CMP
United Business Media

MARCH 2007

# gamedeveloper

## THE LEADING GAME INDUSTRY MAGAZINE

>> **STATE OF THE INDUSTRY**
HOW DIGITAL DISTRIBUTION
IS CHANGING THE MARKET

>> **THE INNER PRODUCT**
SIMULATING FLUIDS

>> **SECURITY MATTERS**
PROTECT YOUR
PLAYER BASE

**POSTMORTEM:**
**EPISODIC GAMING:**
# DISSECTING
# SAM & MAX

# gamedeveloper

# FEATURES

26

# POSTMORTEM

# DEPARTMENTS

# COLUMNS

**COVER ART:** STEVE PURCELL

# [ GAME PLAN ]

## SERVING DUTY

**WHAT'S THE WORST THING THAT SOMEONE CAN** say about the creative business that you're a part of? Probably that you don't care. But this was the exact specter that Doug Lowenstein conjured up in his speech at D.I.C.E. in Las Vegas, effectively his farewell to the game business after an epic stint heading up E3 organizers and anti-game law blockers the Entertainment Software Association.

And what a farewell it was. Very specifically, Lowenstein said, "It's very easy on the keyboard to type up a bunch of criticisms about how ESA isn't doing this or that right. Then you look and where are these people? Are they getting in the fight? Are they making political contributions? Are they going to their senators? No, they're sitting on their hands. I'm sick and tired of people sitting on their hands."

### POLITICAL PUSH
Perhaps one of the problems for some *Game Developer* readers has been that the ESA has felt a little remote from the day-to-day realities of game development—being publisher funded, and heavy on the corporate membership, at least from the impression I got from attending Nite 2 Unite and other ESA charity events. But with the association awaiting a new head, as of press time, and the downsizing of E3 radically changing the way its lawsuits and lobbying are funded, this seems like a good time to reflect on who we, from the most junior tester to the most experienced CEO, really do want to swoop in and save us from those evil politicians.

Here's where I think the issue is. If it weren't for the ESA, we'd all be screwed in the North America game business. For one, there would be practically arbitrary state-based laws, which would mean video games were banned, differently rated, or differently handled at retail in any number of different parts of the U.S. For another, we could see an influx of politicians who might otherwise stay out of the debate, drafting more games-as-porn bills due to their successes in other states or floating the idea of pressing criminal charges against clerks selling M-rated games to underage buyers.

But, though the ESA and Lowenstein seem to care deeply about games' right to free speech and to not be censored, the fact is that the association is funded at the highest level by corporate interests who essentially care about their profits. There's nothing wrong with this, of course, but compare it to an organization like the American Civil Liberties Union, which gets a great deal of its income from individual membership dues.

### IN SERVICE OF THE INDUSTRY
So the ESA has started the Video Game Voters Network, and Lowenstein was also pretty steamed that very few in the industry had even stepped up to join that. In fact, I felt so bad that I just now this evening went and signed up.

Having done so, I didn't really understand, well, what I could do next. But I do feel better that I did it. Yet ... I didn't give them any money, and I don't really understand how signing up can help to make a difference, other than to be used for citation purposes. There's no forum to discuss issues or other ways to interact with the Voters Network community, though emails are sent out.

There's also an alternative now, thanks to Hal Halpin's ECA, or Entertainment Consumers Association. Membership costs a yearly fee, but, with all respect to Halpin, the effort is still rather nascent. You sign up, but where do your membership fees go? Why can't you interact with the community here either? Can the now ECA-affiliated GamePolitics.com actually include ECA-related calls to action in it, or does that break its impartial point of view?

### A UNITED FRONT
We need unity. We need the publishers to keep supporting the ESA. And we need more grassroots organizations like the ECA to start springing up.

But we need a bigger church to unite under, one that could include the IGDA, the ECA, the ESA, and beyond. Are there more clever ways to protest these issues, via Flash games, or flash mobs, or good old-fashioned signs outside courthouses, or intelligent discourse in the court of public opinion?

Most of all, Lowenstein is right. We need to care, because if we don't care about this now, we won't be allowed to develop the art form in the ways that we should and could.

*S!*

*Simon Carless*
*Editor-in-Chief*

# ALL GA

## DEVELOPERS ASSESS THE APPLE iPHONE

**The company that seemed to take over the modern world with the iPod has recently unveiled an even more chic little device, the Apple iPhone. This snazzy little gadget boasts a dual-input touch screen, hard drive, MP3 and video player, and proper Mac operating system, not to mention a $500 minimum price tag.**

**On paper, the iPhone sounds like a great machine for playing games, though Apple has not announced any plans for games as of press time.**

**What does the handheld industry make of this device? Is it the next generation of mobile games?**

—*Mathew Kumar*

# ESA HEAD SAYS GOODBYE, BERATES INDUSTRY

**DOUG LOWENSTEIN IS LEAVING THE GAME** industry, and he's angry. Behind him lies a legacy formed through the ESA, the game industry's primary political voice and creator of the ESRB and E3, both impressive industry staples. Just who is he mad at? You, me, and everyone in this industry.

"There is nothing more important than defending the first amendment," said Lowenstein, in his keynote speech at the D.I.C.E. Summit in Las Vegas this February. "It is a fight worth fighting. It's a fight I'm proud to fight, and we need to keep fighting. But ... sometimes that's not easy. Plenty of things are put out that are art that don't necessarily ennoble the culture, but they're protected."

What really bothers Lowenstein, who announced his resignation from the ESA in December after serving as its founder and president for more than 12 years, is that most developers just don't stand up for themselves after releasing games that offend a certain section of American politics. "If

**Doug Lowenstein**

you want the right to make what you want, if you want to push the envelope, I'm out there defending your right to do it," he said. "But, dammit, get out there and support the creative decisions you make. If you want to be controversial, that's great. But then don't duck and cover when the shit hits the fan. Stand up and defend what you make."

Lowenstein's statements come on the crest of a wave of controversy about violent games, for which the ESA has been the primary public defender. Game companies themselves typically assert that they have "no comment," while the ESA carries the flag into battle. It's true that publishers

pay for what the ESA does, but there's also something to be said for fighting one's own battles, especially as developers are often primarily responsible for these games' violent or sexual content.

The ESA is facing criticism not only from political groups, but from within the industry itself. "[Industry figures] always seem to think there's a better way to do something," Lowenstein said. "It's very easy on the keyboard to type up a bunch of criticisms about how ESA isn't doing this or that right. Then you look and where are these people? Are they getting in the fight? Are they making political contributions? Are they going to their senators? No, they're sitting on their hands. I'm sick and tired of people sitting on their hands."

Lowenstein's statements may be harsh, but it is worth thinking about what we as an industry have done to make him, our chief civil defender, so disappointed in us.

—*Brandon Sheffield, Frank Cifaldi*

# DGET, NO GAME?

### MIDORI YUASA, PRESIDENT, CAPCOM INTERACTIVE

The iPhone has the potential to completely change the landscape of our industry. One of the biggest problems with mobile games is that many consumers still find it difficult to download, install, and find mobile games on their handsets. It seems likely that, given its track record, Apple will address this problem with an elegant and consumer friendly solution.

   In addition, with Apple's expertise in consumer marketing, there is the potential for publishers to further penetrate the market by potentially expanding their product offerings via an off-portal distribution system like iTunes. That possibility alone is cause for excitement.

### H.E. MAH, VICE PRESIDENT OF MARKETING, OZURA MOBILE

Apple expects to sell about 10 million iPhones by the end of 2007, while market leader Nokia has more than 30 percent of the worldwide mobile phone market, which means about 90 million handsets as of Q3 2006. Mobile game developers are still going to focus on existing platforms, particularly as iPhone is a completely closed system.

   As the iPhone has no keypad and the controls are all based on touch screen, it's really going to affect the users' experience when it comes to mobile gaming. Perhaps the iPhone is not a true multi-entertainment device.

### MATT BOZON, CREATIVE DIRECTOR, WAYFORWARD

I love the idea of a button-free phone. Look at what touch did for the Nintendo DS. ... A dynamically changing interface allows the phone and games to each use what works best for them instead of being in constant conflict.

   Even if there are mightier Pocket PC phones out there that already do this, they still invite productivity, and the fun-loving iPhone looks like it's more about checking out than checking in.

   A handful of touch games like COOKING MAMA would be great to have available for download, especially if presented like an expanding library instead of the flavor-of-the-week. Since it's Apple, the company could enforce standard technology so that games would perform the same on every iPhone model—something that mobile gamers in particular tend to miss out on.

### MIKE YUEN, DIRECTOR, GAMING GROUP, QUALCOMM

The iPhone presents an interesting proposition, but there are a couple of worrying aspects once you look past the design. The phone itself is somewhat limited, as initially it's only going to be running GPRS so the connection will be slower than a true 3G network, and that's going to limit what you can do with it from a game perspective. Also, it's completely touch screen, and so like the iPod before it, the games that will work on it will necessarily be more casual.

   The other thing is the price point. $500 to $600 is a little high for the average consumer. That's not a mass market consumer phone.

## IN THE NEXT ISSUES OF GAME DEVELOPER ...

"Game Developer's 6th Annual Salary Survey," April 2007 issue

The Professional Career Advancement issue, May 2007 issue

6th Annual Game Developer's Game Career Guide, Fall 2007 special issue

# THE NEW USER INTERFACE
# EXPERIENCE

## Scaleform® GFx

Enhance your game with hardware accelerated
Flash® UI and vector graphics

## Scaleform® GFC

Comprehensive C++ framework to support
demanding PC UI requirements

"Using our new state of the art CryENGINE 2 and Scaleform GFx,
Crysis will deliver an unrivalled cinematic experience."

Cevat Yerli, CEO & President of Crytek

"Mass Effect is only the beginning of a long-term commitment by
BioWare to deliver an incredibly immersive cinematic UI experience."

Greg Zeschuk, President of BioWare

"The Scaleform GUI system has been extremely valuable in
helping us create a streamlined, easy-to-use interface in Civ IV."

Jeff Briggs, CEO Firaxis Games

www.scaleform.com

## Scaleform®

# SKYMATTER'S MUDBOX 1.0

## BY RONNIE ASHLOCK

**UNTIL THE PAST FEW YEARS, ARTISTS** have had few real methods of modeling three-dimensional sculptures on the computer that were even remotely similar to sculpting in real life. Forced to model with polygons, the technical demands of creating virtual sculptures required seemingly counter-intuitive ways of working.

As computers and software have become more capable, the constraints have loosened. Skymatter's recently released Mudbox 1.0 is one of a few new products that tries to unshackle artists even more, offering them greater creative freedom in digital clay.

### FROM MUDBOX ARE WE CREATED
Like Pixologic's Zbrush 2, another revolutionary sculpting package released a few years ago, Mudbox is a free-form, high-frequency mesh sculpting application designed to give artists the ability to sculpt polygonal surfaces much like virtual clay.

Comparisons to Zbrush 2 are inevitable. Both applications are subdivision surface modelers that create super dense meshes that can be sculpted. Both applications are tailored around working almost exclusively with a digitizing tablet. And both are designed to fit nicely into existing pipelines, with the ability to generate normal and displacement maps.

But the similarities quickly end when the artist begins to dig a little deeper. Mudbox is focused purely on sculpting whereas Zbrush combines 2D painting with 3D sculpting in a sometimes convoluted package. There are no texturing or digital painting tools in Mudbox, and some artists may be disappointed to discover they will have to look to other packages (at least currently) for working on color or specularity maps. This is one area where Zbrush has an apparent advantage over Mudbox. The material system in Mudbox is also significantly simpler than the one found in Zbrush.

The sculpting tools in Mudbox 1.0 are highly responsive and easy to use, and the model is in true 3D at all times. There's no need to pop into a separate



Mudbox 1.0 is a sculpting program and SDS modeler with a layering system similar to Photoshop's.

mode (à la Projection Master in Zbrush 2) to add high frequency details like wrinkles, dents, or scars. This greatly speeds up modeling.

Also, an artist can have as many objects in a scene and work on these objects as freely as he or she chooses. Large adjustments in mass and fine detail work are all done with the same tools in the same window.

### SDS MODELER
Mudbox is amazingly simple and quick to learn, favoring a layout and camera scheme that behaves almost identically to the controls found in Autodesk's Maya. Mudbox also has a scene editor and "channel box" menu that will be instantly familiar to Maya users. Brushes, stamps and stencils (images used to displace areas on a mesh) are easy to use and edit. Mudbox comes with a nice selection of pre-built stamps and stencils, which artists can augment by loading new images via a handy image browser.

Mudbox does not have a hard-coded cap set on mesh subdivision levels. It's no joke to say that this program can

handle unbelievably dense meshes with relative ease. Hardware processing power is the main limiter, so the beefier your system, the better performance you'll see. The local subdivision system, however, keeps the playing field relatively level for those who don't have bleeding edge hardware.

Instead of subdividing a whole mesh globally, artists are encouraged to paint a selection of faces, hide the rest of the mesh, and subdivide only the areas they need to detail. It's possible to have, for example, parts of a mesh like the area around a character's nose and eyes at a higher subdivision level than the area around the ears or back of the head. This keeps a mesh from exploding into a ridiculously heavy and unmanageable mess.

### ALL LAYERS, NO CONSEQUENCES
Mudbox has a very robust layer system that behaves similarly to the layer system found in Adobe Photoshop, only in this case, it works in 3D. Artists can take a base mesh, create a new layer, and sculpt new details or make mass

adjustments, and the changes reside exclusively on the new level.

A transparency slider works on each layer allowing the artist to slide the strength of the layer from 0 to 100. The transparency sliders behave almost like mixers or morph target editors, allowing artists to dial in the strength of each layer so that changes can be strong or soft depending on the setting of the slider. Additional layers can be added the same way, and new details can be added separately to each layer. This may be the single most powerful feature in Mudbox because artists can have several different versions of a single model built into one mesh. The penalties for making changes are basically eliminated. Don't like how something is turning out? Delete the layer. Does the art director want you to elongate the nose and soften the chin? Add a new layer and make the changes without affecting the other layers.

The layers system in Mudbox is nothing short of revolutionary—and the product is only in phase 1.0. Personally, it'll be hard to go back to other ways of modeling now that I'm in the swing of it.

Skymatter also bequeathed to young

Mudbox 1.0 a very robust normal map and displacement map generation tool. Easy to use and understand, it works with one mesh with many subdivision levels, as well as arbitrary meshes that can easily be imported as .OBJ files. The interface for baking these maps, like the software, is fairly easy to figure out intuitively. An advanced features roll out gives users plenty of options in setting up customized normal map burns.

## GOING PRO

Currently only available for purchase via download, the transaction for the application is straightforward. Authorizing the software was easy and uncomplicated, as well. Mudbox comes in two flavors: Basic, which retails for $299 and limits some of the better features offered in the package, and Professional, which goes for $649.

The Basic edition has all the same functionality but is limited to only five layers per object and doesn't support 32-bit High Dynamic Range images. It's not intended for commercial use. While the Professional version represents the best value for the money, the Skymatter

developers have said on the company web site that they will support an upgrade to the pro version later on, should the customer choose to purchase it.

Both Basic and Pro versions include access to the private forums and online documentation where most questions are answered quickly. The forums are handy and the documentation is solid and easy to understand. Video tutorials go over the basics and should help users who don't understand some of the features.

With a very streamlined, stable, and instantly understandable interface, extremely powerful, well thought-out tools, and low learning curve, Mudbox is an amazing program and is highly recommended. Artists looking for a fast, intuitive, and stable high-frequency modeling tool should seriously consider climbing on board with Mudbox. The modeling tools are the very best on the market for organic sculpting. ✕

**RONNIE ASHLOCK** *has more than six years experience making games. Email him at* **rashlock@gdmag.com**.

# product news

**AUTODESK MAYA 8.5**
**AUTODESK**
In mid January, Autodesk announced a half-point release of Maya, one of the company's 3D animation, modeling, and rendering software products. Version 8.5 offers many enhancements for
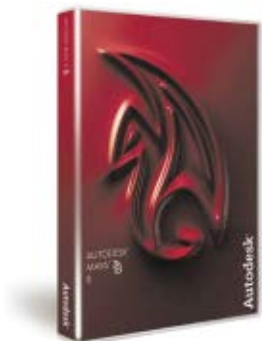
Autodesk Maya gets a mini facelift with version 8.5.

animators in the film industry, with only a few additional perks for game developers, including Python scripting. Maya 8.5 is available currently as a Universal application release for Intel- and PowerPC-based Macintosh computers, as well as on Microsoft Windows and Linux platforms.
*www.autodesk.com/maya*

**MORPHEME ANIMATION ENGINE**
**AND TOOL CHAIN**
**NATURALMOTION**
NaturalMotion is releasing its new animation engine, called morpheme, for next-generation game development. The company, which also developed endorphin and euphoria, will be showing the new engine at the Game Developers Conference and is set to beginning shipping February 28. Morpheme consists of two components: morpheme:runtime (a

run-time engine optimized for PlayStation 3, Xbox 360, and PC) and morpheme:connect (a 3D authoring tool).
*www.naturalmotion.com*

**HAVOK 4.5**
**HAVOK**
Havok has officially released Havok 4.5, the latest update to its modular suite of artist tools and run-time technology. The new version is optimized for the current roster of game consoles—PlayStation 3, Xbox 360, and Nintendo's Wii—and reportedly can run a typical game scene on the PlayStation 3 five to 10 faster than Havok 4.0.
*www.havok.com*

**FILEHAMSTER**
**MOGWARE INC.**
Mogware recently put on the market a new piece of software called Filehamster, a small and non-

invasive content management solution for game developers. The version-tracking software, which automatically renames files as users work on them, doesn't require a client/server setup and is available through download only. Better yet, it's free. Mogware says it has created Filehamster with expandability in mind by making it plug-in ready.
*www.filehamster.com*

**PAINTER X**
**COREL**
Corel last month unleashed Painter X, the latest upgrade to its virtual painting program. One thoughtful new feature is a set of composition tools that lets the user turn on guidelines showing divine proportions (also known as the golden ratio) and the rule of thirds. Software prices start at $219 for an upgrade or $419 for new recruits.
*www.corel.com*

# Navigate Your 3D World
## Starting at $59!

**SpacePilot™** $399
Adaptive sensing technology
& built-in display

**SpaceExplorer™** $299
Advanced ergonomic design

**SpaceNavigator™** $59/$99*
Ideal for home, school & work*

**SpaceTraveler™** $199
On-the-go navigation control

3Dconnexion's award winning 3D navigation devices deliver intuitive touch-sensitive control of three-dimensional models and environments. Fly through scenes and manipulate models with an agility and precision unachievable with a mouse and keyboard. Unify the navigation systems of multiple 3D programs in one simple to use device. Achieve unparalleled productivity with professional 3D applications.

**At work, home, school or on the road, 3Dconnexion offers a navigation device for your 3D world.**

*SpaceNavigator™ PE (Personal Edition) license supports non-commercial use of device only. SpaceNavigator™ Standard Edition suggested retail price is $99.

Learn more at www.3Dconnexion.com
# 3Dconnexion
A LOGITECH COMPANY

Outdated
Character Tools
Have Control
of You?

>> paul hyman

# STATE OF THE INDUSTRY:
# DIGITAL DISTRIBUTION

**SOME SAY IT WAS ITUNES IN 2001. OTHERS GIVE PROPS TO XBOX** Live, which launched a year later. Regardless what lit the spark under digital distribution, it's clear that it's no longer just PC gamers who are increasingly purchasing their software instantly without leaving the couch instead of trudging out to brick-and-mortar retailers. Now that Sony's PlayStation 3 has joined Microsoft's Xbox 360 in offering console games via download, the sky's the limit in terms of where digital distribution is headed.

While some industry observers say digital distribution will never outsell stores, others are far more enthusiastic about what downloadable software will achieve—and soon.

"I think it's going to be the dominant form of game distribution within the next two years for console games," predicts John Smedley, president of Sony Online Entertainment. "I'm saying specifically that, in that timeframe, more game revenue will be generated digitally than via discs."

Smedley's advice to game developers is to start building some games for digital distribution now. "Look at the sales of disc-based games and how many of them have sold 200,000 units or more," he says. "But Xbox Live Arcade has sold 200,000 units of UNO, of all things. Admittedly that's at $5, and that's not an insignificant factor. But eventually, volume is going to win out. Just like Pay-Per-View, it's the convenience of being able to sit in your home, click a button, and get a game when you want it that's going to win out."

Digital distribution is nothing new to PC gamers who have been downloading small casual games for years. But, in the last year or two, several factors have put digital distribution smack in the middle of every game marketer's launch plans. "What's happened is broadband penetration grew faster than anyone thought it would," says Doug Lombardi, director of marketing at Valve Software. "At the same time, online services like Napster and iTunes took the lead in the downloading of music files. Most of us in the game industry would say that our community is

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

much more technologically savvy than anyone in the music industry, yet somehow music jumped light years ahead of us on this one. So now we're all sort of playing catch-up, experimenting with ways to turn digital distribution into a moneymaking thing."

## TAKING IT TO THE CONSOLE

Of all the players in that game of catch-up, perhaps the most innovative was Microsoft, which four years ago foresaw the possibility of expanding the downloading of games to the console sector.

