THE LEADING GAME INDUSTRY MAGAZINE VOL18 NO6
JUNE/JULY 2011 INSIDE: KILL ALL JAGGIES!

POSTMORTEM

FEATURES

DEPARTMENTS

CONTENTS JUNE/JULY 011

VOLUME 18 NUMBER 06

THE PREDICTABLE PROTAGONIST

EMBRACING DIVERSITY IN INTERACTIVE ENTERTAINMENT

LIPS SNARL AND TEETH GNASH. Blood spatters across the ground as the camera pans up to show a monstrous figure eviscerating a human corpse, in front of a panoramic view of a ruined America, overrun by bestial invaders. Suddenly, the figure's head explodes in a shower of green goo, as the camera whip-pans over to reveal the source of this new destruction—a massive weapon is outstretched, in the capable hands of our hero; a Caucasian male with close-cropped hair and a steely gaze. The logo fades in as you yawn—we've just seen the character reveal of nearly every video game protagonist ever.

BACK TO THE BLAND

>> There's a serious lack of variety in our game heroes. Caucasian, heterosexual, and male are pretty much given components of any new game protagonist, and any distinguishing characteristics are built from that base. Characters of other races and genders tend to be relegated to background characters or comic relief, if they're included at all.

I've said it time and time again, but diversity of all types is necessary for the game industry to continue to evolve. We have advanced in many arenas, but our diversity is definitely weak, both within studios and in our game characters. I'm not calling for video game affirmative action, per se, but too much of the same thing leads to an insular medium. And why is it that so many of our protagonists look the same? Why should the protagonist be Caucasian and not African? Or Indian? Who will it alienate?

Some games get around the protagonist issue through character creators. FALLOUT: NEW VEGAS, MASS EFFECT, and their ilk allow players to choose their own ethnicity with sliders, but this has little effect on gameplay. And while racial issues are dealt with to some degree in these games, they are truly "racial," in that they deal with races of beings, be they elf, mutant, or space alien. It's a step, but why is race-oriented dialog always abstracted from reality?

WHO ARE WE FIGHTING FOR?

>> In the late 2000s, USC researcher Dmitri Williams looked into ethnic portrayals in games, using the bestselling titles from 2006–2007. They sampled 150 games, recording a half hour of gameplay from each, logging the ethnic makeup of every character they came across, for a total of 8,500. They compared this data with that of the U.S. Census.

What they found was that white characters were overrepresented by 7%, and Asians were overrepresented by 26%, while black characters were underrepresented by 13%, Hispanics by 78%, Native Americans by 90%, and biracial characters by 42%. And that's just speaking of the U.S.—when the test was implemented, Caucasians represented 75% of the population. Consider then, how overrepresented they may be in the many other markets in which games are played. And remember that this is across all characters in games. Speaking strictly of protagonists, this becomes even more pronounced.

Williams found, further, that while the in-game representations didn't match the U.S. population, it did match the ethnic makeup of the IGDA. So, it seems, we make characters that look like us, not like our players: A late 2010 Nielson Group study showed that in a sample of Americans aged 18–49, African Americans on average spend more time playing console games than any other ethnic group.

THE SAFEST PATH

>> White people are the "safest" group to include in entertainment media. You can make them heroes, you can vilify them, and nobody will bat an eye. Make your protagonist black, and you're likely to get some backlash about the portrayal, no matter how hard you try. That's nothing compared to the can of worms you open if you make a non-white character a main villain. The backlash may not be severe, but I think developers are still wary of it.

Any kind of imagined backlash shouldn't discourage you. I think there are a lot of rewards to be gained through different ethnic depictions. Would more Indians play your game if you had an Indian protagonist? Maybe not. Would your writers and designers get more opportunity to explore different narrative territory? Absolutely, and that's not only freeing, it's the kind of thing that allows simple innovations. Though some may be wary of tackling an ethnicity that's not our own, the best writers can bring any character to life, regardless of gender, orientation, or origin.

The traditional game industry is very risk-averse. But as we've seen time and time again, many of the companies with the highest profit margins these days are smaller studios taking bigger risks.

WHITE WASHING

>> Games aren't the only guilty party. The recent *Avatar: The Last Airbender* live action movie scrubbed the cast a pasty white. Likewise, the frankly blasphemous proposed *Akira* remake is bringing out an all-white cast for its vision of "Neo New York." But the world of film has done much more to advance racial understanding than it has to hinder. Spike Lee, Pedro Almodóvar, and others have done great work to bring other viewpoints into the public sphere through entertainment. With our interactive medium, couldn't we do better?

In my recently cancelled game, I attempted to create four main characters of different races, genders, and backgrounds. They were quite different from my life experience as a Caucasian male. I am certainly not among the best writers or designers in the industry. So, couldn't you and your team do better? Shouldn't you?

—Brandon Sheffield
twitter: @necrosofty

Sources: http://bit.ly/byS9DR http://bit.ly/eeKcjj

GAME DEVELOPER MAGAZINE
WWW.GDMAG.COM

United Business Media
303 Second Street, Suite 900, South Tower
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*

FOR DIGITAL SUBSCRIPTION INFORMATION
www.gdmag.com/digital

EDITORIAL

PUBLISHER
Simon Carless | *scarless@gdmag.com*
EDITOR-IN-CHIEF
Brandon Sheffield | *bsheffield@gdmag.com*
PRODUCTION EDITOR
Jade Kraus | *jkraus@gdmag.com*
ART DIRECTOR
Joseph Mitch | *jmitch@gdmag.com*
DESIGNER
Jessica Chan
PRODUCTION INTERN
Tom Curtis
CONTRIBUTING WRITERS
Tom Carroll
Jesse Harlin
Damion Schubert
Aras Pranckevicius
Steve Theodore
Kim Pallister
Matthew Wasteland
ADVISORY BOARD
Hal Barwood **Designer-at-Large**
Mick West **Independent**
Brad Bulkley **Microsoft**
Clinton Keith **Independent**
Brenda Brathwaite **Lolapps**
Bijan Forutanpour **Sony Online Entertainment**
Mark DeLoura **THQ**
Carey Chico **Independent**
Mike Acton **Insomniac**

ADVERTISING SALES

GLOBAL SALES DIRECTOR
Aaron Murawski e: *amurawski@think-services.com*
t: 415.947.6227
MEDIA ACCOUNT MANAGER
John Malik Watson e: *jmwatson@think-services.com*
t: 415.947.6224
GLOBAL ACCOUNT MANAGER, RECRUITMENT
Gina Gross e: *ggross@think-services.com*
t: 415.947.6241
GLOBAL ACCOUNT MANAGER, EDUCATION
Rafael Vallin e: *rvallin@think-services.com*
t: 415.947.6223

