



United Business Media

▶▶ PRODUCT REVIEW NEXTENGINE 3D SCANNER

SEPTEMBER 2008

gamedeveloper

THE LEADING GAME INDUSTRY MAGAZINE

▶▶ **LET'S GET PHYSICAL**

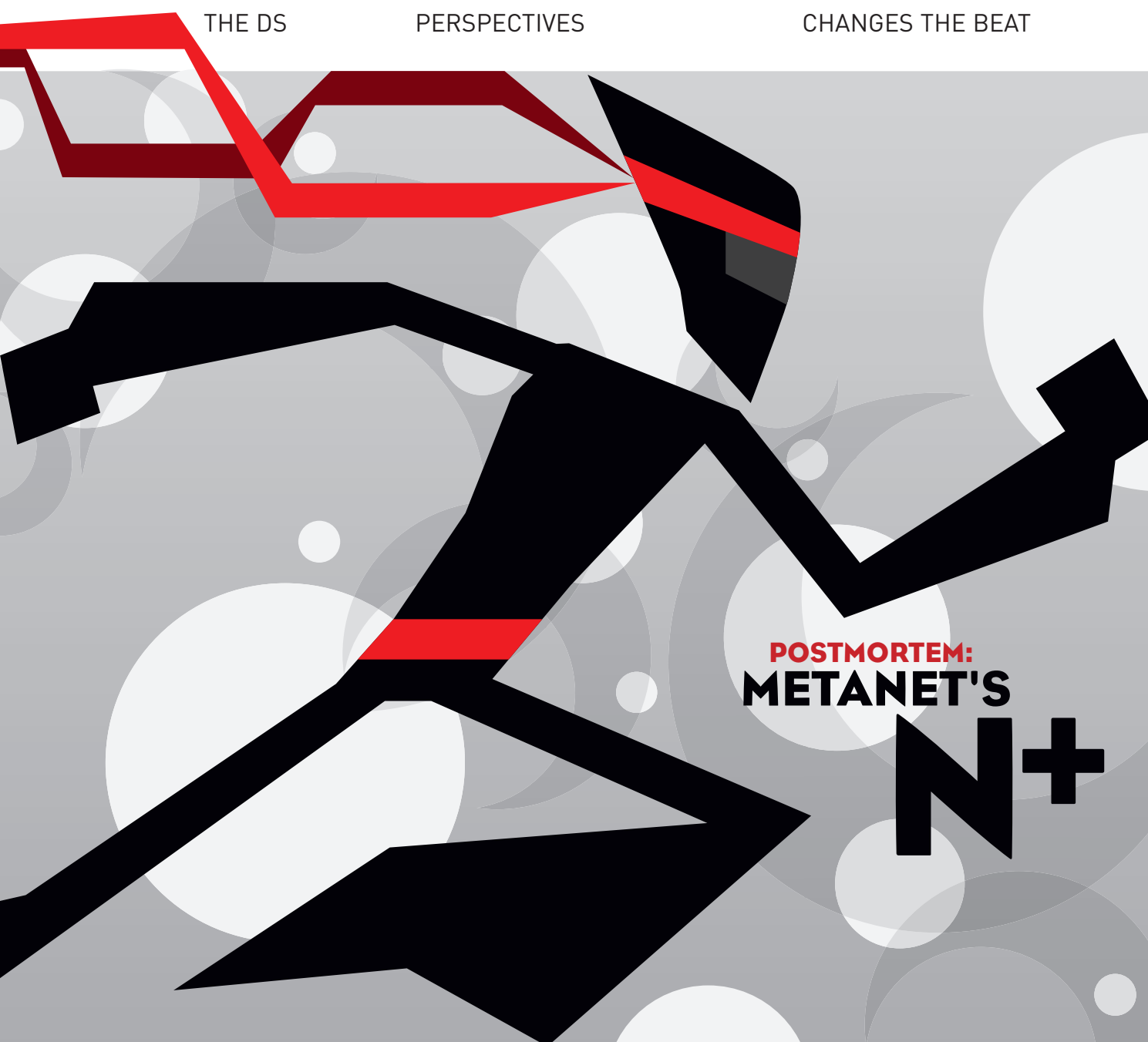
RAGDOLL CODE ON
THE DS

▶▶ **STATE OF THE INDUSTRY**

GAME OUTSOURCING
PERSPECTIVES

▶▶ **MATSUURA'S MANIFESTO**

PARAPPA'S CREATOR
CHANGES THE BEAT



**POSTMORTEM:
METANET'S**

N+



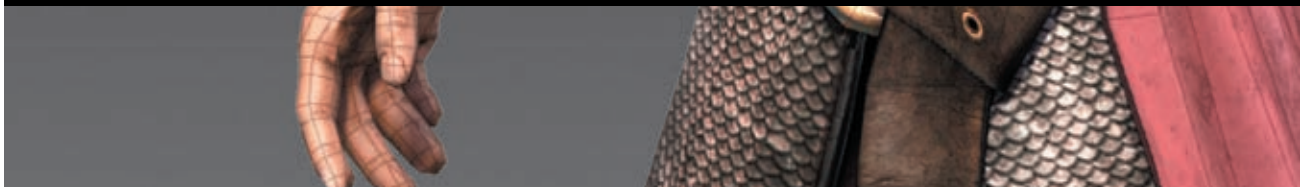
Austin Game Developers
Conference
Visit us at booth #434

“A perfect fit for us.”

Dietmar Meschede, Head of Programming at UBISOFT® Studio BlueByte



Images courtesy by Spellbound/Dreamcatcher and Firefly



VISION7 multiplatform support



Tools & Middleware
PLAYSTATION 3





20

POSTMORTEM

20 METANET'S N+

N+ is the indie success story of the year thus far, with over 100,000 downloads on Xbox Live Arcade. Here, the developers chart their path from the original N to its high definition iteration, contending with TCR problems, networking issues, and cut features along the way.

By Raigan Burns and Mare Sheppard

DEPARTMENTS

2 GAME PLAN *By Brandon Sheffield*
Cahiers du Jeux

4 HEADS UP DISPLAY
Overheard at E3 and Comic-Con, Game Developer Research purchasing survey, and more

37 TOOL BOX *By Tom Carroll*
NextEngine 3D Scanner

COVER ART: GIN GRAVETTE

FEATURES

7 STATE OF THE INDUSTRY: OUTSOURCING

Outsourcing is no longer something game companies are simply "looking into." Outsourcing is now a reality of modern game development, both on the large and small scales. This state of the industry report talks to both outsourcers and outsourcees, to determine where this increasingly important sector of game development is going in the near future.

By Paul Hyman

13 RAGDOLL PHYSICS ON THE DS

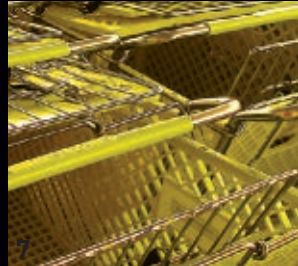
They thought it couldn't be done, but then did it anyway. This article takes you through the steps necessary to implement ragdoll physics on the DS, without eating up your budgets.

By Eric Brown

28 A SENSE OF FUN

PARAPPA THE RAPPER creator Masaya Matsuura feels that music games are missing their full potential. Not only that, he feels that games in general focus too much on competition and negativity. In this op-ed piece, he calls on developers to extend themselves in new directions.

By Masaya Matsuura



COLUMNS

41 THE INNER PRODUCT *By Noel Llopis*
Managing Data Relationships

[PROGRAMMING]

45 PIXEL PUSHER *By Steve Theodore*
Punch the Clock

[ART]

48 DESIGN OF THE TIMES *By Soren Johnson*
Game Economics

[DESIGN]

52 AURAL FIXATION *By Jesse Harlin*
Breaking New Ground

[SOUND]

64 ARRESTED DEVELOPMENT *By Matthew Wasteland*
Depress Releases

[HUMOR]



CAHIERS DU JEUX

DEVELOPERS AND GAME "JOURNALISTS" HAVE

a love-hate relationship. On the one hand, writers for the enthusiast press can drum up interest in a game among the hardcore. BIOSHOCK is a notable example, wherein the dev team used an exclusive GameSpot preview to drum up interest in the game, which was as-yet unfunded. This was a response to publishers thinking they didn't need to spend money on a successor to SYSTEM SHOCK 2, which was not the greatest financial success. In this case, the developer/journalist relationship worked quite well.

But then there are review scores. Developers often feel that journalists harp too much on one point or another, or that the score doesn't represent how they really feel, and myriad other misunderstandings. At the same time, developers seem to respect the journalists, inasmuch as they often will feel that Metacritic ratings are an actual arbiter of quality, or even read reviews to influence a purchase. Journalists, for their parts, often aspire to be developers, and possess a certain idolatry for anyone in a respected company, though they rarely understand exactly what it takes to do the job.

WHO ARE THESE PEOPLE?

Here's what most game developers don't realize about game journalists. First, the grand majority have no training in criticism, English, or journalism. You may not think this matters, but when you feel that a reviewer has unfairly focused too much on one element, such as lack of difficulty in a kids game that wasn't designed for them, the problems start to come to the fore. Most game journalists simply like games, and this is their "qualification" for working in the industry. They are fans, first and foremost, and they understand what they like and don't—but they may not understand why, or how it's done, or how to express it with concrete examples. Journalists generally don't understand the development process, what's simple to execute and what is not, and what can and can't be done. And realistically, they often don't have the time or interest to find out.

A related problem is subjectivity versus objectivity. I've seen many people—developers, and fans alike—call for more objectivity in game reviews. This is completely ridiculous. You don't expect the JRPG fan to like the new MADDEN title. Games are subjective experiences, and that's part of what makes them special. That's what lends them the pretense toward art. All art is subjective! You can't ask for objectivity in game reviewing or criticism—it's simply not realistic. When you have a MADDEN fan reviewing a NARUTO game, you're in serious trouble.

The current system in which reviewers exist is partially at fault. It is completely broken. Reviewers have to cover almost every game that comes out, and most of these persons are male, ages 22–35. As normal members of the human race, they like certain kinds of things, and don't like others. This means they necessarily have to review games within genres they have no interest in, or which aren't targeted toward them in any way. This inevitability, plus the pressure of deadlines, plus the lack of formal training, can lead to exasperated reviewers, and less thorough and so-called "inaccurate" review scores.

A NEW HOPE

Now that you've got the extremely truncated infodump of what's wrong with game journalism ([I have much more to say!]), here are my proposed solutions. First, reviewers should not be aiming for objectivity. If you can get certain reviewers to be known entities, and you understand their opinions, then you have something valuable. An example would be Jerry "Tycho" Holkins from Penny Arcade. Millions of people read Penny Arcade, and understand Holkins' opinions, likes, and dislikes. So when he has an opinion on something, you understand where he's coming from. Same goes for Ebert with movies. If you understand the journalists, you can measure their opinions against yours. But they need to be given the structure and vehicle through which to express this.

Legitimate, understandable criticism is an incredibly important step toward video games entering the mainstream entertainment sphere in full force. It's not that average people need to read reviews, but like with film reviews, criticism and analysis help bring to light industry terms, concepts, and methodology, not to mention the important creators. This is how knowledge of auteur directors and words like cinematography, or concepts like film editing trickled down into the minds of the average moviegoer. If you're a film director, you don't have to tell your family what you do, they understand it. They may not know what a grip is, but it's a start. If you're a game designer, well, that's a different story, isn't it?

A scant few journalists do work in this direction, highlighting important advancements in development, getting developers' names known, writing subjectively, attempting criticism, and advancing the state of the art. If you give these people your acknowledgement, support, and information, we'll all be a whole lot better off.

—Brandon Sheffield

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

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

EDITORIAL

PUBLISHER
Simon Carless scarless@gdmag.com
EDITOR-IN-CHIEF
Brandon Sheffield bsheffield@gdmag.com
SENIOR CONTRIBUTING EDITOR
Jill Duffy jduffy@gdmag.com

ART DIRECTOR

Gin Gravette ggravette@gdmag.com

PRODUCTION EDITOR

Jeffrey Fleming jfleming@gdmag.com

CONTRIBUTING COPY EDITOR

Della Wyser

CONTRIBUTING EDITORS

Jesse Harlin jharlin@gdmag.com
Steve Theodore stheodore@gdmag.com
Noel Llopis nllapis@gdmag.com
Soren Johnson sjohnson@gdmag.com
Damion Schubert dschubert@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 *GreenScreen*

ADVERTISING SALES

MEDIA ACCOUNT MANAGER

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

GLOBAL SALES MANAGER, RECRUITMENT & EDUCATION

Aaron Murawski amurawski@think-services.com t: 415.947.6227

ACCOUNT MANAGER, EDUCATION AND RECRUITMENT

Gina Gross ggross@think-services.com t: 415.947.6241

SR. EVENTS ACCOUNT MANAGER, SOUTHWEST

Jasmin Davé

ACCOUNT MANAGER, WESTERN CANADA, INDIA, AUSTRALIA, & ASIA

Amanda Mae Miller

ADVERTISING PRODUCTION

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

REPRINTS

PARS INTERNATIONAL

Joe Nunziata tnunziata@parsintl.com t: 212.221.9595 e: reprints@parsintl.com

THINK SERVICES

CEO THINK SERVICES Philip Chapnick

GROUP DIRECTOR Kathy Schoback

CREATIVE DIRECTOR Cliff Scorso

AUDIENCE DEVELOPMENT

GROUP DIRECTOR Kathy Henry khenry@techinsights.com

DIRECTOR Kristi Cunningham

kcunningham@techinsights.com

LIST RENTAL Merit Direct LLC t: 914.368.1000

MARKETING

SERVICES MARKETING DIRECTOR Karen Tom ktom@think-services.com

SERVICES MARKETING COORDINATOR Laura Robison lrobison@think-services.com

UBM TECHNOLOGY MANAGEMENT

CHIEF EXECUTIVE OFFICER David Levin

CHIEF OPERATING OFFICER Scott Mozarsky

CHIEF FINANCIAL OFFICER David Wein

CHIEF INFORMATION OFFICER Kevin Prinz

CORPORATE SENIOR VP SALES Anne Marie Miller

SENIOR VP, STRATEGIC DEV. AND BUSINESS ADMIN. Pat Nohilly

SENIOR VP, MANUFACTURING Marie Myers



United Business Media

morpheme™

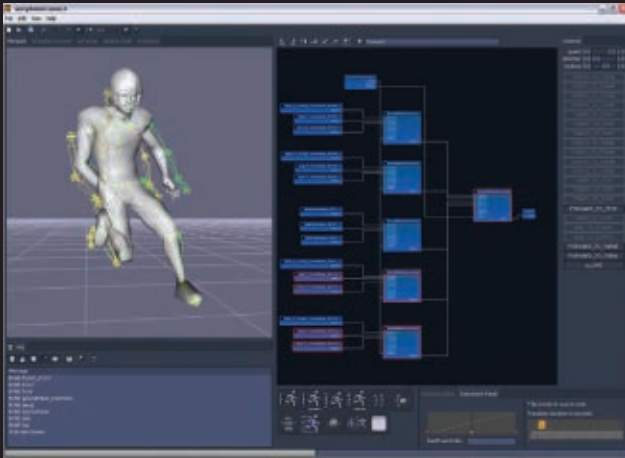
advanced animation system

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



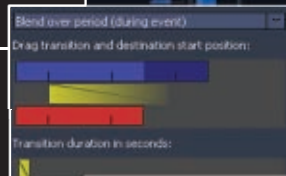
blend tree

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



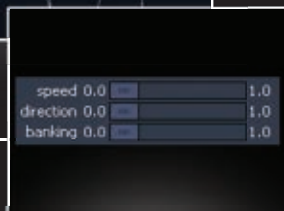
network

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



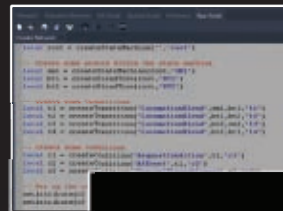
control parameters

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



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



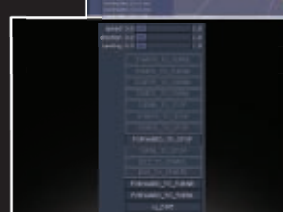
animation browser

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

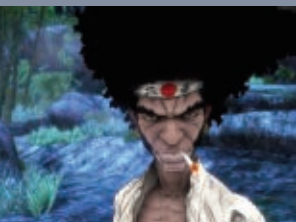


transition requests

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



OVERHEARD AT E3 & COMIC CON

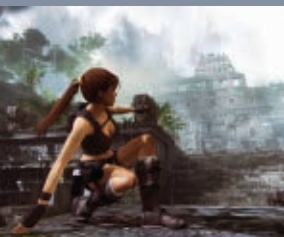


DAVID ROBERTSON

SENIOR PRODUCER OF AFRO SAMURAI FOR NAMCO BANDAI GAMES AMERICA

ON THE COMPANY'S HEAVY USE OF OUTSOURCING: "Easily 70 percent of all art, backgrounds was done by a company called Igloo, in Malaysia. They did an amazing job. We had a huge, year-long vetting process with outsourcers. And they really kick but with their communication strategy. Because you know, art is kind of easy now—people can generate art. It's the communication strategy [that's important], where we need you to not have to re-do the art. Especially in a platform game, that's all you're doing, pushing verts all day long. If you don't own it, that presents a huge problem."

"At the time we started building the game, we had no memory map yet. We were actually building the game at the same time we were building the engine, which was a whole host of drama. You're not supposed to do that. But we didn't have a lot of money, and this was a high-risk project, so we just had to go for it."



ERIC LINDSTROM

CREATIVE DIRECTOR OF TOMB RAIDER UNDERWORLD FOR CRYSTAL DYNAMICS

ON WHETHER YOU COULD MAKE A TOMB RAIDER WITHOUT GUNS OR COMBAT: "There's a feature in the game called player tailoring, which we haven't talked about much yet. It's not about difficulty; it is about people having different ideas about what they want [the underlying TOMB RAIDER concept of] exploration and discovery to be. So for the hardcore user that really likes that action Hollywood experience, there's a help feature where you can ask Lara, 'How do I solve this puzzle?' And then she tells you, and you do it, because you don't want to spend 20 minutes to an hour figuring out this puzzle—you just want to get on to the next thing. Conversely, if there are players that want to spend more time with that puzzle experience, and feel like they just don't want spend time fighting these predators and enemies, you can actually turn that down. We don't have the ability to turn it off completely, because we think those punctuation points are important for the overall pacing, but people can de-emphasize them in order to get the exploration that they're looking for."



SIMON VIKLUND

CREATIVE DIRECTOR ON BIONIC COMMANDO REARMED FOR GRIN

ON DECIDING WHAT TO KEEP WHEN MAKING A REMAKE OF A 20-YEAR-OLD GAME: "That was definitely the most difficult, and as the creative director, most important aspect of my job. To kind of try to look through that haze of nostalgia and realize that not everything about the original is perfect. There are things you can improve upon, so what are those things? What can we change, without the fans—and myself—getting upset? I needed to be confident in what I decided, and hope people wouldn't get mad that I wanted to change certain things around. Like [in the original] you can only bring one radio, you can bring one weapon to an area, and I wanted to change that, so whenever you get an upgrade for the radio, you don't land in an area and find out you brought the wrong radio with you. That happened a lot in the original. And you could only bring one weapon with you, so once you got the rocket launcher in area 5, you pretty much used that all the time, so we wanted to balance that out and also give you the ability to change weapons at any time."



JOSEPH OLIN

PRESIDENT OF AIAS

ON THE EVOLUTION OF E3: "My perspective is that the quality of the work, the variety of the work, is better than ever before, but what's missing for me is that the venue is not as impressive as the work that's contained within it. We're in a medium and an industry that's celebrating unprecedented growth, and unprecedented financial success, certainly on the publishing side, and in terms of how customers spend their money, and this is a reasonably quiet show, don't you think?"

"E3 was always about 'light it at both ends, we are going to burn brightly tonight, and showcase our best people, and our best work about what we do, why [mainstream media] should be here, and why you should be promoting our games for the next six months.' And I don't think that as much as individual publishers and platforms have tried, they didn't have explosions this year in terms of the way they presented their wares that they have in the past. And part of that may be the restrictions that the ESA's E3 committee has gotten agreement to. I'm not privy to that, but it certainly seems like everyone's been told not to be too loud, and I think we should shout. This is the one time of the year I think everyone involved in the game making process should be shouting."

GAME DEVELOPER RESEARCH'S STATE OF GAME DEVELOPMENT REPORT

GAME DEVELOPER AND GAMASUTRA'S SISTER SERVICE

Game Developer Research released its seventh report, the *2008 State Of Game Development Survey*, in August.

The 180-page report was compiled by surveying almost 2,000 video game professionals from North America and beyond who read Gamasutra.com or subscribe to *Game Developer* magazine.

It includes answers to over 55 questions about the platforms Western game creators develop for, the market sectors they are working in, the tools they use, and the amount of money they spend on them. Some of the highlights of the report include the following:

Overall, 70 percent of those replying are making games on the PC or Mac, with 43 percent creating for console and 28 percent for web platforms—just 16 percent are making games for handheld platforms such as the DS or PSP.

Of those developing games for personal computers, 87 percent had chosen Windows 2000/NT/XP as their target OS with 56 percent developing for Windows Vista. Web browsers were also a popular choice for running games (23 percent), followed by Mac OS X (19 percent) and Microsoft's XNA/XNA Studio (15 percent).

When asked what the top three reasons for selecting the target platform for their current or most recent game, developers cited ease of development (47 percent), market penetration/installed base (40 percent), and team skills (36 percent) as being the most important considerations. Interestingly, the performance and speed of the target platform rated lower on the list of concerns at 19 percent.

Of the surveyed console developers, which represent a notable cross-section of the entire industry, 73 percent are creating games for Xbox 360, 58 percent (including some of the same respondents) for the PlayStation 3, and 42 percent for the Wii—with 15 percent still creating games for the PlayStation 2.

This implies that the greatest amount of Western console developers by sheer numbers are creating games for Microsoft's console—but due to team size differences, this doesn't

necessarily imply that more games will appear on the Xbox 360 than other consoles.

In terms of handhelds, of the largely North American and European developers surveyed, Nintendo's DS had the largest amount of creators by numbers, with 75 percent of those handheld developers surveyed making games for it—and with 45 percent making games for Sony's PSP.

Developers were also asked what genre best fits the game they most recently worked on or were currently working on and 34 percent said action, followed by role-playing (20 percent), strategy (15 percent), adventure (13 percent), and education (11 percent). Sports and racing titles were last at 4 percent each.

Looking at the serious games sector of development, Game Developer Research found that 50 percent of the projects were oriented toward the education market. This was followed by science (24 percent), social change (22 percent), military/defense (21 percent), in-game advertising (19 percent), corporate training (18 percent), government (16 percent), health (11 percent), and emergency services (9 percent).

Another particularly interesting result came out of the survey's examination of trends in programming language. Of those

responding, 76 percent are currently using C++ to make games, with 31 percent using C#, and 19 percent using Java/J2ME in their programming efforts. In addition, 9 percent of those replying still use assembly language in some way.