Greg Canessa, who was group manager of Microsoft's Xbox Live Arcade when interviewed for this story (and is now vice president of video game platforms at PopCap Games), notes that console developers are still in an adjustment phase. "While the PC had been connected for a very long time, we're just learning how to do that," he says of console developers. "These last two years we've created a reason for consumers to connect the console in their living room to an online service. We've provided the technical environment that allows developers to create games that plug into that environment. And we've come up with a critical mass of consumer features—like billing, authentication, a points system, profiles—all the great things that help users have a really simple and seamless download, trial, and then purchase experience. [It's] an iTunes-like experience of these small pick-up-and-play games that didn't exist before our Xbox Live service," Canessa says.

"We've removed 11 of the 12 steps involved in downloading casual games on the PC. You press one button, you get the completely free demo. If you like it, you press the same button again and—boom—you've bought it for $5 or $10 without having to download it again. It tends to be a very compelling value proposition for consumers," he says.

Clearly, Microsoft was doing something right. In July 2004, two years after Xbox Live launched, it signed on its one-millionth subscriber. With the release of HALO 2 later that year, the subscription base doubled within the next 12 months.

About 20 percent of the Xbox Live Arcade portfolio comprises casual games that are also available on other services, like Yahoo! and MSN Games. The rest are either retro classics or new pick-up-and-play games (such as the popular GEOMETRY WARS)



John Smedley, president of Sony Online Entertainment



The classic card game UNO has become one of the top sellers on Microsoft's Xbox Live Arcade.

that have been built by independent developers.

"A key tenet of our vision is to open up the console to independent game development as well as large publishers," notes Canessa. "We have an organization in the Arcade Group that is the publisher for everyone who doesn't have a major publisher. And so, for everyone from PopCap and GameHouse all the way down to the garage developers, we are their publisher, providing everything that a publisher provides other than financial funding: production assistance, design assistance, testing, localization, ratings, et cetera."

Canessa wouldn't discuss how revenue is shared with developers other than to say that terms are specific to each developer and publishing partner. "[Microsoft's] philosophy is that the developer-publishing partner keeps the majority of the

revenue," he says. "We're in business to provide a platform. It's their product. They're developing it, they're paying for it, and they should keep the majority of the revenue."

Nor would he be specific about how much money can be made in a development deal with Xbox Live. Canessa calls that opportunity "significant," especially since, in September 2006, Xbox Live announced it had surpassed the 12 million download mark, with several Xbox titles selling 300,000 units or more.

The average download-to-purchase conversion rate is 24 percent, significantly higher than the industry average of 1 percent. Canessa attributes that to the "impulse buy" price of $5 to $10 per game and the simple download-and-purchase experience.

Equally important to the developer, says Canessa, is that the bar for success is much lower with Xbox Live Arcade than for deals made elsewhere.



Doug Lombardi, director of marketing at Valve Software

"It's really about return on investment," he says. "When you've got a $20 million, full package product with a development cycle of two, three, even four years, publishers aren't willing to take risks with innovative games, which leads to lots of sequels because if the game doesn't sell millions of units, everyone is in trouble. With the games we have on Xbox Live Arcade, a small indie company can build one for a few hundred thousand dollars, which means that even if it only sells 50,000 units, it's profitable."

Still, there's tremendous competition among developers for the limited space on Xbox Live due to the fact that Microsoft manages its portfolio carefully, unlike most PC portals that tend to tout volume over quality.

"The PC space is flooded with lots of rip-offs and derivative content, a veritable wild, wild West of games," notes Canessa. "We have more of a record company approach: We believe it's our responsibility to wade through the clutter and pick out only the best games."

As a result, Xbox Live currently serves up only about 40 titles, adding roughly one a week, usually on its Wednesday release days.

If a developer is thinking about getting its game on Xbox Live, Canessa strongly recommends thinking outside the box and coming up with a title that is new, cool, innovative, and significantly different from what's already available.

"If you're coming to us with a chess game, chances are we already have one of those because we've got all the easy, low-hanging fruit already. I would focus on creative concepts for gameplay that haven't been tried before," he says. But innovation is just as important as marketability, and for Microsoft, a game with a strong community component has the best chances of being well received. "Concentrate on games around which we can build a community because the more people who play socially and competitively on the network, the better an experience it is for everybody. That may mean including competitive as well as

# Unreal® Technology News
## by Mark Rein, Epic Games, Inc.

*Canadian-born Mark Rein is Vice President of Epic Games based in Raleigh, North Carolina. Epic's Unreal Engine 3 has won Game Developer Magazine's Frontline Award for Best Game Engine for the past three years and Epic was recently awarded Best Studio at the Spike TV Video Game Awards. Since 1992 Mark has worked on Epic's licensing & publishing deals, business development, public relations, academic relations, marketing and business operations. Epic recently shipped Gears of War, which won Gamespot's overall Game of the Year and broke Xbox 360 sales records. Epic is currently working on Unreal Tournament 3 for publisher Midway.*

**Upcoming Epic Attended Events:**

**Game Developers Conference**
San Francisco, CA
March 5-9, 2007

**E3 Media and Business Summit**
Santa Monica, CA
July 11-13, 2007

**Develop Conference**
Brighton, UK
July 24-26, 2007

**Tokyo Game Show**
Tokyo, Japan
Sept 20-23, 2007

*Please email:
mrein@epicgames.com
for appointments.*

### WELCOME TO GDC!

It seems like just yesterday we were shutting down our booth at GDC but here we are back in San Francisco, California for the 2007 Game Developers Conference.

We invite you to come visit us at our "Exposuite" which is a fully-enclosed booth in a quiet section of the North Hall show floor. Our suite features a theater and private offices, where we can meet with current and prospective licensees to talk about engine technology, provide technical support or discuss business matters. In the theatre we provide twice-hourly demonstrations of the latest Unreal Engine technology and hold two press conferences complete with engine/partner technology demos and Q&A sessions.

Appointments are recommended. If you're reading this prior to GDC and wish to attend please email me (*mrein@epicgames.com*) with three possible day-time combinations and we'll let you know which works best. If you're already at the show just drop by and we'll do our best to accommodate you.

### UNREAL ENGINE SEMINAR AT GDC 2007

Unreal Engine licensees, qualified prospective licensees, and members of our Unreal Engine Integrated Partners Program are invited to sign up for our Unreal Engine 3 licensee full-day seminar at the Moscone Center in San Francisco on Tuesday March 6, 2007.

The agenda for the seminar includes separate tracks for technology creators (i.e. programmers) and content creators (artists, animators, level designers, audio, etc.) There will be plenty of Q&A time in the schedule and a chance for licensees to meet face-to-face with key Unreal Engine 3 development personnel.

Note: GDC passes are not required to attend the Unreal Engine Licensee Seminar. Lunch and coffee breaks will be served and attendance is free of charge.

Please head over to the following web page to register or receive more information: *www.unrealtechnology.com/seminar/*

# GameDevelopers
## Conference

### BOOTH LOCATION AND HOURS

We're located in the North Hall in booth ES-5718. Exposuite booths are open earlier than the normal show floor exhibits. Hours are Wednesday 9AM-6PM, Thursday 9AM-6PM and Friday 9AM-3:30PM.

### EPIC GDC SPEAKING SESSIONS

Epic will be speaking in the following GDC sessions:

Wed March 7, 2007

2:30pm - 3:30pm (Visual Arts) "Adapting Digital Cinematography Techniques for Game Development" – Jerry O'Flaherty, Art Director

Thu March 8, 2007

12:00pm - 1:00pm (Audio Track) "The Sound Design of Gears of War"– Mike Larson, Audio Director, Jamey Scott, Sound Designer, Daniel Vogel, Programmer.

12:00pm - 1:00pm (Business & Management Track) "PC Gaming in an Age of Connected Consoles"– Michael Capps, President.

5:30pm - 6:30pm (Game Design track) "Designing GEARS OF WAR: Iteration Wins"– Cliff Bleszinski, Lead Designer.

### ON A PERSONAL NOTE...

In the 15 years I've been at Epic we've won our fair share of awards but the Academy of Interactive Arts & Sciences award show in Las Vegas this year was something I never even dreamed I'd experience.

Gears of War was nominated in 10 AIAS award categories and we managed to win 8 awards including overall Game of the Year. What makes these awards special is that they are voted on by industry peers and presented in a flashy Hollywood-style award show with an audience filled with game industry celebrities. I hope all of you get a chance to experience something like this during your career. It was truly fantastic!

Last year was an outstanding year for video games and we would like to thank everyone who voted for Gears of War as well as congratulate all of the other winners and nominees.

For UE3 licensing inquiries email:
*licensing@epicgames.com*

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

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

# DIGITAL DISTRIBUTION

cooperative multiplayer play, perhaps supporting the Xbox Live Vision Camera with video chat and 'gesture play.'"

## NEW GAME IN TOWN

The good news for developers is that, since the November 2006 launch of Sony's PlayStation 3, Xbox Live isn't the only online service for video game consoles. Microsoft likes to say that Sony's PlayStation Network validates the Xbox Live digital distribution strategy—and Sony doesn't disagree.

"I do believe that Xbox Live did a really terrific job bringing awareness to a very mainstream audience of how great digitally distributed console games could be," comments Sony's Smedley, "which is why, from the get-go, digital distribution was a core part of our PlayStation 3 strategy—and why, in fact, our models all have hard drives."

The games, however, won't be limited to the small "pick-up-and-play" variety. "What separates [the PlayStation Network] from Xbox Live is that we will offer whatever kind of game people can think up," Smedley says. "If it's a great game, it can be up there."

For example, Sony is working on a full-blown MMO that will have no retail component and will only be available later this year via download. Sony has also built a subscription system into its service, which some day might offer episodic content or a line of arcade classics.

While Smedley believes small casual games will make up a popular portion of the PlayStation Network menu, there's no official limit on the physical size of each downloadable title. In the past, large downloads have proven problematic for gamers who may have no tolerance for them, so Sony is adding the capability to download "in the background," which also enables pre-loading.

"Pre-loading is something that we do routinely on the PC—as does Electronic Arts, Valve, and others—but it's something new to consoles," says Smedley. "It means you can pre-order a game that will be released in, say, a month, and start immediately downloading a bite-size piece of it every day. On the day the game launches, you've already got it all on your hard drive and it's good to go. You don't have to wait to play it."

Smedley believes the growing popularity of digital distribution will result in a proliferation of new game styles that are disc-less and, therefore, less expensive to build. While he claims not to be a big fan of episodic games, he admits they fit the digital distribution model perfectly. More importantly, he believes gamers will see a huge influx of more creative titles on which publishers would not have taken risks in the past.

"A great example is LUMINES LIVE! [currently on Xbox Live Arcade for the Xbox 360], which I think is a terrific puzzle game, but not one that would have made a lot of sense to sell in a box at retail because it is such a niche product," he notes. "But it's turned out to be hugely popular, the kind of innovative game you're more and more likely to see for download, like the poker game we're working on for the PlayStation 3 that, I think, is going to be massive. Everyone is going to love these new games—the gamers, the publishers, the developers—everyone, that is, except the big retailers that don't change with the times."

A few retailers have already chosen to go with the flow, like GameStop, for instance, which has launched its own digital distribution site called Download Now. Smedley calls the

company's adaptation "an amazing thing," explaining that "a year ago, I personally talked to management there and it was clear they had their heads buried in the sand. They were flat-out saying that they didn't think digital distribution would be big. Six months later, they were reminding us to be sure to release our games digitally and on disc simultaneously," he says. "You could tell right then that they'd finally gotten it."

As for Wal-Mart and other major retailers, Smedley predicts they will turn to digital as soon as they take notice of the massive audience that digital distribution brings to the table.

In the meantime, Sony is courting indie developers, creating opportunities for smaller teams with smaller budgets. But, like Microsoft, Sony won't discuss the royalties it pays out, other than to say that the numbers are similar to what Microsoft pays. Nor will it discuss how many games it is selling.

"All I'll say is that digital distribution is the next big thing in games, and because of that, I think it will help the growth of our overall industry," says Smedley—but only if players feel comfortable downloading games instead of buying them on disc, which is probably the biggest hurdle to growth.

"It was the same situation back in the pre-Amazon.com days," he says. "Buying online was no big deal for the early adopters but, for the masses, it took a while before they felt comfortable. The demographics might slow growth a tad before all the console gamers are downloading from Xbox Live and PlayStation 3, particularly because the audience is a young one, many without credit cards. But, once everyone makes that mental shift, you're going to be real surprised at how fast it grows."

## ON PC, WHERE IT ALL STARTED

Just about the same time Microsoft was unveiling its Xbox Live service, Valve Software was taking the wraps off Steam, the content delivery system that has become the standard for digital distribution of PC games, both casual and hardcore. In June, for example, HALF-LIFE 2: EPISODE ONE (the first of a trilogy of expansion episodes to HALF-LIFE 2) became available on Steam, and EPISODE TWO is scheduled to be released during 2007.
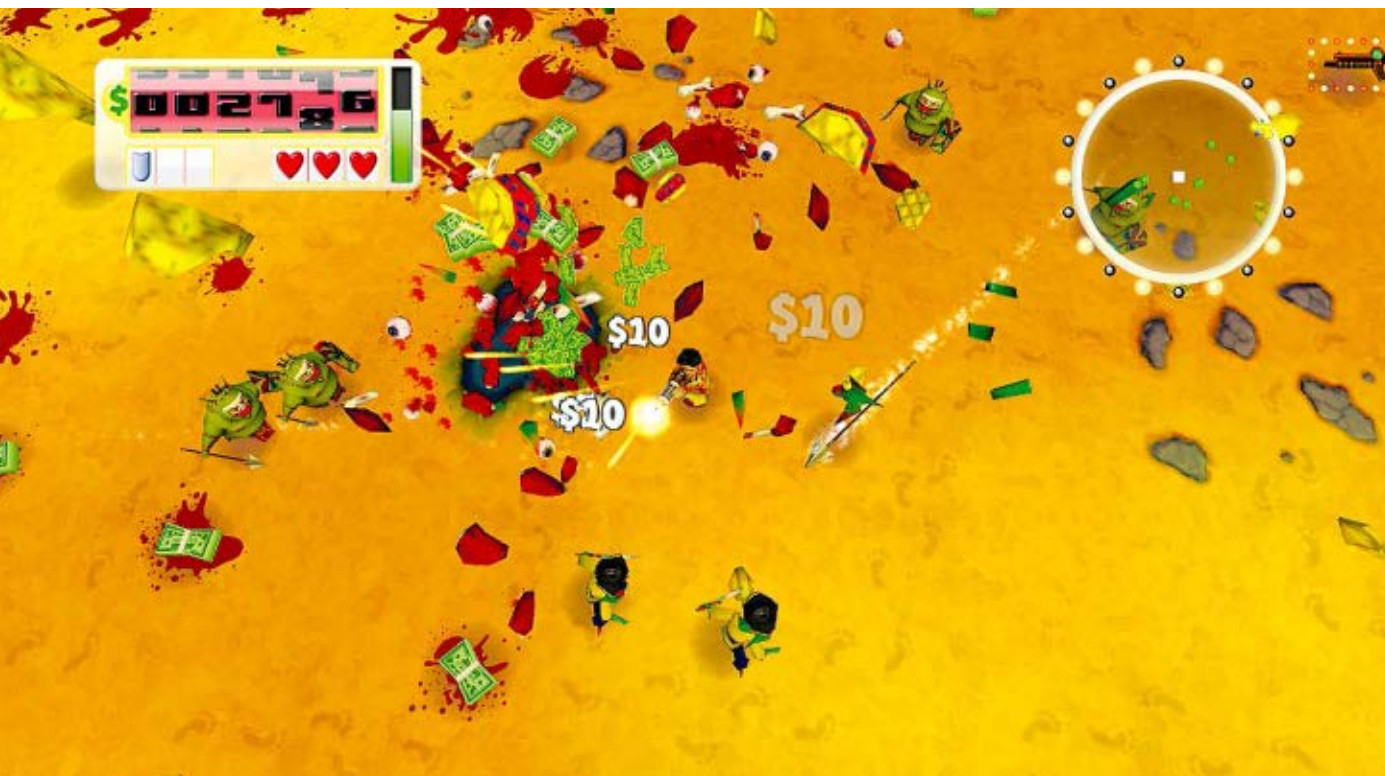
Steam is more than just a delivery system for Valve; it's a service that other developers can use to digitally distribute their titles directly to consumers. In August, for example, U.K.-based Introversion Software, the developer of the award-winning DARWINIA, signed on to deliver its new online strategy game DEFCON via Steam and other retail outlets.

"Steam has made Introversion a commercial success," says Thomas Arundel, commercial director of Introversion. "DEFCON is proof that the creation of exciting, unique games doesn't rely on years and years of development nor a 100-strong dev team."

**Greg Canessa, formerly group manager of Microsoft's Xbox Live Arcade, and current vice president of video game platforms at PopCap Games**

**Jamie Berger, vice president and general manager of consumer products at IGN Entertainment**

One advantage of using the sort of services that Steam provides is that it enables developers to stay in touch with their customers. "When you buy a game on disc from a retailer, you pay your money and there's sort of a handshake and an *adios*," says Valve's Lombardi. "But you can write Steam into your program so that it actually becomes part of your application and enables your customers to be updated automatically with new levels and maps and stuff. You can do all sorts of great promotions, like free weekends, that keep you in ongoing communication with the folks who buy your games."

In addition, Steam is equipped with anti-piracy protection, leaving Valve completely free of the sort of piracy that occurs when games are replicated on discs that sit around in warehouses, easily accessible to would-be pirates.

In two years, Steam's catalog has grown from a few different versions of Valve's popular HALF-LIFE first-person shooters to a portfolio of 104 games as diverse as the casual games of PopCap (BEJEWELED 2 DELUXE) to the god games of Firaxis (SID MEIER'S CIVILIZATION III).

"We try not to be like a traditional publisher that has to green-light a developer's game before we make it available," says Lombardi. "We're more like middleware. We sort of have this open-door policy. You know, 'Here's Steam. If you want to use it, here are the terms.'"

Valve's business model involves taking a percent of game sales, but Lombardi won't discuss what that percentage is or how many units Steam sells. "What I can tell you," he says, "is that it looks much more like a distribution fee than a publisher

relationship. The large majority of the dollars go to the person who brings us the game."

Lombardi's advice to a small development team is simply to try Steam. "There's a reason you don't see [retail] games for much under 20 bucks," he explains, "and that's because, after everybody who touches the game gets their cut, 20 bucks doesn't go very far. In order to have production values that are competitive these days, you need to sell your game for at least $29—or $40, really, after the platform fees, which means you need to build this really big game and invest $4 to $6 million minimum. That's awfully risky.

"I would argue that, instead, you can build an episode of a single-player game or a smaller version of a multiplayer game, and test it by putting it out on Steam. Sell it for $10 or $20. If it's good, the press will write about it, you'll sell some copies, and then you'll be in the driver's seat on the next deal you make."

## KEEPING IT HARDCORE

While casual games have become a staple of digital distribution, at two-year-old Direct2Drive, the only games available are hardcore PC titles. But, in order to differentiate itself from its competitors, the service likes to pepper its portfolio with some of the more creative hardcore games that conventional retailers won't touch.

"If you're looking for DINER DASH or the latest casual game from PopCap, you're not going to find it with us," observes Jamie Berger, vice president and general manager of consumer products at IGN Entertainment, the parent company of Direct2Drive. "We want gamers to know that when they come

HALF-LIFE 2: EPISODE ONE was released on Valve's Steam for digital download.

here, we've got a certain style of content, like the next F.E.A.R. expansion pack or MEDIEVAL II: TOTAL WAR."

What has changed since the service launched in September 2004 is that, two years ago, its catalog was filled with year-old games being sold at $19.99 each. Today, Direct2Drive sells the very latest titles, including pre-ordered games that are not yet available in stores.

"Our focus is very much on the core gamer, the early adopter, the young man who is 18 to 34," says Berger. "There are plenty of sites that service the casual gamer. We're not one of them."

Direct2Drive is very much interested in supporting independent developers who believe that digital distribution can help them find and expand their audience. "Digital distribution is probably the only way that guys with smaller teams and smaller budgets are going to make money these days," says Berger. "The good news is that it's expanding the opportunities for niche titles. There are some developers who say they're going to focus purely on digital distribution and maybe go episodic. Even though theoretically they may not sell as many units, they actually may be more profitable because they're eliminating all the expense of putting their games on disc and selling through retail."

For example, he cites SPACE EMPIRES 4, a strategy game developed by Malfador Machinations. "They're a small group of independent guys who have created a really cool game with a very tight and focused audience," says Berger. "Digital distribution has been a great way for them to expand that audience on a shoestring."

Direct2Drive's business model is strictly e-commerce: no subscriptions, no try-before-you-buy, no sponsorships, no advertising. While Berger wouldn't discuss the terms of the deals he signs with developers or publishers, he notes that, in
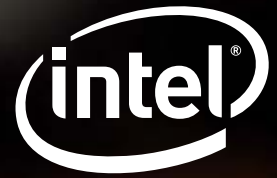
the past year, the company's unit sales have grown approximately 300 percent. But the numbers are just 5 to 10 percent of what retail sales are on the same games.