ADVERTISING PRODUCTION

PRODUCTION MANAGER
Pete C. Scibilia e: *peter.scibilia@ubm.com*
t: 516-562-5134

REPRINTS

WRIGHT'S MEDIA
Ryan Pratt e: *rpratt@wrightsreprints.com*
t: 877.652.5295

AUDIENCE DEVELOPMENT

TYSON ASSOCIATES Elaine Tyson
e: *Elaine@Tysonassociates.com*
LIST RENTAL Merit Direct LLC
t: 914.368.1000

United Business Media
WWW.UBM.COM

BLIZZARD IS HIRING

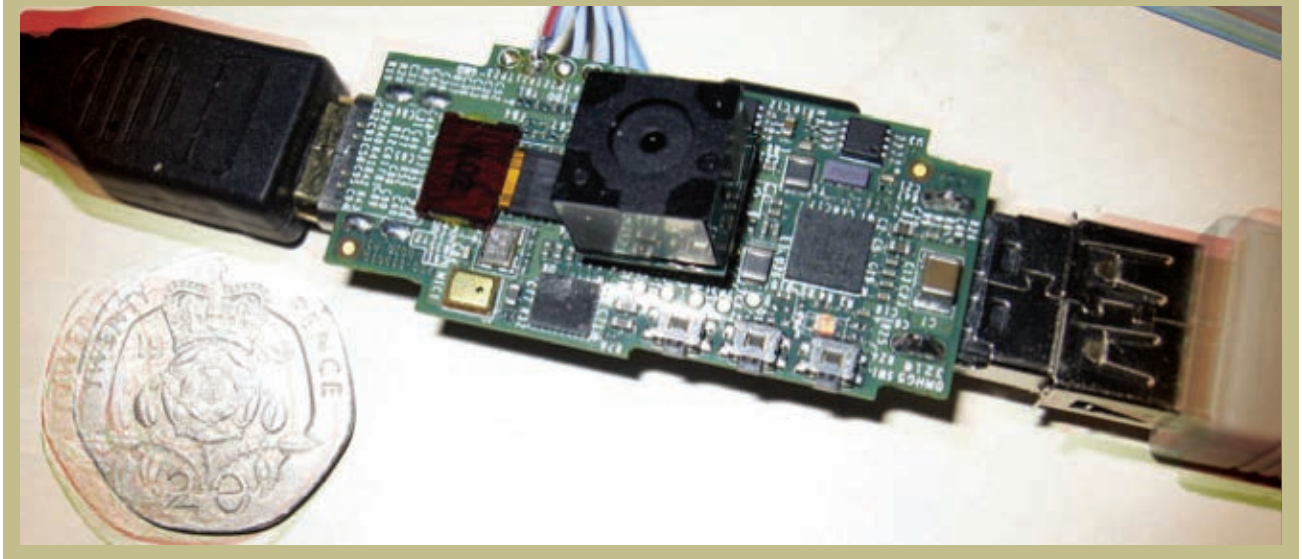
We are actively recruiting across all disciplines for the following locations:

IRVINE, CALIFORNIA | AUSTIN, TEXAS | VELIZY, FRANCE | CORK, IRELAND
SINGAPORE | SHANGHAI, CHINA | TAIPEI, TAIWAN | SEOUL, SOUTH KOREA
SAO PAULO, BRAZIL | BUENOS AIRES, ARGENTINA | MEXICO CITY, MEXICO

jobs.blizzard.com

braben's raspberry pi

A delicious little device that could revitalize hobby programming



\\\ Late last week, when David Braben announced Raspberry Pi—the $25 computer on a stick—he expected a reaction. But nothing like the one he got. The YouTube video announcing the gadget is currently standing at over 415,000 views.

"It's been shocking. Twitter went bananas," he told us. Speaking from Cambridge, England, he outlined his plans for the device, which he hopes will bring real creativity back to how people interact with computers.

Raspberry Pi sports a 700MHz ARM11 processor with 128MB of SDRAM, locked together with a USB port, and a HDMI connection or composite video. Users are invited to stick it in their TV, connect a keyboard, and begin playing with the Linux system.

It's a gadget, but first and foremost, it's a charitable effort to get youngsters excited by the possibilities of programming. Braben says he became "very upset" at the dwindling number of applicants to his games company Frontier Developments (KINECTIMALS, LOSTWINDS) showing up with computer science bona fides. Turns out, people are less willing to take up computer science when they leave school than in years gone by. Braben lays the blame squarely with teaching methods that stress an "office skills" approach to learning about computers. To that end, his big plan is to roll the device out to school kids.

Braben says, "There are lots of creative tools at the high end, if you already have a lot of computer knowledge, but there's a big gap between the shallow creative things like drawing pictures and designing levels in LITTLEBIGPLANET, to doing full on programming. There's very little in between. In my day we had computers like the BBC Micro and the Sinclair Spectrum which you could tinker with and make quite simple programs, and they can easily engage you."

In addition to schools, Raspberry Pi could also be released as a commercial product, to retail. Braben says, "We've been talking about that. We'd have to charge for it. What we are considering is that we would charge a slight premium to help subsidize the [educational] part."

Who does he think would buy such a thing? "Me! I'd love one. There are lots of people who'd want to just use it as a gadget. Sure, people who are geeks, like me, people who are computer fans. There's no shame in that."

Braben and his team plan to release Raspberry Pi out to beta in a few months time, as a software base begins to emerge. A platform will go online to allow educators to upload and download educational software, all for free. (Braben says he's tired of seeing "amazing, beautifully coded programs that are only ever used by the teachers who made them.")

Next, he goes knocking on the door of the government and corporate sponsors to fund releasing the computers to an entire year of school-kids—that's about 750,000 students in the UK. (It would be great to see an appropriate brand step up to help fund this important work.)

Perhaps the biggest challenge is persuading schools to add computer science to their study courses, but he says it's in their benefit. Most schools use horribly out-dated PCs that are constantly breaking down, and require expensive sys-admin care. Raspberry Pi can just reset, and could be used to teach simple IT courses as well.

"The real problem is that kids are getting engaged as consumers of electronics but they are not getting engaged as people who use them to create," he says. "And I think that's a loss."

It doesn't matter to Braben what they create, although the gamer in him would like to see games being made. "I would hope some of them will make games. Some of them might make financial software. Even better, maybe some will make things that we've never seen before."

You can help by visiting www.raspberrypi.org or following Braben on Twitter. (@DavidBraben)

—Colin Campbell

atari 810 micro SD drive

\\\ Older disk drives are not the most reliable storage format, but they certainly have their retro charm. Someone who goes by Rossum on the internet has taken the form factor of the Atari 810's floppy drive and miniaturized it, making a microSD-compatible drive that recognizes the vintage computer's emulated files.

"Overwhelmed by a recent wave of nostalgia from playing ZORK for the first time in 30 years," says Rossum, "I have built a working model of an Atari 810 that uses 8Gb microSD cards instead of 5 1/4 inch floppies to emulate up to 8 drives."