The remainder of the survey offers extra data into the purchasing habits and development choices of the game development industry, with market share information in areas such as AI tools, game engines, 3D art software, compilers, books, motion capture suites, and computer hardware.

The full *2008 State Of Game Development Survey*—including a table of contents with questions asked—is available for purchase from the Game Developer Research web site at www.gamedevresearch.com.

—Staff

CALENDAR

Women in Games Conference

University of Warwick
Coventry
United Kingdom
September 10–12
Price: 200 £
www.womeningames.com

Virtual Worlds Forum Europe 2008

The Bridge
London
United Kingdom
October 6–8
Price: 445–895 £
<http://virtualworldsforum.com>

Tokyo Game Show

Makuhari Messe
Tokyo
Japan
October 9–12
Price: 1,000 ¥ per day
<http://tgs.cesa.or.jp/english>

Project Bar-B-Q

Canyon of the Eagles
Burnet
Texas
October 16–19
Price: \$2,395
www.projectbarbq.com

Fun and Games

Eindhoven University
of Technology
Eindhoven
Netherlands
October 20–21
Price: See Web Site
<http://fng2008.org>

Animation has a new face

- ▶ Superior facial animation for games and film
- ▶ No markers, make-up or special equipment required
- ▶ Used on some of the world's biggest games, like Grand Theft Auto IV
- ▶ Winner: 2008 Develop Award for Technical Innovation

"I officially pronounce that Image Metrics has finally built a bridge across the uncanny valley and brought us to the other side."

- Peter Plantec, Writer, VFXWorld

image metrics

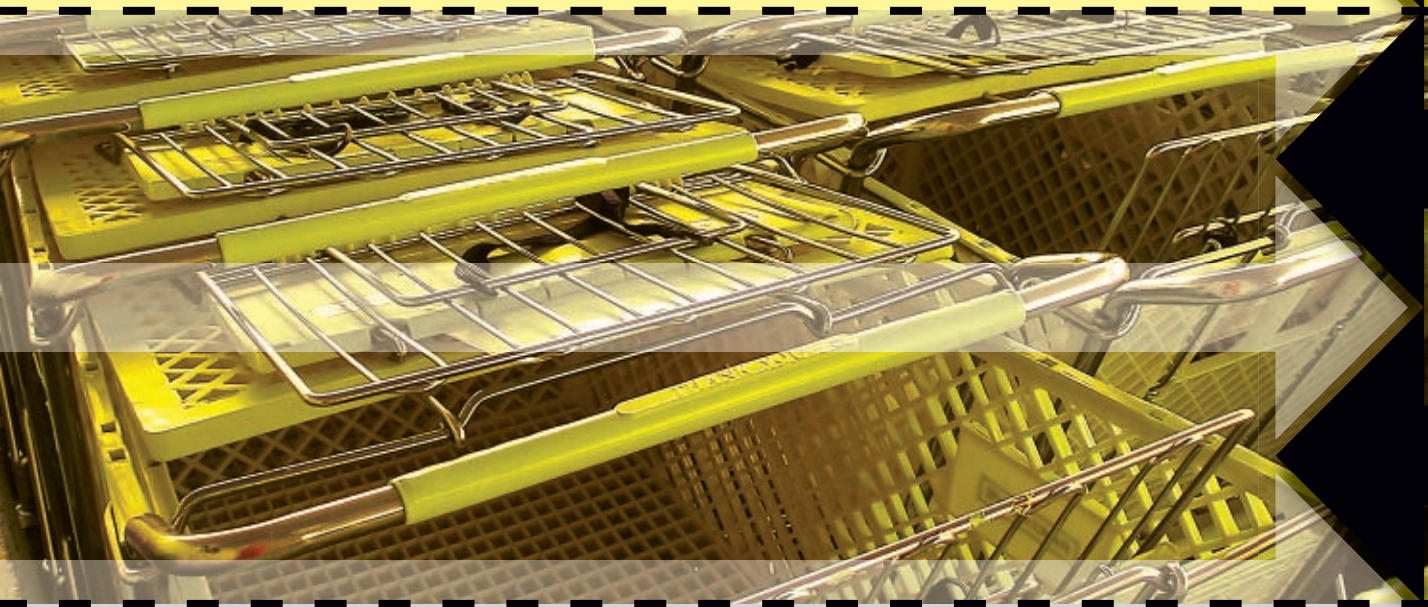
Superior Facial Animation.
Simplified.

www.image-metrics.com

© 2008. Image Metrics, Ltd. All rights reserved.



STATE OF THE INDUSTRY



OUTSOURCING: VIDEO GAME ART IS INCREASINGLY "TO GO"

WHETHER A DEVELOPER RELIES ON JUST A FEW assets to be created elsewhere or depends on contractors to complete the bulk of its production, outsourcing has become more the rule than the exception compared to just a few years ago.

Indeed, some developers—like THQ, Kuju, and Wideload—consider outsourcing to be such an integral part of their corporate strategy that they have taken steps to make their companies more than a little dependent on non-employees to help them meet deadlines.

Meanwhile, outsource companies—like Virtuos and Shadows in Darkness—are gladly accepting their roles as specialists who can partner with developers to save them money, increase flexibility in staffing, and create content with skills that are sometimes superior to those of their clients.

In today's video game space, most developers and contractors agree—if you're not outsourcing, maybe you should rethink why you've chosen not to take the plunge.

THQ'S XDG CALLS IT DISTRIBUTED DEVELOPMENT

In March of 2006, Calabasas Hills, CA-based THQ launched what it has called its XDG unit—which

stands for "External Development Group"—to better manage its outsourcing efforts.

At the time, Shiraz Akmal, then VP of operations and product development, explained that XDG started as "sort of a business development group that keeps an eye on product development and the requirements of our games. We do all sorts of due diligence, and we basically save our studios the time and hassle of determining where the work should go. We make sure that the outsourcers actually exist, that they have the resources and the quality they claim to have, and that they are financially stable."

He predicted that THQ would be expanding its outsourcing efforts from then 20 percent to maybe 40 percent or perhaps 50 percent "in the next few years."

Flash forward almost two and a half years to today ... and outsourcing has become a mainstay of game production at THQ. When we spoke, the company was just about to announce the opening of a new office in Shanghai "from which THQ will spearhead the expansion of local partnerships to develop and publish both online and console games," according to Kevin Chu, now corporate

PAUL HYMAN was the editor-in-chief of CMP Media's GamePower. He's covered the games industry for over 15 years. Email him at phyman@gdmag.com.

OUTSOURCING

director of XDG, who will be moving his office to Shanghai to head up global XDG operations from its base there. "I believe that shows how big a commitment the company has," he says.

First out of the gate will be a free-to-play, micro-transaction revenue-based game called COMPANY OF HEROES ONLINE, designed specifically for Asia, in collaboration with THQ's Chinese



Jonathan Newth, president of Kuju Entertainment.

operating partner, Shanda Interactive Entertainment, Ltd.

In fact, THQ has transitioned from outsourcing to what Chu calls "distributed development," a process in which outsourcers function as an extension of the developer's internal team rather than merely an external producer of piecework.

"We're talking about their contributing to pre-conceptualization, pre-production, prototyping ... everything we do here back in the main studio," Chu explains.

The goal, he says, is to go far beyond traditional outsourcing that might sustain just 20 to 30 percent growth. "If you really want to get up higher than that—perhaps to 50 percent or even 60 percent—you can only do it by changing the way you think about making games, by achieving a level of integration with vendors that I think not a lot of developers are willing to invest the time and training to do. But, through distributed development, we have outsourced up to 40 percent of the assets and we hope to reach 60 percent in the upcoming year, perhaps with a game called DARKSIDERS: WRATH OF WAR—which is an action/adventure RPG scheduled for release on the Xbox 360 and PlayStation 3 this January.

THQ's Vigil Games studio has had artists working with "partner vendors" for over a year now, says Chu, and almost all the concept art for DARKSIDERS was pre-visualized at the company's studios overseas. "XDG has been working very closely with them, trying to train and improve their staff so that they become more experienced with the style of game and more proactive as far as interacting with the THQ team, which is the first stage of a true distributed development environment."

KUJU AIMS FOR 80 PERCENT OUTSOURCING

Similarly, at Kuju Entertainment—one of the UK's largest independent developers—president Jonathan Newth rates

outsourcing a "seven" on a scale of 1–10 reflecting its importance to his company's success.

While an average of 20 percent of Kuju's artwork is done outside the studios, there are some games—the Electronic Arts-published RAIL SIMULATOR, for example—where almost 80 percent was outsourced. And Newth aims to bring the average up to about a third.

"The number one reason—by a significant margin—for our using outsourcing is that it has minimized our fixed costs and given us flexibility in scaling our resources up and down," Newth explains. "We certainly couldn't consider always keeping a full set of teams on staff and trying to time it so that, as one project ends, another one is ready to start. Instead, we maintain a core team of skilled staff and then we increase or decrease resources as we need them. This has enabled us to bid for projects we otherwise couldn't even consider."

In addition to resource flexibility, outsourcing serves up skill sets that may not exist internally at Kuju.

"There are some areas of animation and character creation, for example, that we prefer to outsource simply because we know outsourcers who are capable of producing exceptionally high quality work in these areas—as high quality as we could produce internally and, in some cases, with creativity and experience we might not have in-house," Newth admits.

Kuju—which creates about a dozen games annually—is comprised of six studios in the UK and one in San Francisco with a total of about 330 people on staff, about half of whom are permanent and half are fixed contractors. Then, on top of that, Kuju works on a fairly regular basis with about 20 people who are outsourcers.

The company, which was founded in 1998, began outsourcing its art, music, and voiceover almost from day one and, over the



Alex Seropian, president of Wideload Games.

last few years, has been trying to outsource code as well.

What has also changed over the years is the ease with which Kuju is able to find skilled outsourcers.

"When we started looking at outsourcing, there were fewer around and we knew much less about the whole process and so it cost us a lot of money just to move along the learning curve,"



Newth recalls. “The process we used was to do some research, create a long list of, say, five to seven outsourcers for each separate project, and then do due diligence on everything from their network security, what kind of insurance they had, how long their staff had been with them, and so on. Then we checked references. When the list had shrunk to maybe three to five outsourcers, we sent them our standard RFPs and gave them each a test which was an asset or an animation which we would normally have done in-house. Typically we ended up with two or three outsourcers on the project in order to spread the risk.”

Today the process is much simpler. Kuju has its list of trusted partners and selection is based on costs among other factors.

Newth believes that eventually Kuju might head down the same “distributed development” path as THQ’s XDG, perhaps not on art but possibly on bigger projects where it might try outsourcing scripting or actual code.

“We have certainly had some of the senior members of the outsourcing teams on our site at the start of a project so they can get to know the job and the team they are working with and to understand that they are a big part of the work we’re doing,” says Newth. “So I guess we’ve gone part of the way down that road already.”

OUTSOURCING IS WIDELOAD’S MASTER PLAN

If outsourcing rates a “seven” in importance at Kuju, at Chicago-based developer Wideload Games it rates a “10-plus,” according to president Alexander Seropian, whose five-year-old company’s business strategy was based on the outsourcing concept from day one.

“The fundamental part of our business model is that we have a core staff of full-time employees—now numbering 25—which gets the extra manpower it needs to do all the production work from outside the company,” Seropian explains. “That means outsourcing all the art, animation, sound effects, music, voiceover, and even some of the engineering stuff. On our most recent game, HAIL TO THE CHIMP, for example, we had the help of 15, maybe 20 outside companies. Does that strategy work well for us? Phenomenally well!”

Consider the motion graphic interface in HAIL TO THE CHIMP that needed to look like a TV news program. The Wideload team believed that was a skill that didn’t exist in-house and that it needed to go out and find.

“We did a bunch of due diligence on motion graphics companies that did work for CNN and 20/20,” recalls Seropian. “It just goes to show that some things are better left to outside experts.”

Seropian believes that the primary benefit of outsourcing is not necessarily to save money but to best employ company resources in the most efficient manner possible.

“To me it makes no sense at all to try and hire 100 people, which takes a long time,” he says. “And it’s even harder if you’re trying to hire the best of the best, especially here in Chicago, which is a great city but isn’t exactly the heart of the video games industry. So it’s a big effort, a big risk, and at the end—when you’ve completed production on the game—you’ve got 100 people on the payroll and you only have work for five

or 10. I’m sorry; that model is broken and it’s not one we intend to use.”

But despite the advantages of outsourcing, Seropian is quick to admit that there are also high hurdles, all of which he’s encountered in Wideload’s five-year history.

High atop that list is what he calls “simply process”—which includes setting expectations very clearly and providing outsourcers the tools they need to succeed.

“To make it work, you need to treat the outsourcers—who may be halfway around the world—like they are on the team,” he explains.

“And, for us, that means getting them into our source control system, allowing them to preview their work in mid-game as we are previewing our work, and providing them with a pipeline of assignments and feedback and expectations. For instance, we tell them that when they do a character model, we usually review it about 20 times before we call it done.

Because, if we don’t say that, when they get to the fifth iteration, they’re going to be like, ‘Ah, come on, man. Isn’t it done yet?’ It’s a lot more than just having good communications. It’s treating them like they’re working in the same room as you are. We have come to understand that and to build it into our culture that we aren’t going to succeed if the outsourcers don’t knock the ball out of the park. So we have to do everything possible to enable them to do that.”

CHINA’S VIRTUOS IDENTIFIES THE HURDLES

Gilles Langourieux couldn’t agree more. As founder and CEO of one of the largest outsource companies in the industry, Shanghai-based, four-year-old Virtuos Ltd., he is on the receiving end of all that developers do—and don’t do—to make the outsource relationship work.

While his goal is to take a load off the studio’s shoulders, to do that successfully, he says, the studio needs to put in some extra effort in a few key areas.

“Pre-production is a must,” he explains. “Every asset to be produced must be clearly identified on an asset list and must be linked to clear specs, references, and samples. There should be no room for interpretation, which means that production documentation needs to be more detailed than if it were for internal use only.”

Another potential risk when outsourcing is quality consistency, which can be lower, he says, because the developer’s art director or lead artist can’t walk around the artists’ workstations all day long and pinpoint issues in real-time.

One last hurdle occurs when the developer requires the use of its in-house tools and either doesn’t supply them or doesn’t train the outsourcer in their use.

Given Virtuos’ size—with over 350 staffers in Shanghai and in the recently-opened operation in Chengdu, China’s fourth largest



Kevin Chu, corporate director for THQ’s External Development Group.

OUTSOURCING

city—Langourieux says his company has teamed up with most of the major developers and provided services that focus on art, animation, co-development, programming, and Q/A. Last year, Virtuos produced 3D art and animation for 13 of the top 20 game companies.

"Let's just say that when you play a leading shooter or a sports title or an MMO or the latest RTS," he says, "you have more than a 50 percent chance of coming across art that we produced."

His experience has been that while the due diligence process at most of his clients is fairly standard, the extent to which it is carried out is not. Some developers visit Virtuos for a quick one- or two-hour meet-and-greet visit and run a simple test run on one asset, while others subject Virtuos to full-day audits and several consecutive pilot runs with a different focus each time—quality, consistency, timely deliveries, and so forth.

Although Virtuos rarely has the opportunity to do the same due diligence on its clients, says Langourieux, it does its best to get a feel for the developer and for the project.

"We rely on the developer's reputation, the meetings we have with them, and the pilot runs which greatly help us assess whether we can be successful working together," he explains. "Sometimes we push back on projects which we don't see as appropriate for offshore outsourcing. If both the complexity and the risk of the project are high—and it is the first time we're working with that client—we believe it is our

responsibility to propose to the developer that perhaps they ought to start with smaller steps before outsourcing offshore."

SHADOWS IN DARKNESS' UNIQUE SOLUTION

Naturally, outsource companies feel the same pressures as do developers who outsource—especially to employ company resources in the most efficient manner possible.

But if developers turn to outsource companies to keep their teams trim, who do outsource companies turn to?

At Fort Lauderdale, Florida-based Shadows in Darkness, the six-year-old art outsource company has a unique solution.

Last year, the company—which is relatively small with just 20 staffers—created a separate sister corporation called Darkside Game Studios that is taking on outside programming work, most recently assisting with CIVILIZATION REVOLUTION for the PS3. Its goal is to eventually take on full game development.



Gilles Langourieux,
CEO of Virtuos, Ltd.

"It's really a quite clever plan if you think about it," says Hugh Falk, who is president of both companies. "Just as developers are concerned about what to do with their people when they are in between projects, we—as an outsource company—have the same concerns. Our solution is to let our people with downtime work on projects for Darkside and vice versa, which is essentially outsourcing to ourselves. It eliminates downtime altogether and allows us to keep quality control in-house. What could be better?"

Not that Shadows in Darkness has much downtime these days; the frail economy seems to be taking care of that. When the U.S. dollar was stronger, most of the industry's outsourcing work went to India, China, and Eastern Europe where rates were relatively inexpensive.

But, says Falk, in an unusual twist, some offshore game developers are now outsourcing to the U.S.

"I would say that a good 70 percent of our clients are now based in countries—particularly in England, Canada, and Australia—where outsourcing to the United States makes good economical sense for them," he says.

SOME GOOD ADVICE

Also working in favor of Shadows in Darkness, Virtuos, and similar companies is that outsourcing is increasingly becoming just good common sense.

"Three or four years ago, finding clients took a lot of cajoling mainly because developers just weren't ready for companies like ours," recalls Falk. "They'd tell us, 'Yeah, we're looking into outsourcing but we haven't really done any.' It was tough to get business. Now it seems like everyone is doing it. There's usually someone in charge of outsourcing at every developer, someone who you can speak to directly, who understands the jargon. They know what they're looking for and it has become a much more streamlined process all around. So our biggest issue is keeping up with the demand as opposed to trying to create demand."

And what about developers who haven't stuck their toe in the water yet? What advice would experienced developers offer?

"I remember sitting on a panel about five years ago where there was a very active debate about the value of outsourcing," recalls Kujū's Newth. "There were some very strong proponents of outsourcing, like I was ... and there were those who said they'd never do it ... that they really needed all their staff in-house."

"I'm sorry to say that the majority of those in the non-outsourcing camp are now out of business, mainly because it's very, very difficult to maintain any significant size team using full-time staff," Newth maintains. "That's because as soon as you grow, you suddenly find that there are gaps between projects that are very hard to manage unless you outsource. Would I recommend outsourcing to any developer who hasn't tried it yet? No question. But only if they want to stay in business long-term." ❖



Hugh Falk, president of Shadows in
Darkness and Darkside Games Studio.

What tools are you using?



The Ultimate MMO Toolset



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

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

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

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

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

Isn't it time your tools grew up?

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



www.heroengine.com

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



RAGDOLL PHYSICS ON THE DS

ERIC BROWN is a senior programmer at Sensory Sweep Studios. Eric has worked in the video game industry since receiving his bachelor's degree in physics. He has played a fundamental role in developing the physics and collision engine used for six contracted titles for the Nintendo DS. He is also interested in 3D transforms, especially involving animation systems. Contact him at ebrown@gdmag.com.

WHILE THE NINTENDO DS ENABLES SOME OF THE MOST creative and unique gameplay currently available, it does not have the reputation of being a computational powerhouse. Developing a game for such a platform represents a unique challenge. Having a restricted budget of resources is not new to game programmers, but trying to innovate on a tight platform that you hope can coexist with the gaming experiences provided by the (comparatively) unlimited resources of the next-gen consoles requires a bit of bravery. It is very true that, for average gamers, creativity rather than technological mastery forms the basis of quality for a game on the Nintendo DS. However, it is a very satisfying experience to successfully push the boundaries, not only of the technology, but also of the expectations of performance on a limited device.

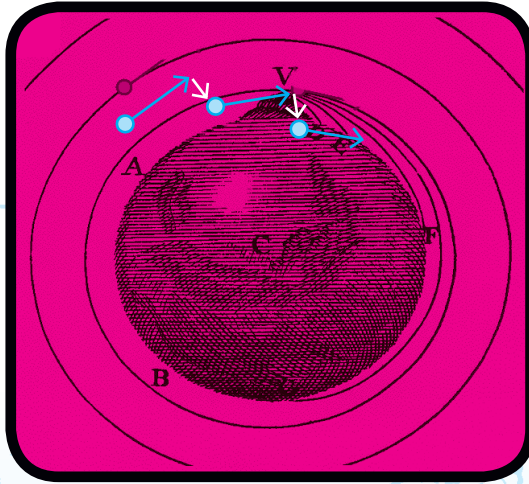
For me it was my own restricted expectation of performance, rather than the lack of computational power, that presented the largest barrier to implementing ragdoll physics on the Nintendo DS. The little brother of gaming actually held its own very well. The main idea that I used for the ragdoll was the common Verlet integration technique.

PREREQUISITES

Ragdoll is a procedural animation technique that represents the interplay of physics and animation. On the physics side, you

will need to have decent collision detection. Collision does not need to be too complex. My collision consisted of static sphere vs. sphere intersection tests for collision with objects, and a dynamic swept sphere vs. triangle test for collision with the world. The triangles that composed the collision mesh of the world were organized by spatial partitioning with a K-D tree. I did not concern myself with self-intersection of the ragdoll, and it turns out that I didn't need to.

In terms of the animation system, you will need to be able to use the information you get from updating the physics on the ragdoll to specify the position and orientation of different joints on the body. If you are using a proprietary animation system, you will need to determine the best place to insert this information. Generally, the transformations of the joints are represented in local space—meaning the transformation is relative to the parent joint—until all key frame and animation blending calculations have been made. Before skinning an animation, the transformation of the joints must be converted to animation space—relative to some character origin—or to world space. Generally the code block where this conversion takes place is the best spot to inject the ragdoll information. If you don't have an animation system, or you feel like modifying your proprietary one will be too hard, you can use the animation system that comes with the Nitro SDK. This animation system



When the state of a particle is represented by a position and a velocity, the next position and velocity can be determined given a particular acceleration. The deviation in the trajectory has been exaggerated in order to show how integration introduces error.

has an appropriate callback that allows you to specify the position and orientation of a joint in animation space.

VERLET INTEGRATION

The basics of a Verlet particle system are documented very well in Thomas Jakobsen's paper "Advanced Character Physics" (which you can find at www.teknikus.dk/tj/gdc2001.htm). The Verlet particle system is ideal for representing a ragdoll skeleton because constraints can be infinitely rigid without causing the system to diverge, thus the distance between joints can more rigidly be maintained. For this reason, you may find that a Verlet particle system is a nice thing to have around, even if you are not planning on doing any ragdoll physics.

The main idea behind the Verlet integration approach to ragdoll is that the positions of the joints are determined by point particles. The "bones" of the skeletons, represented by length constraints that enforce two neighboring particles, are an exact distance apart. Extra length constraints can be added as necessary to prevent unnatural poses, as well as self intersection of the ragdoll. Updating the positions of the particles is relatively inexpensive, and calculating a length constraint is the equivalent of resolving a sphere collision, so the overall cost of updating the particle system is pretty low.

