OCTOBER 2005

gamedeveloper

THE LEADING GAME INDUSTRY MAGAZINE

Thanks to our publishers for helping us create a new world of video games.

GameTap™ and many of the video game industry's leading publishers have joined together to create a new world where you can play hundreds of the greatest games right from your broadband-connected PC. It's gaming freedom like never before. **START PLAYING AT GAMETAP.COM**

GAMETAP™
EXPAND YOUR PLAYGROUND.

gamedeveloper

FEATURES

POSTMORTEM

DEPARTMENTS

COLUMNS

[GAME PLAN]

TOP CAT

AS YOU MAY WELL HAVE NOTICED FROM THE STAR
design on the cover, 12 months have slowly ticked over, and it's now time for the third annual *Game Developer* Top 20 Publishers issue. Our yearly round-up of who's top in the world of game publishing—based on not only revenue, but also release SKU numbers, average review scores, and anonymous feedback—paints a fascinating portrait of who's hot and who's hottest in the game publishing biz. And, well … we won't say anything else. Let's just let the rankings speak for themselves.

PRESIDENT ELVIS
As for a postmortem to go alongside our special Top 20 Publishers feature, we've managed to grab Yoshiaki Hirabayashi, the lead cinematics director on Capcom's critically acclaimed RESIDENT EVIL 4, to discuss some of the efforts that went into fashioning one of the best looking GameCube titles of all time.

Particularly focusing on the improved graphical development workflow, facial animation advances, and skillful use of real-time cutscenes in the chilling ratcheting-up of the sometimes sedate RESIDENT EVIL franchise, Hirabayashi reveals the workings of a title that's currently rated as one of the top five games of all time at GameRankings.com.

TALK! BANG! PIRATES!
We thought it was about time to feature another in-depth interview, and this time around we've chosen someone a little different: Daniel James, founder and CEO of Three Rings Design, a San Francisco-based developer and operator of MMO games for the mass-market casual audience.

Three Rings' independently developed casual MMO YOHOHO! PUZZLE PIRATES has garnered plenty of fans, due to the way it neatly weaves a casual community aspect and addictive multiplayer puzzle games together with wider persistent-world goals. Now, Three Rings has announced plans for a second title, BANG! HOWDY, that will be free to play but will allow players to purchase in-game items and upgrades with real-world money.

In this exclusive interview, James talks about why he feels this business model is ready to take off in the West, and the problems of distributing casual games like PUZZLE PIRATES as an executable, as well as some of his visions for the future of the so-called broadcore games that Three Rings specializes in creating.

USUAL SUSPECTS
We're firing on all cylinders through the rest of the magazine, with Heads Up Display wrapping up the proceedings of Game Developers Conference Europe, regular art, design, and business columns, and Ubisoft's PETER JACKSON'S KING KONG cutting a swagger as part of our Thousand Words art feature.

In addition, we have a new face on The Inner Product: Mick West, a co-founder of TONY HAWK'S PRO SKATER creator Neversoft, who, in his inaugural code column, discusses the problems of debug and release modes for game build configurations, and intriguingly argues that they should be replaced by a single configuration—develop mode—that should be used at all stages of development.

SITTING IN A TREE
Finally, a rumination on an old friend. Since we in the *Game Developer* office grabbed copies of NINTENDOGS and ADVANCE WARS DS, we've been happily humming a "we love Nintendo's first-party games" tune. But, unfortunately, this seems to be just about the only tune it's possible to hum when playing Nintendo consoles.

Although you might know better than I whether NINTENDOGS will be as much of a phenomenon as its currently sold-out status seems to imply, the DS, and particularly the thus-far mysterious Revolution, seem again to be setting Nintendo on an island all its own when it comes to development community outreach.

When even long-time Nintendo second-party developers, such as Factor 5, have transitioned to working on PlayStation 3 games, isn't it true that many publishers will grudgingly support The Big N because they have to, rather than because Nintendo is actively courting them?

Then again, this doesn't yet seem to have affected Nintendo's direction or profitability, or, indeed, the delightfulness of many of the game consoles or software titles the company creates. It's just, well, obtuse, and the gossiping masses of game development are wondering if Nintendo's downfall is now, or ever, imminent. Well … if it makes you feel any better, we don't know either. �֎

S!

Simon Carless, editor

GDC EUROPE SHOWS CARDS

GAME DEVELOPERS CONFERENCE EUROPE, HELD AT THE CAFE ROYAL IN London from August 30 to September 1 and organized by the CMP Game Group (also owners of *Game Developer*), showed an array of talks both technical and trend-based, but what really shone were the intriguing diversity of keynotes and the messages contained therein.

On the one hand, Sony Computer Electronic Entertainment executive vice president Phil Harrison spoke frankly and at length in a Q&A format with GDC chair Jamil Moledina, discussing both Sony's forthcoming Cell-powered PlayStation 3 and the state of play on both the PSP and the still-thriving PlayStation 2.

Pressed on the apparent success of Xbox Live compared to the PlayStation 2's online strategy, Harrison indicated that the PlayStation 3 would have significant online capabilities, but stressed its nature as an "open platform," commenting, in an obvious reference to Microsoft, "Distinct from our competitors, we are happy for publishers to make their own financial agreements directly with consumers."

In addition, Namco's Keita Takahashi, creator of the cult classic and mainstream-covered phenomenon Katamari Damacy used his keynote to comment on the state of gaming and his experiences in creating the sequel, We Love Katamari. Takahashi asked a serious question, as much to himself as to the audience: "Is it fine to continue to create games that are totally superfluous forever?" If they are as charmingly described as the cat controller concept that Takahashi won the Game Design Mash-Up roundtable with at GDCE, he should have no problem continuing to exploit his superfluousness and being adored for it.

Finally, in his keynote, Chris Satchell, general manager of the Xbox Game

Developer Group at Microsoft, discussed what he and his company envisage regarding consumer and content trends. Some of the most interesting responses came in the Q&A when, quizzed about the two options for the Xbox 360, one including hard disc, and one without, Satchell simply commented, "Both of them are gaming devices," suggesting that Xbox 360 users with a hard drive are going to be downloading and customizing their experience a great deal more through Xbox 360 Live, but that would be the main difference.

As for the naysayers regarding lack of hard drive support for Xbox 360 titles, Satchell simply stated, "The games look great, they don't need a hard drive, they're working great off DVD."

Overall, if you factor in the tutorials and game postmortems alongside GDC Mobile and multiple thriving technical and business tracks, most attendees seemed to agree that GDC's little brother is back in relatively rude health, after a couple of years of not cleaning behind its ears, despite recent confusion on the European trade and developer show scene.

The CMP Game Group has confirmed that GDC Europe will return in 2006, with the location and timing yet to be announced.

—*Simon Carless*

PHOTO BY JON JORDAN

Sony's Phil Harrison

WIGI ESTABLISHES COLLECTIVE VOICE

WOMEN IN THE GAME INDUSTRY LACK a collective voice.

Or at least they did until this summer. Women in Games International (WIGI), founded by Sheri Graner Ray, a forthright advocate for women's issues who also works at Sony Online Entertainment, is a newly formed non-profit organization that aims to unite the voices (and possibly the pleas) of women who work in games. The group was co-founded by two other highly recognizable women: Laura Fryer of Microsoft Games Studio and Kathy Schoback of Ageia.

"There are usually only about two or three women at every [game] company and they tend to feel isolated. We want

to give them a common voice," says Graner Ray. Additionally, the group will dedicate much of its energy to educating both male and female game industry professionals about how to attract women—and other under-represented populations—into the industry and keep them there.

WIGI co-founder Laura Fryer explains. "We're interested in getting more diversity. Changing the industry is an individual process, and it takes time. We're in it for the long run. Lifestyle changes take time. People want to do the right thing in this industry. People want to be more diverse. Sometimes, they are unaware of the issues."

Though panels on women's issues have

been prolific at various conferences, and special interests groups have likewise persevered, no single group has been the representative front for all the causes. The WIGI seems to be ready, almost naturally so, to fill this role, especially because Graner Ray, Fryer, and Schoback (among a handful of others) had been the people closest to doing it prior to the WIGI's formation.

Hearing strong interest from the rank and file of female game developers, the WIGI really formed out of a need, says Graner Ray. "It's kind of like we got critical mass. I'm excited that we have enough women to get this kind of movement."

"The thing that makes WIGI

different [from other women's groups and events] is it's about the role of women in the industry. ... How are we making games? What games are coming in the future?" says Fryer.

"We're really all about raising awareness of issues," says Graner Ray, citing that one of WIGI's focuses is to help women who are interested in the game industry actually make it into the career.

"We've got momentum building right now," says Fryer. "There were more women at GDC this year, and I don't know if that means there's more women just attending GDC or if there are more women in the field, but we've got the ball rolling right now."

—*Jill Duffy*

Sheri Graner Ray

Laura Fryer

QUAZAL NETWORKS FOR PSP

MONTREAL-HEADQUARTERED QUAZAL, maker of the Net-Z and Rendez-Vous online multiplayer middleware technologies, has announced that the company has been approved for the Tools and Middleware program for Sony's PSP, and that a version of its middleware is now ready for evaluation by interested PSP developers.

"Handheld multiplayer gaming is full of unique opportunities and challenges, and we're happy to have our technology meeting those challenges," says Martin Lavoie, CTO of Quazal. "Having handheld to home console cross-platform capability adds another great dimension of multiplayer for developers to work with."

By incorporating Quazal's technology into their game titles, developers and publishers can potentially open the door to novel interoperable gaming opportunities between the handheld and home console game spaces. The company claims that, because of the cross-platform nature of Net-Z and Rendez-Vous, which is also available for PCs and major console systems such as the PlayStation 2, these ideas will be far easier to implement than before.

Recent Quazal-using titles include Ubisoft's TOM CLANCY'S SPLINTER CELL: CHAOS THEORY, Crystal Dynamics/Eidos' PROJECT: SNOWBLIND, and High Moon/Capcom's DARKWATCH.

—Simon Carless

NDL, EMERGENT IN MERGER

NDL, THE MAKERS OF GAMEBRYO 3D GRAPHICS tools and engine, and Emergent Game Technologies (formerly Butterfly.net) announced in late August that the two companies will merge into one middleware provider under the name Emergent Game Technologies.

John Austin, president of NDL, commenting on the merger says, "We want developers to have tools to create better games so they can focus on the content creation."

Geoff Selzer, CEO of Emergent, adds that NDL saw eye-to-eye with his company about the implications of the middleware market. "The Renderware acquisition by EA [which happened in 2004, shortly after Selzer became CEO of Emergent], indicated two things. One: it indicated that middleware was accepted by one of the dominant publishers in games ... Two: it left a massive hole in the middleware market."

The decision to join the two companies, and ostensibly fill that hole, was spawned by the desire to be "a broader middleware company," says Austin. "We really needed to have a broader product offering."

Emergent plans to release some details regarding new product offerings at the Austin Games Conference at the end of the month. "I think the industry is going to be pretty excited about the breadth and depth of what we're going to offer in the next 16 to 18 months," says Selzer.

When asked about how smaller, independent developers will fit into the middleware market, Selzer says, "I think you'll see some creative business models out there that will enable the not-so-well-funded developers" to stay in business. Austin adds that the "key visible customers" for Gamebryo "are mid-sized ones that can really take advantage of our technologies," and include development houses that have at least one strongly established title, such as Firaxis Games (SID MEIER'S PIRATES and CIVILIZATION IV), Bethesda Softworks (OBLIVION, scheduled for release on Xbox 360), Mythic (DARK AGE OF CAMELOT), and Irrational Games (FREEDOM FORCE).

"People are going to start seeing that we're not just looking at middleware but tools ... not just providing middleware, but evolving into providing a pipeline," says Selzer, who also mentions that the company is planning Emergent's "rapid expansion in Europe and Asia."

Talk of pipelines, as opposed to specific tools, suggests that now more than ever before, game developers are concerned with workflow, and possibly creating a standardized development method, if only to keep a firm handle on larger teams, multi-million dollar projects, and entertainment technologies that evolve at staggering speeds.

"The industry is terrified of that word, 'standardized'," Selzer says. "The need for standardization falls into the fact that the industry is maturing. What comes out of that [for Emergent] is, yes, we believe in standardization, but we also believe that standardization is going to mean something else in each studio." Selzer also believes that "modular solutions" were the way of the future—or at least the way of Emergent's future. "Flexibility is going to be one of the keynotes," he says. "We don't want to confine. We want to support."

—Jill Duffy

CALENDAR

IDV INC.'S SPEEDTREE RT 1.7
BY TOM CARROLL

HENRY DAVID THOREAU ONCE WROTE,
"I frequently tramped eight or ten miles through the deepest snow to keep an appointment with a beech tree, or a yellow birch, or an old acquaintance among the pines."

And I thought gamers had no social life.

Because Atari's PONG was still 110 years away when Thoreau penned those words, how could he know that game fans would share his liking for hiking, often trudging around for hours—days!—in various MMORPGs? Or that they'd be just as picky about the look of those beeches, birches, and pines? Luckily, those trees and 27 others are found in SpeedTreeRT 1.7, the latest version of IDV's unique software that's used for creating and populating virtual landscapes with trees.

Put yourself in the shoes of any game producer or environmental art lead. You've got to fill a landscape with realistic moving flora in a timely and efficient manner, and within the ridiculously short time allotted. You have two choices.

First, you could do things the old school way, which would mean a) building a variety of trees within your 3D package of choice; b) providing for lower-poly collision objects to match each tree; c) rigging them for animation; d) producing three to five level of detail (LOD) models of each; and finally e) placing them onto a game panorama that could be as small as New York's Central Park or as big as Washington, Oregon, Wyoming, and Montana all stitched together. Most pros give up half way through step d).

Alternatively, you could fill your landscape the new school way by using SpeedTreeRT—enough said. It does it all.

THE $5 SPEEDTREE TOUR
SpeedTree actually has a family of offerings, comprising SpeedTreeRT, a C++ API; SpeedTreeCAD, a stand-alone Microsoft Windows application; Tree model library, which contains more than 170 species of trees and plants, and almost 1,000 models, representing various seasons, growth phases, and graphic densities for those species, as well as bark maps, leaf maps, and normal maps; and SpeedTreeMax and SpeedTreeMaya, plug-ins for the respective programs. Although all these tools can work in conjunction with each other, you can mix and match the components that you need to fit your development environment. This review focuses only on SpeedTreeRT.
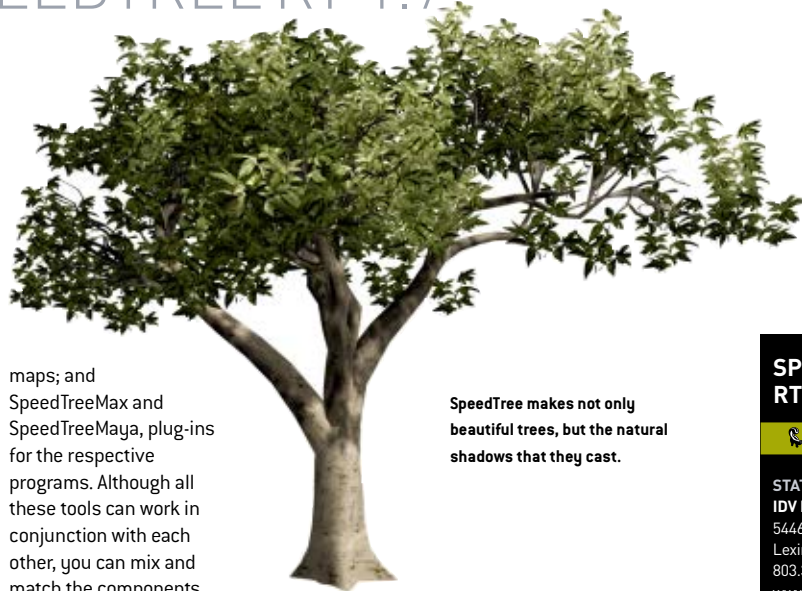
Before I begin, let me warn you that this review contains a really big spoiler. And here it is: When I first opened SpeedTree, I saw—a tree!

There was, in fact, a beautifully rendered picture of a tree in a large window.

I then realized that it was the 3D tree I would be working on, one that would enable me to edit existing SpeedTree trees or build new ones from scratch. After my initial reaction wore off (in seconds), I observed the interface box to the left of the picture and felt a bit leery of it. It had all the ambiance of something designed by Sgt. Joe Friday (he of the "just the facts, ma'am" fame): acres of gray framing, lots of selection tabs, and white numeric entry boxes, with just enough colored icons to give it some graphical appeal.

But get past the menu's utilitarian look and you find unabashed functionality that's so pure it belies the need for a manual at all.

If you haven't downloaded the free SpeedTree demo, here's my quick take on what I experienced with the package. Playing with the row of graphical icons across the top of the screen provided me with instant gratification, since I was able to see or not see various object types on my visible tree (specifically branches, leaves, and fronds). I could

also easily toggle the ground and sky textures, various light sources, and the wind source.

THE F5 TRAP
Nearly all other functions within the package require that you press the Compute (F5) button before you can see your work on the 3D tree. This was only a bother at first, since it took my teensy brain just a few tries to latch onto the fact that I wouldn't see any change until I hit F5.

Once the software modified my behavior, I started cooking with gas, partner. While some aspects of SpeedTree—most notably the Wind and LOD tabs—require you to type in values via the keyboard, others let you use curve editors instead. I took to these quite quickly because they were more fun and intuitive. With about as much accountability as a kid in a candy store, I began modifying my tree's trunk, branches, fronds, leaves, wind, lighting, and LOD values.

One quick tip: Experiment with the Global tab right away. Within the Seeds section, check Randomize for both Main seed and Flare seeds, then hit F5 several times. Each time, you'll see a randomly generated version of the tree you're working on. Randomizing allows you to quickly see how much (or little) variety a forest of your trees will have.

But I digress. Before you could say, "Miracle Gro," I had experienced all the

SpeedTree makes not only beautiful trees, but the natural shadows that they cast.

madcap hi-jinks that I'm sure most people do when they discover how easy SpeedTree is to use. I made a mutant Red Oak the size of the Sears Tower, adjusted the wind to hurricane status, and set the trunk values such that my tree bent nearly horizontal. I also increased the leaf size to such an extent that they looked like elephant ears waving on pick-up stix.

And was I ever pleased.

STEMMING BEYOND THE BASICS

Working with SpeedTree's basic trees enabled me to make flora that had more than enough verve to suit my needs, but I pushed myself still further and that's when I could tell I was making mistakes. That was when I reviewed SpeedTree's manual and the help section it provides.

Creating an entire tree from scratch is for the birds, especially if you're new to the package. You'll get going much faster by simply modifying one of the basic trees that are provided in the package. For example, when I wanted to create new foliage, I had far less to worry about if I simply substituted my custom leaf texture for the one specified in the aforementioned Red Oak.

I also found that no matter how much I tried, my hand-painted leaf textures didn't look as believable as when I started with downloaded images of real foliage. Although Photoshop allowed me to add detail and completeness to the textures, as well as the alpha channel, real leaves are the best starting point.

One of the mistakes I made more often than I'd like to admit (and you will too if it's late at night or right before a deadline) was to leave the pivot point for a 3D leaf cluster in the centered position. Once I imported the cluster and loaded it onto my tree, my leaves tended to twirl around and around instead of waving as they should do. You have to manually move the pivot point to the spot where the stems join together.

Simplicity really works well for SpeedTree. For example, I wanted to create new leaf textures for a rather complex tree that had four individual leaf types, so I took a look at how many

individual textures it required and made substitute ones. Upon loading them into the package, my new textures needed adjusting to make them appear the way I wanted. To fix the tree, I needed to make the same changes to all four leaf types. In retrospect, I should have made one leaf, loaded into all of the slots for the tree's leaves, observed the results, then, when the leaf was perfected, create three variations to match it. As they say, hindsight is 20/20.

GETTING FROM OUT HERE TO IN THERE

Once I got my custom trees in shape, it was time to put them to use in a real-time application: my game.

I had to make a composite map of all the textures my trees would use within the game world. After consulting the manual (real gamers use manuals), I opened the package's Export Composite Leaf/Frond/Billboard menu. To the left is a list of all available trees, so I selected the new ones I had created.

I was able to set the values of some variables, such as the size of the resulting composite map, and include various billboards of the finished tree to use as the lowest LOD. This menu also allows for 360 degree views of the tree to be stored to provide the most convincingly real low-level LOD.

After clicking Add, I could preview the maps required for each tree. I then clicked Create In Folder to move the composite map and all associated files to a folder of my choosing. It was that simple.

Although I didn't need to use it for my own purposes, the Export Composite Leaf/Frond/Billboard menu also lets you export single images of trees—to use as billboards, or to use within Photoshop for numerous other reasons—as well as various components for constructing a self-shadowing tree.

Once the correct files have been exported, a programmer can use SpeedTreeRT to interpret and manipulate the tree data within the framework of a real-time game engine. At this point, anything within reason is possible. For instance, those crazy guys at NCsoft have made it so people playing AUTO ASSAULT can destroy the gently waving,

beautiful trees using a wide array of mesmerizing and destructive weapons. Why not? It's their game.

In another game demo, a helicopter being flown around terrain clad in SpeedTree-style flora generates wind that makes individual trees sway in a realistic manner. Changes in prop speed are reflected in real time. Remember games that had trees looking as though they were frozen in amber? Those days are going, going, gone.

BARK UP THE RIGHT TREE

If misery loves company, so does excellence. SpeedTreeRT is the choice of so many game developers because it is simply the best at what it does: making trees … and bushes, and cacti, and, well, whatever you'd like. Think about that when you're contemplating a little eight- or nine-mile trek through the deepest snow.

Once you take in the full measure of SpeedTree, you'll realize the amount of flexibility it gives you. The only limitation is how bad you desire to do something cool—with trees, that is.

