsourcing, there is just as much of a cost to integrate the work that comes from outsourcers," says Insomniac's Oliff-Dieterle.

On the other hand, Wideload operates on the principal of having a small, core, in-house staff that does the game design and most of the engineering; the majority of the artwork and audio work is done by contractors.

"That has been our strategy from day one and it has worked outstandingly well for us," says Seropian. "The difficulty for developers has always been that

when you start a project you need just a few resources. But, when it comes time to ramp up production, you need a lot of resources. Transitioning from the finished project with a large staff to beginning the next one with a small staff is tricky for independent developers. And so we have kept a core team whose size is consistent and when we increase production, we add to the team with contractors."

Those contractors are treated as team members who usually work at desks within the Wideload facility and concentrate on specialty work that the



Activision's Kate Ruddon, director of talent acquisition.

permanent Wideload team isn't equipped to handle.

"We're building a game called HAIL TO THE CHIMP whose main menu needs to look like a fake news station," explains Seropian. "That's a very specialized skill using motion graphics and it's something we've never done before. But there are outside companies that do that and we hired one that does it for the TV show 20/20. We would never hire them full-time because that task wouldn't apply to another game. In this game, however, they hit it out of the park for us and they did it efficiently, quickly, and better than we could have ever done it. That's a great example of where outsourcing makes total sense."

NO SUCH THING AS TOO MANY

With the industry constantly beating the bushes in search of talent, has that resulted in an overabundance of any particular skillset?

Not yet, says Wideload's Seropian.

"Ours is an industry where there's much demand for talent, even though that may not be true in all hi-tech industries," he explains. "Every manufacturer of devices with screens wants gaming content, and so there is a need for lots of production which means a need for lots of people who can create that supply."

"This is an outstanding time for anyone looking for a job," he adds. "There are a ton of opportunities. And not a lot of developers who are out of work—at least not for very long." ❖

"Traditional entertainers are starting to see gaming as a new distribution channel and we are benefiting from that."

— Kate Ruddon, Activision

Take a bite out of your development time with Havok Behavior™.



Havok Behavior™ will save your team time in the creation of character assets and behaviors. Created with the Technical Artist in mind, it is an innovative SDK for game programmers with an intuitive composition tool for artists and designers. **Havok Behavior** allows your characters to physically interact with the game environment during game development.

- Out of the box integration with our award winning Havok Physics™ and Havok Animation™
- Over 15 “triple A” titles currently in development using Havok Behavior
- Backed by our first class, industry leading Developer Support
- Optimized for leading platforms including Wii™, PlayStation®3, Xbox 360™ & the PC



Dublin • San Francisco • Tokyo

Contact us today for more information and to see if you qualify for a free evaluation copy of Havok Behavior.

www.havok.com



DEADLY SKILLS?

DEFINE THE FUTURE OF GAMING

WITH THE AWARD WINNING TEAM BEHIND

GOD OF WAR™

SCEA Santa Monica Studio

is looking for: programmers . designers . artists . animators . producers

Apply online: <http://www.us.playstation.com/Jobs>

PLAYSTATION 3

"PlayStation" and the "PS" Family logo are registered trademarks of Sony Computer Entertainment America Inc. Equal Opportunity Employer. The Sony Computer Entertainment logo is a registered trademark of Sony Corporation.

SONY



COMPUTER ENTERTAINMENT



DAMION SCHUBERT



PROFESSIONAL CAREER GUIDE

DESIGN OF THE TIMES

IDEA SYNTHESIS

A Philosophy of Pooling Ideas

TOO MANY JUNIOR DESIGNERS HAVE A

habit of hoarding their ideas like precious gems, not collaborating with others, and avoiding showing design documents until they're absolutely perfect. They think that their job is all about idea generation—what they don't know is that believing this leads them to all sorts of bad habits. They're overprotective of their ideas. They're obsessed with getting credit and, at the same time, utterly terrified that their ideas will be rejected, feeling that it reflects poorly on them in the highly competitive field of game design.

All too often, this self-consciousness results in designs that are too big, too safe, or too weird.

The best senior designers I've worked with have a different mindset. They understand, inherently, that most ideas are bad—even their own. They have less investment in getting their ideas into the game, and more in being sure the game rocks, no matter whose ideas get in. Consciously or not, they focus on idea synthesis, a term I use to describe the informal game design philosophy that focuses heavily on collaboration, mass idea generation, and focused execution.

KNOW WHAT YOU'RE BUILDING

It's astonishing how many designers will go off into a bubble and design a system or a level without any clear idea of what this design is attempting to accomplish or how it will serve the rest of the project. The output that results from this way

of working is almost invariably doomed before the designer first moves his mouse across the computer screen.

Every designer should have a deep, almost innate, understanding of the project she or he is building. What is the product's unique selling point? What will make it stand out? Who is the market for the game? What are the publisher's expectations for it? What does the license demand of the game? Designs and ideas that support these goals have a much greater chance of succeeding.

But that's not enough. When assigned a feature to design or an area to build, designers should try to schedule a kickoff meeting with their lead, in an effort to lay down all the expectations and boundaries, and to nail down any unknowns. Why is this feature here? How central is it to the experience? What does the market expect? Being sure that you have a clear idea of your lead's vision is one of the most essential paths to making a name for yourself on your design team.

KNOW YOUR LIMITS

Perhaps the most important part of the conversation with the lead designer is to actually form an understanding about how ambitious each feature is meant to be. Is this a core feature of the game? Is it meant to be trumpeted as a key point of differentiation from competitors' products? Or is the team committed to just matching the competition's games? More drastically, is the feature there just to say you have it, so the marketing team can justify putting the bullet point on the back of the box?

Junior designers often feel the need to innovate extravagantly on every feature that they are asked to design. But too much innovation can actually hurt the game. If a game has too much innovation, or too many innovative components, it risks spreading implementation

resources too thin and doing none of them well. Furthermore, reckless innovation risks being unfamiliar and confusing to new players and genre fans, and increases the odds of actually angering die-hard fans who prefer the old way of doing things.

Ultimately, most projects can only choose to truly take bold and daring risks on a handful of key ideas that the project leadership has identified as flagship features. Unless you've been told otherwise, your feature probably isn't one of them. Delivering something that is too radical and out there is a sure-fire path to rejection. Ask your lead how ambitious he was thinking the feature should be, and keep these boundaries in mind as you begin brainstorming.

YOUR IDEA IS NOT A SNOWFLAKE

Your ideas are not like beautiful and unique snowflakes. Most of your ideas are bad. But then again, most ideas are bad.

Designing with an idea synthesis philosophy requires that you understand these three things: ideas are cheap; great ideas are rare; and creative minds must sometimes be prepared to kill their babies. For many designers, adopting that attitude is perhaps the most important shift in mindset they ever make.

It's not just bad ideas that get killed. Some great ideas are killed because they are technically or financially unfeasible to build. Other fantastic ones die because they're just inappropriate for the game. Every now and then, you'll stumble upon one so good the producer wants to save it for a sequel that may never come. These things happen. Get yourself some thick skin.

One of the first steps new designers need to take toward becoming a more successful designer is learning how to keep their ideas in perspective. It's a folly worthy of Narcissus to fall in love with any single idea. Throwing temper

DAMION SCHUBERT is the lead combat designer of an unannounced online title at Bioware Austin. He has spent nearly a decade working on the design of games, with experience on *MERIDIAN59* and *SHADOWBANE* as well as other virtual worlds. Damion also is responsible for Zen of Design, a blog devoted to game design issues. Email him at dschubert@gdmag.com.



DESIGN OF THE TIMES

tantrums because your pet feature died in committee is a rookie mistake, too, and in six months, you won't even remember the sting. You cannot expect to magically pull the best answer out of the hat while flying solo. What you need is volume.

The key is mass idea generation. If you know that most ideas are bad, the best thing you can do is play the numbers. Reach out and gather as many stray design musings as you can. Pick through the lot to find the best ones, then hold full-blown brainstorming sessions. Hold informal lunch meetings to sort through them and pick the brains of your colleagues during coffee breaks. Use any excuse possible to get second, third, and fourth points of view.

Don't be afraid to reach out to subject matter experts as well. Embrace the person who better knows the system than you. Ask his or her opinions. And don't be afraid of the designer who always has ideas better than you. That's the person you should always invite.

Find out what worked on previous games, and more importantly, what didn't work. Do your research. Don't just look at the top competitors in your space. Look at the little games and old forgotten ones. And don't be afraid to read a book or watch a documentary out in the real world.

Write down every idea, no matter how silly. Don't be afraid to show your work. Then let other people review the silly ideas. All too often, the crazy ones can build into amazing ones, simply when other people bring different perspectives on how to make it work. Do everything you can to encourage a positive and fertile environment for new ideas to sprout.

CULL THE WEAK

The most successful designers I've worked with know that their job is not

to come up with a great idea, but rather to identify the best idea and execute it. Once you have a whole bunch of great ideas, it's time to thin the herd.

Don't rely solely on your own instincts to trash the weaker ideas. Start by asking other people. Show them a whole bunch of ideas, and ask them to put a check mark next to the one that immediately grabs their attention. What you're looking for is resonance. Some ideas just grab people, the way a catchy pop song does. If you're lucky enough to get one of those ideas, lock onto it like a laser sight.

While you're soliciting feedback, don't lose track of the big picture. Measure the ideas you've been generating against

“ **Most projects can only choose to truly take bold and daring risks on a handful of key ideas.** ”

the project's goals and the boundaries defined for you with your lead designer in your kickoff meeting. The best idea in the world won't get greenlighted if it undermines the project in other ways.

KEEP IT SIMPLE

Fun is an elusive concept, almost impossible to predict before the game is tangible. Your first design document should be kept as simple as possible, with an eye toward getting the feature in place quickly so the design idea can be validated in context with the rest of the game.

Keep your first draft simple. Write a lightweight design document focused on the fun that highlights the innovative aspects and calls out the edge cases where the feature might break if not accounted for. It's a rookie mistake to over-design, to design every unit, every bonus, every possible permutation of variables and flags a feature may ever

need. The resulting behemoth can actually hurt a feature more than it helps it.

Over my career, I've seen dozens of great features killed by design docs the size of the Old Testament. Programmers or project managers often torpedo a feature like that before they even finish printing the document out.

As game designers spend time in their professions, they become more adept at predicting when that level of complexity with fine-tooth details is appropriate. Until you develop that sense, don't worry about it in the first draft of a design document. Push to get a lightweight version of the feature built quickly. Let play testers get their hands on it so the

design can be played with and iterated.

Embrace all feedback. Once a feature is tangible, it is tremendously easier to have a clear perspective of it and how it works.

Once you know what worked, what didn't

work, what's fun, and what's tedious, it becomes almost trivially easy to target the correct parts of the design that need more complexity and depth.

EYE ON THE PRIZE

Last but not least, give credit freely to those who contributed ideas to you, but don't obsess over it yourself. The credits page doesn't list the designers for individual features or levels, and even if it did, having a mega hit on your resume is vastly more important.

The days of one-man design teams generating blockbuster games in the garage are over. The industry today is about huge teams working cooperatively to chase after a very elusive concept of fun. Good designers understand this, and work within these constraints. But the great ones don't see this as a constraint—they learn to leverage this extra brainpower to make better designs and better games. ❖

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



JESSE HARLIN

» AURAL FIXATION

THE THREE RS OF AUDIO LEADERSHIP

PROFESSIONAL
CAREER GUIDE

LAST YEAR (AN INSIDE JOB, MAY 2007), we talked about making the jump from freelancer to on-staff sound designer. Being on staff eventually brings with it assent up the rungs of the corporate ladder. With enough years and completed titles under your proverbial belt, the opportunity to lead a project will inevitably present itself.

While the temptation of a higher salary and an increasingly impressive resume can be a powerful draw, and while all audio leads need to have a rock-solid understanding of the process and pitfalls of sound design, the role of audio lead is actually a radically different one than that of sound designer. At its core, the job of audio lead is a management position and with management comes an entirely new set of expectations and job requirements—frequently at the expense of actual sound design. While titles, companies, and personalities will vary greatly, the common facets of all audio lead positions fall into three separate categories that we'll refer to here as The Three Rs: Resources, Representation, and Responsibility.