Once you've updated the positions of all of the particles, and then applied the constraints to ensure the appropriate distance relationships between them, it is time to do some collision. I decided to allow the particles to collide with the world as independent spheres, rather than trying to construct a closed collision volume that encompassed a ragdoll limb. This decision was based on two factors. The first was the cheapness factor—sphere collision is much cheaper than some kind of ellipsoid or oriented bounding box collision. The second was that I succeeded in convincing the artists and designers who were creating the collision worlds to "play nice" with the ragdoll—meaning that they should try to make the static collision geometry as smooth as possible, with no really sharp kinks or protrusions that the ragdoll might get hung up on.

However, I still ran into pretty big problems using this approach. If you are using spheres to represent the collision volumes of joints, then the spheres generally need to be pretty small. After all, they are representing things like a hand, shoulder, or hip. Using such small spheres, I immediately began to notice that all of my collision routines would generally fail, due to the numerical imprecision introduced by fixed-point arithmetic. I'll cover some of the things that I learned in order to overcome these precision-related issues.

FX32 ARITHMETIC

FX32 is a 32-bit fixed-point format that is used by the Nitro SDK with 12 bits of decimal precision. With 12 bits of precision, the smallest number that can be represented is 2^{-12} which is approximately 0.0002. It is a mistake to think that 0.0002 is indicative of the amount of precision to expect when performing calculations. Consecutive multiplications accumulate errors very quickly.

For notational convenience, I will define a fixed-point "unit": $FX = 4096$. A fixed-point number can be thought of as a multiplication of a real number by this unit, followed by truncation of the decimal portion. So if I wanted to represent 3.7 in a fixed-point format, I would have:

$$3.7FX = (3.7) * (4096) = 15155.2 \implies 15155 = 3.6999FX$$

We see that we can't actually represent 3.7 in our chosen fixed-point format. Truncating to get an integer representation gives the nearest representable number that is less than 3.7. This error, due to the restricted representation, does not affect collisions until the size of the objects that you are colliding approaches the size of the smallest representable number—namely 0.0002. Indeed, for the sake of collisions, it is probably just fine to consider that the integer 15155 represents the fixed-point number 3.7FX. We will see that there is a much more significant source of error that occurs during calculations.

Using this FX notation, we can see that addition does not introduce any type of error, since normal integer addition can be used to add the fixed-point representations:

$$X(FX) + Y(FX) = (X + Y)(FX)$$

Thus, using integer addition to add together the fixed-point representations of the number X and Y results in the fixed-point representation of the number X + Y, which is exactly what we want. Again, if you want to be careful, you would specify that X and Y are representable numbers. However, for the sake of collision in games we will just assume that all numbers are representable, until we start to collide things like molecules or bacteria. Now consider the case of multiplication. If we merely perform a normal integer multiply on the fixed-point representations of the numbers X and Y we would have:

$$X(FX) * Y(FX) = (X * Y)(FX * FX)$$

However, the right side of the equation is *not* the fixed-point representation of the number $X * Y$. We have an extra power of FX. To compensate for this extra power of FX, we can define a new fixed-point multiplication routine that performs an integer multiply on the fixed-point representations, and then divides



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 Best Engine Front Line Award for the past three years, and "Gears of War," the 2006 Game of the Year, sold 5 million units on Xbox 360 and PC.

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

Upcoming Epic Attended Events:

CEDEC

Tokyo, Japan
September 9-11, 2008

Austin GDC

Austin, TX
September 15-17, 2008

Tokyo Game Show

Tokyo, Japan
October 9-12, 2008

Please email:
mrein@epicgames.com
for appointments.



UNREAL ENGINE 3 AND THE BOURNE CONSPIRACY

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

High Moon Studios recently shipped Robert Ludlum's *The Bourne Conspiracy* for Xbox 360 and PlayStation 3. When asked about its critical development decisions, the studio said the first step it took in bringing a virtual Bourne to gamers was licensing Unreal Engine 3.

"Unreal Engine 3 brought a decade of development in tools which allowed content creators an unprecedented level of control and power," said Clinton Keith, chief technology officer of High Moon Studios. "Not only was the team productive from day one, but we all learned better ways of creating games as well."

Keith added that although this was the studio's first time with Unreal Engine 3, the support services that Epic provided were a great resource during and after the climb up the initial learning curve that the introduction of any new technology brings.

"Apart from the tools, the benefit from having the entire pipeline working on day one was huge," explained Keith.

"While you may develop your own engine, there is always an overhead of problems and constraint on how many resources you can dedicate to solving those problems. By using an engine from a company that has a support business model, you get the benefits of having more resources dedicated to solving problems."

High Moon worked with Oscar-nominated screenwriter Tony Gilroy, who wrote the three *Bourne* films, as well as Jeff Omatta, the fight choreographer, to bring the cinematic style and tone of the *Bourne* universe to consoles.

Sean Levatino, High Moon's senior technical designer, said that using Unreal Engine 3 allowed the team to focus on new gameplay elements right out of the box.

"Epic provided us with a robust suite of tools that allowed our artists to start developing the game from day one," said Levatino. "The Kismet system is a programming language, but it's very visual. It allowed designers to create complex setups basically on their own without requiring programmer intervention."

One key gameplay feature in *The Bourne Conspiracy* is the ability of Jason Bourne to turn any object and any environment into a lethal weapon. He doesn't use guns as the hunted Bourne, which means injuring enemies must be done creatively. As a result, there are over 300 takedowns in the game in which the environment is used to eliminate threats.

"Kismet allowed us to create the takedowns and the fighting system," explained Levatino. "We have hundreds of iterations of certain assets in the game. All of our props are destructible, and everything in the environment can be used as a takedown. Unreal Engine 3's prefab system helped immensely with that."

"Kismet really changed the way we make games. Tools like the animation system, the character system and the package and asset pipeline have all been good for us."

Valdez said the *Bourne* cover system was based on the *Gears of War* cover system, but High Moon modified it to fit the tone and pace of the game.

"All of the takedowns work through the cover system," added Levatino.

"We recognized early on that all of our takedowns are based on walls or

something waist-high like a desk or railing. The cover system detects what type of object you're against, and it allowed us to tag these takedowns across every square inch of a level."

Unreal Engine 3 also enabled High Moon to gear up development for a simultaneous release for Xbox 360 and PlayStation 3.

"Launching simultaneously across platforms is always difficult no matter what your platforms are," added Valdez.

"Unreal Engine 3 certainly made it that much easier to do. In addition, 99.9 percent of the assets in the game are exactly the same on both consoles."



Screenshot from Robert Ludlum's *The Bourne Conspiracy*

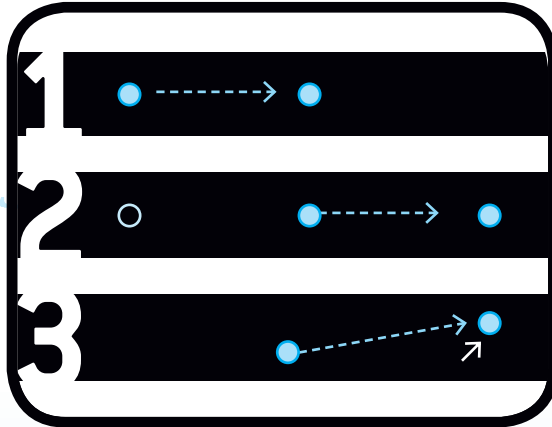


For UE3 licensing inquiries email:
licensing@epicgames.com

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

WWW.EPICGAMES.COM

RAGDOLL PHYSICS ON THE DS



Verlet integration can be conceptually understood in 3 steps. 1) The state of a particle is represented by its last 2 positions. The velocity vector is not stored, but it is implied by the positions. 2) We would expect that inertia will carry the particle an equal distance over the course of the next frame. 3) The position of the particle is modified slightly by the application of acceleration.

by FX. For FX32s, dividing by 4096 is the same as shifting right by 12. A likely macro that performs a fixed-point multiply would look something like this:

```
#define FX_MUL(X,Y) (X * Y)>>12
```

The surprising thing to notice is that with this definition of multiplication, the smallest number that can be multiplied by itself without resulting in 0 is 0.015. This is a very drastic decrease in the precision of representable numbers. This imprecision no longer affects collision of microbes and bacteria, but rather, the collision of things like ants. If your collision routine involves more than one multiplication, then it will affect collisions of things like hands or shoulders or hips. And this is what was causing all of my collision problems.

One solution to this is creating a new representation that allows more precision. For instance, I have recently built a fixed-point math library called FX32e, which has 27 binary digits after the decimal. This obviously puts a very tight constraint on the largest representable numbers, but it is very useful for making objects that represent normalized vectors, quaternions, or rotation matrices, where all of the components are close to unity. However, this was not how I fixed the collision problems I was having. Rather, I found a couple of ways to cheat the precision difficulties that arise from multiplication.

CHEATING FX32 IMPRECISION

The thing that causes the imprecision in the fixed-point multiply is the bit shift. Integer multiplication by itself will not introduce any error whatsoever, in the same way that integer addition does not. However, if we don't perform the bit shift, the result is not the fixed-point representation of the product that is desired. The question that needs to be asked is, "Why do I care?"

The answer is simple: We cannot add numbers or perform comparisons unless the two numbers involved belong to the same "FX space." The integer multiplication can be thought of as leaving the resulting product in the FX² space, and the bit shift

can be thought of as projecting the product back down into FX space. We cannot add or compare two numbers unless they are homogeneous in FX, or belong to the same FX space.

Consider a simple intersection test for two spheres:

```
bool SphereIntersection(Vec Center1, FX32 radius1,
Vec Center2, FX32 radius2)
{
    Vec d = Center1 - Center2;
    FX32 r = radius1 + radius2;

    FX32 d2 = FX_Mul(d.x, d.x) + FX_Mul(d.y, d.y)
+ FX_Mul(d.z, d.z);
    FX32 r2 = FX_Mul(r, r);

    return (d2 <= r2);
}
```

In this routine Vec is a Vector data type with fixed-point components, and FX_Mul is the macro that was defined earlier. The result of this function is a comparison of two numbers that are in FX space. If instead of using the FX_Mul macro, we just used a regular integer multiply, the comparison would be between two numbers in FX² space. An example of how to increase the precision (and add a slight speed boost!) is to merely replace the FX_Mul macros with a regular integer multiply:

```
bool SphereIntersection(Vec Center1, FX32 radius1,
Vec Center2, FX32 radius2)
{
    Vec d = Center1 - Center2;
    FX32 r = radius1 + radius2;

    FX32 d2 = d.x * d.x + d.y * d.y + d.z * d.z;
    FX32 r2 = r * r;

    return (d2 <= r2);
}
```

The lack of the bit shift means that the representable size of a sphere must be smaller to prevent overflow on the integer multiplication, but the routine will still provide correct (in fact much, much better) results. This is such an easy change to make, and had such a drastic improvement in terms of precision, that I immediately began to doctor up all my other collision code.

I developed some general rules for safely replacing fixed-point multiplies with integer multiplies:

- FX32 numbers accumulate a power of FX for every integer multiply, so keep track of the power of FX mentally or by comments in your code.
- Before adding or comparing two numbers, bring them into the same FX space. You can do this by multiplying the number with the lower power of FX by the appropriate power of FX. For instance, if you wanted to compare A in FX² space with B in FX³ space, you would need to first multiply A by FX to get it into FX³ space.
- The larger the power of the FX space that you are in, the smaller the numbers are that you can represent. Thus, if



you fear that you will overflow, you can bit shift to get into a smaller power of FX space.

- Don't pass your result out unless it is in regular FX space, i.e. FX^1 space. (Unless you want to make a new data type for each FX space.)
- If the calculation is sensitive to precision errors, do everything in your power to never bit shift, i.e. try to replace every instance of `FX_MuL` with an integer multiply.

At some portions of my collision code I got up to FX^5 before I was done with the calculation. Of course I had to use 64-bit numbers to do it without overflowing. I expected that doing this would slow the collision code down. Profiling revealed that this was not the case. This is probably because the fixed-point multiply defined in the Nitro libraries already typecasts to a 64-bit integer representation before multiplication in order to increase the range of representable products.

Before employing these tricks, the ragdoll would collide with the collision geometry about 1 percent of the time or less. After employing these tricks, the ragdoll collided perfectly. In fact, the last time I touched the collision code for the ragdoll was to implement these fixed-point tricks, and that was more than 10 months ago. The ragdoll went from being the worst colliding entity in the game to the best colliding entity.

After updating the physics of the Verlet particles, applying the constraints on the skeleton and colliding the independent joints with the surrounding environment, we have a skeletal pose that we need to somehow get the animation system to recognize.

INTERFACING WITH THE ANIMATION SYSTEM

The challenge of interfacing with the animation system stems from the fact that the joints of the ragdoll are being represented as points. A point is merely a position in space. However, the joint of an animation must also carry rotational information in order to properly deform the vertices of the associated skinned mesh. Because the particles that represent the joints of the ragdoll carry no such rotational information, it must be derived before we can properly interface with the animation system.

Another way to describe the problem is to consider that the joints of an animation are represented by matrices, and the joints of the ragdoll are represented by vectors. The vector that represents the position of the ragdoll joint can be inserted into one of the columns of the animation joint matrix, but we still need to determine what goes in the remaining three columns. The problem is figuring this out.

The three remaining columns of the animation joint matrix are the columns of a rotation matrix.

These columns can be thought of as orthogonal unit vectors that define a basis in space. If we could somehow acquire at least two of these, the third could be determined from the condition of orthogonality.

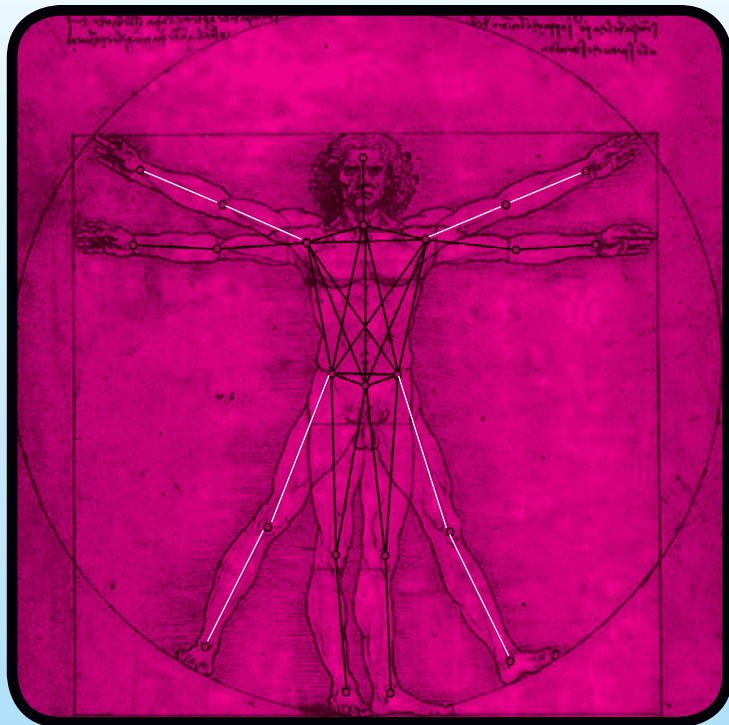
The problem of determining the missing components of the animation joint matrix is reduced to finding an appropriate set of

basis vectors in space. This set of basis vectors represents the "local" coordinate system of the joint—in other words the local x, y, and z directions of the joint.

It is a happy circumstance that when rigging a skinned mesh, a rigger will often align a child joint with one of the coordinate axes of the parent joint. Especially in the case where a joint has only one child, the local transformation of the child is generally translated along only one local coordinate axis. In the case of the character that I was trying to interface with, this was generally the z axis. Therefore, in order to determine the local z direction of a joint, I merely needed to get a unit vector that pointed from the joint to the child joint.

How do we determine the second local coordinate axis? It won't be as easy as determining a unit vector that points from one ragdoll joint to another, since there is no guarantee that the second axis will always be orthogonal to the first one.

Consider that we have a human character standing in a "T-pose" with the arms straight out. The unit vector pointing from the shoulder to the elbow defines one of the local coordinate axes of the shoulder joint. To find the second coordinate axis, we could take a unit vector that points from the center of the character to a neck joint. Such a unit vector would define an "up"

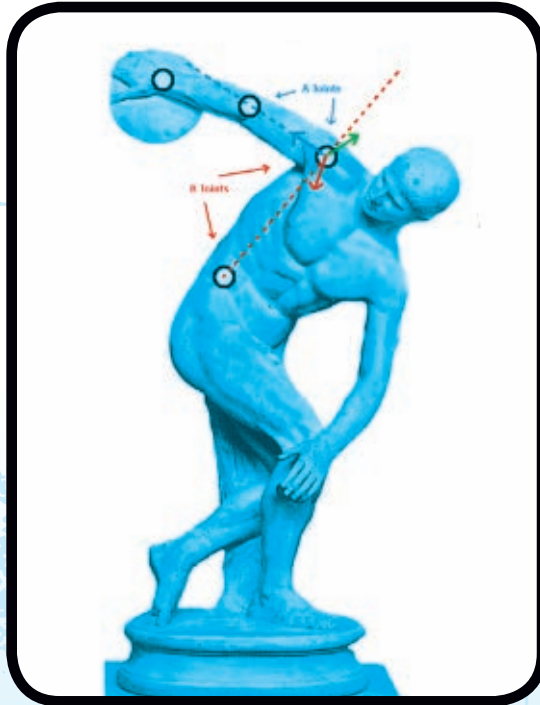


A Verlet-driven ragdoll can be considered a collection of particles that reside at the joints of the skeleton. These particles are constrained to lie within a fixed distance of each other in order to create rigidity in the ragdoll.

direction, and in this particular pose, it is orthogonal to the first local coordinate axis of the shoulder, and could therefore be used as the second coordinate axis.

Once the arm pivots, the axis determined by the up direction is no longer orthogonal, and cannot be used as an axis. To fix this, we find the vector closest to the up direction that is

RAGDOLL PHYSICS ON THE DS



The A axis is the vector pointing from the first A joint to the second—in blue. The B axis is not the vector pointing from the first B joint to the second. Rather, the B axis is normal to the plane that contains both the A and the B offset vectors. In the figure, this plane contains both the blue and red dotted lines. The C axis is determined by the right hand rule, i.e. it is the cross product of the other two axes.

orthogonal to the initial coordinate axis. We now have two orthogonal unit vectors, and the third can be determined by the requirement of orthonormality and the orientation (left/right handedness) of your desired coordinate system.

We can extend this specific shoulder example to a generalized procedure for finding the local coordinate axes of an animation joint.

- Establish a pair of ragdoll joints that represents the first axis.
- Establish a second pair of ragdoll joints that represents the vector that is the goal of the second axis.
- Calculate the second axis by finding the unit vector that is closest to the goal, and also orthogonal to the first axis.
- Calculate the third axis by finding a vector orthogonal to the first two axes, that preserves the orientation of the desired coordinate system.
- Insert these three axes into the appropriate columns of the animation joint matrix.

Of course, if the parent/child transformations do not lie on local coordinate axes, then this procedure will not work. I developed this method based on the fact that I had observed that in the animation rigs I was working with, parent/child transformations almost always would lie on coordinate axes. Of course this procedure could be generalized to specify local directions rather than local axes, but it might be better just to

tell your rigger at the beginning of the project to place child joints on the local axis of the parent. (If you are in the middle of a project, and are in the mood for a fireworks demonstration, simply tell your rigger to re-rig the animation, then take cover before he explodes.)

To facilitate this general procedure, I incorporated a “Ragdoll Mode” into a pre-existing animation tool, where on a per-animation-joint basis we were able to define: 1) which ragdoll joints corresponded to the animation joints, 2) which ragdoll joints were used to determine which axes of the local coordinate transformation.

We called this process “setting up the brads,” where the term brad refers to the set of information that pins—or brads—the animation to the ragdoll. When I was a youngster in elementary school, we would sometimes assemble skeletons out of paper bones. The bones were connected at the joints using brass brads. This seems like a good justification for using the term “Brad,” but the real reason that it is called a brad is because one of our designers—Brad Moss—after making a vital suggestion, asked if we could name some code thingy after him.

The structure of the Brad goes something like this:

```
struct Brad{
    int AnimJoint; //index of the animation joint
    int RagdollJoint; //index of the corresponding
ragdoll joint

    int JointA1; //The first pair of ragdoll joint
indexes that define the "A Axis"
    int JointA2;

    int JointB1; //The second pair of ragdoll
joint indexes that define the goal
    int JointB2; //of the "B Axis"

    int AxisA; //References an enum that
determines if A is the x, y, or z axis
    int AxisB; //References an enum that
determines if B is the x, y, or z axis

    Matrix JointMtx; //The most recently
calculated joint matrix
};
```

The main purpose of this structure is to contain all of the information required to construct a transformation matrix in animation space. After updating the position of the ragdoll is complete, after applying constraints and collision, we loop over the list of Brads, calculate the transformation matrix and store it in the JointMtx element of the structure.

In the general cycle of an animation system, the joints are represented as transformations, but the form of the transformation varies at different points in the cycle. Here is an oversimplified example of the evolution of a single joint through an animation cycle:

- *Keyframe Blending.* Interpolate between two key frames to get the local rotation and translation—local means the transformation relative to the parent joint.



- *Animation Blending.* Interpolate between the transformation values for two different animations—still the transformation is local.
- *Expanding.* Expand the skeleton so that all the joints are expressed in animation space—now the transformations are no longer local. The expansion is performed simply by multiplying the local joint transformation matrix by the expanded parent transformation matrix. If this joint is the first joint in the skeletal hierarchy, then the parent transformation is just considered to be the identity.
- *Prepare for Skinning.* Prepare the list of transformation matrices for skinning by multiplying by the inverse value of the transformation in the bind pose.

The appropriate point to insert the transformation matrix calculated by the brad is in the “Expanding” phase. When the ragdoll hijacks this expand phase, it looks to see if the joint has a corresponding brad. If it does, then it does not calculate the animation space representation of the transformation as normal, rather it just inserts the `JointMtx` that had been previously calculated. This method was the suggestion made by Brad Moss that immortalized his name in our codebase.

PERFORMANCE RESULTS

The animation rig that I used had on the order of 25 bones. I had 15 ragdoll joints with almost 30 length constraints. I would collide each ragdoll joint individually as a sphere using a swept sphere algorithm. The total amount of time to update the Verlet particles, apply the constraints, and collide and construct the brads took around 2000–2500 microseconds. 2500 microseconds is not really a cheap operation, but is definitely manageable considering the 15,000 microsecond allotment of a 60Hz cycle. Actually, we found that when the character was in ragdoll, we could short circuit some other parts of the character update cycle, and it turned out the entire character update cycle went slightly faster when the character was in ragdoll. As such, ragdoll performance never ended up being an issue for us.

