



United Business Media

INTERVIEW: NIGHTS CREATOR NAOTO OHSHIMA

VOL16NO10NOVEMBER2009

gamedeveloper

THE LEADING GAME INDUSTRY MAGAZINE



**DEVELOPERS!
DEVELOPERS!
DEVELOPERS!**

**50+ IMPORTANT GAME
INDUSTRY PEOPLE**

**GAMING ON
THE WIRE**

**THE PROMISE OF CLOUD
COMPUTING ANALYZED**

POSTMORTEM

scribblenauts

Reliable Pathfinding Technology. Havok AI.



havok[™]
www.havok.com
*Turning Creative Aspirations
into Technical Realities[™]*



POSTMORTEM

22 5TH CELL'S SCRIBBLENAUTS

SCRIBBLENAUTS is a pie-in-the-sky concept come to life, and there's a lot for developer 5th Cell to be proud of. Plus, in a game where the object is to write anything and have it come to life on the screen, what could go wrong? A whole lot, of course—from QA to localization to controls, all facts are laid bare.

By Joseph M. Tringali

FEATURES

7 GAMING IN THE CLOUD

Companies like OnLive and Gaikai are promising the gaming world on a platter with cloud computing-supported game services. But how viable is it, really? Programmer Jake Cannell takes a look at the technical and financial logistics behind this emerging space.

By Jake Cannell

13 THE GAME DEVELOPER 50

In concert with our advisory board, we've highlighted 50 important achievements of the last year-or-so, and associated them with specific persons in order to acknowledge individual work. It's no easy feat to distill the work of many into a few names, but we attempted it here.

By Brandon Sheffield, Jeffrey Fleming, and Simon Carless

50 INTERVIEW: NAOTO OHSHIMA

Ohshima is best known as the character designer for SONIC THE HEDGEHOG and director of NIGHTS INTO DREAMS, and here we investigate his motivations, thoughts, and his fascination with time.

By Brandon Sheffield

CONTENTS.1109

VOLUME 16 NUMBER 10

DEPARTMENTS

- | | | |
|----|---|---------------|
| 2 | GAME PLAN By Brandon Sheffield To Be Continued | [EDITORIAL] |
| 4 | HEADS UP DISPLAY Good Old Designs, Sense of Wonder Night 2009, and more. | [NEWS] |
| 34 | TOOL BOX By Zoltan Erdokovy and Bijan Forutanpour Luxology's Modo 401, and Real Time Cameras book | [REVIEW] |
| 38 | THE INNER PRODUCT By Mick West Jump To It | [PROGRAMMING] |
| 42 | PIXEL PUSHER By Steve Theodore Most Likely to Succeed | [ART] |
| 46 | DESIGN OF THE TIMES By Damion Schubert The Art of Fun | [DESIGN] |
| 49 | AURAL FIXATION By Jesse Harlin The Magic of Misdirection | [SOUND] |
| 56 | ARRESTED DEVELOPMENT By Matthew Wasteland Stack Trace Industry Detective | [HUMOR] |



TO BE CONTINUED

RETHINKING PLAYER DEATH

ARCADES HAVE BROUGHT US A LOT OF SIGNIFICANT advances over the years. From the industry's beginning through the mid-90s, arcades were still where you'd find the best game graphics, and the best multiplayer experiences. A great many excellent design rules and guidelines were forged in these fires. Every so often though, I notice a trope carried over from the arcade days that just doesn't seem to fit anymore. One of those is the concept of lives, continues, and player death.

The thought occurred to me while playing *TEENAGE MUTANT NINJA TURTLES: TURTLES IN TIME RE-SHELLED* for Xbox Live Arcade. I played with three friends, and aside from none of us being able to tell which character we were about 75 percent of the time, I noticed that the game defaulted to unlimited continues. Playing essentially amounted to mashing the attack button and hoping for the best, with no real consequences to death, though the game did keep a tally of who among my friends had died the most times.

It really stuck with me—in a scenario in which death essentially means nothing, why have death at all? Granted, this was a port of an arcade game, but a number of kids' games operate under a similar basic principle. Dying either places you right back where you were, or it does so until you run out of lives, and then you continue and start at the beginning of the level, lives fully restocked. The game is basically testing your ability to complete the same actions again and again, rather than your skill. Except in outlying cases, it's testing your willingness to persevere, and not to adapt.

RISE FROM YOUR GRAVE

» This extends in a mild way to the checkpoint systems in modern games. Most AAA games have rid themselves of the idea of continues, or even the concept of limited lives, but death is still not so much a punishment as it is a setback—you simply lose a few minutes' playing time, and probably learn some strategies in the meantime. So why represent this as "death," rather than in some other way? It could well be because we've always done it that way, rather than for any reason anyone spent time thinking about.

DEMON'S SOULS is going to be a hot topic discussion among alternative journalists and academics for some time, perhaps rightly so—the way that game deals with death is well thought out, and actually has an in-world reason behind it. If you die, your (weaker) soul must go out in search of demon souls with which to reclaim your physical body. In *ASSASSIN'S CREED*, "death" is explained as a de-syncing of the player from his host body in history. In *PREY*, players must fight their way back from the valley of death, to reclaim their place among the living. Even *Silicon Knights'* drawn-out resurrection sequences in *TOD HUMAN*

are conceptually relevant. So many games employ outlandish sci-fi or fantasy scenarios that it seems death could be explained away in simple terms—or even better, with some entertaining gameplay.

Continues and their progeny are usually not a nuisance, they just seem unnecessary—evidence of the early framework around which games have evolved. When a developer intends it, player deaths can be entertaining, they just need to be given weight. If you don't care if you die, the stakes of play seem rather minor. In *LEFT 4 DEAD*, you feel like you're letting your teammates down, so death has weight. But in *GEARS OF WAR*, which I otherwise like very much, death is just an annoying setback, and I have to watch cut scenes again and traipse back across the same shattered landscape just to get killed again by the same stupid Troika.

We've had articles in the magazine (see "Press A to Jump," *Game Developer*, October 2009) that cite death as a good time to provide positive feedback for players, and add tutorial hints. Considering the fact that games are meant to be won, the player is essentially invincible in the grand scheme of things, requiring, again, nothing more than perseverance and the occasional GameFaqs hint. Why not try to create a game that contains similar challenges to standard games, but completely avoids death? If the end result is the same, and meaningful (or humorous) consequences aren't built in, why is death even part of the equation? It isn't really death, after all, it's really just the proverbial "flesh wound."

These days we have a buffer against death in games anyway—most first- and third-person shooters have regenerative health, whether it makes sense for the world or not. Once the player's regenerative health is depleted, they crumple to the ground momentarily, only to have time magically rewind, with players finding themselves transported to a very familiar (usually identically instanced) checkpoint save state from their recent past.

THE DEATH (OF MY INTEREST)

» Of course if you're not careful, alternate methods of "death" can be even more annoying. Lengthy resurrection sequences (in fact cut scenes of any kind) will take the player out of the action. Death- or loss-related minigames need to be fun and relevant. Almost any time not playing the game is time waiting to play the game—and that's generally time wasted.

Innovative ways of dealing with player death aren't just for the wacky indies, or the fringe titles—which is why I deliberately didn't mention any. High-end games are starting to change the way they deal with what lies beyond, and it seems high time that more developers start to confront their own (in-game) mortality.

—Brandon Sheffield

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

SUBSCRIPTION SERVICES

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

EDITORIAL

PUBLISHER
Simon Carless | scarless@gdmag.com
EDITOR-IN-CHIEF
Brandon Sheffield | bsheffield@gdmag.com
PRODUCTION EDITOR
Jeffrey Fleming | jffleming@gdmag.com
ART DIRECTOR
Joseph Mitch | jmitch@gdmag.com
CONTRIBUTING EDITORS
Jesse Harlin | jharlin@gdmag.com
Steve Theodore | stheodore@gdmag.com
Mick West | mwest@gdmag.com
Soren Johnson | sjohnson@gdmag.com
Damion Schubert | dschubert@gdmag.com

ADVISORY BOARD

Hal Barwood Designer-at-Large
Mick West Independent
Brad Bulkley Neversoft
Clinton Keith Independent
Bijan Forutanpour Sony Online Entertainment
Mark DeLoura Independent
Carey Chico Pandemic Studios

ADVERTISING SALES

GLOBAL SALES DIRECTOR
Aaron Murawski | amurawski@think-services.com
t: 415.947.6227

MEDIA ACCOUNT MANAGER

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

GLOBAL ACCOUNT MANAGER, EDUCATION AND RECRUITMENT

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

COORDINATOR, EDUCATION AND RECRUITMENT

Rafael Vallin | rvallin@think-services.com
t: 415.947.6223

ADVERTISING PRODUCTION

PRODUCTION MANAGER
Robert Steigleider | rsteigleider@ubm-us.com

REPRINTS

WRIGHT'S REPRINTS
Ryan Pratt | rpratt@wrightsreprints.com
t: 877.652.5295

THINK SERVICES

CEO THINK SERVICES Philip Chapnick
GROUP DIRECTOR Kathy Schoback
CREATIVE DIRECTOR Cliff Scorsio
CHIEF INFORMATION OFFICER Anthony Adams

AUDIENCE DEVELOPMENT

GROUP DIRECTOR Kathy Henry
e: khenry@techinsights.com
LIST RENTAL Merit Direct LLC | t: 914.368.1000

MARKETING

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

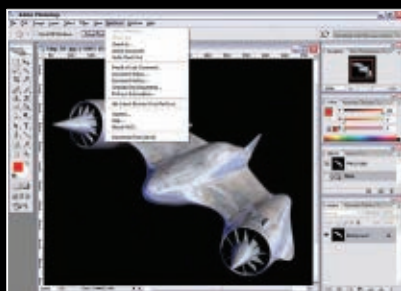
UBM TECHNOLOGY MANAGEMENT

CHIEF EXECUTIVE OFFICER David Levin
CHIEF OPERATING OFFICER Scott Mozarsky
CHIEF FINANCIAL OFFICER David Wein
CORPORATE SENIOR VP SALES Anne Marie Miller
SENIOR VP, STRATEGIC DEV. AND BUSINESS ADMIN. Pat Nohilly
SENIOR VP, MANUFACTURING Marie Myers





Introducing P4GT, a productivity feature of Perforce SCM.



P4GT

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

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

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

PERFORCE
SOFTWARE

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



sense of wonder 2009

At the 2009 Tokyo Game Show, CESA held the second Sense Of Wonder Night, showcasing 10 presentations from creators of interesting, innovative experimental games. This year saw 65 worldwide submissions, with ten games picked for creating a "sense of wonder" by a judging committee that included noted shooter creator Kenta Cho, Enterbrain's MAKER series producer Kenji Sugiuchi, KATAMARI DAMACY creator Keita Takahashi, Vector's Takashi Katayama, IGDA Japan's Kiyoshi Shin, and the writer of this article (IGF Chairman and *Game Developer* publisher Simon Carless).

BALL CARRY (TAKUYA ONO / JAPAN)

BALL CARRY from Takuya Ono was an experimental title made in a week, where you roll a ball toward a goal by pressing multiple keys on the keyboard to make depressions in the ground. To move the ball from the left to the right, you press the A button, then the S button, then the D button. Ono wanted to make an augmented reality game, but it was "very troublesome" to use such complex peripherals and instead settled on the keyboard, saying, "It's not necessary to come up with new devices to create a new experience."

HAZARD: THE JOURNEY OF LIFE (ALEXANDER BRUCE / AUSTRALIA)

Alex Bruce presented HAZARD: THE JOURNEY OF LIFE, an abstract UNREAL TOURNAMENT 3 mod about "philosophy, player expectations"... and learning through experiences in which the player makes choices that result in a moral or life lesson. For example, the player can try to jump over a pit that is not initially make-able, despite an exhortation to "jump." The player must fall down to learn how to get up and keep playing.

SHADOW PHYSICS (ENEMY AIRSHIP / UNITED STATES)

Steve Swink and Scott Anderson of Enemy Airship then presented SHADOW PHYSICS, which stars a character that only exists in a shadow world, and must clamber over shadows of real objects to reach level completion. The duo showed that when you push

on the shadow of an object, the physical object is moved. They noted, if you "use shadows for physical objects... when you change the light, you change your entire world."

INCOMPATIBLE BLOCK (JUN FUJIKI / JAPAN)

Jun Fujiki, creator of the PSP and PS3 downloadable title ECHOCHROME, has produced a number of interesting prototypes using similarly Escher-styled concepts, and showcased one in particular—INCOMPATIBLE BLOCK. In it, the user can drag blocs around the screen from a certain perspective and stack them.

YOU ONLY LIVE ONCE (MARCUS RICHERT / SWEDEN)

Marcus Richert, a Swedish journalism student showed YOU ONLY LIVE ONCE, a cheeky conceptual title that is available in 25 languages. It features a single level, certain failure, but a massive amount of game over screens that tell a story of the player after death. Keita Takahashi said that he liked the "corniness" of the game.

HIS AND HER DISCONNECTED CONVERSATIONS (HIMO / JAPAN)

The Kyoto-based developer behind HIS AND HER DISCONNECTED CONVERSATIONS, Himoto, showed an interesting Japanese-language title in which couples are discussing things with each other in text boxes, and you have one minute to decide which couple matches another by reading what they are saying to each other. The creator noted: "The



PHOTO BY JERASKA

player goes through emotional experiences by going through these multiple stories at one time." In addition, you can edit the conversations in the game via a webpage—a unique version of user-generated content.

ECOLPIT (MISI / JAPAN)

Misi showed ECOLPIT, a top-down 2D title with bug-like characters eating food and fighting with each other. You can either defeat or outgrow all the other characters to complete each level. His "truly foolish" Al nonetheless has a massive memory, recalling when other bugs shot at them, and retaining that hostility for a long period of time. They will also develop hostility toward characters that attack their friends, and camaraderie is also built up over time by each of these odd-looking bugs.

SWARM RACER 3000 (JOSEPH WHITE AND LEXALOFFLE GAMES / JAPAN)

Joseph White of veteran indie

outfit Lexaloffle showed SWARM RACER 3000, his alternative top down racing game. In it, you control an entire swarm of characters, and you can control the size of the swarm while you try to pick up gems. He noted: "The entire game is about learning to control the size and movement of your swarm."

PARA RAIL (KUNIAKI WATANABE AND ONITAMA - ZENER WORKS INC. AND TEAM ONIKU / JAPAN)

The creators at Zener Works and Team OniKu presented PARA RAIL, a vector-style ASTEROIDS-looking spacecraft that is not directly controlled. In fact, all you can do is increase or decrease the amount of games being played simultaneously by deleting any game that is about to end in Game Over. The makers explained: "You don't actually manipulate and play the game, yet still get the feeling of enjoyment." They suggested some future applications to this experiment, arguing that parallel universe

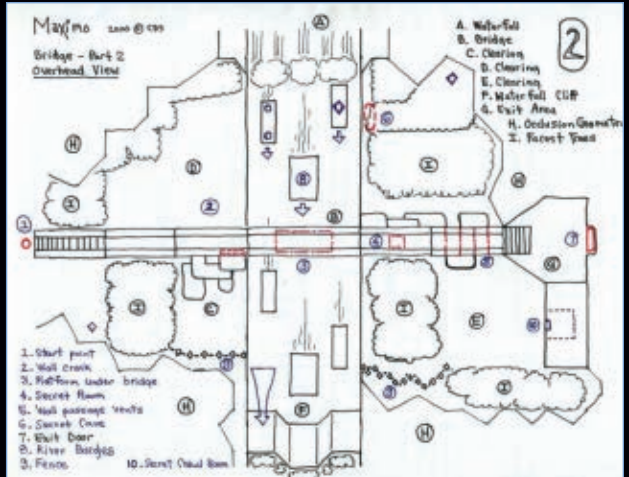
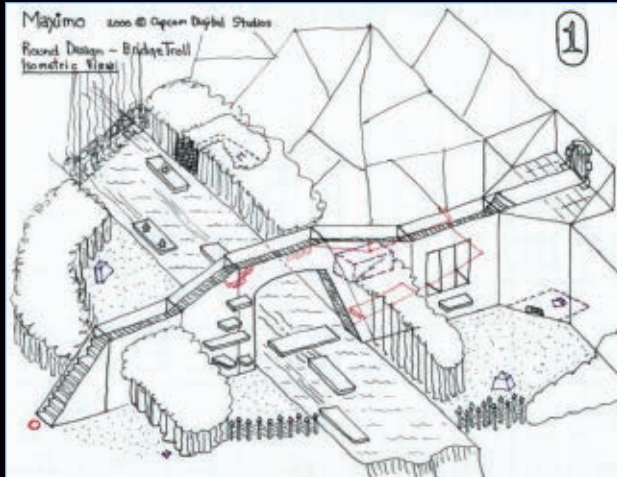
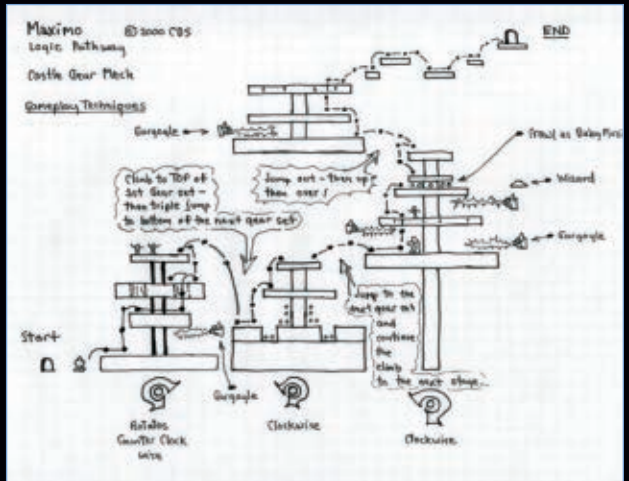
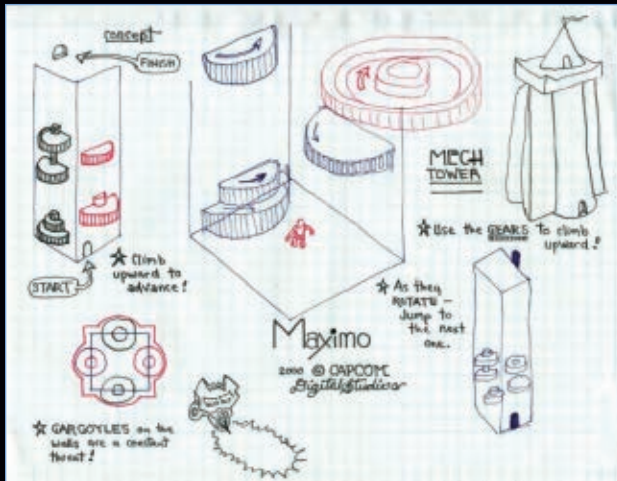
versions of RPG titles and other games could be an avenue of exploration and that multiple players could cooperate across the network to collectively create optimum "replays."

TRANSCEND (FISHBEAT / UNITED STATES)

Fishbeat's Zach Aikman showed off TRANSCEND, an evolution of the IGF Student game-winning SYNAESTHETE, which is currently in development for Xbox Live Arcade. In the title, you have to avoid enemies and attack them by tapping in time to the beat, creating a music action game. The abstract title included circles of trees attacking and giant dancing mushroom bosses in a lush visual atmosphere, with a bright trance-like electronica soundtrack.

KATAMARI creator Keita Takahashi said pointedly that the titles on show at the second Sense Of Wonder Night were "games that the people here were not forced to create, but games that they really wanted to create."

—Simon Carless



Good Old Designs Spotlight on David Siller

DAVID SILLER IS THE CREATOR OF AERO THE ACROBAT, AND WORKED WITH SUNSOFT'S U.S. OFFICE FOR MANY YEARS BEFORE CONTRIBUTING DESIGN TO CRASH BANDICOOT (through publisher Universal), and then MAXIMO and other titles with Capcom. In those days, he was part of the Japanese/Western collaboration experiments, analogous to STI and Sonic Team's work on SONIC 2. Recently, he's been showing some of his old designs via his Facebook page—we've been trying to convince him to keep a repository of this stuff, but for now, we'll highlight a few pieces of his work on MAXIMO, still done the old fashioned way, with pen and paper.

—Brandon Sheffield



IGDA Leadership Forum

INDEPENDENT GAME CONFERENCE WEST 09

Marriott Marina Del Rey, Marina Del Rey, CA
November 5–6
Price: \$119–\$325 www.icgwest.com

PROJECT HORSESHOE

Canyon of the Eagles, Lake Buchanan, TX
November 5–8
Price: \$1,995–\$2,195 www.projecthorseshoe.com

november calendar

IGDA LEADERSHIP FORUM

San Francisco Airport Marriott, Burlingame, CA
November 12–13
Price: \$400–\$1,750 www.igda.org/leadership

NETGAMES 2009

University of Paris, Paris, France
November 23–25
Price: EUR175–EUR375 <http://netgames2009.lip6.fr>

**KILLER
TRACKS™**



**NEW MUSIC
SAMPLER**

Call 800.454.5537
to request a
complimentary
copy of the
Killer Tracks
video game
music sampler.



KILLER

**Video Game
Music**

Perfect For:

- Action
- Fantasy Adventure
- Shooter
- Strategy / Military
- Sports
- Music Games
- Trailers & Cinematics

ASK ABOUT OUR NEW GAME TRAILER CUES FROM AAA GAME
COMPOSERS BILL BROWN, LENNIE MOORE & ALEX KHARLAMOV!

Web: www.killertracks.com • Email: sales@killertracks.com • Phone: 1.800.4.KILLER

gaming in the cloud

CLOUD COMPUTING HAS COME INTO ITS OWN IN RECENT YEARS. UNLIKE MOST BUZZWORD CONCEPTS, IT REALLY does represent a true shift in the computer industry, from the devices and services users purchase to how they are designed and operated. With the rise in sales dominance of cheap laptops, the announcement of Google's plans to turn Chrome into a web operating system, and the general shift to software as a service, it should come as no surprise that some see the game industry heading in the same direction.

But streaming a fully interactive video game in real time is a very complex problem, far more so than typical web-based applications, and surely still far off into a future world featuring a faster, more reliable internet. Or is it? Just this year at GDC, newly-unveiled startup OnLive announced that it will be bringing that future into the present this year, with its fully cloud-based microconsole system, which immediately brought a competitor out of hiding in the form of David Perry's new venture, Gaikai. These two aren't alone: start-up Otoy is partnering with AMD to create large render farms that can be used for gaming, and several other start-ups are working on streaming games over existing cable set top boxes.



If OnLive or one of its competitors overcomes the myriad technical and economic hurdles to launch a viable cloud gaming system, the impact on the industry over the years ahead could be transformative. It could be the end of the console cycle. Almost everyone has something to gain out of this change. Consumers gain the freedom and luxury of instant, on-demand access to ultimately all of the world's games, and the ability to try before they buy or rent. Publishers get to cut out the retailer middle-man, and avoid the banes of piracy and used game resales.

But the biggest benefit ultimately will be for developers and consumers, in terms of the eventual game development cost reduction and quality increase enabled by the technological leap inherent to cloud computing. Finally developing for one common, relatively open platform (server-side PC) will significantly reduce the complexity in developing a AAA title. But going further into the future, once we actually start developing game engines specifically for the cloud, we enter a whole new technological era.

Its mind-boggling for me to think of what can be done with a massive server farm consisting of thousands or even tens of thousands of densely-networked GPUs with shared massive RAID storage. Engines developed for this system will look far beyond anything on the market today and will easily support massively multiplayer networking, without any of the usual constraints in physics or simulation complexity. Game development costs could be cut in half, and the quality bar for the big budget AAA titles could close the gap with feature films.

But can it work? And if so, how well and when? Skeptics such as Richard Leadbetter (see resources) doubt that the fundamental obstacles of Internet latency, limited bandwidth, and real-time video compression are close to solvable. And even if they are, the cost of all the server hardware and bandwidth could be astronomical. Some have also doubted the true value added for the end user: even if it can work technically and economically, how many gamers really want this?

LATENCY

» The Internet is far from a guaranteed delivery system, and at first the idea of sending players' inputs across the Internet, computing a frame on a server, and sending it back across the Internet to the user sounds fantastical. But to assess how feasible this is, we first have to look at the concept of delay from a neurological perspective. A user presses the fire button on a controller, and some amount of time later, the proper audio-visual response is presented in the form of a gunshot. If the firing event and the response event occur close enough in time, the brain processes them as a simultaneous event, like two expanding and reverberating ripples of thought that merge together. Beyond some threshold, the two events desynchronize and are processed distinctly: the user notices the delay. A large amount of research on this subject has

determined that the measurable delay threshold is around 100–150ms. It's obviously a fuzzy number, but as a rule of thumb, a delay of under 120ms is typically unnoticeable to humans. The brain has a massive number of neurons and connections (billions and trillions respectively), but signals propagate across the brain very slowly compared to the speed of light. For reference I highly recommend *Consciousness Explained* by Daniel C Dennet. Here are some interesting timescale factoids from that book:

- » saying, "one, Mississippi" 1000msec
- » unmyelinated nerve fiber, fingertip to brain 500msec
- » speaking a syllable 200msec
- » starting and stopping a stopwatch 175msec
- » a frame of television (30fps) 33msec
- » fast (myelinated) nerve fiber, fingertip to brain 20msec
- » basic cycle time of a neuron 10msec
- » basic cycle time of a CPU(2009) .000001msec

In the time it takes the "press fire" signal to travel from the brain down to the finger muscle, Internet packets can travel roughly 4,000 km through fiber (light moves about 200,000 km/s through fiber). This is about the distance from Los Angeles to New York.