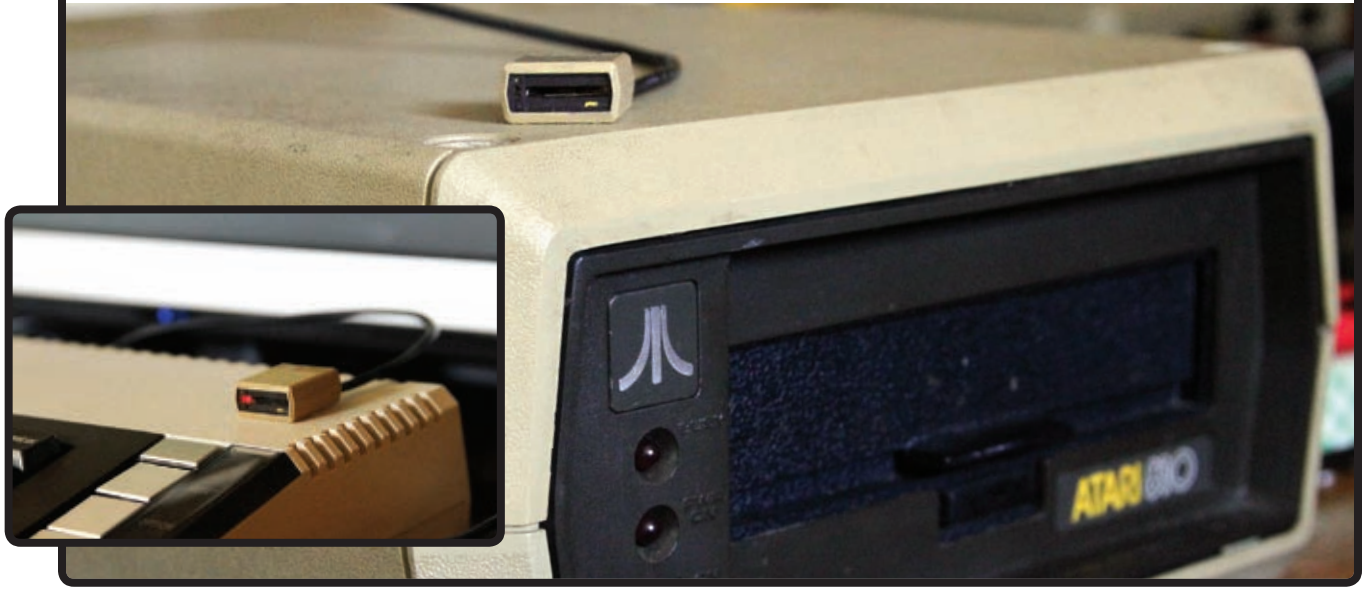
With an 8 gig micro SD card, you can store some 90,000 810 discs, Rossum figures, at an insane fraction of the physical size of the original drive. "The hardware is pretty simple," says Rossum. "A LPC1114 microcontroller, a microSD slot, a 3v3 regulator, a LED and some caps."

The enclosure was created in a 3D modeling program and then manufactured by 3D printer Shapeways, and was hand painted to emulate the colors of the original.

So how does it work? "The microcontroller code emulates up to 8 Atari drives," Rossum begins. "At power on it checks for a microSD card, mounts a Fat16 or Fat32 file system and scans the card for .ATR and .XFD disk image files commonly used with Atari emulators. It also looks for XEX files which are Atari executables, another emulator mainstay. The code then 'inserts' the BOOT. RUR image into drive 1 and waits for the Atari to start sending commands during bootup."

Though this pet project may not be as modern-facing as the recently announced Commodore computers, it actually works with the original device, which, so long as data is properly stored, means you can use your Atari 810 for years to come, regardless of whether the standard drives may fail you. Read more about this, and other interesting projects from Rossum, at http://rossum.posterous.com.

—Brandon Sheffield



cultured mag volume 1 released

\\\ Michael Brown, a student at San Francisco State University, has put together an independent magazine which focuses on bringing the process behind video game development to everyone—more specifically, those with no prior knowledge of the industry. This ambitious solo project had Michael interviewing, writing, and designing the magazine entirely by himself, with various journalist advisors, including some from Gamasutra.com. This first issue includes interviews with a number of notable developers: Dave Grossman of Telltale Games, Nathan Vella of Capybara Games, and Zack Karlsson of Double Fine Studios. Also highlighted is the Game Developers Conference, with discussions on both the history of the show, and how it helps the industry. Cultured Mag is available free for iPad and digital download, and print copies are available for $13.50 from www.michaelrbrown.com.

—Jade Kraus

UNREAL TECHNOLOGY NEWS

BY Mark Rein
Epic Games, Inc.

UNREAL ENGINE 3 FUELS TRANSFORMERS SUMMER BLOCKBUSTER

High Moon Studios has been using Unreal Engine 3 for more than five years now. The studio is creating its second straight *Transformers* game for Activision and Hasbro, although *Transformers: Dark of the Moon* is actually its first Hollywood-licensed adventure. Last year's *Transformers: War for Cybertron* was an original game not set within the timeline of the films.

High Moon knows a thing or two about Hollywood properties, having created 2008's *The Bourne Conspiracy* based on Robert Ludlum's books and the hit film franchise. For its second *Transformers* game, the studio has worked with Michael Bay and his team to mold a prologue story that will bring gamers up to speed on the events that occurred before the start of the big summer movie.

"We are telling a prologue story, rather than the normal model of 'see the movie, play the movie,'" said Terry Spier, senior designer at High Moon Studios. "This affords us many creative freedoms. Our game fits between the second and third movie, covering the events that set everything in motion for the new film. We were given the chance to explain things that the movie does not, while seamlessly weaving our own fiction to support it."

The new *Transformers* game was created with a core team of about 40 people, thanks to the implementation of UE3. The studio has become well-versed in making the most of the game engine.

"Understanding the pipeline for asset creation and iteration has been critical for us," explained Spier. "Also having a good grasp on just how big our levels could be helped us avoid many time-wasting mistakes. It's been fantastic working with Unreal Engine 3 technology. The engine gives us the ability for great visual richness, which is a pillar for us."

Spier said that tools like Unreal Matinee and Unreal Kismet put a lot of power into the hands of the studio's game creators.

"Our cinematic artists and designers have the ability to prototype quickly, often times without the need of code support," Spier said. "Being able to work in a way that is so agile, to get up and running with prototype game play, is such an advantage. We are able to quickly identify if an idea is fun without impacting too many people. Many of our boss battles were approached in this manner, using Matinee."

The studio's familiarity with UE3 over the years allowed the *Transformers* team to focus on some of the newest tools that were developed for *Gears of War 3* like Unreal Lightmass and Unreal Swarm.

"We were able to utilize Lightmass for the first time on this project," said Spier. "Global illumination gave us more realistic lighting than we could achieve without it. It helped give our artists the ability to create the stunning environments you will see in *Transformers: Dark of the Moon*. To go along with Lightmass, we also used Swarm. Swarm distributes the processing required to bake lights over our entire network, saving us precious time."

With its strong UE3 foundation, High Moon was able to implement a custom destructible creation system for the new *Transformers* game.