TOM CARROLL *is a 3D environment artist with Rockstar San Diego. He also tries not to abuse the English language (too much). Contact him at **tcarroll@gdmag.com**.*

3DCONNEXION'S SPACEPILOT
By Spencer Lindsay

ABOUT FIVE YEARS AGO, WHILE WORKING on a game deadline, my right wrist started to tingle. I ignored it.

A few days later the whole edge of my right hand had fallen asleep and was seemingly un-responsive. I couldn't work on my project and had to sit around testing the game instead of building it for a few days. A technical artist co-worker was using a Wacom tablet instead of a mouse for all of his 3D and 2D work and, seeing as he had no problems with his wrists, I ran out and got one. It was a

moment of discovery that you don't get very often. Suddenly, I could move around in 3D and in 2D without having to crimp my hand into some grotesque claw. I was happy. I was satisfied.

Until now.

SPACE GENIUS

Now, I've found a new gizmo, a gizmo that has again transformed my workflow and comfort in ways I never thought possible.

3Dconnexion's SpacePilot is a device that sits to one side of your keyboard and acts as a conduit into your 3D scene. It's designed to work effectively for both right-handed and southpaw artists.

After a few minutes of fiddling with it, I was able to edit and maneuver my 3D scene in a very intuitive and comfortable way.

Sleek, with display panel glowing blue, the hardware itself is both attractive and ergonomic. It looks like something Captain Picard would have embedded in his command chair. A series of six buttons across the top allows for a vast array of user settings so you can set it to have different functionality based on which task you're performing.

In addition to the user-configurable buttons across the top, there are four handy keyboard modifier keys to the left of the controller cap (Shift, Escape, Control, and Alt) that allow for fast selections and de-selections without having to return to the keyboard. This feature may seem superfluous at first glance, but after using this piece of hardware for a few hours, I found I could do all the maneuvering I needed without putting my left hand on the keyboard at all.

To the right of the controller cap is a very useful set of buttons that provide access to Top, Side, Bottom, and Front views as well as a handy Fit button that I used quite a bit while figuring out how to operate this thing; it lets you easily jump from looking at large and small scale models. Above the orthogonal Fit button is a set of tuning keys which let you adjust the sensitivity of the device, a feature that really comes in handy when learning how to use the SpacePilot.

Not only is the device itself easy to install and use, but the customer support system that 3Dconnexion provides is stellar.

PILOTING IN 3D

The SpacePilot is compatible with more than 100 applications, including 3D Studio Max, Maya, and Adobe Photoshop. Although I didn't try it out, I understand you can even use it with Microsoft Office.

Setting up the SpacePilot was a breeze, and the software tutorials that started immediately after I finished installing were extremely helpful. I had all the information I needed to start working with this thing within five minutes of opening the box.

The SpacePilot was recognized by all the applications on my system it was compatible with, and I encountered very few problems upon starting up.

However, if the computer was left idle long enough for the screen saver to come on, the SpacePilot was sometimes sluggish for a few seconds when I came back to it. Also, 3DS Max would crash if I rotated the model and hit the render button at the same time. (This last issue will no doubt be handled by a new driver release.)

LAUNCH

In order to give the SpacePilot a fair shake, I decided to use it for the full cycle of one of my modeling and animation projects. For the first few hours, I found it to be a bit disorienting. I wasn't used to being able to pan, zoom, and roll simultaneously; but once I broke through the rather shallow learning curve, and after a solid day of working with it, I found that being able to do all three of these functions at the same time sped up my modeling and animation tasks dramatically.

Being able to rotate and pan my camera while scrubbing the animation slider gives artists the ability to view and edit models and their keyframes at the same time. This is something I hadn't been able to do before without making a camera rig to attach to parts of my animation.

My animation tasks were greatly sped up by virtue of the SpacePilot because I didn't need to stop the animation, move the camera, and restart the animation. I simply scrubbed back and forth through my keyframes while keeping the camera on the object of interest.

The SpacePilot also increases modeling productivity. I was able to edit quickly and

The SpacePilot is a peripheral hardware device for artists—and indispensible for carpal tunnel syndrome sufferers.

efficiently while simultaneously rolling my work around the screen. It gives you the ability to move the model and tweak verts, animation keyframes, and shader parameters, all while maneuvering around the object like a dogfighter.

I've also noticed that the cramped feeling I usually have in my mouse hand after a few hours of modeling has disappeared.

In the projects I've used it with so far, the maneuverability and ease of use of the SpacePilot has made for some of the fastest and most accurate production I've been able to accomplish. ✖

SPENCER LINDSAY *is an independent consultant who works with game development teams and design firms. Email him at* ***slindsay@gdmag.com***.

>> t r i s t a n   d o n o v a n

TOP 20 PUBLISHERS

>> **THIS YEAR MARKS THE THIRD ITERATION OF THE** *GAME DEVELOPER* **TOP 20 PUBLISHERS RANKING,** and we're confident that while the list certainly has its share of familiar faces, it's also home to a few surprises. As the market shifts and consolidates, so too do the denizens of our ranking.

The obvious non-appearance this year is Acclaim, which lapsed into bankruptcy between last year's guide being written and published. Stronger international showings by Bandai and Capcom have knocked Koei and Empire out of the top 20, and with more acquisitions in 2005, and more likely on the horizon, our list is likely to be even more interesting as time goes by.

TRISTAN DONOVAN is a freelance journalist based in the U.K. Email him at ***tdonovan@gdmag.com***. Brandon Sheffield contributed to this article.

EA ELECTRONIC ARTS™

1. ELECTRONIC ARTS

Year formed: 1982
Headquarters: Redwood City, Calif.
Studios: Criterion (Guildford, U.K.); EA Black Box (Vancouver); EA Canada (Montreal and Vancouver); EA Los Angeles (Playa Vista, Calif.); EA Japan (Roppongi, Japan); EA Redwood Shores (Redwood City, Calif.); EA U.K. (Chertsey, U.K.); Maxis (Walnut Creek, Calif.); EA Tiburon (Orlando)

ELECTRONIC ARTS IS ONCE AGAIN, FOR THE THIRD straight year, the Top 20 Publishers' top dog. The company boasts an annual revenue of more than $3.1 billion, and EA also shows the biggest number of releases of any top 20 publisher, thanks to its commitment to multi-format releases. EA increased its internal development division with the incorporation of Criterion Software in 2004, upped its investment in cell phone games, and has growing plans to establish a studio in China. In late 2004, EA bought a 19.9 percent share of Ubisoft, which was met with strong resistance from the French publisher. EA also struck long-term deals with ESPN and the NFL, giving EA Sports'

forthcoming football titles the distinction of being the only games that may bear official player names, stats, and logos.

The company also took on retail publishing duties for HALF-LIFE developer Valve, which had previously been held by Vivendi Universal Games, and co-founded a record label, Next Level Music LLC. EA's particularly notable titles for 2004 include BURNOUT 3: TAKEDOWN, from aforementioned newly-incorporated studio Criterion, successful THE LORD OF THE RINGS and THE SIMS franchise extensions, and of course, multiple branded sports titles across multiple genres.

ACTIVISION®

2. ACTIVISION
Year formed: 1979
Headquarters: Santa Monica, Calif.
Studios: Beenox (Quebec City); Infinity Ward (Encino, Calif.); Luxoflux (Santa Monica, Calif.); Neversoft (Encino and Woodland Hills, Calif.); Raven Studios (Madison, Wis.); Shaba Games (San Francisco); Toys For Bob (Novato, Calif.); Treyarch (Santa Monica, Calif.); Vicarious Visions (Mountain View, Calif. and Troy, N.Y.); Z-Axis (Foster City, Calif.)

A HIGH NUMBER OF RELEASES AND BOOSTED REVENUES HAVE PROPELLED
Activision to number two on our list, well up from last year, on the back of strong releases such as the critically acclaimed ROME: TOTAL WAR, a second iteration of the TONY HAWK'S UNDERGROUND series, and a host of successful license tie-ins, from X-MEN LEGENDS to SPIDER-MAN 2. Armed with years of know-how as the first ever third-party publisher and a healthy bank account, Activision has also been expanding its already sizeable roster of internal studios during the past year. As a result, Toys For Bob, Beenox, and Vicarious Visions have all

been added to the Activision empire. Even after those purchases, the company, which has joined the ranks of publishers boasting an annual turnover in excess of $1 billion, has more than $700 million in cash to play with, proving that Activision's policy of leaving studios more autonomous after purchasing them works just as well as EA's consolidation method. Always adventurous, Activision released games on even the most obscure of consoles, with games licensed for the now-defunct Tapwave Zodiac PDA game system and Nokia's N-Gage.

Microsoft game studios™

3. MICROSOFT GAME STUDIOS
Year formed: 1975
Headquarters: Redmond, Wash.
Studios: Bungie Studios (Redmond, Wash.); Ensemble Studios (Dallas); FASA (Redmond, Wash.); Rare (Twycross, U.K.); Microsoft Game Studios Japan (Tokyo); Redmond, Wash.

SINCE THE LAUNCH OF THE XBOX, MICROSOFT'S RISE UP THE TOP 20 HAS
been swift. In 2003 it came in respectably at ninth, but has since pushed its way up the charts. Aided by smash hits like HALO 2, Microsoft is increasingly focused on Xbox titles, and has arguably focused less on its PC game releases, though some, such as RISE OF NATIONS and ZOO TYCOON 2, still thrive. Because the Xbox has continued to struggle with market penetration in Japan, a significant part of the market is largely closed to Microsoft titles (though the Microsoft Game Studios Japan-developed title PHANTOM DUST was critically appreciated in all markets). With the impending launch of the Xbox 360, Microsoft will increasingly focus its efforts in that direction, also signing high-profile deals with several Japanese developers, such as FINAL FANTASY

Publisher	Rank			Score	No. of releases	Percent of externally developed titles	Producer rating	Milestone payment rating	Pay and perks	Average game review score
	2005	2004	2003							
Electronic Arts	1	1	1	254	126	17	8.52	8.50	8.91	81
Activision	2	7	4	229	76	63	9.25	9.60	9.40	65.40
Microsoft	3	2	9	227.5	20	60	9.60	9.85	10.00	80.56
Nintendo	4	10	3	219	69	32	8.50	10.00	8.50	76.08
Sony Computer Entertainment	5	3	2	213	41	37	9.18	9.55	8.50	75.07
Ubisoft	6	5	12	205	65	55	9.48	8.05	9.25	70.25
Konami	7	15	8	203	78	15	8.80	10.00	8.50	67.03
THQ	8	4	13	199.5	94	55	9.31	9.42	8.50	65.68
Sega Sammy	9	14	10	198.5	75	31	8.65	7.60	9.40	70.55
Take Two	10	8	6	195.5	42	67	9.40	7.60	8.50	69
Namco	11	20	16	194.5	35	51	9.10	9.40	10.00	74.30
Vivendi Universal Games	12	11	5	180	43	72	8.73	8.92	9.00	68.93
Atari	13	9	7	174.5	67	70	8.84	7.68	8.20	67.63
SCi/Eidos	14	6	19	167	35	51	9.70	9.70	8.50	70.94
Capcom	15	–	14	142	39	21	8.50	8.50	8.50	73.39
Square Enix	16	16	11	134.5	19	21	8.50	8.50	8.50	71.90
Bandai	17	–	15	123	49	51	8.50	8.50	8.50	63.44
Codemasters	18	12	–	121.5	17	29	7.60	9.40	8.50	75.86
Midway	19	17	20	121	22	32	8.50	8.50	8.50	78.20
LucasArts	20	–	–	115.5	9	89	8.50	8.50	8.50	81.60

creator Hironobu Sakaguchi's new company WindWalker and Tetsuya Mizuguchi's Q Entertainment.

While Microsoft has some important studios under its wing— Bungie, Ensemble, and Rare—it continues to use external studios heavily, with outsiders producing around 60 percent of its titles.

4. NINTENDO

Year formed: 1933
Headquarters: Kyoto, Japan
Studios: Brownie Brown (Tokyo); Intelligent Systems (Kyoto, Japan); NDCube (Tokyo); Nintendo Entertainment Analysis and Development (Kyoto, Japan); Nintendo of America (Seattle); Retro Studios (Austin); Systems Research & Development (Kyoto and Osaka, Japan)

ANOTHER YEAR OF STRONG SALES FOR ITS SOFTWARE, AND THE SUCCESS OF the DS and Game Boy Advance show that Nintendo can still hold its own against Sony and Microsoft. Software for the DS in particular has been strong, and Nintendo continues to develop its POKEMON franchise on the Game Boy Advance and elsewhere, which has boosted the company back into our top five. Though Nintendo certainly likes to reuse its characters, there's no shortage of innovation from the granddad of Japanese game developers. The DS was a bold hardware-based move, and Nintendo has followed that with bold, critically acclaimed software, such as ELECTROPLANKTON and NINTENDOGS, also strengthening revenue by virtue of first-party game dominance. Internally, there have been some changes: the firm has merged its main Kyoto-based studios into a single entity, Nintendo Entertainment Analysis and Development. This brings the makers of WARIO WARE together with the makers of THE LEGEND OF ZELDA, which could potentially produce some interesting properties.

5. SONY COMPUTER ENTERTAINMENT

Year formed: 1993
Headquarters: Tokyo
Studios: Bend, Ore.; Cambridge, U.K.; Contrail (Tokyo); Foster City, Calif.; Incognito Entertainment (Salt Lake City); Liverpool, U.K.; London; Polyphony Digital (Tokyo); San Diego; Naughty Dog (Santa Monica, Calif.); Seoul; SN Systems (Bristol, U.K.); Tokyo; Zener Works (Tokyo)

THE ARRIVAL OF THE PSP AND THE EVENTUAL RELEASE OF THE PLAYSTATION 3 have prompted a small expansion in Sony's impressive collection of internal development studios. To help produce games for the PSP, it set up a studio in South Korea and more recently purchased British middleware developer SN Systems to aid PlayStation 3 game creation. Interestingly, as the console hardware market leader, Sony's internal development has been largely original IP-focused. Games such as the critically acclaimed GOD OF WAR and RATCHET AND CLANK 3 have served to boost Sony's credentials as a triple-A developer and publisher, but without the sheer software revenue volume of an EA, and releasing on fewer SKUs than the third parties, Sony slips to number five this year.

6. UBISOFT

Year formed: 1986
Headquarters: Montreuil-sous-Bois, France
Studios: Annecy, France; Barcelona; Blue Byte (Düsseldorf, Germany); Bucharest; Casablanca; Milan; Montpellier, France; Montreal; Quebec City; Paris; Red Storm (Morrisville, N.C.); Shanghai; Wolfpack (Austin)

WITH 13 STUDIOS AND ABOUT 3,000 EMPLOYEES, UBISOFT'S INTERNAL development setup is one of the largest in the world. The company has long been keen to reduce its reliance on external development, and the continuing expansion of its internal studios in Quebec and Shanghai will, in the short term, doubtless help. The company also fended off an acquisition attempt from American rival EA, with several potential methods floated before ultimately settling on restructuring, and a push toward further expansion, to that end buying up the development division of Canadian publisher Microids, and announcing a new line of sports titles.

The company continues to intelligently balance original IP, such as RAYMAN and PRINCE OF PERSIA: THE WARRIOR WITHIN, with key licenses, such as the Tom Clancy line of games.

7. KONAMI

Year formed: 1973
Headquarters: Tokyo
Studios: Hudson Soft (Tokyo, Sapporo, San Francisco); Konami Computer Entertainment (Tokyo); Konami Digital Entertainment (Honolulu, San Jose); Konami Software Shanghai (Shanghai); Kojima Productions (Tokyo)

AFTER YEARS OF KEEPING ITS INTERNAL DEVELOPMENT GROUPS SEPARATE, Konami has rolled its Japanese studios into a single Tokyo-located entity. Improved efficiency is the main impetus behind this move, Konami claims,

Titles by platform (percent)				No. of internal studios
Computer	Console	Handheld	Mobile	
17.46	66.67	14.28	1.59	10
39.48	39.48	18.43	2.63	10
10	85	5	0	6
0	24.64	75.36	0	7
0	78.05	19.51	2.44	14
30.77	58.46	10.77	0	13
10.26	61.54	23.07	5.13	6
20.21	32.97	22.34	24.47	11
16	66.66	11.99	5.33	7
7.14	88.10	4.76	0	13
0	71.42	28.57	0	2
37.21	37.21	20.94	4.65	6
32.84	44.78	20.89	1.49	6
28.57	65.57	3	2.86	5
2.56	64.10	25.65	7.69	4
15.79	31.58	15.79	36.84	5
28.57	60.57	8	2.86	3
23.53	76.47	0	0	2
9.09	81.91	9	0	6
33.33	66.66	0	0	1

which doesn't seem to have slowed it down any. The company also divested itself of all interest in Japanese niche game company and toymaker Takara (through stock owned by Atlus), which went on to merge with fellow toymaker Tomy. As a result, Konami now owns just three Konami-branded studios and three under the brand of Hudson, a developer and publisher it acquired in April 2005.

These studios pump out games at a regular pitch, so just 15 percent of Konami's games originate outside its own studios. Konami's American publishing arm has been working hard to increase that number, though, bringing externally-developed games such as Death Jr. to consoles, while maintaining popular internal series such as Metal Gear Solid, Castlevania, and Pro Evolution Soccer/Winning Eleven.

8. THQ

Year formed: 1989
Headquarters: Calabasas Hills, Calif.
Studios: Blue Tongue Entertainment (Melbourne); Concrete Games (San Diego); Cranky Pants Games (Kirkland, Wash.); Heavy Iron Studios (Culver City, Calif.); Helixe (Burlington, Mass.); New York; Rainbow Multimedia Group (Phoenix); Relic Entertainment (Vancouver); THQ Australia Studios (Spring Hill, Australia); THQ Wireless (Calabasas Hills, Calif.); Volition (Champaign, Ill.)

DESPITE SLIPPING BACK DOWN THE TOP 20 A LITTLE, THQ HAS BEEN building its industry presence greatly during the past 12 months. It now boasts 11 internal development studios, including the recently acquired Blue Tongue Entertainment, and is now the Top 20's second most prolific publisher—only Electronic Arts beats THQ in this department.

THQ is one of the foremost developers with interest in Australia, with two studios in the up-and-coming area, and is still best known for its licensed titles, publishing everything from Sega- and Rare-branded Game Boy Advance titles, to its lucrative Nickelodeon and Pixar licenses and Game Boy Advance titles, through major wrestling licenses. Even so, the company is moving further into original IP, Destroy All Humans! being one of the more notable, thanks to developer Pandemic.

9. SEGA SAMMY HOLDINGS

Year formed: 1952 (Sega); 1975 (Sammy)
Headquarters: Tokyo
Studios: Creative Assembly (Horsham, U.K. and Fortitude Valley, Australia); Racing Studio (Birmingham, U.K.); Red Entertainment (Tokyo); Sega Shanghai Studios (Shanghai); Sega Studios (Tokyo); Sega Studios USA (San Francisco)

WITH THE MERGER WITH SAMMY DONE AND DUSTED, SEGA HAS UNDERGONE some serious changes in a bid to get the company back to profitability after a decade of losses. Among the changes were the sale of Kush Games and Visual Concepts to Take Two, the purchase of Creative Assembly (developer of Rome: Total War), the loss of high-profile developer Tetsuya Mizuguchi, who formed his own studio (Q Entertainment), and the merging of Sega's seven Tokyo studios. In addition, Sammy's U.S.-based internal development team Sammy Studios became an independent company called High Moon Studios, and Sega's American and European management was merged. Finally, Sega has begun putting together a U.K.-based studio dedicated to developing racing games. But while the restructuring has been significant, troubles remain. In Japan the company was hit by a tax-evasion scandal and the death of a customer at one of its amusement arcades. Even so, the merger has largely been successful, with Sega back in the black, floated by tie-ins with Sammy's lucrative pachinko business and somewhat more cautious development and publishing choices. Sega loves original IP and has continued its Sonic, Shining Force, and Phantasy Star brands to decent success, and in this year of restructuring, has spent more time re-releasing its back catalog, and developing unique licensed IP.

10. TAKE TWO INTERACTIVE

Year formed: 1993
Headquarters: New York
Studios: Cat Daddy Games (Bellevue, Wash.); Frog City (San Francisco); Indie Built (Salt Lake City); Kush Games (Camarillo, Calif.); PopTop Software (Fenton, Mo.); Rockstar Leeds (Leeds, U.K.); Rockstar North (Edinburgh); Rockstar San Diego (San Diego); Rockstar Toronto (Toronto); Rockstar Vancouver (Vancouver); Rockstar Vienna (Vienna); Venom Games (Newcastle, U.K.); Visual Concepts (San Rafael, Calif.)

WITH THE MULTI-MILLION SALES OF Grand Theft Auto: San Andreas (12 million at the last count), Take Two has enjoyed another bumper year. The cash from its flagship series has fueled the buyout of several developers,

methodology

So real it renders fear.

Idea:
Create the most gripping and realistic stealth action game on the market.

Realized:
Ubisoft™ modeled and animated the realistic characters and backgrounds of *Tom Clancy's Splinter Cell® Chaos Theory*™ with Autodesk's 3ds Max to build on one of the most popular series ever. The work-horse capability of 3ds Max helped Ubisoft stay on top of their grueling production schedule and garner a 9.9 out of 10 by *Official Xbox Magazine*. To see how Autodesk can help you realize your most innovative ideas, visit www.autodesk.com

Autodesk®

including Venom Studios and Sega's Visual Concepts and Kush Games. The expansion has prompted the creation of two new publishing labels for the firm—2K Games and 2K Sports—to add to its existing ones: Rockstar Games and Global Star. But there has been some bad news too. The "Hot Coffee" mod for SAN ANDREAS, which allows players to access previously inaccessible sex scenes, has forced a recall of the game. An SEC investigation into the publisher's accounting has ended with Take Two paying out a $7.5 million civil penalty, while neither denying nor admitting any wrongdoing. Our current methodology lumps reviews for any given company's titles into one group, and Global Star's value brand has hurt Take Two here, as budget titles are generally reviewed less well. These factors account for Take Two's slip in the Top 20, but as the company aggressively turns out its well-rated 2K Sports brand with content from VISUAL CONCEPTS, and continues pushing the boundaries of the M rating with innovative original IP, next year is anyone's game.