Each component of a game system (input connection, processing, output display connection) adds a certain amount of delay, and the total delay must add up to around 120ms or less for good gameplay. Up to 150ms is sometimes acceptable, and beyond 200ms the user experience rapidly breaks down as every action has noticeable delay.

But how much delay do current games have? Gamasutra has an interesting article about this (see Resources). There, Mick West measures the actual delay of real world games using a high speed digital camera. He finds a "raw response time for GTA IV of 166 ms (200 ms on flat panel TVs)." This is relatively high, beyond the acceptable range, and GTA has received some criticism for sluggish response. And yet this is the grand blockbuster of video games, so it certainly shows that some games can get away with 150–200ms responses and the users simply won't notice or care. Keep in mind this delay time isn't when playing the game over OnLive or anything of that sort: this is just the natural delay for that game with a typical home setup.

In the modern era, a typical network ping to a server is in the range of 30 to 60 ms. So now you can imagine that lengthening the input and video connections out across the Internet is not as ridiculous as it first appears. It adds additional delay, which you simply need to compensate for somewhere else.

How would a server-based rendering system compensate for this delay? The idea for existing games is simple: You can just run the game at a much higher FPS. Controllers and displays add some latency, but the bulk of the delay comes from the game engines themselves, which typically internally buffer 3–4 frames of data for pipelining between the numerous processors of modern systems. Running at 30fps, every full frame buffered costs a whopping 30ms of delay. If the PC version of a console game runs at 60 FPS, and it still keeps 4 frames of buffering, the internal delay is reduced from 120ms to 60ms. If you combine that with a low-latency controller and TV, suddenly it becomes realistic to play a game in Los Angeles from a server residing in San Jose. In fact, with a fiber connection such as FIOS and good home equipment, you could conceivably play from almost anywhere in the U.S., in theory. The key is that many console games have already maxed out the maximum delay (when running on the console), and modern GPUs are many times faster. In the future, engines designed to support server side rendering can further reduce the delay by reducing or eliminating internal buffering.

VIDEO COMPRESSION AND BANDWIDTH

» Clearly latency alone is not a final obstacle to cloud gaming. A server can not send a raw, uncompressed frame buffer directly to the user (at least, not at any acceptable resolution on today's broadband). For practical streaming video, compressors need to squeeze those frame buffers down to acceptably tiny sizes, and more importantly, they need to do this rapidly or near instantly so as not to introduce additional delay. Is this possible? What is the state of the art in video compression?

All modern video compressors, such as H.264, are based on two key techniques: intra [within] frame compression using whole image techniques such as JPEG, and inter [between] frame compression using motion compensation, which compresses image chunks similar to previous chunks by encoding a motion offset and a residual difference. There are upcoming alternatives such as wavelets that may eventually give H.264 a run for its money, but they all work on the same general principle of combining state-of-the-art image compression with motion compensation, getting 10–30 times the compression rate of image compression on its own.

Because of their use of motion compensation, video compression systems have a large variation in their instantaneous bitrate. Good H.264 encoders can achieve reasonable quality for 720p at around a 2 Mbps average bitrate, and even around 1 Mbps is passable. However, the instantaneous or maximum bitrate can spike up to 10 times this, which happens on scene cuts or rapid camera motion when full frames must be sent.

Using JPEG for comparison, about 1 bit per pixel is a reasonable benchmark for whole frame compression. At this rate a 720p frame would take up a little under a megabit. Even if compression took no time, with a 5 Mbps connection, that problem frame would take 200ms to just send across the pipe, followed by dozens of much smaller frames as motion compensation kicks in. That's not a

Ultimately, the real opportunity of a cloud computing service lies in exclusive titles which take advantage of the unique networked server hardware to create a beyond next gen experience.

problem for a web video player, such as flash, because it can buffer up dozens or even hundreds of frames and then leisurely play them back at a smooth framerate. But cloud gaming requires ultra low latency, so buffering is not an option. Fortunately, the difficult frames are exactly the frames that will experience the most motion blur due to camera movement, so reducing the resolution or quality some in these cases and allowing it to smoothly blend in over several frames is a reasonable trade-off to reduce latency and smooth out the framerate.

The final consideration for any cloud gaming service is the CPU cost to compress all this video, as each user logged into the system needs their own live feed. The highest quality H.264 encoders, such as the x264 software encoder, can compress in real-time on a modern Intel quadcore system, using all 4 processors. Even faster H.264 encoders are now possible on the GPU, but the fastest and most cost effective compression would use custom hardware. Some low-latency hardware H.264 solutions already exist, such as the PureVu family of video processors

from Cavium Networks. I can't say that their compression quality at low bitrate is good enough for cloud gaming, or that they are cheap enough to deploy in massive scale, but it's interesting nonetheless.

What is clear is that cloud gaming demands video of high quality, low bitrate, near zero latency, and cheap encoding. A tall order, but clearly not impossible, given what's doable today with H.264. OnLive in particular has proclaimed a multi-patented revolution in zero latency video compression implemented on custom hardware, with a maximum bitrate of 5 Mbps for 720p, and 1.5 Mbps for 480p, and an average bitrate of about half of that. General consensus at the moment is that the quality is quite variable, usually good but noticeably blocky under heavy motion or poor Internet conditions. Impressive, but again in line with what's possible with H.264.

Not to be undone, David Perry recently posted a video and more details about Gaikai on his blog. He hints that his company is using flash for playback (which itself is H.264 based) and thus requires no plugin download. It's a little difficult to predict what the end quality will be like from watching a compressed video of Perry playing a compressed video stream, but from what I can gather the quality looks decent but the games are running in lower resolution, possibly 480p. He claims a bitrate of 1–2 Mbps. Perry can not resist taking a little stab at OnLive, saying "We don't claim to have 5,000 pages of patents, we didn't take 7 years, and we do not claim to have invented 1 millisecond encryption and custom chips. As you can see, we don't need them, and so our costs will be much less."

SERVER ECONOMICS

» As OnLive, Gaikai, Ottoy, and other cloud gaming services have now demonstrated, the dream is at least technically possible and the real question comes down to one of economics. How expensive are these data centers going to be, and how much revenue can they generate?

There are two key parameters that determine the cost scaling: the peak occupancy and the average occupancy. A full on-demand service such as OnLive must provide enough hardware and bandwidth for the peak occupancy. A paying customer who turns on his microconsole only to see a "sorry, try again soon" message is not likely to be paying much longer. It's difficult to get a good

estimate, but some data from the IGDA suggests that gamers average roughly 10 hours per week overall, and 20 hours per week for MMOs. This translates to about 5–10 percent average occupancy depending on the game. Looking at published peak concurrent users and subscriber numbers from EVE

ONLINE and WORLD OF WARCRAFT, I estimate 20 percent for peak occupancy. If the servers can be fully virtualized, this can help tremendously, as you could then scale framerate based on player load. In rare times of peak occupancy, providing a somewhat lower fps experience is probably better than providing no play experience.

The main costs are bandwidth, server hardware, and power. The first two scale with the peak occupancy, and the latter scales more to average occupancy.

To be viable, a cloud gaming service will have to buy bandwidth in bulk in what are known as transit arrangements. A little Google searching reveals that wholesale mass transit bandwidth can be found for around \$10 per Megabit/s per month in most US locations (comparable to end broadband customer cost, actually). Further searching suggests that in some places like LA it can be had for under \$5 per Mbs/month. This is for a dedicated connection or peak usage charge. Thus outgoing bandwidth roughly costs \$8–\$15 per connected user per month for 1.5 Mbps video, or \$25–\$50 per connected user per month for 5 Mbps video. These prices have been steadily falling year after year.

For server hardware, a cloud gaming service can potentially be more net efficient, as each connected user only needs a few CPUs, some RAM, and a GPU. Storage can be achieved with shared network RAID drives to reduce costs. If full GPU virtualization is achieved, one fast GPU could even render multiple game instances. Even ignoring that, at this point in time the basic components to run modern games at 60fps can easily be had for under \$360 on pricewatch, with a real high-end system costing about \$720. Spread over three years, that's just \$10–\$20 per connected user per month, cheaper than the cost of bandwidth. As explained above, I'm

EVEN IF
YOUR COMPANY
BUDGET EQUALS
YOUR CREDIT
CARD LIMIT,
your game can be Unreal.



No matter what size your budget. No matter what type of game.
Unreal can be your game engine. Email Mark Rein at getunreal@epicgames.com.

assuming the video compression is handled with cheap dedicated hardware, but even using extra CPUs to do it is not completely unrealistic.

This finally leaves us with power and cooling requirements. A gaming PC's typical full power load draws about 50 percent of the power supply's rated maximum, around 150W for a typical 300W gaming PC and 300W or so for a 600W maximum high end rig. However, based on the peak/average occupancy model, the servers are idle about 50 percent of the time (mainly at night and in the mornings), so the continuous power draw per server averages 75W–150W over the month. From what I've gathered, every watt of power requires about a watt of cooling, so we can double the base cost to get a net total of roughly 150W–300W per server. With power at about \$0.10 per kilowatt/hour, and 720 hours in a month, we get roughly \$10–\$20 a month per server in power and cooling costs.

In total this works out to a range of about \$50–\$100 per server per month, much of which will be bandwidth (clearly video compression is a key differentiator in the near term). Only if all the customers get HD and high end hardware is it closer to the latter number. Finally, we then apply the time-sharing, as each server is shared amongst a number of subscribers. Factoring in a peak occupancy of 20 percent (one server for every 5 subscribers) this works out to \$10–\$20 per month per subscriber, or \$5–\$10 per month for 10 percent peak occupancy. At even \$20 per month, cloud computing can be cheaper than running the equivalent hardware at home, because of the time sharing. The user's high end gaming rig sits idle most of the time.

THE GAME INDUSTRY POST-CLOUD

» Based on the rough estimates above, it would seem that cloud computing could be economically viable in terms of the per-user costs. However, to truly compete with Nintendo, Microsoft, and Sony, a cloud service would have to scale up very big. The size and quantity of the datarooms required to support even just the U.S. gaming population is quite staggering. We are talking about billions of dollars invested in perhaps millions of servers across a dozen different data center locations, drawing the combined power of an entire nuclear power plant. And that's just for the U.S. It would be like a fully subsidized console launch. However, we already have a very successful example of a company that has organically built up a massive distributed network of roughly 500,000 servers in over 40 data centers.

Yes, that company is Google.

To challenge the current consoles, the cloud upstarts will have to build an even bigger and more massive network. But to support such a massive investment will require a deep content library to attract customer revenue. And that presents something of a problem.

At the end of the day, the games themselves are all that really matters. And here is where OnLive's ambitions to compete directly with the consoles may run afoul with reality. The initial excitement of playing a game with only a dumb terminal quickly fades.

Most of OnLive's lineup consists of current generation console games, and even though OnLive will probably run them at nominally higher fps, this is mainly to compensate for latency. Video compression over the Internet will result in an end user experience that is no better, and often worse, than simply playing the console version. OnLive's one high end PC title—CRYSIS—is probably expensive for them to run, and will be seen as somewhat inferior to gamers who own high end rigs and have played the game locally. It will be more like the console version of CRYSIS. But

resources

RICHARD LEDBETTER

www.eurogamer.net/articles/gdc-why-onlive-cant-possibly-work-article

RESPONSIVENESS

www.gamasutra.com/view/feature/3725/measuring_responsiveness_in_video_.php?print=1

DAVE PERRY'S GAIKAI INTRO

www.dperry.com/archives/news/dp_blog/gaikai_video/

unfortunately for OnLive, Crytek's already working on that itself. I think Nintendo of America's president Reggie Aime said it best:

"Based on what I've seen so far, their opportunity may make a lot of sense for the PC game industry where piracy is an issue. But as far as the home console market goes, I'm not sure there is anything they have shown that solves a consumer need."

To be clear, cloud services will immediately solve some consumer needs. On-demand gaming, the ability to instantly start playing a game on any device, then decide to rent or purchase it later, will surely have an impact on the industry, even moreso than download services such as

Steam, Direct2Drive, and Impulse already have. Here Gaikai's entry strategy of focusing on playing PC games in the browser makes some sense. This may fit a more immediate (if less ambitious) real world need, as no one likes the hassle of downloading and installing a multi-gigabyte demo and its patches, just to play it a few times and then delete it.

But the true drivers of the game industry are the mega hit blockbusters which customers will go to great pains to play. So ultimately, the real opportunity of a cloud computing service lies in exclusive titles which take advantage of the unique networked server hardware to create a beyond-next-gen experience. This is where OnLive, Gaikai, or QToY would be wise to woo key developers. Imagine, for example, if Blizzard decided to release their next game exclusively on a cloud service. As they are still PC-focused, it could allow an easy entry into the console space, and a single Blizzard title could make a system. Another interesting candidate would be id Software. The company's latest engine, id Tech 5, is based around the premise of massive quantities of unique compressed texture data, which is having trouble fitting on standard DVDs. Id could easily release a special version for a cloud service such as OnLive that uses terabytes of data instead of gigabytes to achieve much higher quality. Interestingly, id has already invested into the idea of on demand gaming with QUAKELIVE.

Out of all these potential players, Microsoft and Sony will likely make the next important moves. And although both companies now seem focused on capitalizing on Nintendo's recent success, they both have much at stake. Microsoft in particular is in a good position to move into cloud gaming. The company already has a large install base, beating out Sony in this generation, and could make a transition to server-side computing without having to sell any new hardware. Furthermore, Microsoft's general software model is already focused on chasing Google into the cloud.

The longer-term advantages are clear, and the current costs of compression and bandwidth will only go down. So it seems likely that gaming is headed into the cloud. But when? The real shakeout will play out over the next several years, as Microsoft and Sony finalize their plans for a new console generation. With Intel's upcoming Larrabee chipset, CPUs are becoming like GPUs in order to reach higher parallel performance, just as Nvidia and ATI's GPUs are becoming like CPUs because they can now afford to be more general. We are moving into a world where any high end computing tasks, including video compression, physics, or rendering, can run efficiently in virtualized form across a sea of processors somewhere out on the cloud. This spells the demise of the home console's long standing ace in the hole: cheap, dedicated, high performance hardware, trumping it with cheap, on-demand, high performance service. In this strange new world, anything is possible, and only one thing is certain: much will change. ☹

JAKE CANNELL works as a graphics programmer at Pandemic Studios, and has been creating graphics engines and demos at work and home for about ten years. He maintains a blog to speculate on such matters at www.EnterTheSingularity.com. Email him at jcannell@gdmag.com.

calling all content creators

create opportunities

The Arab world has an estimated 300 million people looking for quality content. twofour54, based in Abu Dhabi, is the region's premier centre for Arabic content creation in the media and entertainment sector across media platforms including television, radio, film, publishing, online, mobile, music, gaming and animation and we can help you reach them.

Create skills

twofour54 tadreeb is developing and enhancing the skills of media professionals enabling them to deliver immediate success to their business.

Create ideas

twofour54 ibtikar provides media and entertainment industry funding and support for start-up and early stage businesses and individuals with creative ideas for Arabic content and a passion for success.

Create content

twofour54 intaj leads the Arab world with fully equipped HD production studios, the latest post-production technologies, media management, digital archiving, playout and uplink services.

Working in partnership with some of the world's major content creation companies, twofour54 provides your business with the right opportunities to grow in the Arab world.

find us. join us. create with us.
destination Abu Dhabi
+971 2 401 2454. twofour54.com

create.

twofour54
Abu Dhabi

content creation community

thegamedeveloper

50

We spend a lot of time focusing on companies in this industry, and sometimes not enough on the individuals. While it can often be difficult to attribute the achievements of a game to one to one particular person, we have attempted just that—mentioning leads where necessary, and independent accomplishments whenever possible. If possible, where leads are mentioned, teams should interpret this as a group honor.

And so, this is our list, compiled by the editors and our advisory board, of 50 important accomplishments of the last year (give or take) as filtered through the specific people attached to them, in the categories of art, design, programming, business, and evangelism. Within each category, there is no ranking—names are ordered alphabetically. Thanks to all the below for doing what you do, and keep striving for future excellence!

50 game developer

» ART

Jakub Dvorsky AMANITA DESIGN

Games from Jakub Dvorsky's Amanita Design have a distinctly European look that combines the sadness of a faded daguerreotype with flashes of sly, post-modern humor. Mixing collage, hand-drawn art, and animation, games like SAMOROST and the recently released MACHINARIUM have a complex patina and visual density that seems to transcend their Flash origins, but show their Czech animation roots



Amanita Design's
MACHINARIUM.

in ways that speak to a country with its own, rarely seen visual aesthetics.

Dhahib Eng, Jeff Ballinger, and Jason Mitchell

VALVE

Valve's TEAM FORTRESS 2 is a prime example of how art direction can support game design to create a smoother experience for the player. First and foremost, TF2's art direction scores for its whimsical retro vs. space age bachelor pad vibe, with clever cinematics and characterization making potentially generic

characters like The Spy stand out by a mile. In the fast-paced environment of TF2, differentiating units on the battlefield swiftly is critical for player survival. Thus, Valve's smart art direction also has a crucial role in helping players recognize the unique visual signature of their team and opponents.

Kareem Ettouney MEDIA MOLECULE

Most video games look like ... well, video games. But Kareem Ettouney's richly textured, handcrafted visuals for LITTLE BIG PLANET are far more Etsy

and *Real Simple* than cathode ray tube. Not only is LBP's constructed look a perfect fit for the game's concept of user-generated content, but it gives the PlayStation 3 what might be its first broadly recognized visual brand identity. The multitudinous awards the title over the last year have vaulted Ettouney and his team to the top of the visual heap.

Matt Korba and Matt Clausen

THE ODD GENTLEMEN

The Odd Gentlemen's soon-to-be-released game

The MISADVENTURES OF P.B. WINTERBOTTOM began life as a USC student project, and the school's film studies orientation made a deep impression on its design. Referencing the look of German expressionist films of the 1920s, the game's art direction by Matt Korba and Matt Clausen (with an additional nod to illustrator Vidal Perez) is a clever metaphor for its deconstruction of linear time. The main character's comically villainous façade also helps to support his nefarious quest to steal all the world's pies.

Jouji Kamitani VANILLAWARE

Proving that 2D games still have a place in the hi-def world, Vanillaware has perfected its craft across three console generations. By making a total commitment to its aesthetic of painstakingly realized 2D graphics, the studio has evolved a distinctive house style that is instantly recognizable, and transcends regional classification. Kamitani is art director, but also the president of the company, and his claim that the studio is "100 percent artists" shows how seriously he takes the artistic side of Vanillaware's games, with titles like MURAMASA re-interpreting classic Japanese legends in beautifully abstruse ways.

Tomohisa Kuramitsu (AKA Baiyon)

PIXELJUNK EDEN

Baiyon is part of an emerging class of

borderless digital artists that move effortlessly between graphic design, fine art, film, music, and fashion. His work on Q-Games' PlayStation Network downloadable game PIXELJUNK EDEN as both composer and art director presents a thoroughly modern aesthetic that mixes minimalist pop art with gently pulsing techno. As games move further into the mainstream of popular consciousness, these kinds of blurred distinctions between media will become more important, but as an artist who happens to have done a game-related project, Baiyon is a great exemplar.

Ryan Lesser HARMONIX

In creating THE BEATLES: ROCK BAND, Ryan Lesser and his team of artists at Harmonix faced an unusual challenge: How should one respectfully portray the Fab Four in a video game? Particularly when The Beatles' faces and personalities are so deeply etched on the hearts of music fans? Their approach, which avoided the obvious pitfalls of photo-realism as well as the easily dismissed "kiddie" look of cartoon caricature, struck an elegant balance between the two extremes. Lesser and his team show that a tasteful eye combined with lots of research and the traditional skills of hand-animation is the surest way to connect with an audience's emotions—and a little psychedelia goes a long way.

Mickail Labat UBISOFT

The world of PRINCE OF PERSIA had become a comparatively dark and grungy place, rather at odds with its *One Thousand and One Nights* inspired origins. What a pleasant surprise it was, then, to step into the newest PRINCE OF PERSIA and find blue skies over perfumed gardens—a game that seemed to be an oasis far from the death encrusted space marines that critics charge are so prominent in the current generation. Ubisoft Montreal's Mickail Labat is to be commended for leading the charge on creatively reimagining this long running franchise.



Dan Paladin.

Dan Paladin THE BEHEMOTH

Dan Paladin's work on 2D hits ALIEN HOMINID and CASTLE CRASHERS is a welcome reminder that form follows function. His chunky, clean line style is perfectly tuned to indie developer The Behemoth's brand of instant gratification game design. Rather than trying to crowbar over-designed graphics into the relatively low-spec environment of Flash or XBLA, Paladin plays to the strength of these platforms by creating art that is bright and eloquently simple. His art also lends itself

well to more humorous scenarios—it's not often you see poop-prone deer jetting across the screen.

Lee Petty DOUBLE FINE

BRUTAL LEGEND is as much about the look of heavy metal as it is the sound, and under Lee Petty's art direction, the music's lowbrow high fantasy album cover art comes to life. Beyond simply bringing Joe Petagno's *Orgasmatron* cover into 3D, Petty incorporates a wealth of fantasy art influences into BRUTAL LEGEND. From Frazetta's rough-hewn figures and Brom's leather strapped fetishism, to Beksiniski's cyclopean forms, the Tim Schafer-headed game presents a vast canvas of fantasy art history.

» DESIGN



Jenova Chen

Jenova Chen THATGAMECOMPANY

Chen and colleagues at Los Angeles studio TGC this year released PSN title FLOWER to great critical acclaim. Not only does the game play differently from others, it eschews violence and many traditional play mechanisms in order to create a new sort of

experience. Chen's design for FLOWER has sparked no end of critical discussions, design debates, and GDC topics, and has inspired other designers to look further into the fun of experiential and experimental gameplay. Chen's well-spoken philosophical ideas and theories have cemented his position as one of the more forward-looking designers in games.

Todd Coleman KINGSISLE

KingsIsle lead designer Todd Coleman has claimed for years that MMOs can be successful targeting smaller niches. His first attempt, the free to play/subscription hybrid PC online game WIZARD 101, merges a little *Harry Potter*, a little POKÉMON, and a little *Magic: The Gathering* into a boutique MMO that just snared its five millionth registration. The youth and tween space for MMOs is a crowded one, but Coleman and company have proved that specific targeting and clever game design, rather than generic catchalls, can be a formula for success.

George Fan POPCAP

The entire PopCap design team has done much to legitimize casual game design by producing titles that combine easy accessibility with deep play mechanics. Recent releases such as PLANTS VS. ZOMBIES demonstrate the PopCap style of creating games that are deceptively simple on the surface, but remain rewarding over the

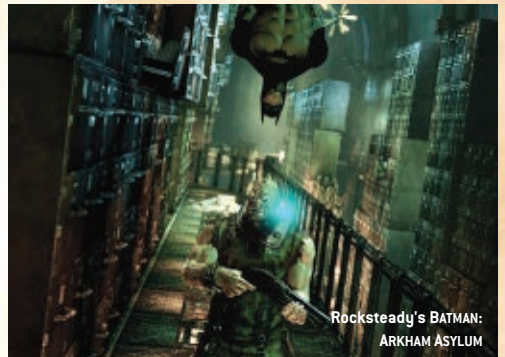
long-term. In PVZ, Fan's take on the alternately hardcore or faceless genre of "tower defense"/strategy is cleverly personalized and masterfully iterated, without alienating the more strategy-minded players.

Nate Fox INSOMNIAC

The whimsical cartoon world of SLY COOPER was Sucker Punch's bread and butter during the PlayStation 2 era, but with the transition to latest gen consoles, the competition has gotten altogether darker in tone. Sucker Punch's response was to update the proven action elements from their well-crafted franchise with a modern, open-world design and edgier art direction. The resulting INFAMOUS is the perfect intelligent fix for an audience that grew up with cartoons, but now craves more adult fare, and gives the PlayStation 3 another much-needed system exclusive.

Sefton Hill ROCKSTEADY

Hill is the creative director of Rocksteady, a company which came from relative obscurity to create the greatest *Batman* game yet made (BATMAN: ARKHAM ASYLUM). Rather than turning the hero into "punching batarang guy" as so many developers have before, Hill and company created a thoughtful, stealthy combat-oriented game, with well-designed levels, clever tricks, and scads of new ideas throughout. From Batman's "detective vision" to the flowing combo-oriented



Rocksteady's BATMAN: ARKHAM ASYLUM

combat, to the silent takedowns, ARKHAM ASYLUM manages to make cautious play enjoyable, without being slow.

Tom Leonard VALVE

LEFT 4 DEAD and its sequel reinvigorated cooperative multiplayer games in such a logical way that it seems astounding it wasn't done before. Players stick together and help each other out because the enemy AI is designed in such a way that you're a fool not to. You actually want to work in partnership, instead of being forced by the old "you need two people to open this door" gag. By forcing most of the large scale battles to take place in motion, rather than from single defensible positions, Leonard and his Valve colleagues have created a new variant of co-op that has captured the imagination of many.

James Ohlen BIOWARE

Most people know BioWare for "the doctors"—Ray Muzyka and Greg Zeschuk, but James Ohlen is a key design brain behind many of the BioWare classics,

including BALDUR'S GATE I and II, as well as KNIGHTS OF THE OLD REPUBLIC and NEVERWINTER NIGHTS. Ohlen is currently heading THE OLD REPUBLIC, BioWare's entry into the MMO space, ensuring that the tricky franchise—on a massively multiplayer scale, of course—is properly infused with the studio's brand of thoughtful play.