"What developers need to know is that digital distribution opens up a completely new business model for them to explore. If someone tells them that one model is always better than the other, they're lying," says Berger. "Developers now have a lot of choices in front of them about how they want to monetize their work. And what we and the other services are doing is increasing their opportunities to come to market in a way that, frankly, didn't exist three or four years ago—or was very difficult to access."

While Sony's Smedley sees digital distribution overtaking retail sales for console games in two years, Berger's beliefs are far more conservative when it comes to PC games.

"I'm not one of those guys who says physical distribution is going away and digital distribution is going to take over the world," he elaborates. "Over the next five to eight years, you're going to see just what you're seeing now: that digital distribution will be a completely acceptable alternate way of acquiring content, just as acceptable as walking into a store and buying it physically. And then it's just a matter of what the consumers choose to be the best way for them."

At Valve, Lombardi doesn't disagree. "Developers had been sitting on the sidelines, reluctant to take the plunge into digital distribution," he says. "But now, they seem to have decided that this isn't one of those goofy technologies that comes around and disappears. Lord knows there have been enough of those. Developers are saying, 'Okay, this is real. Maybe we ought to give it a shot because this could be big.' How big? I think it's going to take us a few more years before we figure that one out." ✖

# 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™ and PC, and **morpheme:connect:** a highly-customizable 3D authoring application.
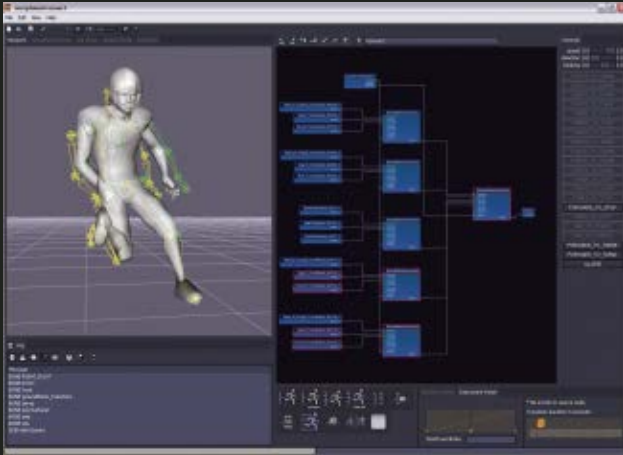
**morpheme** gives animators and developers unprecedented control over the look and feel of their animations in-game: blends, transitions, compression, etc. can all be previewed and modified graphically in **morpheme:connect** and live on the target platform.

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

For more information, visit www.naturalmotion.com

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

## blend tree
Advanced graphical tools for building complex blend trees. Real-time visualization of animation source contribution through node highlighting
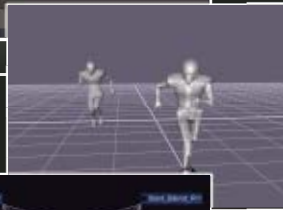
## timeline
Graphical mark up of animation data to add one-shot and duration events, for highlighting footfalls, sound effects, etc.

## blending
Graphical control of transition blending between states in the transition graph

## node palette
Advanced blend notes (time-warp) for dragging and dropping into transition network. Fully customizable node types through C++ and scripting
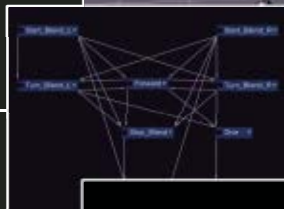
## multiple characters
Visualization of multiple runtime characters in morpheme:connect for easy authoring and analysis of character interaction

## animation browser
Easy browsing and selection (drag & drop) of source animation. Animation list is automatically updated to reflect changed source files (after editing in Maya)

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

## transition requests
Exposure of custom transition messages. In-tool emulation of interaction between morpheme:runtime and game AI system
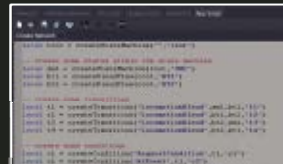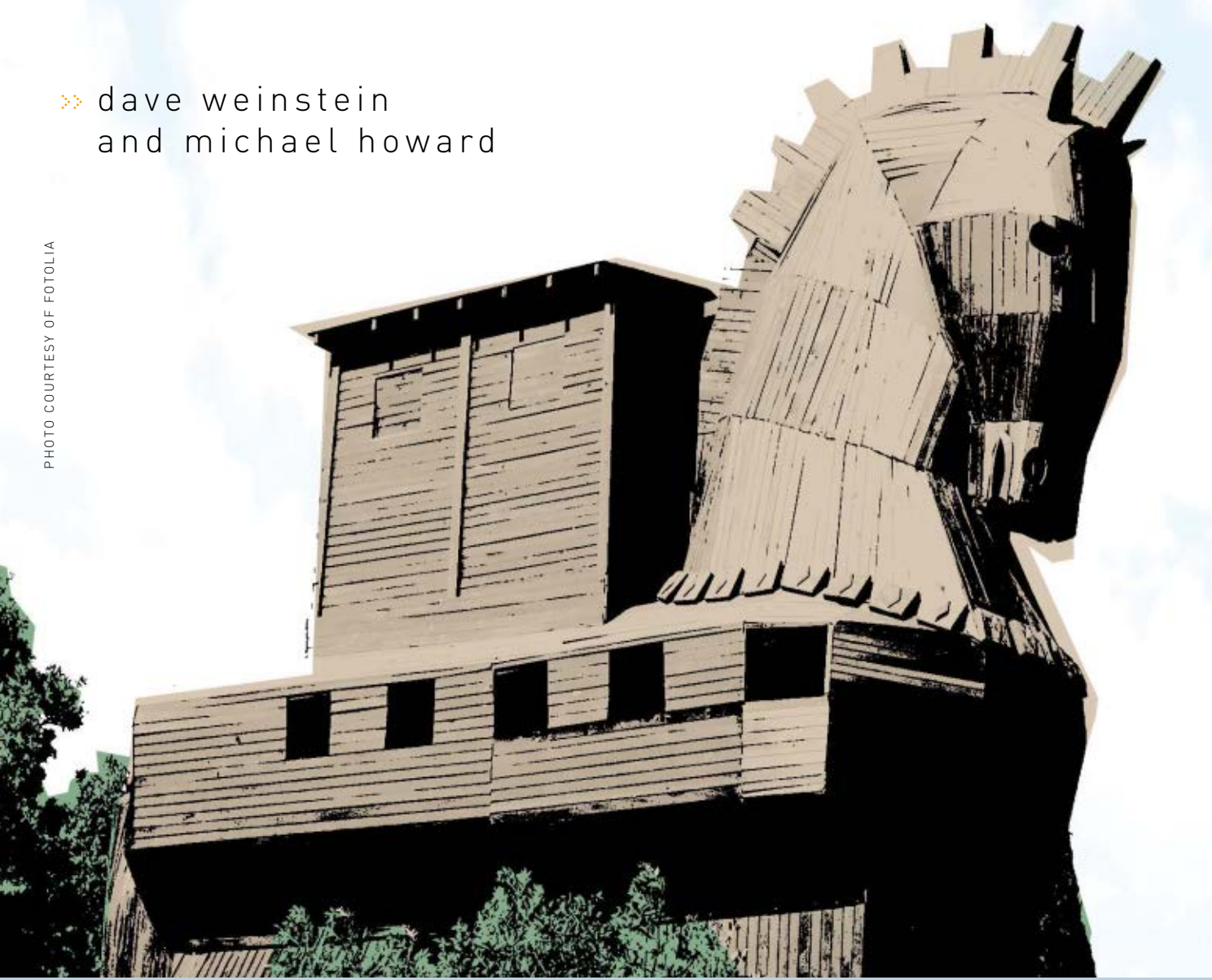
## control parameters
Exposure of custom high-level controls for entire animation system. Real-time manipulation through sliders or game pad controller

**naturalmotion**

>> dave weinstein
and michael howard

# INSECURE PLAYERS
## HOW SLOPPY CODE LEAVES GAMERS EXPOSED

**WHEN GAME DEVELOPERS THINK ABOUT** security, what usually comes to mind is either copy protection or cheating. They seldom, if ever, worry about miscreants or more ominous criminals using games to take control of the owner's machine. This scenario is most common in online games (largely for PC) that provide multiplayer modes or allow user-created content in-game. Unfortunately, many developers don't expend serious thought on the issue, as it's difficult to prove before a game is launched. The risk is greatest for online games or games that allow user-created content in-game.

This article examines other ways and reasons game developers should think about matters of security. It's time to think about whose assets are really at stake.

## SECURITY FOR WHOM?
Copy protection—to the degree that it works— benefits the publisher, and to a lesser extent, the developer. It's at best neutral and often unpleasant for the legitimate customer. And while mitigating cheating is certainly good for the customer, do we really think players would rather be protected against other players who move too fast or shoot too well, than criminals trying to take over their computers?

**DAVE WEINSTEIN** *left the game industry after 11 years and six platforms to write security tools as part of the Microsoft Secure Windows Initiative. These days, he still occasionally works on (or works over) Microsoft published games, as part of their final security review.*

**MICHAEL HOWARD** *is a senior security programmer in the Security Engineering group at Microsoft. He is an architect of the Security Development Lifecycle.*

*Send comments about this article to dweinstein@gdmag.com.*

# INSECURE PLAYERS

The most prevalent and dangerous bug type is the buffer overflow. Most games are written in C or C++ mainly for performance reasons, but C was designed as a replacement for assembly language, and as such gives the developer direct access to the metal. A simple coding error managing a buffer copied from an untrusted source can lead to the attacker running the code of his bidding.

## TRUST AND TRUST BOUNDARIES

Security often revolves around two issues: misplaced trust and protecting secrets. In this discussion, we focus almost exclusively on trust. Programming errors aside, software developers primarily get into trouble when they start making flawed assumptions about what they can trust. They start by trusting the integrity of data that comes from a source completely outside their control.

Security professionals talk about "trust boundaries," an interface across which you have to assume data will be coming from a hostile source. For game developers, the primary trust boundaries are those that occur when we interact with other machines and when we deal with content created by other users.

When users run a game, they don't really have clear knowledge of what that game is doing to their computers. If the game requires administrator privileges (shame on you if it does), the software could theoretically do anything to their machines. But they've made the decision, often supported by corporate branding, to trust the authors not to act malevolently.

If the user trusts the game, then she or he certainly will trust the data files that shipped with the game. Anything the authors could do with malformed data they could do with explicitly malevolent code. In other words, there is no additional risk to loading content written by the same authors who wrote the actual game.

What players probably don't realize is that the code that loads those campaigns is most likely very brittle. It assumes that the values are within valid ranges and most likely, it doesn't do a lot of testing against malformed data, especially if the data files themselves are generated by a content creation tool. After all, if the editor only creates valid files, then the game doesn't need to worry about it—that is, assuming that the game only loads official files (and strictly speaking, it should also use a cryptographic signature to confirm that it is loading official files).

But what if the game is designed to allow user-created content? We have now crossed authors, and even though a player might be willing to download new scripts, maps, and skins from someone on the internet, she or he certainly doesn't expect that loading the new skins is equivalent to running a program on the machine. If the game doesn't handle malformed data (and again, most software that has not been specifically hardened will almost invariably

exhibit problems with malformed data), then that image, sound, or script may be "code equivalent."

For developers publishing through Microsoft, hearing that a data format has been deemed "code equivalent" by security auditors is unlikely to be beneficial to your ship schedule.

When looking at network connections, we again consider issues of trust. Players can trust any server run by the developer, because again, if distributing malicious code were the goal, it would be easier for the developers to simply embed the code in the executable that the players are already running. But as a player, we cannot trust the other players, nor can they trust us, and to be perfectly honest, the servers run by the developer shouldn't trust any of us anyway. To make matters worse, it's a decidedly non-trivial task to determine if players are talking to the developer's machine in the first place. (See the sidebar "Network Source Information.")

It's one thing to naively trust the data and allow people to claim abilities they shouldn't have in the game. And it's this level of trust that game developers have traditionally focused on. Something far worse happens when we developers trust the data, especially if the result is a perpetrator who is enabled to get remote execution of code and take control of a player's machine.

## CODE DEFECTS, VULNERABILITIES, AND EXPLOITS

For the scope of this article, we are concerned with code defects that occur when the code makes assumptions about un-validated data that are incorrect. One of the classic examples is assuming that something we expect to be a string is null-terminated or contains some other format character that the code looks for.

This code becomes vulnerable when there is a route by which an attacker can provide it with malicious data. In the security world, this is referred to as "attack surface" (or sometimes just "surface").

The vulnerability becomes an exploit when a sufficiently skilled and motivated attacker is able to develop a specific payload that results in actual remote execution of code. While

## network source information

```
int recvfrom(
SOCKET s,
char* buf,
int len,
int flags,
struct sockaddr* from,
int* fromlen
);
```

It is obvious that the data received (shown at left) came from an untrusted source. What may not be obvious is that the source address is also inherently untrusted.

Even though it's separated out in the function call, the source sockaddr came from the same source as the data itself, and cannot be considered trustworthy.

there are numerous resources on how to generate an exploit, as a developer, the details aren't particularly important. What's important is that if an attacker can cause the application to crash, she or he is already controlling the program execution.

The best case is a denial of service attack, whereas the worst case is that your customer's machine (or in the case of an MMO, possibly your central servers) is controlled by the attacker. The mantra in our group at Microsoft is, "A denial of service bug is a remote code execution bug once you've spent a little more time working on an exploit."

## PARSERS AND THE EVILS OF DATA

From a security perspective, data is almost inherently evil. It controls program execution flow even in the best of cases; in the worst of cases, it controls program flow in ways the developer never intended.

From an attacker's perspective, attack surface is most often found in parsing code. Few programmers enjoy writing parsers, and the parsing code often assumes that the data coming in is valid. Even when third-party libraries are used for part of the parsing (for example, an XML parser), the code that extracts the content from the format still often contains unchecked assumptions as to what it will find.

When auditing game code, we start with the UDP `recvfrom()` function (or equivalent network functionality) and work backward. We know that we can inject data into the network stack. The questions then become: Who parses that data, and what assumptions are they making about it?

While this is easier to do with the source code, it's also quite possible to do with only the binary and a good disassembler.

## LISTING 1 arithmetic underflow

```
int dataSize = packet->Size - HEADERSIZE;

if( dataSize < maxBuffer )
{
        memcpy( destBuffer, packet->Buffer, dataSize );
}
```

You should also analyze whether any code that validates the data is correct to make sure the code has been written correctly and defensively.

## THE BASICS OF BUFFER OVERRUNS

The classic attack is a buffer overrun, in which the data exceeds the size of an internal buffer but the application copies it anyway. Stack buffer overruns have traditionally been particularly dangerous because they are highly predictable and can allow the attacker to overwrite a function return address and take control of the code execution when the function returns.

For years, it was assumed that heap overflows were, in general, not exploitable. That changed when attackers started using heap overflows to overwrite the heap control information and use those structures to control the values of arbitrary memory locations when the memory was freed. Because game developers often write their own memory management routines, they are more vulnerable to these sorts of attacks because they are not able to take advantage of the hardening of the system heap that has been done as part of securing operating systems.

Even if the heap and the stack were hardened, application-specific mistakes can lead to trivial exploitation. Consider the following data structure:

```
VERTEX    Vertices[MAX_VERTICES];
PARTICLE_FUNCTION  Effect;
```

If we can overflow the vertex list, we can set an arbitrary function pointer, and that means we can take over the code execution by loading a malicious or malformed special effect.

## INTEGER OVERFLOW AND UNDERFLOW (ADVANCED BUFFER OVERRUNS)

The first step in protecting against a buffer overflow is to check the length of the data before any data is copied. As it turns out, this isn't quite as easy as it seems.

Sizes (in C and C++) are generally unsigned values. However, programmers have been in the habit of using signed variables, and using these values interchangeably. Consider Listing 1. If we craft a malformed packet in which we claim that our size is less than the HEADERSIZE constant, we get an integer underflow, and we will sail through our size check and into an extremely large memcpy. This will eventually result in an access violation, but if the attacker has been able to arrange things correctly, that access violation may itself be the mechanism for taking control of code execution.

# INSECURE PLAYERS

We have similar problems when we overflow an integer. There are two classic ways that this happens. The first occurs whenever we get a count of items. If the validation code simply multiplies the count by the item size, we can roll the integer over and end up validating or allocating an extremely small value, before using a large one.

The second occurs primarily in memory allocation. Memory allocation is almost invariably done on some multi-byte boundary (often 4 bytes, often more than that). The allocation size is therefore generally rounded up to the next boundary, with code that looks something like:

```
allocatedSize = ((requestedSize+BOUNDARY) & ~BOUNDARY);
```

The problem occurs when our requested size is within BOUNDARY of an integer limit. When that happens, allocatedSize becomes zero, and for historical reasons, zero-byte memory allocation requests are generally considered valid by the OS.

If you treat zero-byte allocation requests as failure cases, it will help you to find this particular attack pattern very easily. You may be wondering where the security bug is; the security bug is when you call memcpy:

```
memcpy(destBuffer,srcBuffer,allocatedSize);
```

The last argument is not zero, but you have allocated zero bytes.

The Assert macro is a powerful tool. It makes it easier to detect bugs during development and allows for cases where the code breaks assumptions to be found more readily. However, as a tool for validating data, it works very poorly.

Unlike code, data can be provided by outside sources at runtime. If the assert code is left compiled in for release, we can trivially crash the game server. While this is a definite improvement over taking control of the server, it's still going to be a serious problem for the game and the customers.

If the assert code is left out of the release, we have an even bigger problem. If assert was used to validate the data, then we have untrusted data that is being handled with no validation code, and that makes it far more likely that we can find something to exploit.

## FUZZING THROUGH DATA

There are some significant problems with relying on code auditing to find security issues. First, the skills required to identify security bugs as part of a code audit are not common in the game industry—and people who have them aren't cheap labor. Second, even a skilled auditor will have both false negatives (failure to find the problem) and false positives (a problem that's seen but is not actually exploitable). This lends itself to pushback from the development team, who will often insist that exploitable problems aren't real, and who might demand "proof" before correcting the issue. Finally, even assuming that you have found someone with expertise and received a solid and careful audit, as new code is written or final bugs are fixed, it's entirely possible that new security flaws are introduced.

Fortunately, an easier and far cheaper solution exists. Developed in 1989 by Professor Barton Miller and his team at the University of Wisconsin, fuzz testing, or "fuzzing," is the deliberate use of malformed data to test applications. It can range from pure random noise to a deep understanding of the actual underlying formats, but even in the simplest case, it's frighteningly effective at finding exploitable security issues.

While there are a number of available fuzzers, the simplest can be done inside code as part of development. Simply invert random bits as the data is read in (whether from a file or the network) and monitor the game under the debugger as it runs. The cases you are looking for are:

1. access violation exception,
2. CPU spike or infinite loop, and
3. memory allocation spike.

The latter two are denial-of-service attacks, although a large allocation spike could indicate an integer-overflow issue. An access violation is at best a denial-of-service (since it will crash the application), and at worst allows for remote execution of code, compromising the machine.

In our discussion of fuzzing, you may have noted that a simple checksum would solve the problem. While checksums are useful in detecting accidental data corruption, they're useless in stopping an actual attacker. They are mitigation for fuzzing. They will not stop or even appreciably slow an attacker.

## GAMES FOR TARGET

Software security issues aren't going to go away, and from an attacker's perspective, games are an attractive target. They are often online; many of them still require administrator privileges to run; and the increased use of middleware in the game industry means that attacks found against one game may be easily translated to attacks on another.

All security issues revolve around failures of either code or design. Even the best engineering won't protect against a designed feature such as "Server can download and execute an arbitrary binary to the client machine," and even the most paranoid designer will not protect against sloppy implementation and programming.

In the 1990s, game developers began asking, "What would a malicious player do with this feature?" especially when designing multiplayer games. But malice has progressed beyond causing grief and cheating to now encompass real threats to the integrity of your customers' machines. Worse, it uses many of the features the users desire, such as multiplayer gameplay and user created content, as an attack vector.

There's really nothing that separates games from any other software that's attacked on the internet. The very fact that games are often networked and games players have items of value held within the virtual bounds of the game makes games attractive targets. Games often also use UDP-based protocols, which are much more prone to attack than TCP-based protocols. And of course, most games are written in C or C++.

It's critical that developers focus on the design flaws and code defects, including buffer and integer overflows, that make their products vulnerable to attacks. ✕

# IN THIS GAME, QUEBEC IS THE RIGHT MOVE.//:

**://A**BLOCKBUSTER SUCCESS

In Quebec, you'll find top talent and a creative milieu that's unmatched in the world of gaming.

You'll also discover what developers, publishers and major studios such as Ubisoft and EA Montreal simply love about this location: low operating costs and a winning environment.

Contact Invest Quebec and find out why Quebec wins hands-down.