RESOURCES

As a good friend of mine describes his job, the audio lead is responsible for everything that comes out of the speakers. This means that simply having folders full of music, voice, and sound effects assets on a hard drive or checked into Perforce isn't enough. The company is relying on the audio lead to make sure that those assets make their way out of the speakers at the highest quality level possible while coming in on time, on budget, and conforming to all necessary console compliance specifications. To accomplish all of these goals, an audio lead has to become proficient at juggling resources.

Unless the game is small, the audio lead will most likely need a support staff of at least one additional sound designer. Larger games require elaborate audio teams that can include both on-staff sound designers and teams of outside contractors. Regardless of the size of the team, the audio lead is responsible for a staffing plan that realistically determines the number of designers, when those designers roll on to and off of the project, and which is flexible enough to change if the needs or scope of the project changes during the development cycle. This might mean coordinating with human resources and top management to interview and audition new hires.

Manpower isn't the only resource audio leads track. It also falls to the audio lead to decide what audio engine will be used. This decision might mean going with an established piece of tech that has been in use at the company for years. Even then, some changes to the tech may be needed, and the audio lead is responsible for specifying these changes to engineering staff. If no tech exists, the audio lead will need to either evaluate available commercial middleware or spec out the design of new internal technology.

Lastly, the audio lead is responsible for daily administrative work. This could be anything from early pre-production documentation such as milestone roadmaps or Microsoft Project files to assigning designers audio bugs to signing timecards and answering the inevitable mountain of email.

REPRESENTATION

Resources are the most tangible aspect of the gig. The other Rs are more nebulous, but no less important. The audio lead is the public face of audio and will be the most visible member of the audio team to everyone from level designers to testers to executive management. This means representing audio at a vast array of relevant meetings, from weekly check-ins with producers and directors to audio

staff meetings where tasks are delegated, needs are assessed, and roles are defined.

As such, it falls to the audio lead to instill a sense of competence and confidence that all audio tasks are being handled professionally. This means having a head not only for compression ratios and memory footprints, but also for interoffice politics, conflict resolution, and diplomacy. Audio leads must make sure that they serve as the channel of communication between audio and the rest of the development team, ensuring that tool, tech, and vacation requests from the sound designers find their way up the chain just as milestone dates, demo needs, and overtime requests find their way down. Since the lead is the one talking audio with the rest of the team on a daily basis, he or she must also make sure that to pass along praise and other feedback to the sound designers who might otherwise feel as though they're creating content in a vacuum.

RESPONSIBILITY

As a manager, an audio lead has special responsibilities to both team and company that have nothing to do with audio. Managers are viewed as agents of their company and as such are expected to adhere to higher standards. This may mean taking part in HR-mandated sensitivity training as well as employee performance reviews and decisions involving bonuses, vacation approval, and career growth.

Managers are role models for their employees and their attitudes will affect team morale and performance. Careless complaints about crunch time or executive management set an example to the team just as much as pep talks and a positive attitude do. Being responsible for everything that comes out of the speakers means no longer simply safeguarding your own reputation, but being entrusted with the reputations of your audio staff, your development team, the game title, and your company. ❧

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.



MATTHEW WASTELAND

❖ ARRESTED DEVELOPMENT

IT'S A TRAP!

How to avoid getting caught in managerial rhetoric

THE PATH OF CAREER DEVELOPMENT

in the game industry is fraught with strange hazards, not the least of which are the odd delusions that may be foisted upon you by your Management such that you may be a more productive and harmoniously integrated Work Unit. Now, far be it from me, an established professional both famous and important, to stir up a hornet's nest of raving discontent—I simply mean to help you keep a careful eye on your situation. It can spiral out of control very quickly and, like the lovely and tragic GLaDOS, you may find yourself in a kind of frenzied denial about how things really are.

In that light, here are some patterns you should learn to recognize so that you may have some signposts to guide you into the bracing realizations that may eventually lead you along the path of truth—one which may lead you to greener pastures or to a fortune sought elsewhere. Or at least to an embittered and spiteful temperament that will scare the producers off so you'll be left alone.

"We just have to hunker down and ship this game, and then you'll all be rich—like me!"

A potential bonus is not something that should be used to justify complete insanity on your part. We all know that money is a great motivator—all the better when the people in charge don't actually have to pay any of it. Hinting at some breathtaking royalty payout that's just around the corner after the game is a huge hit (which it is practically guaranteed to be, surely!) usually does

wonders for the crew, so it happens rather too frequently.

Here's a funny experiment you can try: ask someone in charge to describe how the royalty is actually calculated. You're likely to get a lot of puzzling airy statements, hand waving, and suddenly important phone calls. In the very best case, you may get a fancy chart or graph that tantalizingly teases at some kind of meaning, but never quite reaches it. If you're truly curious, the real process is this: all that money goes into a big black box, and the tiniest little drop comes out of the other end for you, the diligent worker. That's given that you have a game that sells well at all, which sometimes you don't, exactly.

"Come on, you aren't being a team player—support your buddies."

Everyone else buys into this whole "crunch for no reason and bad project management" thing, so why don't you? Maybe you're the crazy one! Blatant attempts to coerce team dynamics like this are another management machination of which you should be very careful. Don't let them try to leverage to their own ends the soldier's camaraderie that develops when people are thrown into crappy situations together. Your grandchildren aren't going to ask you to tell them about the war video game you worked on once. They probably won't even know what video games are, let alone care.

"Well, you could always be flipping burgers."

When people complain about low pay or poor conditions, you sometimes hear the "you could always be flipping burgers" defense (or, "you could always be programming databases" for engineers, since they want to be speaking your language, of course). This is a strategy meant to distract the grumbler from realizing his job sucks by trying to get

him to think of something even worse, never mind that a job flipping burgers might actually have reasonable hours.

Passive-aggressive overtones aside, this kind of justification is logically suspect since it confers upon the game industry some kind of special, extraordinary status. It's different than all other industries of the world! Getting utterly, totally screwed is somehow the price one must pay for the glamour and mystique of making video games for a living (or working in porn?). And if you can't deal with it, go get a "real" job. It would be sublimely ironic if these ended up being the same people who wonder why the game industry doesn't get the respect and cachet it deserves from the other creative media, but that's probably too perfect to really be true.

"At least we have a really fun work environment."

The team is stuck at work late and on the weekend trying to get a disaster of a project into a box and out the door. Is it bad? Yes, it's terrible—but wait! You get to eat barbecue! And there are fun distractions around the office. Why, it doesn't even feel like work at all, so what does it matter if you're there for all of your waking hours? Standing around having a blast playing GUITAR HERO with all your office mates in the wee hours of the morning is sure to have absolutely no effect on the pile of work that must be done to ship the game—the pile of work being the reason you started crunching in the first place!—so all you're doing is munching on the hair of the dog that bit you. I mean that metaphorically in most cases; the others—you know who you are.

"Okay, we learned our lesson. We'll never run a project like this again. Promise."

After it's all over and the game has shipped, you'll hear this one. The reason is that suckers believe it every time. ❖

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



INTERVIEW: MIKE ZAK

ENVIRONMENT ARTIST ON HALO 3

CHRISTIAN NUTT is features editor of *Gamasutra.com*. He used to work in the enthusiast press, but promises he never reviewed your game. Email him at cnutt@gdmag.com.

IF NOTHING ELSE, HALO HAS STYLE. MORE TO THE POINT IT has a style. The look and feel is far from generic, and one would expect nothing less of a game that manages to move as many copies as it does. Here, *Gamasutra.com* features editor Christian Nutt spoke with Bungie environment artist Mike Zak about the game's unique visuals, Bungie's new found independence, and designing art for areas that only five percent of the game's audience will ever see.

Christian Nutt: *HALO* has always stood out from a lot of first-person shooters through its naturalistic environments, and even the city or urban environments have that certain... there's something distinct. If you look at any screenshot from *HALO*, even if the HUD weren't up, you'd know it was from *HALO*.

Mike Zak: That's really important to us on the art staff. There's a precedent set by *HALO 1*, and there's a lot of gritty, realistic games out there that do that really well, but for us, we're more inspired by slightly more imaginative spaces and more awe-inspiring vistas and ideas. Personally, it's a dream franchise to work on in environment, because I get pretty tired of bricks and rusty pipes. Not that I don't admire a lot of games that have really impressive bricks and rusty pipes, but you get the opportunity to let your imagination go with *HALO* environments, and really build spaces there. You might want to hang out there, to not necessarily fight in, but to have a picnic or go for a hike.

CN: *Do you think that serves an important gameplay function, too? The way you build your environments ... how does that affect and enhance the gameplay?*

MZ: I think they give better choices. I wouldn't know if those inform the gameplay as much as the tone, so it's not in a direct way. It depends on how you want to define gameplay, but if you want to talk about gameplay as the specific mechanics of character interaction and the fighting mechanics, probably the most direct way that the aesthetics and the level environments match up would be, for example, the Forerunner architecture style really lends itself to building playgrounds that are architectural and structural, but are inscrutable enough in function so that you can really go to town with things that might be a little too gamey. For example, in a contemporary shooter where you're going through bombed-out streets, you can't have platforms and ramps and weird conduits the way that we can. In some ways, the theme really supports that, but at the end of the day, we can do a lot of that with rocks and terrain and whatnot. You can have the earth serve the same mechanical functions.

CN: *How has your art pipeline evolved with next-gen systems?*

MZ: We're constantly evolving those target and development pipeline—evolving the art bar, is what I mean by target—and that art bar is defined by budgets that we're engineering now, especially with the 360 and the GPU and the CPU and the complexity of current-gen shader systems and physics and so many things going into the game. There's no simple answer. When I started in the industry, it was pretty much, "What's my poly count? What's my texture budget?" Now, I can't even get a straight answer from the programmer, because there is no straight answer. They have the most elaborate profiling tools and are constantly tweaking performance based on, like, "Are

CONTINUED ON PG 34



Need A Performance Boost?

Shift productivity into high gear with The Archive™, the world's largest single source of high-quality 3D content. The Archive contains 16,504 production-proven 3D models, including over a thousand fully digitized vehicles, landmarks, accurate military models, animals, human anatomy, 3D geography and more. Save modeling time, reduce project costs, and leave your competition in the dust.

Ideal for any game developer's toolbox.

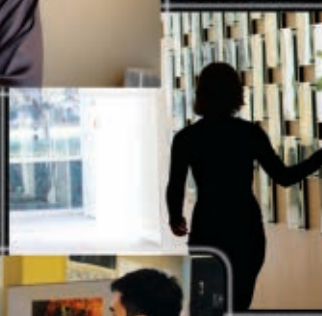
Each model has been carefully crafted in industry-standard OBJ format for compatibility with virtually any production pipeline. The Archive comes wrapped in a game developer friendly site license, available for only **\$4,995**.

Find free sample models and a demo at:

www.digimation.com

Digimation





A Great Place



to Work and Play

Kickstart your career: www.jobs.ea.com



INTERVIEW: MIKE ZAK

CONTINUED FROM PG 30

you throwing a lot of grenades?" That's very different if you were driving six Warthogs, or if it's just you and a sniper rifle.

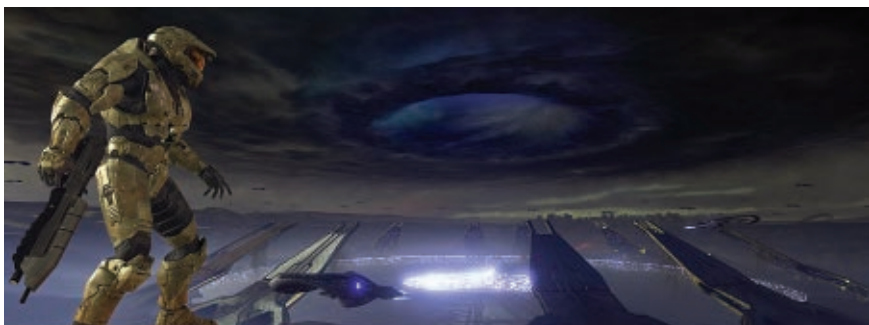
CN: Now that players can do flythroughs and examine the levels in complete detail, what considerations does that create for you?