Hirotoshi Shiozaki STREET FIGHTER IV TEAM

STREET FIGHTER IV brought lapsed fighting game players back into the fold, and even created some new fans. This was a marketing win, to be sure, but the design of the game for lower-level play lowered the barrier of entry such that players were less daunted. The challenge modes also served as a tutorial for the combo system, teaching players how to link certain moves together. Shiozaki and the folks at Dimps and Capcom have paved the way for a resurgence of 2D fighting games in general.

Shuntaro Tanaka and Ryutaro Nonaka SEGA

Strategy RPGs have traditionally been a hyper-

niche genre, beloved by its adherents and almost inscrutable to outsiders. Into this hidebound form stepped producer Ryutaro Nonaka and director Shuntaro Tanaka with VALKYRIA CHRONICLES, a game that put a fresh, youthful (and blue skied) face on the typically grognardian elements of small unit fire and movement tactics. Massive sales didn't necessarily result, but perhaps when the franchise moves on to the PSP it will introduce even more players to the pleasures of tactical games.



Jarrad Woods
CAPTAIN FOREVER

CAPTAIN FOREVER developer Farbs has had about eight years of commercial game development experience at companies like BigWorld and 2K Australia—on either canceled or unreleased games. But in his free time he started making independent games, and the fascinating and transgressive ROM CHECK FAIL—which mashed up classic arcade games in cheeky ways—was just the start, a *Grey Album* for games, if you will. His new project, and his first as a full-time indie, CAPTAIN FOREVER, just won the IGF China Best Game award for the top Asian-Pacific indie title.



Chris Hecker
» PROGRAMMING

Chris Hecker
EX-MAXIS

At Maxis, Hecker was responsible for pushing forward the procedural animation, character tessellation, and rendering found in SPDRE, which, regardless of the game's overall success, did some spectacular technical things. Hecker is also the cofounder of the original Indie Game Jam, of which there are now multiple international derivations, inspiring indies to collaborate and work together in new ways. Now that Hecker is once again independent, one can only anticipate what technical (or, indeed design!) delights he may create with his "free time."

Kristján Jónsson
CCP GAMES

Running a gigantic MMO on a single server carries a host of difficulties—from latency issues and player capacity to time synchronization. However, when having a large, persistent universe is the primary selling point for a game, using a single-sharded architecture is the only way to effectively

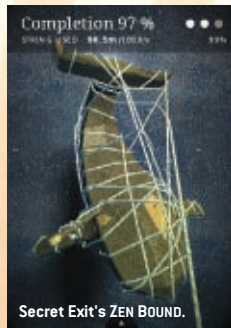
create that environment. Kristján Jónsson's team on EVE ONLINE continue to impress technically by delivering a consistent universe on a vast scale, ever more eye-opening because of the game's growing subscriber base, now over 300,000.

Rob Knopf and Nyung Chul Kim
CN FUSIONFALL TEAM

The Cartoon Network has jumped into the free-to-play MMO space with FUSIONFALL, a new web-based game built on the Unity game engine and incorporating Cartoon Network characters. The project is a great example of how smart, low-cost technology can give media companies an easy path to their increasingly connected audiences. In the browser space, 2D is much more common, and for an MMO, even that is difficult. Taking it to 3D and keeping the "massively" bit intact is what puts Knopf (at Cartoon Network), Kim (at Korean developer Grigon), and the FUSIONFALL team on our list.

Jetro Lauha
SECRET EXIT

As a past and present Finnish demoscene coder, and as part of the team that created the first licensable



3D mobile game engine, Lauha is at the cutting edge of making 3D work on low spec machines. So it should be no surprise that he managed to get complex rope-wrapping working in Secret Exit's iPhone hit ZEN BOUND. He's also gotten ragdoll physics working in the device in the form of a port of his 2006 title STAIR DISMOUNT, which is not yet released, but is fully playable. Lauha makes the list for his continued pursuit of power in small places.

Taku Murata
SQUARE ENIX

Square Enix's custom-built Crystal Tools engine has only recently gotten to step into the limelight with playable demos of FINAL FANTASY XIII, which bring the company's signature baroque visuals to real time, in incredibly high fidelity. As the general manager of R&D at the company, Taku Murata has spent years shepherding the Crystal Tools framework, and despite long development times, Square Enix's lush engine, on display in real-time for all to see at Tokyo Game Show, appeared a genuine wonder to behold. Summon, anyone?

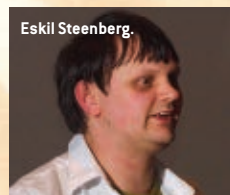
Jean-Charles Perrier
QUANTIC DREAM

Few companies in the entertainment sector of games are doing more to pursue visual realism than French developer Quantic Dream with PS3 exclusive HEAVY RAIN. Though the viability of realism as a method of expressing artistry in games can be debated, QD's singular focus on what they believe

in, both from a gameplay and technical standpoint, is admirable. Perrier's tech work, building the engine and the tools that support it, helps lead a team that is creating realism without sci-fi or post-apocalyptic scenarios to fall back on for "It's supposed to look like that" excuses.

Adam Saltsman
FLIXEL

Austin-based Saltsman is a prolific indie game creator, probably best known for Flash and iPhone roof-jumping game CANABALT. But he exemplifies a sharing trend among indies that's great to behold. During the course of developing his games Saltsman built up a library of custom ActionScript 3 files called Flixel that takes a lot of the grunt work out of Flash game creation. Flixel provides handy implementations for collision, particles, parallax scrolling, sound utilities, tile maps, fast rendering, and more. The best part? In the true indie spirit, he provides everything for free at flixel.org.



Eskil Steenberg
QUEL SOLAR

Swedish indie developer Steenberg is in a class all his own. His multiplayer online FPS, LOVE, was created entirely from scratch, engine and tools included. And Steenberg

The image features a collection of colorful, 3D-style icons representing various aspects of game development and technology. These include a telephone, a dollar sign, a person, a lightbulb, a rocket, a person with a gear, a speaker, and a person with a plus sign. The icons are arranged in a circular pattern around the central text. The background is a vibrant green and teal abstract design with circular patterns and lines.

GD10

Game Developers Conference®

March 9-13, 2010 | Moscone Center, San Francisco

www.GDConf.com

THINK
SERVICES
A DIVISION OF UNITED BUSINESS MEDIA, LLC

Learn. Network. Inspire.

50 game developer

did it all himself. The visuals are gorgeous procedurally-generated impressionistic artistry, the world builder is amazingly involved and detailed, and he's been extremely open about the whole project. Steenberg is one of those rare games programmers that just doesn't seem to realize he's doing it the hard way—because to him, it isn't.



Steve Swink (right).

Steve Swink and Scott Anderson

ENEMY AIRSHIP

SHADOW PHYSICS is an intriguing prototype that takes the 3D lighting effects we take for granted in games and uses the technology to create a new spin on traditional platforming challenges. Based on the idea of manipulating objects in the foreground in order to cast shadows that become the actual playfield, SHADOW PHYSICS is thinking outside the light box, and is a great showcase both for Flashbang's Swink (also working on 3D web games for the Blurst portal) and full-time mainstream industry participant Anderson, as well as the role of technology in indie iteration and experimentation.

Hiroki Ueno

YUKE'S

As executive technical director for Yuke's, Ueno

has been dealing with motion of and interactions between human bodies for some time. The developer is known for its wrestling titles, but with UFC UNDISPUTED 2009, a large step forward was taken in body interpretation. The human body isn't easy to deal with, because the user knows how it should work, and how it shouldn't, and it's especially tricky for close-contact titles like MMA. That the UFC team managed to do chokeholds and kimuras so convincingly is a testament to their skill, and the depth of experience they have with the subject.

>> BUSINESS

Ron Carmel and Kyle Gabler

2D BOY

These two ex-EA indies formed 2D Boy, and are creators of the peerless WORLD OF GOO. But we're awarding them for business savvy, since the duo have been very vocal (Carmel especially) about DRM as a pointless way to stop piracy. Running with a 90 percent piracy rate on PC, as many other games do, Carmel employed every trick in the book to maximize



Ron Carmel.



Satoru Iwata.

profile and revenue, with cut-price bundle deals and even a "pay what you like" policy for the game, allowing players to pay 1 cent and up for a normally \$20 game. Experiments like this embody the indie spirit of business and entrepreneurship—and actually work, too.

Satoru Iwata

NINTENDO

As the Wii and DS suite of consoles take the world by storm, Iwata and Nintendo continue to play a game of patient business and design innovation. The Wii was the last console to take a price cut, and the release of the new DSi, though a small hardware upgrade, was very successful. Of course, not all attempts to shift boundaries were quite so successful, as Wii Music proved that Nintendo doesn't simply print money with every effort. But the important lesson is that these less successful

experiments don't cause Nintendo to stop trying risky things, and Iwata's development and iteration-centric leadership is a big part of the firm's continued massive business success.

Steve Jobs

APPLE

Apple revolutionized album and single purchasing with its iTunes store, and for a long time, games looked longingly at the resulting shift in music digital distribution. Then along came the iPhone and the App Store, and within weeks, it was clear that another paradigm shift was born, this time in gaming. Though it's not without its problems—particularly regarding monetization—the service is tremendously successful, both for Apple and for select game developers who get on board at the right time. Most excitingly, as Jobs perhaps imagined when

planning the service, the miniscule barrier to entry means that the next iPhone game rags to riches story is only just around the corner.

Peter Moore

EA SPORTS

CEO and bona fide biz personality Moore left the Xbox business in 2007, joining EA Sports as its president not much later. EA's sports brands have long been successful for the company, but the 2009 release of EA Sports Active was not only a great success for the sports imprint, it was EA's best Wii launch ever. With smart moves into the fitness and even MMA areas as some "evergreen" franchises like MADDEN wilted marginally, EA Sports' ability to adapt and innovate has certainly increased under Moore.

Rob Murray

FIREMINT

Melbourne-based Firemint is pointed to as one of the big winners in the iPhone development scene, and rightly so. While the studio has done work-for-hire for publishers in the past, it's really starting to break out with original IP. The company has two games that have apparently grossed over one million dollars, in two very different price brackets and categories. There's the casual, pick up and play 99 cent FLIGHT CONTROL (now over 1.5 million sales!), and the bigger budget 3D, fully optimized REAL RACING, which has sold for \$6–\$10. CEO Murray has successfully diversified his game sizes



Firemint's REAL RACING.

and revenue streams in a smart way, and it seems to be paying off.

Mark Pincus ZYNGA

Now that everybody and their grandmother has a Facebook account, social networks are looking like the new Gold Rush country. Quality games and ruthless user aggregation tactics will be the differentiating factor between VC money sinkholes and long-term success. In competition with companies like fellow honoree Playfish, former Tribe.net supremo Mark Pincus is positioning Zynga as the dominant player, with efficient, well-advertised titles like FARMVILLE, MAFIA WARS and YOVILLE in the company's lineup. Social network gaming shouldn't be underestimated, despite some skepticism of business model and design chops from the conventional game biz, and given reports of 9 figure yearly revenues at Zynga, it behooves everyone to pay attention.

Kristian Segerstråle PLAYFISH

In late 2007, Segerstråle and several other early

members of the Glu Mobile family branched off to dive into the world of social gaming, launching titles like PET SOCIETY. Initially the newest in a long line of redheaded stepchildren of the gaming industry, the space, especially in Facebook games, has grown by leaps and bounds. Increasingly complex, artistically relevant titles are being twinned with impressive revenue numbers, thanks to application of Web 2.0-style concepts and iteration to the gaming space, and Playfish is one of the companies at the forefront.

Martin Tremblay WARNER BROS. INTERACTIVE ENTERTAINMENT

Warner Bros. has made multiple attempts to enter the game industry, but it appears that this time, it's actually working. The acquisitions have so far been smarter to outside observers (Monolith and TT Games both have their strong suits), and the licensing more canny, as the company's work with Eidos on BATMAN: ARKHAM ASYLUM was one of the more celebrated games of the year. What impressed us more than that was WBIE's marketing of SCRIBBLENAUTS,

since it's very uncommon for a publisher to put a full marketing push behind an original DS title from a smaller developer. For this, WBIE and president Martin Tremblay deserve credit.

Paul Thelen and Jeremy Lewis BIG FISH

Founder Paul Thelen and CEO Jeremy Lewis of Big Fish have been growing their casual games business for seven years, based on one of the most dominant portals in the PC casual game biz, a solid foundation of in-house quality games and canny international expansion. This past year has seen the launch of the casual MMO FAUNASPHERE, and the inking of a partnership with Harlequin Enterprises that will allow for games based on the book publisher's romance series as well as tie-in novels to Big Fish's games. Even if the company was one of the main participants in the potentially developer-catastrophic PC casual game price war—well, they've come out on top, haven't they?



Twisted Pixel's THE MAW.

Michael Wilford TWISTED PIXEL

Austin-based independent developer Twisted Pixel, like The Behemoth, is one of those indies that really knows how to make the business work. They make

good games, certainly, but it's that ability to get the games—like XBLA hits THE MAW and 'SPLOSION MAN—in front of the eyes of customers that keeps them going, so they can make more good games. Announcing your new game as an April Fool's joke sent to a host of game journalists (as Wilford and team did with 'SPLOSION MAN) isn't a bad idea, either.

» EVANGELISM

Jason Della Rocca PERIMETER PARTNERS

As the executive director of the IGDA from 2000 through early 2009, Della Rocca has played a key role in advocating for game developers and the games industry in general on a multitude of issues. His strong leadership on elements such as quality of life issues and game crediting was important in growing the IGDA by a massive factor during his tenure. While his departure—to run a consultancy devoted to counseling local and national governments on attracting game development talent—was surprising, his selfless work deserves both mentioning and honoring here.

Jerry Holkins and Mike Krahulik PENNY ARCADE

As far as evangelists within the community, Holkins and Krahulik are practically bar none. Their discussion of games is intelligent and relevant, if occasionally too spiky



Jerry Holkins.



Mike Krahulik.

for developers, and their Penny Arcade Expo is an undeniable nerd haven. But they also promote the goodwill of gamers and developers alike through their Child's Play charity, which donates money, games, and toys (to the tune of \$1.4 million in 2008) to children's hospitals in the U.S., Canada, and elsewhere. Their good work reflects well on all of us, partially because we are participating directly in their successes.

Dennis McCauley EX- GAMEPOLITICS.COM

Before McCauley, there were no journalists dedicated specifically to the intersection of games, censorship, social issues, and politics. With his site GamePolitics.com (which he recently left to pursue non-journalistic avenues), McCauley essentially created a vital news beat all of his own. Though the web site was purchased by game consumer advocacy

5 game developer

group the ECA in October 2006, and McCauley is no longer involved in running the still-extant outlet, his work in documenting and reporting on the legislation (and Florida-based lawyers!) that affect games, and the information that allows game developers and players to get involved in shaping the course of our own industry, has been vital.



Petri Purho (with monocle).

Petri Purho KLOONIGAMES

For a sustained period, Finnish indie creator Purho made a game every single month. Purho's efforts have not only inspired further indies, but his theatrical, whip-smart attitude to both game development and public speaking help exemplify the role of the independent game creator as the neglected personal angle. Nowhere was this better demonstrated than his "creation of an entire game in 5 minutes" at the Indie Games Summit rant session, something he was asked to replicate at several other worldwide shows. Whether the feat was true (spoiler: it wasn't), it wrapped the glory of performance and the joy of game creation into one breathless whole. Bravo.



Tim Schafer.

Tim Schafer DOUBLE FINE

BRUTAL LEGEND is the result of a strongly-realized vision by Tim Schafer and crew, but more important for the industry, Schafer makes games look cool. Not just his own game, with its mass of big stars with cultural cache, but through his humorous and personable appearances in the public eye, most notably on *Late Night with Jimmy Fallon* on NBC, Schafer has managed to make game developers look like real, funny, witty people. While Will Wright holds up the supernerd end, Schafer pushes forward the personable and sociable side.

Kiyoshi Shin IGDA JAPAN

As the head of IGDA Japan and a prominent game columnist and media commentator, Kiyoshi Shin has done a lot of work to bridge the gap between Japanese and Western developers. Aside from his work advocating for independent development in Japanese outlets like Nikkei, the two-year-old Sense of Wonder Night is one of his most notable achievements. The event is held during the Tokyo Game Show, and like GDC's Experimental Gameplay Showcase for an international group of indies, gives creators

10 minutes in which to present their games to a global audience. The cultural sharing and networking gleaned from this event is to be undervalued.

Tommy Tallarico and Jack Wall VIDEO GAMES LIVE

Tommy Tallarico and Jack Wall have been very active in promoting their traveling worldwide concert, Video Games Live, for several years now. The show brings game scores to the masses via live orchestral and rock performances, highlights and features on stage the creators of many of these scores, and helps to keep game music from being relegated to background noise. Indeed, the Brazilian government subsidized the group's show in the country as furthering the artistic development of students. Anything that advances popular and governmental recognition of games as an important entertainment medium warrants a place on our list.

Yoichi Wada SQUARE ENIX

Yoichi Wada is the president and CEO of Square Enix, but he's on this list because of his chairmanship of the Computer Entertainment Supplier's Association (CESA), Japan's equivalent of the ESA. Wada has become increasingly vocal about Japan's reasons for falling behind in the game software world, citing the country's lack of development hubs and discussion forums,

which would promote an exchange of ideas, as well as Japanese companies' tendency to solve the same problems separately. Wada and CESA—through outlets like Japanese developer conference CEDEC and Tokyo Game Show roundtables—are serious about helping the Japanese industry reclaim some of its lost luster.

Will Wright STUPID FUN CLUB

Though he no longer works at Maxis, where he created SIMCITY and THE SIMS, and has moved on from working exclusively on games, Will Wright may now glow even brighter as a beacon for electronic entertainment. Freed from the burden of SPORE, he can innovate anew, and anyone who can discuss games and Russian space technology in the same breath is helping to elevate games from the

"murder/smut simulator" stigma which at times has threatened to seep into the industry's pores from popular opinion.

Derek Yu TIGSOURCE, MOSSMOUTH

An active indie developer himself, Yu's games, from DIABOLIKA to IGF Award-winning AQUARIA to SPELUNKY, provide inspiration for the independent games community. But we're honoring him here for running TIGSource, which is a wonderfully random and helpful indie game community. Not only has Derek and the site been active in helping defend indies when they get raked over the coals for unexpectedly fragrant issues (see the Edge Games controversy), the regular TIGSource competitions birth often brilliant games on startlingly esoteric topics. 🎮



Derek Yu.

ACCESS IT IN REAL TIME

gamedeveloper

DIGITAL EDITION



Game Developer magazine's 6 month and 1 year Digital Edition subscriptions give you new issues delivered promptly, full searchable access to more than 50 back issues, plus downloadable PDF versions, easy access from any web browser, and more!

Subscribe today!

[HTTP://WWW.GDMAG.COM/DIGITAL](http://www.gdmag.com/digital)

THINK
SERVICES
A DIVISION OF UNITED BUSINESS MEDIA, LLC



postmortem

Scribblenaut



OUR CREATIVE DIRECTOR AND COFOUNDER JEREMIAH SLACZKA SAT WITH A SINGLE SHEET OF PAPER, ABOUT to pitch our next big original IP to a major publisher. The paper had a sketch of a tree with a star in it, and a box underneath. Jeremiah slid the paper across the table, and simply asked "What would you write to get the star?" The publisher rep stared at the paper for a minute or two, and then glanced up and said, "Wow."

That was the first external reaction to SCRIBBLENAUTS.

The actual SCRIBBLENAUTS story started months earlier, with the founders of 5TH Cell gathered around our conference table. It was early 2008, and we had just received our first royalty check from DRAWN TO LIFE, and wanted to invest it in a new project that would propel our company to the next level. For the past several years we had dutifully continued with publisher-funded development of our internally designed original IP.

uts

post mortem



This had allowed us to build up tech and staffing, but we had this strong creative desire to make a game our own way, not only with product design, but with control over our development process and planning. We believed the best path would be to call the shots ourselves, from the start of development until it was time to announce the brand. There was a quiet excitement in that conference room as we discussed the platforms we could afford to self-fund on, and what was the best path to take. After some discussion, we agreed to stay with what we knew best: the DS.

Jeremiah set the success bar for this title at “Compete with the top Nintendo first-party titles,” with a time frame and budget of a much smaller third party title. We all knew our move onto consoles was coming soon, and we all wanted our grand finale for original IP on the DS platform to be memorable, and completely accessible to everyone. We set BRAIN AGE, NINTENDOGS, and WARIOWARE:

TOUCHED as our comparable games. We exited that meeting with a plan on where we wanted to go, but were still not completely set on a game concept to take us there.

A bit later, Jeremiah had a dream where he was stuck in a room and had to solve a puzzle using objects in the room to escape to the next room. He filed this idea away as “rooms.” At the time, the idea behind SCRIBBLENAUTS was more about using the stylus and typing words to create your own short stories. The idea was cool, but it was more of a toy than a game, and Jeremiah felt it was missing the necessary gameplay to make it marketable. At some point, the game concept behind “rooms” merged with the word concept, and SCRIBBLENAUTS was born.

Jeremiah presented it, and there was consistent internal feedback that the concept of “writing anything and having it appear and behave as it would in real life” was technically impossible, especially on the

Nintendo DS. Our tech director, Marius Fahlbusch, took one look at the idea and fell in love with it. He said it could work, and that was good enough for us. We green-lit the project internally and started development. The final version of SCRIBBLENAUTS allowed the user to write any word, that word would appear as an object, and the user could manipulate that object to reach the specified goal. Our tagline was “Write anything, solve everything”, and we were successful on delivering on that promise.

WHAT WENT RIGHT

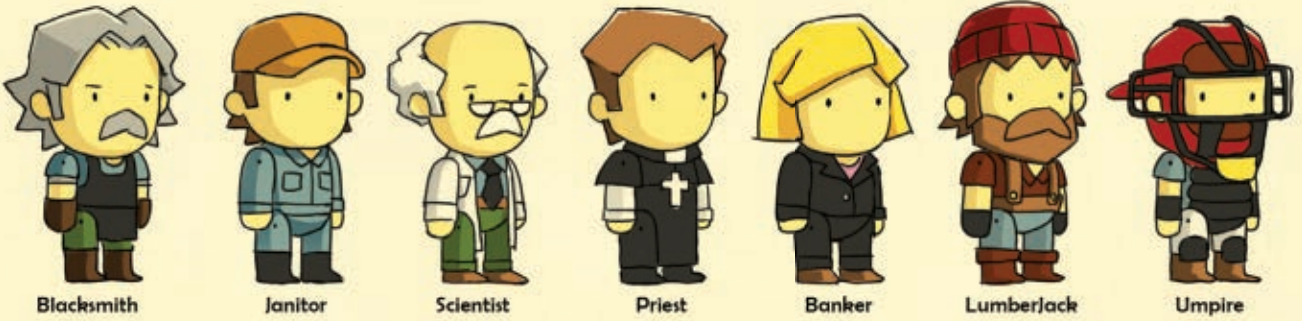
1} Right Partner, Right Time. For many reasons, SCRIBBLENAUTS was un-publishable as a concept. I’ll go into more detail below on how everything works, but in order to experience the game, the entire technology framework had to be in place and functional. There was no early demo or “vertical slice” possible with SCRIBBLENAUTS. It either worked, or it didn’t. Publishers want to see

visual progress, and we wouldn’t have been able to show any. In fact, it took nearly six months before we had anything on screen, which is a typical full development schedule for a third party Nintendo DS title.

At 5TH Cell, we subscribe to a development ethos of iterating until we have something we’re happy with. The game has to be fun, right? If something doesn’t work, it takes time to change, and you can’t build that time into a schedule that was agreed upon two months before the change needed to happen. Also, the higher the bar is set, the more iteration is required, which is incompatible with most publishing deals. There are some developers that might be able to tell a publisher “Give us a blank check and we’ll talk to you in a year when we have something we’re happy with,” but we were not one of them. Publishers are successful by minimizing financial risk, and it makes sense to see something early, with the goal of mitigating that risk before investing most of the funds to finish the game.

So we kicked off SCRIBBLENAUTS on our own, focusing on tools and tech first, while tweaking our style and brand appeal. Maxwell, the main character, took three months and dozens of concepts before we were satisfied with him. Our engineers tackled the data inheritance system piece by piece, and our artists worked on the huge amount of art. Wordsmiths perused dictionaries, wikis, encyclopedias and the net to reach our goal of including “everything.” We made steady progress without having to worry about a publisher withholding a milestone, or stopping natural development to jump through unnecessary hoops. Everyone was focused on making the game, and making it the smartest way possible, which was critical to how the game eventually came together.





While shopping SCRIBBLENAUTS around, we had three criteria that our partner should meet. First, we wanted an established publisher that would be willing to invest substantially in marketing and also have the resources to support a successful product at retail. Second, we wanted a publisher with a smaller slate of releases so SCRIBBLENAUTS would have more attention within the organization. Finally, we wanted a partner that would work directly with us in a collaborative environment to market, promote, and build the brand. Warner Bros. Interactive Entertainment exceeded our expectations in all three of these categories, and the enthusiasm and hard work from everyone there is a huge factor in the success of the title.

2) Objectnaut. The first reaction from fans (and often the first question from media inquires) is “How did you manage to cram so much stuff in the game?”