**IQ Invest**issement **Québec**

Invest Quebec
393 Saint-Jacques Street, Suite 500
Montreal, Quebec H2Y 1N9
Canada
1-866-870-0437

Los Angeles
(310) 209-3332

Chicago
(312) 645-0398

New York
(212) 843-0981

London
+44 20 7766 5931

Munich
+49 89 2420 8760

Paris
+33 (0)1 42 99 61 50

Beijing
(86-10) 6465 1520 ext.129

Tokyo
81-3 5733-4001

www.investinquebec.biz
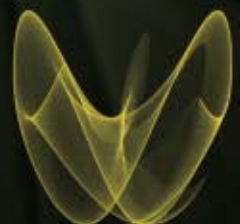
# SAM &

*A Journey*

**SAM & MAX ARE A DOG AND RABBITY-**
thing crime-fighting team, created
by Steve Purcell, who got their start
in the indie comic book scene. Sam
is the straight-laced gumshoe with a
sardonic wit and unparalleled
verbosity. Max is the slightly-
psychotic sidekick who's happiest
when he's causing any sort of
mayhem. They carry big guns, can
travel through space and time, and
have a special affinity for chintzy
American pop culture.

At Telltale Games, we are deep in
the development of the
anthropomorphic duo's episodic
gaming debut, Sam & Max: Season
One. The season launched last fall,
and as of the printing of this article,
three episodes have been released
(one per month, give or take). Due to
the project's episodic schedule,
we're coming at this postmortem in
an unusual way—from the middle.
We dissected the project at the
halfway point, with three episodes in
the bag and two left to go.

There's been a lot of buzz recently
about episodic gaming, but not
everyone agrees on what it means. It
seems like everything from yearly
mission packs to simple

*Written by:*

**DAVID BOGAN**, *art director*
**KEVIN BRUNER**, *CTO*
**DAN CONNORS**, *CEO*
**BRENDAN FERGUSON**, *designer*
**DAVE GROSSMAN**, *senior designer*

**HEATHER LOGAS**, *designer*
**EMILY MORGANTI**, *marketing coordinator*
**RANDY TUDOR**, *lead programmer*
**STEVE PURCELL**, *Sam & Max creator*
*Email comments to editors@gdmag.com*

# MAX

## *Through Episodic Gaming*

## GAME DATA

**DEVELOPER**
Telltale Games

**PUBLISHER**
Telltale Games and GameTap

**RELEASE DATES**
October 17, 2006 (Episode One), December 20, 2006 (Episode Two), January 25, 2007 (Episode Three), February 22, 2007 (Episode Four), March 29, 2007 (Episode Five), April 26, 2007 (Episode Six)

**PLATFORM**
Windows XP/Vista

**NUMBER OF DEVELOPERS**
14

**LENGTH OF DEVELOPMENT**
14 months

**PROJECT SIZE**
Approximately 240,000 lines of C++ code and 10,000 lines of scripting code per episode

**TECHNOLOGY USED**
Proprietary 3D engine, Maya, Photoshop

downloadable multiplayer levels are claiming to be "episodic," but most of these don't fit Telltale's definition of the term. For us, episodic games are complete game experiences, regularly and reliably scheduled, that feature characters and plot elements that are tied. In other words, they're the game equivalent of a TV series. From our standpoint, SAM & MAX: SEASON ONE is the first gaming project to fit this definition.

## WHAT WENT RIGHT

**1 RELEASE SCHEDULE.** SAM & MAX: SEASON ONE isn't the first episodic gaming project, but it's the first to attempt a once-a-month release schedule. With projects such as HALF-LIFE, SIN EPISODES, and our own BONE games that had episodes spaced out over six months or longer, episodic gaming was getting off to a slow start. Without a set release schedule, players didn't know when to expect the next part of the story. This led to a lot of complaints about episodic games being a rip-off, as well as uncertainty over whether episodic projects would continue on their promised path or die out after only one or two installments.

With SAM & MAX, we announced long before the premiere that SEASON ONE would have six episodes, released monthly. Establishing this schedule set player expectations and also helped us know how to tackle the project. We set up our production schedule based on this release plan, giving ourselves a few months up front to develop the initial characters and environments and get a head start on the first episode. The production schedule is staggered, which means that like on a TV series, we're working on later episodes as the earlier ones come out.

We have run into some scheduling problems, but from a release standpoint the once-a-month schedule has been very successful. It allowed us to start seeing revenue sooner than we would if we had first produced the entire series and then released it one episode at a time. The schedule also puts us in a position to follow what people are saying about the series and make changes as needed. We've been able to incorporate some player feedback from earlier episodes into later ones, and have even worked in some gags and plot twists based on comments made on our forums. Since episodes are coming out monthly, players stay involved and the series stays top-of-mind in a way it wouldn't if we only released an episode every six months.

Convincing gamers that episodic gaming is a good idea has been a big hurdle. We certainly see a difference between episodes, sequels, and expansion packs, but these distinctions are not always made in the industry, which can make it very difficult for gamers to understand what Telltale is trying to do. Fortunately, with the release of our first three SAM & MAX episodes, people are finally understanding what episodic gaming can be—and they like it. Every time we come out with a new episode, we see comments all over the internet about how SAM & MAX is episodic gaming "done right." This feedback is incredibly encouraging.

**2 THE RIGHT PARTNER.** We knew that teaming up with a partner would allow us to share the risk of trying this new experiment and put us in a position to achieve a lot more than if we were flying solo. The only question was who the right partner would be. Retaining creative control over SAM & MAX: SEASON ONE is important to us, and we had to do some looking around to find the right fit. Not everyone we spoke to understood what would be best for the license. We met some resistance over the episodic model and the type of game we wanted to make. One company we spoke to even wanted to turn SAM & MAX into a racing game.

Then we hooked up with GameTap, Turner Broadcasting's game subscription service. GameTap was a good fit for SAM & MAX for many reasons, not the least of which is that key members of GameTap's management team are SAM & MAX fans. GameTap shared our vision of a monthly episodic release schedule. They also understood the type of gameplay we envisioned in a SAM & MAX game, which is incredibly important in an industry where many potential partners would have asked us to turn the game into something with more perceived mainstream appeal.

GameTap and Telltale worked out a deal that has been beneficial to both companies. Each episode runs exclusively for two weeks on GameTap, after which we sell the episodes from the Telltale web site. From a marketing standpoint, GameTap has been able to introduce SAM & MAX to an audience Telltale would not have been able to access, and they have given the game exposure through magazine and TV ads that we wouldn't have been able to afford. GameTap has also been nothing but supportive regarding our creative choices—not too surprising considering the Turner heritage, which has fostered Cartoon Network and Adult Swim.

**3 DEVELOPMENT TOOLS.** Even before we started work on SAM & MAX, Telltale put a major focus on developing tools that would enable us to release episodic games on a tight schedule. The tools are designed to spread out the work of building a game, rather than putting the majority of the implementation on the programmers' plates. The tools are set up so that anyone in our office, from the lead programmer down to the marketing intern, can use them. Even our soundtrack composer can do his own implementation of cutscene music and sound effects straight into the project, and he's an outside contractor with no programming background.

Once a base layer is wired up by the programmers, our writers, animators, and choreographers can go into the game and build the content using tools specially designed for their needs. The programmers can then focus on high-level scripting, bug fixes, and base programming for future episodes, while the rest of the team builds content.

The tools facilitate this group effort by allowing multiple people to work on different parts of the game simultaneously, and to get their changes into the game quickly. For example, when an animation is finished, there's no lengthy process required to get the animation into the game; within minutes, an animation can be exported from Maya into the game engine and a choreographer can start editing that animation into a

cutscene. When writers edit dialogue files or programmers make changes to game logic, the changes appear in the game as soon as they've been checked in and can be accessed immediately by anyone else working on the episode. The only time we need to compile is when we're on the verge of a release.

These tools have been in development since we started the company, and we built several games with them before embarking on SAM & MAX: SEASON ONE. Only when we felt confident that the tools were ready to handle the once-a-month schedule did we start work on SAM & MAX.

4 **PRICING STRUCTURE.** One of the loudest arguments against the episodic model is that it's a way to rip off gamers by charging full price for only a small piece of a game. When we were deciding how to price SAM & MAX: SEASON ONE, this argument was front and center in our minds. We had no intention of ripping off our customers, but we didn't want to devalue our work either, and we spent many meetings talking through the best way to approach SEASON ONE pricing.

Ultimately we decided on $8.95 for one episode, or $34.95 for the entire season of six episodes. The per-episode price is slightly less than the cost of a movie ticket, for a slightly longer experience. The full season price is on par with retail PC games, and buying the season provides a discount of about $20 over buying each episode individually. This pricing structure, which rewards the customer for buying up front, has already proven successful for downloadable content through outlets such as the iTunes music store, but it hadn't yet been tried for an episodic game. Knowing that owning a physical product is a big deal for some players who are still on the fence about digital distribution, we threw in the option for season customers to get a CD version once SAM & MAX: SEASON ONE is finished for only the additional cost of shipping.

As much as we discussed it beforehand, we really didn't know how the public would react to the prices. Soon after we made the announcement, we started seeing posts on forums and blogs from people who thought the prices were reasonable, which was a big relief. As we had hoped, the SEASON ONE offer and the option to get a CD version convinced a lot of people to buy up-front,

The Telltale Games team.

rather than waiting until the end of the run. Of course, one of the reasons this is good is because we collect more money sooner, but it has also been beneficial from a planning perspective. Because so many of our customers have bought the entire season, we have a much better idea of how big our customer base is than we would if we were only selling one episode at a time. This advance knowledge allows us to provide better customer support on release day. It also helps us focus our marketing efforts as we move into the second half of the season, and gives us an idea of how many CDs we'll need to produce come spring.

5 **THE RIGHT LICENSE.** Choosing Sam & Max's universe as the one to use for our leap into true seasonal episodic gaming was an easy decision to make, and none of us are surprised that it turned out to be a good call. Since our plan was to build a series of short games released at regular intervals, it was important to work with a license that would lend itself well to this tempo, and Sam & Max is perfectly suited for telling short stories. Virtually anything can happen in a Sam & Max story, and we enjoy that lack of restriction come design time. Sam & Max also has a distinctive sense of humor, which gives us some freedom. This is important when making so many games in such a short span of time. We can take a lot of chances with other aspects of an episode, so long as we make the game funny.

Sam & Max is a versatile license with many fine qualities, not the least of which is its creator, Steve Purcell. Steve is a top-notch artist and writer who is genuinely interested and involved in what we do with the games. (The fact that he used to work at a game company probably has something to do with this.) He gives us feedback on characters, dialogue, voice casting, and visual design, and we've been able to get him in for concept and story meetings at the beginning of each episode. Steve has been outrageously helpful.

SAM & MAX: SEASON ONE is the first episodic game based on the series, but it's not the first SAM & MAX game. LucasArts released a graphic adventure based on the license in 1993, SAM & MAX HIT THE ROAD. In the 13 years since, a lot of expectation had built up in the minds of the fans. This conceivably could have been cause for trepidation, but the Sam & Max characters made a lot of issues easier on our end. Response to the release of the first three episodes tells us we did right by the license, and we know it did right by us.

## WHAT WENT WRONG

1 **RECONCILING SCHEDULES AND DESIGN SCOPE.** When we first defined the scope of the project, we decided on six episodes that would each have approximately 60 to 90 minutes of gameplay. The schedule for the entire project was based on this estimate. What sounded like a great idea on paper became unsatisfactory once the design team sat down and started designing. It soon became apparent that 60 to 90 minutes of gameplay would not be enough. In the interest of quality and player experience, the designers made a conscious decision to design episodes that would be closer to two or three hours in length, despite the fact that it was a larger task than the schedule had been created for. The design scope grew, but the schedule couldn't grow to match it.

Since design happens before everything else, there was a domino effect. Artists couldn't build assets until they were

designed. Programmers were waiting for art in order to wire things up. The polish phase couldn't go far without the recorded voice, which was waiting on the script. Everyone on the team was under pressure to return results faster than anyone was comfortable doing.

We couldn't change the schedule, so we dealt with the unanticipated workload by looking for ways to buy extra time for the designers wherever possible. We reorganized art production to be more efficient and eventually brought in additional writers to make up time during the scriptwriting process. Fortunately, the schedules were so short to begin with that it wasn't possible to get all that far behind, but it would have been nice to have a larger window of design time up front before the games started rolling out.

This sort of episodic project was new to all of us, and no one could have anticipated beforehand that we'd end up doubling the length of the episodes. Now that we do know this, we'll go into our next episodic project better equipped to come up with a realistic schedule.

2 **VOICE RECORDING DRAMA.** An especially big scheduling challenge for an episodic game is the beloved, yet dreaded, voice recording. We don't have the luxury of polishing a script over the course of many months, meaning we have to record the dialogue for an episode soon after we finish writing it.

Thus we had to find actors who could come back to the studio for several recording sessions, which made finding people to fill the big shoes of Sam and Max a particular challenge. (Okay, Sam and Max don't actually wear shoes, but you know what we mean.) It didn't help that virtually everyone on the planet had a strong opinion about how Sam and Max should sound. Luckily for us, we managed to find a great crew of voice talent whose quality work has quieted the ubiquitous critics.

Unluckily for us, we managed to lose one of our great crew of voice talent who was suddenly unavailable to record the second episode. We were right in the middle of production, with no time for delay, so we found a replacement who was available immediately. We felt, though, that the new voice didn't sound quite consistent enough with the previous one, so we recorded yet again with a brand new actor, and then had to deal with the issues of re-choreographing many of the animated dialogue scenes. Though it proved costly, it was well worth the effort as the reaction to the new voice has been positive.

There was another experiment that didn't go as planned. We tried to record the second and third episodes simultaneously. This meant that EPISODE THREE was being recorded before environments or basic wiring was complete, which made it hard to finalize dialogue that was very dependent on the environment. Similarly, EPISODE TWO had to ship before the next recording session, leaving us no opportunity to record last-minute missing lines. Ultimately, we decided it was worth recording each episode individually, which definitely keeps us on our toes, but gets us that much closer to our dream of making an undeniably perfect game.



SAM & MAX has an almost Vaudevillian sense of humor.



Concept art for a character in SAM & MAX.

3 **MANAGING ASSETS ACROSS EPISODES.** With an episodic game, sharing assets is a critical part of the plan. Reusing models, textures, and environments saves time and money on the art and production sides, and working with a recurring cast of characters and sets makes sense from a story standpoint. The difficulty is trying to coordinate these shared assets across episodes.

Each of the six episodes stands alone as a self-contained project, and it can be hard to maintain consistency as changes are made to shared assets. The fact that we don't have a formal production database doesn't help.

Deciding which assets should be shared means sorting through thousands of resources, including models, scenes, animations, sounds, and lines of dialogue. Sharing items in the database is a tedious task at best, especially since it's not always clear which resources should be shared from episode to episode. Sharing is also prone to errors. We had a lot of problems with files being changed in one episode and inadvertently affecting another.

There comes a point in the production of each of the episodes when we need to branch all shared resources and isolate the episode that's closest to shipping. Branching files early can prevent unintentional changes to the same files in another episode, but branching too early can have negative repercussions. After we branched EPISODE ONE assets, for instance, a lot of fixes and last-minute changes were made, which then had to be brought in to EPISODE TWO manually.

We also had some problems with sharing language resources between episodes that had overlapping schedules. Early on, we decided that to keep the dialogue in each of the episodes separate, we would maintain individual language databases for each one and bring in any lines we wanted to reuse when we created a new episode. In theory that sounds good, but in practice, we ended up missing lines that were written later in
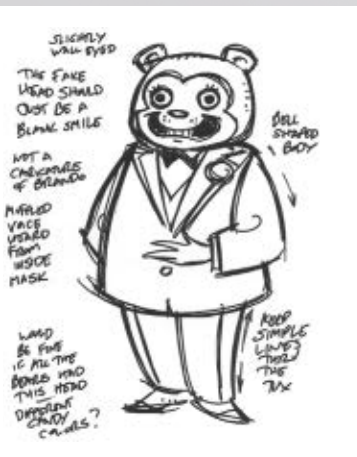
the process. That meant we had to take the time to manually import the language resource, voice file, and lip sync animation every time we found a line that had been left out or changed. That's a pretty simple task, but it requires a broad knowledge of the game's implementation and can add up to a good amount of time lost on any given episode.

The show must go on, and we'll continue to deal with these sorts of problems as we ride out the season. Based on our experiences with shared resources in SEASON ONE, we're planning a robust production database that should make it a lot easier to manage these sorts of issues across episodes and projects.

4 **WORKING WITH CONTRACTORS.** For most game companies, working with outside contractors is a fact of life. We rely on contractors for much of our art, and selecting the right people to work with takes time. Once we settle on which contractors we're going to use, even more time goes into explaining exactly what we need. Since the SAM & MAX schedule is very tight, we don't have a lot of time to account for corrections or mistakes due to inadequate contractor talent or miscommunication. For example, we felt a lot of pain when a contract studio sold us an A-team, then once the contract was signed, gave us a B-team whose work fell short of the standards we agreed to. This situation inevitably led to extra time spent managing the contractors. Communication became even more difficult when the contractors weren't local (we had some on the other side of the planet). When the contractor's workday takes place during the middle of the night for us, what would normally be a 10-minute conversation turns into a two-day exchange.

Another issue, which is specific to episodic games, is that the scope of what contractors want is different than what we need. Contracting studios typically want large amounts of work contracted to them all at one time, and they want work that will continue over an extended period. Due to the small size of our SAM & MAX episodes, it can be hard to find large chunks of work that can be farmed out. The projects we do send to contractors tend to be small, and we need assets delivered in regularly spaced intervals. This scenario doesn't mesh with what many contractors are looking for.

All of this said, using contractors is unavoidable. When contractors are managed effectively, they help us save time and money, and the fact that we can get more work done in a shorter time period is critical for our schedule. One way we've been able to maximize the work contractors do is to find a few good people who are local—people we know and trust. This way we know what to expect when the work comes back, and since they've worked with us before, less time is spent getting them up to speed. It's also important for us to keep track of their availability, particularly on a project with as regimented a

schedule as SAM & MAX, so we know we'll be able to get who we need when we need them.

5 **PEOPLE WERE SPREAD TOO THIN.** Having a small and talented team can be a bit of a double-edged sword. When we set out to start on SAM & MAX: SEASON ONE, we looked at our collective skill sets and knew we'd be able to accomplish a tremendous amount of work with a small group of people. Everyone on the team is very talented and capable of handling almost any aspect of building a game, but we soon learned that it's impossible for everyone to do all of the things they're capable of doing, all at the same time.

Faced with the task of creating a cascading production schedule for a series of six games to be completed within a year, we made projections based on our experience with larger, non-overlapping titles. What we didn't take into account was that when schedules overlap, any one person may need to be doing two very different tasks at the same time. This can quickly become overwhelming. for example, the art lead found himself having to finalize the animation in EPISODE ONE at the same time he was supposed to be doing concepts for EPISODE THREE. The programmers were building new technology for later episodes while also fixing bugs in earlier ones. And the designers were concurrently writing the script for one episode while designing the one that followed it.

While we have a very capable team of people who can wear all these different hats, we've learned that expecting one person to own art direction (or programming or design) on multiple episodes at the same time is not the most effective way to produce episodic games. This is probably why TV shows rotate directors and writers from episode to episode. The next time we plan a project like this, we'll probably want to staff it more like a TV show and rotate our crew rather than attempting to get the same level of work from everyone on every episode.

## NEXT TIME ON SAM & MAX ...

Even for those of us with years of experience in the industry, every day of development on SAM & MAX reveals something new. One of the great benefits of working on an episodic project is that we don't have to wait for the sequel a few years down the line, to make use of the lessons we've learned; we can start applying this knowledge to the very next episode. As people who are constantly striving to make the best games we can, being able to continually improve upon our own formula is a satisfying way to work.

Undoubtedly, there will be a lot more for us to learn in the second half of the project, some of which we won't completely understand until the season is long finished and we can step back and look at it objectively. But for the time being, there's one thing we can say for sure: SAM & MAX: SEASON ONE is an example of episodic gaming "done right," and that's something the entire team is extremely proud of. ✖

**Concept art from SAM & MAX: EPISODE ONE.**

# Never tried SlickEdit? What's your excuse?

## Top 10 Reasons NOT to Use SlickEdit

1. The more I type, the better I get at it.
2. The office is so peaceful after 10pm.
3. Real Programmers don't need no fancy tools.
4. Nothing is more satisfying than formatting my code by hand.
5. I'll get disability if my Repetitive Stress Injury is severe enough.

## 6. I get paid by the hour.

7. Milestones? What are milestones?
8. Traffic is so much lighter later at night.
9. If I finish my work on schedule they'll just give me more to do.
10. I never liked spending time with my family anyway.

**There's really only one good reason <u>not</u> to use SlickEdit — *you've never tried it.*** How can we be so sure? User feedback. For 19 years we've listened to power programmers and continued to try to exceed their expectations. Our multi-language development tools and advanced code editors are fast, powerful, and flexible — whether you're developing on Windows, Linux, UNIX, or Mac OS. Whatever your environment, SlickEdit has a tool for you. Our product suite includes SlickEdit®, SlickEdit Tools for Microsoft® Visual Studio® 2005, and the SlickEdit® Plug-in for Eclipse™.

**slick**edit.

**www.slickedit.com**

**Download a FREE trial at www.slickedit.com/trial or call 1–800–934–3348.**

# NO PLAYER LEFT BEHIND

## AN INTERVIEW WITH
## GAMELAB'S PETER LEE

>> **GAMELAB IS A RATHER UNUSUAL CASUAL GAMES COMPANY. NOT ONLY** does it strive for original designs (such as the hit DINER DASH), rather than rehashes of existing games or concepts, but it also runs a nonprofit and works in the educational and serious games spaces. The company's new project, GAMESTAR MECHANIC, aims to teach young children game design. We spoke with GameLab co-founder and co-CEO Peter Lee about his push for originality in casual games and why games in education is good for the entire industry—not just the children themselves.