"With giant *Transformers* parading around, we knew we needed a system to quickly create destructibles and get them in the game," explained Spier. "Inside the system we have the ability to fire off scripted events, making the lives of our designers much easier. We have also been able to create custom shaders that integrate with UE3's native material system to allow us to create spectacular visual effects like the clouds in our Starscream flying level."

All of this technology allowed the team to focus on making a fun gameplay experience. UE3 also made the sharing of assets from the Paramount Pictures film a seamless and quick experience. Spier and his team utilized everything from character models and set photos to the script in order to tell this original interactive story.

Dark of the Moon's single-player campaign focuses on letting players jump into the action as Autobots and Decepticons through various points in a prologue storyline. Spier said each game level is specifically crafted for the Transformer moving through it to provide a robust experience. Since High Moon contains a group of rabid *Transformers* fans, gamers can expect to see iconic battles between some of the most popular Transformer heroes and villains.

The new *Transformers* game will also offer online multiplayer modes where fans can play with or against friends. And for the first time in a *Transformers* game, players will have the chance to play online as their favorite named iconic Transformer.

"My Optimus Prime against your Optimus Prime, anyone? See you online," said Spier.

The time constraints that Hollywood-licensed video games put developers under and the intense scrutiny that gamers place on these games make it important for studios like High Moon to be able to focus on the gameplay and rely on a strong foundation of technology to allow team members to be creative. Spier said that UE3 has empowered his studio to create not one, but two *Transformers* games that the team -- as fans of the IP -- enjoy playing. And critics, and the public, heralded the studio's last game as the best *Transformers* game ever made.

Canadian-born Mark Rein is vice president and co-founder of Epic Games based in Cary, NC. Epic is the creator of the mega-hit "Unreal" series of games and the blockbuster "Gears of War" franchise. Epic's Unreal Engine 3 has won Game Developer magazine's Best Engine Front Line Award five times along with entry into the Hall of Fame. UE3 has won three consecutive Develop Industry Excellence Awards.

Follow @MarkRein on Twitter.

UPCOMING EPIC ATTENDED EVENTS

E3 Expo
Los Angeles
June 7-9, 2011

Develop
Brighton, UK
July 19-21, 2011

Comic-Con
San Diego, CA
July 21-24, 2011

GDC Europe
Cologne, Germany
August 15-17, 2011

Please email: mrein@epicgames.com for appointments.

UNREAL TECHNOLOGY

EPIC GAMES

WWW.EPICGAMES.COM

TOP 30 DEVELOPERS

OUR PICKS FOR THE TOP GAME STUDIOS OF 2010

As of 2010, the move toward smaller studios making big waves is no longer speculative. The ability of smaller teams to better adjust to the rapidly-changing face of games has been proved multiple times over, with hits like MINECRAFT, ANGRY BIRDS, and LIMBO tearing up the proverbial charts. But the year had its share of blockbusters as well, not to mention those gems that fell through the cracks. In this list, we celebrate 30 teams who did something significant to distinguish themselves from the rest.

ACCESS GAMES
OSAKA, JAPAN

Access' cult favorite DEADLY PREMONITION has clumsy combat, PS2-level graphics, and maddeningly long sequences of driving from nowhere to a slightly differently textured nowhere. But lift the curtain a bit and you will see a lively game world, where people go about their daily business, regardless of player interaction.

The game's dialog is so perfectly, imperfectly written, that it feels as though it belongs in a category all its own. It also has one of the most intelligent narrative framing devices, allowing the main character's alternate personality to take lead in a way that other games have certainly attempted, but never succeeded at (if we say more, we may spoil something).

DEADLY PREMONITION took five years and many near-cancellations to put on shelves, and is the kind of game that nobody makes anymore. As the industry changes and shifts, who knows if they ever will again. But after game director Swery's compelling GDC talk on creating memorable characters in games, we can only hope the momentum will continue.

AVALANCHE STUDIOS
STOCKHOLM, SWEDEN

JUST CAUSE 2 is one of the first games to elevate explosions into an in-game currency. With lush graphical detail in a National Geographic photo-spread of a world, it was one of 2010's most enjoyable places to visit—both as a tourist and as a terrorist.

Avalanche somewhat stealthily created 2010's most open-world console game, which allowed incredible freedom through the main character's superhero-style tether that allows him to latch on to and interact with most surfaces, whether they are static, mobile, or live enemies. This was all possible across miles and miles of explorable terrain, with secrets aplenty. (And incidentally, JUST CAUSE 2's tether mechanic gave the makers of Spider-Man games a very difficult benchmark.)

Players can thus accomplish some of the most ridiculous feats of emergent gameplay we've seen. Want to have a tug-of-war between a tank and a motorcycle? You can! Want to "fly" a boat? Not impossible. The "go anywhere, do" anything nature of the game puts Avalanche on our list.

BIOWARE MASS EFFECT TEAM
EDMONTON, MONTREAL, CANADA

For a developer with deep roots in classic PC-based role-playing games like BALDUR'S GATE and NEVERWINTER NIGHTS, over the past several years BioWare has shown an increasing understanding of what kind of gameplay works for console players, without

losing the rich storytelling that its PC games are known for.

MASS EFFECT 2, creatively led in Edmonton but supported by Montreal (which also did the DLC) sports a sprawling universe, unexpected and clever story developments, and a more memorable ending, which has led this RPG to find fans on both sides of the PC-console divide. BioWare's commitment to the written word (the company has an entire department dedicated to the craft) garnered the studio a Best Writing award at the Choice Awards in 2010.

BioWare has yet again crafted an experience infused with memorable characters, planets and events that feel as though they exist within a consistent universe—an achievement in any game.

BLIZZARD ENTERTAINMENT

IRVINE, CA

Blizzard is not exactly known for shipping products frequently, but in 2010 managed to ship three. Among those was the long-awaited RTS sequel STARCRAFT II, alongside the world-beating MMO expansion pack WORLD OF WARCRAFT: CATACLYSM, and more subtly but perhaps most importantly, a completely new implementation of its Battle.net service. The changes to Battle.net transformed it from a simple matchmaking to a complex and comprehensive service that covers the networking and community functions for the company's slate of games, present and future.

While this didn't go off without a hitch (the company ran into a bit of controversy with its plans to show users' real names, which has since been modified), Blizzard has made a concerted effort to develop a solution which is in line with the popularity of its games and the future of the market—a ferociously difficult task.

Its games, as always, seem to hit their targets creatively—conservative, perhaps, but polished and massively popular. Blizzard was firing on all cylinders in 2010.

CAVE

TOKYO, JAPAN

In the days of yore, Japanese game companies were known for their excellent arcade-style titles, with vibrant colors, larger-than-life sound, and light gameplay. With the new generation of consoles, that drive seems to have been lost. Cave is one of the only companies that has stuck to its guns and continued to work within that arcade-y arena, while also embracing downloadable platforms, when most other companies in the region have been hesitant.