11. NAMCO
Year formed: 1955
Headquarters: Tokyo
Studios: San Jose, Calif.; Yokohama, Japan

NAMCO SEEMS TO BE AGGRESSIVELY PURSUING TWO VERY DIFFERENT
directions with its two studios. First, the Japanese branch seems keen on developing original IP, such as TAIKO NO TATSUJIN and KATAMARI DAMACY, both cult hits in the U.S. as well as their native Japan. The company also fleshes out its established series whenever possible, with new iterations in the TEKKEN and RIDGE RACER series, the latter of which seems to show up as a launch title for every new console.

Second, the U.S. division is ramping up its publishing business with a number of deals in the children's sector. The company has supplemented this with the creation of a handheld game division at its San Jose studio and a significant push to improve its presence in the PC games market. The company's efforts in these various areas have boosted them up the Top 20 into a stronger position. Also of note: Namco announced this year that it will be merging with Japanese toy, game and animation maker Bandai, a deal that will be official in late 2005, and will marry Namco's internal original IP development with Bandai's marketing power and multimedia approach.

12. VIVENDI UNIVERSAL GAMES
Year formed: 2000
Headquarters: New York
Studios: Blizzard Console (Aliso Viejo, Calif.); Blizzard Entertainment (Irvine, Calif.); Massive Entertainment (Malmö, Sweden); Radical Entertainment (Vancouver); Sierra Entertainment (Bellevue, Wash.); Swordfish Studios (Birmingham, U.K.)

VIVENDI UNIVERSAL GAMES HAS SPENT THE PAST YEAR BOLSTERING ITS
internal development activities with a hat trick of buyouts—Radical Entertainment (most notable for reviving *The Simpsons* brand in video games), Swingin' Ape Studios (now Blizzard Console), and Swordfish Studios. Coupled with the massive success of WORLD OF WARCRAFT both in the U.S. and internationally, things are looking up for the firm. The rather bitter, lawsuit-spawning end of its relationship with HALF-LIFE developer Valve is a bit of a blow, but even so, while a relatively new entity, Vivendi Universal remains a potent player, and studio names such as Blizzard only strengthen its position.

13. ATARI
Year formed: 1983
Headquarters: Lyon, France
Studios: Atari Interactive (Beverly, Mass.); Atari Melbourne House (Melbourne, Australia); Eden Studios (Lyon, France); Paradigm Entertainment (Dallas); Reflections (Newcastle, U.K.); Shiny Entertainment (Newport Beach, Calif.)

STILL STRUGGLING TO COPE WITH A CASE OF INDIGESTION AFTER ITS LATE
1990s buyout bonanza, Atari has steadily been descending the Top 20. Though Humongous Entertainment was still internal for the financial year ending March 2005, Atari sold Humongous to parent company Infogrames for $10.5 million in April. As has been its modus operandi for the past few years, Atari is still largely relying on key, potentially large-selling titles such as THE MATRIX: PATH OF NEO, and is banking on its original IP (MISSILE COMMAND, CENTIPEDE) for use in a range of retro compilations.

14. SCI/EIDOS
Year formed: 1988 (SCi); 1990 (Eidos)
Headquarters: London
Studios: Beautiful Game Studios (London); Core Design (Derby, U.K.); Crystal Dynamics (Palo Alto, Calif.); IO Interactive (Copenhagen); Pivotal Games (Bath, U.K.)

WHILE SMALL UK PUBLISHER SCI FAILED TO MAKE THE TOP 20 IN 2003 AND
2004, its ambitious takeover of Eidos (a far bigger company) has propelled it to global attention. Since the merger has only just gone through, SCi is still working out the new structure for the firm, but so far Eidos' four studios and SCi's Pivotal Games (CONFLICT: DESERT STORM) remain in place. But one studio that won't be part of the new venture is Ion Storm, the former Eidos studio that was shut down prior to the merger. Since a joint revenue figure for the post-merger company is not yet available, we calculated the group's revenue by adding the last year-end results from both firms. SCi/Eidos has also made another major shift, though this one is internal—the key TOMB RAIDER franchise has made a lateral move from original developer Core Design to Crystal Dynamics, a team with more contemporary critical acclaim after the popular LEGACY OF KAIN series.

15. CAPCOM
Year formed: 1979
Headquarters: Osaka
Studios: Clover Studios (Osaka); Studio 8 (Sunnyvale, Calif.); Team 1 (Osaka); Team 2, (Osaka);

CAPCOM HAS ALWAYS ROOTED FOR THE UNDERDOG, WHICH LED TO THE
company's rash of exclusive titles for the GameCube throughout the life of the console. The years 2004 and 2005 marked a generic end to this exclusivity, though, as the tactic was hurting Capcom financially. As such, critically lauded titles RESIDENT EVIL 4 and KILLER 7 have made their way from GameCube exclusives, to a PlayStation 2 port for the former, and PlayStation 2 simultaneous launch for the latter. Capcom has also been investing in other developers, partnering with Grasshopper Manufacture for the development of KILLER 7 and Nudemaker for TEKKI. The Japanese parent is now trusting the U.S. branch far more than in the past, granting the FINAL FIGHT license to Capcom USA's Studio 8. Capcom has also become increasingly bold when it comes to publishing the work of others. It has brought the GRAND THEFT AUTO series to Japan, and with it, the wrath of

Badda bing, Badda boom

For every action in games there is an equal and opposite reaction.

- Newton's Third Law of PhysX

[momentum]

ageia
make it real.

The AGEIA PhysX™ platform provides the most powerful and advanced tool set for creating dynamic physical environments. With unmatched support for collision detection, fluids and advanced multi-threading for massively parallel architectures, your games can feature the next level in immersive game play.

AGEIA PhysX has you covered for today's software effects and tomorrow's game play with powerful hardware-based physics. Learn why more than 60 game development studios use AGEIA PhysX technology.

Visit us at www.ageia.com to learn more.

Kanagawa prefecture, in the form of a regional ban. Capcom's increase in multi-SKU games, combined with solid reviews and a firm dedication to original IP keeps the company in the Top 20, but relatively low revenue and a slightly reduced number of releases keeps the firm from the top 10.

16. SQUARE ENIX

Year formed: 1975 (Enix Corp., as Eidansha Boshu Service Center); 1986 (Square Co. Ltd.)
Headquarters: Tokyo
Studios: Beijing; Los Angeles; Osaka; Tokyo; UIEvolution (Bellevue, Wash.)

THE PAST YEAR HAS SEEN SQUARE ENIX INCREASING ITS EMPHASIS on online and mobile games through the purchase of U.S.-based network applications business UIEvolution and the creation of a mobile phone and network games studio in Beijing. The company, whose 2004 flagship title was DRAGON QUEST VIII, has also made a significant investment in development talent by continuing the Digital Entertainment Academy, a division that runs game design schools in Japan. In addition to game and middleware development, Square Enix publishes comics and books in Japan and has a licensing division for its game brands (the most notable of which are DRAGON QUEST and FINAL FANTASY). China is also high on Square Enix's agenda, and the company has been quick to establish a presence in the country.

Remakes and re-releases were a big part of the company's last year, and history is clearly important to Square Enix, as the company made a bid in August 2005 to integrate publisher and developer Taito, Japan's first arcade game maker and owner of the SPACE INVADERS IP. Taito is known for being a more niche and arcade-based publisher, while Square Enix is firmly in the RPG camp, and anything outside of that realm, up until now, has been little more than dalliance.

17. BANDAI

Year formed: 1950, as Bandai-ya
Headquarters: Tokyo
Studios: Banpresoft (Tokyo); Bec Co. Ltd. (Tokyo); Tokyo

PLAYING IS BELIEVING.™

THOUGH NOT IN LAST YEAR'S TOP 20, BANDAI HAS HAD A VESTED INTEREST in the industry ever since the Nintendo Entertainment System days, with its million-plus selling game MUSCLEMAN-MUSCLE TAG MATCH. The company is well-known for its animation and toy divisions in America, but is not as big of a player in the U.S. game space. Bandai, like Sega and Nintendo, has launched several home consoles, from its ambitious handheld Wonderswan line, to the recent Western-targeted Bubble edutainment system.

Aside from the Bubble, though, no Bandai console has ever seen a Western release. In the game publishing arena, Bandai has a penchant for external developers, owning only a few internal teams. Instead, the company either relies on third parties or invests in them heavily for future collaboration. This lack of a large internal team is likely one of the prompts for the company's merger with Namco, in which each company will profit from the strengths of the other, in a fashion not wholly dissimilar from Square Enix's model. Key Bandai titles are generally based on Japanese anime, from *One Piece* to *Inu Yasha* to *Cowboy Bebop*, but since the company also makes anime (*Digimon*), some of this work could be considered original IP.

18. CODEMASTERS

Year formed: 1986
Headquarters: Leamington Spa, U.K.
Studios: Leamington Spa, U.K.; New York

THE DECISION BY VENTURE CAPITAL FIRM BENCHMARK CAPITAL EUROPE to buy 40 percent of Codemasters has ushered in several changes at the veteran U.K. publisher. First off, co-founder David Darling has handed the reigns to Rod Cousens, the former head of Acclaim. In addition, the company has been bracing itself for the next generation of consoles with the help of a multi-million dollar war chest to buy into the games that it hopes will take it into the big league. While the company did not release many games in the financial period analyzed, those titles it did release were reviewed quite well, COLIN MCRAE RALLY 05 and TOCA RACE DRIVER 2 being particularly appreciated within their genre.

19. MIDWAY

Year formed: 1988
Headquarters: Chicago
Studios: Adelaide, Australia; Austin; Chicago; Los Angeles; San Diego; Surreal Software (Seattle)

SINCE *GAME DEVELOPER*'S LAST TOP 20, MIDWAY HAS BEEN BUSY SNAPPING up developers. Beginning in 2004, Midway added Surreal Software, Inevitable Entertainment (now Midway Studios Austin), Paradox Development (now the Los Angeles studio), and most recently Australian studio Ratbag Games (now Midway Studios Australia). On top of this Midway has expanded its international operations, opening a sales and marketing office in Germany in order to get a bigger slice of Europe's biggest market. Midway has been working more and more in the field of original IP, publishing the company's own properties, MORTAL KOMBAT and THE SUFFERING, as well as putting out Japanese RPG SHADOW HEARTS: COVENANT in the U.S. Midway was once well known for its BLITZ and JAM lines of sports titles, and with the new NFL deal to EA, is now looking to revive its alternative, non-licensed sports lines in a big way.

20. LUCASARTS

Year formed: 1982
Headquarters: San Francisco
Studios: San Francisco

AFTER SOME SHAKY MOMENTS IN 2004, LUCASARTS HAS COME BACK with a vengeance. Last summer the company cut back its San Francisco-based development studio, laying off 31 people and reducing the amount of games in its pipeline. The cancellation of fan favorite SAM AND MAX 2 added to the gloom. But a year later, the games division of George Lucas' privately-owned Lucasfilm empire seems to have a new lease on life. Its access to the *Star Wars* franchise was, predictably enough, central to this recovery, with several smash hits including STAR WARS BATTLEFRONT, the biggest-selling *Star Wars* game of all time. LucasArts has also pledged to release more original products, doubtless encouraged by the strong sales of MERCENARIES. Despite few releases, the company's success is not just a story of sales, since it boasts the most critically acclaimed games of any Top 20 publisher, scoring an average of 82.2 percent, though this is spread out across the fewest releases. ✖

Simply cool.
Mobile gaming with 3D.

```
int n=0;
for(int i=2; i<NUM_POINT_SPRITES*8; i+=8){ // ROTATION
 TEXTURE[i] = TEXTURE[i] >= 4096?0:TEXTURE[i] + (int)((4096/
360)*ROT[n]);
n++;
}
for(int i=0; i<NUM_POINT_SPRITES*3; i+=3){
 POINT[i+1] += 1; // change y pos
}
g3d.renderPrimitives(floorTexture, 0, 0, layout, effect,
QUAD_COMMAND, 1, floorPOINT, floorNORMAL, floorTEXTURE,
color);
g3d.flush();
g3d.renderPrimitives(brickTexture, 0, 0, layout, effect,
QUAD_COMMAND, 1, wallPOINT, wallNORMAL, wallTEXTURE, color);
g3d.renderPrimitives(smoke_texture, 0, 0, layout, effect,
POINT_SPRITE_COMMAND, NUM_POINT_SPRITES, POINT, color,
TEXTURE, color);
g3d.renderFigure(barrelFigure, 0, 0, barrelLayout, effect);
```

It's easier than you may think to add cool 3D effects to your mobile games with 3D sprites and transparency. For the latest and greatest in mobile Java 3D development, go to *www.SonyEricsson.com/developer/java3d*.

Creating cool 3D effects in your mobile games is easier than you may think.

At Sony Ericsson Developer World you'll find everything you need to get up to speed in mobile Java™ 3D development. From technical documentation to responsive tech support. Not to mention mobile plug-ins for Mascot Capsule v3 and JSR-184 (m3g) that let you use your favorite tools like 3D Studio Max, Maya and Lightwave. And while you're exploring all the new dimensions of mobile game development, Sony Ericsson Developer World is there to help you on the fast track from mind to market.

Go to www.SonyEricsson.com/developer/java3d to get the best in mobile Java 3D development. And to find out just what mobile Java 3D can mean for your games and your business.

Sony Ericsson

www.SonyEricsson.com/developer

A PIRATE'S LIFE

INSIDE THREE RINGS' STUDIO WITH DANIEL JAMES

>> **YOHOHO! PUZZLE PIRATES IS A MASSIVELY MULTIPLAYER** online game that thrives in an atypical crowd that includes teenagers, middle-aged women, and disgruntled ex-hardcore gamers, all participating together in the same online society. San Francisco-based Three Rings Design, which owns PUZZLE PIRATES, has carved out a unique business model to sustain its employees and its flagship game despite the disparate player base. *Game Developer* recently caught up with Daniel James—designer of PUZZLE PIRATES and CEO of Three Rings—to discuss the evolving business model of online games, the future of MMOs, and Three Rings' newest (though still in development) game, BANG! HOWDY.

Brandon Sheffield: *Can you tell us some details on the* PUZZLE PIRATES *launch?*

Daniel James: We started PUZZLE PIRATES with three people: two engineers and myself. Within the first year, we ramped up to six: three engineers, two artists, and me. With that team, we put the game into open alpha testing. It was a year and a half until we could get it to where people could play it, and then another year and a bit before we could charge money for it.

BS: *Did you have investors?*

DJ: A few. Three Rings was founded by myself and Michael Bayne, and most of the money came from us. We also took some money from friends, and then about a year ago, just about 9 months after launching the game, we got some funding from external investors. Not much in the overall scheme of things.

BS: *What have you learned in terms of attracting and keeping customers?*

DJ: We're pretty good at keeping them! Attracting them is more of an issue, which we learned after a year and a half of operating the game and trying to keep it appealing. You know, we didn't go through the traditional big launch route, where you sell a lot at the beginning, then it sort of trails off. We've been doing gradual growth.

In terms of what we've learned by gradually accumulating players, it's tough to do without substantial marketing or the big bump from something like a successful retail launch.

I think one of the biggest barriers for us is that downloading software at all is becoming a huge issue for a lot of potential players. A lot of players, particularly in the casual gaming space—the older, predominantly female market that groups like Pogo specialize in, and the younger audience like teenagers—they're just not into downloading software. Probably it's because their parents have told them that if they break the computer by installing any adware-ridden software, they'll maybe kill them. I think that's increasingly going to be a problem.

A PIRATE'S LIFE

BS: *In terms of the executable issue, what do you think about browser-based delivery?*

DJ: Web browsers are really interesting, and when we made the decision to make PUZZLE PIRATES a stand-alone game, we weren't aware of the executable barrier. I think it's gotten even worse with Windows XP and virus protection and so on. But also, we were convinced that we wouldn't get the right performance running inside a web browser. If I were to start from scratch now, I would probably address that issue head-on and try to build something within a browser.

If you look at RUNESCAPE, I think one of the reasons it has done so well is that it runs inside a web browser. In some ways, it's kind of contrary to common sense, in that you'd think that having something on the players' desktop would be a good thing. But actually, it turns out that people just don't want to install software. It depends on the audience as well, like for the Penny Arcade guys, or the core gaming sorts of people—those guys are fine. They'll install software with no problem. It's the kids and older casual players who are scared of it.

BS: *How is the internationalization of PUZZLE PIRATES going?*

DJ: It's interesting. We have a version out in Germany, and it's doing okay. It's not as successful as we'd hoped, but it has a small community and is trucking along.

We did have a Chinese partner, and they imploded. That was a small company that got the rights, and then basically just fell over and kicked its feet in the air.

We're looking now at China, Korea, and Japan.

China is really interesting, but it's very difficult to get any money out of China. There may be one and a half million people playing WORLD OF WARCRAFT, but I don't think Blizzard's actually getting paid very much for that. It's kind of the Wild West. Maybe if the right partner comes along; it's not the sort of thing we can do ourselves. The other thing is, even if we could, marketing and so on is very difficult in those countries, so we wouldn't really know where to start.

BS: *Do you want to keep support for existing games as you bring out new ones?*

DJ: Absolutely. The way I look at PUZZLE PIRATES, and pretty much any game we do, is that it's never going to go away. People are going to be playing PUZZLE PIRATES in 20 years. We may not be supporting it anymore, but we might just open-source it and give it to the community, but there will be people playing that game. And that's cool. I started another game 15 years ago, and people are still playing that.

BS: *What game is that?*

DJ: It's called AVALON. It's a text MUD.

BS: *Do you think it helped your initial street credibility to be a small independent developer, rather than part of a larger corporation?*

DJ: Sure, yeah. I think it depends on who cares.

There are two groups of people who care. There are game developers, who think, 'Oh, wouldn't it be nice to work in a small company?' The other group is the people actually involved in the game and the community. They care.

One of the luxuries we have as a small company is that we can be very present and available to the community. That's something we'd like to continue, even if we get quite a bit bigger. It's a huge help for how the players feel in the game, their loyalty, and so on.

BS: *Can you introduce BANG! HOWDY from a design standpoint?*

DJ: BANG! HOWDY is trying to do some different things from PUZZLE PIRATES. It's an experiment in doing 3D for us as a company, and an experiment in the 'pay for stuff, free to play' model. We're doing that with PUZZLE PIRATES, but we wanted to make a game out of the gate using that model.

It's an experiment in what they call casual games in Korea, which are games with short play cycles. Each round of the game is five to 10 minutes. You can drop in, play a round, then leave. It's much more of a pick up and play strategy game. It's also a much smaller project than PUZZLE PIRATES, which was kind of a kitchen sink game. In PUZZLE PIRATES, we had all these great ideas, and we put most of them in, then we got some more great ideas, and now we have a feature list as long as my arm.

With BANG!, we've got two guys working on it right now, hope to have three or four working on it by the end of the year, and hope to have it shipped pretty much around the beginning of next year—and in public testing before that. We're optimistic. It's a good model. If we can make games in six to 12 months instead of 18 to 24 months, then we can make more games, which we'd like to do. This sort of game is really big in Korea right now, and we're really curious to see if it'll work in the U.S., and if we can export it to Korea.

BS: *Do you gauge this as being potentially profitable in the U.S.? I guess it's a bit untried here.*

DJ: The gameplay is fairly untried, but the big thing is the item payment model. With item payment, it's kind of untried with the Western audience, and nobody's really doing it in the U.S. apart from us on the games side. Item payment is looking pretty good for PUZZLE PIRATES.