**Brandon Sheffield:** DINER DASH was an original design for a casual game. What does it mean to GameLab to design a game from the ground up versus rehashing something older?

**Peter Lee:** Well first of all it's more fun, but Eric [Zimmerman] and I started the company wanting to explore different game design mechanics. We were very game design-centric. For us, I don't think it makes sense to recreate a game that already exists. For us as game designers, taking somebody else's idea is not that exciting.

From a business perspective, it's also bad [to rehash an existing game] because you want to be the first person who comes up with the main content. I think copying is happening in every industry—there are leaders and there are followers when something's a hit.

I think we always wanted to be in the forefront and find a new way to play rather than follow what's hot now. We'd rather be someone who forges new trends.

**BS:** What do you think about the trend toward rehashes in the casual games space?

**PL:** For somebody to just copy a game, it takes probably less than three months. For most of these games, if you start from scratch—it's still nothing compared to a console game—it takes about eight months because if you are trying new game mechanics, you have to iterate it out.

So there's actually more risk involved in making copies and probably more reason not to do it in the casual industry. If you're just making a copy, there are probably a hundred others doing the exact same thing. Then the market becomes oversaturated with the same content.

**BS:** How do you test the market for casual games? It seems like it might be harder since the market is not well defined.

**PL:** That's very true. It's really hard to predict what to expect from the market. There's some general sense of it, but the portals are so secretive about the users.

If you look at different portals, there's some sort of playing behavior from portal to portal, and we get some information from that. ... But we don't actually believe in focus testing because we don't think it actually works.

We do like to do lots of playtesting. It's not like we have a huge testing pool, but we have about 200 or so testers, a testing community that we've built. As we develop games, we bring those people in to do random testing. Some might come for an hour or two, some might play all day. We just watch them and see how they do it.

**BS:** How are you going about the film-based funding model? Do you search for funding for each specific game?

**PL:** That's correct. Basically the only model here in the game industry is to partner with a publisher who will fund games. It's pretty limited. It limits the things you can make, what kind of distribution model you have.

I'm Korean, so I've visited Korea many times and have talked to developers in Korea. Basically, they use every kind of funding model they can find and try everything, not just the developer-publisher model. It helps them make more games than they would be able to otherwise.

It's not easy because nobody has done it yet, but we funded two downloadable games with that model, and we're still talking to people. It requires lots of education on our part because venture capitalists and their

ilk are not used to this sort of thing. We've also been talking to a lot of film investors, the types that fund indie films, and we've been getting some positive responses from them. This is definitely the direction we want to head. It puts us in a better spot in terms of ownership of the property and distribution model.

We can do other interesting things as well; we got funding from Curious Pictures, which does animation and advertising. That was a good relationship for us because one other thing we want to do is move into TV and movies.

When we're developing these casual games, we're actually paying attention to having a rich narrative and universe so that we can build an actual show out of it. Once a game launches, we'll start shopping around show concepts, whether that's TV or digital distribution.

**BS:** *Tell me about the GameLab Institute of Play.*

**PL:** We've been working with nonprofit organizations for the last few years, doing consulting work and creating some games, and one of the games we did last year, AYITI: THE COST OF LIFE, is getting lots of publicity. There's also the Game Designer Project, now called GAMESTAR MECHANIC, which is getting funded by the MacArthur Foundation. ...

We've been interested in game design itself as a useful discipline that can be applied to a broad range of things. There are lots of movements like serious games or games for change, but most of those are focusing on teaching math, or physics, or certain established disciplines. We feel [people are] missing out on the fundamental benefits of game design itself. It doesn't mean we're making a bunch of future game designers, but it's basically like liberal arts. If people learn this discipline, it's very applicable to other disciplines. Game design is all about interacting with other players and people. It has broad application.

It also covers visual design, information design—there's writing involved, programming, if that's what you want to do. So there are specific skills you can develop, but at the same time, there's a problem-solving process in a more general sense.

**BS:** *Is this more like education on an individual scale or is it also about global awareness of games as a medium?*

**PL:** For us it's the latter. The GAMESTAR MECHANIC project itself is still in progress, but it's targeted at a younger audience. Our general target is more broad range, though. I think there will be different products or programs that come out of that which cover different age groups.

**BS:** *You're working with the University of Madison–Wisconsin, so I assume that includes James Paul Gee. How is he involved with the project?*

**PL:** [Gee's] main role is—since we're game designers and not educators—to help us design the curricular aspect of it. He's playing content expert in a way. Once the tool is ready and the game is built, he'll determine how the supporting curricular material can be generated. That's the university's main role. They organize testing with the kids, as well.

**BS:** *How do you envision this being used in schools?*

**PL:** At the moment it's sort of targeted at the after-school programs. I don't know how aware you are of school issues, but there are a bunch of hurdles you have to go through before you can get anything into schools. There are financial hurdles, tech support issues, or not actually having computers. Plus there's a privacy issue. You're developing this tool, and it's community-oriented, and some schools have problems with students getting access to the internet because they're minors. Lots of issues.

We're going to start with after-school programs because it's a little easier to get into them. I think lots of other research is going to flow from this to evaluate the usefulness of actual lessons and what kind of learning can take place. That's one of the reasons we're doing this. Lots of people have been talking about the benefit of games, but nobody has proven it yet. ... Also, I think it will do good things for the game industry. We have that bad reputation as a time-waster, even though we're growing so much. Hopefully, getting some credibility in the educational field will benefit the industry itself.

We're also doing research right now about opening a game-themed high school. If that happens, it would be another way to explore this further. As you know, the Bush administration and the No Children Left Behind Act are kind of destroying education. There are lots of people, including the MacArthur Foundation, who want to change this and do something good. This is an attempt to do experiments and research to see what we can actually do here. GAMESTAR MECHANIC will be the first project, but we'd like more things like that to test these concepts.

**BS:** *It almost seems like it's a political statement as well as a movement for the game industry.*

**PL:** That's true, in a way. It's kind of sad. People sort of respect the industry, but also don't at the same time. I think there's huge potential here. The video game industry is still new, and these kids growing up today play games all the time. I have a kid now, so I want to make sure there's a better school system, too. That's one reason I'm interested in starting up this high school.

**BS:** *What made you personally get into the game industry?*

**PL:** I had some programming and visual design background, and I was doing work for both. At some point, as I was doing graphic design, I felt somewhat dissatisfied doing web sites and advertising. I don't mean to insult anyone doing that work, but my problem was that while I was doing it, the things I made were only valuable because they did something else. My creations didn't have their own value. Advertising exists because it sells something else.

Then I started to hear and learn about game design. Games are a very pure medium. Games exist because they are games. There are influences from other things, but a game is valuable because it's a game and because it's fun. ... Not that I wanted to be an artist; I always wanted to be in a commercial industry. But I wanted something where what I create has more value because of what it is. I found games to be very satisfying in that aspect.

The other reason was just the fundamentals of game design. Eric Zimmerman, my partner, wrote a book about rules and play, and those theoretically opposed ideas of rules making play was one of the reasons I got into it. I think games can really change people, and that's a really powerful thing. ❖

# THE RIGHT TO BAER GAMES

## AN INTERVIEW WITH THE FATHER OF VIDEO GAMES

**RALPH BAER IS THE FATHER OF VIDEO GAMES. WANT TO CONTEST** that crown? Then produce a document that can prove it—one that's earlier than his. An engineer at heart, Baer has kept meticulous records of his work, from the first dot he displayed on a screen, through SPACE WAR, which became a proper game in its own right. In this interview, Baer discusses his roots, his work ethic, the problems with patents, and what drove him to create the first video games.

**Benj Edwards:** *I've read that you were born in Germany in 1922. When did you first come to the United States?*

**Ralph Baer:** 1938, in August, which was three months before Kristallnacht, when things got really nasty. I was damn lucky to get out just in time.

**BE:** *How did you feel about the U.S. when you first came here?*

**RB:** Well, I spoke English because I had English in school. I took English lessons, so it wasn't like I couldn't communicate or didn't know what was going on around me. Within a week after we arrived in New York, I'm working from eight in the morning until six at night in a factory for the next two years. Six months later, I see somebody in the subway with a magazine. On the back of the magazine [it said], "make big money in radio and television servicing," and somehow, that was me.

I spent a buck and a quarter every week out of my $12 of wages on the correspondence course. I finished that in a few months, took the advanced course, finished that. Left the factory and started servicing radios.

[I] did it all: I picked up all deliveries, fixed radios, fixed early television sets, put up antennas all over the place on rooftops in mid-Manhattan, and then Uncle Sam came along and, yeah, I went back to where I came from. First England, then to France, you know, as a GI in '43.

**BE:** *Could you describe the atmosphere of your lab at the defense electronics company Sanders when you were developing the first video games?*

**RB:** I had a little bitty room that once was the company's library when they first started on the fifth floor of the Canal Street building in Nashua, New Hampshire. And you entered that directly opposite the elevator. You entered it, made a left turn, and you were in that little bitty room.

I gave my technician, the engineer who worked with us for a little while, keys to the door, and I had a key and nobody else had a key. Nobody knew what was going on in that room. It was a floor above where my division was, and it was nobody's business; and in fact, it would have been ridiculed by a whole lot of people if they knew.

**BE:** *How did it feel to be creating games, of all things, within this military contractor that's probably usually pretty serious?*

**BENJ EDWARDS**

is editor-in-chief of the Vintagecomputing.com weblog, covering both retro gaming and retro computing. He has been collecting vintage computers and game systems for more than 13 years. Email him at **bedwards@gdmag.com**.

An early light gun
prototype from Baer.

**RB:** To tell the truth, it was a piece of Jewish chutzpah. I'm running a division, and it's certainly hard enough to think about commercial products, never mind come up with things like video games. But, you know, I'm a creative guy, so I do what I do with it.

I run a big division, so I've got certain prerogatives, right? [I get] a couple of guys, I put them on the bench, and make them do something I want done, and nobody needs to know about it. It doesn't even ripple my overhead. If two guys out of 500 [work quietly on something else], it doesn't even show up. So I just did it.

When we had something that looked like it was neat, I called Herb Campman, our corporate director, and he saw it immediately. Then he gave me $2,000 and made me honest. But within a month, I had to do a demonstration for the president of the company, and all I saw was long faces. "What the hell are you screwing around with?" The board of directors was meeting that day, and they were all there. They were all watching my demo. But there were two people who were very supportive right off the bat, guys with insight, guys who had foresight. They sort of tolerated it.

*BE: All your patents ended up making them a lot of money later, didn't they?*

**RB:** Well, when the money started coming in, everybody told me how supportive they had been. Yeah, sure. But when the money started coming in, my name and the name of the corporate director of patents, who had negotiated the licenses, were up on the towering neon lights.

We'd be in the quarterly meeting, along with other division managers, looking at the performance of various divisions, and our licensing income was always bigger than that of the biggest division of the company. So we could do no wrong. That's when I started going out and doing my own toy and game design.

I got out of running a division and became the first engineering fellow at Sanders—later, a Lockheed fellow—and I could do what I wanted. No one's going to question me as long as I bring in all that money.

*BE: I'm sure that freedom is priceless for an engineer who wants to be creative and just do his own thing.*

**RB:** Oh, yes. I mean, I've had a ball for the last 25 or 30 years. If I hadn't gotten out of running a division, I'd probably be dead and buried by now, with all the daily stress that goes along with an ordinary job, when you have major responsibilities.

All that stress ... I didn't have any of it once the money started coming in, and once I was identified [as the one who had] the money coming in, I was constantly in Chicago, or San Francisco, or Montreal, or other places in court, defending the patents and making tons of money for them. Money's what counts.

*BE: Do you think it's important for an inventor's success to keep great notes and records?*

**RB:** Absolutely. From day one. Even if they're just hen scratches. As you work on stuff, you make notes. They're called schematics if you're building electronics hardware. Everything has to be kept, everything has to be dated and signed, and if you think you've got something really serious going on, get somebody else who understands the stuff to read it and sign it. Sign it, "Understood, blah blah blah, name, date." It makes all the difference. Memory doesn't count for diddlywink in court, because it's totally unreliable. Even with the best of intentions.

*BE: Where did you acquire that record-keeping acumen?*

**RB:** I'm just a logical guy. I can't help it. Some people hate it, you know? I'm the kind of guy that goes into the kitchen and sees a piece of napkin on the floor, and I gotta pick it up. I can't take it. If the faucet in the sink is wet, I gotta wipe it down. It's just me.

It never occurred to me that I wouldn't put down what I'm doing, and I certainly required it of everyone who worked for me. At the time I wrote that four-page disclosure document, I was running close to 500 people in a division, and everybody there was required to keep a daily notebook for the same reason. Mainly, it was required because the military requires it. If you work on a military program, you must keep a log and you must sign it, and counter-sign it, in some cases.

*BE: You've obviously benefited from the patent system pretty well. These days, there's a lot of criticism about the U.S. Patent system. Do you think it still works, or does it need fixing?*

**RB:** I don't think it ever worked. You look at the patents, and three out of four are garbage. Especially since it's so easy to do patent searches on the web. It's very easy. You look at that stuff, one piece of crap after another. How the hell did that ever get in there and clog up the system to where stuff that should have really been handled in an expeditious manner didn't make it through the damn office for three years or even longer? That's problem number one.

Number two: The kind of response you get from the examiners is very much a function of who the examiner is. Some of the examiners were barely out of law school; it was their first job. They had no practical experience. Some of them had been engineers before, others had not. It was a totally variable cast of characters, like the guy who handled our first video game patent application. He was very nice, but we worked him by having the patent lawyer sit across the desk from him, discussing the various objections he had to the various claims, while I'm setting up a little 10-inch black and white GE television set and an early Ping-Pong game. I hooked it all up, and the examiner didn't want to look at it, no-how. Because they don't, on principle, want to look at stuff.

Within half an hour, he's playing games. He has half the corridor come in. People up and down the corridor come in and play games.

*BE: Yeah, nothing works quite like showing people exactly what you're talking about. You can never really tell them, exactly. You have to show them.*

**RB:** We didn't know it at first. We had the fifth model Ping-Pong in late 1967, and we demonstrated it to TelePrompter in January

w/gun and
...lers

of '68. Did we know that that was all we really had to do? No. We went on through three more models with all kinds of additional stuff, including, of course, shooting at the screen with a light gun. And we could have stopped a year before and would have had all that we needed to have. Who knew that Ping-Pong was all that was needed?

**BG:** *Before you created the first video games, was there any other application that used a regular TV set to do anything other than receive broadcasts?*


**SPACE WAR, seen here in vector-based arcade format.**

**RB:** In the military, they used television sets and modified them. In one case, in a German patent, which I had to defend against many, many times in court, they used a spot that traveled across the screen that was supposed to be a missile that you launched. And then there was another spot somewhere that was supposed to be a tank you were supposed to hit. They did simulations like that. But nobody thought of doing any of that on a home television set for individual, normal use. That was really the leap of imagination, to do something for people with their 40 million TV sets.

**BG:** *You originally called your games "TV games." Do you know how the term "video games" originated?*

**RB:** They were always called TV games. I have no idea who coined the term video game. That happened somewhere in the coin-op period, maybe around '73 or '74. Nobody called the original PONG game a video game. Nobody had heard the term.

To me, it was meaningful because the term "video" is now in use sort of generically for any kind of graphics, especially moving images, on a screen-any kind of screen. But that's not how it was coined. A video signal was a very specific thing. It was a definitive term that only applied to raster-scan television.

**BG:** *Do you play games now?*

**RB:** I'm not a game player. I love making games because I love to be able to come up with a design, a concept or a design. I like the creative process. I'm like a painter, a portrait painter or an artistic painter. And really, what I do is an art form. The engineering part is a part of it. But after you've done 50,000 different things, most of what you do is a combination of what you've already done 17 times before. But the concept of a game-that's always new. It's always fresh. I love to do that stuff.

**BG:** *When's the last time you played Ping-Pong in real life?*

**RB:** Not so long ago. ... [My son Mark has] a Ping-Pong table, and I've played my grandson. I still play Ping-Pong reasonably well. I got lucky, you know. I have leukemia, but I've been in remission for about five years now. I feel stronger and younger now than I did five years ago. I can really move. I have no endurance, but I can move. You'd never know I'm 85. So I can play Ping-Pong, but not very long. [Laughs] I get tired.

**BG:** *Do you think real Ping-Pong is more fun than your video game version?*

**RB:** Well, it's different fun, but it's a lot of fun. And to think about Ping-Pong, I have one more digression. One of the complaints that his highness Nolan Bushnell had was, "You didn't have any scoring on screen," to which I respond, "Well, it's kind of funny. We've been playing real Ping-Pong for the last hundred years. And tennis. And guess how you score tennis and Ping-Pong? You call out the score, you know, nice and loud. Nobody needed any scores on the screen."

That was a real iffy addition. I had no way of doing it with the technology available to us for a price in 1966–67. But it was not necessary to play an interesting tennis game. You just call it out. Who needs scoring?

What was stupid on our part—and I couldn't believe in retrospect—was that we didn't have any sound. Yeah, that was the big addition that made it much more lively a game that Alan Alcorn and Bushnell came up with, adding a pong sound when you hit the ball. Why we didn't think of that, in retrospect? I can't believe we didn't do that. Part of it was that I wasn't really a game person, ever. It only grew as I worked with the stuff. ❖

## » THE INNER PRODUCT

# PRACTICAL FLUID DYNAMICS: PART I

**FLUID EFFECTS, SUCH AS RISING SMOKE** and turbulent water flow, are everywhere in nature but are seldom implemented convincingly in computer games. The simulation of fluids (which covers both liquids and gases) is computationally very expensive. It's also mentally expensive, with even introductory papers on the subject relying on the reader to have math skills at least at the undergraduate calculus level.

In this two-part article, I will attempt to address both these problems from the perspective of a game programmer who's not necessarily conversant with vector calculus. I'll explain how certain fluid effects work without using advanced equations and without too much new terminology. I'll also describe one way of implementing the simulation of fluids in an efficient manner without the expensive iterative diffusion and projection steps found in other implementations.

A working demonstration accompanies this article and can be downloaded from www.gdmag.com; example output from this can be seen in Figure 1.

### GRIDS AND PARTICLES

There are several ways of simulating the motion of fluids, but they all generally divide into two common styles: grid methods and particle methods. In a grid method, the fluid is represented by dividing up the space a fluid might occupy into individual cells and storing how much of the fluid is in each cell. In a particle method, the fluid is modeled as a large number of particles that move around and react to collisions with the environment, interacting with nearby particles. Let's focus first on simulating fluids with grids.

The simplest way to discuss the grid method is in respect to a regular two-dimensional grid, although the techniques apply equally well in three dimensions. At the most basic level, to simulate fluid in the space covered by a grid you need two grids: one to store the density of liquid or gas at each point and another to store the velocity of the fluid.

Figure 2 shows a representation of this, with each point having a velocity vector and containing a density value (not shown). The actual implementation of these grids in C/C++ is most efficiently done as one-dimensional arrays. The amount of fluid in each cell is represented as a float. The velocity grid (also referred to as a velocity field, or vector field) could be represented as an array of 2D vectors, but for coding simplicity it's best represented as two separate arrays of floats, one for X and one for Y.

In addition to these two grids, we can have any number of other matching grids that store various attributes. Again, each will be stored as a matching array of floats, which can store factors such as the temperature of the fluid at each point or the color of the fluid (whereby you can mix multiple fluids together). You can also store more esoteric quantities such as humidity, for example, if you were simulating steam or cloud formation.

### ADVECTION

The fundamental operation in grid-based fluid dynamics is advection. Advection is basically the moving of things on the grid, but more specifically, it's moving the quantities stored in one array by the movement vectors stored in the velocity arrays. It's quite simple to understand what's happening if you think of each point on the grid as an individual particle, with some attribute (density) and a
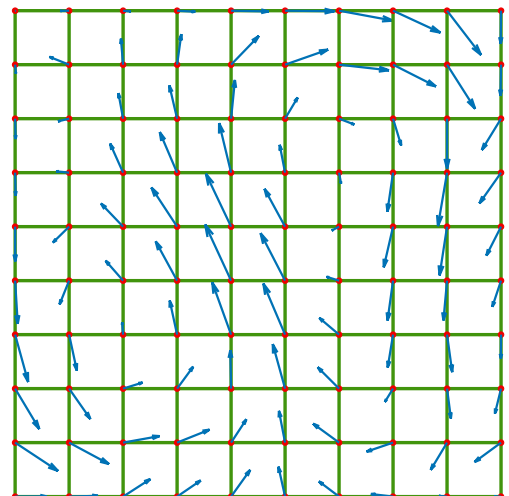
**MICK WEST** was a co-founder of Neversoft Entertainment. He's been in the game industry for 17 years and currently works as a technical consultant. Email him at mwest@gdmag.com.
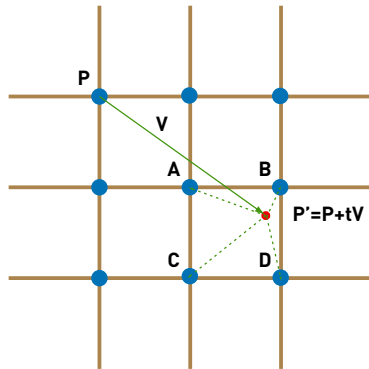
**FIGURE 3** Forward advection: The value in P moves forward to A, B, C, and D. This dissipates it when moving diagonally.
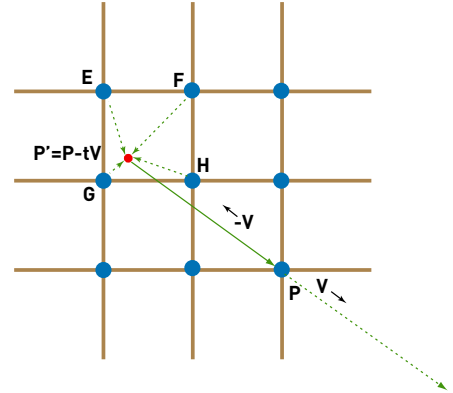


**FIGURE 4** Reverse advection: The new value in P is gathered from E, F, G, and H, one of which (H) is usually the same point as P.

velocity. Then you are familiar with the process of moving a particle by adding the velocity vector to the position vector. On the grid, however, the possible positions are fixed, so all we can do is move (advect) the quantity (density) from one grid point to another.