We made it possible with a tool we built from scratch called Objectnaut. Early in the concept phase, our technical director had to figure out the specific way we could cram thousands of unique objects into the game and have them behave—and interact with all other objects—in a believable way. Brute force wasn’t feasible, so he devised a data inheritance method which defined how our world operates. Models and animations were imported from 3ds Max and augmented with additional hotspots in the hierarchy to add information for attachments, lights, equipment, and others. We built around 40 key properties to determine an object’s behavior, including physical properties, material, electricity, temperatures, and many more attributes. Lists could define how AI would interact with certain categories or specific objects, what tools to use in combination with other

objects and what specific “living” objects would be afraid of. A trigger system allowed designers to tie a series of actions to all gameplay events, defining how an object is used, what happens when it melts, and what happens if a cat is put into a fridge. In all this, filters could be applied to either match entire categories, subcategories, single objects, or even to exclude collections from the filter, all to manage the huge amount of possible interactions.

Objectnaut and the data inheritance system allowed our designers the ability to edit whole categories or to delve into the specific properties of a unique object that needed extra attention. SCRIBBLENAUTS would have shipped in 2015 (optimistically) if we had to manually and individually assign properties to thousands of unique fish, flowers or humanoids, let alone everything else.

3) Art Direction. We had good estimates at the start of development regarding scope of content. Our previous titles were 2D, and we hand animated all the sprites. Hand animation for 2D art is a huge undertaking, and the raw amount of art and animation required for SCRIBBLENAUTS demanded another approach.

For animation, we went with a skeleton system that could be applied to general object categories. For example, most humanoid characters in SCRIBBLENAUTS share a single set of general animations such as run, walk, fall, and so forth. After the general animations were complete, we visited the more unique objects and created special animations. This system worked for the most part, but still required Nathan Hernandez, our animator, to spend significant time on objects such as the Hydra, Leviathan, and other fantasy creatures that didn’t

fit into a category. These objects remained the exception to the rule, and animation remained manageable throughout the project. We finished object animation ahead of schedule, which allowed us to add fun animations at the end for some of the internet memes like Keyboard Cat.

The DS operates using a palette-based system for the art format. We ran into significant issues on previous projects with using too many unique palettes with too many colors. Tracking all the separate palettes, and technical information associated with them resulted in excess documentation overheard as well as stability issues during the art import process.

Amazingly, our lead concept artist Edison Yan was able to come up with only 24 different colors for the SCRIBBLENAUTS palette. This means every object in Scribblenauts consists of a combination of the game’s 24 specific colors, which is hard to believe at first glance. Having such a limited palette allowed us to write a tool that instantly converted everything, resulting in art assets moving from the artists in production into the game within a few days of delivery.

4) Level Design from Project Start.

We devoted a good chunk of time to designing levels on paper early on in the project. Because SCRIBBLENAUTS is a game where the designers could use anything they could think of in a level, it was really important to break away from habits in traditional level design and really take advantage of our dictionary. While initially our level designs were simple and familiar, the more time we spent designing on paper, the more interesting and outlandish our designs became. The amount of time we spent brainstorming, iterating, researching, and challenging each others’ ideas meant we had an increasingly varied landscape to pull from, and from



post mortem



there we chose which levels would go into the final game—to the tune of more than seven-hundred level designs, sampling as much of our dictionary as possible to give players a sense of its scope.

5) Keyboard and Handwriting. The original design of SCRIBBLENAUTS called for writing letters, with the stylus to serve as the main method of word input—we loved the visceral feeling of writing and watching an object appear. Our team implemented a keyboard mode as backup, and this acted as an easy way to test objects while we built our own custom letter recognition.

Eventually we realized that no handwriting recognition software could operate faster than a keyboard. We still spent time refining the letter recognition, but it was clear keyboard input would be the primary input method for SCRIBBLENAUTS. Moving away from a system that we invested so much time into, and still thought was innovative, was difficult but ultimately the right decision.

WHAT WENT WRONG

1) Testing Level Content Came Late. Unfortunately, a by-product of building a game that needed everything at the front end resulted in our level designers missing final functionality until about three months before final submission. As stated above, we had hundreds of great level concepts on paper and built in our tile editor, but no way to know if they worked or how they actually would play. We also had key missing functionality due to a huge backlog of dependencies on item properties.

In the last two months we had to pull our creative director Jeremiah Slaczkka and lead designer Matt Cox into level creation, which detracted from their focus on polishing the game. They sat together every day and ended up modifying and

tweaking most of our level content. Thankfully, the process of building and scripting a SCRIBBLENAUTS level was straightforward, and the tools were robust. Because of time constraints we had to cut some extra level content that was in the game, but we hit our goal of 220 levels.

2) Localization. Our products have always released worldwide, so we planned for localization in SCRIBBLENAUTS from the start. We built in localization support for the dictionary



in Objectnaut, and planned to handle in-game localization as we had in our previous titles. Unfortunately, SCRIBBLENAUTS ended up with far more text than we anticipated, and because of the level finalization coming late, our hint text that displayed at the beginning of each level wasn't localizable until the last minute.

Our localization system for previous games had relied on a custom tool which pulled text from .txt strings and exported to Microsoft Excel, which could then be sent to the localizers. What was sent back was imported using the same tool and showed up in game. The tool

itself was very useful, but optimized for previous titles that only had game text. With SCRIBBLENAUTS, we had game text plus a huge dictionary, and they both utilized different systems.

We delivered SCRIBBLENAUTS in English, French, Spanish, Brazilian Portuguese, German, Italian, Dutch, U.K.-English, Danish, Finnish, Swedish and Norwegian. We had unique dictionaries for every language, and these word lists were constantly being updated. Version

could only spend a limited amount of work on it. We discussed a secondary D-pad control option midway through development only to come to the conclusion it would take a single person 3–4 weeks to integrate it. On our self-funded schedule, that route was not an option.

The root cause of this issue was limited time, and our decision to focus more on delivering what we had promised with the “Write anything. Solve everything.” slogan was the right one in the end, but it hurt the second most important part of the game, the controls. We still feel stylus control was the correct decision for SCRIBBLENAUTS. The DS is a casual platform by design and allowing non-gamers the ease of use of a pencil-like system for a game targeted at everyone over a D-pad only scheme was overall the best choice. But in retrospect; our implementation of those controls was only at 50 percent of where it needed to be.

It would have been better to cut a secondary feature early on, such as the Wi-Fi connection, to allow us the time to fully realize stylus controls. However by the time we realized controls would take weeks to get right from play testing, it was too late to cut features.

So how did this happen? The stylus in SCRIBBLENAUTS had a dual purpose, it controlled Maxwell's movement, and it also controlled interaction with objects. The system was trying to manage far too much, and there were only a few safeguards in place to ensure the user didn't tell Maxwell to move by mistake. The small screen size of the Nintendo DS and miniscule touch regions of many objects also caused the user to move Maxwell when they were trying to tap on an object.

Stylus movement was just one part of the problem though. Jamming a fully functional physics engine written from scratch on the DS was another

control became a nightmare as we tracked dozens of files for both dictionary files and word lists.

The system, although severely stressed, made it through, and we shipped all languages. If I could revisit pre-production, we should have either invested in building or licensing a single web tool to manage all the game text.

3) Game Controls. The sensitivity of the stylus controls are easily the biggest issue in SCRIBBLENAUTS, a fact that has been reinforced by reviewers and users. We knew this was going to take a big hit from reviewers, but we



huge challenge that just couldn't be wrangled to the point it needed to be within the time frame we had, thus causing objects to bug out very easily.

4) Did you mean...? There was a disconnect in terms of how homonyms (words spelled identically that have different meanings) were handled in English vs. other languages. This issue was an extension of the localization concerns, but was severe enough to warrant its own point. For English, each homonym had a clear category we could place it in, and when the user typed or wrote the word there was a specific description for it.

We ran into three major issues with this system when we started localizing. The first issue was with character length. Many German and Scandinavian words and descriptions overflowed our character limit for the GUI. We had to shrink fonts for that menu, as well as limit the number of word choices to three, instead of the five we had originally planned for.

The second issue was with how the words were localized. The homonym issue was not clearly communicated to the localization team, so they went through and added a single localized word as an alternate for numerous unique objects, resulting in an unplanned, but highly necessary scrubbing of the entire word list in every language, very late in development.

The final issue came with the categorization of homonyms in foreign languages. Most languages have fewer words than English, and only have a single word that makes sense that can describe two homonyms. With the international SKUs we ended up with many cases of the same descriptor for the same word. In French for example, we had "Did you mean SPATULE [accessoire], SPATULE [accessoire], or SPATULE [accessoire]?" This was due

to three different kitchen tools being translated separated as SPATULE and no available word to describe it other than ACCESSOIRE (Accessory in English).

5) Bugs, bugs and more bugs. QA on this game was a lot like trying to clean the debris from the inside of a tornado by throwing yourself into its center and hoping you can catch everything flying around. A game with tens of thousands of words and infinite interactions was bound to break, and while we nailed down over 5,000 bugs in our tracker, players still found countless crashes and game breaking bugs like the now famous handcuff-plus-refrigerator/vending machine exploit that allows users to cheat through levels and instantly bring the Starite to them. Or the fairy godmother plus basilisk bug that would create infinite roses until the game was brought to its knees by an overflow of flower sprites.

These are the kinds of issues that are nearly impossible to test, or even find, especially with the very limited window we had between the Beta and launch candidate.

DON'T WRITE IT OFF

» E3 2009 was an amazing experience. Jeremiah, along with myself and our lead designer Matt Cox, flew down to Los Angeles, not really knowing what to expect. We had done trade show events before, and while there was interest in DRAWN TO LIFE and LOCK'S QUEST, we felt the hype behind both of those titles was lacking.

WBIE had prominent signage and several kiosks for SCRIBBLENAUTS. On day one of the show we had traffic, mostly from the hardcore that had heard of the game and wanted to see if it was the real deal and the members of the press invited by WBIE PR. That night, something amazing happened. The press that

GAME DATA



PUBLISHER Warner Brothers Interactive Entertainment

DEVELOPER 5TH Cell

PLATFORM Nintendo DS

RELEASE DATES
September 15th North America
September 30th Australia
October 9th Europe

NUMBER OF DEVELOPERS
12 programmers
11 artists
5 wordsmiths
4 designers
3 quality assurance
2 producers
1 sound and music guy

LENGTH OF DEVELOPMENT
14 months

FINAL NUMBER OF PROJECT SOURCE CONTROL COMMITS
5,187

heard rumors that SCRIBBLENAUTS was under consideration for "Best of Show" at several major press outlets.

The user-generated hype train pulled out of the station, and we took home three E3 2009 Best of Show awards. No handheld game in the history of video games had ever won Best of Show, and when IGN, GameSpot and GameSpy all separately announced our title as the winner, we were blown away. SCRIBBLENAUTS went on to take more than 40 E3 2009 awards from various publications.

At 5TH Cell, we always had big dreams, but they were tempered by the constant feedback from outside that said our expectations were unrealistic. An independent studio developing on the DS had no business getting press, and we should be happy and content that we had the opportunity to develop our own IP. Arriving back in Bellevue after E3, we felt the remaining weight of lingering doubt from the past year lifted from our shoulders.

No individual or company can be 100 percent right, 100 percent of the time. But all doubts that we're less qualified than others to shape our direction, build a brand, develop a AAA product without external direction, and succeed against the biggest players in the space have disappeared. We're excited about the holiday season and reading about all the unique ways people have completed the levels.

5TH Cell is hard at work on several new projects, but with every new SCRIBBLE-related blessing that comes our way, we remember that first external pitch, and think "Wow, indeed." 🙌

JOSEPH M. TRINGALI is the general manager of 5TH Cell and co-founded the company in 2003 along with Jeremiah Slaczkka, Brett Caird, and Marius Fahlbusch. He was the executive producer on SCRIBBLENAUTS and you should email any missing words to jtringali@gdmag.com.



Threaded Game Engine Helps Create New World In EA BioWare's *Dragon Age*: Origins*

By Geoff Koch

It's May 2004 at E3 in Los Angeles. Fresh on the heels of its success with the *Neverwinter Nights series and *Star Wars*: Knights of the Old Republic**, Edmonton, Canada-based BioWare offers a glimpse of a planned new role-playing game *Dragon Age*: Origins*. The cursory preview amounts to not much more than a "tease for PC RPG fandom," writes Sean Molloy for 1up.com.**

Then the company goes more or less mum on the subject for the next five years.

Fast forward to late 2009. After half-a-decade of work, the launch of *Dragon Age: Origins* is imminent. The North American release for Microsoft Windows* is slated for November 3, to be followed three days later in Europe. And the buzz about the game is building.

The New York Times' Seth Schiesel hailed *Dragon Age: Origins* as one of the 10 video games you can count on for this holiday season. BioWare, "masters of the role-playing genre, are going back to their roots with this dark and sometimes sexy . . . epic," declared the Toronto Sun's Steve Tilley, who named it one of eight must-have games for 2009.

One reason for the five-year development process was the company's own must-have technical



criteria for the game—that it take maximum advantage of multi-core chip architectures. Though technologies such as Intel® Hyper-Threading Technology had been around since 2002, when *Dragon Age: Origins* was announced in 2004 it was still mostly a single-core world. By the end of that

year, profound change was in the air. In a December 20 article, Associated Press technology writer Matthew Fordahl explained the momentous shift this way:

"For decades, computer performance has been driven largely by the increasing numbers of ever-smaller transistors squeezed into the machines' silicon brains. With each generation, speeds jumped and prices dropped. Though the tiny switches built in silicon are the heart of the digital revolution, they can't shrink forever. And in recent years, chip companies have struggled to keep a lid on power and heat—the result of some transistor components getting as thin as a few atoms across. Now, the world's leading semiconductor companies have unveiled a remarkably similar strategy for working around the problem: In 2005, microprocessors sold for personal computers will sprout what amounts to two heads each. Instead of building processors with a single core to handle calculations, designers will place two or more computing engines on a single chip. They won't run as fast as single-engine models, but they won't require as much power, either, and will be able to handle more work at once."

By the middle of 2005, Intel was shipping multi-core for desktops, laptops, and servers. And the cognoscenti at BioWare began to realize their game would be released onto a very different platform than most PC gamers had used up until then.

Moving to Embrace Potential Of Multi-Core

"Multi-core, multi-threaded development, along with the emergence of HD graphics, was the foundation of the huge technology transition that happened within *Dragon Age: Origins* development," said Mark Darrah, BioWare's executive producer of the *Dragon Age* franchise. "It actually sent the team back to the drawing board at least once, asking themselves, 'What's the right way to take advantage of this shift?' The team was clear that they weren't



going to build a multi-threaded platform just for the sake of it. They wanted real, tangible gameplay benefits for our players. So, the programming staff actually worked with our designers to explain the technology and then to collaborate on a plan of how best to use it. The result is several aspects of the gameplay that are enhanced on multi-core systems."

Here's another all-too-familiar result of coding for multi-core chips: headaches. That's because embracing parallelism is flat out difficult. Often a development team applies a Herculean effort to, for instance, adding gobs of synchronization primitives to prevent multiple threads from accessing data at the same time. Then, without fail, many of those threads spend inordinate amounts of time idle and waiting for other threads to execute. A meager payoff for a huge amount of trouble is unfortunately not uncommon.

Given the inexorable push toward more multi-processor, multi-core computers (in February 2007, Intel demonstrated an 80-core processor produced as a research project at the International Solid State Circuits Conference in San Francisco), game developers

may eventually need to toss out their age-old reliance on object-oriented programming (OOP) and begin focusing from day one on data-oriented design. This means grokking the various types of data; how and where they are stored in memory; and how input data are read, processed, and turned into output data at interactive rates during gameplay.

“Step back for a minute and think of the last game you worked on,” wrote Noel Llopis in the September 2009 issue of *Game Developer* magazine. “How many places in the code did you have only one of something? One enemy? One vehicle? One pathfinding node? One bullet? One particle? Never! Where there’s one, there are many. OOP ignores that and deals with each object in isolation. Instead, we can make things easy for us and for the hardware and organize our data to deal with the common case of having many items of the same type.”

Of course, exhortations such as Llopis’ run smack into the realities of deadlines, constrained budgets, and the difficulty of simultaneously retraining while working on a looming release. BioWare’s approach to multi-core can be summed up with three bits of advice: iterate toward success with the entire team of developers and artists, lean on Intel, both for its development tools and its engineering expertise, and always strive for simplicity.

Darrah is adamant that one of BioWare’s most important decisions was to insist that the designers and artists work to understand some of the pitfalls and challenges of multi-core development, and not just in a cursory way. This sort of mutual understanding helped everyone pull toward the same set of realistic goals to benefit gamers, such as providing more creatures on the screen or achieving smoother load transitions.

“Designers had to learn too,” said Darrah. “With a game design as broad and deep as *Dragon Age: Origins*, it’s easy to fall into the traps associated with multi-core development.”

When it comes to encouraging its designers and programmers to collaborate, BioWare does much more to insist that everyone speak the same language about new chip architectures. Four years ago, the company implemented Scrum methodology, an iterative incremental framework for managing

complex work that is commonly used with agile software development. The hallmark of Scrum is a development cycle of short two- to four-week sprints during which a cross-functional team works to finish a particular product increment.

“We assign our Scrum teams by system or functional area,” said Aaryn Flynn, senior director of development operations at BioWare, responsible for overall staffing of the projects in development at the company. “We might have a team or two working on storytelling aspects, a team working on creatures, one working on the combat system, and another working on the GUIs. Overall we try to keep

the teams as cross-functional as possible, just to keep the collaboration going.”

Dragon Age: Origins will be BioWare’s first product of Scrum development, which is also being used in its work on *Mass Effect* 2*, a title developed alongside *Dragon Age: Origins* at BioWare’s Edmonton studio, with support from BioWare’s new team in Montreal. For all the advantages of agile methodology, Flynn is quick to point out that BioWare’s chimeric teams benefited from more than just a faster, smarter approach. They also received various kinds of support from Intel.

“Multi-core, multi-threaded development, along with the emergence of HD graphics, was the foundation of the huge technology transition that happened within *Dragon Age: Origins* development.”

—Mark Darrah, BioWare’s Executive Producer of the *Dragon Age* franchise

At the top of the list was Intel® Thread Checker, an analysis tool that pinpoints hard-to-find threading errors, such as data races and deadlocks in 32-bit and 64-bit applications. Flynn said that throughout much of the development cycle, two BioWare programmers would daily “fire up Thread Checker and look at stuff.” Intel® VTune™ Performance Analyzer was also used regularly for overall optimization.

“I think it’s really important that along the way you use tools that are available to make sure that what you’re attempting to do is actually being done, that you’re actually getting the advantages that you thought you would from multi-core development,” added Darrah.



Intel also provided BioWare with development hardware, made several visits to the Edmonton studio, and otherwise made its engineers available for technical discussions with BioWare programmers.

Revvng Up a New Game Engine

The most prominent multi-core payoff of this collaboration is the heart of the game, the Eclipse Engine*. (For now the company is sticking with the name, despite the existence of the Eclipse open source community and its eponymous development platform.) *Dragon Age: Origins'* multi-threaded engine uses a job dispatching scheme that takes core engine tasks, such as AI actions, physics calls, and so on, and deploys them to the available thread.

Building the Dragon Age Lore

“Run, Maric! And run he did. His mother’s dying words whipped him into action. The image of her grisly murder still burning in his mind, Maric reeled and plunged into the trees at the edge of the clearing. Ignoring the clawing branches that scraped at his face and clung to his cloak, he blindly forced his way into the foliage.”

So begins the novel *Dragon Age: The Stolen Throne*, released in March 2009. The book, written by the game’s lead writer, David Gaider, is part of BioWare’s effort to create the lore and characters that it hopes will capture the imagination of RPG players when *Dragon Age: Origins* is released in November, 2009. The Stolen Throne is set 30 years before the events of the game and tells the back-story of several important characters. Here’s the product description from Amazon.com:

“After his mother, the beloved Rebel Queen, is betrayed and murdered by her own faithless lords, young Maric becomes the leader of a rebel army attempting to free his nation from the control of a foreign tyrant. His countrymen live in fear; his commanders consider him untested; and his only allies are Loghain, a brash young outlaw who saved his life, and Rowan, the beautiful warrior maiden promised to him since birth. Surrounded by spies and traitors, Maric must find a way to not only survive but achieve his ultimate destiny: Ferelden’s freedom and the return of his line to the stolen throne.”

Perhaps more interesting than the synopsis, at least for those wondering about BioWare’s ability to create a compelling story and setting for the game, is the steady trickle of positive Amazon.com reviews for the book.

“I was reading it more to ‘get in the mood’ for the game than anything, and I had very low expectations, to be honest,” wrote Peter A. Smith in a four-star March 21 review. “And I was blown away.”

“Books associated with games usually are not great literature,” added Sarah Jacobsen 11 days later, who went on to compare Gaider with such Science Fiction/Fantasy lions as Terry Goodkind and Robert Jordan. “This book is truly an exception.”

The release date for the second book in the series, *Dragon Age: The Calling*, has yet to be confirmed. ■

“We might have a team or two working on storytelling aspects, a team working on creatures, one working on the combat system, and another working on the GUIs. Overall we try to keep the teams as cross-functional as possible, just to keep the collaboration going.”

—Aaryn Flynn, Senior Director of Development Operations at BioWare

The engine was designed intentionally to be generic and easy to use so that BioWare programmers could iterate over the development cycle to move more and more of the engine functionality to dispatched jobs.

For now, the engine will be the foundation for extensive post-release content plans for *Dragon Age: Origins* and will act as a platform for the hoped-for avalanche of user-generated content that will flow from the planned Dragon Age Toolset. Eventually, Flynn and Darrah say, the engine may be used not only for future Dragon Age titles, but also by other Electronic Arts studios.

All the work on parallelization appears to have paid off. According to an Intel analysis, *Dragon Age: Origins* showed a 20 percent performance jump on Intel’s next-generation Kings Creek desktop platform supporting four cores and eight threads.

“Not bad considering we did all the work before that hardware was even available!” said Darrah, who added that BioWare reaped real benefits from profiling done by Intel of several earlier builds of the Eclipse Engine.

Of course, creating a game means more than writing lines of code. Intel’s Steve Pitzel likes to point out that in fact the majority of staff at most game studios don’t write code at all, and instead focus on storytelling, from writing narrative to illustrating characters and scenes. And in the case of *Dragon*

Age: Origins, the storytelling challenge was paramount. That’s because unlike BioWare’s success with titles such as *Star Wars*: Knights of the Old Republic** and *Sonic Chronicles*: The Dark Brotherhood**, *Dragon Age: Origins* is set in a brand new world of the company’s own creation.

“When you’re working with licensed intellectual property (IP) you actually have an established set of lore and story and understanding of the universe,” said Darrah. “When you’re building your own IP, you have to build all that lore, all that back story yourself in order to create a compelling vision, and then you have to use that to fill in the world that you tell the story within. So you have to build a world as well as build a game.”

Forget the Dark Winter Weather; BioWare’s Future is Bright

BioWare seems well positioned to build on its recent success. For starters, the company is now backed by the global marketing reach of EA, the world’s leading independent publisher of video games. Next, with a growing library of titles, including the enthusiastically received *Star Wars: Knights of the Old Republic*, BioWare has a won a sizable cadre of rabid fans. However, it’s a third point that may figure most prominently in the company’s future.

We hear repeatedly that the twenty-first century will be defined by globe-trotting knowledge workers. Given that talented software developers can live in



"We have people who joined us from the U.K., from Australia, from Europe, from Mexico," said Flynn. "It starts off with one person joining us and then they recommend a friend, and that friend is a good person so we hire him or her, and then on and on. We're very proud of the fact we're a very cosmopolitan, very multi-cultural studio right here in Edmonton." ■

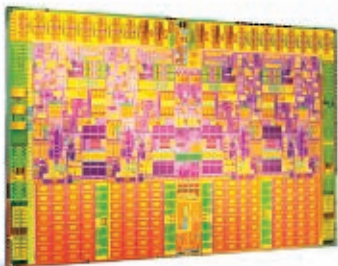
many places in the world where the month of January is marked by conditions more hospitable than Edmonton's just under eight hours of average daily sunlight and average daily temperatures of -9 degrees Celsius, it's noteworthy that BioWare is winning rave reviews as an employer. The company was named a Top 100 employer in Canada, and is situated in arguably Canada's most business-friendly province—Alberta. And already, said Flynn, more than a quarter of BioWare's staff is from somewhere besides Canada or the United States.



ABOUT THE AUTHOR

Geoff Koch is a contributing writer for the magazine. He last wrote about composer Justin Lassen's *Synaesthesia* project in *Intel® Visual Adrenaline* magazine, Issue No. 5.

Stay current on the latest software developments in Visual Computing at:
<http://visualadrenaline.intel.com>



Intel does not make any representations or warranties whatsoever regarding quality, reliability, functionality, or compatibility of third-party vendors and their devices. All products, dates, and plans are based on current expectations and subject to change without notice. Intel, Core Inside, the Intel logo, Intel Core, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

*Other names and brands may be claimed as the property of others. | Copyright © 2009. Intel Corporation. All rights reserved.



LUXOLOGY MODO 401

REVIEW BY ZOLTAN ERDOKOVY

MODO WAS DEVELOPED BY several industry veterans who played a pivotal role on the creation and shaping of LightWave 3D. In 2001 they left NewTek and started Luxology to produce a 3D application from the ground up.

They began with the creation of Nexus, a software architecture that serves as the foundation for modo. This framework allowed Luxology to develop a clean, generalized, and flexible application. It doesn't carry the legacy or backward compatibility issues that often bloat older packages as new features are duct-taped onto an aging core.

The downside of being a relatively young software tool (the first version was released in 2004) is the lack of certain features, like physics simulation, particle systems, or character animation.