I don't think there's anything intrinsic about the U.S. audience that's going to turn this model down. With smaller products, like Habbo Hotel, which is more of a chat site than a game, it's doing crazy business purely in furniture sales, where you can buy things
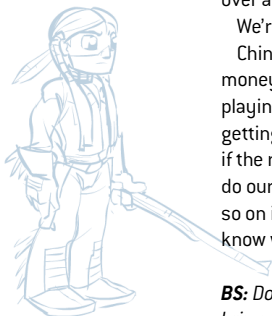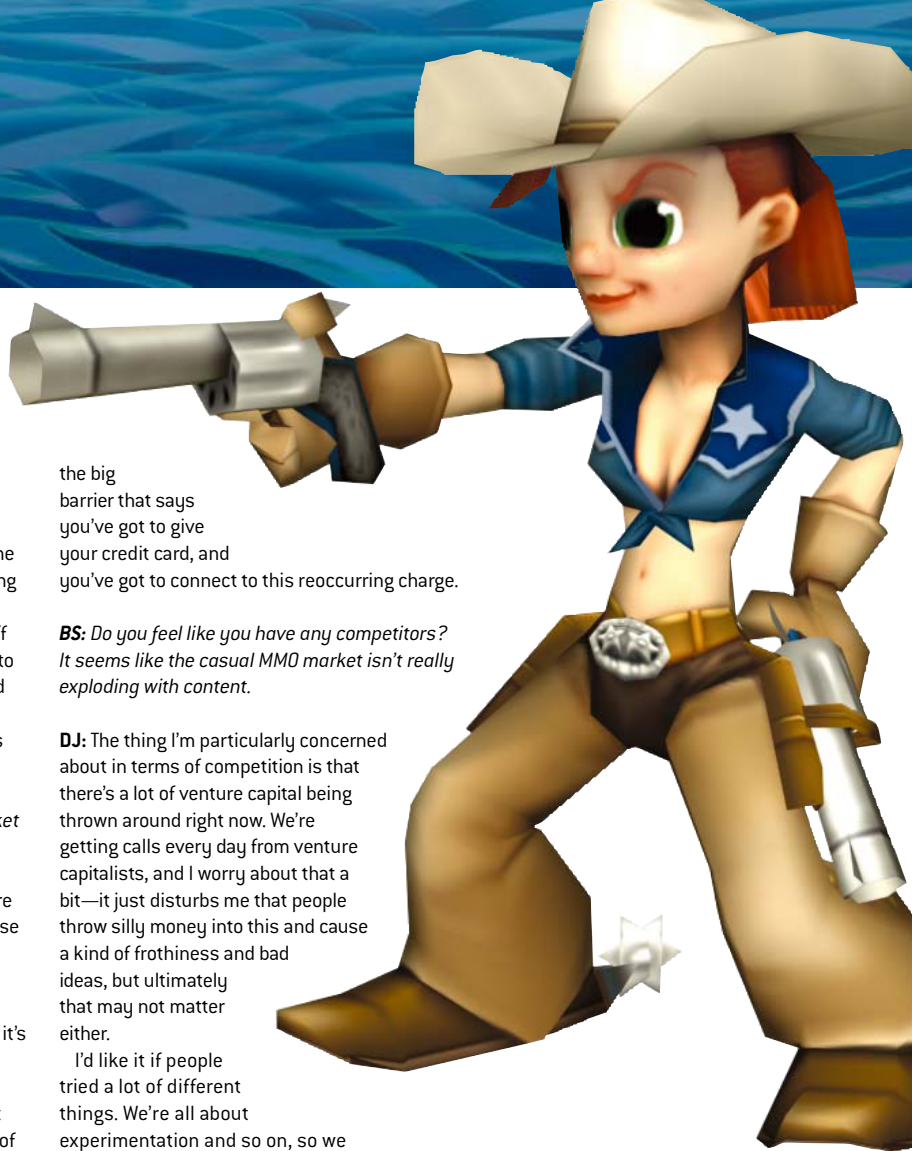
for your little hotel room. That's driving nearly a $20 million business for them this year. It's clearly something that people will do in Europe, and increasingly in the U.S. as well. I don't know that there are any big question marks there; the question is really how best to deliver it.

We have an exchange that's player-driven in [PUZZLE PIRATES]. You can exchange one currency for another [there are two currencies in the game]. Right now, you can log onto one of the servers and buy the fruits of people's labor, which they're being paid the equivalent of 25 to 50 cents an hour to create. That exchange is taking place, so people who just want to buy stuff are essentially facilitating free play for those who don't want to pay for anything. You can just roll up with your credit card and buy things, but you're basically subsidizing other peoples' experiences very directly. It's a fair trade. I think that model is peculiar but gentle.

BS: So how did you plan the game with the international market in mind?

DJ: UTF encoding, so it's easily localized. To some extent, we're working a bit in the dark because it's difficult to play all of these Korean games and really understand what's going on, since they're all in Korean. But we've done some of that and kind of worked it out.

A lot of the feedback we've gotten about PUZZLE PIRATES is that it's not anime enough for the Korean audience, even though the pirates are like big-head guys. So this time [with BANG! HOWDY] we're going a little more anime, but we're trying to bridge that gap between Western and international tastes. I think the proof will be in the pudding, ultimately.

We've got some interest from potential partners, and then when the game's up and running we'll shop it around and see if we can find someone who will take it out there. They'll probably want us to adapt it as well. Everyone who's wanted to take PUZZLE PIRATES into another territory has said, 'We want to change this, and we want to do that.' We're okay with all of that. Making the characters have bigger heads, that's okay.

BS: How does this model work with people who pay set fees for services like Pogo has?

DJ: It doesn't. It would be a value add. Like on Pogo they could have the basic PUZZLE PIRATES, with some of the pay-for features, but free to the user as part of the subscription. But then there would be additional billing on top of that, which is not necessarily something Pogo wants to do.

You've got to look at what Pogo's doing, and think, 'They're getting $20 to $30 out of a customer per year,' and they're probably looking at that and thinking, 'Maybe we could make more money.' I know they can make more money than that.

The whole subscription thing is going to go through some mutations. It's very appealing business, but there are some big barriers there. There's the big barrier that says you've got to give your credit card, and you've got to connect to this reoccurring charge.

BS: Do you feel like you have any competitors? It seems like the casual MMO market isn't really exploding with content.

DJ: The thing I'm particularly concerned about in terms of competition is that there's a lot of venture capital being thrown around right now. We're getting calls every day from venture capitalists, and I worry about that a bit—it just disturbs me that people throw silly money into this and cause a kind of frothiness and bad ideas, but ultimately that may not matter either.

I'd like it if people tried a lot of different things. We're all about experimentation and so on, so we don't claim to own the market at all. It's just that no one yet has managed to bring something exactly the same. That said of course, I think a lot of people have looked at PUZZLE PIRATES and said, 'Well they did okay, but they didn't blow it out of the park, so maybe we won't do that. We'll make a WORLD OF WARCRAFT clone, because those guys blew it out of the park.'

And you know, good luck with that one, and have fun raising the $30 million you need.

BS: In the casual market, do you think the console space is at all viable? Microsoft seems to be doing some pretty interesting things with Xbox Live Arcade.

DJ: I think Arcade's cool, and Microsoft is doing a great job of leading the market there, and I think Sony and Nintendo will have to catch up big time.

I'm a little bit nervous about the console space. We were kind of excited about the PSP and the Nintendo DS, but even with those platforms you're seeing budgets get really big, you're seeing production values being kind of a primary part of the selling message. All of this just really bugs me, and basically for us to do a console game, we'd have to have a publisher, and that brings us into the whole developer/ publisher relationship that's the bugbear of any independent third party developer.

I'm also very reluctant to staff up my company with a bunch of people, because if we get a publishing deal and the publisher says, 'You've got to spend all this money, and add
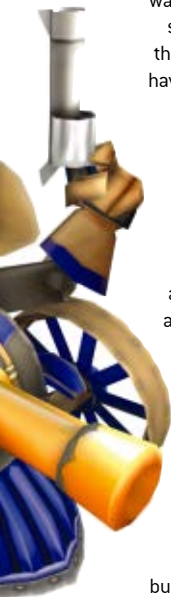
15 people,' and we hire 15 people, and the publisher says, 'We decided we don't like this idea anymore,' and then they can us, and now I've got 15 people on my payroll.

In terms of how to fund expansion, I much prefer the venture capitalists because at least they give you the money. They don't have milestones. No—they give me the money, and I can expect to have that money in my bank account. That's certainly more reasonable.

The other thing about console is that you're in retail. PUZZLE PIRATES was interesting because it was direct download, but for the most part, console is retail, and then you're in that nightmare. If you've got no long tail to it, if you don't go big out of the door, you're never going to get anything. It only trails off from there, you can't do a slow build. You have no direct connection to your customers. It's just unappealing as compared to the interweb.

The interweb's great! You go out onto the interweb and there are just so many options in getting to people, and so many people out there, and that's got so much freedom to it, and very low cost to build. There's all sorts of things with console like how you have to finish the damn thing before getting any kind of player feedback on the platform. But with the interweb, like with BANG!, we're going to put it up in the first stage with very little content and get a very good idea of how fun it is to play.

To my mind, as a small company, and probably the rest of the industry is thinking this way too, how can you reduce risk? How can you avoid spending $15 million to find out something's not going to work?

BS: *Do you think it's possible for independent developers to survive in the current market?*

DJ: Sure. It's easier than it has been for a while, if you pick the right thing. I split independent developers into two categories. There's independent, not funded by publishers, and then there's independent, making their dollars by doing work for publishers. I think that for many of the reasons I just talked about with respect to consoles, business is extremely tough for the third party developer with a publisher, and I think the guys in this area will tell you the same. I just read Double Fine's postmortem [of PSYCHONAUTS], and I can't imagine the stress those guys must have been under when their product had just gotten canned, and they've got everyone on payroll, and what do you do?

I guess the only other thing is mobile, which is also becoming really really tough. You can't go direct to carriers anymore, and there's no way around the carriers. At least with the internet, you can put up a web site, and some people are going to find your game. And if the game's really good, then they're going to tell other people, and you'll get some momentum. With mobile there's no way to do that, and then you're dealing with a publisher, and then you're back into developer/publisher relationships, just with smaller budgets. I will say that on the developer/publisher side there are a number of publishers in the casual games space, like there are for mobile that have smaller budgets. I can think of three companies that are funding titles. One of them is PopCap, and those guys are really cool, and they've open sourced their code, so you can just download it and make your game, and either sell it yourself, or you can present it to them, and maybe they'll pick it up and distribute it. That's the way I would do it if I were starting from scratch. And try not to do the kitchen sink thing. If I were to go back, I'd make PUZZLE PIRATES simpler and more achievable, rather than a more ambitious thing.

BS: *So more of an established design?*

DJ: Well, there's kind of a debate about that. The classic model is to pick one thing, and make one part of it better. Lots of games did this like EVERQUEST, or WORLD OF WARCRAFT. But I think getting it out there with something is the key. The big thing for independents is getting moving, and not waiting around. I used to get a lot of people mailing me saying, 'We want to make the next great MMO.' And I told them, 'Don't do that.' They wanted elves and goblins and stuff like that. Do something different.

BS: *Have you looked into the serious games space at all?*

DJ: I think the serious games space is interesting. It's certainly a way to get started, but it's basically work for hire. And I don't see a tremendous up side there for the developer. The nice thing about making a game that goes out to consumers is that if it catches fire, you could make a ton of money. Like GameLab did DINER DASH, and now they're in a way stronger position, they're getting royalty checks, they can go out and do more publishing deals, and they've got a lot of options, because they shipped a hit game. But if you do a serious game, you're basically doing client work that gets done, and then it's done.

BS: *What do you think is missing from online gaming now?*

DJ: I think probably the biggest single thing that really gets on my nerves now is the distribution platform executable problem. Windows executables, virus scanners, and XP 2 cause problems. People don't want to download software. Java: in a bit of a state and not very widely installed. Flash: performance issues. Shockwave Director: not very widely installed, neglected, out of date platform. How can I just make games that people can play?

Oh please! I know nobody's just magically going to solve it, but that's one thing I think could be improved. And I think that anything that makes it easier for people to develop is good. ▪
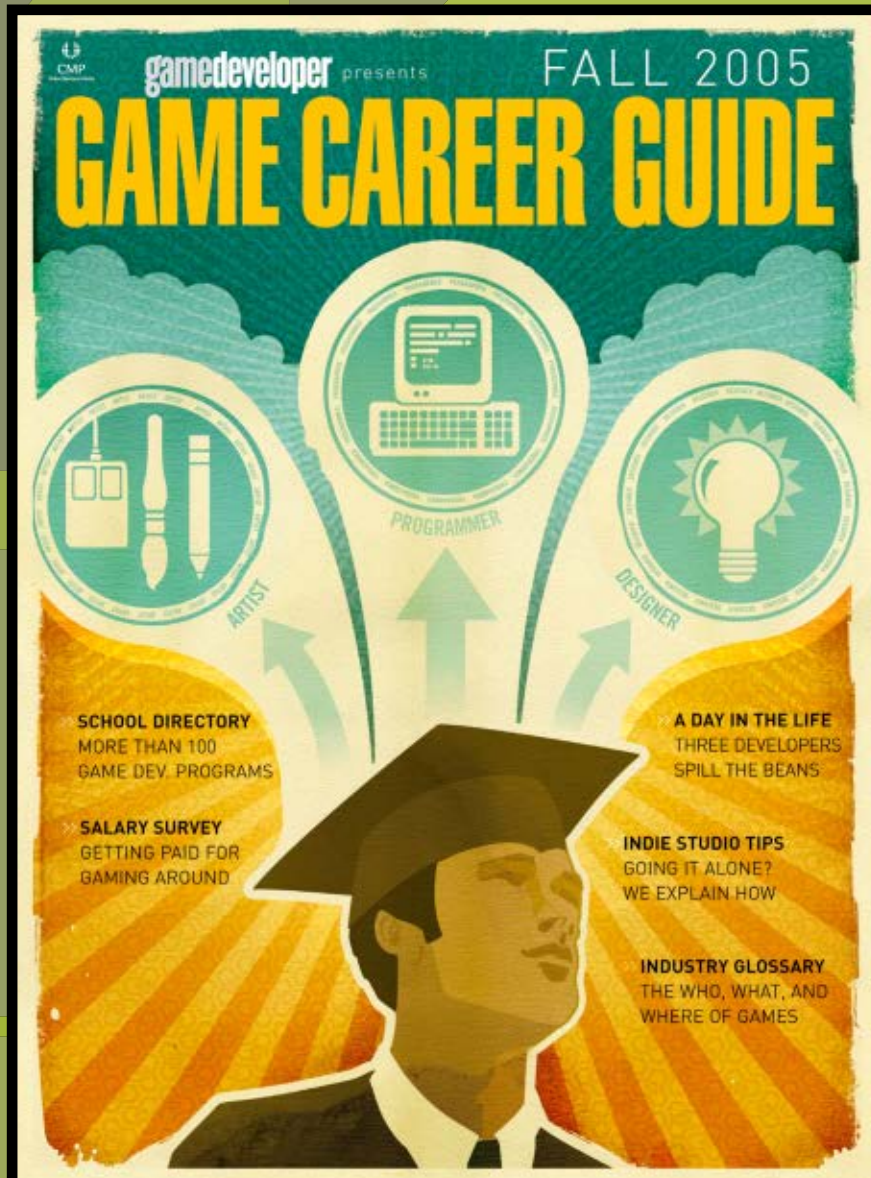
POSTMORTEM

THE GRAPHICAL STYLING OF resident evil®

YOSHIAKI HIRABAYASHI *joined Capcom in 2001, working on the GameCube port of the original RESIDENT EVIL. He is principally a graphics designer, specializing in real-time cutscenes, which he worked on for both GameCube RESIDENT EVIL ports. This experience earned him the title of lead real-time cinematics director on RESIDENT EVIL 4. Email him at yhirabayashi@gdmag.com.*

THE RESIDENT EVIL SERIES HAS A broad fan base, and in order to meet players' expectations, we decided to create a totally new entry with RESIDENT EVIL 4. Because the series has been around for so long, we really wanted to address the feedback from both our fans and our development team in order to revamp the game. This meant looking at everything from presenting a new way to experience fear, creating more frightening enemies, implementing new ways of using items, and much more.

For this postmortem, we'll use one element, which was also one of our biggest challenges, as the archetype for the game's development: the successful creation of the title's graphical style. I'll provide an overview of what this entailed and how we were able to achieve it with some specific examples from the game.

WHAT WENT RIGHT

1 **CUTSCENE INTEGRATION.** When we began the project, one area we focused on was how playable portions of games usually shift into atmospheric pre-rendered movies. This seemed like an area that, if done well, would improve critical reception. As gameplay shifts to a cutscene, the change is usually quite noticeable. It's possible that in that moment, players regard what is on screen as just imagery rather than a true part of the game. The change might be appealing to those people who simply enjoy cinematics for the higher quality of the cutscene graphics, but in terms of keeping players focused on the game, it's possible that these moments interrupt the flow of the experience. We thought that if we could facilitate a seamless transition between gameplay and the in-game movies, people would be able to stay involved throughout the entire experience without interruption. Our solution was to keep the cutscenes in real time.

The action button system we implemented for RESIDENT EVIL 4 was very complementary to our use of real-time movies. By incorporating an action button into the cutscenes, we made it possible for players to interact with the in-game movies. In a traditional game scenario, players change from being active participants to bystanders as the cinematics begin and play out. The player might not pay close attention or might even put the controller down, and either way, that's not what we want.

2 **IMPROVED TECHNOLOGY.** In the current generation of consoles, the technical capacity of hardware has improved vastly over the last,
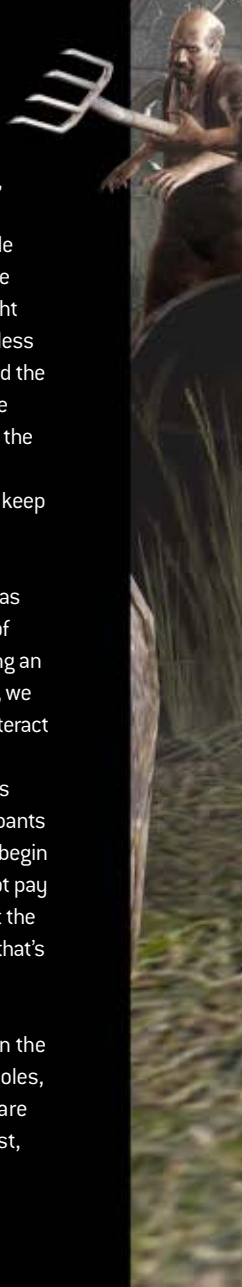
GAME DATA

TITLE
Resident Evil 4

PUBLISHER/DEVELOPER
Capcom

PLATFORMS
GameCube, PlayStation 2

RELEASE DATE
Jan. 11, 2005 (GameCube),
Oct. 25, 2005 (PlayStation 2)

NUMBER OF DEVLOPERS
80, including part-time

DEVELOPMENT TIME
3 years, including concept
planning

**SOFTWARE USED
(FOR GRAPHICS)**
Softimage XSI and Alias Maya

HARDWARE USED
NEC Workstations

and our technology itself has also increased to the extent that we can maximize the full potential of that hardware. Technologically speaking, this advancement has made it possible to express scenes in real time that would have previously only been possible in pre-rendered cutscenes, for example those that incorporate complex facial animation. Up until now, we didn't have the processing ability or capacity to realize complex animation of the sort we have achieved in RESIDENT EVIL 4—it was simply outside the hardware's capabilities. We solved this issue through programming and by packaging data intelligently. The same solution was applied to areas that required a lot of special effects, such as projection lights and explosions.

Using real-time movies also made it easier for us to change elements of the story according to the game specs and design. For example, in a pre-rendered situation, if a character or enemy in a movie had to be somehow altered, all the time and energy used to create it would have gone to waste. However, by using real-time movies, we could just rewrite a new model onto the existing model data.

3 **IMPROVED WORKFLOW**. Since we had to basically reinvent the series, we needed to improve our workflow—converting data, managing data, and troubleshooting—in order to stay on target. We needed to find a way to complete these tasks more efficiently, which was how we came to use the XML data management system. This workflow management method saved us valuable time that we were then able to apply to the creative elements of the game design.

Figure 1 shows the system we used for converting movie scenes, posting scene data, modeling, and making slight adjustments for texture data, among other things. The XML system allowed us to evaluate the total amount of data in order to assess whether there was an excess or deficiency to support it. The system also eased the process of making multiple alterations to the customized models in each of the game's many scenes.

Looking at our previous system (Figure 2, page 30), you can see exactly why our new system has helped us. The traditional development pipeline in Japan is still quite hierarchical: With the

previous system, you needed a programmer to transfer data to the development tool, which put limitations on the programmer's work capacity. This situation caused a tremendous loss of time.

As opposed to the old flow, which put the heft of the workload on the programmers, the new system (Figure 3, page 30) actually resulted in the designers being able to contribute more frequently and more directly, thus minimizing the amount of time lost.

4 **BELIEVABLE IMAGES, APPEALING CHARACTERS.** Creating believable images doesn't necessarily mean that the images need to be factual and realistic. Rather, they should engross players and be believable within the game's universe. For example, in RESIDENT EVIL 4, a long skirt or long hair moves naturally according to the motion of the characters, but in a slightly enhanced way. This effect helps to immerse players and create a suspension of disbelief. If something swings in the real world, it also swings in the game; however, in real life, it might not move in such an exaggeratedly beautiful way.

We adapted the swings a little and brought the motion close to what players might expect, even though it isn't completely accurate.

In regard to creating more appealing characters, there's a wide range of opinions on the topic. In our meetings with Shinji Mikami, the game's director, he stressed that the most important factor in making a character appealing is to create believable facial expressions and gestures. Characters should exhibit appropriate feelings and expressions for a given situation, while also expressing their individual personalities.

With Mikami's goal in mind, we put a great deal of effort into making characters' expressions believable (see Figures 4 and 5), and to that end, each characters' fingers have joints that move and articulate, for example. Still, we wanted to focus on even more refined details. In order to achieve this, we used a very large number of face targets (or facial expressions) for each character.

At key moments, we conveyed the tension that characters might be feeling by using extra lighting elements to highlight a character's facial expression. We actually created quite a lot of facial expression patterns for the characters. We also used special higher-quality textures to make the expressions look as good as possible.

For Ashley, the main heroine, 36 facial targets were used in total, which is one and a half times the normal number. Regardless of the effectiveness of the increased targets, we couldn't have that much data for just one character, so we used a method called face target packaging. Even though 36 targets were prepared, they were not all necessary for each scene. We included targets required by each scene into models that implemented them on a scene-by-scene basis.