In addition to advecting the density value, we also need to advect all the other quantities associated with the point. This would include additional attributes such as temperature and color, but also the velocity of the point itself. The process of moving a velocity field over itself is referred to as self-advection.

The grid does not represent a series of discrete quantities, density or otherwise; it actually represents (inaccurately) a smooth surface, with the grid points just being sample points on that surface.

Think of the points as being X,Y vertices of a 3D mesh, with the density field being the Z height. Thus, you can pick any X and Y position on the mesh, and find the Z value at that point by interpolating between the closest four points. Similarly while advecting a value across the grid, the destination point will not fall directly on a grid point, and you'll have to interpolate the value into the four grid points closest to the target position.

In Figure 3, point P has a velocity V, which, after a time step of $t$, will put it in position $P'=P+t*V$. This point falls between the points A, B, C, and D, and so a bit of P has to go into each of them. Generally, $t*V$ will be significantly smaller than the width of a cell, so one of the points A, B, C, or D will be P itself. There

are various inaccuracies when advecting an entire grid like this, particularly that quantities dissipate when moving in a direction that is not axis-aligned. But this inaccuracy can be turned to our advantage.

## STAM'S ADVECTION
Programmers looking into grid-based fluid dynamics for the first time will most often come across the work of Jos Stam and Ron Fedkiw, particularly Stam's paper "Real-Time Fluid Dynamics for Games," which he presented at the 2003 Game Developers Conference. He describes a very short procedure for making a grid-based fluid simulator. In particular, he shows how to implement the advection step using what he calls a "linear backtrace," which simply means that instead of moving the point forward in space, he inverts the velocity and finds the source point in the opposite direction, essentially back in time. He then takes the interpolated density value from that source (which, again, will lay between four actual grid points), and moves the value into the point P. See Figure 4 for an example.

Stam's approach produces visually pleasing results, yet suffers from a number of problems. First, the specific collection of techniques discussed may be covered by an existing patent (U.S. #6,266,071), although as Stam notes, backtracing dates back to 1952. Check with a lawyer if this is a concern to you. On a more practical note, the advection alone as Stam describes it simply does not work accurately unless the velocity

field is smooth in a way termed mass conserving, or incompressible. Consider the case of a vector field in which all the velocities are zero except for one. The velocity cannot move (advect) forward through the field, since there's nothing ahead of it to "pull" it forward. Instead, the velocity simply bleeds backward. The resultant velocity field will terminate at the original point, and any quantities moving through this field will end up there.

We can solve this particular problem by adding a step to the algorithm termed projection. Projection essentially smooths out the velocity by making it incompressible, allowing the backtracing advection to work perfectly and making the paths formed by the velocity "swirly," as if it were real water. The problem with this approach is that projection is quite expensive, requiring 20 iterations over the velocity field in order to "relax" it to a usable state.

Another performance problem with Stam's approach is that he uses a diffusion step, which also involves 20 iterations over a field. It's necessary to allow the gas to spread out from areas of high density to areas of low density. If the diffusion step were missing, solid blocks of the fluid would remain solid as they moved over the velocity field. Diffusion is an important cosmetic step.

## ACCOUNTING ADVECTION
If a velocity field is not mass conserving, then some points will have multiple velocity vectors from other points pointing toward them. If we simply move

# A.I. for the masses ...
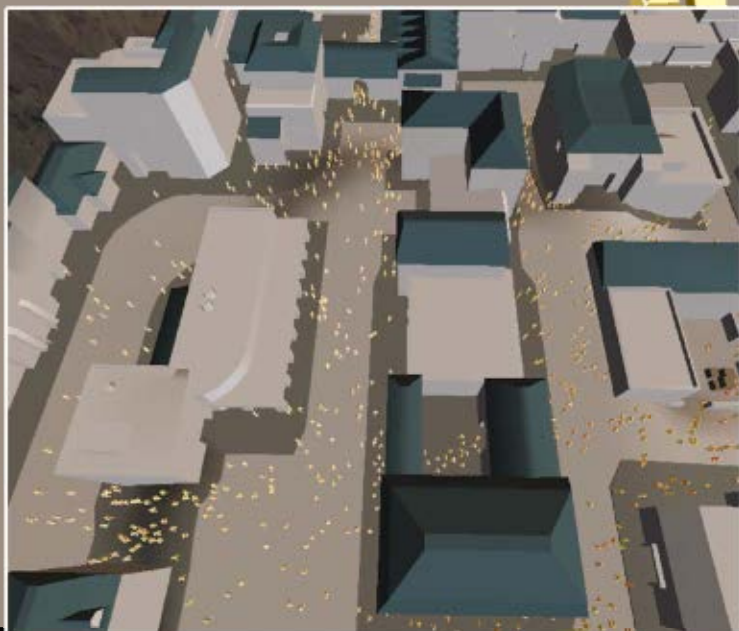


## Kynapse 4.0 is the next generation in A.I. middleware.

Kynapse provides a flexible, production tested solution for your titles' requirements.
Kynapse is truly multiplatform - we support Xbox 360™, PLAYSTATION®3, Windows, Linux,
Xbox™, Playstation 2, PSP™ and Gamecube -  and can be easily integrated into your production
environment. Our new large scale A.I. gives developers free rein to provide complex, immersible
environments for gameplay.

Leading companies such as Artificial Mind & Movement, Lionhead Studios, Real Time Worlds,
SEGA, Spark Unlimited and Turbine have trusted Kynapse to help supply a truly next generation
gaming experience.

**Contacts**

France, headquarters
12 rue de la Chaussée d'Antin
75009 Paris
France
Phone : + 33 156 035 980
Fax    : + 33 153 346 508

United Kingdom
Leatherhead
Surrey
Phone : +44 (0) 1372 454083

**www.kynogon.com**

Montreal
1080 Cote du Beaver Hall,
Suite Gr-2
H2Z 1S8, Montreal, Quebec
Canada
Phone: +1 514 667 0871
Fax: +1 514 931 8865

# kynogon
### no mind no game

our scalar quantities (like density) along these vectors, there will be multiple quantities going to (or coming from) the same point, and the result will be a net loss or gain of the scalar quantity. So, the total amount of something such as the density would either fade to zero or gradually (or perhaps explosively) increase.

The usual solution to this problem is to make sure the vector field is incompressible and mass conserving. But as mentioned before, it's computationally expensive to implement. One partial solution is to make the advection step mass conserving, regardless of whether the velocity field is actually mass conserving. The basis of this solution is to always account for any movement of a quantity by subtracting in one place what is added in another.

Advection uses a source and destination buffer to keep it independent of update order. In Stam's implementation, the destination buffer is simply filled one cell at a time by combining a value from four cells in the source buffer and placing this value into the destination buffer. To properly account for compressible motion, we need to change this copying to accumulating, and initially make the destination buffer a copy of the source buffer. As we move quantities from one place to another, we can subtract them in the source and add them in the destination.

With the forward advection in Figure 3, we're moving a quantity from point P to points A, B, C, and D. To account for this, we simply subtract the original source value in P from the destination value in P, and then add it (interpolated appropriately) to A, B, C, and D. The net change on the destination buffer is zero.

With the reverse advection in Figure 4, as used by Stam, the solution would initially seem to be symmetrically the same: Subtract the interpolated source values in E, F, G, and H from the destination buffer, and add them to P. While this method works fine for signed quantities such as velocity, the problem here is that quantities such as density are positive values. They cannot drop below zero, as you cannot have a negative quantity of liquid.

Suppose that point E was one source point for two destinations P1 and P2, both of which wanted 0.8 of E. If we follow our initial plan and subtract 0.8*E from E and add 0.8*E to both P1 and P2, the net effect is zero, but now the value at E is negative. If we clamp E to zero, then there is a net gain of 0.6*E. If we subtract 0.8*E from the source value of E after updating P1, then

when we update P2, it will only get 0.8*0.2*E, when clearly both P1 and P2 should get equal amounts. Intuitively, it seems they should both get 0.5*E, and the resulting value in E should be zero, leading to a net zero change.

To achieve this result, we create a list that for each point, noting the four points of origin for each and the fraction of each point they want. Simultaneously, we can accumulate the fractions asked of each source point. In an ideal world, this would add up to one, as the entire value is being moved somewhere (including partially back to where it started). But with our compressible field, the amount of the value in each point that is being moved could be greater than or less than one. If the total fraction required is greater than one, then we can simply scale all the requested fraction by this value, and the total will be one. If less than one, the requesting points can have the full amount requested. We should not scale in this case, as it will lead to significant errors.

With the mass conservation of advection fully accounted for in both directions, it turns out that neither forward nor backward linear advection alone will produce smooth results. After some experimentation, I determined that applying forward advection followed by backward advection worked very well and gives a smooth and artifact-free flow of fluid over a compressible velocity field.

## NOW WHAT?

We can now perform both forward and reverse advection in a mass-conserving manner, meaning we can move fluid around its own velocity field. But even though our velocity field does not need to be mass-conserving, we actually still want it to be, since the velocity fields of real world fluids generally are incompressible.

Stam solves this problem by expensively forcing the field to be fully mass conserving after every change. It's necessary, since the reverse advection requires it. The key difference is that since our advection step does not require the field to be mass-conserving, we're really only doing it for cosmetic purposes. To that end, any method that rapidly approaches that state over several time steps will suit our purpose. That method, and the method of diffusion, can be found in the accompanying code, and I will discuss how they work in next month's article. ✖

## ›› PIXEL PUSHER

# THE BLACK ART OF PIVOT ANIMATION

**ANIMATORS KNOW HOW IMPORTANT GOOD** pivot placement can be, and to prove it, Pixel Pusher has devoted several columns to anatomy lessons and pivot points. But pivots serve two very different purposes in animation. In a skeleton, they're where the bones come together, whereas in a control object, like a foot target, they represent the place were leverage has to be applied. The "pivot" of a foot might be the edge of a shoe heel, or the ball of a flexed foot, or the tip of a toe. If your animation rig assumes that the pivot of the foot is always just the ankle, you're on the hook for a lot of annoying counter animation to keep that foot from slipping around.

The antidote for this kind of hassle is a rig that lets you rotate the foot from the toe or the heel as the situation requires. Pivot choice, though, is important for more than just heel and toe work. Getting the proper pivot point is critical for simulating all sorts of complicated and messy collisions and leverages. Mastering the art of faking pivots is essential for any serious rigger.

### ONE SPELL WONDER

The power (and the perversity) of pivot animation is that it basically combines rotation and movement into one operation, which explains why good pivot control helps you avoid tedious counter-animation. It's also the reason objects with animated pivots can be confusing for animators: things aren't necessarily where they seem.

Pivot animation works best when the animated object has a well-understood range of motion, like a foot or a hoof. Pivot manipulation can't stand in for full-on physics, but simple scenarios, like a flopping box or a rocking chair, are easy to fake. A dropped weapon realistically colliding with the ground, on the other hand, has to be left to a physics engine (or a ton of keyframes!).

Since there's no way to cover the entire topic of pivot animation in a single column, we'll stick to the basics: simple methods for animating pivots using only extra transforms and simple expressions.

Maya actually lets you animate the position of an object's pivot directly, but it's safer and less mysterious to work with a hierarchy of regular transforms instead. Setting up a pivot animation system sometimes devolves into trial and error, so being able to see and choreograph all the pieces of the puzzle is important to productivity.

### REVERSE FOOT PIVOTS

Most animators learn pivot tricks from the faithful heel and toe rig, also known as reverse foot. The reverse foot setup uses a hierarchy to provide separate pivots for the heel and toe. The simplest version (Figure 1) involves two controls. First, there's a move-and-rotate control, located where the heel hinges on the ground. The second control is the rotate-only child, located where the toe flexes off the ground. The IK effector that drives the leg is attached to the toe bone's terminus. The actual foot bone is orient-constrained to the toe control as well.

This setup has the virtues of simplicity and robustness. Moving and rotating the heel control rotates the model's foot from the heel. When the heel is rotated down flat, the toe control can be used to rotate the foot from the toe instead. This simple



**FIGURE 1** This typical reverse foot setup for heel and toe work shows how the orange heel control can lift the foot from the heel; the yellow toe control raises it from the toe. The skeleton's ankle IK (green) and foot are constrained to the control.

**STEVE THEODORE** *has been pushing pixels for more than a dozen years. His credits include MECH COMMANDER, HALF-LIFE, TEAM FORTRESS, and COUNTER-STRIKE. He's been a modeler, animator, and technical artist, as well as a frequent speaker at industry conferences. He's currently content-side technical director at Bungie Studios. Email him at* **stheodore@gdmag.com**.
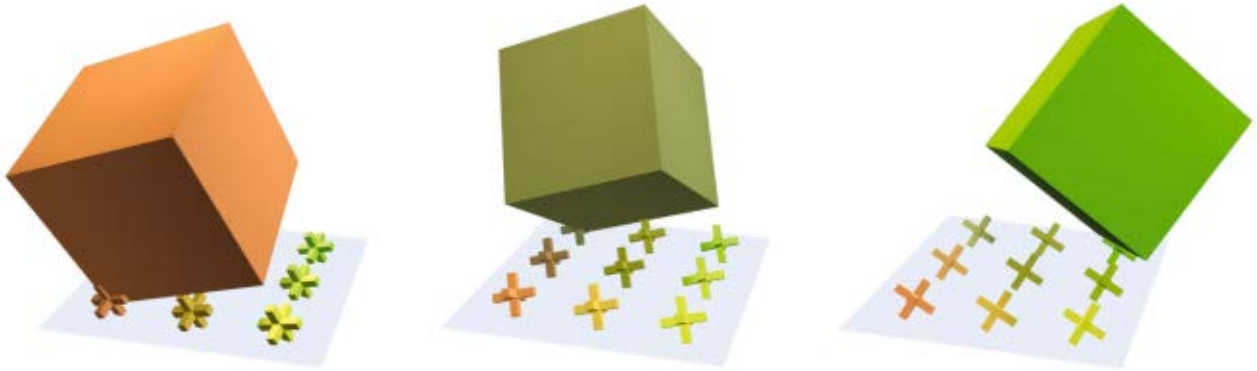
FIGURE 2 In this constraint-based rocker, constraint hands control to the correct pivots as the cube rolls.

idea—using extra controls to simulate different pivots for that final effector—can be easily extended to more complicated arrangements. For example, it's easy to insert a control that will allow the foot to flex if the model has articulated toes or an extra pivot for twisting around the ball of the foot. And of course, feet aren't the only application. Any object that has to lever off multiple points of contact could use a similar setup.

Reverse foot style pivot faking isn't perfect. The effects of the controls overlap, which can cause confusion or hitches. For example, if the toe control isn't zeroed out, the foot won't appear to pivot around the heel properly or vice versa. The animator also has to coordinate two different controls to produce the pivot handoff, which makes it harder to read f-curves and time keys. Finally, the complexity and potential for overlaps add up fast as additional pivots are added. Switching pivots from side to side as well as front to back, for example, involves at least nine possible pivots— way too much work for manual control.

## PARENTAL GUIDANCE

When the number of fake pivots gets large, it's better to skip the hierarchy and use a set of parent constraints instead. Using constraints lets you combine all the pivot switching into a single control, making graceful handoffs between pivots much easier and avoiding overlaps. The general outlines of a constraint-based system are pretty much the same, whatever the specific task. Figure 2 shows a pretty clear example of how this occurs, but here's a rundown of the six major elements: control object, counter-rotating transform, anchor transform, dummy pivot, constraint system, and translation control.

*Control object.* The control object, also known as the "rocker," is the object the user interacts with. It's also what you use to drive the actual model. For example, it's the object to which you attach IK handles or orient constraints in the final rig.

The rocker control is rotate-only, and it rotates only in the pitch and roll axes. The two-axis limitation prevents gimbal locks that would confound the pivot switch expressions. Keeping the translations on a separate control is important because animating pivots can make the "real" position of an object hard to decipher otherwise.

*Counter-rotating transform.* The parent of the rocker control shares the same location and pivot point. It's not intended to be touched by the user, so it should be invisible and un-selectable. The counter-rotator is wired to cancel out the motion of the control object. The rotations on the rocker will be fed to the alternate pivots, so this transform is there to make sure they aren't applied twice.

*Anchor transform.* The counter-rotator also has a parent, with the same location/pivot as the rocker. This one is likewise invisible and un-selectable. The transform is driven by the constraints, creating the

illusion that the original control is rotating with different pivots. Anchor transforms are necessary because the counter-rotator's local orientation is driven by the rocker, so the counter can't be directly constrained.

*Dummy pivots.* The pseudo-pivots are Max dummies or Maya locators, placed wherever you need them. For the common case of a box-like rocker, the standard arrangement is nine pivots at the corners, center, and edge-midpoints of the control object's bottom face, as in Figure 2.

The dummy pivots are wired to the original control object so they repeat its rotations. This connection has to be done with wires or expressions, not constraints, because we need to pipe the local values to them rather than world-space values.

*Constraint system.* The dummy pivots are all used to constrain the anchor
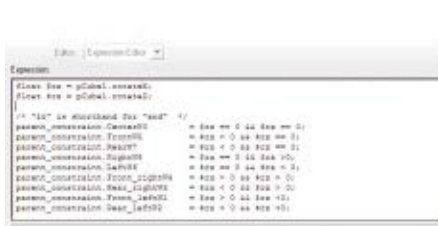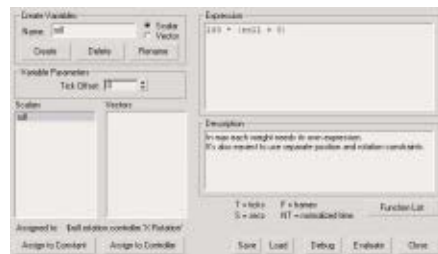


FIGURE 3 Sample expressions for driving pivot handoff are shown for 3DS Max (top) and Maya (bottom).

**FIGURE 4** This screenshot shows handoff expressions for faking a rounded base.

transform (the "grandparent" of the original control). An expression on the constraint weights will tell the control which pivot to use at any given time. In Maya, use the dummy pivots to parent-constrain the anchor directly. Max users have to give each dummy pivot a child with the same initial position and rotation as the anchor, and then point- and orient-constrain the anchor to them.

The expressions driving the constraint depend on the shape of the rocker you're trying to simulate. For a box structure,
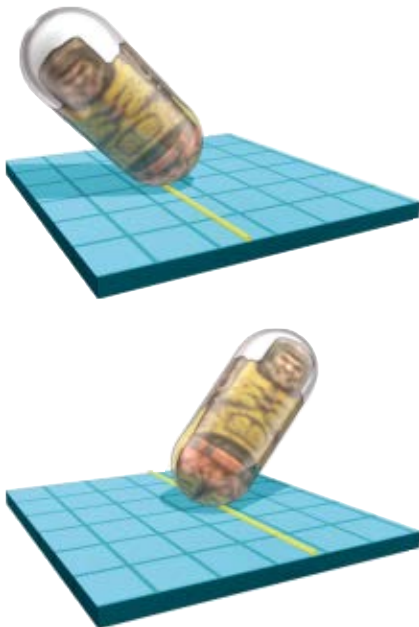


**FIGURE 5** An added translation, driven by the rotation values, creates the illusion of a rolling control.

the weights can use a simple Boolean test to turn them on or off as the control rotates around. Using a Boolean like those in Figure 3 ensures that control is passed from one pivot to another only when at least one of the axes is at zero, which hides the transition.

*Translation control*. The last element in a rocker system is the translation control, which is the parent of all the dummy pivots and of the anchor transform (thus the "great-grandparent" of the actual rocker control). This control's elevation and roll should be locked to prevent overlaps with the effect of the rocker. Its appearance ought to provide a clear visual indication of the real position of the whole system so that the effects of the pivot animation are easy to debug. For example, it might look like a footstep if the control is driving a leg rig.

The translator's yaw rotation should be animatable, but it has to be used with care. Without a big increase in complexity, the yaw rotation won't share a pivot with the other two rotations on the rocker, so if the translation control rotates while the rocker is supposed to be planted, the rocker will appear to slide. The setup as a whole becomes most useful when the ground contact is locked rather than twisting—that is, a walk cycle rather than a dance move or a boxer's shuffle.