USER INTERFACE

» Modo's modern software design is really apparent in the tool's user interface—it's very flexible and can be customized in great depth. From keyboard shortcuts to mouse navigation, menus, forms, popups, and pie menus—the whole layout can be changed and adapted to almost any workflow.

Modo can be a focused modeling application, a sculpting-painting solution, an animation and rendering tool, or all of the above, as you can switch between layouts with one click.

MODELING TOOLS

» Modo started life as a "modeling only" package, and because this area has been continually updated and polished since then, it's probably the most refined part of modo. Creating

both high definition and low poly models is where this application really shines.

A generalized approach makes the modeling tools simple and very intuitive. For any task, three elements are combined: a base tool, an action center, and a falloff. For example the Shear tool is just a move base tool combined with a linear falloff. Swirl is a rotation with a radial falloff.

These elements can be freely mixed, allowing you to create your own tools, or customize existing ones. Once you've learned that *vertex map* falloff adjusts the tool's strength based on the selected vertex map, you can expect that it will work the same way with every tool—from transformation and polygon reduction to sculpting tools. After understanding the basic principles, modo becomes transparent, and you can start doing what you want without really thinking about it.

While polygonal models and spline patches are supported, subdivision surfaces are where the extensive sculpting tool set works best. Although modo is no ZBrush, common tasks like wrinkled cloth, elaborate landscapes, or even human figures can be done in high quality. Mesh instances and replicated geometries are used to add repeating details to scenes such as populating a garden with plants or placing rivets on the hull of a battleship.

High-resolution geometry creation is well covered by modo and the same can be said when it comes to low poly modeling. The quad drawing pen tool, combined with a background constraint provides the same capability as the Retopology tool in ZBrush—you can quickly draw quad strips over a high-resolution mesh.

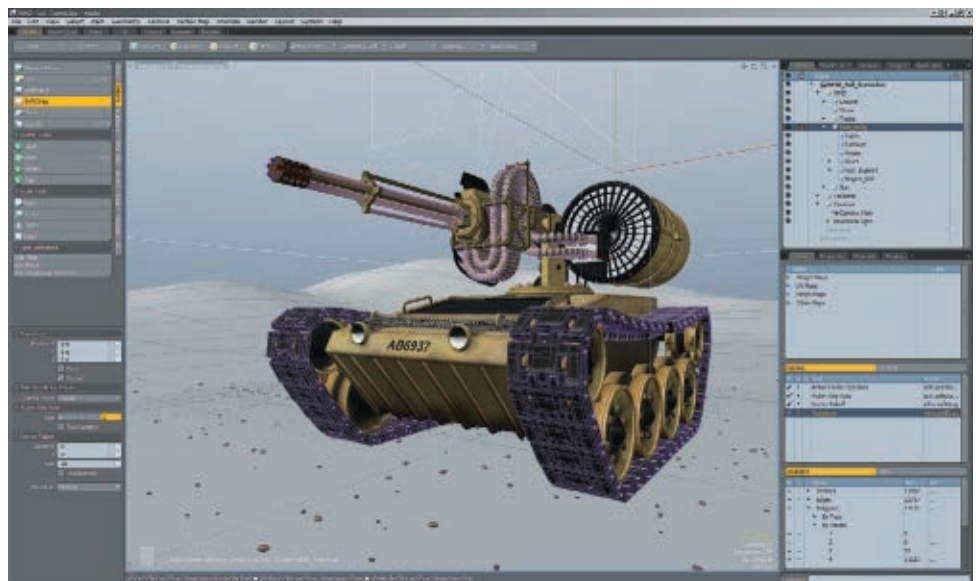
The polygon reduction tool is another great way to create a low

poly version of a geometry—it creates a tidy topology, with almost no artifacts to clean up manually. Combined with the *vertex map* falloff, you can control where and how much the model should be simplified.

When UV editing, the basic tool functions such as falloff and action center work the same as in modeling, and the specialized UV tools do their job reliably. The UV Unwrap unfolds the surface, Relax removes mapping distortions, and UV Pack efficiently arranges the polygon islands.

SHADERS, RENDERING, AND BAKING

» Modo uses a layer-based shader system, called Shader Tree. While it's more advanced than LightWave's old layered system, it's not nearly as powerful as node based solutions such as Maya or the newest version of LightWave. Quick for simple



Luxology MODO 401

✂ **STATS**

Luxology, LLC
2525 East Charleston Road
Suite 104
Mountain View, CA 94043
www.luxology.com

✂ **PRICE**

\$995.00
(Floating License \$1,195.00)

✂ **SYSTEM REQUIREMENTS**

Win XP / Vista (32 and 64 bit)
2 GB available hard disk
space, 2 GB of RAM, Pentium
4 processor or better.

Mac OS X 10.4 or later,
Macintosh G5 or Intel
processor, 2 GB available hard
disk space, 2 GB of RAM.

Monitor resolution of 1024 x
768 or greater, Graphics card
with OpenGL support.

A modo license allows the
software to run on either a
Mac or a Windows computer
and can be switched back
and forth between systems
as needed.

In addition, modo can run
in rendering ("slave") mode
on up to 50 machines on a
network to render animation
sequences or extremely high-
resolution graphic images.

✂ **PROS**

- 1 Highly adaptable user interface.
- 2 Excellent modeling and UV editing tools.
- 3 Enthusiastic and productive community, dedicated developer support.

✂ **CONS**

- 1 Shader tree makes complex shader setups difficult to manage.
- 2 Sub-par normal baking capabilities.
- 3 Young software, many features are not yet implemented (e.g. character animation support or baking exotic map types).

setups with only a few textures, Shader Tree becomes a real pain as things get more complex. Without consequent layer naming and well-planned grouping, the tree quickly grows unmanageable. While the technical capabilities are just fine, the problem is how a complex shader structure is presented to the user.

The advanced OpenGL view is far from 3ds Max 2010's HLSL enabled viewport. Although it shows a few effects and multiple UV sets, anything more than a very simple shader tree will confuse the perspective view. You have to fiddle with layers, hiding some of them and adding new ones in a certain order, before the viewport starts to show the texture you're looking for.

Complex material setups are best viewed in the preview render window—it gives a very fast and accurate render of the scene. It has no problem handling millions of (instanced) polygons with all the bells and whistles—global illumination, caustics, blurry refraction, Micropoly Displacement, etc. It can show separate buffers, which is very helpful when debugging the shader tree. The full render is fast, relatively easy to set up, and capable of network rendering.

Generalization also enhances usability when dealing with render buffers—if a buffer type is in modo's list of supported buffers (which is quite extensive) then it can be previewed, saved (individually or as a multilayer PSD file), and baked.

At first glance this would make baking simple and easy. Unfortunately there are several problems that can make baking a frustrating experience. While layer baking (collapsing several layers onto one texture) works as expected, object baking (transferring the look of a highpoly model to a lowpoly one) has several shortcomings.

When rendering tangent space normal maps, only geometry data is used—normal and bump textures on the source object are

discarded. Only displacement maps show up, because Micropoly Displacement generates new geometry during rendering. The downside is that the object will be heavily tessellated, so render speed and memory consumption becomes a problem, especially on 32 bit systems.

Another inconvenience is that modo doesn't use baking cages to define the start and end points of the traces. Their length is constant, which can cause problems in tight corners or when the highpoly and lowpoly topologies differ significantly. Even worse, the baking distance is asked before every single bake. Because its value is not stored in the scene or item, you must remember it each time. Hopefully a community-made script will address this issue. Special map types like "cavity-," "thickness-," or "radiosity normal" maps are not yet supported.

When it comes to shader setups and baking, modo is a mixed bag. Since modo exports and imports the most common file formats (Collada, FBX, Obj, Lwo), for some it might be easier to use modo only as a modeler and finish the job in another package. Yet, for others, staying in one application during the whole workflow might be worth jumping through hoops. It really depends on you, your task, and other 3D software you may have.

ANIMATION

» At this point modo has no bone support, so it's better suited for mechanical animations. There is no support for named animation sequence management (like ActorX for Maya and Max, used in Unreal circles). Still, with some effort, a decent solution could probably be implemented using modo's scripting capabilities.

SCRIPTING

» Modo can be scripted using Lua, Python, or Perl. The system provides access to most parts of modo and almost everything done through the UI can also be done

from scripts. There are a huge number of scripts made by the community, which solve a wide range of tasks and also provide material to learn from.

SCENE MANAGEMENT

» The item and hierarchy management is solid. Custom data can be stored in items, and along with item instances and external references, a few custom scripts could potentially turn modo into a basic level editor. The appearance of "locators" ("dummy" or "null" objects) can be changed to represent different entities or define zones, and lines between linked items that express relations, are some of the features possible.

SUPPORT

» An outstanding aspect of modo is its support. The community is very active and helpful. The script database, the material and scene preset library are goldmines of information. The official forum is a great place to ask questions or find tutorials, discuss problems, or find inspiring artwork. The developers are in touch with the users—they provide tips, help to find workarounds for existing bugs, and reply to feature requests. The official website has an extensive collection of video tutorials, both free and commercial. Also, there is a form for submitting bug reports where you can check their status as well.

A WORKHORSE

» Most of the time using modo is fun. While it could stand to improve in the shading, baking and OpenGL departments, version 401 has no show-stopping issues for a game developer. It's a workhorse and can easily become an invaluable part of any developer's toolbox.

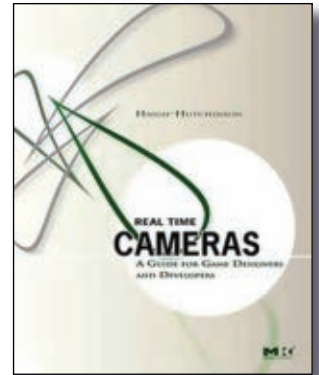
ZOLTAN ERDOKOVY is a freelancer technical artist. In the past decade, he has created artwork, game levels, special effects and support scripts for several projects, most of which were based on the Unreal engine. Email him at zerdokovy@gdmag.com.



REAL TIME CAMERAS

A Guide for Game Designers and Developers By Mark Haigh-Hutchinson

REVIEWED BY BIJAN FORUTANPOUR



A FEW YEARS AGO I HAD THE TASK OF writing a camera system for a game. The design spec was simple: Look at two given games for reference, and implement something similar, an approximate hybrid of the two. Having never written a camera system before, it was an interesting and challenging task. The camera behaved differently in different situations, so it quickly became obvious that the problem had to be broken down into smaller pieces and categories. Along the way, many things went wrong and many things right, and that is when I developed an appreciation for the art and science of camera programming. When the opportunity arose to write a review for *Real-Time Cameras, A Guide For Game Designers and Developers* I jumped, because this book wasn't around when I needed it. Furthermore, I wanted to see whether what I had figured out on my own was close to the "correct" solution(s) proposed by the book.

The secret sauce to creating a good camera system is to realize that if it is done well, no one will notice. Conversely, if the design or implementation is poor, everyone will notice. A game camera needs to feel natural without jarring the player out of the world they are immersed in. This is not an easy task, and one where *Real-Time Cameras* leads the way. The book is organized into three parts, Core Concepts, Design Principles, and Camera Engineering.

CORE CONCEPTS

» The Core Concepts section lays a basic foundation for the reader, covering a typical game update

loop where input is received from the controller, game objects are updated, and messaging and events are processed. Finally the camera is updated, and the final render performed. Basic camera fundamentals, behaviors, and terminology are then covered in the following chapter, followed by a quick lesson in cinematography. The Core Concepts section of the book is a good starting point for beginners new to game development and camera design. For experienced programmers unfamiliar with filmmaking concepts, it is also worth a quick read to help communicate with the artists and designers on the team.

DESIGN PRINCIPLES

» This section of the book is oriented toward game camera designers and the information presented in these chapters is focused on theory and concepts. Because game development teams will have different game engines as well as different art and design production pipelines the book takes a general approach. It covers the different types of scrolling 2D camera behaviors such as continuous, character-relative, screen-relative, and directional lag, among others. Then 3D cameras are discussed, covering different types of projections, followed by a good list of dos and don'ts when designing camera systems.

The next chapter goes through an exhaustive list of game genres and the types of cameras appropriate for each genre. The section on camera design is concluded with a chapter on camera scripting as a method for describing camera behaviors. The

topics of script objects and object relationships are covered, as well as camera hints, pathing, trigger volumes, and script debugging.

CAMERA ENGINEERING

» Camera Engineering is the real strength of the book, and the largest section. Different problems and requirements are presented, along with proposed solutions. It begins by covering different methods for computing desired camera position and orientations. The list of behavior types is complete; stationary cameras, slaved cameras, path constrained, surface and volume constrained, framing, object-framing relative, and axis rotational cameras are all explained. Camera orientation methods are also exhaustively presented: constant,

object tracking, look-at offset, locked, object position prediction, object framing, idle wandering, and automated orientation controls.

The next two chapters continue the discussion and present the more difficult challenges, covering the topics of navigation, occlusion, motion, and collision. The lesson to take away from these chapters is that there are many ways to solve a given problem. The more simple and successful solutions involve using spline-based pathing as a way to guide the camera through difficult terrain situations. Path splines may be built apriori by the game designers, or they may built by the camera system on the fly, perhaps using preexisting hints, objects, and volumes, among others. The best way to solve a tricky problem is to avoid it. The alternative is to simplify the problem and always have a fail-safe escape plan when the worst-case scenario actually occurs. Typically, this would be doing a jump cut of the camera to a new location, or simply making problem objects transparent or semitransparent. There is much more advice and trickery up the author's sleeve, which makes these chapters very valuable.

Next comes a chapter on the mathematics used in the algorithms presented, focusing mostly on different spline types commonly used in pathing. Different types of interpolation are discussed as well, although sometimes at a cursory level. For instance, quaternions are mentioned, though not described in detail.

The book concludes with a chapter on a proposed organization for a camera system, namely the

REAL TIME CAMERAS
A GUIDE FOR GAME DESIGNERS AND DEVELOPERS
 BY MARK HAIGH-HUTCHINSON

Morgan Kaufmann
 3251 Riverport Ln
 Maryland Heights, MO 63043

<http://realtimcameras.com>
<http://realtimcameras.googlecode.com/svn>

✂ PRICE \$59.95

✂ PUBLISHED April 14, 2009

✂ PAGES 544

✂ ISBN ISBN-10: 0123116341

different types of managers as they might be mapped into C++ classes.

PROS AND CONS

» *Real-Time Cameras* offers a wealth of information and is very broad in scope. Its goal is to teach beginners theory and terminology, as well as give game designers some design guidelines and ideas for what is possible, while at the same time helping programmers with implementation details. While a very noble goal, this makes the book slightly frustrating to read cover to cover, because the same concepts are repeated multiple times, depending on the target audience. To use programming terminology, the book takes a breadth-first traversal of the material, making a first pass at individual subjects (e.g. camera POVs, navigation,

occlusion, etc.) in order to keep the discussion brief and in context, and then revisits each topic again a few chapters later to repeat as well as go into further detail. This approach has advantages and disadvantages.

The solution for the reader is to simply begin reading at the section of the book targeted toward their particular interests. As a programmer, I was looking for a depth-first approach to the material, a kind of *Game Camera Gems* that quickly cut to the details of algorithm and implementations. The best approach in this case would be to just begin reading at Chapter 7, where the Camera Engineering section of the book begins. Similarly, a game designer looking to fill his or her playbook with camera design strategies should focus on Chapters 4, 5, and 6.

Real-Time Cameras does have some brief code snippets interspersed throughout. There is a website that accompanies the book which was stated to contain the implementation of a simple camera system. Unfortunately, the Microsoft Visual Studio project files do not compile because they reference a significant amount of missing source code. The book also states that the website provides movie examples and analysis of camera design in existing games, but those do not appear on the website either. We hope the website will be completed in the near future, as it holds the promise of becoming a valuable resource for developers.

IN FOCUS

» *Real-Time Cameras* contains a metric ton of real world experience

gathered through many years of dedicated work in game camera development by the author. It leaves no stone unturned, giving an exhaustive list of problems, solutions, and gotchas to watch out for in the solutions. As I mentioned at the beginning of this article, I was able to implement a real-time camera system for a game without any prior camera programming experience, but there was a lot of trial and error involved. Having this book as a guide would have helped save a lot of time, and is well worth the read.

BIJAN FORUTANPOUR is a senior graphics programmer at Sony Online Entertainment in San Diego with 16 years experience in the visual effects and games industries. He is also the author of *Enzo 3D paint for Photoshop* (www.enzo3d.com). Email him at bforutanpour@gdmag.com.

UNITED STATES POSTAL SERVICE Statement of Ownership, Management, and Circulation

1. Publication Title: Game Developer. 2. Publication No.: 1073-922X. 3. Filing Date: September 11, 2009. 4. Issue Frequency: Monthly with a combined June/July issue. 5. Number of Issues Published Annually: 11. 6. Annual Subscription Price: \$49.95. 7. Complete Mailing Address of Known Office of Publication (Not Printer): Think Services, 600 Harrison St., San Francisco, CA 94107. Contact Person: Roy Beagley. Telephone: 203-775-9465. 8. Complete Mailing Address of Headquarters or General Business Office of Publisher (Not Printer): Think Services, 600 Harrison St., San Francisco, CA 94107. 9. Full Names and Complete Mailing Addresses of Publisher, Editor, and Managing Editor: Publisher: Simon Carless, Think Services, 600 Harrison St., San Francisco, CA 94107; Editor: Brandon Sheffield, Think Services, 600 Harrison St., San Francisco, CA 94107; Managing Editor: None. 10. Owner: United Business Media LLC, 600 Community Drive, Manhasset, NY 11030-3875, an indirect, wholly owned subsidiary of United Business Media LLC, Ludgate House, 245 Blackfriars Rd., London, SE1 9UY, U.K. 11. Known Bondholders, Mortgagees, and Other Security Holders Owning or Holding 1 Percent or More of Total Amount of Bonds, Mortgages, or Other Securities: None. 12. Tax Status: Has Not Changed During Preceding 12 Months. 13. Publication Title: Game Developer. 14. Issue Date for Circulation Data Below: September 2009.

| 15. Extent and Nature of Circulation: | Average No. Copies Each Issue During Preceding 12 Months | No. Copies of Single Issue Published Nearest to Filing Date |
|--|--|---|
| a. Total No. Copies (Net Press Run) | 27,436 | 24,000 |
| b. Paid and/or Requested Circulation | | |
| (1) Outside County Paid/Requested Mail Subscriptions Stated on Form 3541. | 19,542 | 15,996 |
| (2) In-County Paid/Requested Mail Subscriptions Stated on PS Form 3541 | 0 | 0 |
| (3) Sales Through Dealers and Carriers, Street Vendors, Counter Sales, and Other Paid or Requested Distribution Outside USPS | 1,966 | 1,782 |
| (4) Requested Copies Distributed by Other Mail Classes Through the USPS | 0 | 0 |
| c. Total Paid and/or Requested Circulation [Sum of 15b. (1), (2), (3), and (4)]: | 21,508 | 17,778 |
| d. Nonrequested Distribution (By Mail and Outside the Mail) | | |
| (1) Outside County Nonrequested Copies Stated on PS Form 3541 | 2,906 | 2,776 |
| (2) In-County Nonrequested Copies Stated on PS Form 3541 | 0 | 0 |
| (3) Nonrequested Copies Distributed Through the USPS by Other Classes of Mail | 0 | 0 |
| (4) Nonrequested Copies Distributed Outside the Mail (Pickup Stands, Trade Shows, Showrooms, and Other Sources) | 2,505 | 3,000 |
| e. Total Nonrequested Distribution | 5,411 | 5,776 |
| f. Total Distribution (Sum of 15c and 15e) | 26,919 | 23,554 |
| g. Copies Not Distributed | 517 | 446 |
| h. Total (Sum of 15g and 15h) | 27,436 | 24,000 |
| i. Percent Paid and/or Requested Circulation (15c Divided by 15f Times 100) | 79.90% | 75.48% |

16. Publication of Statement of Ownership: This Statement of Ownership will be printed in the November 2009 issue of this publication. 17. Signature and Title of Editor, Publisher, Business Manager, or Owner (signed): Cary Mitsuyoshi, Date: September 15, 2009.



JUMP TO IT

THE MATHEMATICS BEHIND EFFECTIVE JUMP MOTION IN GAMES

IN DAMION SCHUBERT'S ARTICLE, "Resonance" (*Game Developer*, September 2009), he talks about how jumping is a vitally important part of the initial feel of many games, and how poor jumping can immediately prejudice a player. Yet jumping is often not given appropriate priority in development. Great attention to detail can pay off big. In this article I give an in-depth look at the coding behind just one small yet vital aspect of jumping in video games: Jump height. We'll discuss how it works, how it could go wrong, and how to fix it.

JUST JUMP?

» Jumping in video games seems very simple. You just press a button, and the player character jumps. He goes up, he comes down. All very straightforward, right?

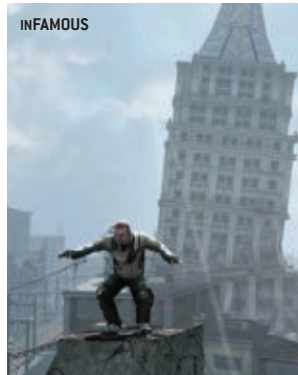
Well, yes, it is actually very simple, and from a programmer's point of view, it's quite possible to program it in a very simple manner. But, as you may have noticed while playing games, the "feel" of a jump seems to vary wildly from

game to game. The way a jump has been implemented can have huge effects both on control feel, and on actual gameplay. Various factors can affect the height of a jump, and hence the distance jumped. Inconsistencies here can have a negative effect on the feel of the game, as the player can sometimes make a jump, and sometimes not, seemingly arbitrarily.

WHEN I SAY JUMP...

» How high is a jump? Here we are talking about the standard video game jump, the type you would use to jump up onto a wall or a crate. This is a "vertical jump", and in real life is more akin to the jumps performed by basketball players than high jumpers. In the real world a typical vertical jump is up to three feet, and takes about 0.8 seconds.

In video games, of course, gravity is not always the same as the real world, and is typically tweaked to make the game play nicely, rather than because of any pretense at reality. Characters in video games can often jump their



own height, and their standing jumps can range in duration anywhere from 0.5 to 2.0 seconds. For the sake of this discussion assume we are talking about a jump height of about six feet, and a jump time of a bit under a second. That's about the same as the standing jump of the main character in INFAMOUS on the PS3. This means that the gravity used during the jump is going to be set a bit higher than the real world, and our character is somewhat superhuman.

In any game involving jumping we are doing some kind of simulation of Newtonian mechanics. We have a position P , a velocity V , and constant gravity G . We also have a time step t , which is the duration of one frame's worth of physics simulation, and an initial jump speed s , which is the y -velocity of the character at the start of the jump (for simplicity here I'm using 2D, with y being up. It's basically the same for 3D).

Jumping can be implemented with two functions, `jump()`—which is called when the jump button is triggered, and the character is able to jump—and `integrate()`, which is called every iteration of the physics loop. A very simple version would look like this:

```
jump1() {
    V.y = s;
}

integrate() {
    P = P + V * t;
    V = V + G * t;
}
```

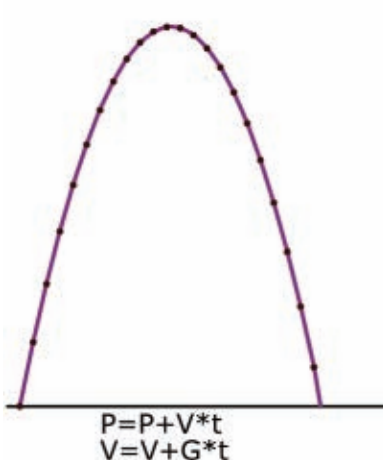


FIGURE 1 A normal jump at 30 Hz, using simple physics.

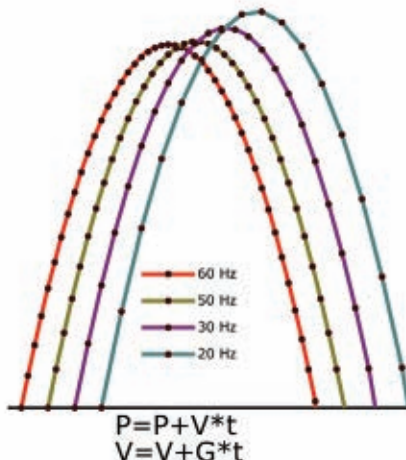


FIGURE 2 Simple physics at different time steps yield differing results.

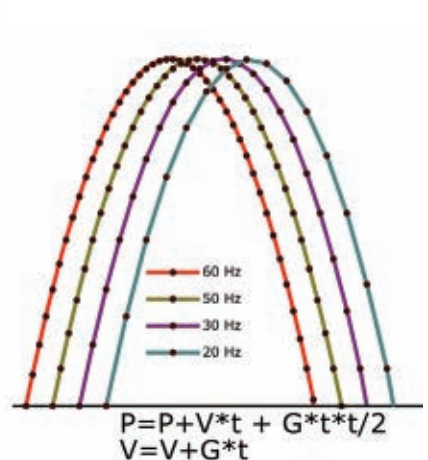


FIGURE 3 Correcting the equation of motion equalizes the jump heights.

FIGURE 4 A variable height jump is done by holding gravity at zero for the first few frames.