Of all of the problems that I faced while implementing this technology on the DS, the largest problem was the psychological aspect. Everyone around me, including myself, believed that ragdoll was way too expensive for the teeny little DS. It seemed like a long shot when the idea was first suggested, and we would often discuss how low we were willing to let the frame rate drop in order to have it in the game. We pressed forward on the assumption that there was only a 5 percent chance that it would actually work well. The greatest lesson I learned from this experience was that sometimes 5 percent is enough to believe in. ❖



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

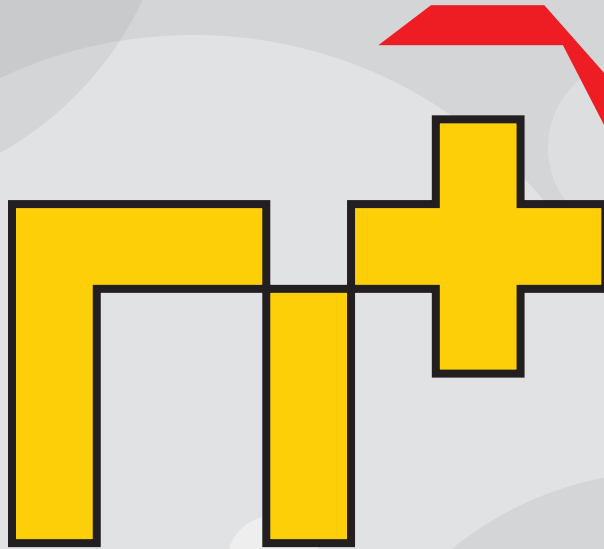
Superheroes Wanted.

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

www.f9e.com/jobs



METANET'S



THE MAIN GOAL OF N+ WAS TO EDUCATE AS MANY PEOPLE AS possible about the threat of robot gold-hoarding, and the brave ninja who work tirelessly to bring justice to the world. We can only hope we've succeeded.

N+ was planned as a next-gen console version of the free downloadable game N, which would feature salvation-inducing upgrades: multiplayer, a handful of new enemies and objects, improved collision and movement/control, impressive HD graphics, a particle simulation including fluid smoke simulation, and an improved level editor with robust online level-sharing capabilities. It's all in the name, really. N, Plus: a faithful reproduction of the game fans knew and loved, plus as many new and exciting things as we could cram in there, without ruining it all.

WHAT IS THIS N+?

N+ is a 2D action-puzzle platformer which captures the feel of classic arcade games but adds an injection of modern physics and style. As a lone ninja, players must use deft acrobatic skill and guts of tempered steel to survive in a futuristic world populated by inadvertently homicidal robots.

The game was co-developed by Metanet Software and Slick Entertainment. Metanet Software is an independent developer based in Toronto composed of Raigan Burns and Mare Sheppard, who provided production, publishing, and level design for N+. Metanet released N on its own in 2004, and bringing N to consoles was the next logical step. For this, they called Nick Waanders and Kees Rijnen of Slick Entertainment. Slick was started in early 2007 by two expatriates of Lost Boys Games, now Guerilla Games in Holland.

The game ships with 500 levels of ranging difficulty (in four types: Single player, multiplayer co-op, multiplayer race and multiplayer survival), 700 more levels on the way via DLC, and a built-in level editor so players can make their own. It was developed in C# and C++, and was built on the Klei engine which was used for EETS: CHOWDOWN. The engine required substantial modifications and additions such as physics, collision, and networking, but provided a solid base for most of the game systems.

WHAT WENT RIGHT

1 THE PROTOTYPE. The existence of N as a "prototype" was extremely useful in the development of N+, from pre- to post-production. N made our vision for N+ very easy to convey—it provided a way for everyone to see and, even better, experience the basic gameplay. N showed the greenlight committee what they could generally expect from N+, and also proved which aspects of the game were successful and which needed to be augmented. The magic of an already-established fan-base, reviews and references to critical reception, and plenty of user feedback gave us a ton of data.

When development on N+ began, N was a finished prototype which answered the questions of design and paved the way for smooth development with available source code and a list of bugs and issues to improve on. It was relatively easy to get N running on the Xbox 360, giving us ample

TOTAL DEVELOPMENT STAFF:
5

TOTAL BUDGET:
\$300k

LENGTH OF DEVELOPMENT:
10 months + 2 months
post-release

RELEASE DATE:
February 20, 2008

PLATFORM:
Xbox Live Arcade

DEVELOPMENT HARDWARE:
Toshiba A200 (1.67ghz Core 2
Duo/2gb/220gb),

Dell Dimension 4550 (2.4ghz
Pentium 4/300gb),
3 A-power PC's (1 server, 2
dev. machines), 4 Xbox 360
development kits.

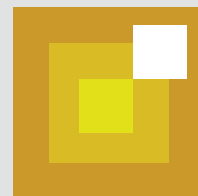
DEVELOPMENT SOFTWARE:
Ned (in-house level editor),
Visual Studio 2005, Visual

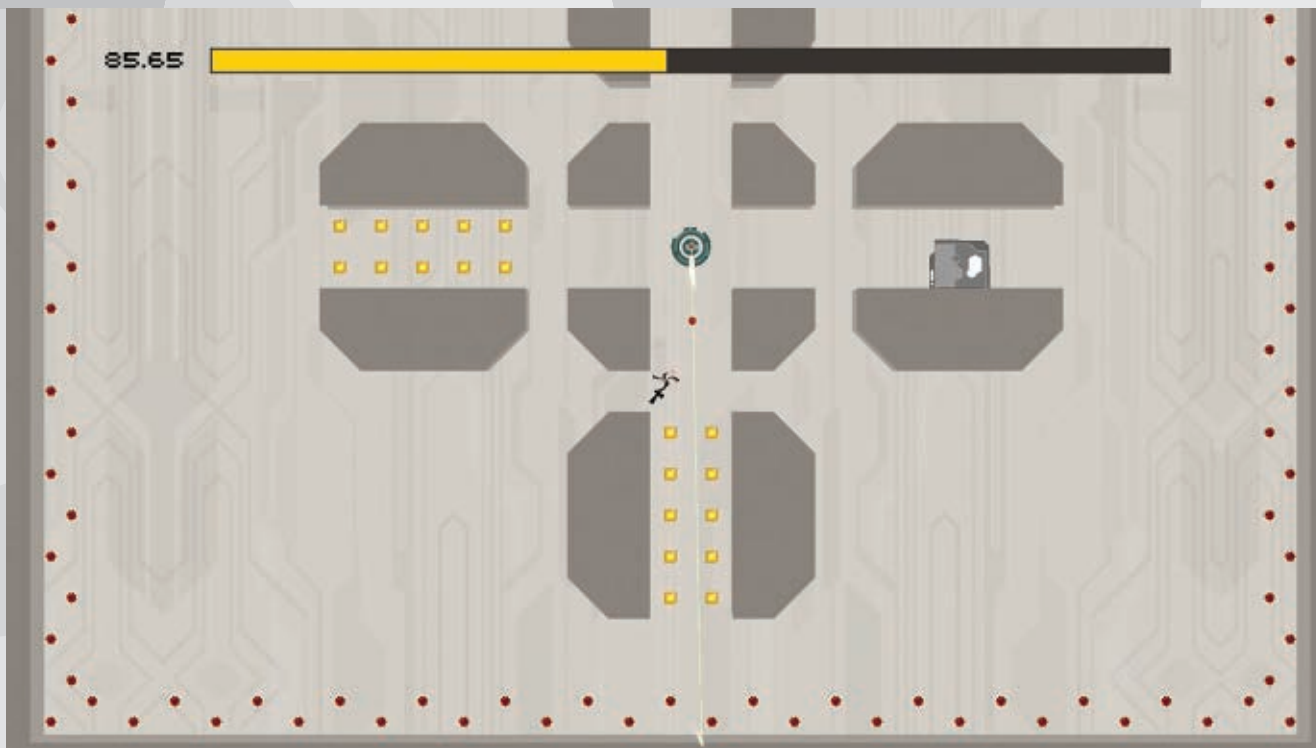
Studio 2008 C# Express,
Adobe Photoshop, Tortoise-SVN.

PROJECT SIZE:
20,1097 lines of C++
code, 11 C# tools built to
support development, 603
logged issues fixed during
development.



MARE SHEPPARD and **RAIGAN BURNS** comprise Metanet Software. They met in 1998 and discovered that when they combine their efforts, they are able to transform into a rather large robot made up of small mechanical cats. Email them at metanet@gdmag.com.





opportunity to tweak the gameplay and get the feel of the controls exactly right, which is the foundation of the whole experience.

2 A SOLID TEAM. N+ had a combination of expert programmers and expert level designers, which led to a well-constructed final product. Slick handled the entirety of the programming, giving Metanet the time it needed to design and rigorously test hundreds of levels (as well as handle publisher-type jobs). That each part of the team was experienced within its respective domains was instrumental to the success of the project.

3 GAMEPLAY FIRST. The basic gameplay was working in the first months, which left us with lots of time for polish and incremental improvements. It took a few tries to get the physics working perfectly, and tweaking the control of the ninja was a small but continuous task throughout development, concurrent with the main tasks of networking, TCRs, and UI. Slick planned and developed dedicated tools to make life easier, which was a huge help given the amount of revision required—the final game has over 60 different screens of UI!

4 KEEPING IT SIMPLE. Near the start of the project, we were under pressure to change the graphical style and really show off the next-gen power of the 360. The main reason we resisted such changes was that when developing N, we discovered that a visually complicated back- or foreground led to a sharp increase in player frustration. Because controlling the ninja requires such precision, everything else needs to be as minimal as possible so players can focus and easily read the state of the world without devoting much extra attention to it.

We had lots of user feedback and critical reviews to back this up, and since Microsoft was open to discussion and debate about the graphics, we were allowed to keep the clean minimal look. This made the update much easier to implement, as there was no need for lots of 3D or complex shader programming, leaving more time for other enhancements such as a robust particle system. The fluid-simulator-based “ninja smoke” was unfortunately cut due to time constraints, but this only simplified the graphical presentation further.

5 EARLY FEEDBACK. We saw huge value in using feedback from N players to improve the design of N+—very few know the game better! We hold the N community in very high regard because their enthusiasm towards providing honest criticism of N and passion for the game in general is unparalleled. We considered several fan suggestions when planning N+ prior to development.

Showing N+ at PAX 2007 was an opportunity for us to get the game into the hands of gamers, listen to their feedback, and make some important changes. One of the things we learned was that the progression of level difficulty, even though it had been refined and simplified several times, was still too hard for the average player. This led to our revising and playing through the levels several more times, to the benefit of gamers everywhere.

WHAT WENT WRONG

1 UNDERUSED NEW FEATURES. Creating new levels using the N+ editor was awkward due to the editor still being under development, and involved several steps to extract and rearrange created levels. This frequently resulted in errors and lost files. Experience and a store of existing levels kept us using

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



"Havok is an essential ingredient in what makes our games great. By consistently integrating the Behavior 5 tools across our projects, our teams are instantly empowered to take their creativity to the next level."

– Josh Resnick, co-founder and general manager, Pandemic Studios

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.

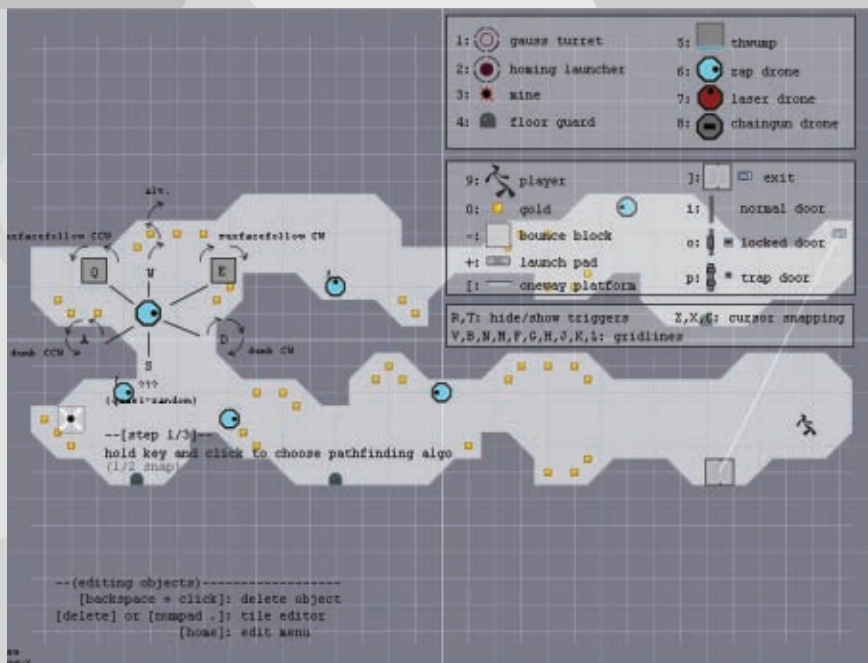
- Out of the box integration with our award winning Havok Physics™ and Havok Animation™
- Over 15 "triple A" titles currently in development using Havok Behavior
- Backed by our first class, industry leading Developer Support
- Optimized for leading platforms including Wii™, PLAYSTATION®3, Xbox 360™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



The Ned level editor built in to N.

the editor built in to N, “Ned,” which allowed us to quickly create and immediately test levels in the N environment.

Because we had been working with Ned for years, we knew how to create successful levels quickly with it—but this meant we had limited access to the new enemies and different level dimensions found in N+. The process for combining Ned-based and N+-based levels was tedious and error-prone; as a result new features were not properly explored, and seldom used. For the DLC, which was created after the game had been released, we dedicated a lot of time to level creation using the finished N+ editor to try to alleviate this problem.

2 NETWORKING. Networking issues were very problematic because this was an area the team had very little experience with. We were aware that this would be a significant challenge, and planned for it with the addition of a dedicated network programmer, and a schedule which saw work start on multiplayer basics only a month after development began. Despite these steps, it still wasn’t enough, and as a result networked gameplay is far from ideal.

The problems were further complicated by the fact that Metanet only tested multiplayer maps locally rather than networked, so were not aware of the network problems where they related to level design, and unfortunately were ultimately not able to design around them.

When VMC Game Labs began testing in August, they uncovered a slew of complex issues that would take a rewrite of the entire network layer to fix, and time was not on our side—the content complete milestone was set for the end of September! The team decided that networking was as good as it

could be given the time constraints, though the severity of the problem was not yet known and so the consequences had not been fully analyzed. Unfortunately, networking glitches are the number one user complaint, which has definitely hurt our review scores, our pride, and our fans. On the plus side, local multiplayer is great!

3 USER-CREATED CONTENT AND LEVEL SHARING. From the start of N+ development, we were in a constant, ongoing debate with Microsoft about how to enable sharing of user-created content. At first, discussion was limited to the technical, “how” it could be done, rather than “if” it could be done. Microsoft supported the idea but expressed a very reasonable need to limit users from creating offensive content such as hate speech, representations of male genitalia, offensive language, and so on. The problem was that severe limitations would cripple the user’s ability to create interesting content, negating the editor’s reason for being. For instance, limiting

levels to only 8x8 tiles would be effective in preventing the tiles from spelling profane language, but would also limit the scope of level creation to an unacceptably restricted degree.

We eventually settled on a number of passable solutions, and development of this feature began. But then, the great FORZA debacle happened.

The gist of the issue is that because of limitations of the leaderboard system, Microsoft was unable to delete the specific offensive content uploaded by a user of FORZA, which could also not be flagged by other users, and was resolvable only by the deletion of the entire user account. Near the end of N+ development, we were told to disable the content-sharing features for launch, with the suggestion that they could be re-enabled when and if the leaderboard back-end was altered to allow effective user-created-content control.

We complied, regretting the backlash that would surely occur, but were optimistic for the future when we could re-enable. Unfortunately, this last-minute change caused a certification



The N+ in-game level editor.

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



WHAT ABOUT YOURS?



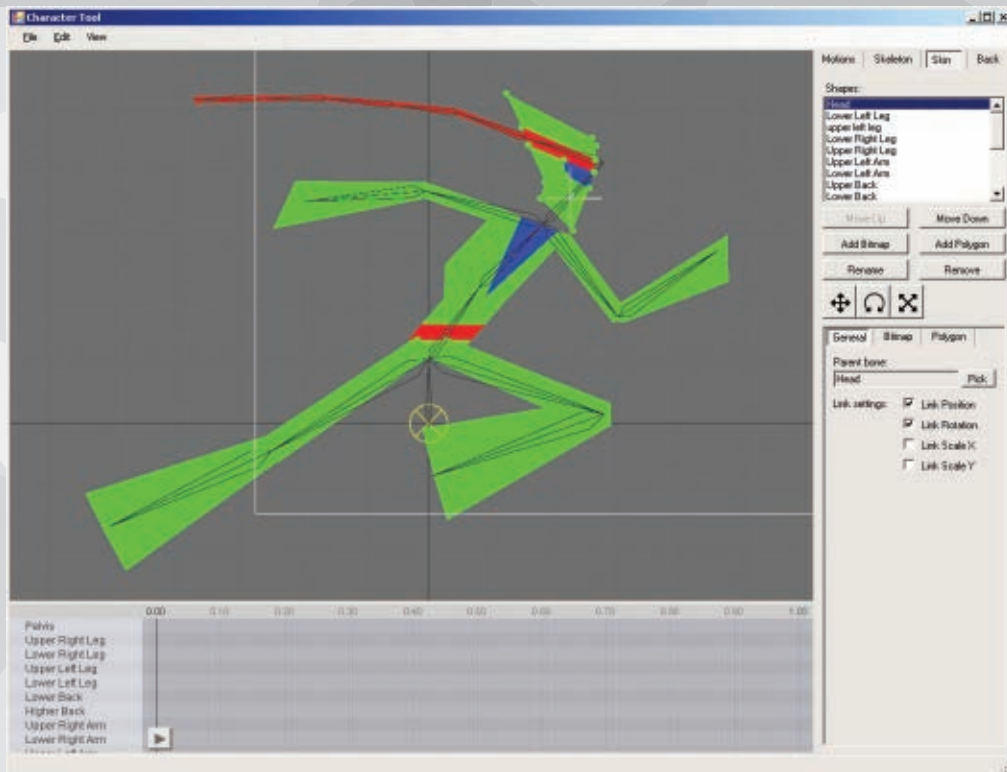
THE #1 VIDEO GAME UI SOLUTION

Integrated with the leading middleware engines
Artist-based **Flash®** UI development
Scalable image and vector graphics animation
High-quality, rich HTML text output
Efficient Asian font and IME text input
Available on PC and all major console platforms

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 Simutronics Corporation. Trinity logo is a trademark of Trinity GmbH. All rights reserved.





failure, and a lengthy launch delay for N+, but we worked to get through the new issues.

Predictably, lack of level sharing is the number two user complaint.

We pushed to keep developing content creation and sharing in because we knew it would benefit everyone: the game would be more popular because players would appreciate the creative possibilities, and it would add a lot of value to N+ as available levels would become essentially infinite. Perhaps, though, we should have just cut it when Microsoft initially started expressing reservations. It was a lot of work for everyone, and since there was a chance that it would be cut, it's possible this was a case of poor risk management on our part. As it stands, this feature is not likely to be allowed any time soon, if ever.

4 CERT HELL. After failing Certification the first time, we were stuck in a burnout-inducing pre-submission loop, a never-ending trickle of bugs that would go on and on until we finally resubmitted in December. There wasn't much we could do except buckle down and grind out the bugs; it was a morale-

destroying end to what had been an otherwise pretty smooth year. This string of required bug fixes unfortunately introduced a "save bug," which was the result of a last-minute Technical Content Requirement (TCR) workaround.

Because the frequency of saving/writing to leaderboards can't exceed a stipulated frequency as per XBLA TCRs, we

were forced to disable post-level saving, and instead save data only when the user returns to the main menu. This saving "bug" can cause loss of progress when users fail to return to the main menu before quitting their game or turning off the console—and losing progress in N+ can be particularly devastating given the difficulty of the game's later levels!

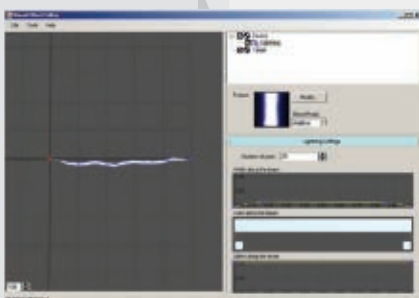
The "save bug" is probably the number three user complaint about N+.

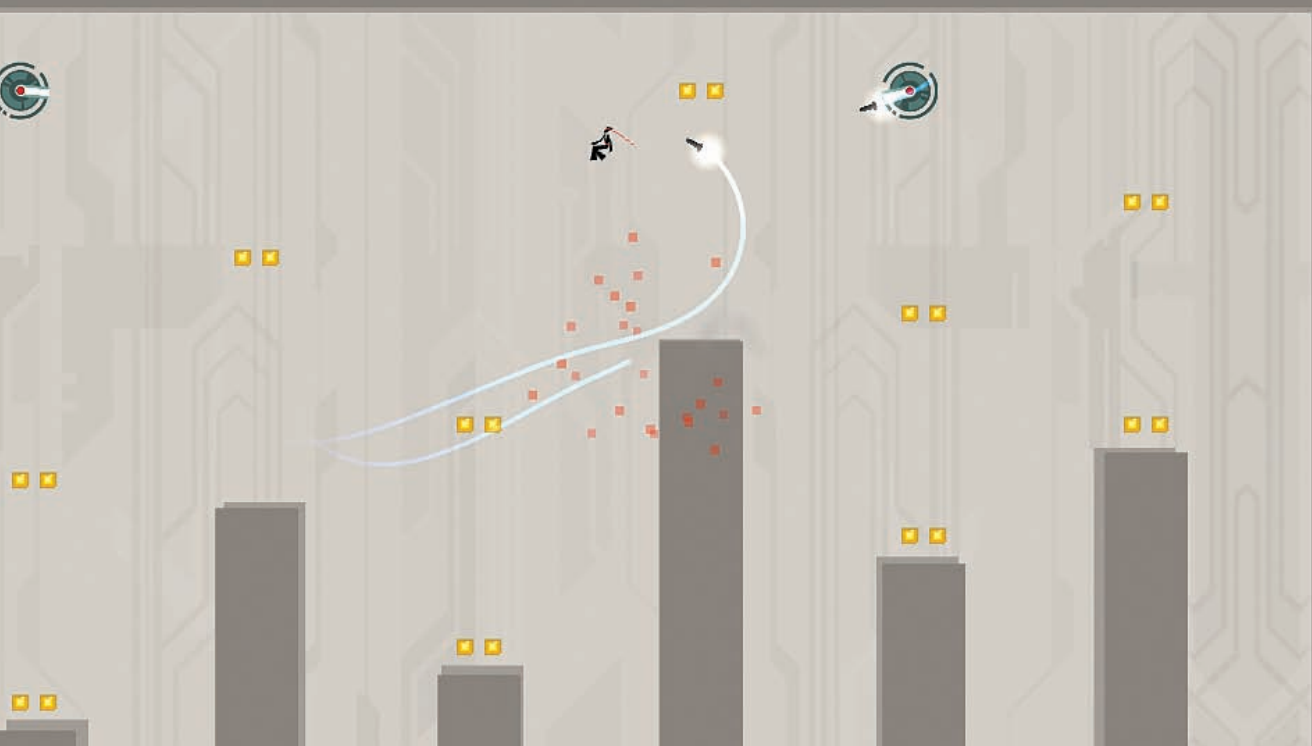
5 NO PLANNING FOR POST-LAUNCH DEVELOPMENT. From the beginning of the project, we knew we wanted to release additional level packs, but we didn't give this due thought when planning—in fact, we failed to schedule any post-release development whatsoever! After the delay from cert failure, we were all running on financial fumes by the launch of the game, and having no budget allocated to post-launch development only exacerbated the problem.