The setup from Figure 2 is the typical box-like rocker that pivots on the edges of its base. Non-rectangular bases are trickier, but a rounded profile (like the flat base of a cylinder) can be approximated with four pivots and a slightly different expression setup. The round-base rig scales the constraint weights by the relative size of the rotations (see Figure 4). With this arrangement, the farther you're tipped on a given axis, the more influence the pivot on that "edge" has over the movement. The resulting behavior doesn't simulate a flat circle exactly, but it gives a fair approximation for very little extra work.

## CIRCUMFERENCE POWER
The rocking box is very handy, but it obviously doesn't handle objects with semi-circular or hemispherical bases, such as the Weeble in Figure 5. A rolling

base rig, more commonly called a "roller," is almost the same as the rocker rig sketched out above. Instead of using multiple pivots, however, you have only a single proxy pivot (and, naturally, no need for a complex constraint expression).

This proxy pivot is located at the "center" of the hemispherical base, as in Figure 5. Its rotations are driven in roll and pitch just like in the box rig; however, its translations in the horizontal plane are also driven with an expression thus moving the point of ground contact as the control rotates. The translate expression involves the dreaded magic number pi, so if you were traumatized by high school trig, you may need to take a couple of deep breaths. Actually, though, it's quite simple.

The circumference of a circle equals the diameter multiplied by pi. (A wheel turning full circle will move that distance.) To create a rolling control, therefore, all you need to do is express the rotation as a fraction of 360 degrees and then multiply that fraction by the circumference (pi*diameter). For example, a 90 degree turn of a ball 1 meter across would move the ball (90/360)*3.1415, or 0.785 meters (3DS Max users, remember to convert those degrees to radians).

## FULCRUMS AWAY!
That's about all we can jam in for this month. Don't forget that rockers and rollers are good for more than just feet. Any kind of lever system, from mechanical objects to forearms of a crawling character, makes a good match for a rocker. Rollers can help with objects from boats heeling over in the water to unicycles. There are plenty of tricks to play with once you've got the concept down.

Deciding how far to push a rig and when to give up and do the counter by hand is the classic animation dilemma. When do you put in a couple of hours today to save yourself hundreds of five-second twiddles in the future? Not even the humble P. Pusher (though always happy to offer advice!) can answer that one for sure. ✖

NOAH FALSTEIN

## »» GAME SHUI

# CASUAL STARTERS

**ONE OF THE DELIGHTS OF WRITING FOR** *Game Developer* each month is that it allows me to rationalize the worth of almost any time spent playing games. I knew the many hours I spent on DICEWARS just had to be research.



**DICEWARS (top) and DOUBLEGAME (bottom) are simple games that burgeoning designers should take note of.**

### ESCAPING THE CATCH-22

A frequent complaint of aspiring game designers is that game companies very rarely hire a designer who has no previous design experience—so how do you get that experience? One way is to get into the industry through a route other than game design. But you can also prove your worth as a designer by making a game yourself or with a friend. Even if you have no programming or art talent, you can make a paper game—a board game or collectible card game or the like. If you have even just a little experience in Flash or can collaborate with someone

who does, you might be able to make a game like one of these two.

### BEYOND RISK

DICEWARS is a very simple game reminiscent of the old board game classic Risk. First, the player chooses to battle against between one and seven computer opponents. Next, the game constructs a board of little hexagon-tile "countries," populates them with up to eight colors of players, and puts between one and eight dice in each country (randomly and relatively evenly distributed at first).

When it is your turn, you click on a pile of dice at least two tall in your color, and then click an opponent's pile that's adjacent to yours. Your dice are rolled, your opponent's dice are rolled, and whoever gets the higher total wins (defender wins ties). If the defender wins, the attacker loses all but one die involved in the battle. If the attacker wins, the defender's dice are eliminated and all but one of the attacker's dice move into the newly captured territory.

DICEWARS has no written instructions, and the only rule that may not be immediately obvious is that you are awarded new dice (again, placed randomly) equal in number to the largest number of contiguous territories you hold when you choose to end your turn. The objective is to conquer every territory on the board.

As simple as that is, it makes for remarkably good gameplay, due chiefly to the interesting variations in random geography, a simple but subtle AI, and the element of chance. DICEWARS is a wonderful example of a game that has taken advantage of chance, which can make games more fun and replayable, as Richard Garfield noted in a recent article (see Resources).

### BEYOND PONG

If DICEWARS doesn't sound like your cup of tea, consider this game, apparently just called DOUBLEGAME.

Moving the mouse left and right controls a small paddle, which is used to

keep a small ball that's in constant motion from dropping below the level of the paddle. This part is like the old classic BREAKOUT (also called BRICKOUT), but without any bricks. It's too simple to really be a game—until a second tilting platform comes into the picture with a larger ball on it. Keep the ball from rolling or bouncing off either end—also extremely simple.

Now, control both the paddle and the platform at the same time with the same left-right mouse movements. Throw in a timer to let players keep track of their own personal best and you have a fiendishly difficult action game. Even better, anyone could learn to program a game like DOUBLEGAME in their very first programming class, and it will run on almost any platform. If I had thought of this back in 1981, I expect it would have been the number one Atari VCS game that year.

These games, from Japan and France, respectively, give me considerable hope that there's still plenty of opportunity for creativity in casual games. Either game in someone's portfolio would serve to interest any worthy game company in hiring the creator.

Finally, I thank Phil Steinmeyer for bringing both of these games to my attention. You'll find the URL to his blog in the Resources. ⟡

### RESOURCES

Dicewars
**www.gamedesign.jp/flash/dice/dice.html**

doublegame
**www.zanorg.com/prodperso /jeuxchiants/doublejeu.htm**

Garfield, Richard. "Luck in Games," *Game Developer*, November 2006.

Steinmeyer, Phil.
**www.philsteinmeyer.com**

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

INDEPENDENT GAMES FESTIVAL

REWARDING INNOVATION IN INDEPENDENT GAMES
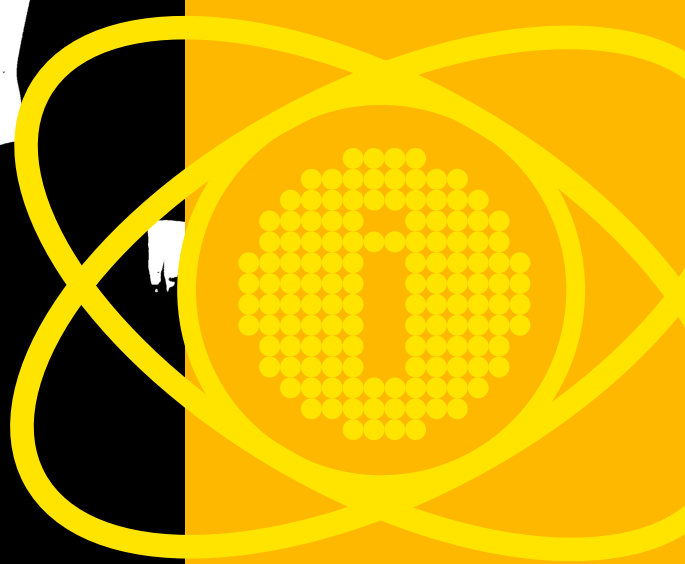
THE 9TH ANNUAL

# INDEPENDENT GAMES FESTIVAL

MARCH 7-9, 2007 • SAN FRANCISCO, CA

## CONGRATULATIONS TO THE 2007 IGF WINNERS!

PLAY THE BEST INDIE GAMES OF THE YEAR

THE IGF PAVILION
AT GDC 2007
MARCH 5-9, 2007
SAN FRANCISCO, CA

## WWW.GDCONF.COM

PLATINUM SPONSOR:

GAMETAP
EXPAND YOUR PLAYGROUND.

MOD COMPETITION SPONSOR:

PhysX
by ageia

GOLD SPONSOR:

SONY

COMPUTER ENTERTAINMENT ®

GOLD SPONSOR:

Microsoft

PLATINUM STUDENT SHOWCASE SPONSOR:

DigiPen
INSTITUTE OF TECHNOLOGY

MEDIA SPONSOR:

GAMESPOT

# WWW.IGF.COM

CMP
United Business Media

JON GOLDMAN

## » BUSINESS LEVEL

# KEEP YOUR EMPLOYEES

**KEEPING EMPLOYEE MORALE HIGH CAN** be one of the trickiest challenges for any company, but it's also one of the most important things to nail. This is especially true in the game industry, where there's a shortage of talent and plenty of options for skilled developers. Recruiting—and more importantly, retaining—your best team members is quite frankly the biggest challenge we face as independent developers.

To do that at Foundation 9 Entertainment, we need to make sure we provide an excellent place to work, with a creative atmosphere and a true sense of community. We have succeeded in some cases, but in others, there is still work to be done in order to foster the culture we want.

### LEANING TOWER OF BABEL

There's one crucial element to creating a strong working environment that's very easy to overlook when other business tasks start to pile up: good internal communications. A constant dialogue between management and all employees can make a big difference in keeping employees informed and involved in the business process, as well as happy and creative in their jobs.

Since 2005, when the company was formed, we've brought in five new studios, more than doubled the number of employees, and had some shifts in our executive structure. It hasn't always been easy for some of the employees to adjust, especially those who've been with us for awhile. These are people who we count on, both for their amazing talent and as champions of our corporate culture, so we can't afford to lose them.

In fact, we found many of our folks were simply terrified by the word "merger." Many of them had gone through mergers at other companies and came to think that layoffs were an inevitable result, even though all of Foundation 9's mergers and acquisitions have been about growing the company, not condensing it. Clearly, we needed to do a better job of communicating our goals and business strategies to our employees, particularly as the company grew and "executives" seemed more like distant individuals. Previously, we all had personal relationships with employees and colleagues. But this really becomes impossible with 750 employees. And, although our entrepreneurial sides rebelled against mass communication and "bureaucratic" announcements, keeping people in the dark was definitely not an option.

### LEARN BY DOING

To be honest, formalized internal communication at Foundation 9 was fairly non-existent until late 2006, when we started to realize how much we could alleviate fears and make everyone feel like a vital part of the company through small efforts. Our executives have always been open to conversing with employees about our general business plans, but most employees were reluctant to start these dialogues on their own.

In addition, as the company grew, so did the number of locations where we have studios. That raised another challenge in getting employees to feel like they were colleagues with everyone else in the company, especially since they work in studios all across North America, which themselves have unique cultures.

After identifying the need for better communication, we put a number of projects in place that have worked out very well for us so far.

Any large announcements, such as new mergers, acquisitions, or executive shifts are announced to the company internally via email before the news is made public, as are all press releases. We learned from our mistakes in this area: When

Foundation 9 was formed, a number of employees first learned about the merger from *The Wall Street Journal* rather than their managers! Our employees have a right to know these things before the general public, and we've been careful to make sure that happens from now on.

Another simple procedure we've put in place is to announce to the entire company when a project has passed first-party approval, as well as when it's released. This gives the individual team recognition and enables the whole company to take pride in the fact that we've successfully shipped another project. This is a change that's been hugely popular at the company.

Our most recent endeavor is a monthly newsletter, featuring spotlight articles on individual employees and projects, updates on our different studios, a calendar of birthdays, anniversaries, and game releases (and important dates from video game history), contests, and a column from me addressing questions and concerns. This newsletter has been very well received by our employees after only two issues, but I suppose no one's going to tell me they think it's a pile of fertilizer!

### BREAK DOWN THE WALL

We're not done yet. We have plans for a more comprehensive company wide intranet, including message boards for employees, as well as regular executive studio visits with informal lunches and dinners so employees can get to know upper management and have the opportunity to ask questions and make suggestions in the way they feel most comfortable.

We hope all these new efforts will result in happier and more productive teams, as well foster a sense of community across our geographically scattered studios. We know that great games don't make themselves, so we're striving hard to focus time and energy on our greatest asset—our people. Anything you can do to keep your employees happy is time well spent. ❖

**JON GOLDMAN** *is chairman and CEO of Foundation 9 Entertainment. He previously co-founded Backbone Entertainment. Email him at* **jgoldman@gdmag.com.**

You're Invited to Attend the 7th Annual

# Game Developers Choice Awards
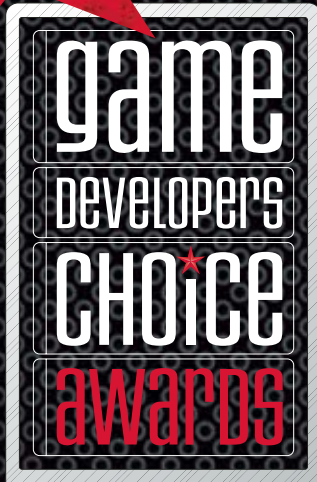
**Wednesday, March 7, 2007**
**Esplanade Room in South Hall**
**Moscone Convention Center**
**San Francisco**

**ALL GDC ATTENDEES WELCOME**

PRODUCED AND HOSTED BY
**Game**Developers
Conference

PRESENTED BY
igda *international game developers association*

Join us at the 7th Annual Game Developers Choice Awards to recognize the industry's greatest achievements.

Hosted by PSYCHONAUTS'
# Tim Schafer!

Congratulations to the best games and game developers of 2006 as voted by you!

**WWW.GAMECHOICEAWARDS.COM**

JESSE HARLIN

## >> AURAL FIXATION

# THE GREAT CONJUNCTION

**CHANGES ARE COMING TO THE FILM** world, changes that will leave a long lasting impact and that the film industry is largely not aware of yet.

Why talk about the film industry in a game professionals' magazine? Because these changes are originating from the game world and will leave an indelible mark on how both games and films are made in the next decade.

### WHERE IT STARTED

Two years ago, a new technology initiative got under way at LucasArts. The goal was to figure out a robust solution to the question of an internal development engine for the next generation of LucasArts games. The new LucasArts marched out into the world on a mission: to assess the state of the industry's technology and find the best solutions for the next generation of internal development.

We assessed middleware engines. We assessed proprietary engines. We talked to development teams in the U.S., Europe, and Asia. In the end, the solution came from a somewhat surprising source: our own backyard.

As it turned out, Industrial Light and Magic (ILM), a Lucasfilm company, had long been working with a proprietary tool called Zeno. Zeno helps ILM build and populate digital stages for their films. It contains texture controls, particle generators, and systems for digital actors. Its only drawback is that it's rooted in rendering; each frame is rendered individually in a time-consuming process, so making changes is very slow going.

But in time, LucasArts brought to the table the grand concept of runtime viewing. Two years into the collaborative development of Zeno, LucasArts is well on

its way toward the original goal of having a robust game engine populated with ILM's computer effects wizardry while ILM is gaining a robust real-time pre-visualization tool that saves time, money, and aids pre-production in the creation of advanced animatics. Both LucasArts' game developers and ILM's film teams are using the same technology with standardized terminology, a standardized workflow pipeline, and standardized tool sets.

### WHERE IT'S GOING

The environment we've been working in for the past two years is one in which many game developers may find themselves in the near future. With standardization and the cross-pollination of tools (which puts game and film design on an even technological playing field), assets can be tossed back and forth between teams instantaneously throughout production. The digitalization of filmmaking and visual effects enables the game development teams to create levels out of scenes still being worked on by the film team. And effects artists, texture artists, and animators can service the design needs of both teams simultaneously.

But how does this affect audio? The conjunction of film and game authoring technology has different implications for audio professionals than it does for our graphics-centric brethren. Film audio exists almost exclusively in a world of post-production and will most likely always be that way. Game sound is rarely a linear experience. The notion of audio simply coming in at the end of a film project, however, is misleading. As budgets balloon, an ever-increasing importance in the film world is being placed on detailed and sophisticated pre-visualizations.

The audio capabilities of current pre-visualization tools are often extraordinarily basic, if implemented at all. Usually, if a pre-visualization program has any audio functionality, it's the ability to play back one or two linear tracks of audio.

ILM began using Zeno's pre-visualization tool with that kind of limited functionality. However, the group then collaborated with the same programming staff who was creating the audio tool sets for LucasArts' audio engine, and as a result the audio capabilities of the pre-visualization tool are now more robust. It even shares similar key commands, UI design, and workflow conventions as the game audio engine.

### JOB DIVERSITY

At its heart, converging technology is efficient, economical, and dedicated to expanding professional opportunities. What's critical to remember is that, while film audio and game audio are often separate beasts, the audio pros who create content for each often aren't.

When industry toolsets do merge, the big conclusion for audio professionals will be whether they can easily hop between film and game projects without needing to learn new software. And if they can, game audio professionals will find themselves available to do pre-production work for films, as opposed to simply coming onto a project late in a post-production role.

Film audio professionals will find themselves versed in software that allows them to step out of the world of linear audio and into the world of interactive audio. All audio professionals will find themselves with a wider pool of job opportunities, which they previously would not have been able to take due to the different skill set required by each industry.

While bi-media development environments are currently relegated solely to the realms of proprietary software, commercially available options are inevitable. It won't be long before dual-use development engines find themselves industry mainstays among other more specialized programs common to both film and game development such as Maya, the Havok physics engine, and Protools. ✖

**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.*

**[SOFTWARE DEVELOPMENT WEST 2007** CONFERENCE & EXPO]

**MARCH 19-23, 2007** | SANTA CLARA CONVENTION CENTER | SANTA CLARA, CA
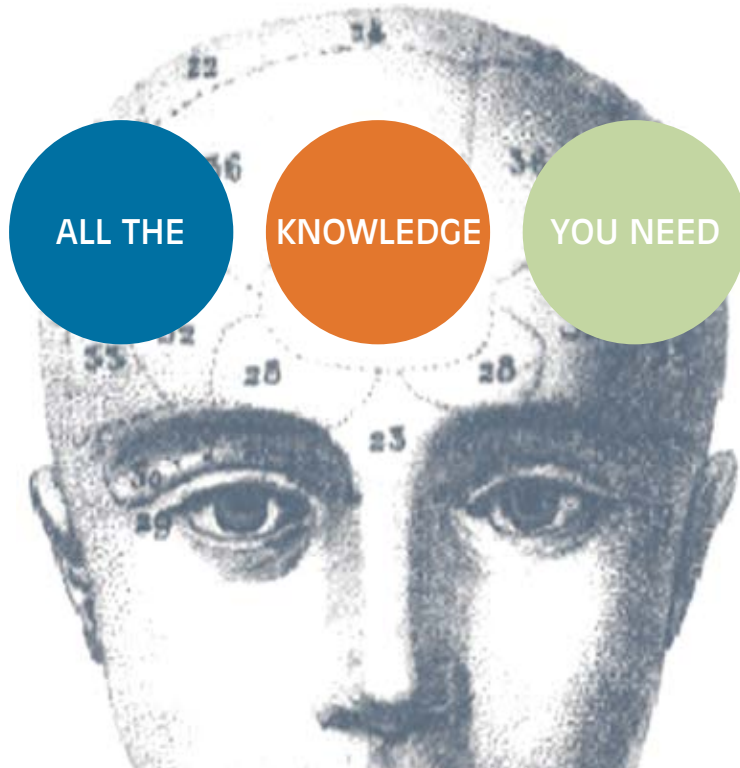
**ALL THE**  **KNOWLEDGE**  **YOU NEED**

**REGISTER**
WWW.SDEXPO.COM
**TODAY**

**20** CELEBRATING **YEARS!**

OVER 200 SESSIONS IN **14** IN-DEPTH TRACKS

- Business of Software
- C++
- Java
- .NET
- Ruby

- Modeling & Design
- People, Projects & Methods
- Requirements & Analysis

- Security
- Testing & Quality
- Web 2.0
- Web Services/SOA
- XML

**STROUSTRUP & SUTTER ON C++**
**BACK BY POPULAR DEMAND:**

Join C++ creator Bjarne Stroustrup and renowned C++ expert Herb Sutter for an in-depth two-day tutorial on C++.

NEW THIS YEAR: RUBY, WINDOWS VISTA AND WEB 2.0!

PLUS: Keynotes, Expo, Birds-of-a-Feathers, Panels, Case Studies, Roundtables, Parties and Special Events

USE THE CODE 7WGDC TO REGISTER FOR A FREE EXPO PASS AT WWW.SDEXPO.COM

**SD**
**WEST 2007**

# Disney INTERACTIVE STUDIOS

+8 Creativity
+7 Innovation
+6 Experience

# Upgrade Your Career Stats

*Visit us in the Career Pavilion in*
*Booth 1113 for opportunities to level up.*
**www.disneycareers.com**

©Disney

FALL LINE STUDIO

PROPAGANDA
GAMES

CLIMAX
RACING

AVALANCHE
SOFTWARE

# INTRIGUE.
## *EXCITEMENT.*
# INNOVATION.

### It's all in a day's work.

Want to work on a title you can be passionate about? Think you have talent? Vivendi Games wants to talk to you. Visit **BOOTH #1209** in the Career Pavilion at the Game Developers Conference in San Francisco, CA.