At first, I talked with a programmer regarding the amount of data which could be used for a character and decided to use 30 targets for each. We divided these into two categories (see Figure 6, page 32). One is the Standard Slot, which is frequently used and is the general default slot. The other is the Select Slot, which dictates less frequently used targets for each scene. We made one package with these two slots. Because the number of Select Slot expressions varied from scene to scene, we were able to efficiently manage the data allocation so
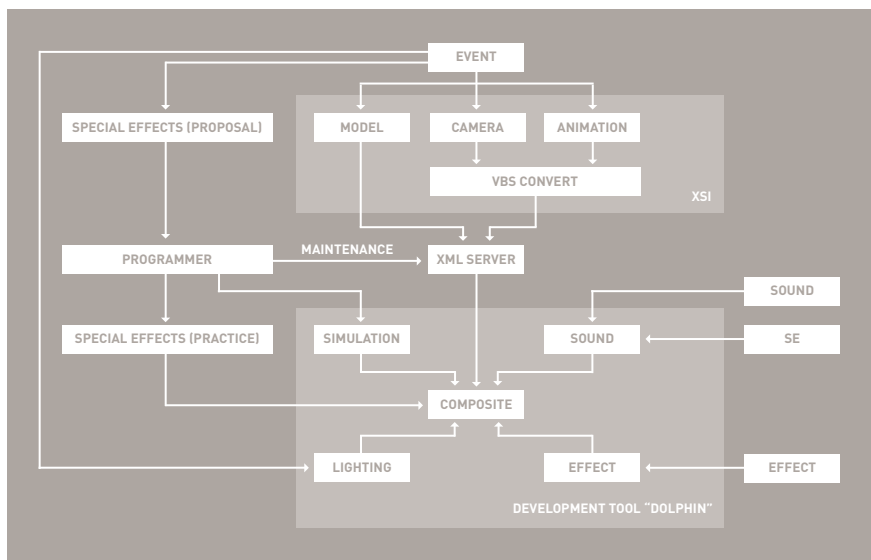
FIGURE 1 **Capcom's system for cutscene implementation.**

TDs: Take Five.

SOFTIMAGE®|XSI® v.5.0

· Work with ten times the detail   · Non-destructive everything
· Easy migration from *Maya*   · Native 64-bit

Introducing **SOFTIMAGE|XSI v.5.0**, the digital character and effects software with features that are carefully designed to work for you, and not against you.

Whether you're a technical director, an artist or a developer, with **XSI** software you're free to do what you're best at. **It's really that easy.**

Order XSI v.5.0 now at **softimage.com/five** starting at **$495**. Promotional upgrade pricing available

Avid
computer graphics

FIGURE 2 Capcom's older, hierarchical workflow.

FIGURE 3 The new, improved workflow.

FIGURES 4 and 5 Examples of RESIDENT EVIL 4's detailed expressions.

there were only 30 targets in the package at any given time. By changing the package according to the scene, we were able to produce facial animations with higher quality.

5 **REINVENTED GAMEPLAY.** Even though the series is linked to horror, it also has become known for having strong gameplay and a high entertainment value. We wanted to make RESIDENT EVIL 4 appealing to an even wider variety of players by raising the game's level of entertainment. Early in development, through trial and error, we found that the game was scary, but had a low fun factor—we needed to rethink both the gameplay system and the fear component. We restarted the project four times in order to ensure that this title would be interesting and fun for the consumer; there had to be more to the game than just fear.

In order to change the gameplay, we challenged ourselves to implement features and elements that had been impossible to realize previously, in terms of environmental interaction, graphical quality, and more frenetic action. We took the existing archetype of RESIDENT EVIL and added extra play value that would make sense within that world. The main character's melee attacks were made to be more robust because he's a police officer and should be better at close-quarters combat than we had demonstrated in previous games. Enemies were made to be more intelligent and much faster; they can pick up weapons and open doors. Enemies also talk to each other, communicating plans of attack. This was used as a new way to inspire fear for the series, giving the player a feeling that he or she was up against a sentient and unrelenting force, rather than shuffling zombies, or simple monsters.

We had intended from the start to make the project something that we ourselves found intriguing and challenging. In demanding this of ourselves, and largely meeting those goals, I think we were able to provide an exciting game for the fans.

WHAT WENT WRONG

1 **FRESHEN UP.** The goal of any game's storyline is to expand the depth and perception of that game's universe, as well as to inform players of what they are to do. In RESIDENT EVIL 4, the storyline was primarily conveyed through our real-time movies. We had a hard time achieving our concept of combining gameplay with the necessary story elements that could evoke a feeling of believability within the in-game movies.

Games are a visual medium, and throughout the series, it wasn't always easy to create fresh concepts. It was even more difficult to develop visuals that would continue to advance the series. This is why we thought of a new direction for RESIDENT EVIL

4's development, which we call "complete concept changes." That is to say, we examined all the previous gameplay features and notions of fear and broke them down to change the entire concept for the game.

2 **DATA CONSTRAINTS.** Because this version of RESIDENT EVIL was far more action-based, we often needed to display a very large number of characters at once (see Cutscene Images A–D, page 32). As a result, we needed to carefully apply model textures for these types of scenes. Normally, an enemy character in the game has about 3,000 polygons and is about 400K of data space. There are some scenes in the game, though, that had up to 25 characters on screen simultaneously. We weren't able to adjust all the memory and image storage to the level that would be required for us to keep those 3,000 polygon models intact.

In order to address this issue, we decided to adjust the amount of data per model in each scene, and even in cuts within the scene. First, we would implement the model that required the most data (usually the closest to the camera). Then we adapted it by deciding how it would be drawn and processed, based on memory levels. Limiting the amount of data on all the models would result in poor quality, so we prioritized the portions of the models that would require the best modeling. We came up with three different scenarios for how to distribute the data for texturing the models, which could then be used in various situations by combining them.

A normal high resolution model for a Ganado, a generic enemy in the game, consisted of approximately 3,500 polygons. Normally, it would be difficult for us to render several high resolution Ganados on screen simultaneously. Therefore, we created two additional variations of the model: mid-resolution Ganado, which are approximately 2,000 polygons, and a low-resolution model with about 1,000 polygons. These three models would switch depending on the situation. Textures were treated in the same way. High-resolution textures were 512x512 pixels, mid-resolution textures were 256x256 pixels, and low-resolution textures were 128x128 pixels.

As mentioned, these models could be switched depending on the situation, according to two conditions: how many models the scene required, and how much data was available for the given scene. We always attempted to be efficient with our data sizes, so we would reduce the model and texture data by taking into account the camera and lighting used for certain scenes.

3 **TEXTURE VS. LIGHT.** In addition to the models, we also adjusted textures for each scene. We sometimes faced problems with how to adjust the way that characters looked when different lights hit them. This happened especially when we used a lot of penetrating or spot lights. We solved this problem by creating textures for each scene or each cut.

Not only was this used for something as detailed as eyes (Figure 7, page 32), but was used widely, anywhere from the characters' bodies, to background objects. This texture adjustment was also useful in addressing problems with
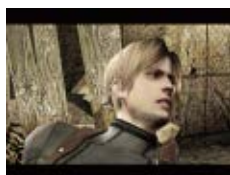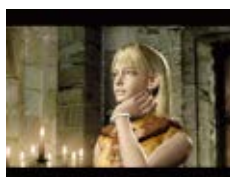
GameDevelopers
Conference

MARCH 20-24
SAN JOSE, CALIFORNIA

WHAT'S NEXT
GDC:06

www.gdconf.com

GDC MOBILE   SERIOUS GAMES SUMMIT

GAME DEVELOPERS CHOICE AWARDS   INDEPENDENT GAMES FESTIVAL   GAME CONNECTION

WHAT'S NEXT IS EXCLUSIVELY AT GDC 2006

From console and PC gaming to the casual and mobile sectors, the interactive entertainment industry is poised more than ever for rapid growth and change. At GDC 2006, you will gain access to the people, technologies, and tools that will define *what's next* for our industry. LEARN MORE AT WWW.GDCONF.COM.

CMP
United Business Media

dynamic lighting situations, such as when dented objects were being lit too brightly. In cases when a model would only be seen from a distance and required little to no shading information, we reduced the data amount because the dimensions visually do not stand out very well, and the difference would not be noticeable. In another case, when a character's shadow would stand out due to the camera's position, we wouldn't use the low-resolution model, as it would not be capable of properly rendering the delicate lighting details. Instead we would use a high-resolution model with low-resolution textures, still keeping data use down.

These scenes use high-resolution character textures and models, but lower resolution background textures in the far depth of field.

4 **DEPTH OF FIELD.** Another problem we had to address was related to depth of field within a given scene. One method we used is a blurring effect: We displaced several pieces of a given image at various depths of field when no camera movement was happening. In this case, processing power was light, and the hardware was not overly taxed.

The more difficult scenarios came up when manipulating depth of field in a scene that required camera movement. In these cases, we specified a distance from the camera past which we would blur the image. We dynamically blurred the image itself, just as you would in Photoshop.

The method was artificial, but it enabled us to convey a depth of field that appears in the actual in-game camera. The processing power was very heavy in comparison with the first case, and therefore, we used the blur only when we needed to move the camera and adjust the depth of field from a creative standpoint.

5 **MULTI-CAMERA STRAIN.** Some situations required real-time rendered textures, which in turn required the use of multiple cameras. By using multiple cameras, the processing strain is increased by about one and a half times the amount used with just one camera. Despite this, we decided to use this method because the workflow system we utilized made it very

CUTSCENE IMAGES A–D
Displaying multiple characters onscreen simultaneously required inventive texture and model assignments.

FACE SLOTS TOTAL = 30 SLOTS

STANDARD SLOTS = 25 SLOTS

SELECT SLOTS = 5 SLOTS
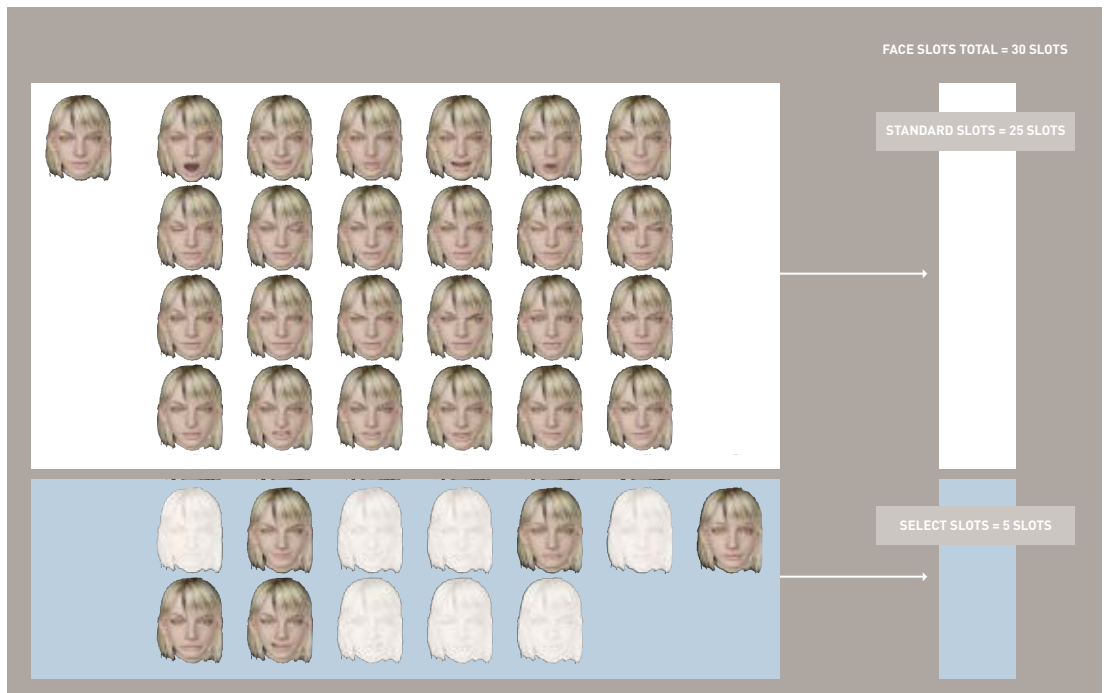
FIGURE 6 RESIDENT EVIL 4's package slot system.
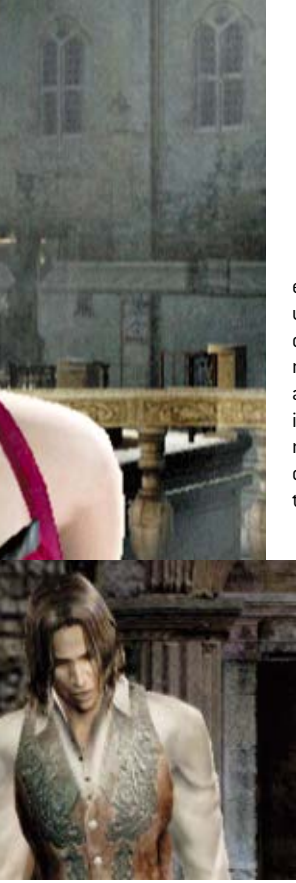
efficient to adjust the images. This technology was used in a scene in which Leon looks at a monitor on the screen. For the "video" displayed on the monitor, we didn't use pre-rendered movies, but animated scenes which were rendered to a texture in real time. With this method, we did not have to render character motion on the monitor in order to change elements of the scene, and so we were able to make changes quite easily

Normally only one camera's data is used for one cutscene, but data from three cameras had to be used here (see Figures 8 A–D). One was the main camera, which projected the entire scene. The second was camera A, which created the image shown on monitor A. The third was camera B, which created the image displayed on monitor B. The screen images which were created by camera A and camera B were cached by memory, scaled down to a lower resolution, and used as the texture for the monitor. After that was done, we compiled the image for the main camera view. This technology was also used for reflections in things like sunglasses and car windows, with translucent adjustments to cached image data.

THAT'S A WRAP

As you might have gathered, the development of RESIDENT EVIL 4 was not necessarily based on innovative new technology, but rather on efficiency. Our improved workflow and our intense focus on details in the game allowed us to achieve the level of quality we had challenged ourselves to produce.

Restarting the game multiple times allowed us to take several fresh perspectives on the game, and the survival horror genre as a whole. I think that ultimately we came up with something that was not only enjoyable, but which also helped to advance the series in a positive direction. The next RESIDENT EVIL is planned for next generation consoles, and will present a whole new host of challenges and opportunities. Hopefully we will once again be able to meet our own high expectations. ✣
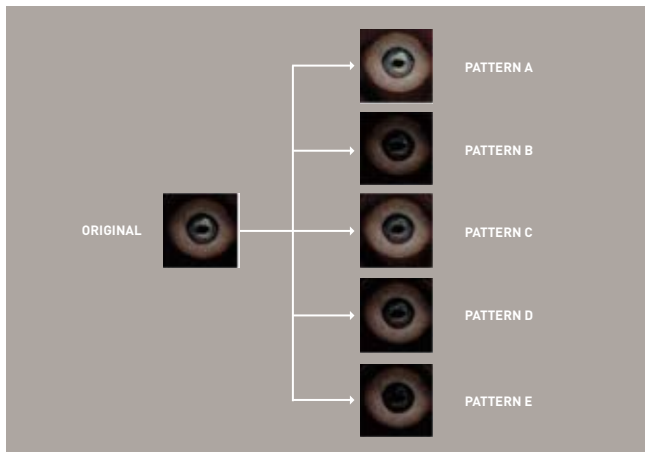
FIGURE 7 Textures change depending on the scenario, as you can see from these five variations of how an eyeball might look.
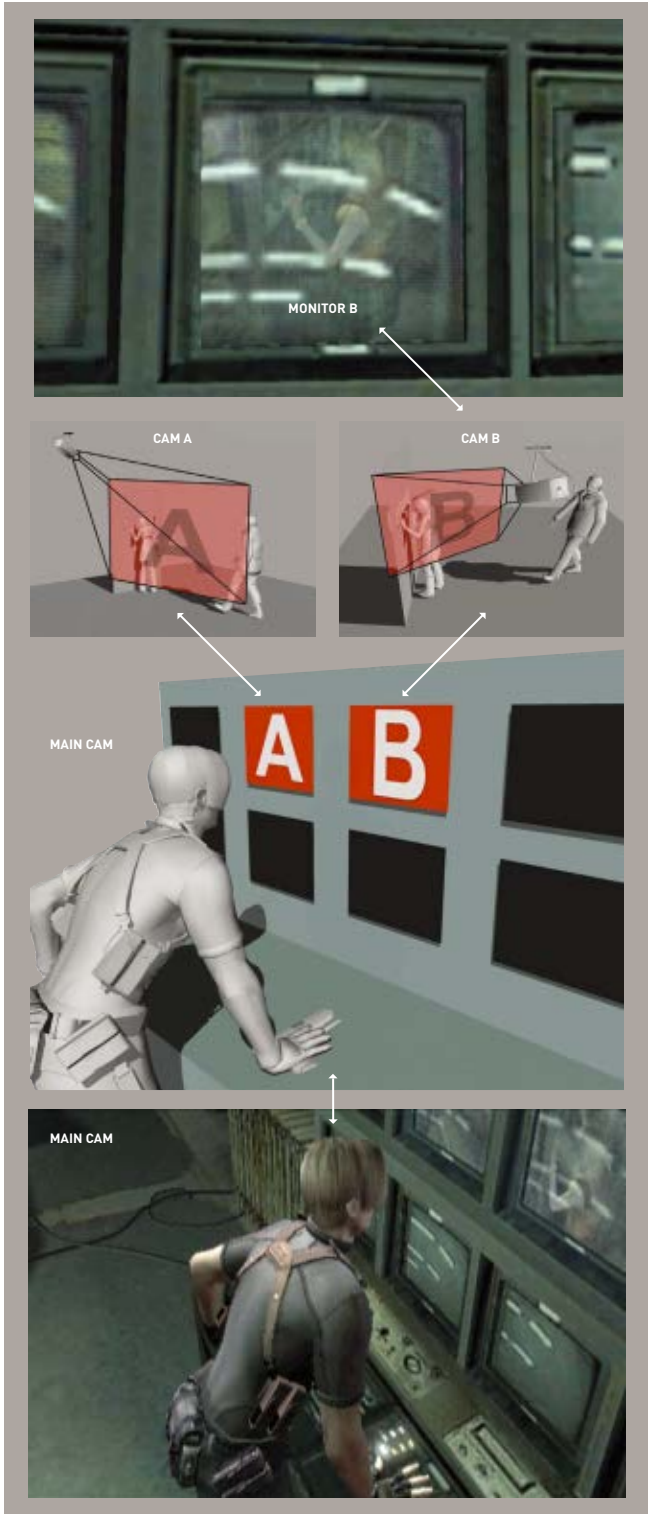
FIGURES 8 A–D Multiple cameras were used to output video-like textures.

MICK WEST

>> THE INNER PRODUCT

DEBUG AND RELEASE

GAME PROJECTS USUALLY HAVE A NUMBER OF BUILD configurations, which control how the code is compiled and linked into the game executable. Each build configuration or build mode builds the game with varying amounts of additional debugging code, and with the optimization options modified to aid debugging. Microsoft's Visual Studio has by default just two: debug and release. The idea is that you will develop your game in debug mode, making it easier to find bugs, and then ship it in release mode, with debugging code removed and optimizations switched on to make your code small and fast.

This division usually starts out fine. At the start of a game development cycle, there are not too many assets in the game. Level maps are small, not much of the logic is implemented, and the CPU is rarely taxed. But after a few months of development, game assets, entities, and logic start to get into the game. As the quantity of these things approaches the level of the final

game, debug mode becomes unusable because it's so slow. At this point, the development team usually switches to release mode for day-to-day development work.

But because the traditional release mode lacks debugging code and assertions, it becomes a lot harder to track down bugs. When programmers attempt to reproduce the bugs in debug mode, they first have to rebuild in debug mode (which can take a very long time), and then the code runs at a really slow frame rate, making it very difficult to reproduce the bug. Sometimes the bugs can't be reproduced at all, due to the different configuration and initialization of memory.

The way I see it, the traditional division of debug and release is inappropriate for game development. Both modes, as traditionally envisaged, should no longer be considered useful options and should be replaced by a single build configuration—"develop mode"—that should be used at all stages of development.

DEBUG AND RELEASE

When you set up a project in Microsoft Visual Studio, you get two build configurations: debug and release. The distinction between these seems obvious. You use debug to debug the project, and

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**.

when it's ready to ship, you switch over to release.

This distinction is usually found in any game development project, and sometimes it's extended to some additional configurations, like "extra debug," "final," or "submit."

The intent of debug mode is to make the code easier to debug. But we've become so used to having the distinction between debug and release over the years that it's high time we stepped back and asked ourselves exactly why we needed a debug mode in the first place.

A common misconception that I'd like to clear up right away is that you can't use the debugger unless you build your code in debug mode. This is simply not true. The debugger will work just fine in release mode, but some aspects will be a bit harder to debug.

The idea behind debugging mode is to make it as easy as possible to track down bugs in your code. This results in two major differences. First, the code is compiled in such a way that the debugger works as well as possible. Second, extra code is compiled in, usually in the form of assertions, to trap bugs as early as possible, and to provide additional debugging information.

Optimization. In debug mode, optimization is disabled, so the code runs a lot slower. Why?

Optimization involves re-ordering the sequences of assembly instructions so there's not always a direct correlation between a group of assembly instructions and a line of C++ code. When you step through your code in the debugger, it's not always clear exactly where you are in the code, and the PC might jump oddly between lines.

Also, optimizing involves keeping values in registers instead of memory as frequently as possible, so local variables that are not used very much are not stored in the local stack frame. In other words, in the debugger, you often can't examine the contents of a local variable in optimized code, as once you step past the code that uses it, the value no longer exists. In debug mode, each local variable always has local storage for itself, so you can always examine its value in the debugger.

Inline expansion. Another form of optimization is the expansion of inline functions. This is done to clarify the flow of control in the debugger. If an inline function is expanded inline, then you can't step "into" or "over" it. It almost becomes invisible to the debugger.

Debug code. Besides the optimizations, you usually add additional "debugging code" to your program to help track down bugs. To this end, you generally have some symbols defined that tell you at compile time which build configuration you are using so you can conditionally compile code for debugging.

In Microsoft Visual Studio, these symbols are DEBUG, which is defined when you're in debug mode, and NDEBUG, which is defined when you're not in debug mode (or in release mode).

Assertions. Everyone should be familiar with assertions. Assertions are additional lines of code sprinkled through the program that ensure that everything is as it should be. Basically, they are macros that check (assert) that a condition you know should be true actually is true. If the assert fails, then the program execution halts and you can see what went wrong. Asserts should be your most useful tool in debugging your code.