As the game reviews and user feedback poured in, we became very familiar with the aforementioned failings of N+, as well as several smaller things we had noticed on our own. We knew that a lot of these issues could be fixed with a title update (TU), which would also allow us to address newly discovered DLC-related bugs affecting achievements and leaderboards. We were also still optimistically waiting for Microsoft's verdict on content-sharing, hoping to enable it as part of the TU.

Unfortunately, we didn't start discussing DLC/TU development until after the game launched—we should have been discussing

The N+ visual effects editor.





it from the start of project! This resulted in some members of the team being unavailable, having started work on other projects once we reached the end of the N+ schedule (and more importantly, the N+ budget). As a result, the DLC was delayed, and the title update was indefinitely postponed.

This whole issue can be chalked up to an unfortunate oversight—we simply failed to appreciate how much development time and budget would have to be allocated beyond the release of the game. In hindsight this seems pretty obvious, but at the time we simply assumed that once we had delivered the final version of the game to Microsoft, our job was done. As a result of this mistaken assumption, we did a lot of unpaid level design, and there was ultimately not enough time or money for a Title Update. We are still, perhaps naively, hoping that this will someday change. N+ ANNIVERSARY EDITION, anyone?

BRINGING JUSTICE TO THE WORLD

Most of the problems we encountered can be attributed to lack of experience. This was the first XBLA game any of us had worked on, so no one had a clear idea of what to expect, and we all had to think on our feet. As a result, mistakes were made that could have been avoided. However, all of us

learned a lot, so this was a positive, if not totally problem-free, learning experience.

Networking and other glitches, plus the lack of content-sharing unfortunately prevented a perfect reception for N+; ratings in the 80s would probably have been 90+ if these few major problems had been addressed and corrected prior to launch.

Thankfully, most reviewers and fans have been sympathetic to the content-sharing issue at least, and overall the game has been very well-received, aside from the networking problems. We think that N+ was a success, and that Metanet, Slick, Microsoft, VMC and everyone involved have a lot to be proud of.

N+ has to date far exceeded our sales expectations, which is terrific, especially for an indie game few Xbox 360 owners had heard of, and one whose graphics didn't exactly grab the attention of the average user. We were thrilled to be able to bring N to new players and a new platform, and we think that the improvements made to the game really paid off. N+ XBLA is definitely N, plus. Worldwide appeal plus positive critical reception have justified our vision and really validated the idea that small teams can be successful in this industry. We'd go as far as to say small teams with ideas which are unique and different should be the lifeblood of XBLA! ❖



A SENSE OF FUN

ANYBODY COULD BE YOUR PLAYER 1

AT THE HEART OF MY INVOLVEMENT WITH VIDEO GAMES, I HAVE always been strongly linked to the "music game" genre. While this is often said of me, in reality at the end of the day I am first and foremost a musician looking for ways to expand my creative process, but I guess the end result is the same.

When PARAPPA THE RAPPER was in development at the beginning of the 90s, there were hardly any individuals attempting similar things. After the game's release in '96, similar titles started to become successful sales-wise, which led to greater variation to the extent that these days, "music games" are recognized as being in a genre of their own. But I want to say that we are very far from realizing our potential in this industry.

The past 10 years have seen a number of transitions that have taken the sector to its present condition. I'm delighted to see that lately music games have seen a great surge of interest in the West, thanks to titles like GUITAR HERO, ROCK BAND and SINGSTAR.

I couldn't verify this, but I was told that the GUITAR HERO

franchise accounted for 20 percent of all game sales in North America last year. Even if that figure isn't accurate, the sales are still astronomical. I can only say that this is an amazing thing. Ten years ago, when PARAPPA was big news, developers from Harmonix visited Japan and showed me some interactive music software that they were working on. I strongly advised them that rather than interactive music software, it had to be a game.

COPYCAT CURSE

The recent success of music games in the West has been based mostly on licensed music. It is necessary to move beyond this. The challenge is to discover the next paradigm in which music and games have a positive and complimentary relationship. I cannot overstate the importance of this. We are currently wasting our use of music.

Most games these days seem to use gorgeous orchestral soundtracks. While these large-scale soundtracks may generally

CONTINUED ON PG 31

MASAYA MATSUURA is president of NanaOn-Sha. He was an amateur musician until 1985, when he signed with Sony Music Entertainment as the leader of a group known as PSY'S. After nine original albums and three "best of" compilations, they disbanded in 1996. His game creations include PARAPPA THE RAPPER, UM JAMMER LAMMY, VIBRIBBON, and TAMAGOTCHI CONNECTION among others. He is currently working on MAJOR MINOR'S MAJESTIC MARCH for the Wii and an as-yet unannounced project. Email him at mmatsuura@gdmag.com.





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





Tools for Creation

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



Understanding your workflow
Enhancing your pipeline



Autodesk

Media and Entertainment
Authorized Reseller

LET US HELP YOU, CONTACT US

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



CONTINUED FROM PG 28

be lovely to listen to, if we really think about it, isn't it all a bit lacking in imagination? Thinking about it from a simplistic visual perspective, while films are basically just watched, games are interactive. The duties involved and objectives set are also different, so film and games cannot really be the same. And yet, music is basically the same in both. Despite this wonderful opportunity to advance music with this new medium, it seems that new and bold ideas are not encouraged in the current climate.

It's reasonable to think that unimaginative forms of expression will slowly die out as a new form of entertainment matures. Just looking at the past 10 years we can see that there have been drastic changes in the way we use music, as the media becomes digital, mobile, and more accessible. If we do not make great efforts to ensure progressive use of music in representative mediums, such as games, we could be faced with a steep decline.

The history of music games is still very young. So what is required to ensure the growth of this category? Maybe collaborating with cool musicians would be a good start, and I mean really collaborating, not just licensing music from famous artists. To all the cool musicians, please take note! I would like you all to dig deep into your own musical expression, and collaborate with other forms in order to extend yourselves.

GOTTA BELIEVE

Most musicians create music in their own style, even when they have been asked to make something specific. Many incapable music directors do not even think of starting musical production until after the taste of the game has been decided by the planning, design, and graphics teams. For me, someone who is engaged in this kind of occupation cannot be called a musician.

This applies for more than just music. If you want to create something of value, then you must continue to create for yourself as well. It's the same for designing games. If an artist only works on things that come to him through work, then all he is doing is production. Development is not coming from within. The scale and scope of the production, as well as the number of people working on the project should not put a damper on this creativity. The possibility of taking the ideas of many and fusing them into one creative expression is to me a wonderful thing. But it is very difficult to try to unite the varied opinions of a large group of people. As a result, one person's strong opinions and thoughts can go unrecognized or unreflective. Choices in games are generally decided through safety and without risk. Don't you think that this vagueness affects the flow and union of your team and business?

Lately, a lot of developers I know—especially programmers—have told me that they are making small scale games by themselves, independently. To develop something solely from one's own potential is, I believe, a thing of importance. The contents of a game are a combination of passion and energy. If these are becoming sub-ordinate to other factors, then it is game over.

I have come to feel that the music game genre is starting to outgrow itself and is now challenging us to expand its

possibilities. It's difficult to describe this, but music really is an art form of abstract elements, which makes me wonder—are games included in these elements?

Music is mysterious. It often has memories and information attached to it that can be very difficult to separate. To me as a musician, when creating music, I noticed I'm very clear in my own mind about its independent existence. While making



PARAPPA THE RAPPER

music is often an individual creative act, once music enters the physical world it becomes a shared property (see sidebar "Non-competitive Fun, AI, and You"). Its life is no longer under the musician's control. Here we can see a hint about why music is so hard to define.

A SENSE OF FUN: POSITIVE EMERGENCE

The premise of positive emergence is two entities attaining a mutual standpoint that is equal and fair. On the other hand, if the standpoints have different merits, or are unequal and strongly opposed, then emergence is restricted and instead we experience an unpleasant "negative emergence."

Naturally, it is not an easy matter to simply divide human emotions into positive and negative. In our adult world, there are a vast number of things that obstruct our ability to gain positive emergence with each other.

We often are faced with negative emergence when placed in circumstances that expose our inferiority or inequality, and this brings about stress. War is an extreme example of where these negative thoughts can lead.

I think it's regrettable that we are flooded with games that promote these negative emergences. It may be one of the most straightforward ways to design a game, but I don't think the future is bright for this industry if we continue to focus on games that motivate the player by using gameplay that employs



physical attacks as a way of establishing levels of superiority and inequality.

I've said this many times before, but in the future—be it a hundred years or a thousand—children will study 21st century history and the video game boom will be a part of that. But what if it's written like this:

"Back then, video games consisted mainly of people and monsters killing each other, similar to the gladiators of ancient Rome, and were a way to experience and enjoy violent behavior on the TV screen."

—Wikipedia, 3008 A.D.

This is not a history that I want to be associated with. I can understand that the energy of youth can evoke aggressive emotions. However, when I was an amateur musician back in the 80s, I played in a punk band where we wanted to scream our defiance to society, but I soon came to recognize the emptiness of this attitude. I only had to look at the wonderfully polished work of respected artists to realize this. For those of us who have been in this industry for a while, what can we do to stop our fresh young talents from being misled?

INTRODUCING ENGAGEMENT

Video games are a very simple way to enjoy virtual experience. All you need is a TV, a console, a controller, and the software. This is an easy system for everyone compared with other forms of entertainment. But like Hollywood, in order to keep the customers paying, the industry is using increasingly exaggerated content. Pressing buttons, moving sticks—these are small actions with grand effects. However, I think it is a slight error of judgment in our industry to believe that actions that in reality would carry great responsibility can be carried out in video games without thought for responsibility.

The Wii has come and put a cat amongst the pigeons of this unbalance. The harder you swing the remote, the faster the baseball bat moves. This more organic relation between imagination and reality is easily absorbed. At the same time we understand that game designs that, for example, require the player to shake the Wii controller strongly to rotate a TETRIS block, are unsuitable for input methods like this. The Wii requires a tighter connection between actual and virtual actions. But think! How can we improve on these

Non-competitive Fun, AI, and You

I AM INTERESTED IN THE INNATE PROCESS

by which music attains shared ownership. In particular, in the case of children we often see examples of primitive group musical expressions. To create these shared expressions, simple, inherently explainable rules are necessary. These are gleaned through interaction and positive emergence, and this led me to think that there might be a hint here for discovering a new form of music.

Thinking of an example, I saw a dance performed by a group of children from the Republic of Chad in Central Africa. This kind of play can of course be seen in playgrounds all over the world, and leads to an idea of what constitutes "fun." It is hard to explain the rules clearly, but for some reason, even without victory as a goal, players find some kind of enjoyment that is related to their conduct in games.

So why do human beings engage in these seemingly meaningless actions? And furthermore, why are they fun? Keeping these thoughts in mind, I would like to discuss Desmond Morris, a zoologist who published a book called *Man Watching* in 1991. He talked about a theory called "Postural Echo." Nowadays, I hear that there are a number of arguments against Morris' views, but I think that for the theme that I am interested in, the

observations and results have sufficient value, so I would like to paraphrase his definition of Postural Echo: When two friends meet and talk informally they usually adopt similar body postures. If they are particularly friendly and share like-minded attitudes to the subjects being discussed, then the positions in which they hold their bodies are liable to become even more alike, to the point where they virtually become carbon copies of each other. This is not a deliberate imitative process—the friends in question are automatically indulging in what has been called Postural Echo, and they do this unconsciously as part of a natural body display of companionship.

Morris asserted that it is an unconscious act in general communication between adults. But in the case of children playing, I wonder—aren't these postural echoes more or less necessary for positive development? Coming to understand rules that you previously didn't know, matching your behaviors in real time—from the beginning we have no way of knowing if we will be successful or not, or even if it will be fun. This kind of behavioral risk-taking, or "positive emergence," is necessary.

Yet for children, their behaviors are not as motivated by judging potential merits. Rather, as Morris suggested, it seems to me that it's

more a case of children adapting to the other humans in their surroundings, and behaving appropriately. If children feel like they have some of this shared emergence, then surely their playing should become more enjoyable. Sharing experiences with a friend, or multiple friends—this "game"-based form of fun is quite enjoyable. It is considered that this phenomenon remains in the sub-conscious even into adulthood.

I heard about a very interesting editing technique by a Japanese film director. He says, for example, when editing commercials, at specific intervals he'll insert a single black frame. At that moment, the audience will blink, without fail. Despite the "partner" in this case not even being human, for some reason humans unconsciously pick up these mimicking behaviors. This is the same if your partner is a computer game.

On the subject of AI, text-based communication like the Turing test soon comes to mind, but I really do think that it is an important mission to try and develop emergent forms of non-verbal based communication such as postural echoing in our relations with computers. I believe that "Kismet," the robotic emotional AI at MIT, is an example that is getting good results [see References¹].

CONCEPTOPOLIS

**WE
MAKE
ART**



CONCEPT ART | PACKAGE ART | TURN-AROUNDS | ENVIRONMENTS



WWW.CONCEPTOPOLIS.COM | 859.534.5500
CONCEPTOPOLIS: AN EXTENSION OF YOUR ART TEAM



kinds of obvious connections? That is the hint to make more advanced games.

Currently, we at NanaOn-Sha are working hard on a game that will hopefully make the best of the Wii platform.

Using the Wii remote for music games seems like a perfect match, but it has many difficulties. Music games need accurate data, quickly—and this is not a strong point of motion sensing controllers. Instead of focusing on traditional rhythm based gameplay, we decided to let the player control the tempo of a musical performance by waving the baton in a marching band. You can dynamically change the bpm of the performance—a completely new form of music game that would not be possible with digital controllers. At the same time we have other issues to consider—such as the player physically tiring out or getting bored of repetitive movements—so we've had to come up with many ideas to create an innovative and rewarding experience.

While we are currently working on a game for the Wii, I think that in the near future we will see games leaving our video displays. I'm anticipating an evolution in tangible experiences. You may think it a joke, but I do believe that we will see a day where STREET FIGHTER games see you actually fighting a robot bare fisted.

One day we will look back with embarrassment on this era when all of our virtual experiences were trapped behind a screen. This advance will have great implications for the role of games within society, and the wider possibilities of tangible experiences could make the word "game" insufficient to describe what we are doing. I would like to introduce some examples of things that have led me to think about this.

I created some music for a Japanese automobile company's pavilion at the 2005 Aichi Expo in Nagoya, which was about a futuristic one-man car. On this occasion, we were challenged with creating music that would convey the sense of interaction that a human might feel with a one-man vehicle as an extension of one's body image. It is said of existing cars that they are an extension of your body image. We particularly prioritize the ability to travel safely at high speeds on expressways. In this kind of dangerous environment, it is natural to want to arm oneself as robustly as possible. I think that the current nature of cars causes

us to think in this way. On the other hand, this vehicle is a one-man vehicle and doesn't make us feel that way. The engine is silent. Also this vehicle is a high-aspiration concept that envisages multiple units cooperating automatically on the highways, but at the high speeds of current cars. At least from a visual perspective this concept looks a little bit like a video game (see References?).

In this kind of environment, of course all kinds of sounds will be necessary for different functions, confirmations

and so on. Nowadays with computers, there are a large number of strange alert sounds and so on that ring away, but nobody seems to mind. After all, computers are just boxes that can't move, right? But when these sounds occur in something like this vehicle, carrying a human passenger and traveling at a high speed, the passenger is unable to stay calm and focused. In fact, if the driver can't quickly understand the meaning of the alert, it's downright dangerous. Therefore, we decided to

undertake this experiment with the ambition to expand the borders of music, and with a concept grounded on attempting to communicate these functions of music.

ANYBODY COULD BE YOUR PLAYER 1

The neurologist Alan Snyder suggests of children with Savant Syndrome that as their knowledge of languages piles up, they lose their natural genius ability to draw incredibly precise pictures.

In other words, as the left brain develops we lose our ability to see detail. Extending this to everyday reality, in my head, when I am informed simply of a general idea, I become less inquisitive about it in my heart.

Games are important media that help us actively experience these new ideas. Outside of games, in some other forms of representative media, we require some prior knowledge or expectations in order to understand and feel the contents.

This is not completely necessary in the case of games. You get an interactive reply to your input in an instant and good games can teach you about themselves. But games also have the potential to reach out and affect deep emotions. This is why games are fun, and also important.

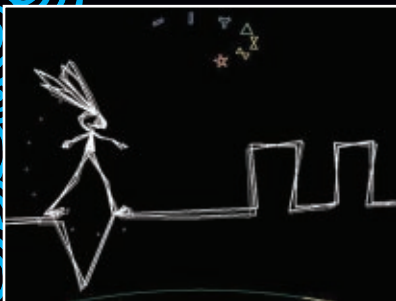
I think in today's world we are suffering from a lack of mutual understanding of each other's differences. That games are fostering this rejection of mutual understanding is very saddening to me, particularly as this is the industry that we work in. As I grew up I was heavily influenced by Western culture, and this passion continues to this day. For this reason, I would like to see Western developers make a bigger effort to develop products that will also appeal to the other markets. Like me, I would like future generations to experience the joys of respecting the culture of different countries.

THE CHOICE IS THEIRS

I mentioned earlier the idea of positive emergence, and I think that the biggest stimulus for this is the sense of hearing. Even if you don't want to hear it, sound makes its way into our ears, and sometimes this is of course an inconvenience. This is why speech became a way to read and consider your companions and environments. With games, the player cannot enjoy the product without buying a package or downloading something. Unlike listening, they have a choice. If we are saying things that they don't want to listen to, they can ignore us. So we need to find a language that appeals to everyone, and if we do, then it doesn't matter who your "Player 1" is—it could be anyone. ❖



TAMAGOTCHI CONNECTION: CORNER SHOP 3



VIB RIBBON

REFERENCES

1: www.ai.mit.edu/projects/humanoid-robotics-group/kismet/kismet.html

2: www.toyota.co.jp/jp/news/04/Dec/nt04_1201d.html

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

Create the Game

Gaming Degree Programs for the Next Generation

Game Art

Associate's Degree Program

Game Development

Bachelor's Degree Program

Game Design

Master's Degree Program



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

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

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

fullsail.edu

Online Programs Available



FULL SAIL
UNIVERSITY

GAME CAREERGUIDE.COM



DOWNLOAD
AND PLAY
THE LATEST
STUDENT
GAMES!

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

- News and Features for students and educators
- Getting Started section – an invaluable how-to guide
- Faculty & Student Research – Salary Survey
- Directory of Game Schools
- Message Boards



Download your FREE digital edition of the 2008 **gamedeveloper** Game Career Guide at:

www.gamecareerguide.com

NextEngine 3D Scanner

By Tom Carroll

DON'T YOU JUST LOVE GADGETS?

Don't you crave putting a Turnsmart Mirror System on your 1981 Subaru Brat, thinking that somehow that it's going to give passing motorists "mirror envy"? Doesn't the Walkstation you saw in a recent catalog give you pause—the one that lets you use your computer while walking on an attached treadmill? And don't you yearn for the USB Missile Launcher that fires three foam missiles

on command approximately seven feet from your desk—compatible with Windows XP and 2000?

Yeah, those gizmos say a lot about you, you gadget lover.

But now for something completely different, and much more useful than anything I mentioned above (well, except maybe the

Walkstation)—the NextEngine Desktop 3D scanner.

The NextEngine scanner can scan nearly anything you can imagine into your computer, making a polygonal replica of it in a reasonably short amount of time and with a surprisingly brief learning curve. Without getting into too

much esoteric detail, the NextEngine scanner operates pretty much like NASA scanning a distant planet. Everyone has seen those cool images that come back from Mars, containing the minutest detail. Because data is shipped back from space in packets, only portions of the planet's surface can be seen at any given time. But NASA developed powerful algorithms that can sense where the edges of one section fit another and join them together without error.

NextEngine's scanner does something similar, though its operation relies on some user interaction to obtain the best results. The company provides a number of great test objects, so users don't have to fiddle around with some overly complicated action figure (the gnarly Sentinel robot from *The Matrix* comes to mind) and give themselves a nervous breakdown within an hour of plugging the thing in.

SCANNING

There is no limit to the size of object that can be scanned. While it's convenient to use the turntable that comes with the system (it automatically turns the object the precise amount the user specifies), any object can simply be set down in front of the scanner, scanned, and manually turned to a new angle and scanned again. This makes it possible to scan anything from a delicate shell to

that massive bust of Beethoven that's sitting on your baby grand piano. What makes this possible is that scan sections are joined together inside the computer using NextEngine's ScanStudio CORE software tools. However, I used the squishy football that the scanner comes with to begin my scanning odyssey and it worked very well. In a nutshell, I took the football and positioned it on a simple turntable that is attached by a data cable to the scanner. The football isn't too tiny, so you can scan it using Macro settings. Objects larger than foam footballs should be scanned farther away from the laser source and thus have their own wide angle settings.

The turntable is free to rotate 360 degrees on command and it is up to the user to determine how many segments they separate the 360 degrees into. The smaller the slices, the better the final scan will be, but there's a trade-off; smaller slices draw out the scan times.

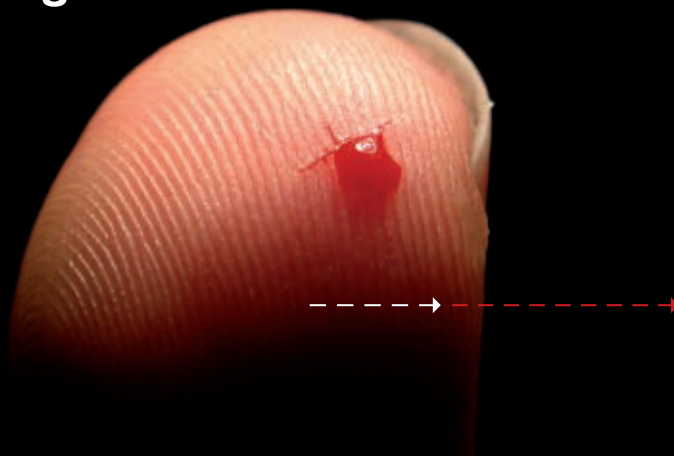
For my football, I used six 60 degree slices and the scanning took approximately 15 minutes to finish. NextEngine says to estimate based on two minutes a slice.

If you watch the screen during capture, you see that the first step in scanning a slice is color capture. Not only does the NextEngine scanner capture the three dimensional aspects of your object, but it also pulls in the diffuse colors that



The NextEngine 3D Scanner.

A minor issue in the beginning.



make up its surface. A finished scan will therefore have the surface colors applied to the 3D model.