MZ: In a way, we were initially kind of dreading that. Like, "Oh man, you're going to be able to peek at all of our laziness." But actually at the end of the day, I think it was kind of good, because it actually more often than not—the screenshots I saw where I would be afraid of exposing a bad angle, users were actually coming up with better viewing angles than we did. A lot of times, they're actually getting to see stuff that we had to model for that one percent edge case, where somebody could grenade-jump up. We were like, "Okay, I have to model that and texture that, but I know nobody's going to see it." But now, tons of users see it, because they're flying up there, and that was awesome. We didn't necessarily author a lot more content than we did on HALO—well, we did more than HALO 2, but for different reasons—than we otherwise would if we didn't have safe zones. In some ways, it's actually kind of satisfying, that some work paid off that you didn't think would.

CN: I remember when the beta came out, and somebody clipped through Master Chief's helmet somehow and found the face texture. Those are the things that must be hard to anticipate.

MZ: Basically our golden rule of making something bulletproof is that it will never be bulletproof. The players will always find a way to break your game, especially when you expand to the player population that we have. We have an amazing test team, but there's no way they can compete with several million users, you know? It's just impossible. There's a lot of smaller games that get away with a lot more because of that.

CN: How do you set up your team structure?



MZ: We sort of cluster by general discipline, but our desks are angled at 30 degrees so we can build—we don't have a grid system, basically, in our office. We have a giant, open layout, and we can cluster people. So we generally have programmers sit in the same general area, and art and design and etcetera, but we try and pair up—like, a programmer who is working on animation systems should sit near the animators. So he'd be on that fringe of the programmer side. And environment designers need to sit near mission designers, because they're constantly iterating together.

We shift it around a lot, too. If somebody changes position or

assignment, then they might move desks. We try to keep it fluid.

CN: I think you said that they have wheels?

MZ: Yeah. All of our wiring is in the floor. We have a false floor and it just comes up through there, so we can just unplug, move, and the IT team must [go crazy.] It happens all the time.

CN: Do you think that's good? What do you think is the best situation?

MZ: It takes some people a while to get used to. They're used to having their own office, or sharing an office with like two people. I came from a studio that had separate offices, but I would never allow that. If I ran a studio, there would be no question.

I'm the most firm believer. How you segregate or integrate in an open plan—there's no perfect way to do it, but I absolutely believe in an open plan. There's just so many conversations that would have never happened, or so many people who wouldn't know each other well enough. There is definitely a problem with being able to concentrate at times, but you just learn to love the headphones. Half the time my headphones are on and I listen to music, because I need to concentrate and dull the noise. People tend not to include you when they come talk, because they know they're interrupting. It's a simple system that's emerging.

The other thing I really like is that it democratizes the ceiling. Producers and studio management sit right in with everyone. There's no corner office. Well, Marty's got his ivory tower, but you could argue that sound actually does need to be isolated.

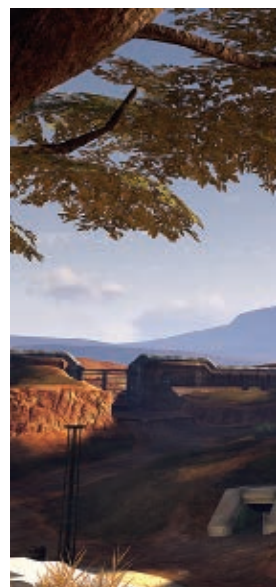
But beyond that, it feels good as a developer in the trenches when your producer, who doesn't actually have a deliverable for that milestone, is still there crunching with you or sitting right next to you.

CN: Is it easier to get the "all hands on deck" feeling?

MZ: Yeah, definitely. It allows you to feel like it's okay to have conversations with people you might otherwise avoid. It's definitely a cultural thing.

CN: What is the mood of going independent again?

MZ: There's an enthusiasm, in a way. You really feel like you're controlling your own destiny. Being part of such a big organization such as Microsoft, where you're a part that plugs into this massive organization that has revenue streams well beyond anything you're ever going to have visibility into... When you're all sitting in one room and know you're all working on the next great thing, you get serious about what you do with your time. Not that we weren't serious before, but it definitely gives you a different voice in the back of your head. I feel more diligent and have a sense of ownership now. We only have ourselves to blame if we don't deliver the best we possibly can.





CN: I've heard a little bit from different people that there was a sense Bungie could never fit properly into Microsoft. Did you ever feel that way?

MZ: Well, that statement could be as true for any discipline, because Microsoft is a giant umbrella, like I said before, that has all these different quote-unquote "cost centers." Microsoft is an entertainment company, first and foremost, and games are entertainment and content-driven. So that was a new business that they're ramping up. But that doesn't mean that Microsoft is not capable of having an incredibly successful entertainment segment to their business.

CN: You hear that echoed by John Riccitiello now. I don't know if any of you guys caught what he said at DICE, but he basically said that EA's failure had been to try to manage outward to the studios they acquired, and they had success when the studio's culture flow inward to them. Maxis and Black Box were success stories, and Westwood and Origin were failures of the EA way, and their new organizational metaphor is a city-state system.

MZ: I've worked at other studios, and honestly, just as just a guy in the trenches making art, the culture at Bungie served a little voice in the back of my mind that's like, "Hey, it's just us now." But the fun mental culture was there when I started with Microsoft, and it didn't change in a significant way. And I never felt, even when we were on the main games campus before we got our own building, we were in a sort of cafeteria—I didn't feel like they were running our lives. In fact, I worked for a studio where even though we were off in our own building in our own city and the next office was halfway across the country, I still felt like there was more heavy-handed publisher down and first-party management than I felt sitting in Redmond. And who knows. I'm just in the trenches making art, but there could be all sorts of cubes and shielding going on in the management levels, but at the end of the day, I can't imagine the relationship was completely antagonistic, because both parties are enjoying the success. ❌



Are You a **Star**?

If so, you'll fit right in at Forterra. We're the market leader in serious Virtual World platforms, and we're growing! Our platform powers 3D worlds, solving real business issues for innovative companies. We work with global partners like IBM and Autodesk.

We're looking for star coders and artists to bring 3D virtual worlds to life.

Visit us today and **get ready to explore a new world.**

www.forterrainc.com/jobs

FORTERRA
SYSTEMS INC.

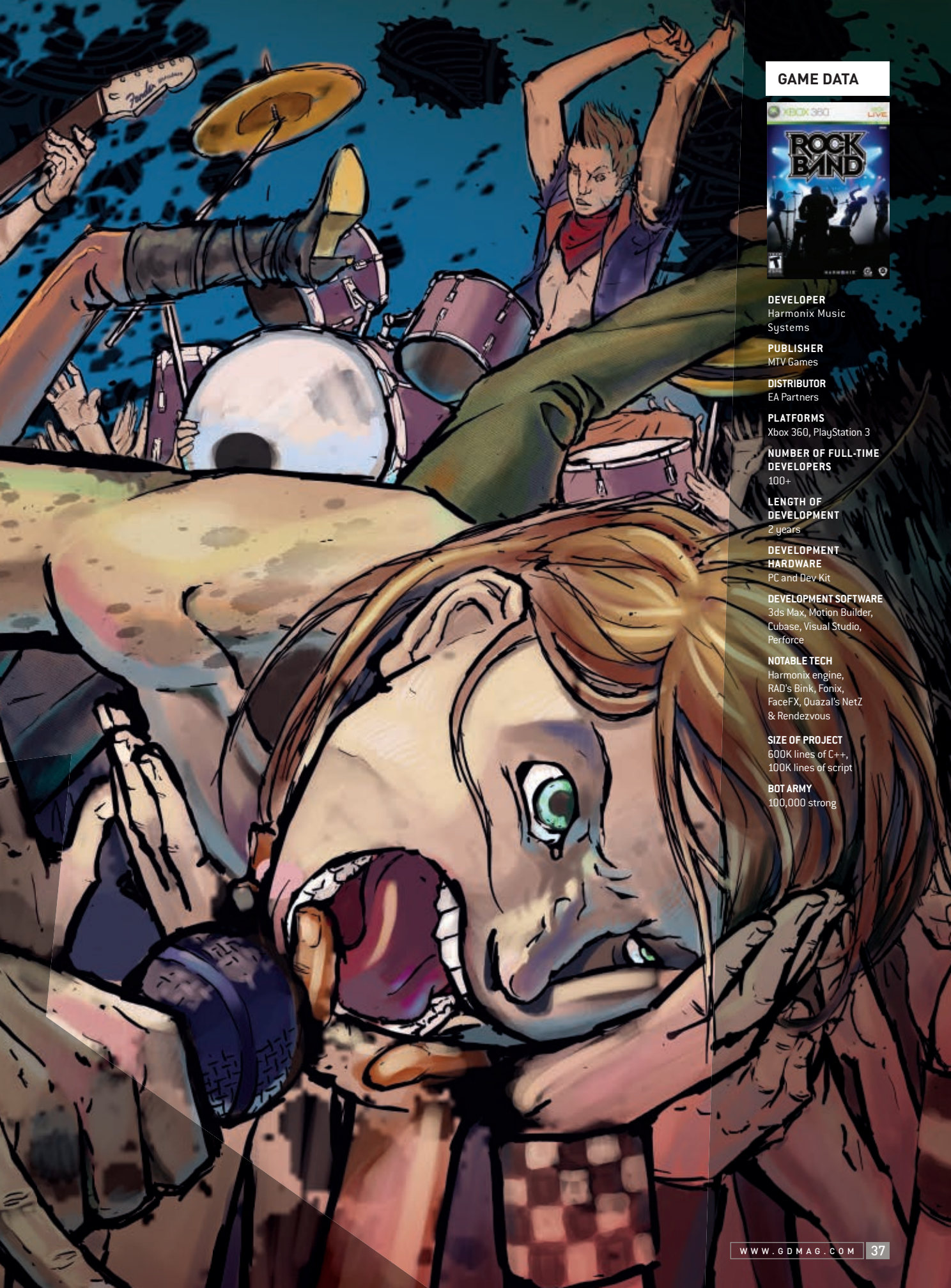
HARMONIX'S

ROCK BAND

ROB KAY is the director of design at Harmonix. This article was written with contributions from Eran Egozy, VP engineering, Ryan Lesser, art director, Dan Teasdale, senior designer, Tracy Rosenthal-Newsom, director of production, Daniel Sussman, director of hardware development, and Greg LoPiccolo, VP of product development. Email him at rkay@gdmag.com.

THE STORY OF ROCK BAND'S BIRTH, AND the transformation we've gone through as a company to create it, is a tale that an accomplished writer will someday pen, but for now you've got me. I was the lead designer on ROCK BAND. Working with our senior management to capture their vision for this game, I did wonder if they'd lost their minds. Their vision was simple: create a music game that takes instrument simulation to its logical conclusion—getting a full band together. The intimidating reality the team faced was that this wonderful vision entailed an enormous workload. The

drumming component gameplay of ROCK BAND would need to be built from a standing start. Guitar, bass guitar, and singing gameplay would have to be refined to work in a band setting. Finally, and crucially, we had to develop new cooperative multiplayer gameplay to tie drums, vocals, guitar and bass components together into one cohesive rock band experience. In short, we strove to create the ultimate party game. But it didn't end there. The vision included customization features to give bands their own identity, a world to tour, and enough depth to keep



GAME DATA



DEVELOPER
Harmonix Music Systems

PUBLISHER
MTV Games

DISTRIBUTOR
EA Partners

PLATFORMS
Xbox 360, PlayStation 3

NUMBER OF FULL-TIME DEVELOPERS
100+

LENGTH OF DEVELOPMENT
2 years

DEVELOPMENT HARDWARE
PC and Dev Kit

DEVELOPMENT SOFTWARE
3ds Max, Motion Builder, Cubase, Visual Studio, Perforce

NOTABLE TECH
Harmonix engine, RAD's Bink, Fonix, FaceFX, Quaza's NetZ & Rendezvous

SIZE OF PROJECT
600K lines of C++, 100K lines of script

BOT ARMY
100,000 strong



bands playing for months. Add in unparalleled DLC that would transform the game into a music platform, and it was one tantalizing package. Of course, then we had to make it!

The team would eventually grow to around 160 people at Harmonix, and countless contractors elsewhere, but it all began with three people making a drum prototype at the tail end of 2005.

WHAT WENT RIGHT 1 PROTOTYPED DRUMS EARLY.