Listing 1 shows Microsoft's implementation of the assert macro. The assert is compiled away to nothing when you build in release mode. This makes sense, as when you release the game, you don't want asserts going off. Plus, they take up space, and all those extra checks slow down the game.

LISTING 1

```
#ifdef NDEBUG
#define assert(exp) ((void)0)
#else
#define assert(exp) (void)( (exp) || (_assert(#exp, __FILE__,__LINE__), 0) )
#endif /* NDEBUG */
```

Debug and browse information. For the debugger to work, it needs to know how to link the code it is executing to the original source. This information is generated during compilation and linking, and is referred to as debug info. In many release configurations, the debug information is switched off, which is generally done to make the project build faster and the executable smaller. An executable .ELF file built with GNU C++ can be anything up to 100MB in size when built with full debug info, compared with a 3 to 5MB file without debug info.

Memory allocation and initialization. Memory allocation is the source of many bugs, so it makes sense to have extra debugging code in your memory allocator. Typically, memory will be initialized to a known value when it is allocated. The pattern of allocations will also differ, as blocks have extra debug info added. Since most games use their own custom allocators, the differences vary. But nearly all games will have some difference in memory allocation between debug and release modes.

Un-initialized variables are a common source of error. Generally, you should be catching such problems at compile time, but if not, you need to realize that their behavior will differ in release and debug modes. In debug modes, local variables are more likely to have actual storage, and depending on your precise build configuration and compiler, they will probably be initialized to zero in debug mode. In release mode, however, the un-initialized variable will be some random value—whatever happened to be in memory or the register used.

Games are real-time applications. They have to run at a reasonable speed in order for you to say that they are working. So if the debug mode is very slow, then it's not going to be practical to use the debug mode as the development build—the build used by artists and level designers implementing assets, and by programmers implementing new features in code or script.

Nevertheless, games are incredibly complex applications. The game engine is often still in development while the game content is being created and implemented. If everyone is using the release mode for development, then it makes it very difficult to track down the bugs that inevitably arise in a development environment.

Clearly, neither of the traditional build modes is suitable for game development. I'd like to make the case that a single hybrid mode should be used for 99 percent of all development. This hybrid mode (develop mode) should be fast enough so that gameplay is not hampered, but also contain enough debugging features so that bugs are caught early and are easily tracked down. Develop mode should also essentially be the configuration mode your game ships with.

DEVELOP MODE

Assertions switched on. Developing without assertions is like driving with your eyes closed. You'll know when you crashed,

but you won't know why you crashed, and your crashes will be much worse. Having assertions on all the time during development will greatly improve the rate at which you find and fix bugs.

Optimization switched on. Your code needs to be fast if develop mode is going to be used by artists and level designers. Sure, you can't tell exactly what's going on in the debugger because you're not able to see the contents of local variables. But you will be able to identify the place in the code where the crash occurs, see the call stack, and roughly follow the logic flow. If you need more information, then often the solution would be to add more assertions. You can add logging calls to track the contents of variables, and if all else fails, you can temporarily switch off optimization.

Inline expansion switched on. The inline expansion is similar to optimization, but with games, the inline expansion's being off is often a far greater source of slow debug code than other aspects of optimization. Most games will have some kind of custom 3D vector class, usually with accessor functions, or overloaded [ ] operators that use inline functions. The fact that these functions are explicit adds a vast overhead to code execution. The benefit is simpler flow of execution in the debugger, something you rarely need. So in develop mode, you would switch inlining on.

Link without debug information. Linking without debug information can be a vast time saver. The link stage of the edit-compile-link-run cycle can take up to several minutes, depending on the type of project. The vast majority of that time is spent generating debug information, when really all you want is the executable. Remember the compilation units are still being complied with debug information, so if the code crashes and you want to go into the debugger, then you can just switch debug information back on and re-link, and you'll have the debug information, but only when you actually need it.

Make your assertions fast. Assertions should never need more than five percent of your total CPU time. If you turn assertions on and your framerate plummets, then there may be a problem in the implementation of your assertion macro (many game developers implement their own version of assert). The majority of assertions are simply comparisons of two values, and usually one of these values is something the compiler will have in a register; the other is usually a constant. Your assertion should compile to two or three instructions that perform the test and then skip over the code that arranges the parameters and calls the assert handler. You should verify that it's working by looking at the compiled code in the debugger.

Use assertions appropriately. As much as I love assertions, there is such a thing as having too many. In very low level functions, assertions are often testing the same thing over and over again. For example, collision detection code might use unit normals stored in the mesh. Since collision code runs a lot each frame, then adding an assertion to verify that the normals were of unit length might make a serious blow on your framerate. Plus, you'd be repeatedly checking the same normals over and over. In this case it might be better to verify the input just once when the mesh is initially loaded, or even when it's originally generated.

In addition, putting assertions in at such a low level of the code is often a hit or miss affair. The conditions that might cause the code to fail could occur very infrequently. You might have to play the game for hours in order to hit the triangle with a malformed normal. In this case, you would want to use automated tests that ensure full coverage of the code and data. These tests can be run constantly and need not be part of the develop build.

BREAKING OLD HABITS

Ideally, your develop mode should also be your release mode, which would automatically eliminate that bane of game development: the bugs that only show up when the game is in release mode. The problem here is that you've got to devote up to five percent of your CPU time—and a chunk of memory—if you want to leave in all your assertions. The solution is obviously to budget for that from the start.

Establishing a develop mode may seem like a lot to ask, since the harsh reality that we're used to is that we're often scrambling for a few thousand extra bytes, and CPU time is never adequate. That said, consider the benefits of having just one build configuration up to and including the version you ship. There's no risk of obscure bugs popping up due to changes in the code. If you ship for console, you can get back more useful information from publisher tests. If you ship on PC, you can add a facility for users to report assertions, which will allow you to get that patch out quicker.

If you are going to ship without assertions, for whatever reason, then make sure you budget enough time to test that version. The majority of your testing should be done on a version with assertions in, as you'll track down ordinary bugs much quicker. But at some point you'll need to test your final version. Just don't switch off the assertions the day before you submit. It actually might be a good idea to occasionally test with assertions removed, as the different code configuration might bring more obscure bugs to the surface. ✹

RESOURCES

Rabin, S. "Squeezing more out of Assert." In *Game Programming Gems Vol. 1*, ed. Mark DeLoura (Hingham, Mass.: Charles River Media, 2000), pp. 109–114.

Etherton, D. "Designing and Maintaining Large Cross-Platform Libraries." In G*ame Programming Gems Vol. 4*, ed. Andrew Kirmse (Hingham, Mass.: Charles River Media, 2004), pp. 37, 38.

Hunt, A. and David Thomas. "Leave Assertions Turned On." In *The Pragmatic Programmer: From Journeyman to Master*. (Reading, Mass.: Addison-Wesley, 1999), pp. 123, 124.

STEVE THEODORE

PENCIL WHIPPED

IN MY CONVERSATIONS WITH OTHER artists, we always end up dreaming about the perfect art tool, a tool that is fast, responsive, intuitive, and flexible. We picture a tool that doesn't require a ton of pre-planning or complex left-brain gyrations, but just lets you get into the subject and pursue a vision.

That tool exists. Unfortunately, it's called a pencil, and while it's as fast, intuitive and as flexible as you could desire, it doesn't exactly scream, "next-generation graphics" to your average executive producer. For any task that's too complex for a pencil, alas, we're still stuck with typical CG software routine: plodding workflows and uncannily prescient planning.

MODELER'S GRIDLOCK
Of all the disciplines, it's modeling that suffers the most from the tyranny of the graphics toolbox. When polygon counts hovered in the hundreds, it was possible to manage every vertex by hand; but as polycounts get larger, the burden of strategizing gets heavier and heavier. Nowadays, technical obsessions, such as NURBS tangency or subdivision vertex valences, suck up a lot of the energy that should be going into experimenting, improvising, or just plain messing around with new designs.

This month, we're going to look at a novel workflow that aims to bring some liveliness back into the art of modeling.

CLAY IS MIGHTIER THAN THE MOUSE
The pencil might not be quite enough for game art, PENCIL WHIPPED, of course, being the great exception (see Figure 1). But one way around the frustrations of high-poly modeling is simply to go back to clay and wire.

Although the game industry tends to prefer all-digital models, some Hollywood studios still prefer clay models to computer ones because they offer both creative flexibility and a reassuring physical presence that can help sell a character design during development.

Clay excels at developing organic forms, although in the hands of a good sculptor it can also embody mechanical forms. Even today, many top industrial designers still rely on clay in preference to computer models.

Clay has a much lower rocket science quotient than CG techniques, but of course, it's not magic. Like with most crafts, building up proficiency in traditional sculpture takes time and dedication, so it's unlikely most of us will find our workstations replaced with potter's wheels anytime soon. Nevertheless, many CG modelers get big creative benefits from working in clay, if only for hobby projects.

FROM PUTTY TO SCREEN
The barriers to getting traditional sculptures digitized into usable CG format have fallen dramatically. A contact digitizer like the MicroScribe from Immersion (see Figure 2, page 38) can be had for around $5,000. Digitizing service bureaus can turn around a clay model, or even a full size human scanning job, for much less. One minor caution to sculptors: contact digitizers like the MicroScribe work better on hard media, such as baked Sculpey, because the tip of the digitizing sensor can sink into softer clays, marring the model and also messing up the scan data.

If you don't have traditional clay skills or are dependent on the comforting presence of the Undo button, there's also Zbrush (see the review in Skunk Works, May 2005).

Zbrush is a paradox for many experienced modelers. It defies most of

the conventions that we've lived with ever since the hoary days of WaveFront and Alias PowerAnimator, which makes it extremely disorienting, even frustrating for old pros. Still, it gets closer than any other 3D application to the ideal of the pencil, offering speed, simplicity, and freedom to experiment.

As interesting as the Zbrush interface is—and that's a subject for a whole column of its own—the real key to the program's power is really just astonishing poly throughput. On my undistinguished machine, a 4 million quad model is still fairly interactive.

While Zbrush is ultimately a subdivision surface modeler, relying on the same math as a Max MeshSmooth or Maya Subdivision node, the profusion of polys lets us bypass many of the painstaking stratagems we need to generate subtle details. Subdivision basics like edge loops and vertex valences don't exactly go away in Zbrush, but they matter much less there than they would in a conventional face-and-vertex modeling environment.

LIMITS FOR GAME ARTISTS
Old fashioned clay and newfangled Zbrush share two important characteristics. Both mediums are great for chasing an elusive idea, experimenting and iterating without a clearly preconceived plan. Unfortunately, they're also alike in the kind of data they output: zillions of tiny polygons. Even if games were capable of displaying the

STEVE THEODORE *started animating on a text-only mainframe renderer and then moved on to work on games such as HALF-LIFE and COUNTER-STRIKE. He can be reached at stheodore@gdmag.com.*

extremely heavy meshes which both processes create, necessary tasks like texturing and animation setup would slow to a crawl. Unlike big movie studios, few game companies can dedicate a whole squad of technical directors to easing multi-million poly models through the pipeline.

This doesn't mean that heavy meshes are useless; it simply means that an extra step is needed to transfer information from that heavy mesh to a lighter, game-friendly one.

Messy, inefficient, ultra-high poly models are the natural complement to normal mapping. In a normal mapping workflow, the modeler can concentrate on the spirit of the design while evolving the high-resolution model and backload the technical problems—efficient UV mapping, animation-friendly geometry, and rigging—onto the creation of the game asset model. This may not be a complete liberation from the computer-y parts of computer graphics, but at least it allows you to concentrate on the creative aspects before worrying about the mathematics.

If your game engine is normal-map friendly and your high-resolution models don't completely sink your interactive performance during the technical setup phase, you may not need to worry about refining the process further. There are many situations, though, in which you may need to consider a different strategy. If your game engine doesn't

support normal maps, or if the costs of using them everywhere are busting your texture budget, you may need to create medium-resolution models—models with tens of thousands of polys, but not hundreds of thousands or millions.

Medium-resolution models can be awkward because they are impossible to work with vertex by vertex. Yet at the same time, they are definitely subject to efficiency considerations that can be ignored when working on super heavyweight models.

RETOPOLOGIZE

If you're working on mid-resolution models, you don't have to completely forgo the benefits of fast and furious high-poly tools. You can still separate out the creative and technical components of the process by using your high-resolution model as a foundation, building a new mesh or subdivision surface over it. This technique is known as retopologizing, and it can be extremely useful.

The simplest form of retopologization involves nothing more complex than old-fashioned poly modeling. Using the Make Live function in Maya, or the Snap-to-Face mode in Max, you can build a low-resolution mesh right onto the surface of a high-resolution model.

This technique may seem unduly slow, since it does involve a point-by-point rebuild of the entire model. However, it's actually much faster than a traditional poly model because using the high-

resolution model as a snapping template means that you only need to worry about two dimensions instead of three, so laying down faces goes very speedily. Tricky technical problems such as deformation-friendly shoulders are also easier to tackle when the basic volumes are firmly established by the high-resolution mesh.
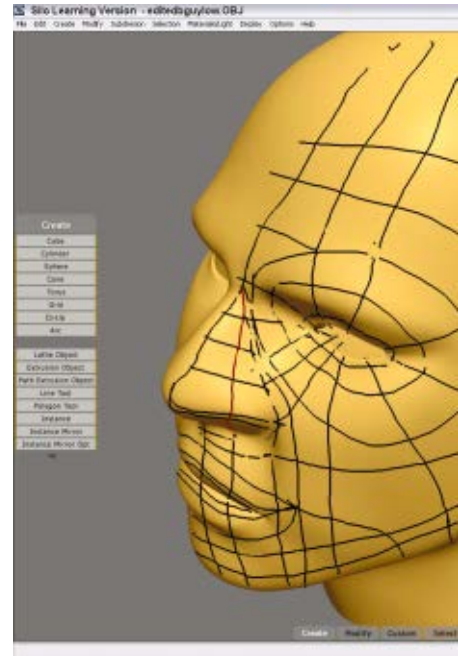
Even if you don't need to work on medium-resolution models, simple retopologization of this kind is a great way to quickly build up the low-resolution shells for normal-mapped assets.

If you find yourself doing a lot of this kind of retopologizing, you should definitely take a look at the topology

FIGURE 4 **CySlice is a program dedicated to retopologization.**
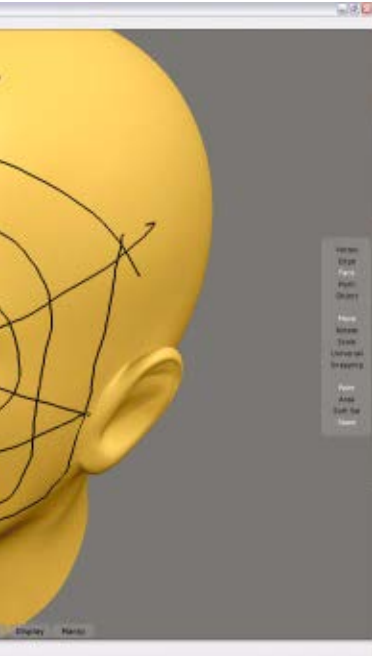
brush feature in the modeling package Silo (see Figure 3).

The process is very fast and mostly painless, although the method of selecting and editing existing strokes can be a bit confusing. It's certainly easier than accomplishing the same task with snaps, and since the entire package sells for only $109, it's definitely worth a look.

On the higher end of the budget scale is CySlice, a dedicated retopologizing program (see Figure 4). CySlice, which ships with most cyberware digitizers, will generate NURBS or subdivision patches from a high-resolution mesh. Part of CySlice's appeal is that it can also extract displacement maps and color maps automatically from the original geometry. This feature has become less compelling nowadays with the common availability of normal casters, which can do the same thing.

BRICK BY BRICK
Even with the assistance of the fancy tools, building models with more than a couple of thousand faces is too cumbersome to do point-by-point. Fortunately, you can use the snap or topology brush tools to lay down a solid edge-loop network as the foundation for a new subdivision model (or simply get your

subdivisions straight from CySlice). This is a popular technique with modelers in the film business. Even though they aren't bound by our performance considerations, they use this trick to separate the technical business of edgecraft from the intuitive parts of modeling.

It's likely that your new subdivision surface will appear a fraction smaller than its high-resolution original. Subdivision surfaces don't pass exactly through their control points. If your subdivision control mesh is fairly dense, this effect won't be very pronounced; but if it is, you can usually correct it by moving all the control vertices outward along their own normals. Alternatively, you can use the subdivision to generate the extra faces you need and then use a script tool to glue each vertex in the subdivision mesh to the surface of your original high-resolution model. You can even use a normal caster to transfer the finer details from your high-resolution mesh to the new mid-resolution mesh as a bump or displacement map, allowing you to keep fine details, like clothing wrinkles, that even a 20,000 poly model can't afford to represent with geometry.

REINVENTING YOUR ROOTS
Retopologizing makes for more powerful workflow, but it's still a long way from the mythical 3D pencil. Working loosely on a high-resolution source model—whether in clay or in Zbrush's digital clay—and then retopologizing lets you push the technical and resource constraints into the background.

Alas, those constraints certainly don't disappear altogether. The freedom to

experiment comes with a cost. You need to build your model twice, and even though the second run through is much faster than the first, it does take time. Moreover retopologizing is a one way gate, so you need to be well satisfied with your high-resolution original before you move on to building a new mesh on top of it.

If you can live with those limitations, however, you'll have a workflow that's very powerful and also a lot of fun. Maybe someday we won't need to obsess over every detail of the construction process, but for the time being we can at least enjoy ourselves a little. ✂

>> BUSINESS LEVEL

LICENSED TO STEAL

OVER THE YEARS, I HAVE HAD THE opportunity to close licensing deals for famous mice, infamous singers, speedy hedgehogs, and classic monsters; and more recently, I have been a soldier in the battles for the major sports franchises. Licenses have become an important part of our business—and they're not going away. Some games need to be based on licensed properties. It's hard to imagine a competitive sports sim today without a league and players' license behind it. Yet if it were left to me, I would opt for creating original intellectual property every time.

WHY WE LICENSE

The economic case for basing a game on licensed properties is straightforward. A famous property brings with it a built-in audience and brand recognition. A recognized title is more likely to get shelf space at retail. Moreover, a successful movie has millions of dollars of advertising that indirectly supports the game. We have even seemed to work out the old bugaboo of trying to release the game with the movie. Now if we can't make the theatrical release, we get a second chance with the DVD release, which comes just a few months later, before the buzz of the film has completely faded.

A game based on a licensed property is a good hedge to ensure that a publisher will at least be able to sell enough units to recoup its investment. Making games from licensed properties is a little bit like attaching a well-known star to a movie. It may cost more to produce, but it also minimizes risk.

However, our industry is at its best when it takes risks.

When I think back, I can't recall a game based on a licensed property that was innovative, groundbreaking, or awe-inspiring. But I do remember being awed by the speed of Sonic the Hedgehog, intrigued (but also bored) by Myst, perplexed with Mario's ability to freely roam on the Nintendo 64, and blown away by the 3D fighting in Toshinden. Maybe I haven't learned much over the years, but the one thing I am sure of is that our industry is at its best when developers and producers work on products they imagine, create, and ultimately develop themselves, all the while pushing the limits of whatever system they are developing for.

The game industry is just coming out of its equivalent of the film industry's silent era. For the first time, we are on the verge of an era when the technology will not impede the designer's imagination. It's a time when the industry will be redefined.

The scale of game production has become staggering. The skill and ability of game developers is amazing, but production costs have become frightening. Basing a game on a successful licensed property is not a bad decision, given all that can go wrong during development. In fact, the right license may ensure substantial success.

SKIRTING CREATIVITY

But there are some nasty elements to the economics of licensing. For one, not all licensed properties succeed. Anyone still playing The Last Action Hero?

Second, while the license may buy a built-in audience and retail shelf space, if a game is successful, ultimately, the licensor will extract an ever-larger share of the profit. You simply can't publish an NFL licensed game without financially involving the NFL. Licenses can become licenses to steal.

Third, and most concerning for our industry, the licensor can control the creative process. Even if the licensor is completely hands off (and most are not), the very fact that the property was designed for another medium establishes consumers' expectations and dictates the manner in which that property is portrayed. There is no getting around that a licensed property is a derivative work. If the property is based on a film, the consumer often expects to be able to reenact scenes from the movie or progress through the game in a sequence very similar to the movie. Developers are seldom given carte blanche to innovate in a licensed game.

It's true that allowing developers to create new properties in the short term is a huge risk, but in the long term it may prove to be the more financially sound decision. If a new property becomes a successful franchise for the publisher, the dividends that it repays are the licensing fees the publisher would have otherwise had to pay to the licensor, and the ability of the publisher to create sequels, derivatives, and licensing revenue of its own. Crash Bandicoot, Twisted Metal, and SOCOM all produced significantly far more revenue for their publisher than most games based on licensed properties. I believe that these dividends are funds that should be reinvested in our industry to create even more new properties.

OUR OWN TWO FEET

Too often our industry is viewed as the stepchild to the other entertainment industries. The game business is younger than film, music, television, and book publishing, and is just coming into its own as an entertainment medium above and beyond the previous portrayal of games as children's toys. To rely too much on properties created for the other entertainment industries will stifle the potential for new creativity and perpetuate our industry's second-class status in the minds of media consumers.

Groundbreaking games, not licensed properties, will define interactive entertainment. We should try to get there sooner rather than later. ✖

RILEY RUSSELL is general counsel for Sony Computer Entertainment America and oversees legal and business affairs for the company. Email him at rrussell@gdmag.com.

Note: All statements are my own opinion and not the opinion of my employer or former employer. I am an unapologetic video game lawyer, love the industry, and as such, admit that there might be more than a hint of bias in my opinions.