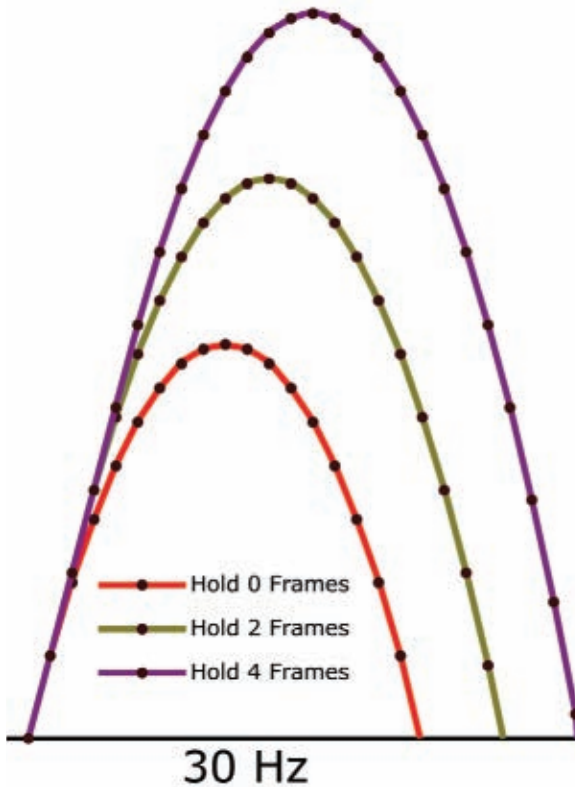
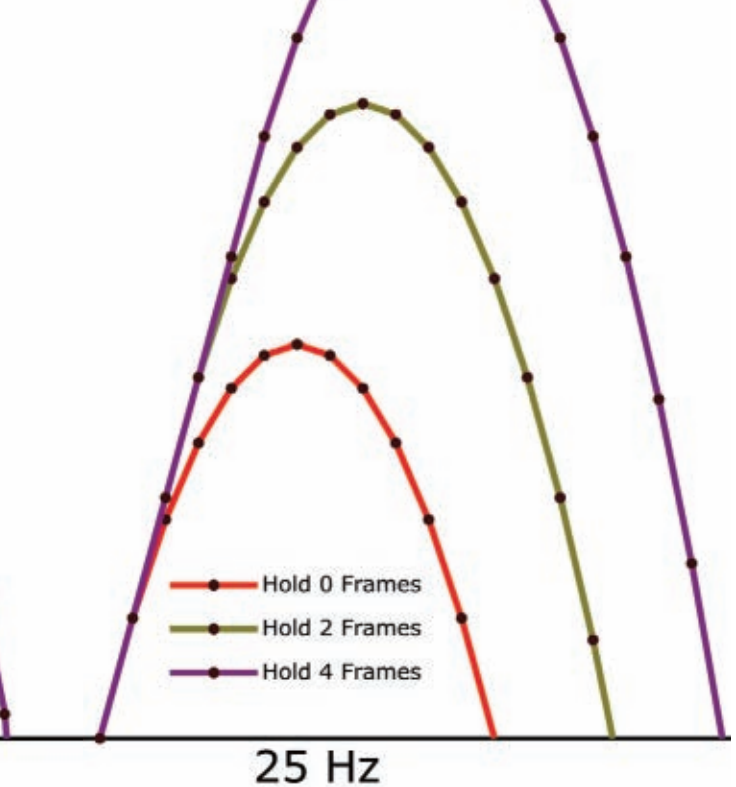


FIGURE 5 Holding for a frame count produces inaccuracies when frame length changes. We need to use time, not number of frames.



On the face of it, there is nothing wrong with this. Many games don't need much in the way of physics, and using these equations usually gives very acceptable results, especially for smaller games. Indeed, if you've got a fixed frame rate, you can even drop the t term, by adjusting your other constants so you can use $t=1$, and simplify to $P = P + V$ and $V = V + G$.

Either way, our simple physics model gives us a very nice jump. Figure 1 shows a 23 frame jump that results from this method. In these figures the vertical axis can be considered height and the horizontal axis can be considered as either time or the equivalent distance with a constant x velocity. As you would expect, we get a nice parabola. 23 frames is 0.77 seconds at 30Hz, so it's quite a reasonable representation of what happens in a video game when the character jumps.

If you jump from the same level of ground, then the height of any jump should theoretically always be the same at any given N time steps after the start of the jump. You can rely on consistent behavior, providing your time step is constant.

When might your physics time step (t) not be constant? There are three main situations:

- **When converting a game between NTSC (60Hz or 30Hz) and PAL (50 or 25Hz), for a console game. (There's also PAL60, but you usually have to support PAL50 as well).**
- **When running on a PC where the frame rate is tied to the amount of CPU and GPU processor power available, so less powerful PCs have a lower time step.**

- **When attempting to run mostly at a fixed frame rate, like 30Hz, but accepting occasional (or not so occasional) drops to 20Hz or even lower.**

You can, of course, try to always run at a fixed physics frame rate, regardless of the rendering frame rate. But this is not always possible, and can lead to other problems.

So what happens with a variable speed time step, using the simplest physics model? Figure 2 shows the exact same initial jump, with the same velocity and gravity, where all that is changed is the time step t . Four jumps are shown, 60, 50, 30, and 20Hz.

The obvious problem here is that the lower the frame rate, the higher the character jumps. The simple reason is fairly obvious when you look at the equations. Consider the difference between one step of $t=2$

and two steps of $t=1$. In the first case we have $P = P + V*2$, and in the second we have $P = P + V + (V+G)$, or $P = P + V*2 + G$. The longer a time step is, the less effect G has on the motion during that time step.

You've probably realized by now that we need to use the proper equations of motion, commonly written as $s=ut+\frac{1}{2}at^2$. Here we can express it in code as $P=P+V*t + G*t*t/2$. I'd encourage you, if you don't already know, to look into where this equation comes from. In particular to look into the Taylor series, integrators, and error orders. But you don't actually need to know the terminology or the math to see what is going on here. For a constant acceleration like gravity, you don't need anything more complex.

So, using this equation for updating the position we get the results shown in Figure 3. Note now the jumps are all the same height. Note also that every $1/10$ th of a



second (6 frames at 60Hz, 5 frames at 50Hz, etc.), the points on the curve line up. This demonstrates that the character is now traveling along the exact same curve, regardless of our time step.

NOT EXACTLY

» All good? Not quite, even though we are moving along the exact same curve, we can never actually be at the highest point in the curve unless a time step happens to fall exactly in the middle of the curve. With large time steps this is less likely, depending somewhat randomly on the initial jump speed. See the table of maximum heights (in feet) reached for simple and corrected motion (see Table 1).

You can see the expected huge variance of half a foot in the simple physics version. The difference between 60 and 50 Hz is a fairly small 0.05 feet, or about 0.6 inches. While this small difference is not going to be visually apparent, it can have a critical effect in gameplay. I encountered a bug based on this very effect in the first TONY HAWK'S PRO SKATER, where a designer had created a wall that was of such a height that you could jump over it in the PAL version, but not in the NTSC version. Testers also complained that one of the two versions seemed sometimes easier, or harder, or different, but were unable to explain how.

With the corrected physics, the problem is reduced, but still potentially there. The difference between PAL and NTSC is now just 0.002 feet, or 0.024 inches. This difference is tiny, but in the unforgiving binary world of collision detection this could make all the difference between a hit and a miss. Such a bug would probably

only arise as an occasional annoyance, perhaps more focused in one particular spot, where a particular jump sometimes works, and sometimes does not.

An even smaller difference occurs in the third column, labeled "500" feet. Here we are doing the exact same jump, but from a position in the world that is just 500

feet higher. Because the relative precision of floating point numbers (the difference between two close numbers) decreases as the numbers increase, the maximum height you get at 500 feet ($y=500$) will be different from the height at ground level ($y=0$).

Note that the maximum height reached is different here even at the same frame rate. So even if you run your physics with a fixed time step, you could still have these problems. You might take a piece of geometry from ground level and paste it into the game at a higher level, or just move it some distance away. In this new position you might find the results of a jump vary, so you now hit or miss something that you did not in the original position.

SOLUTIONS

» Problems like these are not always readily apparent. If the frame rate only drops occasionally,

then the missed jumps may be rare, yet still very annoying to the player. Glitches may occur where the player can leap to a position in the map, but only at certain frame rates. You have to educate your testers to report these kinds of "feel" problems appropriately.

The most important measure you can take is to be as consistent

The reason, of course, is that one frame at 25 Hz is 20 percent longer than one frame at 30Hz, so the jump is in virtual zero gravity for 20 percent total more time.

The solution here, and for other similar problems is to allow the hold for a maximum period of time, not a maximum number of frames. In addition, releasing

The most important measure you can take is to be as consistent as possible. Use the correct equations of motion. Test your game at any frame rate at which it might possibly run. Use a fixed physics time step if possible. Using double precision for world coordinates and performing collision in a local coordinate system can help prevent inconsistencies.

as possible. Use the correct equations of motion. Test your game at any frame rate at which it might possibly run. Use a fixed physics time step if possible. Using double precision for world coordinates and performing collision in a local coordinate system can help prevent inconsistencies. Consider using fixed point if your world layout suits it. If an exact maximum height for a jump is required, and you have variable frame rates, then use pre-calculated jump tables for each frame rate.

VARIABLE JUMP PROBLEMS

» Beware of anything that takes a fixed number of frames. For example, a common way to implement a variable height jump is to basically turn off gravity for the first N frames while the button is held down. That way the player can tap the button for a small jump, and press it slightly longer for a higher jump. This works great, and hardly changes the shape of the parabola. Figure 4 shows this in operation, with the button being held for 0, 2, or 4 frames, running on an NTSC system at 30 Hz.

When this is converted to PAL, and is running at 25 Hz, the picture changes—see Figure 5. The jump with zero frames of delay remains exactly the same, but the jumps with 2 and 4 frames are a lot higher.

the hold can happen at any time during that period, not just on 30 Hz frame boundaries. This introduces additional complications, as for a single frame of physics at the transition you have to calculate partial movement in zero gravity, and then partial movement under gravity. But it's pretty straightforward, and when it's done your 25Hz jumps will look just the same as your 30 Hz jumps.

Remember that just because a jump looks nice and consistent does not mean it actually is. To help your gameplay it is vitally important that you know exactly what is going on when your character jumps: how it starts, how it goes up, what happens at the top, and what happens when it lands. Consider adding visual tools to your code that let you zoom in on the exact path that your character took. Aim for having something like Figure 1, but visible in the game, and reflecting the precise movement your character took. The more detailed your visualization, the easier you will be able to spot and correct problems like these.

MICK WEST was a co-founder of Neversoft Entertainment. He's been in the game industry for over 20 years and currently works as a technical consultant. Email him at mwest@gdmag.com.

TABLE 1 Maximum heights for simple and corrected motion.

| | Simple | Corrected | 500 Feet |
|------|----------|-----------|----------|
| 60Hz | 6.029631 | 5.758031 | 5.758420 |
| 50Hz | 6.080000 | 5.760003 | 5.759679 |
| 30Hz | 6.301233 | 5.758021 | 5.757899 |
| 20Hz | 6.577778 | 5.755556 | 5.755556 |

AWARD-WINNING GAMES
AWARD-WINNING DEVELOPER

BURBANK

DURHAM

WHAT ARE YOU WAITING FOR?

INSOMNIAC
GAMES

www.insomniacgames.com/careers

facebook

Become a fan:
Insomniac Games



Follow us:
insomniacgames



MOST LIKELY TO SUCCEED

CAN SPLINE IK LIVE UP TO ITS POTENTIAL?

TECHNOLOGIES, LIKE HIGH SCHOOL kids, have unpredictable stories. You meet all sorts of new ones. Some are eager and full of potential. Others seem destined for a career in the CG equivalent of convenience store clerking. But, just like after graduation, not everything turns out the way you'd expect. Some new techniques go on to do great things, the sort of successes you follow enviously on Classmates.com. Others, though, end up like the punchline in those "SHE married HIM???" ads—they never quite live up to their bright promise.

Spline IK is a prime example of a tech that never lived up to its early hype. It demos well, and impresses everybody the first time they see it, but it rarely elbows its way into the forefront of day-to-day production. Instead, spline IK is relegated to a handful of specialist applications. Tails, tentacles and snakes are, after all, pretty common in the games business. Surely, that's enough work for a humble rig component?

In fact, spline IK can be useful for a lot of applications beyond merely the tentacular. You just need to put in a little extra work learning its quirks and capabilities,

so this month we're going to make a pitch for the rehabilitation of this oddball technology.

PROMISE RING

» The essence of spline IK is quite simple: You draw a curve and then glue a bone chain to it. This simple mechanism allows for fluid, sinuous motions that would be almost impossible to key by hand. For example, you can easily animate a slithering snake by drawing out a control curve that's much longer than the snake's skeleton and animating the offset of the bones along the spline. [See Figure 1]

Since the spline curve controls bones indirectly, spline IK is a very attractive option for controlling dense bone chains. The would-be squid rigger can control dozens of joints with a handful of position keys on the control vertices of the spline. Unfortunately, it's the way those position keys behave that turns a lot of animators off spline IK.

Unlike a bone chain, a spline has no fixed length. This means you can accidentally lengthen or shorten a spline when posing, and interpolations are often not what you'd want. Consider the example in

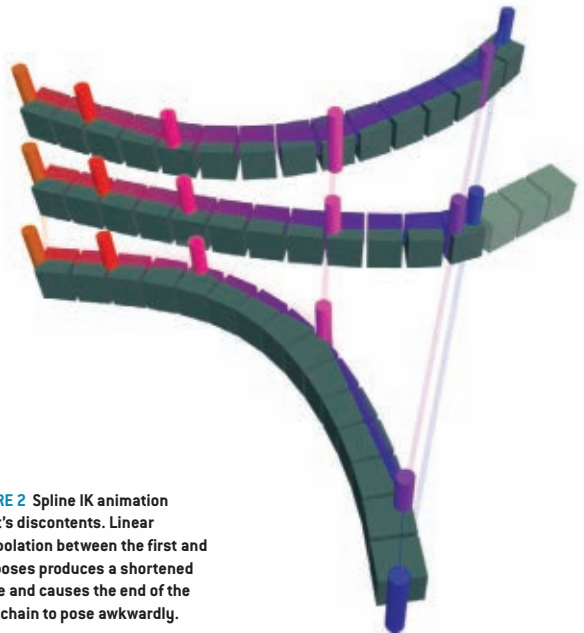


FIGURE 2 Spline IK animation and its discontents. Linear interpolation between the first and last poses produces a shortened spline and causes the end of the bone chain to pose awkwardly.

Figure 2: The splines on the top and bottom are good poses for some kind of tentacle-ish appendage. The interpolated pose in the middle, though, is clearly screwy; linear interpolation of the control points has produced a spline that is much shorter than the other two. Veterans of old-school vertex animations may get a bit of *deja vu* from this kind of artifact—the effect is a lot like the shrinkage that afflicted vertex-animated *QUAKE* characters when their keys were too far apart.

This linear interpolation can play havoc with animations that should be flowing and snaky. If the spline shrinks so much that bones on the end of the chain are left hanging, they'll snap back to their rest poses. This is not attractive, not predictable, and hardly sinuous. You can avoid the worst of this by simply using control curves that are longer than the bone chains they control, minimizing the risk that the end bones will be left hanging, though this imposes some unwelcome overhead on the

animators who would have to deal with long, unwieldy controls.

The other artifact that can arise from spline interpolation is creasing. It's possible for control points to cross each other when interpolating between extreme poses. This introduces a kink into the spline. Particularly if this happens near a long bone, you can get a nasty pinching artifact, and in any case the temporary appearance of a loop in your tentacle is rarely what you want.

Interpolation glitches like this are the main reason splines remain a niche tool. Some cartoony rigs actually exploit the stretchiness of the splines by letting the bones grow and shrink to fit the curve—which is great if you want that look, but it's a bit disconcerting in a more realistic setting [CGToolkit's *The Art of Rigging* has a good chapter on stretchy rigs for Maya users and Louis Marcoux's website www.louismarcoux.com has some good videos for Max users]. For most uses, though, the funky

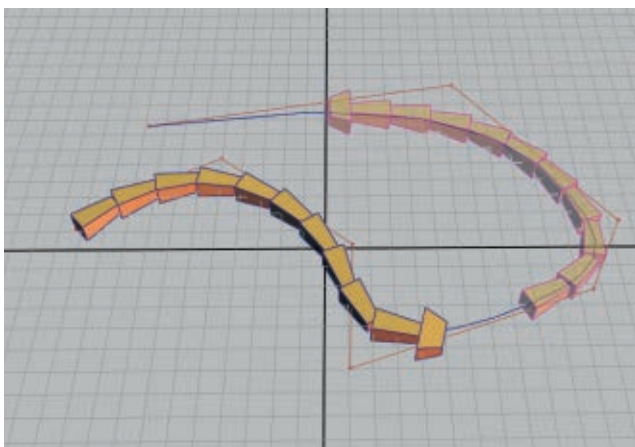


FIGURE 1 The basics of spline IK: a series of bones oriented along a spline curve.

interpolations demand lots of extra keys—which rather undercuts spline IK’s purported advantages of fluidity and lower key density.

CALL OF CTHULHU

» Luckily, there’s an easy way to combine the undoubted virtues of spline IK—smooth interpolation and easy control of very dense bone chains—with the expressiveness and precision that animators need. You can use a simple FK chain to drive your IK spline.

This might sound outlandish, but it’s really quite simple. It hinges on the fact that the control spline is a piece of deformable geometry which can be controlled with bones—or, as we’ll see later, with other deformation tools as well. So, simply create an IK spline rig in the usual way, and then skin the control spline to a second, simpler FK chain, as shown in Figure 3.

This might seem redundant, but each element is contributing something useful. The spline is still controlling the final chain of deformation bones smoothly, making sure they are well coordinated and don’t require tons of hand-laid counter animation keys. The new FK chain poses the spline, preserving its length and eliminating the interpolation glitches. Moreover, FK control does a better job of supporting the whip-like motions you’d expect from your average tentacled beastie. This hybrid setup is easy for animators to work with because it keys like a traditional FK rig.

Riggers, on the other hand, will enjoy the delicate tweakability of the hybrid FK/Spline system. Trying to achieve a similar effect with expressions and constraints would be a nightmare of guesswork and fiddling with arbitrary numerical values. With a skinned IK spline, you can smooth or sharpen the IK spline using the same skinning techniques familiar from poly meshes. A tightly-skinned control spline produces a more jointed appearance. Smooth influence weights with gentle dropoffs produce very smooth, fluid deformations.

Treating the control spline as deformable geometry can also create some striking effects easily. For example, only a masochist would try to recreate the famous pseudopod sequence from the end of *The Abyss* (See Figure 4) in a typical bone-based game engine—the labor of trying to pose the limb while it rippled and distorted would be enormous. With a deformable spline controlling the final deformation skeleton, however, this suddenly becomes possible; inserting a ripple or noise deformer upstream of the skinning deformation that drives the limb allows you to introduce a controlled degree of random distortion without compromising the basic pose.

This kind of setup does demand that the rigger be willing to grapple with the intricacies of deformer order. A rippling deformation that runs up and down a limb would have to be applied before the skinning, to a more or less straight spline as in Figure 5. If you applied the same distortion downstream of the skinning, the final pose would tend to drift away from the pose set by the animator. On the other hand, sometimes that’s what you want: you could, for example, apply a world-space deformation to several limbs after the skinning to produce the appearance of swaying underwater tentacles or eerie ghostlike movements.

Deforming the spline control basically allows you to use modeling tools as part of your animation pipeline. Wrap deformers or lattices can make a moving limb conform to a complex surface. The control curve can be turned into a physics-driven soft-body object and simulated with Havok or Nucleus. Maya’s history-driven NURBS tools offer even more possibilities: You could control bones with a curve-on-surface created by a projection onto a NURBS object, or use the Extend Curve tool to automatically add some extra safe distance on to the end of a control curve. The Rebuild Curve tool is great for uniformly smoothing out a control curve, making it hard for animators

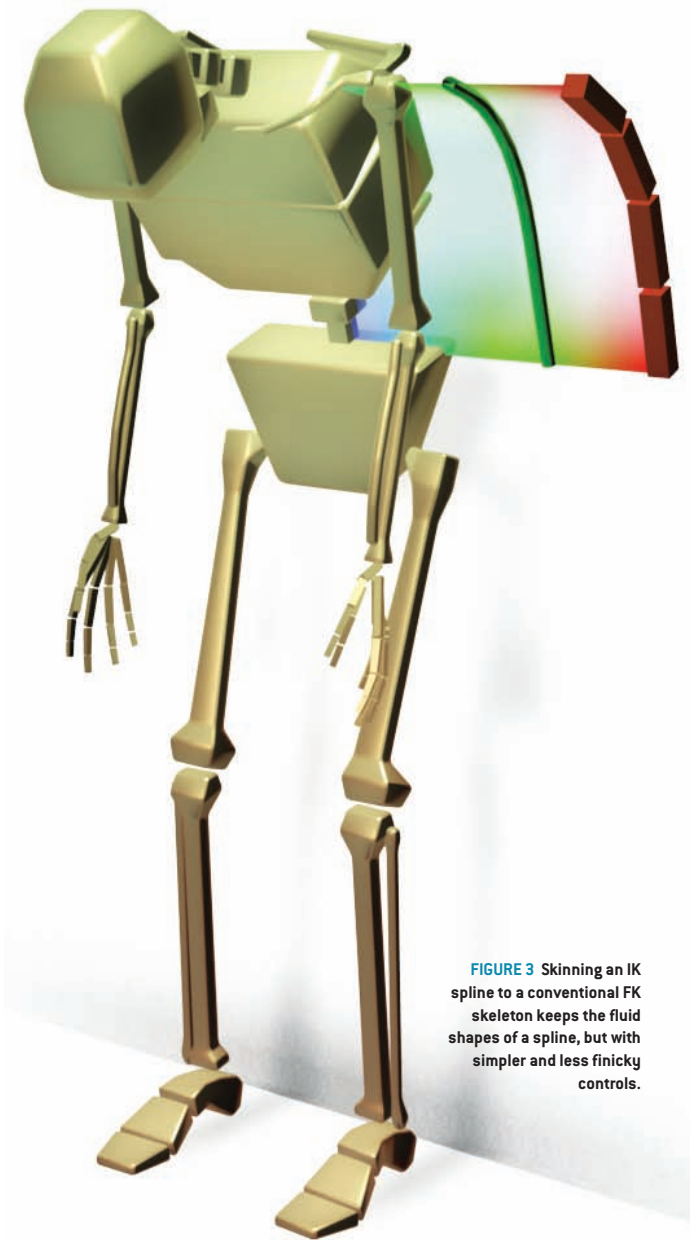


FIGURE 3 Skinning an IK spline to a conventional FK skeleton keeps the fluid shapes of a spline, but with simpler and less finicky controls.

to introduce unwanted kinks. It’s well known that Pixel Pusher disdains overstatement—but the possibilities of control-curve deformation really are almost limitless.

FKING SPLINES!

» Whew. After all that excitement, we need a breather. But, if all that doesn’t convince you to give spline IK another chance, there’s more! This tech is like the Sham-wow, it just won’t quit.

As we’ve often noted in the past, FK bone chains are awkward and unpredictable when they twist around their own axis [for a detailed discussion of the ins and outs of twist management, see “Twist and Shout,” *Game Developer*, April 2004]. Whether it’s herky-jerky interpolation, wonky poses or the dread spectre of gimbal lock, handling rotation in three axes along an FK chain is a major pain for riggers.



FIGURE 4 The famous water tentacle effect from the *Abyss* combined pose animation and rippling deformations—a trick that can be approximated in realtime with spline IK and deformers.

Spline IK offers some excellent tools for managing chains that need to bend and flex and also to twist around their own axes without wobbling. Spline IK rotates bones in two axes to keep them glued onto the control spline. The twist rotation isn't needed to match the spline position, and ordinarily it's simply left out. However, both Max and Maya allow you to control the twist at the start and end of the chain. This grants a degree of fine control over complex chains that FK cannot match.

Both packages allow you to drive the twist using a control object (the "up node" in Max, the "world up object" in Maya). Alas, the "out of box experience" can

serve up some nasty surprises, particularly for Maya users. The mathematical basis of twists suffers from a built-in limitation, similar to the pole-vector "dead zone" that affects conventional IK chains. It's common to find spline IK chains suddenly flipping around their twist axis for no obvious reason. Max's spline chains offer less fine-grained control over twist behavior than Maya's, but at least their default behavior is acceptable. Maya's default is annoying enough that many riggers throw up their hands and swear off spline IK after trying to get them working.

It is possible to whip your twist controls into shape if you keep two rules in mind:

First, be aware that twist angles larger than 180 degrees in any one axis are tricky. If your twist control objects rotate more than 180 degrees in two axes, your chain is going to be highly unreliable. Luckily, at least, those kinds of really extreme twists are rare. If you really need huge twist values, you might consider a secondary twist-only bone chain.

Second, the twist calculations are more reliable if the twist control objects are aligned along the start and end tangents of the control curve. You can use path constraints to achieve this, or attach the twist controls to the start and end of an FK chain driving the control spline as discussed above.

With those rules in mind you should be able to get a reasonably predictable set of twist behavior on a spline IK chain. If you're dealing with really dense bone chains—say the spine of a realistic, high res character—you'll find that the payoff is worth the trouble (for more tips on spline IK for backbones, see "Get Some Backbone," *Game Developer*, November 2004).