We got our drumming prototype up and running before doing anything else. This afforded us time to understand the unique hardware needs, interface challenges, and

learning ramp inherent in learning to play the drums.

As with all instruments in *ROCK BAND*, we wanted something that captured the feel of playing the real thing. We threw around various ideas for the basic form factor, and as our prototype hardware was configurable, we were also able to experiment with different pad layouts before settling on four pads and a kick pedal. We connected this to our PC-based prototype software, gave the drum setup its own room, and encouraged people around the company to come play it and let us know what they thought.

This drum prototype taught us a great deal. Crucially, it revealed an interesting drum-centric problem with our Harmonix 3D interface concept. Our usual row of note lanes (as seen in many of our games, including *FREQUENCY*, *AMPLITUDE*, and *GUITAR HERO*) worked fine for the drum pads but failed spectacularly on the kick drum pedal. The spatial organization of notes on screen didn't match the spatial relationship of the pads and pedal. This made readability of note patterns exceptionally tricky. Through a combination of theorizing and experimentation, we arrived at the solution seen in the final game—an orange line stretching across the whole track. We're proud of this solution, and how early we nailed it considering how many other challenges lay ahead.

Perhaps the greatest discovery of this drum prototype was simply how fun it was to play. It really captured the feel and power of real drumming, which gave us a huge confidence boost.

2 DEVELOPED THE HARDWARE AND SOFTWARE IN TANDEM.

We began *ROCK BAND* as a software developer, but decided to take on the task of designing and manufacturing the hardware ourselves. Designing the controllers from scratch—and starting up a major manufacturing effort in under a year—was an enormous undertaking with a considerable learning curve. The payoff though was more than worth it. For a game like *ROCK BAND*, the hardware is at least as important as the software, if not more. We were able to design and develop the hardware and software together, taking full control over the user experience.

We began hardware development in February 2006, working with an industrial designer (J. Hayes Industrial Design) and a contract manufacturer (Canyon Creations). Because it was the least understood controller, we attacked the drums first. The big challenge was delivering the playing experience of a real drum kit, in a videogame controller that would cost a fraction of the price. The guitar controller was less blue sky, and development started after we had the basics of the drum controller down.

Hardware was a brave new world to us. Throughout the whole eighteen-month project cycle, we made mistakes (see "What Went Wrong") but tried to focus on understanding the critical decisions and making the right calls. In April 2007 we began tooling for both drums and guitar (cutting the steel molds that shape the plastic parts). The molds take about eight weeks to complete, and then you have at least another four weeks of pre-production and tuning. In September 2007 our first containers left the China warehouse. By January 2008 over two million *ROCK BAND* bundles had shipped.

3 CAPTURED THE FEELING OF PLAYING IN A BAND.

The biggest design challenge for *ROCK BAND*'s gameplay became all too obvious the moment we got our first version of a four-player game up and running. It was not a stellar group experience. Obviously something key was missing—player interaction! There was no reason to care how other players were doing, let alone any of the interactions musicians in a real band might recognize. This cooperative multiplayer game mode was to be *ROCK BAND*'s reason for being, so it had to become amazing.

Enter the Harmonix Design Cabal, a mix of wily Harmonix vets and senior designers. This group identified some experiential goals for the gameplay team to go after. The experiential goals

Weekly DLC and Music Store

From day one, we hoped *ROCK BAND* would become a fledgling music platform as well as a game. Our weekly song releases, generally two dollars a song with discounts for song packs, have proved incredibly popular, surpassing our expectations. We passed 2.5 million song

purchases in our first two months alone, and the biggest seller so far has been the Metallica Pack.

Since launch, a Music Store has been added to the game via a free software update. We wanted to get this into the shipped game, but simply ran out of time. The UI has been designed to handle

the sheer volume of DLC we're making available, and improve the experience of browsing new songs [via things like cover art, audio previews, and difficulty ratings].

We've invested a huge amount of time and resources to make weekly DLC and the Music Store possible. It's been well worth

it though, and we'll continue to develop this approach because we believe it's the future of music gaming. Not only do game consoles now have hard drives and broadband Internet connections, but also our players seem to have an insatiable appetite for more music.



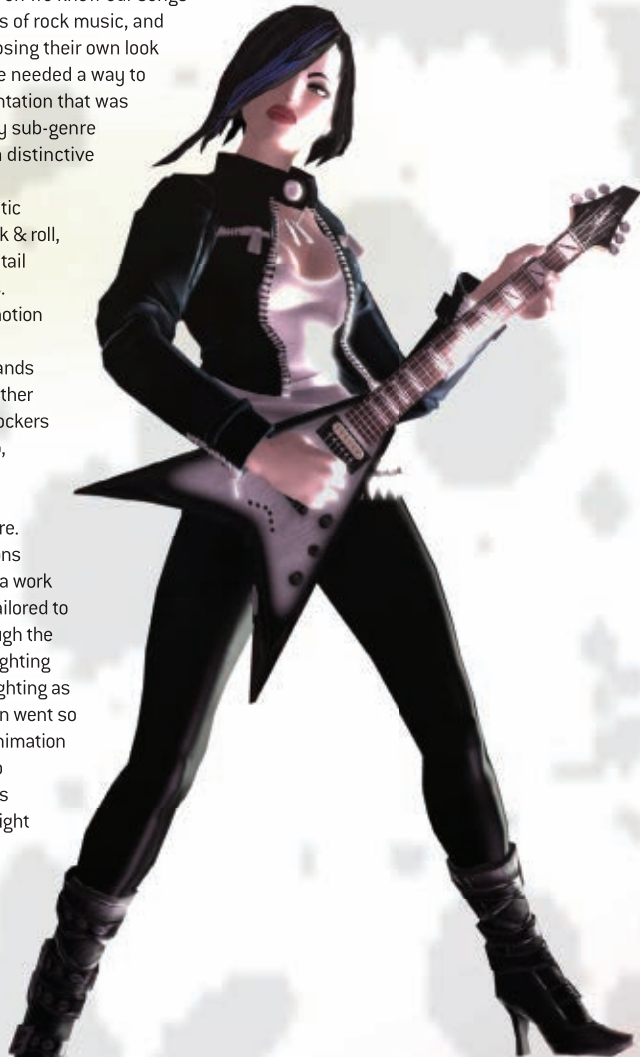
were “togetherness” and “distinct roles.” The gameplay team [three game coders, two UI artists, the audio lead, and myself] experimented for many months with gameplay mechanics and UI that met these goals. We were scheduled loosely with three days a week set aside for iteration time. Each week, we’d meet with the cabal and review progress, which gave us enough freedom to react quickly and experiment, but also enough direction to stop us veering off course. We experimented with many scoring systems and gameplay mechanics to try and meet the high level goals. The whole company played this evolving game, and offered feedback and ideas, which fed into the process on numerous occasions.

Many experiments were failures, but taught us something valuable. Some worked well with four players, but fell apart with two or three. Some were diamonds in the rough, which we kept and revised and eventually honed into a cohesive whole. This was the process that birthed all of the unique ROCK BAND gameplay including individual fail outs, saving fallen band-mates, Unison Bonuses, Bass Grooves, Guitar Solos, Drum Fills, and Big Rock Endings. Together these features formed the “band” in ROCK BAND.

4 EMBRACED “AUTHENTICITY” AS A GUIDING PRINCIPAL. Very early on we knew our songs would encompass all forms of rock music, and that players would be choosing their own look in the character creator. We needed a way to approach the visual presentation that was generic enough to host any sub-genre of rock, and yet still offer a distinctive look and feel.

We focused on an authentic recreation of traditional rock & roll, with serious attention to detail and high production values. Character animation was motion captured from performing musicians who played in bands of a particular sub-genre rather than actors (we got punk rockers to do the punk rock mo-cap, for instance). Clothing was labored over, with shops representing each sub-genre. Instruments were recreations of the real thing. The camera work and post processing was tailored to present the live show through the lens of a music video. The lighting mimicked real rock show lighting as closely as possible. We even went so far as to create a custom animation system for the drummer so that every single drum hit is realistically played on the right drum or cymbal!

The litmus test “is it authentic?” began as an art direction, but quickly



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



Daniel Sussman, Director, Hardware Development.



James Fleming, Lead Programmer.



Eric Brosius, Audio Director.



Ryan Lesser, Art Director.

spread project wide. "Authenticity" became a buzzword in not only art reviews, but also design, hardware, music and writing discussions. The music had to be authentic, so we secured as many master tracks as possible. The guitar hardware had to be authentic so we worked with Fender to create a replica of their iconic Stratocaster guitar. It spread through our collective conscience; everyone heard the "authenticity" mantra and got behind it, unifying the overall presentation and vibe of the game.

5 MADE GOOD EARLY CALLS ON NETWORKING TECH. We made some key technical decisions about our approach to networking early on that really paid off.

We chose to use Quazal instead of making do with the network solution provided by the platforms alone. This decision had to be made far earlier than when our network features were fully designed. As it turned out, the flexibility afforded by Quazal far outweighed the additional complexity of integrating another middleware provider. We came out with an awesome feature set that was custom crafted for our game, including head-to-head battles, online band play, a flexible leaderboard system that can hold thousands of boards, and persistent band data storage. We were also able to create a common code base for both Xbox 360 and PlayStation 3, which helped reduce development time and kept platform differences to a minimum.

We also decided to conduct extensive network bot testing for our online solution before release. We spent months developing a tool that simulates realistic network traffic and user behaviors. We rented time on a huge supercomputer system with a 2 gigabit Internet connection and proceeded to attack the servers with complex scenarios simulating over 100,000 concurrent users. These tests uncovered piles of bugs and helped us make tons of client and server optimizations. The result: when we went live, everything worked! We had hardly a hitch and things continued to run smoothly, even into Christmas Day (which saw the largest number of simultaneous users online to date).

WHAT WENT WRONG

1 DIDN'T GO TO CHINA EARLY ENOUGH. With 20/20 hindsight, we clearly should have made an effort much earlier in our cycle to understand our hardware manufacturing partners. It is so critical that all the people building your game have a true understanding of that game, of the people who will be playing the game, and of the initial design intent of every feature. Good hardware specs are incredibly important but they only go so far, especially in areas involving the subtleties of feel. Also, the language and time difference [China is a full 12 hours ahead] need to be considered when dealing with Asian manufacturing.

In the early stages of our development, we sent just our industrial designer and our manufacturing agent to China to represent Harmonix. It wasn't until March 2007 that a full time Harmonix employee made their first trip. After that first trip, it became clear that we had to have people on the ground there. From June through September, we had a team there basically full time and that was a critical part of our success. Had we sent people there earlier, we wouldn't have needed the intensive 3 month grind that it took to get our drums and guitars on boats.

2 DIDN'T HIRE AGGRESSIVELY ENOUGH. From the outset, it was clear this was the biggest game we'd ever made. The inherent scope of the project was such that it couldn't be significantly scaled back. For competitive reasons, missing our November launch wasn't an option either. The only way to

complete the title on time was to grow the team. We put most of our staff onto the game, and expanded the organization to new levels. Our offices were bursting at the seams. With staff spread across three floors, and no room left for the new hires we needed to finish the game, we bit the bullet and moved the entire company to a larger space mid way through Alpha.

Despite all of this, we still didn't hire aggressively enough. Many years making small, tightly focused games had ingrained an efficiency bias and "smaller is better" mentality that was hard to shake. We were afraid of the additional management required to hire more people, and it resulted in a longer harder crunch for all of us.

Things became most acute on the code team, with pretty much everyone working insane hours to complete the game. For a while in alpha, we fooled ourselves into thinking it was too late to integrate any new hires. Yet when things got really hairy in beta, and we looked dangerously close to slipping, we did just that. The coders we added in the eleventh hour were an essential part of the collective push to finish in time for a Thanksgiving launch. It would have made our lives much easier if we'd made those hires months earlier.

3 LACKED SENIOR BANDWIDTH. This beast of a game ate up senior bandwidth like no other we'd worked on. We assembled a huge team of leads and sub-leads. Every Monday morning we'd meet and fill a large conference room. This senior powerhouse drove the game forward, but was seriously overworked. Our own internal postmortem revealed that in too many cases these people, who included senior management, had been doing the job of two people each.