›› GAME SHUI

NEGATIVE FEEDBACK

AS MANY OF MY CLIENTS CAN ATTEST from my design advice, the following rule is one of my favorites: *Use negative feedback to balance variations in game difficulty and player skill.*

Negative feedback, when used properly, is one of the hallmarks of great game design, and yet it is under-appreciated and under-utilized. In fact, I continue to see comments that games need more positive feedback, surely equivalent to saying that fast food restaurants need to include more fat in their cooking.

Providing feedback involves returning a portion of a system's output to its input. If the feedback enhances the trend of the system, for example making an increasing output increase faster, or a decreasing output diminish faster, it is called positive feedback. It can make a microphone squeal, or a car skid on ice.

Negative feedback, logically enough, works the opposite way. If an output is increasing, negative feedback slows it down and vice versa. Think of a thermostat cutting off power to a heater when the temperature in a house rises.

Positive feedback is designed into most games in many ways. It's hard not to put it in. A player in an RPG kills an enemy, and gains a special sword, making the next battle easier, and as a result of winning he gains enough gold to buy new

Sid Meier's CIVILIZATION series uses appropriate negative feedback to balance the games.

armor, and so on. It's fun at first, but the character may grow to be so powerful the game becomes boring.

Or consider a different player in the same game who loses a battle and is so weak that winning the next battle becomes nearly impossible. Positive feedback can be a dangerous tool.

Negative feedback has often been implemented by designers to keep good players from getting bored or poor players from becoming discouraged. Thinking about that same RPG, negative feedback might take the form of introducing a few higher-level enemies to balance the power of the stronger player, or some extraordinarily weak rats carrying implausible amounts of gold and health potions to help the weaker one. It's not necessarily logical, but to the player it makes the game seem fair, and hence more fun.

THE RULE'S DOMAIN

The rule can apply to any game which has rewards and challenges that change over time—an overwhelming majority of video games. It is often used to best effect in games that would otherwise become quickly unbalanced, making it easier for weaker players to rebound from a minor setback. It's particularly important (and difficult to apply) to games with multiple human players, as the implementations must feel fair to all players, while computer opponents can be designed to act in the player's favor if necessary.

RULE IT TRUMPS

The negative feedback rule trumps this one: *Make rewards proportional to the difficulty of the task required to earn them*. Though it's a good basic rule, almost all games are more exciting if the player who does well feels that victory is always in question, while the player who does poorly feels that there is still a chance to come from behind and win.

RULE IT IS TRUMPED BY

The rule regarding negative feedback is trumped by this one: *Make the game appear to be fair to the player*. Some designers shun negative feedback because they assume that it amounts to punishing the victor and rewarding the vanquished, which seems inherently unfair. But this is where the artistry of proper use of negative feedback comes in.

Many military games impose an increasing maintenance charge on military units, charging more for larger or more sophisticated armies, which results in reducing the funds available to create new units, making it easier for weaker opponents to make up lost ground.

Another very common way that negative feedback is applied is the practice of most RPGs requiring that a character have a geometric increase in experience points to advance to the next level, while the points available increase more slowly, often linearly. Also, realize that a game should appear fair, but it's fine to achieve this by changing underlying and hidden algorithms to increase the players' enjoyment. It's the illusion of fairness—not fairness itself—that's important.

EXAMPLES AND COUNTEREXAMPLES

The negative feedback rule is used throughout good games. It shows up everywhere in the CIVILIZATION series, for example, to balance the player's civilization against the others' in the game, handicapping it with increasing tax and "citizen happiness" spending.

In MARIO KART 64, the player who is in the lead can win power-ups, but (and one might notice the subtlety of this trick) they are fairly useless. On the other hand, when the player in last position earns a power-up, it's highly powerful.

I'd love to hear more examples from readers of their favorite uses of negative feedback too—even as it is applied to my design of this column. ✖

NOAH FALSTEIN *is a 25-year veteran of the game industry. 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****.*

ALEXANDER BRANDON
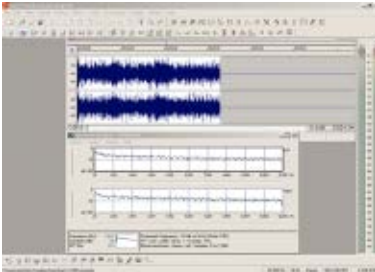
BRING THE NOISE
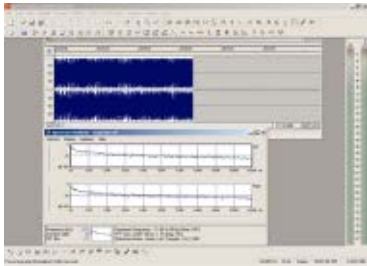
SINCE GAME AUDIO HAS emerged as something comparable to film and television audio (in all areas except voice acting), expectations and paradigms have changed. Ten years ago, anything was fair game, and the closer in-game assets were to film or TV in quality, the more people were impressed.

Now, audio developers worldwide are hearing an almost constant mantra of requests from producers and project leads: "I want more bass," and, "That needs to be more powerful!"

Sound designers constantly fight back, chanting their own mantras: "You have no idea what you're talking about," and, "I can't possibly do that!"

Both groups have good points. Games need to stand out in the marketplace, and bigger and badder sounds will accomplish that.

Unfortunately, big and bad are relative terms when it comes to audio. But there might be a way to satisfy both the lead and the audio developer so everyone is happy, hopefully, once and for all.

THE LEAD: DYNAMIC RANGE

Every project lead needs to understand that you have a limited amount of bandwidth to play with. Imagine your old boomboxes and Walkmen from the 1980s, with their nifty graphic EQ controls for high, mid, and low (treble and bass, as they are also known) frequencies. If two sounds are playing with the same narrow frequency range, they add significantly to the loudness of that range, while everything else is softer. There's only a limited amount of loudness (known as headroom) to go around within each range, so once you go beyond it, the sounds begin to distort. Before that, both sounds are just muddy. So if you have two sounds that are bass heavy with slightly different frequencies within that range, you stand a better chance of creating a cleaner and more professional sounding mix than if you have eight or more sounds with mostly the same bass frequencies.

Take a look at Figures 1 and 2: a fairly well rounded piece of looping music with a bass heavy range and a piece blasted to the max with bass. Figure 2 will sound wonderful in a coffee-can exhaust Honda Civic with the windows down, but it will do no good for in-game sound effects that might have any bass. They will be lost to the ear.

In addition to headroom, you have to consider your delivery system. Are you playing on a TV with tiny speakers, or on a full blown B&W Nautilus system? The difference in headroom and ability to play more frequencies at a louder volume is tremendous between a television and a $300 surround sound system. The problem is, you need to consider these delivery systems separately. You won't get great sound on a TV speaker; you will on a home theater system.

SOUND DEPARTMENT: COMPROMISE

Understand that sounds make graphics pop when done properly. Interactive mixing using SCREAM and/or XACT, or your own custom system is just the beginning of creating a solid mix that will fill a cushy home theater or impress someone with a standard 12-inch television.

Start with understanding what the design needs of your title are. If your title is action based, be ready to set priorities, ducking groups, and mixing requirements for each level as much as possible before testing them in the game.

Your initial assets don't have to sit there waiting. Get a demo with non-textured art assets or even an animatic and plug away at a prototype. Show the prototype to your leads and have them get a good idea of what the soundscape will be before anything graphical is ready. Yes, friends, it is the heyday of audio concept art.

Once you're in production (and here's the real pearl I learned recently) and you get yet another request for more power or more bass, demo your title next to a comparable one in its category. Recently, I did this with SOUL CALIBUR 3. It's easy to notice how un-spiffy a gigantic swing of an axe might be on a TV in your fantasy themed title, but watching the character make the same swing and have the same weak sound will prove the point that TVs just aren't good delivery mediums for bone crunching and heavy linebacker tackles. The same goes for music and voice over.

Finally, demo and compare a TV to a pair of monitors as well as a surround system to your leads. Do this early in the project so you can filter those wasted requests for a "badder" sound. However, if it still doesn't jam on the more high-end systems, you've got work to do.

TURN IT UP!

With these guidelines, you should be able to satisfy everyone on the team with what the concept of a mix is, and sound designers can better address concerns of a demanding team. After all, we wanted this kind of creative control, but with great power comes great responsibility. Audio sense ... tingling! ✖

ALEXANDER BRANDON *has been involved with game audio since 1994 and is currently the audio manager at Midway in San Diego, Calif. You can email him at abrandon@gdmag.com.*

Seapine Software

Winning Game Development with Seapine CM

Successful game development depends on a team of designers, graphic artists, developers, testers, and managers all working toward one goal — delivering the next mega-hit on time with no bugs. Seapine CM, Seapine Software's change and issue management solution, manages and tracks your team's source code, digital assets, bugs, change requests, and issues throughout the development lifecycle. Whether your process is agile or structured, cross-platform Seapine CM promotes team collaboration with features such as advanced code branching, secure remote access, the ability to link code changes to issues, file versioning and archival, and seamless integration into your development and build processes.

Flexible change management

With Seapine CM you know not only what files changed but why. Seapine CM combines TestTrack Pro, an award-winning issue management solution, and Surround SCM, an award-winning version control solution, into a change management powerhouse. This combination allows team members to associate source changes with defects, change requests, and feature requests—even while checking in changes from their favorite integrated development environment (IDE).

To maximize productivity, change management tools must also seamlessly integrate with your other development tools. Seapine CM works with a variety of popular development tools including Visual Studio, JBuilder, and Dreamweaver MX. Administrator-definable triggers, programs you can create that run before or after file actions (e.g., check in, promote), give you additional control over your change management process and allow you to extend Seapine CM's functionality to fit your company's change and build management processes.

Effective project management

Keeping track of your team's digital assets is simple with Seapine CM. From tracking the who, what, and when of your team's file revisions to providing one-click access to bug histories, Seapine CM helps you manage your projects more effectively. Monitor quality control statistics such as who reported the most bugs, how many bugs are open, or how much time a developer spends fixing bugs for increased issue tracking efficiency. Easily archive and retrieve virtually any file for improved change management.

Tracking file versions has limited benefit without a structured branching methodology. Seapine CM features unlimited branching to accommodate your team's development processes. In addition, Seapine CM's private branches enable team members to work independently on common source files and digital assets. Confusion and file overwrites become a thing of the past because changes remain isolated, until team members are ready to share changes and promote them up the branch. With the latest quality-controlled build always protected from changes downstream, development and software quality assurance team leaders can review and approve changes with ease.

Secure, remote access

Protect your intellectual property. Seapine CM's role-based security gives you strict control over who can access issues, files, and commands. Geographically dispersed teams will benefit from fast, reliable remote access to issue databases and source repositories with Seapine CM's secure 512-bit encrypted communications protocol. Advanced password rules further ensure your valuable data is not compromised.

Streamline your game development efforts with Seapine CM and help your team deliver higher quality products on time.

Seapine Software, Inc.
5412 Courseview Dr., Suite 200
Mason, OH 45040
Tel: (513) 754-1655

www.seapine.com/cmsuite.html
jenny@seapine.com

SELECTED FEATURES

- Complete issue management
- Versioning and branch management
- Image thumbnail support
- Triggers, shadow folders, and email notifications
- WebDAV and Unicode support
- Windows, Linux, Mac OS X, and Solaris support
- Integrate with popular development IDEs
- Link source files with bugs, change requests, and other issues
- Advanced 3-way diff/merge utility

IN FOCUS

Rocketbox Studios GmbH
Vahrenwalder Str. 7
30165 Hannover, Germany
Tel: +49-511-9357-280

www.complete-characters.com
info@rocketbox.de

SELECTED FEATURES

- 100 different customizable characters
- Optimized for real-time applications
- Each included in 3 different polycounts (600, 2500 and 5000 triangles)
- Texture size 2048x2048 pixel
- Diffuse textures in Photoshop PSD format with many changeable layers
- Normal bumpmaps
- Specular maps
- Skin weights and bone skeleton
- 100 animations included
- Compatible down to 3ds Max 5
- Maya version available shortly
- Company license priced $3199 USD

Complete Characters
by Rocketbox Studios

A Real-Time graphics library with customizable human characters

Complete Characters is a new product that is suited for real-time applications and therefore also meets the special needs of game developers. It is an extended form of a graphics library that consists of low and medium polygon 3D models of human characters with customizable textures and animations. The three DVD package is developed and distributed by Rocketbox Studios GmbH and contains 100 characters of different ages in contemporary clothing.

As the meshes are included in three different polycounts (approximately 600, 2,500, and 5,000 triangle polygons), possible applications range from mobile and handheld development to the creation of games and applications for PCs and next-generation consoles. Rendering stills and movies is also possible because of the high resolution of the textures and the quality standard of the characters.

The inside of the mouth is also modelled and textured for the medium- and high-poly versions of the characters, so a facial animation set-up can be realized if required. All characters have a bone skeleton and skinning, which is optimized for use in real-time engines. For example in the low and medium versions, no vertex is influenced by more than two bones. Skin weights are set exclusively on vertex basis.

The texture maps have been designed to create photorealistic impressions of human characters. The textures have a high resolution and are in an uncompressed format, therefore they can be customized to meet the requirements of various applications. Each texture comes as a 2048x2048 pixel PSD file whose layers are not collapsed. Texture elements like hair, skin, cloth color, cloth structure, drape/folding, joints, buttons, logos, add-ons, etc. are arranged on separate Photoshop layers. This means that the character texture can easily be modified and customized. The result is a unique set of characters for only a small investment of work. All characters have similar UV mapping coordinates, which makes it easy to interchange texture elements between them.

Each texture has an additional normal bump map and specular map. The normal maps have been created in Photoshop by mixing high polygon sources and converting layers of grayscale height maps.

100 ready made animations are also included in the library, consisting of animation clips and loops commonly used in games, such as walk and run cycles, a drop-dead motion, and a motion for a character picking up an object. The keyframe animations are also optimized for better performance and easy modification.

Complete Characters is mainly designed for the use with 3D Studio Max (version 5 and higher). The animations included in Character Studios are in .bip format. A Maya version is planned for release November 1.

A company license is available for 2,499 euro (or $3,199 for U.S. companies) and can be used for an unlimited number of projects and for an unlimited time. Additional information as well as image and video files for download can be found at www.complete-characters.com.

metrowerks™

CodeWarrior™ Game Development Technology — Stable, High Performance, Full Featured

Metrowerks provides a complete suite of development options, enabling titles to stay on schedule and ship on time. Leading studios and publishers have come to depend on Metrowerks technology from compilers, to debuggers, to analytical tools, to IDE — We understand game development, and have been in the industry since 1996. We offer development and analysis tools for the Sony PlayStation® Portable, Sony PlayStation®2, and NINTENDO GAMECUBE™, including support for leading third-party plug-ins. Our customer roster includes the world's top console game developers and publishers. Look under the cover of best-selling console games and you will see CodeWarrior™ game development tools in action.

CodeWarrior Development Studio for PlayStation® Portable — Offers a comprehensive and consistent environment for fast and easy application creation, including a cross compiler for blazing fast, high-quality object code. The CodeWarrior debugger offers developers the quickest way to find showstoppers, and the intuitive graphic interface offers flexible and accurate debugging capabilities.

CodeWarrior Development Tools for PlayStation®2 — Provides advanced editing capabilities, a powerful and optimized MIPS® compiler, and debugs STABS and DWARF formats. This tool suite makes it easy to port code to other game consoles, allowing you to write code once and then optimize for each platform.

CodeWarrior Game Development Tools for NINTENDO GAMECUBE™ — Includes a highly optimized C/C++ compiler and an assembler designed for the GAMECUBE processor, plus a wide array of debugging support including C++ browsing for all available GAMECUBE development systems.

CodeWarrior Analysis Tools — Analyze and optimize your game with CodeWarrior Analysis Tools for the NINTENDO GAMECUBE™ and/or PlayStation®2 consoles. An intuitive GUI provides easy navigation — results are displayed graphically in charts, lists and call graph trees.

CodeWarrior Analysis Tools with Zero Intrusion Profiling technology for the Sony Computer Entertainment Performance Analyzer — *Available for the PlayStation®2 Platform*. Zero Intrusion Profiling Technology from Metrowerks analyzes the raw data from the Sony® Performance Analyzer and generates detailed and exact profiling information on the application, without intrusion. Zero Intrusion Profiling then displays this information in an intuitive graphical user interface, complete with a Source View, a Hierarchical View and a Call Graph View.

CodeWarrior Communication Utility — A client-server application for designers, artists, and programmers who develop games for the PlayStation® Portable, PlayStation®2 and/or NINTENDO GAMECUBE™. Game artists can change texture, characters, and other 2D/3D animated scenes without recompiling or reloading the entire executable.

"CodeWarrior Analysis Tools are an invaluable asset to Stormfront's development cycle. The hierarchical profiler helped us quickly and effectively eliminate bottlenecks, and the code coverage tool was very helpful in improving our quality assurance process. CATS was a key tool in helping us ship Demon Stone on schedule."

Ken Chao, Technical Director, Stormfront Studios

Contact games@metrowerks.com to obtain a free evaluation copy of our tools*.

Metrowerks Headquarters
7700 West Parmer Lane
Austin, TX 78729
USA
Phone: +1 512 996 5300
Fax: +1 512 996 4950

www.metrowerks.com/games
Email: games@metrowerks.com

SELECTED FEATURES

- Tools for Sony® and Nintendo® platforms
- CodeWarrior Analysis Tools for optimizing code
- C/C++ front-end compiler
- Multi-core debugger
- Support for multiple IDEs
- In-person consultations available
- Support for leading third-party plug-ins

*To obtain an evaluation copy of CodeWarrior tools, you must be a licensed developer with Sony Computer Entertainment (SCE) and /or Nintendo.

©2005 Metrowerks. Metrowerks and CodeWarrior are trademarks or registered trademarks of Metrowerks Corporation in the United States and/or other countries. All other trademarks are the property of their respective owners. All rights reserved.

N FOCUS >>>>>>

FOCUS ON

Annosoft

Annosoft LLC
2050 N. Collins Blvd. #117
Richardson, TX 75080
Phone: 214-427-8183
Fax: 214-227-9489

www.annosoft.com
info@annosoft.com

SELECTED FEATURES

> Advanced Text Based Lipsync in multiple languages
> Reliable lipsync without a text transcript
> Unique articulation system for smooth animations
> Fast support from the programmer who wrote the software
> Flexible output that can fit any animation architecture
> ActiveX Control and Batch Processing for scripting
> Excellent resume and company history

Annosoft provides SDKs and Tools which automatically generate realistic lip synchronization information from audio content. Annosoft offers C++ SDKs, an ActiveX Control, Batch Programs, and The Lipsync Tool, a standalone software product.

Founded in 2002, Annosoft provides technologies that help solve difficult problems associated with the production of non-player characters in games. Annosoft software is generally used at production time and is good fit for customers with proprietary production components.

Annosoft's most popular product is the **Text Based Lipsync SDK**. This software is used to produce precise lip synchronization from audio content and a text transcription. It gracefully handles transcription mistakes and works on short or very long audio files. It is speaker independent and supports English, French, German, Italian, and Spanish.

Artists appreciate the **Textless Lipsync SDK**. It produces realistic mouth animations without the need to write a text transcription. It may lack the some of the precision of its sister, but in most cases, it generates excellent lipsync, and has seen heavy use in many games. It is speaker and language independent.

The Lipsync Tool is a phoneme editing tool that can be used to generate and visually edit lip synchronization. It can be included in the SDK as source code or licensed as a standalone product.

Annosoft technologies are animation independent. The software generates generic information describing the shape of the mouth over time. This output is extremely flexible, providing precise mouth specifications without forcing the customer to a particular character rig, animation architecture, or runtime system.

Reliable automatic lip synchronization is a difficult problem. It involves developing a robust acoustic processor to extract speech information and a physiologically correct articulation system to produce smooth mouth transitions. Annosoft has spent years on these two problems and has developed reliable technologies that are helping game developers produce higher quality animations, faster.

For free evaluations, contact Mark Zartler (mzartler@annosoft.com), or visit us on the web at www.annosoft.com.

"The software has been a dream to use - certainly one of the best 3rd party packages I've worked with. I'll be recommending you."
- Mark Adami, Lionhead Studios.

"Not only is Annosoft fast, but the results are great and could be put into our game without tweaking. With the added ActiveX Control we were able to easily work Annosoft directly into our existing facial animation and export tools. In crunch time Annosoft helped us get over a thousand facial animation files into our game in just a few days."
- Jason Greenberg, Neversoft Entertainment

"Annosoft lipsynced a total of 10,000 lines in four separate languages; it's worth its weight in animators. Support has always been prompt, effective and rarely needed."
- John Scott, Raven Software

FaceFX: Next-Gen Facial Animation Software

Creating facial animation for next-generation titles can be a daunting task. Realistic content can require a combination of bone transformations, morph targets, and material properties. Procedurally controlling these targets for a dynamic solution adds an additional layer of complexity. Motion capture and hand animation solutions do not scale well to thousands of animations, and audio-based solutions have traditionally relied on sub-par phoneme segmentation software that only supported one or two languages.

OC3 Entertainment is bringing facial animation software out of the Dark Ages with FaceFX. If you are struggling to find a facial animation solution that meets your needs here are four reasons to consider FaceFX:

Partnership with Fonix
Great audio-based facial animation starts with great phoneme segmentation. OC3 Entertainment has recently partnered with Fonix, a leader in the speech recognition industry, to provide the most accurate audio-based facial animation solution on the market. Fonix VoiceIn™ speech recognition technology supports US English, UK English, French, Italian, Spanish, German, Korean, and Japanese. OC3 Entertainment and Fonix are also working together to provide a real-time facial animation solution that will put a new face on in-game voice chatting.