In 2010, Cave found success on iPhone and XBLA, to the surprise of many, with games like DODONPACHI RESURRECTION, ESPGALUDA II, and DEATHSMILES. The company has made touch-screen controls work for games that require precise control and speed, and has brought updated versions of its original arcade shooters to downloadable consoles. And that's saying nothing of the original and occasionally experimental titles they have made for downloadable services. Cave, one of the most hardcore, non-casual companies out there, has found a niche that works, which has fueled its western expansion more efficiently than more instantly-accessible companies have been able to.

CHUNSOFT

TOKYO, JAPAN

Chunsoft, whose catalog of celebrated "sound novels" includes 428, MACHI, and other titles passed over by U.S. publishers, somehow managed to bring 999 to the States via Aksys Games, thereby releasing one of 2010's more underappreciated, but worthy titles.

In this anxiety-filled M-rated DS game, players find themselves kidnapped and trapped on a sinking ocean liner, forced to work with eight other similarly unfortunate characters to survive deadly puzzles. 999's Choose Your Own Adventure-style story progression might seem simple at first, but players soon discover their decisions determine whether they (and their companions) live or die.

The sound and visual novel genres have never made much of a splash in the West, but for bringing its expertly-crafted stories to a broader audience, we place Chunsoft on our list.

CRITERION GAMES

GUILDFORD, ENGLAND

Criterion trimmed away the fat of its previous title, BURNOUT PARADISE and returned to the schizophrenic NEED FOR SPEED series' first principle of cops vs. robbers for this startling re-imagination with NEED FOR SPEED HOT PURSUIT . But it's in the introduction of Autolog, an always-on competitive social network overlay, that this release became a game-changer, evolving the humble leaderboard into an obsessive, prodding competitive pursuit.

Autolog is a suite of connected features that automatically picks out what your HOT PURSUIT friends have been up to—latest track times on the "Speed Wall," new photos and comments, and so forth—and encourages you to immediately one-up your buddies. This system gives the game some semblance of socially-driven stickiness by providing players with constantly-changing goals that are set by online friends.

After using something like Autolog, you realize that the head-to-head multiplayer and basic leaderboards that are prevalent in racing games today are pretty archaic as we enter the sixth year of the current console generation.

CROWDSTAR

BURLINGAME, CA

Crowdstar is one of the bigger players in the social game industry, largely through its extensive support of existing titles. HAPPY AQUARIUM and HAPPY ISLAND were its breakout hits in 2009, but in a climate where games are disposed of as soon as the revenue flags, the company has shown that supporting an existing user-base is one of the best ways to build a business.

In 2010 Crowdstar launched its female fashion and culture-oriented social game IT GIRL, while also announcing that Bon Jovi would be using the company's existing social games to sell his digital music. His Greatest Hits album was bundled with Facebook credits and virtual goods for the three aforementioned titles. With more than 50 million monthly active players, Crowdstar's offices are getting quite, well, crowded, as the company expands rapidly to meet demand, even as it turned down an acquisition attempt by Microsoft.

DOUBLE FINE

SAN FRANCISCO, CA

How the heck did Double Fine manage to survive two consecutive console games that cost a lot, and preformed rather poorly at retail? Strength of character, I'd say. The company has managed to soldier on without significant layoffs, and without losing its creative streak. And in 2010, it seems the company has finally found its footing. Creative head Tim Schafer is a hilarious and talented individual, but he can't do everything himself. To this end, the company encouraged several department leads to make their own downloadable games, which would be quicker, smaller, and more targeted than Double Fine's prior sprawling epics.

2010's COSTUME QUEST was the first of these, and proved the model's merits. As subsequent games (such as STACKING) emerged, it became evident that this was going to work for Double Fine in the longer term. Smaller games that focus on smaller ideas fit the detail-oriented studio quite well, and it's not every company

that can survive some serious setbacks and come out of it more critically successful than ever. Whether this translates to financial success for the company is another matter, but our fingers are collectively crossed.

FRICTIONAL GAMES
HELSINGBORG, SWEDEN

Among recent big-name video game entries that have a horror slant, whether it's DEAD SPACE, RESIDENT EVIL, or FEAR, there's been an arms race resulting in protagonists who are armed to the teeth with anything from assault rifles to bazookas.

In the world of Frictional Games' AMNESIA: THE DARK DESCENT, you don't have grenade launchers, M16s, or shotguns. And even if you did, they would probably be of little use. The gruesome creatures within dreary Brennenburg Castle possess a ghost-like, ever-present supernaturalism that makes them terrifying, as if they could appear at will to menace the protagonist.

Through impressive sound design and arresting visuals, the addled Swedes at Frictional have put together an experience that, even with no enemies in sight, can bear down so hard on players that they have to step back and collect their own sanity to remind themselves, that this is only a game. It's no wonder Frictional walked away with IGF awards in Audio and Technical categories.

FRONTIER DEVELOPMENTS
CAMBRIDGE, ENGLAND

David Braben's Frontier Developments is better known for its hardcore ELITE series than for casual titles, so it was a bit of a surprise to see the studio come up with KINECTIMALS, a launch title for Microsoft's Kinect motion-sensing system—a peripheral that recognizes players' movements and translates them into game actions without the use of any sort of hand-held controller. But a good developer is a good developer, and KINECTIMALS is a shining example of a children's game done right. Players are given a jungle cat to play with, but rather than being a simple raising sim, the game takes you on an adventure through an unexplored island, where you learn new skills alongside your pet.

The game has a real respect for the player, regardless of age, treating the user as an equal partner in play, rather than talking down and giving instructions. It also uses clever tactics to create a bond between owner and virtual pet, which is no easy task (the gorgeous animation helps). KINECTIMALS was, for our money, the most successful narrative-driven Kinect product in 2010, and one of the best kids games to boot.

HALFBRICK STUDIOS
BRISBANE, AUSTRALIA

In the years running up to 2010, most of the news out of Australian game studios was very bad. Massive layoffs and studio closures were the norm. But necessity is the mother of invention, and as licensed game work dried up, the smaller studios found ways to become more successful than ever before. Halfbrick is one such studio, achieving massive sales with FRUIT NINJA on iOS, and turning that success into a slew of new titles for downloadable platforms in 2010.

On top of FRUIT NINJA, the company released MONSTER DASH, AGE OF ZOMBIES, and a host of others alongside its irreverent XBLA title RASKULLS. With Australian developer Firemint snapped up by EA, the pressure is on Halfbrick to retain that indie spirit for the Aussies.

HARMONIX MUSIC SYSTEMS
CAMBRIDGE, MASSACHUSETTS

The Kinect launched with massive promise and Harmonix made one of the major standouts. DANCE CENTRAL not only created a very compelling dance experience, it also one-upped Microsoft's own Kinect interface, finding a much more intuitive method of browsing through and confirming content.