Had we realized this during the game's development we could have done something about it, but ironically it's precisely because all of us were so busy that we didn't truly realize how widespread the problem was. Each lead was so busy working with their team to keep their goals on track that collectively we just didn't make enough time to assess the bigger picture.

Our senior figures and team leads often provide the best design insights, and have been instrumental in crafting our earlier games. But they can only do this if they're playing the game regularly, and because of the huge demands on their time this was a real challenge. We plugged them into weekly play sessions, and numerous internal playtests to gain their design insights, but still big swathes of the game didn't get the relentless internal examination they deserved. I'm not convinced that many people outside of Q/A played some of the less mainline game modes like Bass Guitar Tug of War, or online Vocal Score Duel for example. They just didn't have the time.

4 STRUGGLED TO DEAL WITH MULTIPLE UNIQUE CONTROLLERS. We knew going into ROCK BAND development that one of the challenges we'd have to overcome would be getting three different controller experiences working in one game, but we underestimated how entrenched the single controller type mindset is with console manufacturers and certification, and how much that would impact our own plans for shell flow and online play.

The concepts of an "active controller" and "profiles attached to controllers" work great when you have one type of controller that you want a player to stick with (like a joystick), but work horribly when you have multiple profiles signed into multiple controllers that have different functions (like a drum kit, guitar,



and microphone). Since there's no need to swap controller types in most games to be able to play all areas of the game, there is no functionality on the system level to swap profiles between controllers, or allow one controller to use another profile's data. Systems like online matchmaking and invites complicate this even more—inviting is simple on a single controller, but on a multi-peripheral game it raises issues like “what if there is already a drum player in the session?” and “what if I want to play guitar, but my vocal controller is signed in? Can I sign out and sign in without leaving the session?”

For months, we struggled to reconcile our multiple controller types reality with first party certification requirements and system architecture designed for ‘one controller type at a time’ games. As we engaged in the technical and certification challenges with these systems, all essential to solve, it left very little time to properly test and refine usability. We ended up releasing something that functioned and passed certification, but it could certainly have been friendlier.

5 DIDN'T DO ENOUGH TECH DESIGN. Historically, we've managed without much formal technical design documentation. This approach, or rather the lack of lack of one, probably sufficed in the past because our programmers were actually superstar designer-programmer hybrids who considered both technical and design impact as a matter of course.

Expanding our code department for ROCK BAND meant bringing in new programmers who were very talented but weren't all designer-programmer hybrids, or if they were didn't know they were allowed to be. Too often we jumped headlong into implementation of a new system without taking the time to properly examine the implications or test the edge cases of the design. This bit us in a few areas, notably online matchmaking, which had to be redesigned multiple times.

No doubt about it, jumping into development of complex new systems without a technical plan upfront was a flat out mistake. We've now formalized our design process to include code review and a technical design document before implementation of a new system begins.

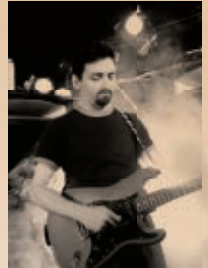
THANK YOU AND GOOD NIGHT!

ROCK BAND was a unique project, with challenges that transformed us as an organization and worked us incredibly hard, but it was an absolute joy to work on. The victories that keep spirits high came thick and fast.

Working on a game with so many firsts was incredibly exciting. Our staff was set out early: this was to be the most ambitious music game ever made, and the spirit of confident innovation was rampant. ROCK BAND is full of features and moments that were born of a team committed to pushing the quality bar as high as possible. Things like the crowd singing along when you're performing really well, the Band World Tour's open ended play, and the Art Maker tool (that allows you to, among other things, design your own tattoos) would not have made it into the game if the talented people who made them weren't so utterly committed to making the game shine.

Seeing the drumming gameplay of ROCK BAND come to life, and realizing how close it is to real drumming has been a definite highlight of the project. We spent a long time breaking down the skills that absolute beginners learn to become competent drummers, and applying that to our note authoring and difficulty tuning. Taking people with zero musical experience, and watching them gradually learn how to drum on ROCK BAND may well be the closest we've come yet to realizing the Harmonix mission: To bring the joy of making music to everyone.

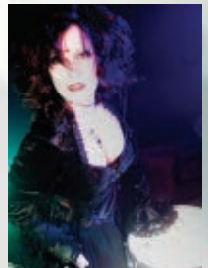
Throughout development and since release, the universal appeal of ROCK BAND has hit home on numerous occasions. The fantasy of being a rock star is one that all of us harbor, and music reaches us on an emotional level that's almost impossible to explain. It has to be felt, and this is what video games do so wonderfully. It's been a particular joy to see children playing with their parents, guys playing with their girlfriends, first-time gamers playing with life-long gamers, and friends playing with strangers. Cooperative multiplayer games are rare, and multi-peripheral games even rarer. In ROCK BAND though, it's hard to imagine it any other way. ❖



Luke Jacobs, O/A Manager.



Greg LoPiccolo, VP of Product Development.



Tracy Rosenthal-Newsom, Director of Production.



Rob Kay, Director of Design.



OUT-OF-THE-BOX INTEGRATION WITH THE LEADING ENGINES



WHAT ABOUT YOURS?



THE #1 VIDEO GAME UI SOLUTION

www.scaleform.com

Scaleform GFX and Scaleform logo are © 2008 Scaleform Corporation. Flash is a registered trademark of Adobe Systems Incorporated. BigWorld Technology logo is registered trademark of BigWorld Pty Ltd. Crytek logo is a trademark of Crytek GmbH. Emergent logo is trademark of Emergent Game Technologies. Unreal Technology, the Unreal Technology logo, and the Circle-U logo are trademarks or registered trademarks of Epic Games, Inc. in the United States of America and elsewhere. Instinct Technology logo is a trademark of Instinct Technology. Hero Engine logo is a trademark of Semutronics Corporation. Trinigy logo is a trademark of Trinigy GmbH. All rights reserved.

Integrated with the leading Middleware Engines

Artist-based **Flash®** UI development

Scaleable image and vector graphics animation

High quality rich HTML text output

Efficient Asian font and IME text input

Available on all PC and Console Platforms



AUDIOKINETIC'S WWISE 2008.1

Sharan Volin

OVER THE PAST COUPLE OF YEARS

some of the biggest game companies, such as Activision, Bioware, and even Microsoft (which has its own audio tool, Xact), have announced that they are using Audiokinetic's Wwise (WaveWorks Interactive Sound Engine) as their audio solution for games. Looking at the latest version, 2008.1, it's easy to see why. Wwise is a very powerful tool, for both the audio programmer and the sound designer. It is available for just about any platform you could possibly need (Wii support was added in version 2007.2).

Wwise consists of 3 components: an SDK for programmers, a sound designer tool, and a game simulator that allows you to connect the engine with Lua for testing. Although the game simulator may only be useful in certain situations, the designer and SDK are extremely useful for any game.

WORKING WITH WWISE

The Wwise authoring application allows sound designers an incredible amount of control over their audio, and once they have worked with their audio programmer to set things up, they can work on the audio all they want and will rarely, if ever, have to contact the programmer. The interface is extremely complex (as can be expected in a tool with so many features), but once you get past the initial learning curve and find what you

need, the actual work is surprisingly easy. You can set up containers to handle random sounds, set up variations in pitch and volume, play sounds in a specific order, and add effects from within the tool (including environmental effects).

Wwise also contains tools to make interactive music easy, allowing you to set up states and even transitions to play when switching between states. Within each of these states you can have many different segments of music and control how you switch between them.

One of the biggest changes in the past year is the Dynamic Dialogue system, introduced in version 2007.2. This easy-to-use system lets you set up different parameters with possible values, and allows you to set up "paths" that contain a specific set of values for these parameters, and set a sound object to each path. Not all parameters need to be set, which allows you to set up fallback sounds that can be used when some parameters are not available. All this can be set up in Wwise and all the programmer needs to know are which parameters to send to trigger the dialogue. This tool should make creating dynamic dialogue much easier for the programmer and give the sound designer much more control over the process.

Wwise is easy to set up so you can have multiple sound designers work on different areas of the project without

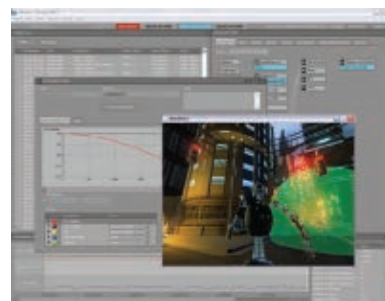
conflicting with each other. It also has some really nice source control integration with the Perforce plug-in, which allows sound designers easy access to Perforce without leaving Wwise, and reminds them to add files to source control whenever they are added to a project (which is sure to come in handy on a busy project).

THE WWISE SDK

The SDK seems to contain pretty much anything an audio programmer might

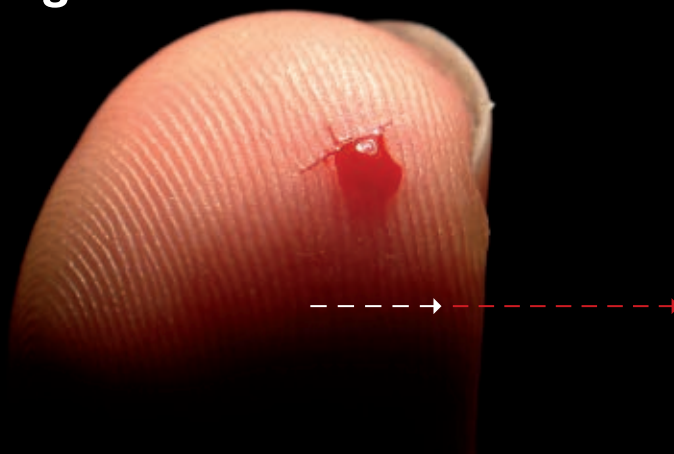
want. Initial setup can be complicated, and it does require a lot of coordination between the programmer and the sound designer up front, but after the initial setup, the rest is fairly simple. The programmer can pass in references to game objects

with positioning to handle 3D sound and can set up real-time parameter controls which allow the sound designer to change audio based on parameters from the game (such as modifying the sounds of a car engine based on RPM). This can be set up as a curve in the sound designer's tool. The designer can also trigger events and states, and trigger changes in interactive



Wwise's profiler enables designers to troubleshoot audio performance.

A minor issue in the beginning.



audio and dialog with only minimal coding. This frees the programmer up to focus more on the gameplay.

Although the SDK is set up to use Wwise's authoring application to create soundbanks, the SDK also gives the programmer many tools to control the audio in code, as well as control over streaming and memory management.

NEW FEATURES IN 2008.1

The main improvement to this new version is the ability to add multiple effect inserts to all sound and music objects. They also made improvements to blend containers, and fixed several bugs.

SUPPORT

Audiokinetic has gone out of its way to help people use its software. Although only paid or evaluation users can contact tech support or use the private forums,

the company has a public forum and wiki available, along with some very good video tutorials available for download through the website to help you get started. It also offers sample projects that demonstrate all the major features, and extensive Windows help files that cover a wide range of features. Although at times the help files were a little confusing, for the most part they were very useful. Audiokinetic also seems to constantly update the software (reviewing it was like trying to hit a moving target) and every release mentions changes and bug fixes that have been requested by customers.

PLANET OF SOUND

In general, Wwise is a very powerful tool and works well in any game project. Although it has a little bit of a learning curve and requires a decent amount of initial setup, once it's set up, it frees both

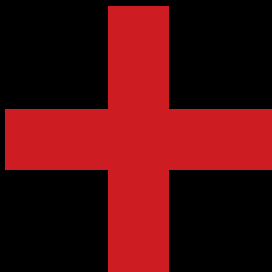
the programmer and the sound designer to focus on creating great audio without having to worry about the technical details. The Dynamic Dialog system makes creating dialog a breeze, and the interactive audio system is also a powerful tool to enhance the game experience. It has a great support system to help you get started and is set up to work well in a real-world production environment. Wwise is well worth considering on any game project, especially if you have any plans to create a complex, interactive audio experience. It's no wonder so many major developers are using it.

SHARAN VOLIN has worked as a programmer for 10 years and is currently a game programmer for Infinite Monkey Factory. She also runs the music website www.rockersonline.com. Email her at svolin@gdmag.com.