Integration with Your Engine
FaceFX ships with full source code and a sample rendering engine. The sample rendering engine can easily be replaced with your game engine for an optimal facial animation pipeline. Why go back and forth between a 3D animation package and your engine when you can instantly see how changes will look in the game?

The Face Graph
You might be wondering how the automatically-generated curves interact with procedural effects like gaze direction, or artist-created curves for emotion and other effects. The answer is simple: with the Face Graph! With FaceFX, there is no need to write custom code to control facial systems that need input from your game at runtime. The Face Graph is evaluated at every frame of the game and lets artists create complex relationships between targets using animation curves and in-game values as inputs.

Flexible Use
From hand-animating to batch-processing to real-time analysis, FaceFX has you covered. Some customers use FaceFX to hand animate from scratch within their game engine. Others prefer to write out animations into their own custom file format. Still others batch process some animations while tweaking important dialogue sequences to perfection. No other solution provides so many different ways to get your characters talking.

OC3 Entertainment, Inc.
1133 Winter Walk Cir
Morrisville, NC 27560
Website: www.oc3ent.com
Phone: (919) 460-4564
Fax: (919) 460-8993

www.oc3ent.com
info@oc3ent.com

FEATURES

- Compatible with any facial rig
- Full source code to tools and SDK
- Speaker independent analysis
- English analysis available now. French, Spanish, German, Italian, Korean, and Japanese available soon
- Generate head rotation, blinks and eyebrow movements automatically
- Tweak animations with FaceFX Studio
- Batch process animations
- Memory efficient animation data
- Supports bones, morph targets and everything else
- Support team dedicated to the success of your title
- Solutions for pre-recorded dialogue or in-game communication

gamedeveloper
FRONTLINE
AWARD '04

CRI Middleware Co., LTD.
650 Townsend Street
Suite 650
San Francisco, CA 94103
+1 (415) 701-3683

www.cri-mw.co.jp/eval/119
bizdev@cri-mw.com

CRI SOFDEC

- Cross-platform, ultra high quality video codec
- Real-time 3:2 pulldown film/ NTSC/PAL conversion
- Multiple language and subtitle tracks in one movie file
- Real-time effects: movie-in- texture, Z movie, alpha movie

CRI ADX

- Cross-platform audio codecs, file system API and file packaging tools
- High-performance asynchronous streaming of voice, music or any kind of game data
- Low CPU requirements: about 0.1 percent for one music track, about 1 percent for one voice track
- Highly efficient at managing large numbers of samples

Game Developer 2004 Front Line Award Winner

CRI Middleware means top-quality video and audio codecs. CRI Middleware has been battle-tested in over 1000 titles from nearly every publisher, including top-ten hits like ESPN NFL Football, Halo 2 Multiplayer Map Pack, Sonic Heroes, Soulcalibur 2, and DragonBall Z.

When you use CRI Middleware, you leverage Sega's four decades of arcade history. And CRI Middleware is ready for the future: we're an authorized middleware provider for Sony PLAYSTATION®3 and Microsoft Xbox 360™.

CRI Sofdec: Movie quality matters

You've spent countless hours perfecting your cinematic sequences. So why surrender all that visual detail to a cheap video codec? Introducing CRI Sofdec! CRI Sofdec integrates quickly and cleanly into your existing game engine. And with dozens of top-ten hits, it's the highest quality video codec available anywhere. For current and next-gen consoles, PC, and handheld systems. When quality matters, use CRI Sofdec.

CRI ADX: Rock-solid audio streaming

ADX is a suite of high-performance audio codecs for games. Its cross-platform compression format enables you to squeeze hundreds of thousands of high-quality voice and music samples into one game. The memory and CPU requirements are low and predictable—less than 1% per playing voice on most systems, and only 0.1% per music track! Think you can get that performance with MP3? Keep your frame rate up with ADX!

Best of all, ADX is a rock. Nine years in continuous development, ADX is a stable, time-tested API with over 1000 successful titles behind it. Tired of beating out the bugs in your current audio library? Use ADX!

"Over the last decade, Shiny has always been searching for new ways to create fresh gameplay experiences. Sometimes it can be done in surprising ways, like the crafty use of the extremely flexible CRI technology. Frankly, we've been amazed by the software quality, and even by the support that these guys provide!"
—David Perry, President, Shiny Entertainment

"CRI's code is lean, mean and effective. Their support staff is knowledgeable and prompt the few times we have had to talk to them."
—Gil Gribb, Lead Technologist, Raven Software (Activision)

"CRI Sofdec met our high bar for video quality for our HALO 2 Multiplayer Map Pack project, and their customer service was outstanding."
—Curtis Creamer, Producer, Bungie Studios

"Midway's experience with CRI's products on such titles as NBA BALLERS, MORTAL KOMBAT, GAUNTLET: SEVEN SORROWS, and others has been completely positive. They provide exactly what we look for in a middleware company: products that fit our needs, innovation to deliver new features, and strong support to back it all up."
—Paul LeFevre, Chief Technical Officer, Midway

Image from Tom Clancy's Splinter Cell Chaos Theory™

Massive Incorporated
627 Broadway, 7th Floor
New York, NY 10012
Tel: 212-228-2296
Fax: 212-228-2161

www.massiveincorporated.com
bizdev@massiveincorporated.com

KEY DEVELOPER BENEFITS

> Significant Revenue Per Title
> Realism that Enhances the Game Experience
> Fully Supported, Seamless Integration Process With No Disruption to the Development Schedule
> Complete Control Over All Ad Content
> Sophisticated Network Operations and Reporting
> Full Engineering and Creative Services Support
> Robust, Industry-Standard Security
> Massive Covers All Costs

Image used under permission from Ubisoft Entertainment. © 2005
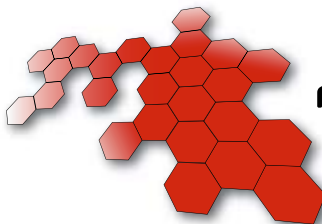
massive

YOUR CHANNEL TO THE GAMING AUDIENCE ℠

The Massive Network Provides Significant New Revenue Source for Developers

Massive Incorporated is the developer and operator of the world's largest Video Game Ad Network. Massive's technology consists of both an AdClient that is integrated into the game at the development stage and an AdServer that dynamically delivers ads through an online connection into the game environment.

Massive's technology enables the delivery of image, audio and video advertising, seamlessly into the game. Massive works closely with the game creators to ensure that all advertising preserves the game's integrity and is contextual to the environment and relevant to the gamer.

Massive's partnerships with 24 major publishers and developers have created a network that will include over 100 titles. By aggregating the world's largest video game audience, Massive is able to attract significant advertising revenue that is shared with developers and publishers.

Ads That Preserve and Enhance the Game Experience
Massive has invested over three years developing its technology and researching gamer preferences for video game advertising. Massive customizes every ad element to fit seamlessly into the game, enhancing realism and the overall gamer experience.

Complete Control
Development teams retain complete control over all ad content delivered into their games. The creative vision of a game and its functionality are never compromised. Ads never distract from or interfere with game play – period.

Seamless and Efficient Integration
Integrating a title into the Massive Network is an efficient process that requires minimal developer resources and runs parallel with the development schedule, with no disruption. Massive provides comprehensive integration support to developers at no cost.

Rigorous Quality Assurance
To ensure title integrity, Massive applies a rigorous QA process throughout a title's life-cycle. Every ad element and each ad campaign is tested in all titles prior to ship, both by Massive engineers and gamers.

Significant Revenue Source
The financial impact for developers and publishers is significant. Dynamic advertising is a new, significant revenue stream for the video game industry. Titles in the Massive Network are generating $1-2 in direct profit contribution per connected unit.

The Massive Network benefits all parties. Developers and publishers gain an additional revenue stream and enhanced realism. Gamers experience a more life-like environment. And advertisers can now reach a key audience through the world's most engaging and dominant entertainment medium.
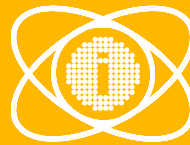
INDEPENDENT
GAMES FESTIVAL

REWARDING INNOVATION IN INDEPENDENT GAMES

THE 8TH ANNUAL
INDEPENDENT
GAMES FESTIVAL

MARCH 20-24, 2006
SAN JOSE, CA
WWW.IGF.COM

The stage is being set for the eighth annual
Independent Games Festival—the most respected event
celebrating the indie game developer. Here's your chance to see
if you have what it takes to bring home an award in 2006!
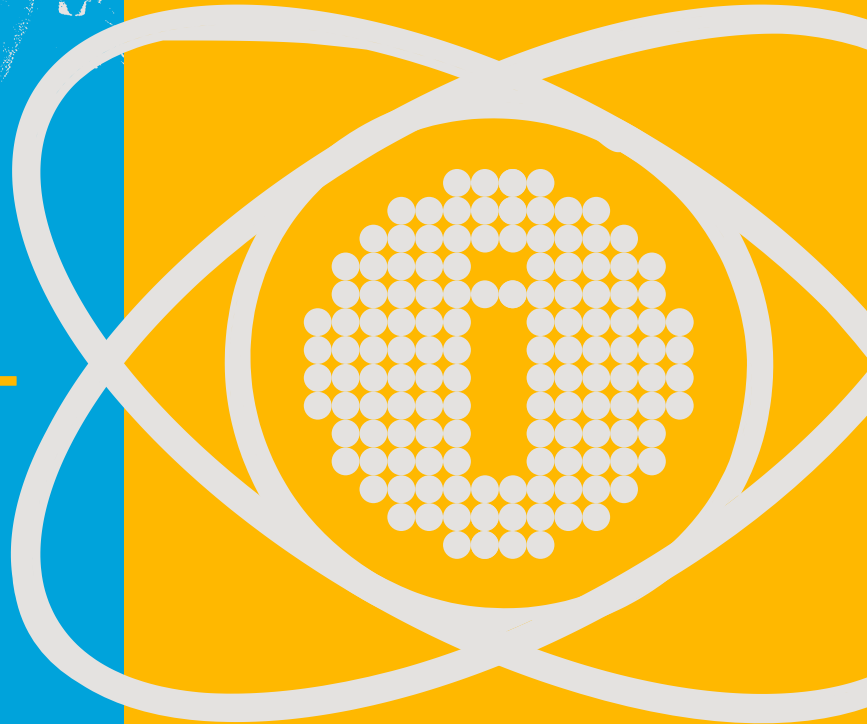
OPEN CALL FOR ENTRIES

IGF COMPETITION
SUBMISSIONS CLOSED
Open to all independent game development
teams, the IGF Competition offers over $35,000
in cash and prizes.

IGF MODDING COMPETITION
DEADLINE FOR SUBMISSIONS: OCTOBER, 10 2005
New Competition! Non-commercial mod teams
compete in one of four categories for a chance
to take home a $2500 cash prize.

IGF STUDENT COMPETITION
DEADLINE FOR SUBMISSIONS: NOVEMBER 15, 2005
A competition for games developed
by full-time students.

**FOR INFORMATION ON CATEGORY DESCRIPTIONS,
ENTRY FORMS, AND ELIGIBILITY RULES GO TO**

WWW.IGF.COM

CMP
United Business Media

GAME ON

NOW HIRING PEOPLE WITH TECHNICAL AND ARTISTIC VISION

rainbowstudios.com

relic.com

volition-inc.com

crankypantsgames.com

heavy-iron.com

helixe.com

locomotivegames.com

concretegames.com

thq.com

bluetongue.com

R rainbow STUDIOS

relic ENTERTAINMENT

volition inc

crankypants games
A THQ COMPANY

HEAVY IRON STUDIOS

helixe

locomotive games

concrete games

THQ studio australia

bluetongue

THQ

THQ Studios... there's a place for you.

Rockstar Games is actively seeking experts in the following fields:

Rockstar North
Tools Programmer
Environment Artists
Character Artists
Animators
Vehicle Artists
Level Designers

Rockstar San Diego
Tools Programmer
Animation Programmer
Graphics Programmer
Networking Programmer
Concept Artist

Rockstar Vienna
AI Programmer

Rockstar Vancouver
Art Director

Rockstar NYC
Motion Capture Specialists

Rockstar Toronto
AI Programmer

Rockstar Leeds
Audio Programmer

Rockstar Games
622 Broadway, New York, NY 10012
Att: Recruiting
Email: resumes@rockstargames.com
Fax: 212-334-6644
www.rockstargames.com
Please indicate the studio and position you are applying for in the subject heading.

MIDWAY

Join the Pulse of Gaming™

JOBS.MIDWAY.COM

Midway Games Inc. is a leading developer and publisher of interactive entertainment software for all major video game systems, releasing some of the most popular games in video game history.

Our goal is to create the most compelling, high-quality games around. We're always searching for world-class talent with energy, commitment to excellence and unrestricted imaginations to join our exceptional teams.

MUNICH

SEATTLE

LONDON

LOS ANGELES

CHICAGO

SAN DIEGO

DALLAS

AUSTIN

ADELAIDE

MONTREAL, CANADA:
Producer

SEATTLE:
Producer
Lead Technical
 Designer
Lead Scenario
 Designer
Lead Writer
Lead Animator
Lead Gameplay
 Programmer
Technical Producer

LOS ANGELES:
Creative Director
Project Tech Lead
Project Art Lead
QA Supervisor
Animator
Programmers

SAN DIEGO:
Development Director
Producer –
 Children's Titles
QA Supervisor
QA Lead

AUSTIN:
Sr. Software Engineer
Software Engineer
Producer
Ai Programmer
Senior Designer
Lead Modeler
Concept Artist
Associate Producer
Producer
Network Programmer

CHICAGO:
Gameplay
 Programmer
Ai Programmer
Mid-level
 Programmer
Software Engineer
Sr. Software Engineer
Technical Director
Sr. Game Designer
Game Designer
UI Artist
Lead Animator
Special Effects Artist
Environment Artist
Lead Environment
 Artist
Environment Concept
 Artist

**ADELAIDE,
AUSTRALIA:**
Junior Artist
Senior Artist
Sr. Character
 Animator
Jr. Character
 Animator
Level Designer
Programmer
Studio Creative
Director
Studio Art Director

TAKE GAME DEVELOPMENT
TO THE NEXT LEVEL.

Microsoft®

XNA™

microsoft.com/xna

Make better games faster. XNA is the newest game platform team at Microsoft, and here, your passion can revolutionize the way next-generation games are made. By creating industry-defining cross-platform software, tools and service components, you'll empower game creators to manage costs and focus their talent where it matters most – on design, creativity and bringing brilliant new game worlds to life.

XNA is now seeking visionary, dynamic people to join our team. To apply online, visit microsoft.com/careers, or email your resume to xboxjobs@microsoft.com.

XNA Now Hiring: Program Manager/Technical Producer *Required:* Experience in game development, development tools and/or Windows/Xbox development. *Preferred:* Interpersonal skills, strong technical background, participation in at least one game product cycle and the ability to translate customer requirements into viable solutions. Lead Software Engineer *Required:* Coding background in C# or C/C ++. *Preferred:* Experience in game development, development tools and/or Windows/Xbox development. Software Engineers *Preferred:* Visual Studio experience. C++ and C# with an emphasis on C# and the .NET framework, UI design and support infrastructure, 3D graphics experience and experience extending one or more of DCC packages like MAX or Maya. Test Focused Software Engineer *Required:* Testing and/or programming experience. *Preferred:* Coding background using C# or C/C++ in developing product or test automation.

XBOX 360™

Student artwork…Sungjoo Yong
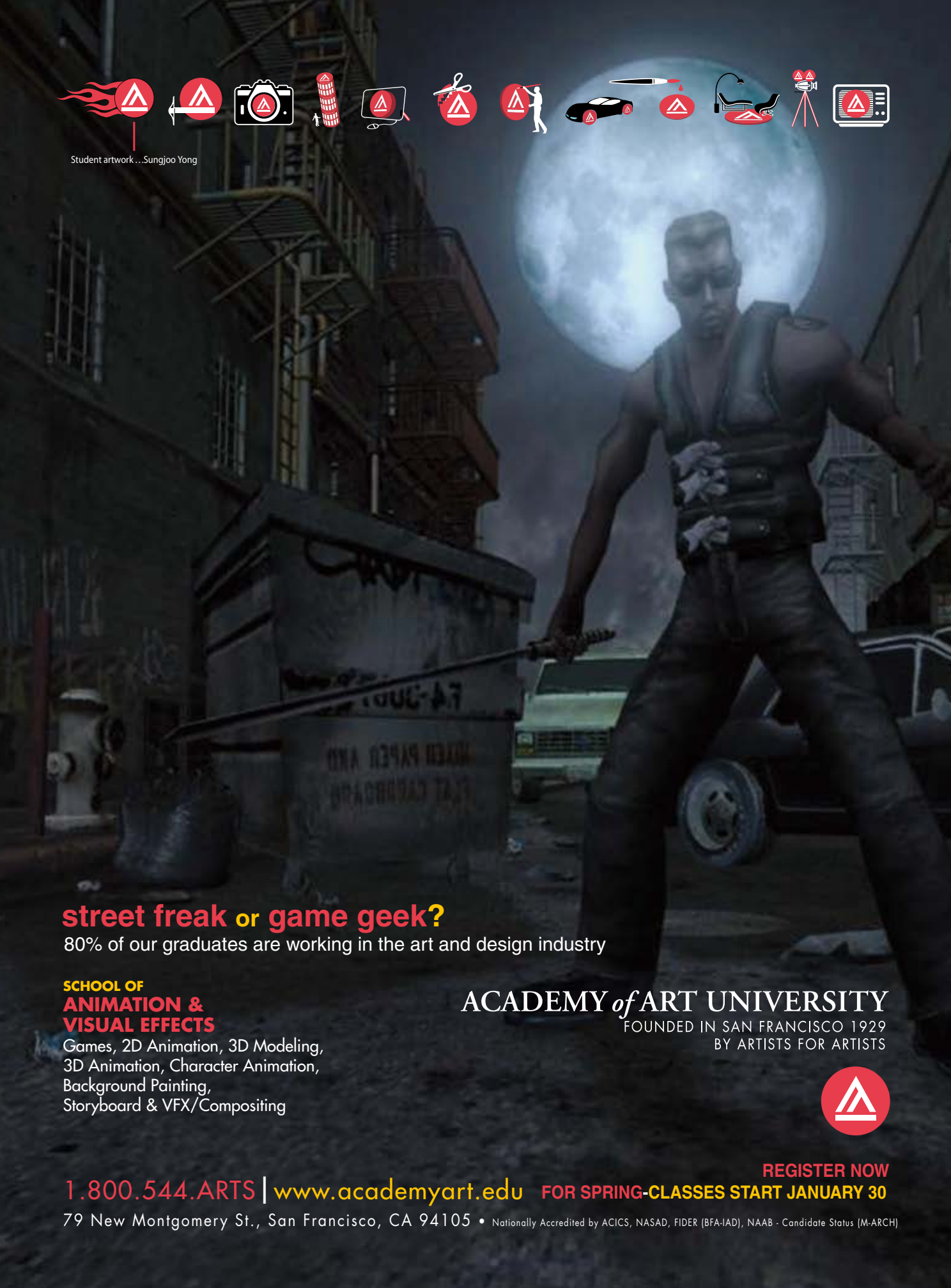
street freak or game geek?

80% of our graduates are working in the art and design industry

SCHOOL OF
ANIMATION &
VISUAL EFFECTS

Games, 2D Animation, 3D Modeling,
3D Animation, Character Animation,
Background Painting,
Storyboard & VFX/Compositing

ACADEMY *of* ART UNIVERSITY

FOUNDED IN SAN FRANCISCO 1929
BY ARTISTS FOR ARTISTS

REGISTER NOW

1.800.544.ARTS | www.academyart.edu   **FOR SPRING-CLASSES START JANUARY 30**

79 New Montgomery St., San Francisco, CA 94105 • Nationally Accredited by ACICS, NASAD, FIDER (BFA-IAD), NAAB - Candidate Status (M-ARCH)

ADVERTISER INDEX

⟫ MARKETPLACE

FLY THROUGH YOUR CREATION

FREEDOM   SPEED   COMFORT

MEET SPACEPILOT™

A bird can't fly with one wing. Why use just a mouse when you can put both hands into your creation. SpacePilot™, the Intelligent Motion Controller, is a giant leap forward in intelligent interface devices for professionals. Featuring adaptive sensing technology that automatically gives you the functions you want, when you want them and an array of extendable keys that put unlimited functions within a fingertip's reach.

Master your 3D world today – learn more at: www.3dconnexion.com/GDfly

3Dconnexion
A LOGITECH COMPANY

CMP
United Business Media

» THE LEADING GAME INDUSTRY MAGAZINE

gamedeveloper
DIGITAL EDITION

» THE GAME INDUSTRY'S LEADING SOURCE OF INFORMATION FOR DEVELOPING NEXT GENERATION GAMES

SAVE 60% OFF THE PRINT SUBSCRIPTION RATE

A THOUSAND WORDS

UBISOFT'S

PETER JACKSON'S
KING KONG

Art by Ubisoft Montpellier art team

[Other competitive solutions require multiple supporting packages to handle our assets. With **Alienbrain**, everything we need is in one package.]

Christopher Bretz Art Director, Secret Level

Brainpower

Avid Alienbrain Studio 7.1 has the intelligence to help your team work faster and smarter.

Its intuitive interface tells you more than the what, where and who of an asset. It tells you what an asset references, what it's referenced by, and whether it's buildable, it's optimized, or it's redundant. It knows what platform an asset is for and what should happen to it next. It even knows an asset's location in a texture map, its resolution in level three, and how much damage it inflicts when eaten. Alienbrain knows. You know. Your pipeline knows.

ASSET MANAGEMENT FOR CREATIVE TEAMS

Free eval copies available at www.alienbrain.com

alienbrain STUDIO

www.alienbrain.com

Avid computer graphics