You can interview onsite with representatives from our studios. If you've got what it takes, you could be doing work that really turns you on at a place that's as passionate about games as you are.

| MASSIVE ENTERTAINMENT | RADICAL ENTERTAINMENT | HIGHMOON STUDIOS | SWORDFISH STUDIOS | SIERRA Online | vivendi GAMES MOBILE |
|---|---|---|---|---|---|
| Malmö, Sweden | Vancouver, BC | San Diego, CA | Birmingham/ Manchester, UK | Los Angeles, CA Issaquah, WA Santiago, Chile | San Mateo, CA |

Please contact us at roadtrip@vgames.com for more information,
And check us out at vivendigames.com

**vivendi GAMES**

# GET IN THE GAME

## CREATE THE WORLD'S GREATEST INTERACTIVE ENTERTAINMENT

## jobs.ea.com
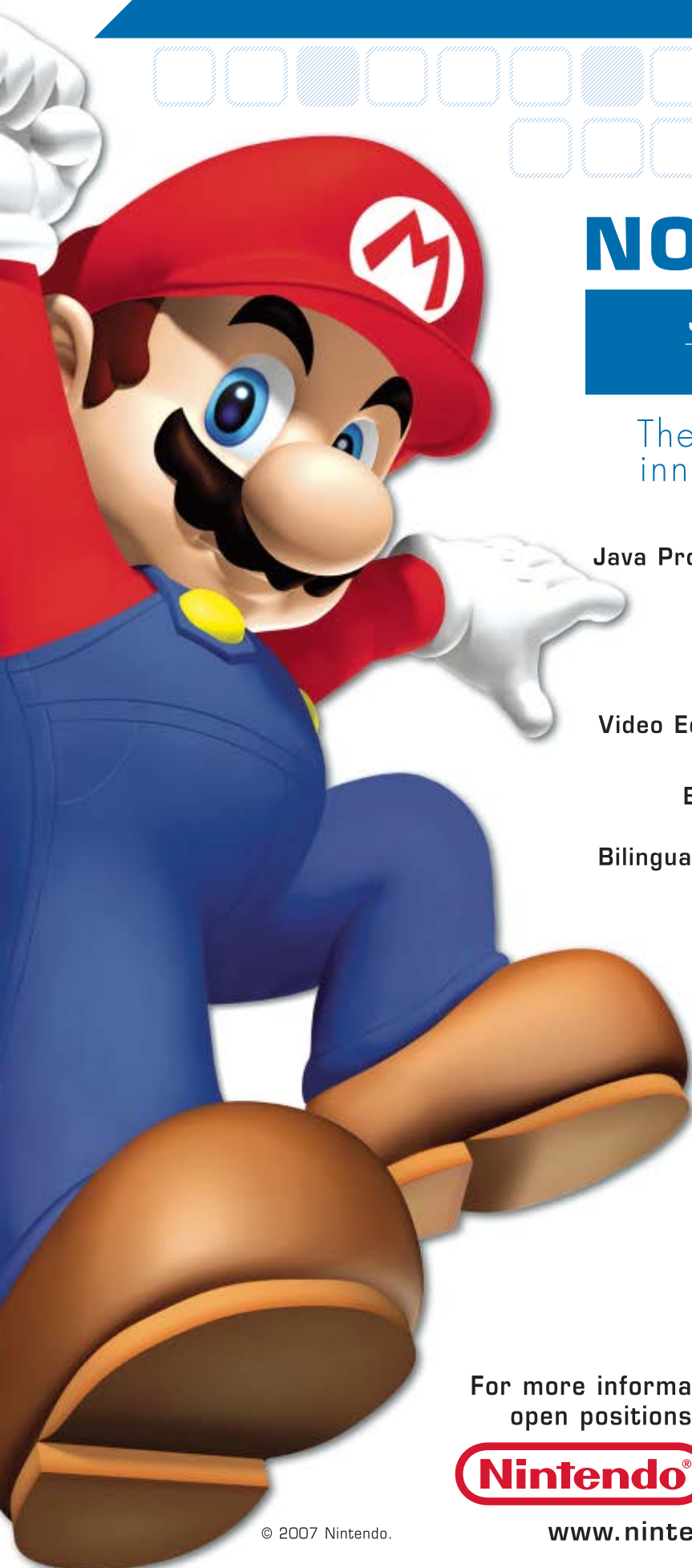
# NOW HIRING

## Jobs at Nintendo of America

### Redmond, WA

The Worldwide Leader and innovator in the creation of Interactive Entertainment

Java Programmer or Sr. Java Programmer (Virtual Console)

Translator (Japanese/English)

Senior Paralegal (Anti-Piracy)

Manager, Gateway Services

Video Editor/Marketing Support Assistant

Bilingual Software Engineer

Bilingual (French Canadian/English) Product Specialist - Localization

Bilingual (Latin American Spanish/English) Product Specialist - Localization

Manager, Public Relations

Training Specialist

Regional Merchandising Manager

Materials Lead

Electrical Engineer

Senior IT Auditor

UNIX Systems Administrator /Web Developer

Accountant

Retail Sales Anaylst

Computer Operations Specialist

Computer Operator

For more information on these positions and other open positions, please visit the websites below:

**Nintendo**®   **RETRO STUDIOS**

**www.nintendo.com/corp/jobs.jsp**
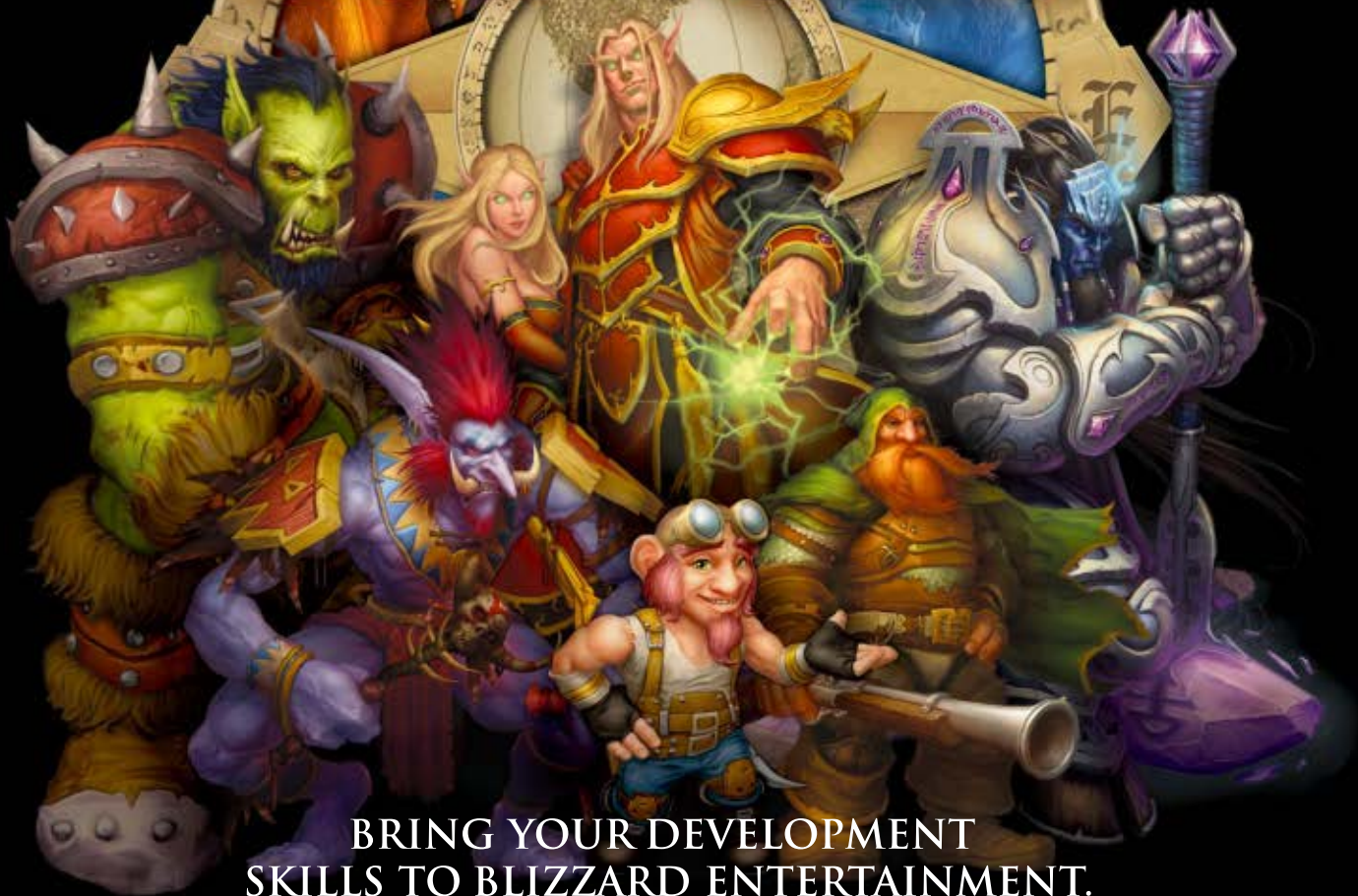
**BRING YOUR DEVELOPMENT SKILLS TO BLIZZARD ENTERTAINMENT.**

WE HAVE OPENINGS IN ALL DEVELOPMENT AREAS.
FOR MORE INFORMATION AND TO APPLY, PLEASE VISIT US AT

#1209 IN THE CAREER PAVILION - WEST HALL, LEVEL TWO

OR ONLINE AT WWW.BLIZZARD.JOBS

EQUAL OPPORTUNITY EMPLOYER.

# NOW HIRING
## SPACE CADETS

CCP, the creator of sci-fi MMOG EVE Online, is a fast-growing company that currently employs 200 people. We are looking for highly talented individuals who share our passion and drive for creating games on the leading edge of technology and game design.

We are currently looking for people to join the EVE Online team in our Atlanta, Shanghai and Reykjavik offices.

For more detailed information please visit our booth at GDC or go to www.ccpgames.com/jobs

**CCP**

# US.PLAYSTATION.COM/JOBS
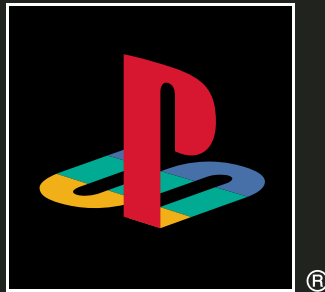
FOSTER CITY, CA • SANTA MONICA, CA • SAN DIEGO, CA • BEND, OR • SALT LAKE CITY, UT

# THE FUTURE OF VIDEOGAMES

AUSTIN

CHICAGO

LOS ANGELES

NEWCASTLE UK

SAN DIEGO

SEATTLE

Visit us at the Game Developers Conference
Booth Number 1406 or at
jobs.midway.com

JOHN WOO PRESENTS
STRANGLEHOLD

MIDWAY   TIGER HILL

RATING PENDING
RP
CONTENT RATED BY
ESRB

Visit www.esrb.org
for updated rating
information.

# ARE YOU READY FOR

Edge of Reality is now hiring

# A NEW MISSION?

## Get in the game!

Edge of Reality is staffing up for two exciting next generation projects; an upcoming title, based on a **Marvel** super hero with **Sega**, and Cipher Complex an original action title.

For more information visit **www.edgeofreality.com**

# gearbox
### s o f t w a r e ™

**CREATIVITY** UNLIMITED • UNRESTRAINED • UNBOUND
TO CREATE WHATEVER WE WANT WITH WHOMEVER **WE CHOOSE**
**DREAM PROJECTS** & ORIGINAL GAMES
THE PRODUCT OF **INDEPENDENCE** • **YOUR CHOICE**

**HAPPINESS** EVERYONE MATTERS • **YOU MATTER** • GET INVOLVED
BE **EMPASSIONED** • YOU WILL GROW • ACHIEVE **YOUR GOALS**
OUR CULTURE: **CAREERS** NOT JOBS • **YOUR PATH**

**SUCCESS** WE SHIP **BEST SELLERS** & **AWARD WINNERS**
**$10 MILLION** IN **BONUSES** PAID TO **TALENT** SINCE 2000
TOMORROW IS BIGGER • STRONGER • **BETTER** • **YOUR FUTURE**

GET IN THE BOX

WANT TO LEARN MORE? GO TO **WWW.GEARBOXSOFTWARE.COM/OPENTHEBOX**

Rise Above the Rest.

www.ravensoft.com

## About Us

Emerging Entertainment Pte Ltd., Singapore was founded in 2006 as a studio dedicated to developing next-gen games with a focus on fun for all ages. Emerging Entertainment Pte Ltd has a strong network of international partners and a team of professionals with excellent track records from all over the region and the world. Our current studio is centrally located in Singapore in a historic shophouse.

We are currently in production on a Fantasy Action Adventure using Maya and Gamebryo as our main tools, and are looking for dedicated and passionate people to join our international team. If you are ready for an exciting challenge in the center of the Asian storm, come join us. We'll do the relocation and work visa for you so it will be a smooth transition to your new life. We offer competitive salaries plus excellent medical and dental benefits that are up to international standards. Drop us a note with salary requirements plus track record, and we'll take it from there.

*Benefits we offer include:*
- Competitive salary (employees also benefit from low taxes in Singapore)
- Full comprehensive relocation coverage, including: hotel costs, apartment deposit, and relocation shipments.
- Singapore employment pass, handled by company lawyer
- Premium health plan, including dental coverage.
- Annual airfare allowance

## Executive Producer

*Job Description:*
We are looking for an Executive Producer to work in our Singapore Studio, who will be responsible for the overall management and development of those products assigned to you. It will be your key responsibility to ensure that your products are delivered on time, on budget and according to Emerging Entertainment's quality and profitability standards. In addition, the Executive Producer uses his or her experience on the front lines to improve production and organization processes, to maintain industry networking, represent the production to business partners and to manage human resource issues for Studio talent.

*Requirements:*
- Minimum 4 years of game industry experience
- At least 2 shipped console title credits
- Able to meet schedules and deadlines
- Minimum 1 year in a similar position
- Excellent problem-solving and interpersonal communication skills

## Technical Director

*Job Description:*
We are looking for an experienced Technical Director to oversee and mentor our programming team and establish the direction for technology and programming, especially with console related technical  issues. The Technical Director will work closely with the Art Director and the Project Manager to ensure that the game engine support all the game functionality and preserve the artistic integrity of our titles. The Technical Director will work hands-on directly with the team-members to resolve technical problems or optimize code and integration of technologies. He or she will also be responsible for reviewing 3rd party tools and integrating selected ones into the engine.

*Requirements:*
- Minimum 4 years of game industry experience
- At least 2 shipped console title credits
- Solid understanding of LUA scripting
- Next-Gen console experience a plus
- Able to meet schedules and deadlines
- Minimum 1 year in a similar position
- Solid understanding of C++
- GameMonkey scripting knowledge a plus
- Strong technical architecture skills
- Excellent problem-solving and interpersonal communication skills

## Art Director

*Job Description:*
We are looking for an experienced Art Director to oversee and mentor our art team and establish the direction for all art assets. The Art Director will work closely with the Technical Director and the Project Manager to ensure that the development pipeline runs smoothly and to troubleshoot problems that artists are facing. The art director will work hands-on with team members on the overall look and and style of the production.

*Requirements:*
- Minimum 4 years of game industry experience
- At least 2 shipped console title credits
- Deep understanding of styles and cultural references
- Able to meet schedules and deadlines
- Minimum 1 year in a similar position
- Thorough knowledge of Maya, MotionBuilder and other graphic tools
- Excellent problem-solving and interpersonal communication skills

## Lead Game Designer

*Job Description:*
We are looking for a Lead Game Designer to oversee and mentor our design & integration team. The Lead Game Designer will be responsible for developing and implementing core concepts, gameplay mechanics and game content in conjunction with other departmental leads; and then championing the overall game vision to other team members. The Lead Game Designer will also be responsible for conceptualizing, creating and maintaining detailed game design documentation throughout the project cycle; as well as continuously balancing and adjusting the gameplay experience to ensure the product's critical and commercial success.

*Requirements:*
- Minimum 4 years of game industry experience
- Minimum 1 year in a similar position
- At least 2 shipped console title credits
- Thorough knowledge of the major game design elements and how they should work with each other
- Thorough understanding of issues facing various types of gameplay, artistic and technical considerations; as well as the ability to forecast the ramifications of decisions made during the concept/pre-production phase
- Proven success in creating compelling gameplay, scenarios, activities and encounters
- Art and scripting (programming) experience is a definite plus
- Excellent problem-solving and interpersonal communication skills
- Able to meet schedules and deadlines

*Emerging Entertainment representatives will be present at the GDC in March. If you would like to schedule an interview, or even just to have a quick chat about the Studio, please contact Cornel at (+65)-8322-5834.*

- **Send all applications to: jobs@emerging-entertainment.com**           www.emerging-entertainment.com

# Head to Queensland, Australia.

## Be the leader of the pack

It is no secret that Queensland is the lifestyle capital of Australia. Queensland's games developers have been discovered by global publishers who are now tapping into our extensive depth of creative local talent. With over 600 people employed across 18 local developers and many more related services, Queensland is the perfect destination to progress your career.

**Find out more at GDC stand 1618.**

Queensland Government

investqueensland.com.au

SD4138 01/06

# We <3 Programmers!

**Big Huge Games is expanding!**

We have a metric ton of cool, industry-redefining work to do and many exciting positions available. Whether you're an industry n00b looking for your first fantastic job experience or a jaded veteran in need of a jumpstart at a fun, driven company, join us as we:

- Develop cutting-edge new IP across multiple next-gen platforms and genres.
- Push the development of RTS games through our award-winning *Rise of Nations*® franchise and beyond.
- Expand our Big Huge Engine Licensing program.
- Continue developing downloadable games like our upcoming version of *Settlers of Catan* for XBLA.

**Never been to Big Huge Games? No problem!**
The first ten applicants to qualify for and attend an onsite fulltimer interview (not an internship) will receive a free Nintendo Wii or Xbox 360 (your choice)! We're looking for gifted, well-rounded programmers at all levels, including internships. At the entry level, the outright requirements are:

- An unbounded excitement and focus for making games.
- A practical understanding of object-oriented programming.

**In addition, console experience would be huge.**
- XBOX 360 development? Step right up!
- PS2/PS3 development? Let's make a deal!

Please send a cover letter and resume to:
**codejobs@bighugegames.com.**

Code samples and demos are also encouraged. Code samples should demonstrate a candidate's skill in writing clear code that solves complex problems.

BIG HUGE
G A M E S

For more information please visit
http://bighugegames.com and http://bighugeengine.com

ACCEPT THE
CHALLENGE

Development

Sound Design

Programming

Animation

# Say it in game.

Take your ideas to the next level.
Join Columbia's community of intensely creative minds.

create...
change

Look for Columbia College's booth (#5607) at
**Game Developers Conference '07** in San Francisco, March 5-9.

# Columbia
## COLLEGE CHICAGO
INNOVATION IN THE VISUAL, PERFORMING, MEDIA,
AND COMMUNICATION ARTS

*GAME.COLUM.EDU*

# Make More Enemies

**Game Design at VFS** shows students how to make more enemies, better heroes, cooler levels, and tighter connections to top game companies.

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

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

**The Results.**
Our graduates work at top game companies including Backbone Entertainment, BioWare, EA Black Box, Next Level Games, and Propaganda Games.

**The Industry Connection.**
Students meet with industry pros in classes and at panels, and have access to events like the Game Design Expo – an annual event that brings the leaders of game design to Vancouver to discuss the state of the industry.

www.gamedesignexpo.com

VFS Student work by Jelmer Boskma

VFS™

vfs.com/enemies

[  **GEEKED  AT BIRTH.**  ]

**LEARN:**

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

You can talk the talk. Can you walk the walk? Here's a chance to prove it.  Please geek responsibly.  **www.uat.edu >** 877.UAT.GEEK

## ADVERTISER INDEX

# GAME DEVELOPERS EXCHANGE

# GDX 2007

**The Southeast's premier game development conference**

**DingDing Chung**, Taipei, Taiwan,
M.F.A. student in animation,
*Night Cherry Blossom*, Maya 7.0, 2005.

**April 26: Women in Games International Conference**
**April 27: GDX Conference**
**Savannah, Ga.**

GDX, presented by the Savannah College of Art and Design, brings together game developers, educators and students with renowned industry experts who share their behind-the-scenes knowledge. Participants choose from a variety of sessions, lectures and panel discussions in four game development tracks: art, technical, design and business. In addition, the Women in Games International Conference, "Advancing Your Career in Game Development" will be held on Thursday, April 26.

**Savannah College of Art and Design®**
ATLANTA · LACOSTE · SAVANNAH · E-LEARNING

**WIGI** WOMEN IN GAMES INTERNATIONAL

**www.scad.edu/gdx**

## BULLET WITCH

Cavia's BULLET WITCH, published by Atari for Xbox 360 in the U.S., stars a witch named Alicia, pictured here with her broom-like gun, in a world in which the human population is rapidly approaching extinction. Cavia has typically been a work-for-hire developer in Japan, and this is the first game in which its name appeared on the front of the original box.

**WHAT**

**DO**

**YOU**

# SEE?

People with extraordinary vision find a home for their big ideas in Orlando. That's why leading game studios, specialized industry training programs and the world's largest concentration of simulation developers make their home here. Look closer, and see why forward-thinking companies and talented workers come to Orlando to turn their visions into realities.

Call 888.867.2489
www.orlandoedc.com/whatdoyousee

Putting imagination to work®
ORLANDO