The results of my first scan were quite satisfactory. The scanner uses multiple lasers during the scanning process so that it can cross validate the information it is gathering, eliminating the jaggies, pops, and/or spikes that populated earlier technology. In macro mode, it obtains .005 inch accuracy, and in wide mode it is accurate to .015 inch.

A STITCH IN TIME

Unfortunately for me I had forgotten to draw white lines on the football with a marker provided in the scanner kit. This is important because when you finish scanning your slices of your object you have to place three colored dots onto the same position on two of the slices. This gives



FIGURE 1 The author's initial scan of his character sculpture looked like this minus pedestal polygons. Many more hours of "play" would be required before he would call it tight.

the software enough correlation to fit the slices together into one solid object. Without lines, or some other identifying marks, I wasn't easily able to place my dots on the surface and the joining together of my initial scan turned out less than perfect.

Within minutes, however, I had used the pen to draw four lines lengthwise on the football and to place a unique letter inside each segment. With these

helpful markers, I was able to stitch my segments together like a seasoned pro.

This initial scan left me with a problem. The top and bottom of the football were mashed against the turntable and weren't scanned properly. With NextEngine this isn't a problem at all. ScanStudio CORE editing tools allowed me to scrape off the offending polygonal mess at the football's poles. Using these tools is similar to using a packages like iPhoto to edit photographs. Why? In a similar way, the software makes a lot of decisions on the user's behalf to smooth out the processing of all that data involved in joining scan lines together. And like with iPhoto, if you don't like the results the software initially gives you, you simply undo it and go it again with different user driven parameters.

Once I had refined my initial football scan (cutting off the two gnarly ends), I rotated the football on the scanner and used it to scan only the two ends. As before, by aligning the two new scans with points on the existing football's shell, they snapped into place with nary a bother.

NextEngine supplies a software application called ScanStudio CORE with the scanner that enables the user to refine the mesh he/she obtained. With ScanStudio CORE you can smooth out entire scans or portions of them, fill holes that may have resulted from the joining process (hey, it happens), and do something called "smart variable density decimation" that ensures that any scan is without holes, seams, or other nagging glitches. The decimator allows the user to specify a resolution tolerance for the end result, which intelligently combines

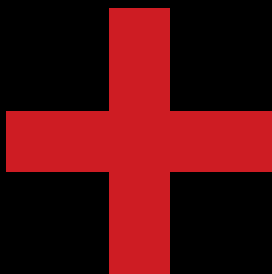
triangles to shrink the dataset size while still preserving detail.

I used ScanStudio CORE to fill in several holes that remained on my football scan once I had joined the slices together. The process was extremely easy.

The final step is outputting the scan so it can be used within popular 3D modeling/rendering programs like Maya, Max, XSI, or Modo. NextEngine supports STL, OBJ, U3D, VRML, and XYZ points. More ambitious modelers can run their models (sample footballs in my case) into Mudbox or Zbrush—all the better to turn them into slump-shouldered aliens with six-pack abs (and football-shaped heads). As with any technical process, users become more proficient and faster with the NextEngine 3D scanner the more they use it. Such companies as Activision, LucasArts, and Sega of America have employed the NextEngine equipment and any company that wants to include source information from the widest variety of sources, including real world assets and artistic sculptures, should consider looking into it.

For me, the next challenge was to sculpt my own creation out of the block of FIMO polymer clay provided in the scanner's accessory kit. See Figure 1 for the results.

The reason? Watertight 3D models built from NextEngine scan data can be printed out on a variety of 3D printers from Dimension, ZCorp, 3D Systems, and others. To be printable, a 3D model file must comprise a fully-healed watertight mesh. Essentially, this means that the mesh of triangles describing the surface must not overlap, and that all the surfaces



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.

that make up the solid model must be closed without perforations in them.

While I have no personal experience with this, I am told these printers create actual physical models of your model or object in plastic or some similar materials, and are very useful in testing parts prior to manufacture, or in making duplicates of scanned originals.

IT SLICES, IT DICES

So far, this review makes it sound like the NextEngine 3D scanner is as easy to use as a potato peeler, but that's not necessarily the case. If the object you want to scan is shiny or completely black (or, eek!! ... both), the scanner's lasers will not return an accurate scan. The remedy: spray paint the object with a dull grey primer, or if it's an extremely expensive Ming vase (cost: \$78 million) you can apply some talcum powder to it that will dull down the surface without rendering it worthless. As with

the football demo, it is also very helpful to have some marks on the surface to help calibrate the scan's final assembly. While the white marker or a Sharpie is quite adequate, pieces of tape can be applied to precious objects and removed after scanning.

My experience with the NextEngine HD scanner almost proved fatal. Because I was able to watch the scanning process—my eyes moving from the screen to the model to the lasers and back to the screen—I quickly discovered that the scanning process was at least as entertaining as watching most network television shows. It was with great effort that I packed the scanner back into its foam packing. The NextEngine scanner is fun to use, easy to use, and somewhat addictive once you begin to see the high quality results you can obtain.

Sadly, my newfound love for the NextEngine Desktop 3D Scanner came with personal loss: I have to somehow

find the courage to call the makers of the Walkstation to cancel my order. My wife says I can't have both ...

Pass the hankies, please.

NextEngine notes that an upcoming release of the 64-bit ScanStudio CORE 1.8 software that is tentatively scheduled for September will result in a jump from 2 million to 100 million-point capacity.

The upgrade will feature Multi-Color Specularity, a 50 percent increase in scan capture speed with a new gearbox for HD models, and a resolution increase from 40,000 to 160,000 dpi.

Anyone who purchased NextEngine scanners after December 2007 will be able to upgrade to the new 64 bit capability without requiring new hardware.

TOM CARROLL is a video game artist currently with a prominent game studio. He is a contributor to myIPD.com, an intellectual property portal. Email him at tcarroll@gdmag.com.

NEXTENGINE HD 3D SCANNER

★★★★

STATS

NextEngine, Inc.
401 Wilshire Blvd., Ninth Floor
Santa Monica, California 90401
Fax 310 883 1860
www.nextengine.com

REQUIREMENTS

Minimum: Windows XP, 2 GHz PC, 2 GB RAM, 128 MB Graphics, USB 2.0

Recommended: Windows x64, 3 GHz PC, 4 GB RAM, 256 MB Graphics, Powered USB 2.0 Hub

PRICE

\$2,495

PROS:

1. Relatively cheap to own, especially considering how much it can do.
2. Extremely portable.
3. Works quickly.

CONS:

1. Preparation of different object surfaces is the highest learning curve, and failure to learn can yield spotty results (which some may blame on the machine instead of themselves).
2. The relatively inexpensive price is still somewhat expensive for individuals looking to scan physical objects or sculptors seeking an entree into 3D.
3. While relatively intuitive, you still have to be pretty darned tech savvy to obtain the best possible results.

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

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

CONCEPT EXECUTION

DG
RC

NC STATE UNIVERSITY

Digital Games
Research Center

h t t p : / / d g r c . n c s u . e d u



WORKING TO CREATE THE SCIENCE OF GAMES

The **NCSU Digital Games Research Center** is a multidisciplinary center focusing on the creation of new games technologies, game play and game design.

With over \$3 million in research funding, the DGRC focuses on

- Serious games for education, training and other contexts
- Core technologies, including procedural content generation, graphics and AI
- Games as social artifacts, including mobile games and games to teach

PREPARING STUDENTS FOR SERIOUS CAREERS

NC State is the leading source for college hires in the growing North Carolina games industry and regularly places graduates nationally

Special tracks on game development/design in both Computer Science and Industrial Design departments prepare our students for the unique needs of the games industry

Coursework exposes students to real-world tools and technologies used to build significant projects in multi-disciplinary development teams





NOEL LLOPIS

» THE INNER PRODUCT

MANAGING DATA RELATIONSHIPS

FROM A 10,000-FOOT VIEW, ALL VIDEO

games are just a sequence of bytes. Those bytes can be divided into code and data. Code is executed by the hardware and it performs operations on the data. This code is generated by the compiler and linker from the source code in our favorite computer language. Data is just about everything else.

As programmers, we're obsessed with code: beautiful algorithms, clean logic, and efficient execution. We spend most of our time thinking about it and make most decisions based on a code-centric view of the game.

Modern hardware architectures have turned things around. A data-centric approach can make much better use of hardware resources, and can produce code that is much simpler to implement, easier to test, and easier to understand. In the next few months, we'll be looking at different aspects of game data and how everything affects the game. This month we start by looking at how to manage data relationships.

DATA RELATIONSHIPS

Data is everything that is not code: meshes and textures, animations and skeletons, game entities and pathfinding networks, sounds and text, cut scene descriptions, and dialog trees. Our lives

would be made simpler if data simply lived in memory, each bit totally isolated from the rest, but that's not the case. In a game, just about all the data is intertwined in some way. A model refers to the meshes it contains, a character needs to know about its skeleton and its animations, and a special effect points to textures and sounds.

How are those relationships between different parts of data described? There are many approaches we can use, each with its own set of advantages and drawbacks. There isn't a one-size-fits-all solution. What's important is choosing the right tool for the job.

POINTING THE WAY

In C++, regular pointers (as opposed to "smart pointers," which we'll discuss later on) are the easiest and most straightforward way to refer to other data. Following a pointer is a very fast operation, and pointers are strongly typed, so it's always clear what type of data they're pointing to.

However, they have their share of shortcomings. The biggest drawback is that a pointer is just the memory address where the data happens to be located. We often have no control over that location, so pointer values usually change from run to run. This means if we attempt to save a game checkpoint which contains a pointer to other parts of the data, the pointer value will be incorrect when we restore it.

Pointers represent a many-to-one relationship. You can only follow a pointer one way, and it is possible to have many pointers pointing to the same piece of data (for example, many models pointing to the same texture). All of this means that it is not easy to relocate a piece of data that is referred to by pointers. Unless we do some extra bookkeeping, we have no way of knowing what pointers

are pointing to the data we want to relocate. And if we move or delete that data, all those pointers won't just be invalid, they'll be *dangling pointers*. They will point to a place in memory that contains something else, but the program will still think it has the original data in it, causing horrible bugs that are no fun to debug.

One last drawback of pointers is that even though they're easy to use, somewhere, somehow, they need to be set. Because the actual memory location addresses change from run to run, they can't be computed offline as part of the data build. So we need to have some extra step in the runtime to set the pointers after loading the data so the code can use them. This is usually done either by explicit creation and linking of objects at runtime, by using other methods of identifying data, such as resource UIDs created from hashes, or through pointer fixup tables converting data offsets into real memory addresses. All of it adds some work and complexity to using pointers.

Given those characteristics, pointers are a good fit to model relationships to data that is never deleted or relocated, from data that does not need to be serialized. For example, a character loaded from disk can safely contain pointers to its meshes, skeletons, and animations if we know we're never going to be moving them around.

INDEXING

One way to get around the limitation of not being able to save and restore pointer values is to use offsets into a block of data. The problem with plain offsets is that the memory location pointed to by the offset then needs to be cast to the correct data type, which is cumbersome and prone to error.

A more common approach is to use indices into an array of data. Indices, in

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 *MCHASSAULT* series. Email him at nllopis@gdmag.com.

addition to being safe to save and restore, have the same advantage as pointers in that they're very fast, with no extra indirections or possible cache misses.

Unfortunately, they still suffer from the same problem as pointers of being strictly a many-to-one relationship and making it difficult to relocate or delete the data pointed to by the index. Additionally, arrays can only be used to store data of the same type (or different types but of the same size with some extra trickery on our part), which might be too restrictive for some uses.

A good use of indices into an array would be particle system descriptions. The game can create instances of particle systems by referring to their description by index into that array. On the other hand, the particle system instances themselves would not be a good candidate to refer to with indices because their lifetimes vary considerably and they will be constantly created and destroyed.

It's tempting to try and extend this approach to holding pointers in the array instead of the actual data values. That way, we would be able to deal with different types of data. Unfortunately, storing pointers means that we have

to go through an extra indirection to reach our data, which incurs a small performance hit. Although this performance hit is something that we're going to have to live with for any system that allows us to relocate data, the important thing is to keep the performance hit as small as possible.

An even bigger problem is that, if the data is truly heterogeneous, we still need to cast it to the correct type before we use it. Unless all data referred to by the pointers inherits from a common base class that we can use to query for its derived type, we have no easy way to find out what type the data really is.

On the positive side, now that we've added an indirection (index to pointer, pointer to data), we could relocate the data, update the pointer in the array, and all the indices would still be valid. We could even delete the data and null the pointer out to indicate it is gone. Unfortunately, what we can't do is reuse a slot in the array since we don't know if there's any data out there using that particular index still referring to the old data.

Because of these drawbacks, indices into an array of pointers is usually not an effective way to keep references to data. It's usually better to stick with indices into an array of data, or extend the idea a bit further into a handle system, which is much safer and more versatile.

HANDLE-ING THE PROBLEM

Handles are small units of data (32 bits typically) that uniquely identify some other part of data. Unlike pointers, however, handles can be safely serialized and remain valid after they're restored. They also have the advantages of being updatable to refer to data that has been relocated or deleted, and being possible implement with minimal performance overhead.

The handle is used as a key into a handle manager, which associates handles with their data. The simplest possible implementation of a handle manager is a list of handle-pointer pairs

and every lookup simply traverses the list looking for the handle. This would work but it's clearly very inefficient. Even sorting the handles and doing a binary search is slow and we can do much better than that.

An efficient implementation of a handle manager is available online at www.gdmag.com/resources/code.htm. The handle manager is implemented as an array of pointers, and handles are indices into that array. However, to get around the drawbacks of plain indices, handles are enhanced in a couple of ways.

In order to make handles more useful than pointers, we're going to use up different bits for different purposes (see Listing 1). We have a full 32 bits to play with, so this is how we're going to carve them out (see Figure 1):

The index field. These bits will make up the actual index into the handle manager, so going from a handle to the pointer is a very fast operation. We should make this field as large as we need to, depending on how many handles we plan on having active at once. 14 bits give us over 16,000 handles, which seems plenty for most applications. But if you really need more, you can always use up a couple more bits and get up to 65,000 handles.

The counter field. This is the key to making this type of handle implementation work. We want to make sure we can delete handles and reuse their indices when we need to. But if some part of the game is holding on to a handle that gets deleted—and eventually that slot gets reused with a new handle—how can we detect that the old handle is invalid? The counter field is the answer. This field contains a number that goes up every time the index slot is reused. Whenever the handle manager tries to convert a handle into a pointer, it first checks that the counter field matches with the stored entry. Otherwise, it knows the handle is expired and returns null.

LISTING.01 the handle structure

```
struct Handle
{
    Handle() : m_index(0), m_counter(0), m_type(0)
    {}

    Handle(uint32 index, uint32 counter, uint32 type)
        : m_index(index), m_counter(counter), m_type(type)
    {}

    inline operator uint32() const;

    uint32 m_index : 12;
    uint32 m_counter : 15;
    uint32 m_type : 5;
};

Handle::operator uint32() const
{
    return m_type << 27 | m_counter << 12 | m_index;
}
```

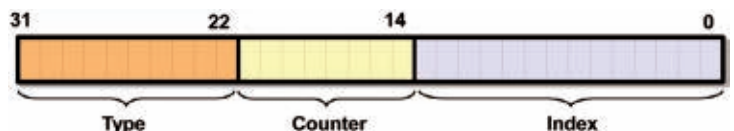


FIGURE 1 32 bit handle.

THE INNER PRODUCT

The type field. This field indicates what type of data the pointer is pointing to. There are usually not that many different data types in the same handle manager, so 6–8 bits are usually enough. If you're storing homogeneous data, or all your data inherits from a common base class, then you might not need a type field at all.

CRANK IT

The workings of the handle manager itself are pretty simple. It contains an array of `HandleEntry` types (see Listing 2). Each `HandleEntry` has a pointer to the data and a few other bookkeeping fields: freelist indices for efficient addition to the array, the counter field corresponding to each entry, and some flags indicating whether an entry is in use or it's the end of the freelist.

Accessing data from a handle is just a matter of getting the index from the handle, verifying that the counters in the handle and the handle manager entry are the same, and accessing the pointer—just one level of indirection and very fast performance.

We can also easily relocate or invalidate existing handles just by updating the entry in the handle manager to point to a new location or to flag it as removed.

Handles are the perfect reference to data that can change locations or even be removed, from data that needs to be serialized. Game entities are usually very dynamic, and are created and destroyed frequently (such as with enemies spawning and being destroyed, or projectiles). So any references to game entities would be a good fit for handles, especially if this reference is held from another game entity and its state needs to be saved and restored. Examples of these types of relationships are the object a player is currently holding, or the target an enemy AI has locked onto.

LISTING.02 HandleEntry structure

```
struct HandleEntry
{
    HandleEntry();
    explicit HandleEntry(uint32 nextFreeIndex);

    uint32 m_nextFreeIndex : 12;
    uint32 m_counter : 15;
    uint32 m_active : 1;
    uint32 m_endOfList : 1;
    void* m_entry;
};
```

GETTING SMARTER?

The term smart pointers encompasses many different classes that give pointer-like syntax to reference data, but offer some extra features on top of “raw” pointers.

A common type of smart pointer deals with object lifetime. Smart pointers keep track of how many references there are to a particular piece of data, and free it when nobody is using it. For the runtime of games, I prefer to have very explicit object lifetime management, so I'm not a big fan of this kind of pointers. They can be of great help in development for tools written in C++ though.

Another kind of smart pointers inserts an indirection between the data holding the pointer and the data being pointed. This allows data to be relocated, like we could do with handles. However, implementations of these pointers are often non-serializable, so they can be quite limiting.

If you consider using smart pointers from some of the popular libraries (STL, Boost) in your game, you should be very careful about the impact they can have on your build times. Including a single header file from one of those libraries will often pull in numerous other header files. Additionally, smart pointers are often templated, so the compiler will do some extra work generating code for each data type you instantiated templates on. All in all, templated smart pointers can have a significant impact in build times unless they are managed very carefully.

It's possible to implement a smart pointer that wraps handles, provides a syntax like a regular pointer, and it still consists of a handle underneath, which can be serialized without any problem. But is the extra complexity of that layer worth the syntax benefits it provides? It will depend on your team and what you're used to, but it's always an option if the team is more comfortable dealing with pointers instead of handles.

DESTINATION DATA

There are many different approaches to expressing data relationships. It's important to remember that different data types are better suited to some approaches than others. Pick the right method for your data and make sure it's clear which one you're using.

In the next few months, we'll continue talking about data, and maybe even convince you that putting some love into your data can pay off big time with your code and the game as a whole. ❖



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.

Austin Game Developers[®] Conference



September 15-17, 2008

Austin Convention Center | Austin, TX



KEYNOTE SPEAKERS



“At Their Service: Making a Difference by Putting Players First”

Lane Merrifield

Co-founder and General Manager | Club Penguin



“Computer Entertainment 35 Years From Today”

Bruce Sterling

American Author



“Is That It? Next Gen Audio”

Jason Page

Audio Manager | Sony Computer Entertainment Europe



Connect, Engage and Interact with over 5,000 designers, programmers, publishers, and game professionals currently in the myGDC community.

Austin GDC Features Sessions Covering:

Audio • Online Games • Writing for Games

Business & Marketing • Design • Social Networking & Community • Technology & Services



ONLINE REGISTRATION CLOSING 9/11. REGISTER NOW AT AUSTINGDC.NET AND SECURE YOUR LAST CHANCE FOR SAVINGS.



STEVE THEODORE

PIXEL PUSHER

PUNCH THE CLOCK

You're shipping soon. Get back to work.

I RECENTLY FOUND MYSELF TRYING TO explain some of the details of how a game gets wrapped up to my Dad, who comes from a universe in which “ship schedules” means “when do the cruise ships depart?” and “gold masters” are a cabal of secretive Bavarian Illuminati manipulating world commodities markets. He commented that release time must be exciting and exhilarating, and I started to agree—but I cut myself short. Sure, it is great to know your baby is going off to conquer the world; and nothing beats the secret, pornographic thrill of lurking around the Best Buy eavesdropping on the folks browsing your boxes. But the actual process of “releasing”—of letting go, of surrendering all hope that what’s actually on that disc will be better than it is right now? That’s anything but exhilarating. It’s downright sad.

Picasso famously said:

“To finish it means to be through with it, to kill it, to rid it of its soul, to give it its final blow, the coup de grace for the painter as well as for the picture.”

But Picasso, was, well, Picasso—by the time people were asking him for quotes, he didn’t have to worry about a publisher threatening to withhold his next milestone payments if he couldn’t get that fourth set of eyes on Dora Maar in time for E3. Picasso was one of those lucky souls who can say “when it’s done” with no irony in their voices. The rest of us are stuck with finishing things on other people’s timelines. The bargain bin of any game

shop is full of games that would have been a lot better with a few more months to spare.

The essence of any medium is really what you can’t do, rather than what you can. Like a haiku, an opera, or a still life, much of what’s beautiful in games comes from seeing how we cope with the limitations of the form. It’s not about the amazing visions in our heads—it’s about the bits we actually ship out for other people to play with. The final work is always less than what we wanted it to be; it’s more pixellated, under-animated, weirdly lit, beset by bugs—but it’s what people see and play and what makes our worlds come alive for them. You yourself remember the Mario or Solid Snake or Big Daddy you got, not the one the concept artist imagined. We are what we ship, so we’d better learn to like it.

TICK, TOCK

But learning to like it ain’t easy. The Grim Shipper is always waiting in the wings. This means that the most critical decision any game artist makes is how to invest that small, never-to-be expanded piggybank of minutes. Time is a finite resource. You can solve some problems by throwing money or computing horsepower or memory at them—but nothing will add more minutes to the day. The number of decisions you get to make in the course of your day, your week, and your project is pretty much fixed. You might be doing something as important and “artistic” as concepting a character, or as petty and bureaucratic as renaming files—either way, the clock is ticking.

The first rule of beating the clock, therefore, is remembering that every choice has costs. You could, for example, spend half a day straightening out UVs you’ve already done once, and repacking your textures to grab a bit more resolution. It would make better art. But is it worth missing a half-day of modeling touch-up on the next character? Don’t be fooled into thinking this is a producer question—it’s

not about scheduling or hitting milestones. It’s an artistic choice! If this model or that texture gets some extra loving, do you know what won’t get done because of it? Do you consistently, routinely do what the work really needs to succeed as art instead of getting bogged down in trivia? That’s almost the dictionary definition of a good games artist.

Teachers in every art school from caves of Lascaux on down have been trying to beat the same idea into our heads. Work on the big picture first, not the details. The teachers in our figure drawing classes, Animation Mentor crits, and studios keep repeating the same things over and over: Don’t work on the eyelashes when you haven’t roughed in the proportions; don’t worry about the finger positions when you’re blocking in your scene; don’t start building individual rivets when you’re still designing the armor ... you know the drill.

And you probably ignore it a lot of the time.

Working from the big gestures to the little details is easy to preach, but hard to practice. People get into the art business because they are hypnotically drawn to the little details. Sure, you may spend most of your polys, frames, or texels on the broad strokes—but deep down you know that the magic lies in the mysterious alchemy of some small gracenotes. Sober good practice is certainly the Right Thing To Do—but it’s always a struggle and sometimes you just have to listen to your muse instead of *Game Developer*.