In the same year, Harmonix also managed to put out ROCK BAND 3, the "last man standing" in the peripheral-based music genre, now that the GUITAR HERO franchise is no more. This latest entry refines and expands the ROCK BAND concept in compelling ways, such as "pro mode" songs, and instruments designed to let players transition from mimicry to true performance.

But just as important is, again, the complete and total refinement of interface and copious customizability of play modes. Harmonix puts real thought into making these often-overlooked elements simple and accessible, putting the studio at the forefront of U.I. design in the game industry.

MOJANG
STOCKHOLM, SWEDEN

The basic human instinct to practice survival through play is woven into the DNA of all video games, but in MINECRAFT, IGF Grand Prize and Choice Award Best Debut Game winner, it's hewn into the very rocks that make up its randomly generated world. You are deposited into a field, your only task to create shelter for yourself from the beasts that rise at sunset. It's survival horror in its purest form; there's no need for cinematic shocks to punctuate the creeping sense of dread as you race to fashion tools from gathered wood and set about digging a hole in which to cower.

MINECRAFT's brilliance is found in the way in which goals, almost all self-made, unfurl in new directions with the passing of time. By giving the player the exact tools they need to express themselves, MINECRAFT is perhaps the closest we have to a true god game.

Outside of gameplay, the innovative model of charging for the game beginning with a compelling alpha build has fueled to the game's 10 million paid downloads (and counting), and has allowed the company to grow organically. MINECRAFT has irrevocably changed the very landscape of gaming, even as we have irrevocably changed its own landscape in kind.

OBSIDIAN ENTERTAINMENT
IRVINE, CA

Obsidian had a large task to accomplish—take what worked in FALLOUT 3 and improve on it. It seemed hard to top the vision of the U.S. capital nearly melted to ash, but the portrayal of a ruined former hub of American decadence in FALLOUT: NEW VEGAS, is just as fascinating.

The distinct influence on the game world of Rome's tragic story of out-of-control power is well thought out, and the player has the freedom to choose to make a celebration or a condemnation of all kinds of excess. As the game starts to draw a story of factions warring for control, the loyalty system in which the player participates provokes lots of thought on the nature of power in a world with laws upended.

Much has been said about the game's bugs, but in a year of big blockbusters, a project with this vast amount of choice, improved writing, and a multilayered story came much appreciated.

PLATINUM GAMES
OSAKA, JAPAN

Platinum Games in 2010 put out titles from two of Japan's best and brightest: Shinji Mikami (of RESIDENT EVIL fame) released VANQUISH, and Hideki Kamiya, known for DEVIL MAY CRY, put out BAYONETTA. Both games had interesting systems, a curious premise, and slick execution, positioning Platinum at the forefront of the Japanese industry.

BAYONETTA seems as if it's trying hard not to be liked: the game is garish and implausible, bursting with filigree, butterflies, and senseless

conglomerations of religious iconography. And yet, somehow it all works as absurdist fiction, like a glorious pulp movie, with its excesses as celebratory as a Tarantino film. It wages such a calculated, eloquent war on taste that it creates its own style, riotously pleasing to play.

Meanwhile, VANQUISH takes speed and polished visuals (literally, everything in the game glitters with a dull gloss) and turns them into frenetic gameplay, that some have said sets a new standard for third person shooters.

PLAYDEAD
COPENHAGEN, DENMARK

Ask any Xbox player which Live Arcade titles were worth checking out in 2010, and it's a fair bet they'll mention LIMBO, alongside words like "incredible," "atmospheric," and "unfair." This visually-arresting Danish labor of love took over five years to develop, but proved its long dev cycle was warranted—the game walked home with the Best Visual Arts award at the Choice Awards, winning out over games boasting high-fidelity 3D.

LIMBO enjoys nothing more than killing you over and over again in the goriest manner possible, while laughing at your crippled remains. Fortunately, whenever death does befall our young hero, he is always placed back down just before the perpetrator, hence the trial-and-error feel to many of the puzzles can seem more humorous than frustrating. Indeed, while playing the game is fun, watching someone else fail time and time again is just as entertaining.

QUANTIC DREAM
PARIS, FRANCE

The hugely ambitious HEAVY RAIN has been critically divisive—everyone agrees on the risks it took, but not everyone agrees that it succeeded. That Quantic Dream tried and continues to try to push games forward as a medium is to the studio's credit. One of the less-discussed successes is in the interface, through which any action can be accomplished using the same controls, from firing a gun to tucking in a child.

Most games that try to sprinkle some sentiment or levity by adding child-tucking-in or woman-kissing hack it in. You push a button. An animation plays. You're just tapping X, you're not really "doing" it. In contrast, what you're usually doing, in those games, with tremendous depth and nuance, is killing things.

No, shooting a gun in HEAVY RAIN isn't as satisfying as the shooting in GEARS OF WAR. But, in HEAVY RAIN, kissing a woman or tucking in a child is as satisfying as shooting a gun. This elevation of all actions to the same plane is essential to what makes the game a success, and is core to Quantic Dream's philosophy.

ROCKSTAR SAN DIEGO
SAN DIEGO, CA

You're half-dead, under heavy fire from the Mexican Army and running out of ammunition. In desperation you call your horse, who gallops over the ridge with the sun rising at its back through the sage. You swing onto the saddle and ride like hell until you reach the next safe town.

That scenarios like this are so common in Rockstar San Diego's RED DEAD REDEMPTION speaks volumes to the reverence with which the company treated the Western genre, effectively ending the game developer legend that "cowboy games don't sell." RDR presented a living open world that players hadn't really been exposed to in games before (at least not this effectively), and the result was a slew of accolades (including Game of the Year in the Choice Awards) and massive sales.

The game does a fantastic job of making the player feel like they're making significant choices, forming relationships (especially with horses), and discovering locations on their own, when in fact their options are limited to story and subquest nodes—that kind of trickery is to be praised, since the ultimate goal is entertainment. With RDR, Rockstar San Diego proved the company can have a great year without a GRAND THEFT AUTO.

ROVIO MOBILE
ESPOO, FINLAND

Has this Finnish team cracked the secret of succeeding on the app stores? With millions of units sold across iOS and Android platforms, as well as a cartoon deal in the works, signs point to "yes." Disappointingly for many, the secret may not be that easily repeatable as there is still a lot to learn from the endless chart-topping success of ANGRY BIRDS.

One major lesson is that success doesn't come quickly. Rovio poured effort into making the title successful, and learned that constant updates drive its continued popularity. If people keep playing a game, they'll keep talking about it. The team also capitalizes on holidays to keep the game fresh and in people's faces.

In short, Rovio has learned that even casual games need a great deal of attention. The interesting question is precisely where and when that attention belongs—and that can only be answered by each individual team.

SIGNAL STUDIOS
BOTHELL, WASHINGTON

TOY SOLDIERS is Signal Studios' runaway hit for XBLA, selling some 500,000 units since its release in March 2010. The game's unique mix of strategic command and third-person action brought the game to prominence, and its "cute" destruction, which shows no blood, kept TOY SOLDIERS family friendly, while still being exciting.