WWISE 2008.1

★★★★★

STATS	PRICE	PROS	CONS
Audiokinetic Inc. 409 rue Saint-Nicolas, bureau 300 Montreal, Quebec H2Y 2P4 Canada www.audiokinetic.com	Console Games: \$15,000 Electronically Distributed Games: \$5,000 Non-Commercial Projects: Free www.audiokinetic.com/4105/wwise-licensing.asp	1. Powerful, easy to use Dynamic Dialogue system. 2. Great tools for creating interactive audio. 3. Multiplatform support, including all three current generation consoles.	1. Complex interface. 2. Initial learning curve for first-time users. 3. SDK requires a lot of initial setup and coordination between programmers and sound designers.



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.

product news

3DVIA MP DEVELOPMENT PLATFORM DASSAULT SYSTÈMES

Dassault Systèmes announced 3DVIA MP, a high-end development platform that features a realistic, real-time rendering engine for PC and Xbox 360. There are two forms of 3DVIA MP—Prototyping and Production. 3DVIA MP Prototyping helps developers build prototypes on PC and test them directly on their target platforms while 3DVIA MP Production is a complete and flexible integrated production pipeline designed for multi-platform game development projects. Dassault Systèmes says that it has plans to introduce PlayStation 3 and web versions of 3DVIA MP this summer. www.3dviamp.com

SLICKEDIT 2008 SLICKEDIT

SlickEdit is a multi-language code editor that gives programmers the ability to code in over 40 languages on a variety of platforms including Windows, Linux, UNIX, and Mac OS X. New features in the 2008 version include message list, adaptive formatting, quick brace/unbrace, Perl regular expressions, URLs treated as links, auto symbol translation, new clipboards tool window, and makefile import. A free trial download is available from the company's website. The company also offers SlickEdit Core for

Eclipse and SlickEdit Tools for Microsoft Visual Studio. www.slickedit.com

PERFORCE UPGRADES VISUAL CLIENT PERFORCE SOFTWARE

Perforce Visual Client (P4V), the graphical interface for Perforce's Software Configuration Management system, has been updated to make it easier for users to visualize source code evolution.

Its Revision Graph tool displays branch and integration history as a tree that includes all file branch points, edits, and merges. P4V also visualizes file history with a Time-Lapse View tool that enables developers to look at how the content history of a file has evolved over time. The P4V's Perforce Visual Merge Tool provides graphical three-way merging and side-by-side file comparisons that use color-coding to simplify the process of resolving conflicts that result from parallel or concurrent development. P4V also sports an improved native look and feel for Mac and Linux platforms. www.perforce.com

3DS MAX 2009 AUTODESK


The latest version of 3ds Max is now available from Autodesk. Key features in 3ds Max 2009 include a new Reveal rendering toolset which streamlines

iterative workflows, a ProMaterials material library for simulating real-world surfaces, and improved interoperability with Autodesk's Revit Architecture, Mudbox, Maya, and MotionBuilder software, as well as other third-party applications. The new release also features numerous biped enhancements with additional timesaving animation and mapping workflow tools. www.autodesk.com

PATHENGINE SDK RELEASE 5.15 PATHENGINE

The new release of PathEngine's pathfinding solution features a broad array of speed increases including speedups in tokenized XML loading code, pathfind preprocess loading, and speedups in loadMeshFromBuffer. PathEngine has been further optimized on Windows-x86 with speedups across the board on the platform. PathEngine running on the Cell Broadband Engine SPU takes advantage of new data structure and code size optimizations as well as the addition of custom code paths. An independent 'collision core' can now be loaded onto the SPU for asynchronous treatment of core line collision queries. There is also a new version of the 3DS Max exporter plug-in, with support added for automatic batch export processes in MaxScript. www.pathengine.com

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



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



NOEL LLOPIS

» THE INNER PRODUCT

IMPLEMENTING DETERMINISTIC PLAYBACK SYSTEMS

Part 1: Back to the Future

"Insanity: doing the same thing over and over again and expecting different results."
— (attributed) Albert Einstein

HOW WOULD YOU LIKE TO BE ABLE TO reproduce every crash report that Q/A adds to the bug database quickly and reliably? How useful would it be to be able to put a breakpoint the frame before a crash bug happens?

You can do all that and more if your game is deterministic and you feed it the same inputs as an earlier run. Sound easy? It is if you implement it early on and you keep it that way during development. If you choose not to make your game deterministic, your team will go insane by Einstein's definition, and maybe by a few other definitions as well by the time the project ends.

DETERMINISM

A system is said to be deterministic if, given the same set of inputs, it produces the same set of outputs. Think of your game as a big, black box, with some inputs and outputs.

The two main input types for most games are:

Player input devices. The state of the gamepad, keyboard, mouse, or any other input device is used to update the simulation.

System clock. Most games query the system clock once per frame to retrieve a current time and delta time. The game then advances the simulation forward by that amount of time. Even console

games that expect to run at a rock-solid 60 Hz are often implemented this way to be able to deal with different TV refresh rates or with slower builds during development.

The main outputs of a game system are the pixels on the screen and the generated sounds. Optionally there are other outputs such as network packets or force feedback.

To make the game deterministic we need to make sure that, for a given set of inputs, it always produces the same outputs. In other words, playing the game twice, entering the same button presses at the exact times, the game will end up in the same state both times (same location, same health, number of lives, NPC positions, etc).

What about random numbers, which we rely on so much in games? Aren't random numbers inputs as well? Yes and no. In games we use pseudo-random number generators. That means that the sequence of numbers from a particular seed is always the same. For a given seed, all runs of the game are fully deterministic, so the random numbers are not really an input into the system. The seed to the random number generator is an input, since it will greatly affect the state of the simulation.

RECORD AND PLAYBACK

Recording all player inputs and the system clock is as straightforward as it sounds: At the beginning of the simulation loop, sample the clock and the player input devices. Then save those values to a file and continue the game simulation as usual. You want to make sure the file is flushed after writing the values for each frame. That way if the game crashes you'll have all the input leading up to that frame and you'll be able to play it back right up to the point where it crashed.

Recording the input has a negligible performance impact and the amount of

data saved is very small. As an example, in our current game, every frame we save the frame number, the clock time, frame delta time, and a 20-byte structure for each gamepad (see Listing 1). Without any compression, that's 52 bytes per frame for a two-player game. At 60 Hz, a 20-minute game would be a tiny 3.6 MB, which is small enough to attach to a bug tracking system, email to the team, or archive with the build itself.

Using the recorded input for playback is almost as easy. Every frame, before the simulation starts, we read the input data from the file and feed it to the game as if it came from the system clock and the input devices. A clean way to do that is to separate the reading of the data from where it comes from. In C++ we can use abstract base classes that define an interface describing how to retrieve the data. For example, a class `IGameInput` can define an interface, and the class `GamepadGameInput` reads the data from the hardware, while the class `FileGameInput` reads it from the recorded file (see Listing 2).

In addition to recording and playing back those inputs every frame, we also need to record the seed to the random number generator before the game starts, and read it and apply it during playback. That will ensure that all the random numbers are the same in both runs of the game.

VERIFICATION

If you're going to rely on this recording system, you need to make sure the playback produces the exact same results as the original session. If they aren't the same, the whole system is worthless. Also, because we're just recording input, if the state of the game during playback ever starts to diverge, it will continue getting more and more out of sync from there. We need a way to make sure the playback produces the same state as the initial recording.

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.

Earlier we identified some of the outputs of the game as the pixels on the screen or the sound produced by the speakers. We could compare pixels with a previous run of the game, but the storage requirements would be enormous, and the performance less than ideal. Besides, subtle changes in shaders, lighting, or a simple texture change would throw the comparison off. An easier solution is to check the game state itself. If an enemy is in a different location in two runs of the game, of course it's going to render to a different set of pixels. It will be a lot faster and easier to compare game states than raw output.

During recording, in addition to the game input, we can save some of the state of the game. There's no need to record it all, just some of the most important and representative state. Good candidates include player and enemy positions, prop transforms, score, etc. Unless it's crucial to your game, there's no need to record things like player animations or enemy AI state. Chances are that if any of those diverge, the positions for those entities will also diverge right away.

Once we have this state recorded, we can verify the state does not change during playback. Every frame, we compare the current game state with the

LISTING 1 Game Input Structures

```
struct FameInput
{
    uint32 frameNumber;
    float time;
    float dt;
};

struct RawControllerState
{
    uint32 buttons;
    float leftStickX;
    float leftStickY;
    float rightStickX;
    float rightStickY;
};
```

recorded game state. If they're different, even by the smallest amount, we know something has gone wrong and we flag it right away. The game isn't quite deterministic and something needs to be fixed. We can choose to record and verify the game state anywhere in the simulation loop (as long as they're both done in the same spot), but if we do it after the simulation step instead of before, we'll be able to see what the inputs were that caused the divergence this frame, which will help when debugging. The main loop is shown in Listing 3.

Unlike recording game input, recording the game state can be a much more expensive operation. Traversing the game structures can be a significant performance hit due to cache misses, and the data saved to disk often results in large files. Because of this, in our

LISTING 2 IGameInput, GamepadGameInput and FileGameInput

```
class IGameInput
{
public:
    virtual void GetInputData(
        RawControllerState& state) = 0;
};

class GamepadGameInput: public
IGameInput
{
public:
    virtual void GetInputData(
        RawControllerState& state);
};

class FileGameInput: public
IGameInput
{
public:
    virtual void GetInputData(
        RawControllerState& state);
};
```

current game, player input is recorded all the time, but recording game state is optional, and we can control it through a command-line parameter. Game state verification is also optional, since sometimes we want to play back a recorded set of inputs even knowing that the state of the game is going to diverge due to changes we've made.

For a few of you, checking the game state won't be enough. If you're writing a middleware graphics layer, you probably want to verify that the values you're generating in the back buffer are the same ones from a previous pass. Or maybe you want to check that improving the performance of a rendering algorithms still generates the same image. In that case, you might want to consider something like the Perceptual Image Difference utility (see References), which will be a much less error-prone than comparing exact values for pixels.

MONKEYING AROUND

Once the game is fully deterministic and the playbacks are rock solid, you want to make sure it stays that way. It's all too easy to introduce bugs that will cause playbacks to diverge. For example, a single, innocent-looking, uninitialized variable can change the simulation depending on the value it happens to have for this run.

The best method I've found to stress test the playback system is to use a

LISTING 3 Main Loop Structure

```
while (!done)
{
    SampleClockAndInputs();
    // from different sources
    through the same interface
    RecordInput();
    UpdateSimulation();
    if (recordGameState)
    RecordGameState();
    if (verifyGameState)
    VerifyGameState();
}
```

recorded session of monkey input. The monkey input method consists of feeding the game pseudo-random inputs (as if a monkey were playing the game).

Recording both the input and game state of a monkey input play session, and then playing it back verifying the game state kills two birds with one stone: You get some nice automated testing of your game, and you verify that the game is fully deterministic. It sounds too simple to be useful, but you'll be amazed at how many bugs your first session of monkey input will uncover.

A clean way to implement the monkey input is writing a new class that implements the `IGameInput` interface. This new class will generate game input for all the buttons and axes of a game controller, but it won't come from the hardware game controller or from a recorded session, but from randomly generated values. Apart from being a very clean way to insert input into the game, this approach has the advantage that the monkey input can be recorded just as if it were regular input coming from the controller. That means we can later replay a session and verify its game state, which makes for a very useful functional test.

It turns out that totally random input values are not ideal, as becomes apparent as soon as you implement the naive random monkey input class. If the state of the jump button is truly random, it will be pressed and released almost every frame, which means the player will hardly ever jump high enough off the floor. A better implementation will use a range with a minimum and maximum press time durations, which will give much better results. You might also want to avoid pressing specific buttons (like the pause button, or the restart button combination). Resist the temptation to make the monkey input too smart and make it behave more like a real player, for example, by limiting the number of buttons it can have pressed simultaneously. Part of the benefit of the monkey input is that it will do very unexpected things that no sane player will ever do on purpose. By doing that, it

will unearth many more problems and issues with the game.

One day—shortly after I implemented the monkey input system—the playback verification started failing. Some game entities were ending up in a different state than expected. After doing some digging, I realized that I was using the same random number generator for the game simulation and for the monkey input. Since the playback was just reading input values from the file and not generating them on the fly, the random number sequence was getting off sync right away, causing entities with some randomness in them to behave differently. Lesson learned: Make sure that nothing in the monkey input affects the game systems. In this case, I solved it by using two instances of the random number generator, one for the monkey and one for the game.