YOU'VE GOT SOME SPLINING TO DO

» So there is our case for the rehabilitation of spline IK. It's a tool with issues, no doubt, but it's also capable of great things if you put in the time to learn its quirks. It's definitely worth some poking and prodding from the ambitious rigger or the curious animator. After all, life in the Facebook era is a like a continuous, real-time high school reunion. It'd be nice to think all those old classmates googling us weren't judging hastily, so let's set an example with a little tolerance and willingness to learn.

Except for that flipping twist-axis nonsense. That really has to get fixed. ☹

STEVE THEODORE has been pushing pixels for more than a dozen years. His credits include *MECH COMMANDER*, *HALF-LIFE*, *TEAM FORTRESS*, *COUNTER-STRIKE*, and *HALO 3*. He's been a modeler, animator, and technical artist, as well as a frequent speaker at industry conferences. He's currently a consultant helping game studios perfect their art tools and pipelines. Email him at stheodore@gdmag.com.

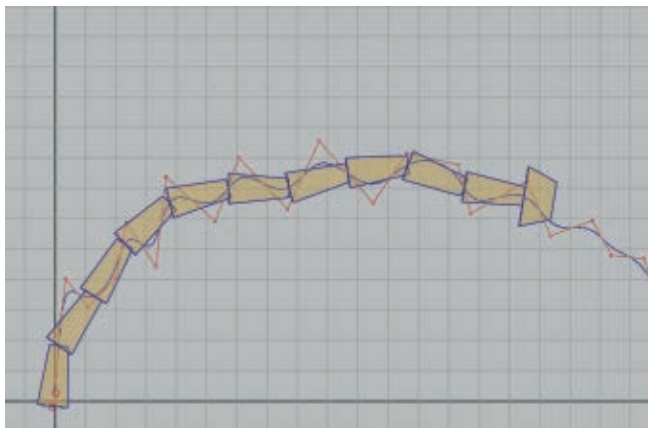
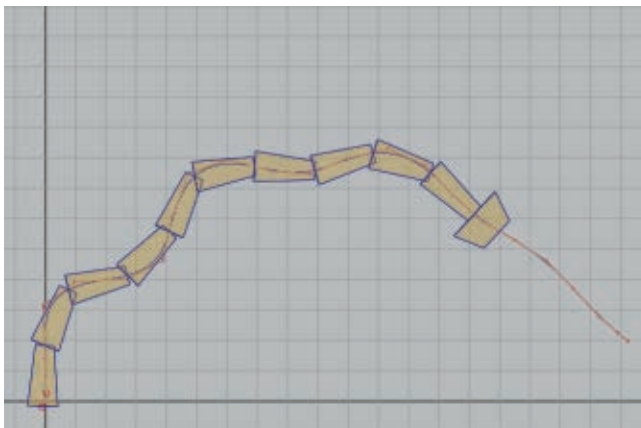


FIGURE 5 Deformation order matters when you're warping an IK control spline. The first spline is ripple-deformed before being skinned; the second is deformed after.



Unreal Technology News

by Mark Rein, Epic Games, Inc.

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

Epic's Unreal Engine 3 won Game Developer magazine's Best Engine Front Line Award for three consecutive years, and it is also the current Hall of Fame inductee.

Epic's internally developed titles include the 2006 Game of the Year "Gears of War" for Xbox 360 and PC; "Unreal Tournament 3" for PC, PlayStation 3 and Xbox 360; and "Gears of War 2" for Xbox 360.

Upcoming Epic Attended Events:

G-Star
Busan, Korea
November 26-29, 2009

GCAP
Melbourne, Australia
December 6-8, 2009

DICE Summit
Las Vegas, NV
February 17-19, 2010

GDC 2010
San Francisco, CA
March 9-13, 2010

Please email:
mrein@epicgames.com
for appointments.



UNREAL ENGINE 3 POWERS PLANET 51: THE GAME

Madrid-based Ilion Animation Studios and Pyro Studios are taking a unique approach to their upcoming Sony Pictures feature film, *Planet 51*, and its SEGA-published video game. Pyro has developed the game while sister company Ilion has focused on the movie, enabling teams behind both releases to share digital assets and work together in a creative atmosphere.

Jose Manuel Garcia Franco, game director, Pyro Studios, said the impact of Unreal Engine 3 on the game's creation has reduced development time as well as the team size. The most important impact that Unreal has had on this project, he said, is that the technology has allowed most of the team to focus on the gameplay aspects, which has resulted in a better game.



Jose Manuel Garcia Franco, Game Director, Pyro Studios

At the peak of production, the team at Pyro included more than 40 developers working with Unreal Engine 3 for the Xbox 360 and PlayStation 3 versions of the game. Franco said that out of this team, just a few coders and a few artists had previous experience using Unreal.

"We needed a very solid engine, and with this being a movie-based game for next-generation consoles, we had time restrictions and a hard deadline," said Franco.

"We chose Unreal because the game had to ship on time. Unreal is a very solid and stable engine, and it gives us a group of tools that enable us to polish our game. It helps us to focus on the game's execution, reducing development time and allowing the team to focus on creativity."

Franco said his team used many Unreal Engine 3 tools, including the robust level-editing system to structure the content with layers -- more than 60 levels in each scenario. In addition, they employed the asynchronous load system for the streaming the world to gamers with no load time. In terms of actual gameplay, the team

used UnrealScript, Matinee and Kismet to bring the action adventure game to life.

"Kismet gives us the opportunity to prototype gameplay quickly, so once mechanics are approved, we have more time to polish them," explained Franco.



Screenshot from *Planet 51: The Game*

"Unreal also helped us a lot developing *Planet 51* across platforms, particularly with the PS3 version, which has been less problematic that we thought it would be at the beginning."

The film was written by *Shrek* creator Joe Stillman, and the open world game adds a lot of new material to the mix through a variety of vehicle-based and pedestrian missions.

"The game feels like you're playing in a real open world," said Franco. "You can control the characters from the movie and utilize their special abilities as you progress through the game."

As Pyro nears completion on its first Unreal Engine project, Franco said it's important to remember that Epic's technology is not just for shooters. "It has a lot of possibilities," said Franco, who points to his own action/platform/racing game, *Planet 51*, as a great example of what can be done with Unreal Engine 3.

Everyone can explore *Planet 51* this Thanksgiving on both the big screen and major game consoles.

Thanks to Pyro Studios for speaking with freelance reporter John Gaudiosi. This interview will be available in full at www.unrealtechnology.com.



For UE3 licensing inquiries email:
licensing@epicgames.com

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

WWW.EPICGAMES.COM



THE ART OF FUN

DISCUSSING GAMES ON THEIR OWN TERMS

WHETHER VIDEO GAMES CAN

be art has been debated for as long as developers have been putting pixels together. It's a question that goes beyond mere academia—the idea of games as art brings professional legitimacy for the industry, and gets to the heart of the concept of games as protected speech.

Naysayers argue that video games will never tell as complex stories, touch as controversial topics, or display emotions as textured as those found in film—largely because games are obsessed with “fun.” This argument suggests that until designers get out of the rut that is focusing on this singular emotion, art will elude us.

I feel this completely misses the point of where the art of video game design actually lives. Do games have to be fun to be successful? Almost certainly. Does this mean that every game needs to push the same emotional buttons? Or that games can't be art? No to both.

Let's back up. There is something holding back the idea that video games could be art. It's the simplistic notion that they will not succeed until they can do what the movies or other story-based media can do. This is extremely shortsighted.

Video games will likely never have the capacity to tell a story as well as a well-crafted Scorsese film. They will struggle to have characters as strong as those found in a Donald Westlake novel. They will

never have the pacing as masterfully handled as an Alan Moore graphic novel. Simply put, timing and pacing is just too important to a pure narrative, and the concept of interactivity gives too much control away to compete with master storycrafters.

Yet fascinatingly, interactive stories in video games always seem to be more artful to me than linear, canned, non-interactive ones.

At Bioware, we work very hard to bring the art of narrative to video games. We strive to push the envelope on narrative, and our devotion to pursuing story-based gameplay is one of the reasons our games are among the best in the business.

All that said, narrative is a red herring in the discussion of games as art. Let's put it this way: can oil paintings be art without great cinematography? Can classical music be great without a killer screenplay? Can a Ming vase be exquisite without compelling characters? These are very silly questions.

So why should anyone insist that video games follow the rules of another artistic medium? Every medium has its own rules for what makes that particular craft capture the viewer's eye and imagination.

This is not a particularly new idea—Marshall McLuhan coined the phrase “the Medium is the Message” back in 1964—talking about a new upstart called “television” that

most critics felt had little or no artistic potential. It took years for television writers to abandon the rules of radio and cinema and create art suited for their genre. Nowadays, many would argue that shows like *Deadwood*, *Battlestar Galactica*, and *The Office* surpass in many ways the westerns, sci-fi, and comedies you can find in a movie theater.

For video games, narrative is an exceptionally powerful tool—one used very well in *KNIGHTS OF THE OLD REPUBLIC* and *STARCRRAFT*, for example. But I posit that many games without story, games like *CIVILIZATION* and *MINESWEEPER*, are elegant, artful games with barely a lick of developer-provided narrative. The art found in these games is less about what you find in a movie theater, and more about what you find in an ancient Chinese puzzle box (boxes which can only be opened through a series of tricks).

AN ARTFUL GAME

» There are many opinions as to what makes good art. I keep coming back to a simple idea—the films, books, and television that I consider art is the stuff that lingers. The stuff I reflect upon afterwards. The films that surprise me, like *Usual Suspects*. The graphic novels that challenge my world views, like *V for Vendetta*. The songs that become deeper and more intricate the more times I listen to them.

Good art is interactive. It is not a passive thing that slides off the viewer's

experience. It involves the viewer, and is made richer by his involvement and his point of view. You reflect on it. You experience it. Great art of all kinds evolves in your mind the more you reflect on it.

Why is the *Mona Lisa* a great piece of art? Because the viewer is left wondering, “Why is she smiling?”

Video games can be art, certainly, because while it will never do other things as well as film, television, novels, photography or oil paintings, it is perhaps the best medium in existence for interactivity.

Certainly, there are video games that are not very artful—but this is also certainly true of a lot of films, television, novels, photography, and oil paintings. The existence of *Porky's* does not invalidate the art found in *Schindler's List*.

The art to be found in a video game is the interactions—how a player interfaces with the game, and what sort of feedback he gets in return. Thus, the art is in the mechanics, the systems, and the simulation.

There are many, many places to go from there—some may find more art in systems that are very naked and easy to manipulate, such as a fine game of *Magic: the Gathering*. Others may find awe and wonder in simulations designed to mimic reality and hide the mechanics. Even others may be most intrigued by the art of how multiple people can share or impede each other in the same system. These are all techniques or

tools for the notion that the interactivity is the art.

Think again to the Chinese puzzle box.

What makes a great Bioware storytelling game? Many things, including some that would be familiar to screenwriters and film directors (a focus on characters and cinematography, for example). But one lost on many observers is the design worldview that the narrative must be interactive—the player must be able to make choices, and the choices should matter. This level of interactivity, self-learning and control is what the movies can't compete with.

WHERE DOES FUN FIT?

» In *A Theory of Fun*, Raph Koster argues that fun is what happens when a player encounters a game system, is challenged by it, learns it, masters it, and then takes it to the next level. In his model, fun is a measure of personal engagement.

In this light, “fun” is the result of successful interactivity. And if you start with the idea that interactivity is the basis for art in games, then it stands to reason that, not only is it possible for fun games to be art, it is very likely that games that will be considered great works of art will be fun.

Saying that a game needs to be fun is kind of like saying that a movie needs good dialogue and direction. Sure, you could have a great movie without a spoken line,



but don't kid yourself, it will be an anomaly.

The trick is that fun is not a simple, one-note emotion inside of the game space. Nicole Lazzaro is a game researcher who describes several different kinds of fun inside successful games: easy fun (the joy of interacting with an environment in a non-threatening way), hard fun (meeting a challenge, being frustrated) social fun (helping someone) and schadenfreude (smacking them down).

And this is still very coarse in texture. The mindless fun in *DINER DASH* is nothing like the cerebral fun of a game of chess. Raiding in *WoW* pushes vastly different buttons than nailing a drum solo in *ROCK*

BAND. Players frequently find different kinds of fun in the same places—interacting with *HALF-LIFE*'s gravity gun is very different than playing some *TEAM FORTRESS*.

Arthouse favorite game *BRAID* is a work of genius not because of a social message, a gripping narrative, stirring characters, or even great art. It's fascinating because it has interesting, novel mechanics. Which, when you figure them out, are fun.

Even more interesting is Brenda Brathwaite's experimental game, *Train*. *Train* is a board game where players try to maximize the number of passengers on their train. They are given no prior clues, but at the end of the game, they are told that the passengers they were loading up were Jews

heading to a concentration camp, and that no one wins. This is an interesting case: most people are shocked, and feel guilty about the fun they had playing the game. It is doubtful that the game would be fun if the players played it a second time in a row. It's all very meta in the whole games and fun discussion.

But the whole message would be lost if the core game interaction of loading passengers weren't fun.

You can tell a narrative in your game, as in *MASS EFFECT*, or discuss a philosophical point of view, as in *BIO SHOCK*. The story can be core to your interactions, or it can be window dressing. It doesn't matter, the game still needs to be fun.

You can have the mechanics of your game

teach an important lesson. Consumerism and success doesn't necessarily make your life easier (see *THE SIMS*). Nuclear war may result in a pyrrhic victory and destroy the earth (*CIVILIZATION*). Killing terrorists won't necessarily make us stronger (web game *SEPTEMBER 12TH*). In all three games, the lesson comes not from preaching, but from interacting, which is a far more powerful way to learn something. But if that interaction's not fun, the messages will never gain traction.

The game industry needs more kinds of games. It needs to reach more markets. The lifeblood of the industry will continue to be to find more kinds of interactions and systems for

players to experiment with. It needs to tap into a wider shell of emotions. It could stand to explore more adult themes. It would be nice to see more games teach real-world relevant knowledge, or encourage players to explore deeper philosophical divides.

But to just base our success on how well we can copy what the movies do? Where's the fun in that? 🎮

DAMIAN SCHUBERT is the lead combat designer of *STAR WARS: THE OLD REPUBLIC* at BioWare Austin. He has spent nearly a decade working on the design of games, with experience on *MERIDIANS9* and *SHADOWBANE* as well as other virtual worlds. Damian also is responsible for *Zen of Design*, a blog devoted to game design issues. Email him at dschubert@gdmag.com.

ACTIVISION®

GREAT GAMES

START WITH

GREAT PEOPLE



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



Visit our site: activision.com



THE MAGIC OF MISDIRECTION

WORKING AROUND AUDIO ENGINE LIMITATIONS

AUDIO IMPLEMENTATION

is like a magic show. With limited resources and a bag of clever tricks, audio implementation gurus can convince a willing audience that anything they hear is real. The more limited the resources are, the more clever the tricks need to be.

Like magicians, a great tool of the audio implementer is the act of misdirection. In magic, misdirection is commonly defined as making an audience look at the wrong thing at the right time. The same practice is true for audio implementation. Audio implementers frequently find themselves needing to misdirect a listener's attention with flashy tricks and loud noises as they finesse smaller, more subtle changes in the game's audio while the audience is distracted. When performed well, the audience has no idea they've been bamboozled.

SLEIGHT OF HAND

» Instance limiting, dynamic loading, and intelligent systemic stream management can go a long way toward saving memory budgets and making life easier on audio implementers. However, implementers can't count on these luxury features being available in every game they tackle. The casual game market has exploded via Flash-based web sites and the Nintendo DS, and for both, audio file sizes can remain a crippling limitation. Apple's iPhone has indisputably become a major gaming device, but

unfortunately has specific restraints other consoles don't, such as the allowance for only a single hardware-decoded stream. Meanwhile, some higher end games are still being driven with legacy audio engine limitations that are leftovers from the last console cycle.

These are just some of the limited frameworks within which audio implementers are forced to work. But indeed, part of the job of audio implementation is trying to make these limitations seemingly disappear into thin air while still providing a rich audio experience under tight restrictions. This is where misdirection becomes most crucial.

Whether on the Wii, the iPhone, or the PlayStation 2, many games can find themselves at the mercy of only a single available stream for audio. Though saddled with limited tech, design teams still expect rich audio worlds and dynamic sound experiences. Mixing ambiences and music together into a single streaming file is a start. However, problems enter when it comes time to change from one file to the next. Without a second stream, there's no ability to crossfade. As such, the existing stream needs to fade down to infinity [zero gain] before the new stream can start, creating an audible gap of silence between the two streams.

While the implementer has no control over the existence of the gap, they can have control over when the gap is created. Changes in stream playback can be



Streaming files that are divided into smaller pieces can be easier to sync with in-game cinematics.

made during pre-scripted events that call the players' attention to other elements of the game. Look to hide stream changes behind reliably scripted instances of explosions, elaborate animations, or dialogue-driven camera cuts. Pull attention away from the stream change by masking it under interesting or louder one-shot instances such as opening and closing doors, elaborate machinery sounds, or item and power-up pick-up sounds. The key is to be positive that your misdirection sounds will always play at the moment that you need them, so make sure to dig into available gameplay scripts that the level designers have created and look for suitable candidates for reliable misdirection.

SMOKE AND MIRRORS

» In-engine cinematics present their own set of challenges. Frequently they make use of special suites

of specific animations, characters, and story-critical dialogue. Rather than loading a slew of files into resident memory for only those moments, pre-scored streaming files can be an effective way to fake detailed audio implementation.

Unfortunately, these sequences are notorious for having their frame rates fluctuate, thereby causing the animations to easily drift out of sync from the pre-rendered audio. The ideal solution is to slave the visuals to the audio, but this isn't always achievable for tech reasons. In these cases, implementers can work with the sound designers to break the single audio file down into smaller constituent chunks. These chunks can then be anchored to scripted events throughout the cinematic as a means of periodically reestablishing sync. By planning the splices at times where single one-shot misdirection sounds can

be triggered, the listener's attention is drawn away from the joins, and they never sense the trick that was performed before their unsuspecting ears.

Like in magic, audio misdirection takes careful planning that is successful only if the audience doesn't notice its inclusion in the act. The key to sneaking it past the audience's detection is reliable audio scripting and a firm understanding of the specific audio engine's limitations. With a few tricks and a knack for making the audience listen to the wrong thing at the right time, audio implementers can find clever solutions to their limited tech options, keep their development teams happy, and wow their listeners with magic. 🎩

JESSE HARLIN has been composing music for games since 1999. He is currently the staff composer for LucasArts. You can email him at jharlin@gdmag.com.



dreamtime
naoto ohshima
interview

WHEN THE SECRET HISTORY OF VIDEO GAMES IS FINALLY WRITTEN, A SPECIAL CHAPTER WILL HAVE TO BE DEVOTED TO Naoto Ohshima. He played a crucial role in exploring the creative possibilities of new console technology by working on some of the most innovative titles of the 1990s. After contributing to early installments of WIZARDRY and PHANTASY STAR, Ohshima created the iconic character designs for SONIC THE HEDGEHOG. The blue mascot's success paved the way for Ohshima's directing turn on SONIC CD, a game brimming over with fresh ideas that few had an opportunity to experience due to the Sega CD's unpopularity. His next major project, NIGHTS INTO DREAMS for the Sega Saturn, was an audacious and beautiful attempt to claim the high ground during the early days of 3D console gaming. He followed NIGHTS with BURNING RANGERS, another game for the ill-fated Saturn that was radically different from its violence-oriented 3D contemporaries. As the 90's came to an end he weathered the shifting fortunes of Sega by forming Artoon, and now heads the sometimes brilliant, occasionally aggravating, always interesting Cavia.

BS: Your first major console game with Artoon was *BLINK*. You might say it was ahead of its time ... games like *PRINCE OF PERSIA* used a time mechanic long afterward and saw great success with it. Where did that idea come from?

NO: It was purely a product of the hard drive included with every Xbox—the original one, not the 360. We wanted to build a console game from the ground up that used the drive effectively.

BS: That's where the play mechanic came from?

NO: That's right. There wouldn't be any other way to do it. The PS2 wouldn't have been able to do it.

BS: Do you think it would be possible to make another mascot-style platforming game in the current era?

NO: Ah, well, I'm making a game like that right now. [laughs] I can't quite talk about that yet, though. In more general terms, the game needs to be something that anyone is able to play, and it needs to have one thing or element that is brand new, that hasn't been done before.

BS: *NIGHTS INTO DREAMS* on the Saturn was the first really 3D game I played, long ago. As the *Nights* character you had a certain path you followed in 2D, but if you went back to human form, you could walk around anywhere you liked. I found a lot of things that way that I couldn't see as *Nights*, and it was a sort of turning point for me; it felt like a real-life world to me, making these discoveries. You don't get that pure feeling of discovery much in games anymore. Was that something you were purposefully aiming for with *NIGHTS*?

NO: Well, with people my age, we didn't really have video games as children. When we came up with concepts for games, we couldn't say "It's some of this game and some of that other game." As a result, especially around that era, you had a lot of games that did not become truly evaluated by the public until long after their release. There just weren't a lot of 3D games back then. Of course, with *NIGHTS*, if you keep

going and going along the ground you eventually run into an invisible wall, so ... We had to think about ways to keep players from going that far off; that's where the Alarm Egg came from [a wandering alarm clock that follows the human player and wakes them from the dreamland, thereby ending the level].

BS: What made you want to put features like that into the non-*Nights* section of the game?

NO: Well, the original inspiration for the game was to create a Peter Pan-like character. *Nights* and Peter Pan share that character trait; they're both capable of things that regular people can only do in their dreams. So I wanted two games here, in a way; one where you were human, and one where you combined with *Nights* to accomplish extraordinary things. Nobody had really played a full-on, free-running 3D game at that time, so we were concerned that people would have trouble comprehending the game if it gave you complete control freedom. As a result, as a human, you have freedom, but only in a small, confined space. Combine with *Nights*, and the game switches to a side-scrolling type, as gamers would've been readily familiar with at that time.

BS: In a way, having a vulnerable human character able to do things *Nights* couldn't is somewhat empowering to the player.

NO: I agree.

BS: *CHRISTMAS NIGHTS* was also ahead of its time—the idea that you could unlock extra stuff depending on what the date is. Those sorts of games are still rare to see.

NO: Indeed. That game is mostly the same as *NIGHTS*, but it was very literally a Christmas present for our audience, a sort of thank-you from Sega to its fans. That was the concept.

BS: I was an enormous fan of Sega, around the Saturn and Mega Drive eras, and that certainly calls to mind the Sega of old. What struck me about *CHRISTMAS NIGHTS* was

that there was a lot of stuff in there, altered graphics and such, that would only be playable for two days' time.

NO: We had the idea to have *Spring Nights*, *Summer Nights*, and so forth, reflecting all of the seasons. It really wasn't a technological challenge, as the textures don't really change much. It didn't take a lot of time, and I think its mission of drumming up interest and excitement among our fans was pretty well met.

BS: What did you think of the new *NIGHTS*?

NO: The Wii one? That project was led by [Takashi] Iizuka, who was the lead designer on the original *NIGHTS*. He really loves that character, and

BS: It's a cliché question, but was that why *Sonic's* main colors are red, white and blue?

NO: [laughs] Well, he's blue because that's Sega's more-or-less official company color. His shoes were inspired by the cover to Michael Jackson's *Bad*, which contrasted heavily between white and red—that Santa Claus-type color. I also thought that red went well for a character who can run really fast, when his legs are spinning.

BS: You were also the director of *SONIC CD*—another game that had time as a play mechanic. Have you liked that mechanic for a while?

NO: Well, I wanted a *SONIC* where the levels changed on you—where



I'm sure that he was able to create the *NIGHTS* that he wanted to create.

BS: It didn't feel the same to me.

NO: It was, perhaps, more Americanized than before. The original *NIGHTS* was chiefly made with the Japanese and European audiences in mind—*SONIC*, meanwhile, was squarely aimed at the U.S. market.

BS: In what way did you position *SONIC* for the U.S. market?

NO: Well, he's a character that I think is suited to America—or, at least, the image I had of America at the time. *NIGHTS* is a more delicate ... well, his gender is deliberately ambiguous, for one.

Sonic would go really fast, like in *Back to the Future*, and bang, wind up in a different place.

BS: Why do you think you've been involved with a lot of games with time elements to them?

NO: I hadn't realized that, actually. [laughs] There must be a part of me that likes that sort of thing, the time.

BS: *SONIC CD* really felt great in action. It doesn't have the full-on speed of *SONIC 2*, but the world feels really alive in the game, much in the same that it did for me in *NIGHTS*—that feeling that the game world would still be alive even if I weren't exploring through it. Was that your intention?

CONTINUED ON PAGE 52

Ohshima Interview

CONTINUED FROM PAGE 51

NO: SONIC CD was made in Japan, while SONIC 2 was made by [Yuji] Naka's team over in the U.S. We exchanged information, of course, talking about the sort of game design each of us was aiming for. But SONIC CD wasn't SONIC 2; it was really meant to be more of a CD version of the original SONIC. I can't help but wonder, therefore, if we had more fun making CD than they did making SONIC 2 [because we didn't have the pressure of making a "numbered sequel"].

What I really wanted to do was just have this sonic boom, with a flash, and have the level change on you instantly. We just couldn't manage it on the hardware, though, so instead there's that sequence that plays while it's loading. [laughs] I kept fighting and fighting with the programmers, but they said it just wasn't possible.

BS: *I bet they probably could have done it.*

NO: I know! [laughs] If Naka was doing the programming, I think it could've been done.

BS: *Do you know why Naka left Sega?*

NO: No, actually.