With multiple modes and 50 different controllable units, the game has shown to have a certain longevity, which Signal has supported with massive downloadable content, continued into 2011. On the back of this success, Signal has begun to license its SigEngine and also looks to port TOY SOLDIERS to PC.

STERN PINBALL
CHICAGO, ILLINOIS

Who keeps Pinball alive if not Stern? No-one, that's who. Gary Stern has been running the American pinball market since 1986, when he ran Sega/Data East's U.S. divisions. Stern Pinball has been the only major manufacturer of pinball games in the world since 1996, making larger runs for operators in pizza parlors like the old days, and smaller runs for private consumers to own themselves. The company on occasion will even re-run older popular pinball cabinets, effectively keeping the entire industry alive by itself.

In 2010, Stern released *Big Buck Hunter*, *Iron Man*, and *Avatar* cabinets, all designed by John Borg, who seems to be the only working non-virtual pinball cabinet designer around. Though Stern's pinball tables may have gotten less visually elegant with time, using glossier and more plastic-y material, they are no less enjoyable, and Stern deserves a place on our list for continuing to fight the good fight.

TEAM MEAT
USA

Edmund McMillan and Tommy Refenes at Team Meat didn't expect SUPER MEAT BOY to be quite the XBLA and Steam hit that it's become, after the extreme difficulty they had making it. But it worked out in the end, as this tough-as-nails platformer is so chock-full of content that it'd be a bargain even at boxed retail price, with alternate versions of each level, retro throwback mini-stages accessible through warp zones, and unlockable characters and modes.

The thing SMB (not an accidental acronym, we're sure) gets most right is the control,

which in spite of using the 360 analog stick, manages to feel precise and sticky, even in a 2D environment. This means that no matter how difficult the game may be, you always know it's your fault when you die, which can minimize frustration (well, a bit at least).

McMillan's irreverent sense of humor doesn't hurt either, extending even to the point of baiting PETA into making a parody of his game, which he then turned around and parodied himself.

TREYARCH
SANTA MONICA, CA

2010 marks the year that Treyarch came into its own as a studio. After years of ports and substandard "off-year" CALL OF DUTY games, the company finally found its footing, funding, and positioning with CALL OF DUTY: BLACK OPS, which was ambitious in concept, and stuffed full of cleverly hidden extra content.

You can knock BLACK OPS' fist-bumping machismo, but the game knows what it is: an unapologetic, violent shoot-fest. An intriguing mind-bending story that plays with history and over-the-top multiplayer means BLACK OPS will keep players busy till the next CALL OF DUTY entry. Oh, and it generated over $1 billion in sales in 2010, too.

Treyarch was previously known as a bit of a grindhouse studio, churning out titles at an incredibly quick rate, which also lead to significant employee churn. But after a refocus, the studio's tactics seem to have changed, with Activision allowing greater creative freedom and direction, and the CoD franchise now lives with Treyarch after the dissolution of Infinity Ward.

UBER ENTERTAINMENT
KIRKLAND, WASHINGTON

Amidst the sprawling demo floor of last year's first PAX East conference in Boston, there was one table that seemed to have a constant throng of people surrounding its demo units. That table belonged to Uber Entertainment, and it's easy to see why that early display of MONDAY NIGHT COMBAT attracted such attention. The game's squad- and class-based shooting matches, cel-shaded aesthetic and heaps of personality call to mind Valve's TEAM FORTRESS 2 and classic Midway games, but the addition of upgradable robot sentries, defensive turrets, and player abilities make each battle resemble the push-and-pull of a football match more than a deathmatch.

After attracting 225,000 players in the few months after its August release, Uber has stoked the community fires with updates that provide new maps and bug fixes, as well as constant balance tweaks handled through Microsoft's Title Managed Storage system, rather than as major DLC. Not bad for a company that started as a team of six working out of a bedroom office.

UBISOFT ANNECY
ANNECY, FRANCE

Ubisoft Annecy opened in 1996, and was previously named Ubisoft Simulations. As multiplayer has taken an increasingly important role in games, so too has the Annecy studio taken on a larger role within Ubisoft. The team has worked with Ubi branches in Montreal and Shanghai on SPLINTER CELL franchises, using extensive playtesting to hone the multiplayer experience for multiple titles.

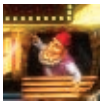
Most recently, with ASSASSIN'S CREED: BROTHERHOOD, Ubisoft Annecy demonstrated that it isn't the guns that make multiplayer, with its own original and engaging competitive mode which places multiple assassins in a sandbox environment, awarding points and evolution across multiple game styles. As the studio moves forward with multiplayer for ASSASSIN'S CREED: REVELATIONS, we can only expect bigger and better things.

WAYFORWARD TECHNOLOGIES
VALENCIA, CA

WayForward has been making clever and colorful 2D games since the Game Boy era, notably with its late-entry GBC original property SHANTAE. WayForward hasn't given up on that core of 2D expertise, and continually tries to bring SHANTAE to a new audience. In 2010, the company found a way, releasing an updated and gorgeously animated entry to the series on DSiWare.

In the same year, WayForward released BATMAN: THE BRAVE AND THE BOLD for DS and Wii, a charming 2D brawler with loads of content. In a world where BATMAN: ARKHAM ASYLUM doesn't exist, this would have been the best Batman game around (but I'm pretty sure we're all glad we don't live in that world). WayForward continues to raise high the banner of 2D gameplay, moving forward with a 2D implementation of the Thor franchise for DS, among other games. For this, we salute them.

ZEN STUDIOS
BUDAPEST, HUNGARY



A pinball simulation on a computer may not be the most video-game-like experience in theory, but with PINBALL FX2, Zen Studios' expertise in designing tables and the very clever, modular nature of buying/importing new content shines through. New tables are downloaded as add-ons, not as new games, which means players keep coming back to the same core game to get new experiences. Downloadable tables are cross-marketed, as pinball tables often are, with licenses like Marvel, STREET FIGHTER II TURBO, and Rocky and Bullwinkle.

Add to all that best-in-class social integration with regard to individual/friend high scores, plus perfect simulation and compelling gameplay, and you get one of most modular and slick game packages of 2010 (provided you like pinball, that is). The company continues to support its ZEN PINBALL on PlayStation 3 as well, also bringing the franchise to iOS and Android.

ZEPTOLAB
MOSCOW, RUSSIA

Zeptolab was founded in 2008 by brothers Efim and Seymon Voinov (formerly a lead artist at Digital Chocolate in Finland). After an initial game in PARACHUTE NINJA, the company found success with its breakout hit CUT THE ROPE. The game boasted multiple-million downloads in 2010, and also managed to win Best Handheld Game in the Game Developers Choice Awards, as well as a BAFTA.

The company looks to continue its success on the iOS platforms going forward, adding content to CUT THE ROPE while also working on new titles. Zeptolab proves that indie downloadable game development is a truly global business.