PLAYBACK IN THE REAL WORLD

At this point you can easily reproduce bugs. You can re-run the game and put a breakpoint right before the game crashes—except that the crash happens 20 minutes into the playback and nobody wants to wait that long.

Fortunately, you can make time go by faster. During playback you don't care about tearing artifacts, so we can turn off the vertical sync signal, which will speed things up a bit. Most importantly, you often don't even care whether you render anything, so if you turn off rendering completely, you can make the playback run significantly faster, particularly if you were graphics bound. One word of caution: If the rendering part of the game does significant work, and especially if you suspect it might be interacting with the rest of the game to cause a bug or some other source of non-determinism, you might want to do the same work in the rendering system, constructing the push buffer the way you would normally do, but never sending it off to the graphics hardware.

Turning off the vertical sync and disabling the graphics rendering is just reducing the amount of time we spend

in each frame. It still takes 10 minutes to get 10 minutes into the game, you just go through many more frames to get there. The last piece of the puzzle to make time go faster is to be able to set a fixed timestep. Now you can go through the simulation at a much faster rate than real time (and the beefier your computer, the faster it will go). In my current game, we can go through 10 minutes of game time in about 1 minute of real time.

One use for recording and playback that I haven't mentioned is performance comparisons. When you're optimizing the game, you can record a gameplay sequence and gather some performance statistics, such as average fps or longest frames. Then you can apply your optimizations, play back the same input sequence, and compare the performance statistics to see how effective the optimizations really were. Just make sure you use real clock time and not the recorded clock time, otherwise you won't see any differences, even with all the hard effort you put into the optimizations.

There are a few details I glossed over that might prevent some games from begin fully deterministic: network traffic, asynchronous file I/O, and threading issues. I'll cover those in detail next month.

You'll soon find that input recording and playback becomes an essential tool for in development process and you'll wonder how you ever lived without it. Spend a few hours implementing it early in the development cycle and reap the benefits many times over during production. You'll be the hero of the day when that dreaded crash bug from Q/A arrives minutes before a deadline. ❄

Next month's Inner Product continues our look at deterministic playback systems with Part 2. The author acknowledges Jim Tilander for being an idea bouncing board and proofreading the article.

RESOURCES

PERCEPTUAL IMAGE DIFF
<http://pdfdiff.sourceforge.net/>



STEVE THEODORE

PIXEL PUSHER

RAW CRUDE

Twists and turns in the concept pipeline

YOU COULD BUILD A NICE COZY

two-bedroom bungalow out of all the verbiage the games business has expended on “the content pipeline” over the years. The term “pipeline” makes it all sound very rational and linear, the sort of high-tech industrial process that involves hard hats and jumpsuited henchmen. But let’s be honest. If what artists do fits into any pipeline, it’s one of those that you find in Dr. Seuss books, full of crazy loopbacks, recirculations, and about-faces.

Partly this is because even the simple part of the pipeline—the software that’s supposed to get the art out of your content tools and into the game—is mutating and morphing constantly as game designs and engines evolved. But mostly it’s due to us. Even the part of this mythical “pipeline” that we control, the concept and design side, is iterative, messy, and nonlinear. We’re artists. We aren’t built for assembly line production.

If you don’t face that fact (and build your process and schedules around it), you’ll ship stuff you hate, full of bugs you know you could have avoided, or full of crappy content that should never have left the building. On the other hand, once you accept that the art process is fundamentally not linear, you can make some common-sense adjustments that will help make the whole crazy business a little more manageable. So this month

let’s look at some of the ways a concept can get jammed in the pipeline, and some of the virtual Drano you can use to muck out that old pipeline and get things moving.

DOWN THE TUBES

If you still subscribe to the myth of the pipeline, you know that creating concepts is the key to production. It seems logical, after all, that you should try to work out all the messy creative issues in sketches, where they can be tackled quickly and cheaply. A concept artist and an art director can spin up lots of images quickly as they grope for the elusive soul of the new character. They can also use those images to sell the new

the creative work, meaning the team can get by with less experienced artists on the production line. There’s nothing particularly innovative about any of these observations, of course; they are the stuff of many a GDC talk. They’re also music to a publisher’s ears since it helps them pretend that their teams have a master plan that can be followed rationally, step by step and milestone by milestone.

Unfortunately, “appealing” and “likely” are two very different things.

PIPE CLEANERS

To illustrate some of the ways in which this tidy model breaks down, let’s look at a concrete example. The HiveHound (see Figure 1) was a typical game industry monster, an acid-spitting cross between a scorpion and a Doberman of the sort you meet all the time in our business. Despite a very thorough, meticulous design process he was blind sided by the unpredictable, infuriating realities of going from concept art to game asset, so he can serve as a cautionary tale about what you can and can’t get on the strength of good illustrations.

Like most characters, the HiveHound was born as a handful of bullet points in a design doc. The spec called for a medium-sized creature that operates in packs, was

basically feral but had some rudimentary intelligence, and shared a kind of hive-mentality. The drawing stage kicked off with lots of pictures from the web, then silhouettes and sketchy thumbnails. As often happens, the gameplay and the visual concepts ping-ponged back and forth, as fresh ideas emerged and fed off one another, so that the spec which emerged from the first round of meetings was somewhat different from the original design concept.

Nevertheless, at this early stage the process was working pretty much going according to plan: lots of quickie



FIGURE 1 An early sketch of the HiveHound, as the basic formula began to take shape.

concept to the rest of the team, getting feedback from other departments to head off potential problems. And of course—ah, sweet naivety!—since the character has been carefully defined in the concept stage, turning it over to modeling and thence to animation is just a matter of execution.

You can see why this is an appealing idea. Strong, thorough concept work is undoubtedly a Good Thing. It keeps the whole production cycle attuned to clear vision and goals. It allows you to iterate cheaply and minimize risk. It also has the seductive side effect of centralizing

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.



FIGURE 2 So far, so good—the basic design in place without too much wasted work.

pencil sketches and Photoshop iterations that gradually honed those vague designer commandments into something like a graphic formula for the character. Thankfully, it was all done quickly, without a lot of polish wasted on unfinished ideas [see Figure 2].

The next task was to get from “what are we making?” to “what does it look like?” Hence, another round of concept art, this time farmed out to the consummate pros at Massive Black, an outsourced art studio. The sketches coming back from the concept team started taking longer and longer and were becoming more detailed as the formula was refined. As the visual vocabulary for the character solidified, we tried hard to anticipate future problems and head them off. Will that arrangement of neck plates be flexible enough? Can he climb a tree with those limbs? How does that ranged attack work?

Each new question sent additional drawings flying back and forth as we tried to settle those questions early, and, it was hoped, cheaply. At this point we felt like we were just about ready to really start flowing through that pipeline like \$100-per-barrel crude.

PIPE DREAMS

Unfortunately, the progress we were making wasn’t as solid as it seemed. Events would soon show that for every problem that was sidestepped in this phase,

another two still lurked behind it. The feeling that all those problems could be spotted and solved in the concept phase was doubly dangerous. Not only did it turn the concept drawings into something more like a mechanical drafting exercise, it also created a false sense of security.

Having avoided [it was felt] so many pitfalls in advance, it must surely be safe to start driving for the final layer of polish.

With a fat folder of pretty pictures in place, it was time to turn the vision into reality. Since one of the HiveHound’s jobs was to star in a prototype vertical slice, he had to be a fully “next-gen” character with all the normal-mapped goodness, high poly counts, and crazy shader effects you need to impress a potential publisher. This meant that it was necessary to create the scaffolding that would hold up this mighty next-gen edifice as well as the art itself. A mini pipeline for assembling a complex, multimillion poly version of the creature had to be put in place as part of the test, and also a shader system that

CLOGGED

In a traditional concept pipeline, the evolution of the character is over by the time the design is handed off to the modelers. In reality this isn’t the end of the design phase—it’s the beginning of the most important and difficult stage in a character’s evolution. A character is not just a drawing, or a model of a drawing. It’s a living entity. Behavior is just as important as graphic identity for selling the character. Imagine combining the body of Shaquille O’Neal with the moves of Richard Simmons. You’d get—let’s say, a very different character than either a standard NBA player or a standard exercise guru. All too often we treat the design of a character’s behaviors as an afterthought, as a detail to be dealt with only after the “real” work of finding the character’s look is done.

This doesn’t help the artistic integrity of our characters, and it’s a big risk in production terms. The case of the HiveHound illustrates both risks all too well. When the beautifully shaded and rendered character [see Figure 3] was prepped for animation, we discovered a critical flaw that had gone completely undetected through all of the concept iterations that came before it: He couldn’t



FIGURE 3 From concept to fancy next gen model. Unfortunately the design was flawed—the character’s rear legs didn’t allow enough freedom to run and gallop

move right. His proportions were loosely modeled on those of a wolf or hunting dog, but his leg arrangements had been made more angular and simpler to reinforce the insect side of his personality. Without the compressibility of canine rear legs, his fastest gait was sort of a baboon-like lope instead of the bounding wolfish run he’d been intended for.

An equally damning problem emerged when it came time to prototype the creature’s ranged attacks. A lot of energy had been expended on brainstorming how the throwing mechanism would work, right down to detailed sketches of flexible “fingers” that let him flourish his throwing spikes with theatrical menace [see Figure 4]. Unfortunately the concept didn’t make clear that, being attached to the flexing spine of a fast moving creature,

could handle the glowing-acidy-lava-tube thingies that were a key part of the design.

Some of this was unavoidable investment. Unfortunately, the need to proof the pipelines translated into even more pressure to treat the design as finished. The zillion-poly version had to be built because the zillion-poly pipeline needed to be tested. The fancy shaders needed to be tweaked because the shader engine needed to be put through its paces. The unhappy result was that the HiveHound reached the animation and rigging stage as a completely modeled and shaded mesh with weeks of work sunk into the high-resolution model, shaders, UVs, and texture work. It looked great, but serious problems were about to pounce on us.

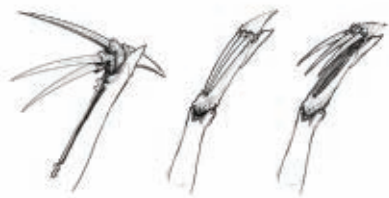


FIGURE 4 Trying to head off problems by detailed concept work is a good idea. Too bad it only goes so far.

NEVERSOFT

THE ASTOUNDING CAPTURE OF ALL HUMANITY!!!

NOW HIRING

- ANIMATORS
- NOTE TRACKERS
- GRAPHICS ENGINEERS
- NETWORK PROGRAMMERS
- SOUND DESIGNERS
- UI ARTISTS (HUD)
- TECH ARTISTS

WE CAME TO SHOOT HUMANS!

ACTION!



APPLY

NOW

AT

WWW.NEVERSOFT.COM



WELCOME TO OUR WORLDS



Blizzard is Hiring!

We are actively recruiting across all disciplines for locations worldwide.

For a full list of openings, please visit us online at

www.blizzard.com/jobs

ACTIVISION®

GREAT GAMES

START WITH

GREAT PEOPLE



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



Visit our site: activision.com

MOM SAID PLAYING VIDEO GAMES WOULD NEVER PAY OFF.

START PROVING HER WRONG TODAY.



Check out our career opportunities now at: careers.thq.com



THQ, THQ Wireless, Rainbow Studios, Cranky Pants Games, Heavy Iron Studios, Paradigm Entertainment, Relic Entertainment, Kaos Studios, Volition Inc., Blue Tongue, Juice Games, Vigil Games, Locomotive Games, ValuSoft, Helix, THQ Studio Australia, XDG, Concrete Games, Incinerator Studios, and their respective logos are trademarks and/or registered trademarks of THQ Inc. and/or their respective owners. All rights reserved.



www.thq.com

THE CREATOR OF **ROCKBAND™**

HARMONIX®

IS HIRING
ARTISTS
PROGRAMMERS

JOBS.HARMONIXMUSIC.COM





Join the **PLAYSTATION®** Team!