BS: *The way he put it, he was too far up—he was doing nothing but management and couldn't do any design or programming work on games. He couldn't even influence SONIC anymore. I thought that was admirable that he went on to try to do creative work again.*

NO: Indeed. In fact, it was really the same deal with me and Artoon. Now, though, the group as a whole has gotten really big—it's sort of a mini-Sega.

BS: *So will you making your type of game in the future, then, or are you going to be making more darker-style games, as Artoon and Cavia have been moving toward?*

NO: Well, the important thing is to make gamers happy—the users playing your game. When you make something that's truly new—well, it's not that you have to with every project, but when you do, you're

expanding the world of games. That's what I want to do; I want to keep on doing new things, and I think that's possible with Artoon, Cavia, and Feelplus.

BS: *The way I see it, if the director makes the sort of games he wants to make, then the end results are going to be more interesting. The gamers are important, of course, but if you make a game for yourself, that adds character to the result.*

NO: Well, for example none of my favorite movies were really hits. [laughs] Myself and the gaming audience, we're different. I like new things, but if something is too new, then gamers won't be able to comprehend it. So you have to think about your audience at least a little bit, or else you'll make something that runs the risk of being incomprehensible. That's why I want to keep them in mind.

BS: *Certainly that's true, but sometimes you see indie films become hits, too. Taking that risk is important, I think.*

NO: Yeah, but the sort of indies that become hits are those that are easy to follow for anyone who watches them. So, in the end, you want to make it just a little new—not completely so.

BS: *With games, though, you have download platforms that can support titles like ECHOCHROME [the PSP version of which Artoon worked on]. Games like that, they don't have to be massive hits.*

NO: Sure. ECHOCHROME doesn't have very fancy graphics at all, but play it, and it's really fun. It really depends on the game.

BS: *What made you decide to move into that darker territory with VAMPIRE RAIN and Artoon?*

NO: Games are a part of the overall realm of entertainment, including movies and music and so on. I think that movies play a sort of big-brother role for the game industry. The Japanese film industry has been around for ages, but once films from other countries began to see



NIGHTS INTO DREAMS

wide distribution here, Hollywood films very quickly became the most popular. Within that environment, Japanese animation has managed to attract worldwide praise, which is great. But we're seeing a sort of Hollywood-ization of the game industry right now, and Japan's traditional strengths in cartoon-style games are going by the wayside. So in thinking about the future, I realized I wanted to do both "real" and "cartoon" games. Now, VAMPIRE RAIN got a negative reception from a lot of its players, and we regret a lot of things with that game, so in the future we definitely want to make games that excite people a great deal more.

BS: *You could say that SILENT HILL has succeeded at that in the past, showing an auteurism without being Hollywoodized—opening up to the world on its own terms.*

Those cartoony games do not sell well in Japan any longer, either—it seems like platformers like SONIC and MARIO are the only ones that do. Why do you think that is?

NO: Because they're all the same. The art and the storylines may be different, but the gameplay is the same.

BS: *Lately I've been thinking about how to modernize games like those. I think the first hour of so of UNCHARTED is close. You don't have a gun or anything—you're just jumping around and searching*

for treasure in this gorgeous environment. I thought that was really fun, until it got to the shooting bits, but I'm not sure if such a game would sell well.

NO: I do have an interest in that sort of thing as well, but I think games that have more of an original idea to them come first in my mind.

BRANDON SHEFFIELD: *You recently became the CEO of Cavia, moving over from Artoon—or is it more complex than that?*

NAOTO OHSHIMA: Well, the head of the company here quit because of ... family issues. [laughs] Since they then lacked a president, I sort of wound up taking over that role, since we're all technically part of the same outfit. [Artoon, Cavia, and Feelplus are all under AQ Interactive.]

BS: *I do like a lot of Cavia's games. Though they often seem unpolished—the games seem to get "almost there," do you know what I mean? BULLET WITCH and GHOST IN THE SHELL: STAND ALONE COMPLEX both have something to them, but these certainly don't feel like Ohshima-style games.*

NO: Well, certainly, I do a lot of management-type work these days. I definitely want to make something again! I really do. ☺

BRANDON SHEFFIELD never says he can't. Nothing can survive the will to stay alive 'cause if you try, you can do anything! Email him at bshffield@gdmag.com.

TURN YOUR PASSION FOR GAMING INTO A CAREER

Game Art

Bachelor's Degree Program
Campus & Online

Game Design

Master's Degree Program
Campus

Game Development

Bachelor's Degree Program
Campus

Game Design

Bachelor's Degree Program
Online



Campus Degrees

Master's
Entertainment Business
Game Design

Bachelor's
Computer Animation
Digital Arts & Design
Entertainment Business
Film
Game Art
Game Development
Music Business
Recording Arts
Show Production
Web Design & Development

Associate's
Graphic Design

Online Degrees

Master's
Education Media Design & Technology
Entertainment Business
Entertainment Business: with a Sports Management Elective Track
Internet Marketing
Media Design

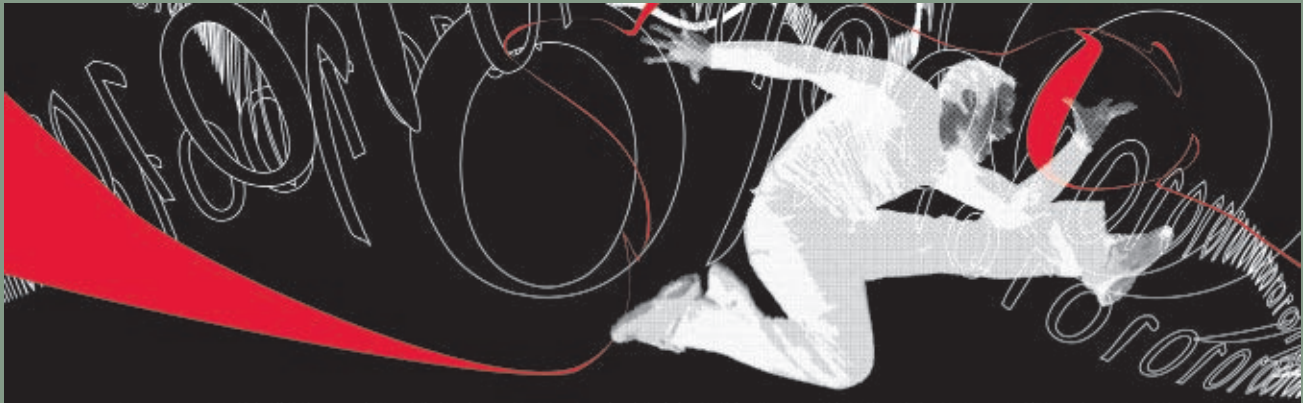
Bachelor's
Computer Animation
Entertainment Business
Game Art
Game Design
Graphic Design
Internet Marketing
Music Business
Web Design & Development



FULL SAIL
UNIVERSITY

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

fullsail.edu



BECOME A MASTER OF DIGITAL MEDIA

We offer a 20-month master's degree in entertainment technology and digital media. This powerhouse graduate program combines industry-facing curriculum, real world projects, and a 4-month internship in Vancouver, Canada: videogame capital of the world.

Learn more. Contact alison_robbs@gnwc.ca or visit mdm.gnwc.ca

CENTRE FOR DIGITAL MEDIA

Masters of Digital Media Program
@ Great Northern Way Campus



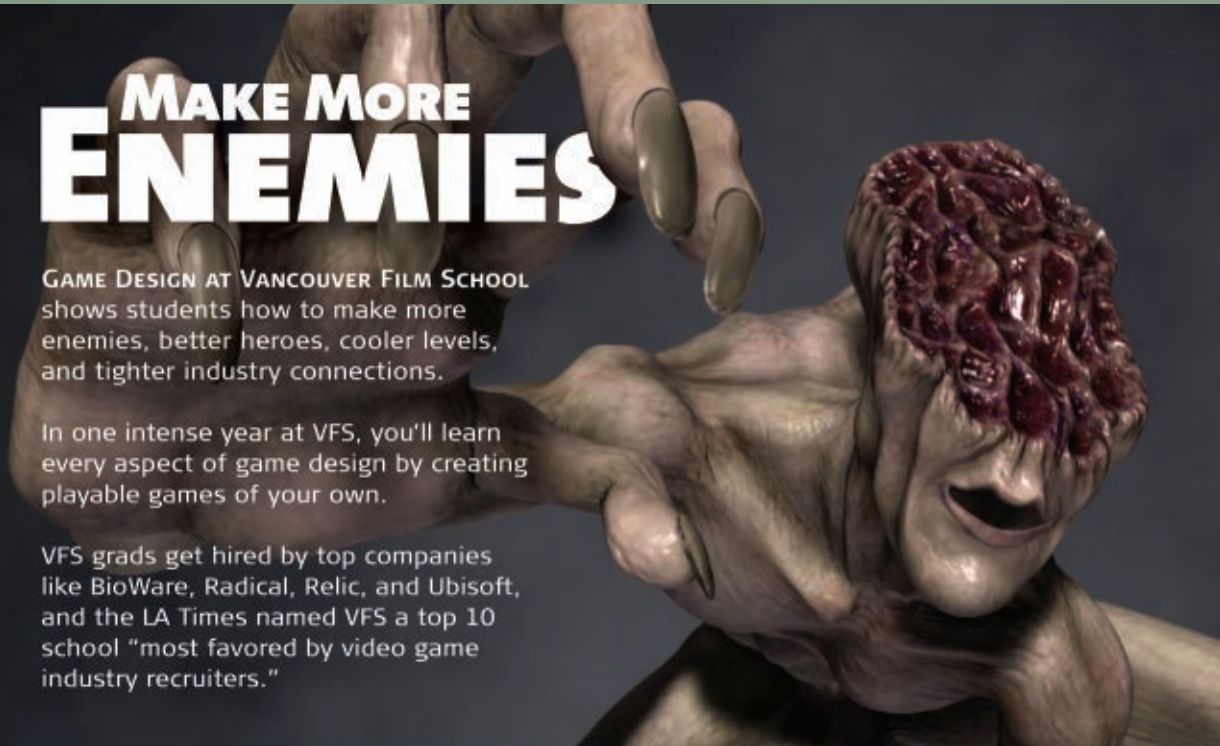
a collaborative university campus environment

MAKE MORE ENEMIES

GAME DESIGN AT VANCOUVER FILM SCHOOL shows students how to make more enemies, better heroes, cooler levels, and tighter industry connections.

In one intense year at VFS, you'll learn every aspect of game design by creating playable games of your own.

VFS grads get hired by top companies like BioWare, Radical, Relic, and Ubisoft, and the LA Times named VFS a top 10 school "most favored by video game industry recruiters."



VFS

vfs.com/enemies

VFS student work by Moo Won Kim

THE LEADING PROFESSIONAL GAME SOURCE FOR JOB SEEKERS AND EMPLOYERS

Access the Web's Largest Game Industry Resumé Database and Job Board



Take Control of Your Future Today!
Log onto www.gamasutra.com/jobs



THINK SERVICES

CENTER FOR DIGITAL IMAGING ARTS AT BOSTON UNIVERSITY



capture
imagination

3D ANIMATION
+ INTERACTIVE MEDIA

GAME ART + CHARACTER ANIMATION

..... certificate programs

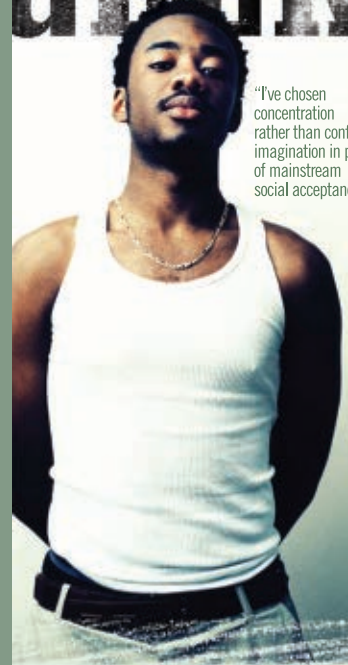
TWO CAMPUS LOCATIONS
WALTHAM, MA & WASHINGTON, DC

Financial assistance and career services available.
Now accepting applications. **Apply today!**



CDIABU.COM

I'M A GEEK.



"I've chosen concentration rather than conformity, imagination in place of mainstream social acceptance."



Come get your geek on.

UAT is a community of geeks—passionate seekers of knowledge and truth in an ever-expanding universe of technology.

LEARN:

- ADVANCING COMPUTER SCIENCE
- ARTIFICIAL LIFE PROGRAMMING
- DIGITAL MEDIA
- DIGITAL VIDEO
- ENTERPRISE SOFTWARE DEVELOPMENT
- GAME ART & ANIMATION
- GAME DESIGN
- GAME PROGRAMMING
- NETWORK ENGINEERING
- NETWORK SECURITY
- OPEN SOURCE TECHNOLOGIES
- ROBOTICS & EMBEDDED SYSTEMS
- SERIOUS GAME & SIMULATION
- TECHNOLOGY FORENSICS
- VIRTUAL MODELING & DESIGN
- WEB & SOCIAL MEDIA TECHNOLOGIES

www.uat.edu > 877.UAT.GEEK

ADVERTISER INDEX

| COMPANY NAME | PAGE | COMPANY NAME | PAGE |
|---------------------------------------|------|--|------|
| Activision | 48 | Killer Tracks | 6 |
| Autodesk | C3 | Masters of Digital Media | 53 |
| Center for Digital Imaging Arts | 55 | Perforce Software | 3 |
| Epic Games | 1045 | RADGame Tools | C4 |
| Full Sail University | 53 | twofour54 | 12 |
| Havok | C2 | University of Advancing Technology | 55 |
| Insomniac Games | 41 | Vancover Film School | 54 |
| Intel | 28-3 | | |

Game Developer (ISSN 1073-922X) is published monthly by United Business Media LLC, 600 Harrison St., 6th Fl., San Francisco, CA 94107, (415) 947-6000. Please direct advertising and editorial inquiries to this address. Canadian Registered for GST as United Business Media LLC, GST No. R13288078, Customer No. 2116057, Agreement No. 40011901. **SUBSCRIPTION RATES:** Subscription rate for the U.S. is \$49.95 for twelve issues. Countries outside the U.S. must be prepaid in U.S. funds drawn on a U.S. bank or via credit card. Canada/Mexico: \$69.95; all other countries: \$99.95 (issues shipped via air delivery). Periodical postage paid at San Francisco, CA and additional mailing offices. **POSTMASTER:** Send address changes to *Game Developer*, P.O. Box 1274, Skokie, IL 60076-8274. **CUSTOMER SERVICE:** For subscription orders and changes of address, call toll-free in the U.S. (800) 250-2429 or fax (847) 647-5972. All other countries call (1) (847) 647-5928 or fax (1) (847) 647-5972. Send payments to *Game Developer*, P.O. Box 1274, Skokie, IL 60076-8274. For back issues write to *Game Developer*, 4601 W. 6th St. Suite B, Lawrence, KS 66049. Call toll-free in the U.S./Canada (800) 444-4881 or fax (785) 838-7566. All other countries call (1) (785) 841-1631 or fax (1) (785) 841-2624. Please remember to indicate *Game Developer* on any correspondence. All content, copyright *Game Developer* magazine/United Business Media LLC, unless otherwise indicated. Don't steal any of it.



STACK TRACE

GAME INDUSTRY DETECTIVE

CASE #1: THE MISSING MUSCLE MILK

ALL THE RAIN IN THE WORLD COULDN'T WASH AWAY THIS CITY'S DIRT, BUT that wasn't stopping the man upstairs from trying. It came down in great sheets, the way it did in the beginning of METAL GEAR SOLID 2. I wasn't jumping off bridges onto tankers, though—I was sitting in my office, expecting another day of my usual work: nothing. Funny how that works. Suddenly, the doorbell rang, and a sassy dame with an air of smoldering sexuality sashayed into my life. Well, that's what I wish I could say, but come on, this is the game industry we're talking about here. A frail guy who looked like he was fifteen squeaked into the room. He tugged nervously at his two-sizes-too-big NVIDIA GeForce t-shirt and quickly exited the GPS application on his smart phone before taking a seat and spilling his story.

"Are you the famous Stack Trace? I need your help," he began. "I don't know who else to turn to at this point. Someone stole my plastic container of Extreme UltraGain Mega 7000 protein powder from my desk last night. I asked the head of the studio about it, but he just laughed it off, and suggested I eat some Red Vines instead." He sighed. "I spent my whole royalty bonus from our last game on that tub. I can't afford another one. Please, Mr. Trace, do you think you could take the case?"

How could I say no? Especially since his request had unintentionally rhymed. Clutching the only clue I had—a crumpled Google Maps printout of the studio's precise location—I put on the trenchcoat I bought after *The Matrix* came out and got ready to do my investigation.

I waited until the wee hours of the morning on Sunday, so I could check the office for clues while it was deserted. I found a grate that revealed a passageway and, since it was the only one around, figured it was the way in. But as I popped open the vent from the ceiling and let myself inside,

I immediately sensed something was wrong. The lights were on and machines were humming.

I felt eyes on me from all directions. It turned out the whole team was there, crunching for an upcoming milestone.

Just when I thought things couldn't get worse, a hulk of a man the size of a freight train barreled toward me. "Woah, easy there, Zangief," I said. Me and my big mouth. I was on the floor before I knew it, Red Rings of Death dancing around my head.

When I came to, I was in the corner office of an old friend, Page Fault. He stared at me with his

small eyes as I struggled with the floor. Zangief had done a real number on my skull.

"Page," I said, coughing. "I figured you'd be running the show around here by now. Tell that guard dog of yours to lay off the steroids. Or are you still mad about that repeatable crash I found?"

Page smirked, his forehead shiny with sweat. "Now, Stack. That was a long time ago. The zero-day patch we had to do because of you is all water under the bridge now. Still, though. You know we can't have people sneaking around the office—least of all gumshoes," he said. "Somebody might find out about the flamethrower." He laughed maniacally.

"What?" I stammered. "Even someone as sadistic as you wouldn't..."

"Nah, I mean in the game. It's under embargo right now; the announcement's set for next Tuesday."

I finally got to my feet. "I'm not here for secrets, anyway. This is about some missing muscle milk."

"Nice alliteration," he said, then furrowed his brow. "You know, that stuff's bad for you."

"Yeah? Worse than Red Vines?"

He snorted and flicked some ash from his cigar onto a stack of unopened artist portfolios.

"Let me ask you something," he said. "Suppose this hypothetical person got his hypothetical protein powder back. Suppose he had the discipline to use it and build up some muscles. Now, everyone knows game designers aren't supposed to have muscles. That kind of wacko thinking might put other ideas in his head, like maybe he starts standing up to us when we need him here at three in the morning on a Sunday, or thinking that the games we're making here are uninteresting. He might even decide to go indie."

"Huh? Working out can lead to going indie?"

Page stood up, putting his palms down onto the surface of his desk. "Stack, old buddy. You, of all people, ought to understand this isn't about the snacks we keep in the kitchen. It's about what's right, the natural order of things. Get it? We're running a business here. And you don't run a business by letting people do whatever they want." I didn't get it, not in the least, but before I could say anything more Page was gesturing at Zangief. "Take our friend here and show him the door."

My puny stick arms were no match for this brute's iron-vise grip. I hit the pavement so hard I thought I caught a glimpse of Jeff Minter's next game. And maybe it was just the impact talking to me, but suddenly I felt like understood. It wasn't the protein powder itself—it was what it represented: the desire to improve one's self at the expense of the studio's profitability. Better to have a staff of anemic pushovers than people who, however haltingly, started to take their destiny into their own hands ... right?

The flash of insight wore off quickly as the rain started to soak through my trenchcoat. I guess you shouldn't buy clothes on eBay. This day was turning out to be a doozy. I'd have to tell my client I couldn't get his UltraGain powder back. I'd gotten bumps and bruises. I wanted a drink. But before that, there was one thing I had to do first. In my wallet, I found my wrinkled old gym membership card. It was about time I renewed. **END**

MATTHEW WASTELAND writes about games and game development at his blog, *Magical Wasteland* (www.magicalwasteland.com). Email him at mwasteland@gdmag.com.

Autodesk Games Insight

The latest scoop from Autodesk Media & Entertainment



“Titles like *WET* are getting a lot closer to interactive films at this point, and the quality of the content is more important than ever. So if we can optimize our pipelines to enable more iterations at a faster pace, then we’re ahead of the game. Tools like Maya, MotionBuilder and Kynapse help us get there”

— Martin Walker
Chief Technology Officer
Artificial Mind and Movement

©2009 WET, Artificial Mind and Movement. ©2009 Bethesda Softworks. Images courtesy of Artificial Mind and Movement

WET: A2M Uses Autodesk Creative Tools and Middleware

Montreal-based game developer Artificial Mind and Movement (A2M) is best known for developing family-friendly games and licensed titles like *Iron Man* and *Indiana Jones*. So when it came time for their first foray into mature gaming – for them to make their mark – they knew it would have to set a new standard in both game play mechanics and in look and feel.

WET centers on Rubi, a feisty problem fixer trying to figure out who double-crossed her and why. It is a third-person shooter with a unique mix of acrobatics and swordplay. The game’s dual-targeting system lets players control one of Rubi’s weapons and auto-targets the other. Rubi can aim and shoot in different directions while doing complex acrobatic moves. Players are rewarded for making Rubi jump off the walls, shoot down ladders, stay in the air and be as acrobatic as possible.

WET was written by Duppy Demetrius and Rubi was voiced by actress Eliza Dushku (Dollhouse, Buffy the Vampire Slayer). *WET*’s gritty look and feel can be compared to such films as *Kill Bill*, *Desperado* and *Sin City*, with a clever use of photoreal 3D and stylized graphic novel-type visuals. In key moments Rubi goes into “Rage Mode”, giving players an entirely different experience. Rubi becomes faster and stronger, and the graphics transform to a minimalist color palette with enemies in contrasting colors.

With a staff of 500 working on as many as a dozen projects a year, A2M needed to standardize and streamline several key development processes with reliable, off-the-shelf tools. To do so A2M used Autodesk® Maya® and Autodesk® MotionBuilder® software to build the game characters and levels for *WET*. Autodesk® Kynapse® middleware was used to implement the artificial intelligence (AI) that drives the game’s non-playable characters.

Using Kynapse AI middleware helped A2M focus its development efforts on the creative look and feel of the game and high-level game-play interactions, rather than on the time-consuming process of path-finding and location awareness.

“Our main philosophy with regards to development tools has always been – if it’s good and it already exists with the right price tag, why reinvent it? That’s why we chose to work with Kynapse as our AI engine for *WET*. In the game, the player confronts a large number of enemies. We needed a path finding and location awareness system to control decision-making for these characters in a coherent manner,” explained Martin Walker, CTO, A2M. “Kynapse has a fantastic hide-and-shoot algorithm that was used to conceal and disperse the enemies in a natural way throughout environments.”

MotionBuilder was used to enhance motion-captured sequences, helping A2M experiment with character

movement. “We used MotionBuilder extensively to clean up motion capture data for *WET*,” said Walker. “Also, when we did our mo-cap, we weren’t sure which villain would take on which moves. With MotionBuilder, we were able to retarget motion data onto different characters even if the skeletal data wasn’t necessarily a perfect match.”

WET is A2M’s first mature AAA game title. “Titles like *WET* are almost interactive films at this point, and the quality of the content is more important than ever. So if we can optimize our pipelines to enable more iterations at a faster pace, then we’re ahead of the game. Tools like Maya, MotionBuilder and Kynapse help us get there,” concluded Walker.

For more information on Autodesk games software and middleware please visit www.autodesk.com/games.

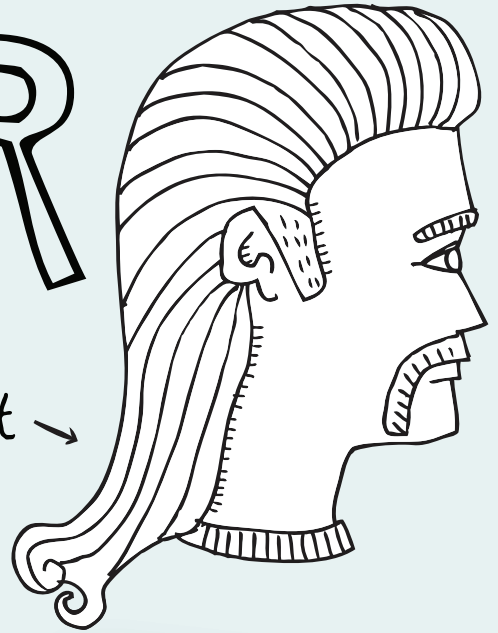


“Rage Mode” stylized combat in A2M’s *WET*.

Autodesk, Kynapse, Maya, and MotionBuilder are registered trademarks or trademarks of Autodesk, Inc., and/or its subsidiaries and/or affiliates in the USA and/or other countries. All other brand names, product names, or trademarks belong to their respective holders. Autodesk reserves the right to alter product offerings and specifications at any time without notice, and is not responsible for typographical or graphical errors that may appear in this document.
©2009 Autodesk, Inc. All rights reserved.

THERE ARE LOTS OF WAYS TO BE AN EVEN

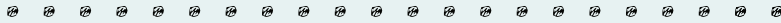
COOLER GAME DEVELOPER



like grow a mullet →

← drive a sweet old Camaro

or you can use **BINK VIDEO**.



You get an amazing, super **FAST** video and audio codec - all in a simple, clean API. And Bink Video

runs on **EVERY PLATFORM!**



USING BINK VIDEO NOT ONLY MAKES YOU
cooler, IT MAKES YOU **rad!**



www.radgametools.com
(425) 893-4300