LIKE SAND THROUGH THE HOURGLASS

Maybe the spectre of the Ticking Clock principle won’t turn you into a disciplined pre-planner, but it can still help you wring more value from the less glamorous parts of your day. The logic of time’s-a-tickin’ is something artists of all stripes should remember when it’s time to discuss nuts and bolts with other

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.

departments. Those inter-departmental games of hot potato often devolve into an elaborate game of “whose time gets wasted,” and artists are too often on the receiving end in that game. We’ve got less practice in wrangling and, well, lower hourly rates.

But you shouldn’t get rolled over just because you’re an artist. Studios often commit artists to clerical stuff (“Oh, just copy all these file names into this excel sheet”) or convoluted processes (“If you want that transparent you just put a semicolon in the name, and parent a dummy object named ‘@transparent’ to it, and export it with the `-t` option”) to save on developer time. Sometimes, this is a reasonable cost-benefit decision, but all too often it’s just squandering time that should be used for making art. Maybe you’re only wasting 10 minutes a day on nonsense? Well, that’s an extra week per year.

This means, among other things, that you have to be an active advocate for your own tools. The engineers can’t wave their fairy computer-science wands and magically eliminate all the hard work of creativity, iteration, and experimentation. We’re artists, the fairy godmother stuff is our business—but what you *do* have to insist on is tools that are flexible, iterative, and above all tools that stay out of your way. Character, emotion, and intensity are your problem—file naming, streaming caches, and shader-fragment management shouldn’t be. It’s totally fine that some engine requirement means that all the animations for a single character need to be loaded from a single file. Great, cool, gotcha. But actually making the animators work in that one file? That’s madness. You want all the textures for this level to live in a giant 4096 combination plate? Good for you! But don’t expect 10 texture artists to share a single photoshop file.

Nobody in any department ever has enough time. Programmers don’t make crazy systems because they’re evil (well, only rarely). You just need to remind them of the real costs of crazy work flows and using valuable creative time for housekeeping. Be patient, sympathetic, and reasonable; but be stubborn, too. This is too important to skip over.

TIME KEEPS ON SLIPPING

Still, before we get all huffy and start mumbling under our breath about other departments, we need to think hard about how we sabotage ourselves. A lot of the busywork that plagues us is of our own devising. Artists love to be in control over every detail of what we make—it’s the seductive power of those little details again. But if we help design systems that need too much handholding, we can end up clogging up the works just as badly as Mr. “just put an asterisk in lines 72, 79, and 103 of `importSettings.txt`” from engineering.

The artistic personality thrives on micromanagement. For example, you’d probably love to have the ability to control the lightmap resolution on any face in your environment. “Ah!” you think, “it’ll be so handy to make sure all the shadow edges are nice and tight!” It’s easy to imagine yourself deftly shuffling resolution to where it’s needed most, stretching those inflexible GPU budgets and killing off some of those blocky texels that drive you nuts. It doesn’t cost the engine anything, and makes you happy—what’s not to like?

But stop for a moment and think how long it’s going to take to actually make all those individual decisions about lightmap texels. How long can you afford to ponder the lightmapping of every triangle in your level—a second? Half a second? Well, for a moderately sized level of 150,000 polys, that’s one solid week of doing nothing but twiddling lightmaps just to hit those faces once. Well, OK, so you’ll probably flood fill them all and then just mess with the problem ones? Sure ... until you have to rebuild that big chunk of terrain to accommodate a gameplay change. Oops, back to the lightmap-resolution tool again. And again. And then somebody comes in and decides your lightmap budget needs to drop by 20 percent—who, exactly, is going to be changing all those fives into fours?

At this point, maybe you’re willing to consider something a little less ambitious? Maybe you could drop a box in the level that tells the engine to up the lightmap res inside the box? Or maybe even an algorithm in the lightmapper

that looks for contrast changes and dynamically remaps them for sharper edges? Neither of these will be as good as a week or two’s worth of loving attention from you. But do you have that week? And the week that comes after that when something changes and it all has to be done over?

The funky intersection of art and tech throws up questions like this all the time, and the problem is always the same. Hand building every convex-hull-collision mesh? Twiddling the compression settings on every single animation, one at a time? Does your shader have more sliders in it than a DJ’s mixing table? In every case, there’s a defensible artistic reason why you’d want the ability to intervene by hand and make things better. Add it all up together, though, and you’re snowed in under an avalanche of little bitty decisions. On the dollars and cents side, that means everything takes longer. On the artistic side, it means you’ll end up unable to fix the things that drive you crazy. If you spend all day working through a magnifying glass, you’ll lose sight of the bigger picture.

To avoid this terrible fate, you need to stay focused on the right questions. “Will this make it look better?” is a good question, but “will it be worth my time?” is also worth asking. Whether you are designing a character or helping to build the pipeline for your next game, you need to balance the inner artist’s demands for perfect control with the little producer in your head who’s got an eye on the upcoming milestone.

TIME RELEASE

We started this month by thinking about letting go—sending your prize baby out into the world on its own. Dealing with the ominous tick of ship-time is all about knowing when to let go—when you’ve done enough to tell your story, when you’ve added enough details and need to move on, when you have to say, “let the computer do this, it’s not worth my time.” Learning to let go, to release, is a critical discipline for any artist. And discipline, of course, means two very different things. Sometimes, it means getting flogged. Sometimes, it means learning, and dedication, and growth. ❖

Jobs on



GAMASUTRA

The Art & Business of Making Games



The Leading Professional Game Source for Job Seekers and Employers



NEW! Job Seeker Center



NEW! Employer Job Manager



NEW! GDC Events Job Section

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

Log onto Gamasutra.com/jobs



SOREN JOHNSON

» DESIGN OF THE TIMES

GAME ECONOMICS

GAME DESIGN AND ECONOMICS

have a spotty history. Designing a simultaneously fun and functional economy is no easy task, as many design assumptions tend to backfire when they come in contact with the player. For example, the early days of *ULTIMA ONLINE* were infamous for the game's wild and chaotic economy. Zachary Booth Simpson wrote a classic analysis of *UO* in 1999, detailing some of the more notable problems experienced at launch:

- The crafting system encouraged massive over-production by rewarding players for each item produced.
- Over-production led to hyper-inflation as NPC shopkeepers printed money on demand to buy the worthless items.
- Players used vendors as unlimited safety deposit boxes by setting the prices for their own goods far above market value.

- Item hoarding by players forced the team to abandon the closed-loop economy as the world began to empty out of goods.
- Player cartels (including one from a rival game company!) cornered



Shown is the effect of a "glock bomb" in *COUNTER-STRIKE: SOURCE*.

the market on magical Reagents, preventing average users from casting spells.

MMO economies have come a long way since then; *World of Warcraft's* auction house is now a vibrant part of the game's economy and overall world, with many players spending much of their time "playing the market" to good effect. CCP, developer of *EVE Online*, even hired an academic economist to analyze the flow of resources and the fluctuation of prices within *Eve's* game world. Indeed, understanding the potential effect of market forces on gameplay is an important ability for designers to develop.

CAN THE MARKET BALANCE THE GAME?

Many designers have used economic game mechanics as a tool for balancing their games. For example, in *RISE OF NATIONS*, every time a unit—such as a Knight or Archer—is purchased, the cost of future units of the same type goes up, simulating the pressure of demand upon price. This design encouraged players to diversify their armed forces, in order to maximize their civilization's buying power. By allowing the "values" of different paths and options to float during a game, designers present players with a constantly shifting landscape, extending replayability by guaranteeing no perfect path to victory.

However, if taken too far, efforts to auto-balance by tweaking the economy

can destroy a game. In 2006, Valve conducted an interesting economic experiment within *Counter-Strike: Source*, implementing a dynamic weapon pricing algorithm. According to the developers, "the prices of weapons and equipment will be updated each week based on the global market demand for each item. As more people purchase a certain weapon, the price for that weapon will rise and other weapons will become less expensive."

Unfortunately, the overwhelming popularity of certain weapons trumped the ability of the algorithm to balance the game. For example, while the very effective Desert Eagle skyrocketed to

SOREN JOHNSON is a designer/programmer at EA Maxis, working on *SPORC*. He was the lead designer of *CIVILIZATION IV* and the co-designer of *CIVILIZATION III*. Read more of his thoughts on game design at <http://designer-notes.com>. Email him at sjohnson@dmag.com.

\$16,000, the less useful Glock flatlined at \$1, leading to some extreme edge cases (such as the pictured “Glock bomb”). A game economy is not a real economy; not everything can be balanced simply by altering its price. Gamers just want to have fun, and if the cost of the option considered the most fun is constantly tuned higher and higher until the price becomes prohibitive, players may not just alter their strategy—they may simply go play another game. The current price of gas may be making our real lives “unfun,” but only one real-world economy exists, leaving us no choice. Gamers are not in the same situation.

Ultimately, designers should remember that achieving perfect balance is a dubious goal. Players are not looking for another game like rock/paper/scissors, in which every choice is guaranteed to be valid, essentially encouraging random strategies. Players are motivated by reasons beyond purely economic ones when playing games. Raising the cost of a player’s favorite weapon is simply going to feel like a penalty and should only be done if the imbalance is actually ruining the core game.

PUTTING THE MARKET INSIDE THE GAME

Perhaps a more appropriate use of economic dynamics is as a transparent mechanic within the game itself. The board game world provides some great examples of such free market mechanics at work. German-style games *Puerto Rico* and *Vinci* both use increasing subsidies to improve the appeal of unpopular roles and technologies, respectively. In the case of the former, every turn no player decides to be the Craftsman, one gold piece is added as a “reward” for choosing that role. As the gold increases slowly, few players will be able to resist such a bounty, which nicely solves the problem of making sure all roles are eventually chosen.

Puerto Rico still has some clearly better and clearly worse options—they just change from turn to turn based on the current reward. In this case, auto-balancing actually keeps the game fun because players are rewarded for choosing less common strategies, instead of being penalized for sticking to their favorites. Perhaps more importantly, the effects of the market are spelled out clearly for the players ahead of time, so that no one feels the game is biased against them.

Perhaps the most elegant example of a pure free market mechanic based around actual resources and prices can be found in *Power Grid*, another German-style board game. In this case, players supply their power plants with a variety of resources (oil, coal, uranium, and garbage), all of which are purchased from a central market. Resource pieces are arranged on a linear track of escalating prices. Every turn, X new pieces of each resource are added to the market, and players take Y pieces away as purchases. As the supply goes up and down, the price correspondingly goes up and down, depending on where the next available piece is on the market track.

By making the supply-demand mechanic so explicit and transparent to the players, the market becomes its own battlefield, as much as the hex grid of a wargame might be. By buying up as much coal as possible, one player might drive the price out of the range of the player in the next seat, causing her to be unable to supply all her plants at the end of the turn, a disastrous event

in *Power Grid*. Thus, with a true open market, price can be used as a weapon just as much as an arrow or a sword might be in a military game.

THE BENEFITS OF FREE TRADE

Similarly, a number of modern strategy games, including *SINS OF A SOLAR EMPIRE* and the *AGE OF EMPIRES* series, have included free markets in which players could buy and sell resources, influencing global prices with their actions. These markets served as interesting “greed

www.rtpatch.com

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

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

DESIGN OF THE TIMES



The market track of *Power Grid* is shown.

tests" in that players are often tempted to sell when they need cash or to buy when they are short on a specific resource, but they know in the back of their minds that each time they use the market, they are potentially giving an advantage to another player. Buy too much wood in *AGE OF KINGS*, and your opponents can make all the gold they need selling off their excess supply.

Unfortunately, the market dynamics of these games tend to repeat themselves, with prices usually bottoming out once the players' total production overwhelms their needs. This effect stems from the fact that the game maps emphasize economic fairness—in *AoK*, each player is guaranteed a decent supply of gold, stone, and wood within a short distance of their starting location. Spreading resources randomly around the map could lead to a much more dynamic and interesting market mechanic but at the cost of overall play balance for a game with a core military mechanic. If your opponents attack with horsemen, what if there is no wood with which to build spearmen, the appropriate counter unit?

However, a game with a core economic mechanic does not suffer from such limitations. In most business-based games, specializing in a specific resource is a basic part of the gameplay. Thus, a free market mechanic can become a compelling part of a competitive game. The ultimate example of such a game is the '80s

classic *M.U.L.E.*, in which four players vie for economic dominance on a newly-settled world. Although only four resources exist (food, energy, smithore, and crystite), economies of scale encourage players to specialize. More importantly, players can rarely produce all the resources they need on their own, requiring them to buy directly from other players.

The game has a brilliant interface for facilitating this trade between players. Buyers are arranged along the bottom edge of the screen, with sellers on the top. As buyers move up, their asking price goes up accordingly. As sellers descend, their offer price decreases as well. When the two meet in the middle, a transaction occurs. Once again, the mechanic is explicit and transparent—player inventories and market prices are all clearly visible to everyone. Players understand that they either have to adjust their own prices to make a deal happen or hope that their rivals cave. Knowing how desperate another player might be to acquire the energy needed to power his buildings or the food needed to feed his labor, the temptation to pull every last penny from him is strong. In such a case, prices tend to fall only if the player is afraid someone else might sweep in to reap the profits. The game mechanic mined here by *M.U.L.E.* is deep and rich. Impoverishing one's enemies can be just as much fun as destroying them. ❖



MONTREAL INTERNATIONAL
GAME SUMMIT 08

IN MONTREAL WE LIVE IN A POLYGON



Photo: © Ait-6

NOVEMBER 18-19
www.sijm.ca



REGISTER BEFORE
SEPT. 30TH AND
SAVE UP TO 30%



Organized by:

 alliance
numérique www.alliancenumerique.com

1500 ATTENDEES + BUSINESS LOUNGE +
200 LEADERS + 600 BUSINESS MEETINGS +
80 SPEAKERS + 5 TRACKS + SERIOUS GAME +
EXHIBITION AREA

WHERE THE GAME BEGINS

Partners:

Québec 

Ministère de la Culture et des Communications
Ministère du Développement Économique, de l'Innovation et de l'Exportation

 Canada Economic
Development  Développement
économique Canada

Canada 

 Canadian
Heritage  Patrimoine
canadien



JESSE HARLIN

» AURAL FIXATION

BREAKING NEW GROUND

THREE YEARS AGO, THE TEAM BEHIND

STAR WARS: THE FORCE UNLEASHED called me into their conference room to look at a tech demo that would later debut at E3. At the time, the demo consisted of a large hall containing panes of glass, a brick wall, and a wooden beam ceiling. The demo's driver proceeded to throw Stormtroopers at everything in the room and I watched bricks crack, wood splinter, and glass shatter—each time different from the one before. The technology at the heart of this new breakable system is known as Digital Molecular Matter, or DMM, and is a physics-based materials simulator developed by Pixelux Entertainment. Its goal is to remove last generation art-swap breakables while allowing for new interactive materials such as bending metal, rubbery plants, or melting ice.

The technology looked impressive, but was completely silent. Each collision resulted in hundreds of thousands of variables and fragmented the original materials into everything from enormous hunks of matter down to invisible splinters. Facing endless variations, the audio team needed a solution that could sell DMM's realism without exceeding memory budgets.

DIGITAL MANAGEMENT MAYHEM

The first challenge was to decide how to generate the wide variety of possible sounds for DMM. One early thought was to approach sound for procedural matter from the data-driven realm of physical modeling synthesis. However, this was quickly ruled out due to the largely academic nature of the field as of 2005, meaning DMM would have to be tackled using thousands of unique audio recordings.

At first, we attempted a literal approach and scored DMM breakables with combinations of hundreds of tiny sounds. Splinters made splinter sounds, shards

sounded like shards, and every chunk of material made its own chunk sounds. After spending a couple of weeks doing material source recordings and tweaking the implementation, the end result was a completely unrealistic mess. It sounded like what it was—hundreds of disparate pieces of wreckage knocking together in front of a microphone.

The solution was to simplify and edge towards a hyper-realistic sound representation akin to that of film post-production. When something shatters, the brain does not process every last shard hitting the floor. Instead, the brain experiences a cacophonous impression of chaos defined by the behavior of an undetermined number of non-uniform pieces of debris.

The DMM engine consisted of 350 different material types that we were able to pare down into 20 different DMM sound materials, such as "organic hard" or "metal strong hollow." At a macro level, DMM allowed us three main behavior categories: collision, fracture, and bend. For instance, hitting a DMM wood prop might make it collide and fracture while hitting a metal prop might collide and then bend. Additionally, all DMM materials came in a range of small, medium, and large sizes. Of all the materials, glass was hardest to manage due to thousands of small particles generated from each shattered pane.

When it came time for implementation, each game level had its own .xml file that detailed all potential collision permutations applicable for only that level. In practice, fractures and bending did not require any data be kept for material-on-material relationships since each only dealt with a single material. While collisions and fractures were essentially instantaneous sounds, bending necessitated the use of bending loops—on average 3–4 seconds long and variable depending upon the size of the bending object—that were then augmented by banks of up to 15 randomized sweeteners. With collision, fracture, and bending behaviors figured out individually, the next step towards

the rich realism we wanted from DMM came when we began combining behaviors together allowing for instances where large bending doors could scrape or bang into the dirty ground.

With hundreds of sounds now triggering per DMM behavior, the last piece of the puzzle was a three-tiered system of voice instance limiting. The audio engine allowed us to first limit at the cue level, allowing us to set a maximum number of times each cue could trigger per frame. Then, supplementary instance limiting was added to the DMM-specific sub-bus in the game's main audio mixer. Lastly, because the PS3 was limited in available channels more so than the Xbox 360, we added an additional priority-based limiting system to help sort out the most important elements of each moment in-game.

LET SLIP THE PROPS OF WAR

Not every object was infused with DMM, though. Anything that could be picked up and thrown with The Force, non-deformable objects like walls and floors, and enemy bodies all used the Havok physics engine. In conjunction with the Euphoria AI engine, Havok also fueled our foley and footstep systems.

Havok had its own matrix that dealt with Havok-to-Havok object material collisions. Like DMM, Havok also allowed for the inclusion of small, medium, and large sound categories plus three levels of hit sensitivity. With these three hit sensitivities, a thrown object might bang hard against a wall, fall to the floor with a medium intensity, and then settle itself with a soft sound. Lastly, our proprietary audio engine allowed for standard audio parameters such as volume and pitch randomization, distance-based fall-off, and another level of Havok-specific instance limiting.

In the end, by combining intelligent instance limiting with multiple interlocking systems of physics-based collision and materials behavior detection, the result is a richly detailed world full of breakable materials that never sound exactly the same way twice. ❖

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.

THE CREATORS OF **ROCKBAND™**

HARMONIX®

IS HIRING
ARTISTS
PROGRAMMERS

WWW.HARMONIXMUSIC.COM



ACTIVISION®

GREAT GAMES

START WITH

GREAT PEOPLE



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



Visit our site: activision.com



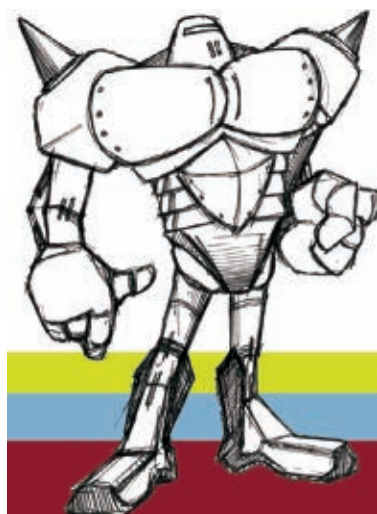
THE MAKERS OF EVE ONLINE ARE ACTIVELY RECRUITING LEADING TALENT

» ATLANTA - REYKJAVIK - SHANGHAI «

QA MANAGERS
ARTISTS
ANIMATORS
SOFTWARE DIRECTOR
SENIOR PROGRAMMERS

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

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



**CD-ED's Digital Arts
Technology Training Institute**
500 Kings Road, Suite 300 B1S 1B1
1.866.567.3010
902.567.0229
<http://www.darttinstitute.com>
info@darttinstitute.com

"With my ever-demanding schedule of working a full time job, managing a new business, and dealing with day-to-day family life, I thought I would never have the time to go to school to further advance my education. From the schedule DArTT provided me, I do have time for my family, my job, my new business and to advance my education to the next level. From my two years of experience dealing with the Center for Distance Education, I would recommend any of the courses that are offered and know they are backed with the education and friendliness of the instructors and staff."

Robert Prince now has a job at IMAX, doing compositing, rotoscoping, and modelling.

Disney Interactive, Ubisoft and Pixar Call the Digital Arts Technology Training Institute. Maybe You Should Too!

The Centre for Distance Education's DArTT Institute is the leader in online education, with a staff who has more than a century of combined experience in delivering quality college diploma programs. Their innovative, industry-driven, fully online and interactive programs give students the opportunity to immerse themselves in the digital environment to the degree necessary to truly excel in the gaming industry.

New software has been brought on board to permit instructors to conduct webinars - one-on-one or group real-time lectures that will provide levels of demonstration and support previously unavailable to this extent. Text-to-audio software has also been added which will allow students with learning disabilities to be on a level footing. This technology reflects DArTT's educational philosophy, which allows for all three of the basic learning styles.

DArTT's learning methodology places a heavy focus on portfolio development, and as every hopeful game artist knows, the portfolio will make or break you.

Online education graduates are in high demand, and employers often prefer them to traditional in-class grads for their self-directed habits, adaptability, commitment, and time management skills.

Online education means having wider choices. DArTT has the courses employers want, when and where the student wants to take them. With minimal disruption to their daily lives, a student can become a Character Animator or a 3D Game Artist regardless of where they live or work.

Self-paced flexible study methods allow students to earn a diploma on their terms. Continuous intake means courses start when the student's schedule allows. Personalized study packages help the students to learn with ease, and dedicated instructor support is unfailingly positive.

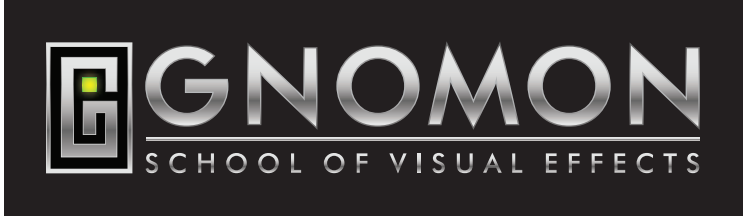
The college believes that their textbooks have to be excellent – and their instructors have to be better than that. The instructors have experience in the field and in addition to writing proprietary tutorials focused on the exact technical requirements expected in real live jobs, they can also prepare each student personally for what the job market is like when they've graduated. Directed Study projects provide for individualized portfolio commentary as well as career counseling just before graduation, and the school also has a job find placement program.

Online learning is a new experience for many people. To make the transition easier, students are able to interact freely with fellow students and staff concerning academic, social, and personal issues. DArTT's leading-edge online course management system allows the student community to interact more than ever before possible with an online program. Students report that although they learn on their own, they never feel alone.