Sony Computer Entertainment America, Inc. (SCEA) markets the PlayStation family of products and develops, publishes, markets, and distributes software for the **PlayStation Portable®**, **PlayStation®2** and **PlayStation®3** computer entertainment systems, for the North American market.

We are currently looking for talented and experienced game development professionals for the following areas:

Foster City • California

Software Engineer
Product Marketing
Technical Manager
Programmer
Producer

San Diego • California

Web Developer
Software Engineer
Animator
Technical Support

Santa Monica • California

Designer

At **Sony Computer Entertainment America, Inc.**, we offer our employees exceptional benefits, an open and creative working environment, and the opportunity to be apart of a team that performs at **1.8 trillion calculations per second**.

For additional information, ability to send your resume, add updates, and establish a new business relationship, please visit our website at:

www.us.playstation.com/jobs



PLAYSTATION 3

PlayStation®2



PlayStation Portable

"PlayStation" and the "PS" Family logo are registered trademarks of Sony Computer Entertainment Inc. The Sony Computer Entertainment logo is a registered trademark of Sony Corporation. Equal Opportunity Employer

SONY



**COMPUTER
ENTERTAINMENT®**

Work for a studio with great character



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

Go to www.f9e.com/jobs for current job listings

SOME PEOPLE ARE JUST TOO LUCKY.

FIRST THEY SPEND 2 YEARS ANALYZING AND DEVELOPING GAMES
IN A DEDICATED, CUTTING EDGE ACADEMIC ENVIRONMENT.

THEN THEY GET A JOB, WHERE THEY CONTINUE

DOING WHAT THEY ARE ENTHUSIASTIC ABOUT.

TAKE A LOOK AT www.itu.dk/msc
AND MAKE UP YOUR MIND BEFORE MAY 15TH

STUDY GAMES_

DEVELOP. ANALYZE. PLAY. MSc in IT*

itu.dk/msc

* MSc in IT = en kandidatgrad i IT



APPLY BY MAY 15th 2008



ACADEMY of ART UNIVERSITY

FOUNDED IN SAN FRANCISCO 1929 BY ARTISTS FOR ARTISTS

Study Online or In San Francisco

Pre-College Summer Art Experience Programs
for High School Students
Continuing Art Education Courses

AA | BFA | MFA & M-Arch

Advertising
Animation & Visual Effects
Architecture (*MFA & M-Arch*)
Computer Arts: New Media
Digital Arts & Communications (*AA & BFA*)
Fashion
Fine Art
Graphic Design
Illustration
Industrial Design
Interior Architecture & Design
Motion Pictures & Television
Photography

Accredited member WASC, NASAD, Council for Interior
Design Accreditation (BFA-IAD), NAAB (M-ARCH)

Enroll Now!

79 New Montgomery Street, San Francisco CA 94105
1.800.544.2787 | www.academyart.edu



GNOMON

SCHOOL OF VISUAL EFFECTS

Gnomon School of Visual Effects specializes in training for careers in high-end computer graphics for the entertainment industries.

HIGH-END CG
CERTIFICATE
PROGRAM

PROFESSIONAL
STUDIES
PROGRAM

MAYA
FAST TRACK
PROGRAM



GNOMON|ONLINE

HIGH END TRAINING ANYWHERE, ANYTIME

Gnomon Online offers hours of web based training in high-end computer graphics to help artists become more productive, regardless of their physical location.

5 & 10 WEEK DIGITAL CLASSES OFFERED
ON A QUARTERLY BASIS

NOW REGISTERING FOR SUMMER TERM 2008

www.GNOMONSCHOOL.com | www.GNOMONONLINE.com

For more information about GNOMON, email us at info@gnomonschool.com or call 323-466-6663. 1015 N. Cahuenga Blvd - Hollywood, CA 90038

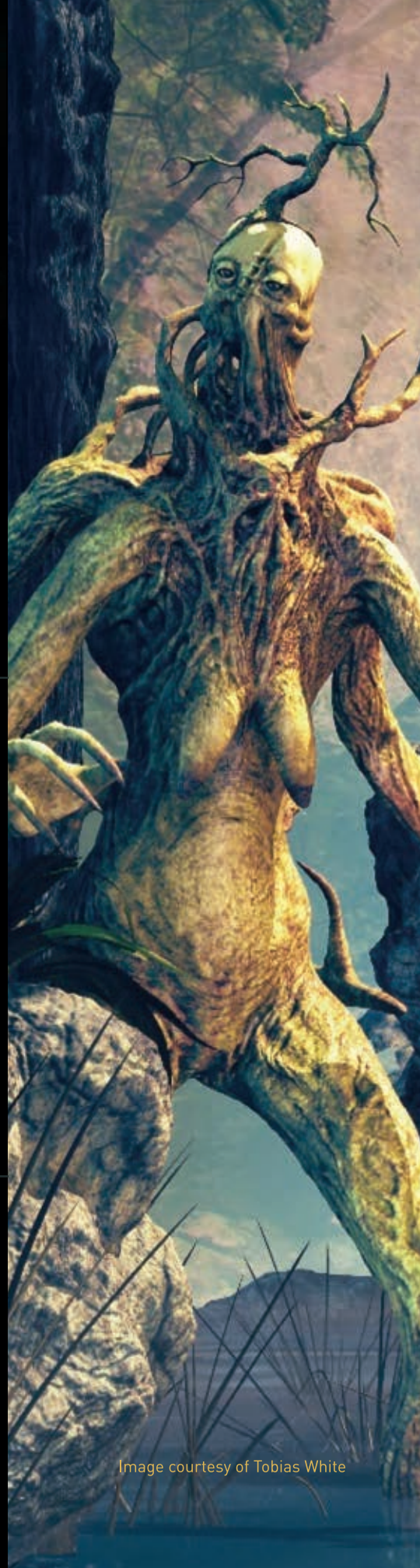


Image courtesy of Tobias White

DREAMERS WANTED.



SCHOOL OF GAME DEVELOPMENT

COMPUTER ANIMATION > DIGITAL ARTS & DESIGN > ENTERTAINMENT BUSINESS > FILM > RECORDING ARTS > SHOW PRODUCTION & TOURING
Financial aid available to those who qualify • Career development assistance • Accredited School, ACCSCT

ONE OF THE TOP FIVE GAME-DEGREE PROGRAMS IN THE WORLD

— Electronic Gaming Monthly



800.226.7625
3300 University Boulevard
Winter Park, FL 32792

fullsail.com

© 2006 Full Sail Inc. All rights reserved.

Ai
The Art Institute
of Portland®

1122 NW Davis St.
Portland OR, 97209
503.382.4804



TURN FANTASY
INTO REALITY & GET
YOUR GAME ON.

Call today & learn more about the
Media Arts degree programs we offer:

- Game Art & Design
- Media Arts & Animation
- Visual Effects & Motion Graphics
- Visual & Game Programming



MAKE MORE ENEMIES

Game Design at Vancouver Film School

The Leader.

Dave Warfield, 15-year veteran game designer for over two dozen titles.

The Know-How.

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

The Results.

Our graduates work at top game companies including BioWare, Radical, Relic, and EA Mythic.



VFS™

vfs.com/enemies

VFS student work by Justin Walters

Do you want to work on a Serious MMO?

We're Hiring: Level Designer
Environment Artist
Character Artist
Interface Designer

Apply online:

<http://sgdi.sbccc.edu>

Serious Game Design Institute

Santa Barbara City College

READY TO GET YOUR GAME ON?

Enroll in NATIONAL UNIVERSITY's **Master of Fine Arts degree in Video Game Production and Design** and be part of the industry that promises continued growth and opportunities for innovative game designers and producers.

You do not need to have prior experience in programming; a passion for video games is the most important prerequisite for this degree!

call today

1.800.NAT.UNIV

or visit our website:

getinfo.nu.edu/mfa-vgp



The University of Values

© National University 2008

CREATE
your
WORLD



CERTIFICATE PROGRAMS:
3D ANIMATION | 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

ADVERTISER INDEX

COMPANY NAME	PAGE	COMPANY NAME	PAGE
Academy of Art University	58	Paynova	39
Activision	53	Perforce Software	16
Autodesk	CV2	Point Of View	19
Blizzard Entertainment	52	Qube Software	6
Center for Digital Imaging	63	Rad Game Tools	CV4
Digimation	31	Replay Solutions	CV3
Electronic Arts	32, 33	Santa Barbara City College	62
Epic Games	15	Scaleform Corporation	42
Fabiano Communications	63	Seapine Software	9
Forterra Systems	35	Sony Computer Entertainment	24, 56
Foundation 9 Entertainment	57	IT University of Copenhagen	57
Full Sail Real World Education	60	Techexcel	27
Gnomon School of Visual Effects	59	The Art Institute of Portland	60
Harmonix Music Systems	55	THQ	54
Havok	23	Trinigy	21
National University	62	Vancouver Film School	61
Naturalmotion	3	Versant Corporation	43, 44, 45
Neversoft Entertainment	51		

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.

GET EDUCATED

[GEEKED AT BIRTH.]



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

- LEARN:**
- DIGITAL ANIMATION
 - DIGITAL ART AND DESIGN
 - DIGITAL VIDEO
 - GAME DESIGN
 - ARTIFICIAL LIFE PROGRAMMING
 - COMPUTER FORENSICS
 - GAME PROGRAMMING
 - NETWORK ENGINEERING
 - NETWORK SECURITY
 - SOFTWARE ENGINEERING
 - WEB ARCHITECTURE
 - ROBOTICS

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



FIGURE 5 The end product of the concept stage was compelling. Too bad it wasn't practical.

CONTINUED FROM PG 50

the throwing arms would gyrate wildly in motion, rendering a convincing throw animation almost impossible.

Fixing the fully detailed and textured model to allow for more appropriate animations was (it needs hardly be said) a costly boondoggle that involved throwing away a lot of otherwise good work. It also played havoc with the character's schedule. Of course, there's nothing unique in this little tale. Indeed, that's why it's worth recounting.

The kinds of problems that emerged as the HiveHound passed from concept to game asset are typical of the nasty little surprises that lie in wait for most would-be game characters (Figure 5 shows the finished concept piece). Movement, gameplay, shader interactions—there are innumerable ways in which a character can confound your plans. The HiveHound's thorough concept phase probably did head off many potential problems before they transpired, but it didn't and couldn't catch them all. It's naive to expect a concept team to

magically anticipate every potential gotcha. The problems are too varied and the expertise necessary to anticipate them belongs to too many people—animators and riggers, designers, engineers, as well as concept artists and modelers. There is no way to plan it all out on paper.

LIGHT, SWEET, CRUDE

The only safe way to avoid long and costly detours into untenable concepts is to be absolutely rigorous about not overcommitting to any design feature until it has actually passed through the whole process from initial sketch to animated character. The concept phase shouldn't end when a sketch or model sheet is passed from the concept team to the modelers. It's not a handoff. It's taking the concept development into a new stage with its fundamental goals—fast iteration and exploration—intact. Turn the sketch into a 3D proportion study, and the study into a set of test animations, before anybody from concept side to

the animation staff starts talking about “polish.” Treat the early models and rigs as tools for refining and proving the concept, not early jumpstarts on the grind of production. Even rapid prototyping can't completely eliminate glitches, but it can help to insure that they are fewer in number and that they are less costly in terms of work wasted and time lost.

Involving modelers, animators, riggers, and designers in the concept phase has broader implications. Expanding the concept effort can be frustrating if it leads to endless meetings or design by committee. On the other hand, enlisting the expertise of all the different disciplines can also be a huge plus. Not only does it help you anticipate problems, it also democratizes the sense of ownership and creative input that makes great work possible. We started off with the notion that pipelines start with artists, and that's always the most important fact to remember about what we do. ❖

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



RAD GAME TOOLS PRESENTS

MILES

Sound System



COMPLETELY HANDLES ALL ASPECTS OF SOUND

WWW.RADGAMETOOLS.COM/

MILES

**MULTI-CHANNEL DIGITAL AUDIO, INTERACTIVE MIDI,
RED BOOK CD AUDIO, 3D AUDIO, INTERNET
VOICE CHAT, DECOMPRESSION, EAX, AND MORE!**

TUNE IN AT:

425-893-4300

**MP3,
OGG, ADPCM**
DECOMPRESSION



**ALL PLATFORMS:
PS3, XBOX 360,
Wii, PC, MAC**