The Guidance Department ensures every student's success by providing support services to anyone experiencing problems. This is very important to DArTT, who will continue to develop services that will provide students with the support and information they deserve.

Graduates from DArTT are in high demand. DArTT graduates are compositing and modeling at **IMAX** and contributing to **XBOX 360** games.

So what did **Disney Interactive Studios**, **Ubisoft**, and **Pixar** want to know when they called? Contact the Admissions Department at 1.866.567.3010 and find out!



Gnomon School of Visual Effects

Think you could be the next Tom Chilton working as a lead game designer on an epic like World of Warcraft? Or how about serve as a character designer on the next best-selling adventure game? Do you dream about working as a 3D artist on a game like Halo?

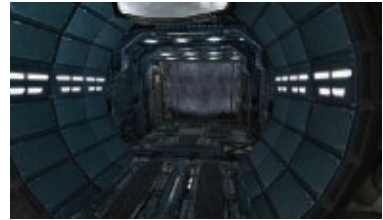
Whether your passion is for first-person shooters, multiplayer games or even military applications, earning a Certificate in High-end Computer Graphics at Gnomon School of Visual Effects is your hot ticket to an even hotter industry. Pick a specialty, like 3D modeling and animation, digital illustration, environment creation for games, just to name a few. Then, the faculty and staff at Gnomon can simplify things for you. Our innovative training facility offers a full curriculum for individuals seeking careers in the 3D industry, so you can easily connect with the programs and classes that meet your specific goals. At Gnomon, we not only provide the instruction and tools to make your career goals in the gaming industry a reality, but also provide an in-depth knowledge about how the gaming industry works as well as extensive networking opportunities and job placement. Gnomon is able to accomplish this like no other educational facility by providing our students with instructors who are all working professionals in the games, film and visual effects industries. Having this opportunity to learn from some of the most relevant and recognized CG professionals in the business consistently opens doors to a variety of game development careers for our current students and alumni.

With 2006 revenues in excess of \$7 billion (source: Entertainment Software Association), the gaming industry provides plenty of room, upward mobility, and creative license for you to flourish. Gnomon looks forward to being your primary resource for acquiring the attributes and skills that gaming employers look for. Those basics, listed on our website (www.gnomonschool.com) are followed by a brief rundown of various fields within the gaming industry- a list that might help you narrow your goals to fit your personality and skills. Finally, we'd be glad to provide you with any additional resources you may need to help guide you in your pursuit of your career goals, either in person with our Admissions Counselor or via email.

Gnomon School of Visual Effects specializes in training for careers in high-end computer graphics for the entertainment industries. Located in the heart of Hollywood, Gnomon offers a full curriculum for individuals wanting careers in 3D, professionals in need of specialized training, custom training for production studios and online courses. In conjunction with several major studios across Los Angeles, our instructors and our esteemed Advisory Board, Gnomon's curriculum and facilities have been designed to constantly evolve to reflect any new demands that arise in the entertainment industry. Founded by Alex Alvarez in 1997, Gnomon is located at 1015 North Cahuenga Boulevard, Hollywood, CA 90038. On the web, Gnomon School of Visual Effects can be found at <http://www.gnomonschool.com>.



Created by Fausto de Martini



Created by Ioannis Karathomas

Gnomon

1015 N. Cahuenga Boulevard
Hollywood, CA 90038
Phone: 323.466.6663
www.gnomonschool.com
Email: info@gnomonschool.com

"The graduates from your program will have no trouble joining the ranks of the digital effects industry. I recommend your courses to both novice and advanced professional users and hope that you have room in your classes to accommodate them all. Thank you for your hard work and dedication to education in computer animation."

Sande Scoredos
Senior Vice President,
Artist Development and Training
Sony Pictures Imageworks

"I've worked with Gnomon staff and faculty for the past several years. During this time, I've had the opportunity to meet, interview and hire many Gnomon-trained artists to join Imageworks' growing roster of Digital Artists and Technical Directors. I commend Gnomon's reality-based approach toward digital visual effects instruction and their dedication toward preparing students for today's market. I look forward to a continued relationship with Gnomon staff, faculty and of, course their ever expanding and multi-talented roster of students."

Stan Szymanski
Director, Digital Production
Sony Pictures Imageworks



SCAD prepares talented students for professional careers. Alumni of the SCAD School of Film and Digital Media work for companies such as Electronic Arts, Pixar, Sony Pictures Imageworks, Blue Sky Studios and Cartoon Network.



David Hale, undergraduate student in visual effects, Crab, Visual Effects Studio.

SAVANNAH COLLEGE OF ART AND DESIGN

SAVANNAH:

ADMISSION DEPARTMENT
P.O. BOX 2072
SAVANNAH, GA 31402-2072
912.525.5100 or 800.869.7223
admission@scad.edu
www.scad.edu

ATLANTA:

ADMISSION DEPARTMENT
P.O. BOX 77300
ATLANTA, GA 30357-1300
404.253.2700 or
scadatl@scad.edu
www.scad.edu/atlanta

eLEARNING:

ADMISSION DEPARTMENT
P.O. BOX 2072
SAVANNAH, GA 31402-2072
912.525.5100 or 800.869.7223
elearning@scad.edu
www.scad.edu/elearning

DEGREES OFFERED

- Bachelor of Arts (B.A.)
- Bachelor of Fine Arts (B.F.A.)
- Master of Arts (M.A.)
- Master of Fine Arts (M.F.A.)

SCAD

ATLANTA • eLEARNING • LACOSTE • SAVANNAH

SCAD School of Film and Digital Media

Named one of Kaplan's "25 cutting-edge schools with an eye toward the future," the Savannah College of Art and Design offers an innovative curriculum designed to provide an excellent arts education and effective career preparation for students. With three locations as well as online programs, the college attracts students from all 50 states and from more than 90 countries.

Animation

SCAD's animation program was ranked among the top three animation programs in North America by 3D World magazine. Students use industry-standard software and study under faculty with extensive field experience. Gary Goldman ("The Secrets of N.I.M.H." and "All Dogs Go to Heaven") was the Winter 2008 artist-in-residence.

Broadcast Design and Motion Graphics

Broadcast design and motion graphics students train to create main titles for television and film, commercials and network design, and design and animation for new formats such as cell phones and iPods. At SCAD's annual Inspire symposium students hear from top motion graphics and media art talent and network with industry leaders such as Psyop, MTV, süperfad and PES.

In 2007, two alumni made were finalists in the Chevy Super Bowl XLI College Ad Challenge; and one student won the Film Society of Lincoln Center's Student Trailer Competition, presented by HBO Films.

Film and Television

Film and television students produce films using digital technology and audio workstations, nonlinear editing systems, and a Movie Magic lab for screenwriting, scheduling and budgeting.

SCAD's annual Savannah Film Festival features student and professional films screenings, panel discussions and casual interaction with industry professionals including Michael Douglas, Charlie Rose, James Franco and Todd Wagner.

Interactive Design and Game Development

Interactive design and game development students create interactive media and entertainment, Internet applications, video games, virtual online environments and a variety of computer applications.

Each year SCAD hosts the Game Developers eXchange, where game developers, educators and students hear about behind-the-scenes knowledge of the game industry. Students also attend the South by Southwest Interactive Festival, the Game Developers Conference, Flashforward, E3, NAB, BDA and SIGGRAPH.

Sound Design

SCAD was the first university to offer a degree in sound design. Sound design students explore sound for moving pictures, music production and sound art.

SCAD has a three-room studio boasting a Digidesign ICON mixing board with Pro Tools software, 5.1 speakers and mixing facilities. More than 2,000 hours of sound effects, 1,000 hours of production sounds and numerous original scores are accessible via an online library.

Visual Effects

Visual effects students learn through a combination of technology and art, working individually and collaboratively within a framework of cooperative activity that reflects the real-world experience of film production. SCAD production facilities offer students access to digital tools including Autodesk Maya, Shake, RenderMan and Houdini.



High School Students Interested in Video Game Development Career Should Look to Southern Methodist University in Dallas

The Guildhall at Southern Methodist University, and the Division of Art at SMU's Meadows School of the Arts and SMU's School of Engineering have started the first-of-its-kind degree programs in the nation offering a bachelor's degree in fine arts or computer science and a master's in interactive technology degree in digital game development (MIT) within five years.

Students who choose the dual degree plan complete their general education and School of Arts or School of Engineering major requirements at the SMU main campus and then move to the SMU campus in Plano to complete The Guildhall at SMU program. At the Plano campus, students concurrently finish their undergraduate degrees and start the master's program.

Gaming has increasingly broad penetration into all forms of cultural expression, and likewise draws more and more from the framework of the arts, physical and social sciences. Because of the depth of immersion possible within the graduate studies of the Guildhall MIT program, the breadth afforded by the BFA degree and the BS in computer science will give the next generation of game designers unique tools and experiences to bring to the profession as well as the game environment itself.

The BFA/MIT and the BSCS/MIT programs form a unique collaboration between these three disciplines, with exciting implications for the future of art, engineering, and interactive simulations.

Preparing students for fantasy jobs in the real world, The Guildhall at SMU program has a 95 percent placement rate. In four short years since the program opened, 200 of its graduates now work for more than 70 of the world's leading video game companies. Combining the foundation of a fine arts or computer science undergraduate degree with the technology, project team management and game theory knowledge encompassed in The Guildhall at SMU graduate program truly prepares students for careers in the gaming industry.

Students interested in the program can get more information by contacting the Division of Art at SMU's Meadows School of the Arts at 214.768.3217 or SMU's School of Engineering at 214.768.3041.



SMU's main campus is in the heart of Dallas, one of the top places in the U.S. to live, work, learn, and play.



Shantytown was a game developed by SMU graduates.

How to Contact:

The Guildhall at SMU
5232 Tennyson Parkway
Building 2
Plano, TX 75024
972.473.3539
<http://hs.guildhall.smu.edu>

How to Contact:

"Involving experienced, successful and respected professional game developers with writing The Guildhall at SMU's curriculum was a very smart move and will likely give an advantage to all students in the program as they strive to become game makers themselves."

Randy Pitchford
President, Gearbox Software

GDC

09 learn
network
inspire

www.GDConf.com



Game Developers Conference

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

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

MAKE MORE ENEMIES

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

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

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

VFS™

vfs.com/enemies



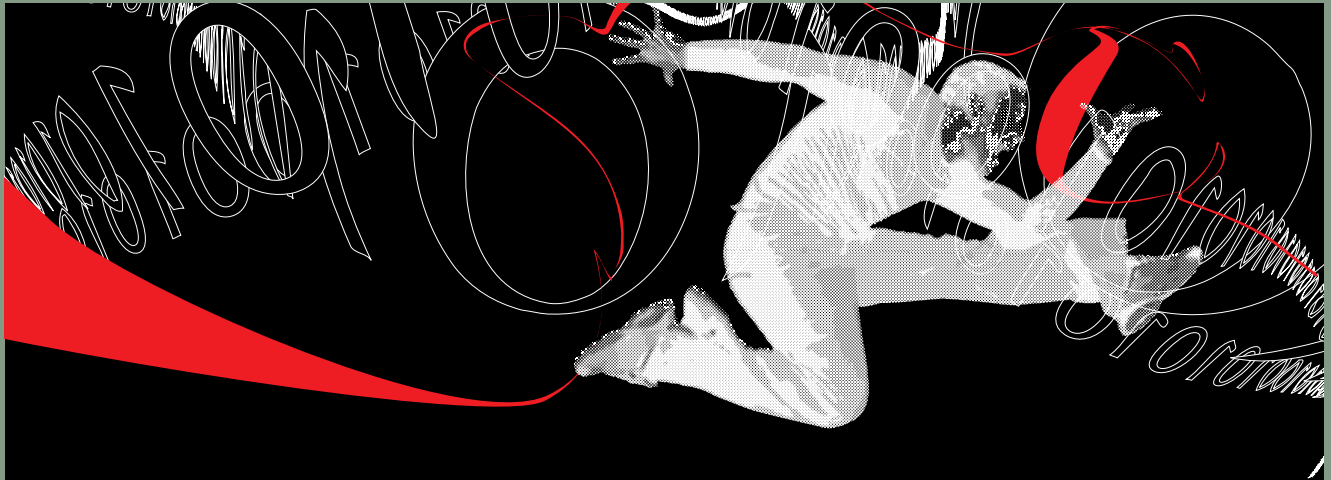
VFS student work by Jason Martin

Serious Game Design Institute

SANTA BARBARA CITY COLLEGE

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

Visit: sgdi.sbccc.edu for current opportunities



BECOME A MASTER OF DIGITAL MEDIA

We offer a 20-month master's degree in entertainment technology and digital media. This powerhouse graduate program combines industry-facing curriculum, real-world projects, and a four-month internship in Vancouver, Canada - videogame capital of the world.

MASTERS OF DIGITAL MEDIA PROGRAM @ THE CENTRE FOR DIGITAL MEDIA
Great Northern Way Campus | Vancouver, CANADA | mdm.gnwc.ca

GREAT NORTHERN WAY CAMPUS »



ADVERTISER INDEX

COMPANY NAME	PAGE	COMPANY NAME	PAGE	COMPANY NAME	PAGE
3Vis	30	Foundation 9 Entertainment	19	Pocket Soft	49
Activision	54	Full Sail Real World Education	36	Replay Solutions	12
Alliance Numerique	51	Harmonix Music Systems	53	Santa Barbara City College	61
CCP North America	55	Havok	23	Scaleform	25
Center for Digital Imaging	63	Image Metrics	6	Simutronics	11
Cogswell Polytechnical College	63	Masters of Digital Media Program	62	Techexcel	35
Conceptopolis	33	Metro Orlando Economic Development Commission	CV3	Trinity	CV2
Digimation	29	NaturalMotion	3	University of Advancing Technology	63
Epic Games	15	North Carolina State University	40	Vancouver Film School	61
Forterra Systems	43	Paynova	50	Versant Corporation	37-39

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

CENTER FOR DIGITAL IMAGING ARTS AT BOSTON UNIVERSITY

WALTHAM, MA | WASHINGTON, DC

CREATE your WORLD




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

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

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

GET EDUCATED

It's the first time in my life
that I've felt like I actually fit in.
Or at least I'm happy now that I don't
Truly **OBSESSIVE COMPULSIVE**
dedicated video **MANIAC**

who has found true fulfillment
& that means 

a career in **GAME DEVELOPMENT**
State of the Art—hardware and software
AT MY DISPOSAL

Game Design & Programming Game Art
Implementation of common programming APFS
Writing code, what about you?

「The LAN」
tournament
「ROCKED」

My demo reel turned them
into web monkeys

So you want to learn how to create long hair
using Maya, paint effects and cloth?



WE SPEAK G33K

Geeks transform into Alpha-Geeks
by becoming fluent in:

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

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



Offers bachelor
degrees in:

Digital Art and Animation
Game Design
Modeling
Animation

Digital Audio Technology
Audio Engineering

Digital Arts Engineering
Software Engineering
Computer Engineering

WASC Accredited
Financial Aid Available

admissions@cogswell.edu
www.cogswell.edu

1176 Bordeaux Drive
Sunnyvale, CA 94089
800.264.7985

Robot model by John Corvella
© Cogswell Polytechnical College



MATTHEW WASTELAND

ARRESTED DEVELOPMENT

DEPRESS RELEASES

SPEND SOME TIME FOLLOWING THE industry and things are likely to settle into patterns after not too long. The seasons flit by with their games and their dramatic occurrences, and behind it all is the constant, droning noise of press releases, which all begin to sound the same after a while. They usually go something like this.

WIDE-EYED GAMERS CRAP THEMSELVES FOR ACTION TITLE® 4 The wait is over! The trillions of gamers across the nation who have been frothing at the mouth for more wild, destructive, pulse-pounding action can now satisfy their lust for carnage with this explosive new ride through the brutal, post-apocalyptic world of ACTION TITLE® 4: VICTORY WAR™: DUTY OF OPS™ for the Sony® PLAYSTATION™ 3™™™ and Microsoft™ Xbox® 360™™™™!

On store shelves today, ACTION TITLE® 4 puts gamers in complete control of the most advanced and powerful assortment of weaponry ever conceived of in a video game! Jaw-dropping visuals, fiendish artificial intelligence, and words like ops, tactics, duty, obligation, cell, and 6, will keep them coming back for more! ACTION TITLE® 4 is the latest brainchild of the star developers behind such mega-hits as ACTION TITLE® 2: THE WAR IS STILL HAPPENING™ and ACTION TITLE® 3: MORE WAR FIGHTING IS NOW™!

SORT-OF FAMOUS PERSON TO BE ASSOCIATED WITH GAME Middle Tier Publisher, LLC today announce that genuine middle-tier Hollywood talent will now be associated

in some capacity with their upcoming middle-tier video game title.

"I am extremely excited to be a part of this project," said the perhaps ironically-named Talent. "I have always loved this video game series. The name escapes me right now but it's the one I just signed the deal for."

"Being associated with middle-tier Hollywood talent brings us the kind of middle-tier legitimacy that we've always chased after," said Middle Tier Executive. "Now our consumers can say, 'hey, this game has that guy that was in that movie, you know, the one with the guns and the running around and stuff?'"

The game will also feature music by a composer who has worked on real Hollywood projects, like for real. Please notice us!

RANDOM STUDIO LICENSES A MIDDLEWARE PACKAGE

Random Studio Nobody Has Heard Of today announced that it has licensed the industry-leading Amaze-o® QuickEvents™ middleware application program.

"This agreement furthers Random Studio's strategy to arm itself with the top technology in the industry," said Figurehead Developer. "With this new weapon in our high-powered arsenal, we'll be able to create incredible experiences the likes of which have never before been seen, ever."

"The Amaze-o® QuickEvents™ allows game developers to create unbelievable Quick-Time Events with no work at all in exactly zero seconds," said Middleware Salesman. "With its library of thousands of pre-made button combinations and timings, anyone who buys our package can legitimately call themselves a master game designer."

The Amaze-o® QuickEvents™ solution has been licensed by over three studios worldwide.

GAME DEVELOPMENT CONFERENCE GOING ON SOMEWHERE

The Alaskan Game Developers Association has announced that the 14th Annual Alaskan Game Developer's Conference will be held at the Red Dog Saloon in Juneau, Alaska this summer. Attendance is expected to increase over last year's record attendance of over 12 game developers and tourists who wandered in from the cruise ship outside.

SOME BUSINESS THING HAS JUST HAPPENED

Big Giant Publisher, Inc. today announced that some kind of business thing has occurred that will be really good for its stock price. In fact, the news is so good that you should buy some. Like right now.

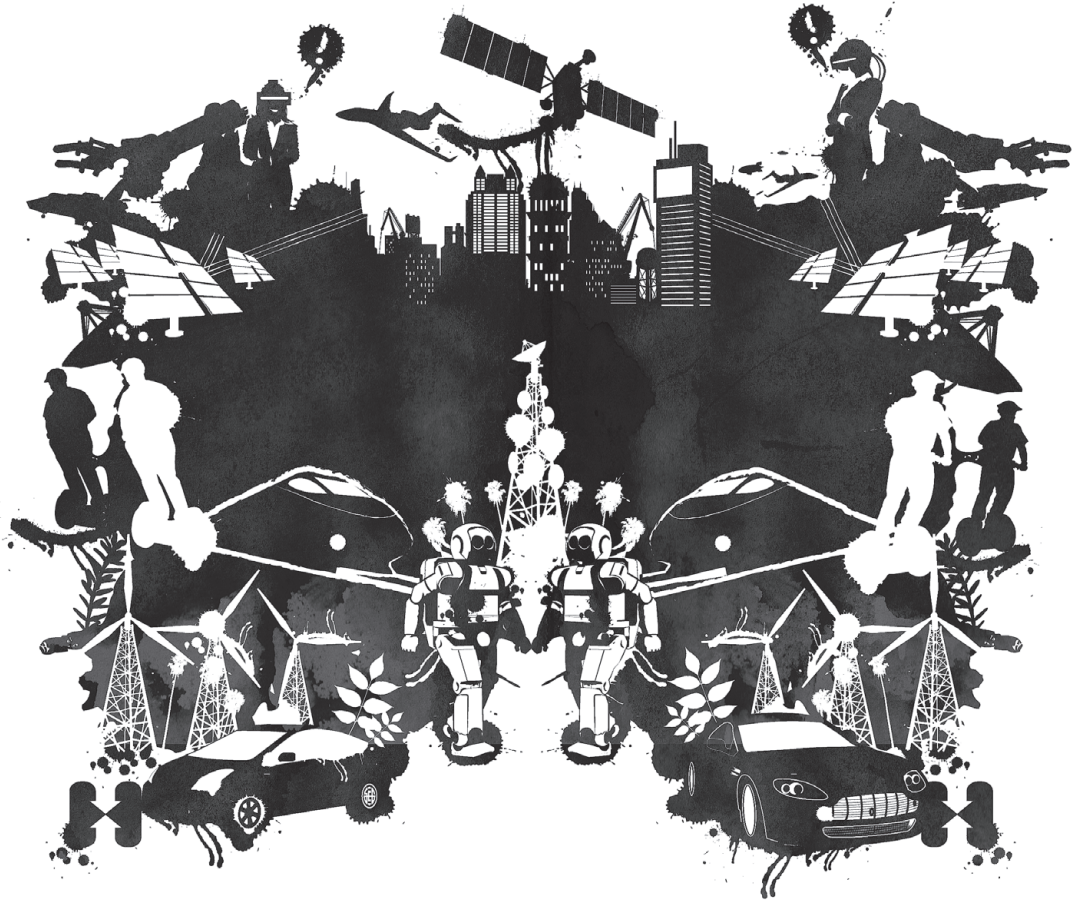
"Now more than ever, we are ready to leverage synergies in cross-media franchise development strategic business ... what? I've lost my place," commented the CEO. "Oh, here I am. Don't forget to mention our robust holiday portfolio and key initiatives in emerging markets across the globe."

Big Giant Publisher Inc. is a big, important player in the game industry. Ask your nephew and he'll be sure to tell you all about them. The company maintains operations in many fancy-sounding areas.

Disclaimer: statements in this press release that involve expectations of the future are predictions, and involve a number of risks and uncertainties. Big Giant Publisher often uses words such as "maybe," "perhaps," "not really," "actually, no," and "psych!" to help identify these statements. Factors that may cause the company's actual future results to differ from the stuff we made up could be things like our games sucking, or our inability to ship them on time. But none of that will happen, because we have all the new innovations this time. For reals! ❖

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

WHAT DO YOU SEE?



Future + Planning

In Orlando, we see tomorrow. A place charged with creative energy and entrepreneurial spirit. An economy built on a wide variety of emerging industries and led by a diverse group of innovative thinkers. A community determined to live up to its reputation as one of the world's "most fiercely competitive" locations for business. Look closer. You'll find tomorrow has arrived today in Metro Orlando!

Call 888.867.2489

www.orlandoedc.com

Metro Orlando Economic Development Commission

Putting imagination to work[®]
ORLANDO



WARNING: This product can make you more efficient.
425-893-4300 www.radgametools.com/granny3d