ZYNGA EAST
BALTIMORE, MD

The path forward for social games is far from clear. Funding is flowing everywhere, and each company is trying to pave its own road. One studio that has made a great stab at charting the future of social games is Zynga East, developer of FRONTIERVILLE.

Parent company Zynga realized it would need experienced development talent to move the social game industry forward, so it enlisted the help of seasoned strategy developers like Brian Reynolds, who left Big Huge Games to transition into the social space.

FRONTIERVILLE has proved that there is a way forward. The team at Zynga East has found a way to marry not just traditional design ethos, but traditional design talent with Facebook and the newly emerging best practices of social games. Appropriately enough, it was proved by a game with a Wild West theme.

THE **38TH** INTERNATIONAL
CONFERENCE AND **EXHIBITION** ON
COMPUTER GRAPHICS
AND **INTERACTIVE TECHNIQUES**

VANCOUVER
SIGGRAPH2011 MAKE IT HOME

CONFERENCE 7-11 AUGUST 2011 **EXHIBITION** 9-11 AUGUST 2011

Wherever SIGGRAPH travels, HOME is where smart people gather for an irresistible mix of energy, artistry, and the latest ideas in computer graphics and interactive techniques.

VANCOUVER CONVENTION CENTRE

Sponsored by ACM**SIGGRAPH**

WWW.SIGGRAPH.ORG/S2011

JORGE JIMENEZ, JOSE I. ECHEVARRIA, BELEN MASIA, FERNANDO NAVARRO, NATALYA TATARCHUK, DIEGO GUTIERREZ

DESTROY ALL JAGGIES

MLAA ON THE GPU – THE OPEN WAY

You've got your mind-blowing brand-new graphics engine featuring state-of-the-art lighting and shaders ... But looking closer, there are jaggies everywhere! You obviously need some kind of anti-aliasing in order to improve the final image quality, but which solution? Your typical choice would be to use multisampling anti-aliasing (MSAA). But you may already know (or are probably going to face) some of its limitations, such as big performance penalties for high qualities (4x and above), the hardware limitations of current consoles, and the troubles of combining it with multiple render targets and deferred shading. Then there is the inclusion of alpha to cover for proper anti-aliasing of transparencies, artifacts when resolving HDR framebuffers, plus depth-related artifacts when rendering objects after MSAA resolves, and the biggest limitation of all, additional memory cost for MSAA buffers. So, what can you do?

For the past few years, the solution has been to move anti-aliasing techniques to the shader units and apply them as any other postprocess in screen space. This has resulted in a plethora of custom solutions featured in games like S.T.A.L.K.E.R, Tabula Rasa, Crysis, and Brutal Legend (just to name a few). It turns out that all of them share one core idea: edge detection and smoothing. Among these solutions, directionally localized anti-aliasing (DLAA—featured in Star Wars The Force Unleashed 2) shines with its smart use of blurs. Edge detection and smoothing is also one of the underlying ideas of morphological anti-aliasing (MLAA), a technique that was originally published by Intel, and has been getting some buzz on the internet during the past few months. As opposed to those custom solutions, with MLAA this smoothing is done adaptively by taking into account the coverage areas coming from a perceptual re-vectorization of the image. Compared with DLAA, MLAA is more accurate in terms of resemblance to the ground truth MSAA, while DLAA results, although good-looking, often present some blurriness.

MLAA detects edges based on color information. These are then classified according to a series of patterns. The final anti-aliasing is performed by blending each pixel with its neighbors, depending on the previously detected patterns. Intel's CPU code was later followed by the first practical in-game implementation in Sony's God of War III, with great success in image quality. This implementation, running on PlayStation 3 SPUs, has been featured recently in other first-party titles like Little Big Planet 2 and Killzone 3. Concurrently, Kalloc Studios developed another SPU-based anti-aliasing technique for the PS3 version of Saboteur, which is similar in spirit to MLAA. Hybrid CPU-GPU implementations on the Xbox 360 have surfaced in Double Fine's Costume Quest and Stacking. On the PC side, AMD introduced its own proprietary implementation at driver level with the launch of its 6000 series, making it a great choice for a bunch of games that were completely lacking AA. However, their results tend to be blurry, and its activation via driver panel means that even the GUI is processed (something not desirable). While each has its own strengths, they either come from proprietary technology with varying degrees of customization, or are tied to specific platforms, so you may want a more open and universal solution.

In this article, we will discuss how to easily implement MLAA exclusively on the GPU, making its integration in your graphics engine child's play. This integration boils down to a standard pixel shader post-process, allowing for total control of when exactly you want

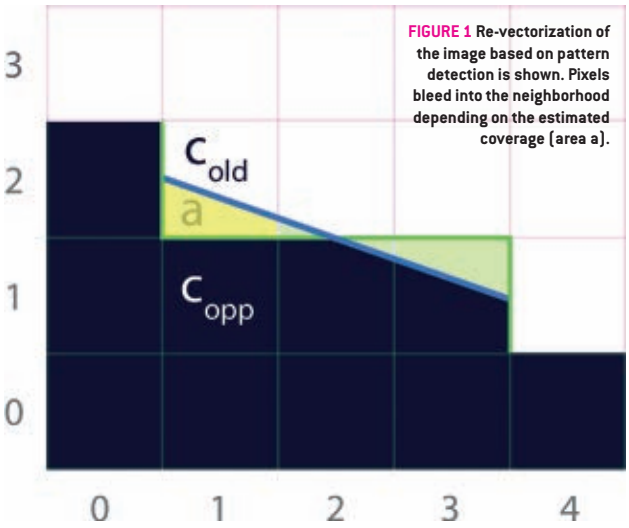


FIGURE 1 Re-vectorization of the image based on pattern detection is shown. Pixels bleed into the neighborhood depending on the estimated coverage (area a).

it to kick in, boosting flexibility and final image quality. This lightweight implementation takes just 0.44 ms, on average, on a mainstream NVIDIA 9800 GTX+ for a resolution of 720p. On the Xbox 360, for this same resolution, it takes 2.6 ms (which can be further reduced depending on the edge detection method used and specific engines). You can run it on any DX9-class GPU (and above), including the Xbox 360. Also for comparison, our implementation runs around 9x faster than AMD's MLAA on a Radeon 6870.

Are you ready to join us in the battle against the jaggies? If so, let's begin with the basics to set the stage for the real action!

FIRST THINGS FIRST: MLAA ABC

/// The main goal of anti-aliasing as a post-process is to work at 1x resolution throughout the graphics pipeline, optimizing memory consumption and sparing processing power. The color buffer you are going to apply AA over is typically going to be 1x. This means that you will not have subpixel information at hand. In that case, how do you know which pixels you have to use for the final blending? The answer is in the make-up of our own eyes, which assumes that on the edges, things behind one object will have the same color as their background neighborhood. Therefore, by properly mixing pixels on the edges, you can obtain a nice perceptual anti-aliasing.

As mentioned before, MLAA searches for specific patterns among the detected edges in order to perform the final blending. This process can be viewed as a re-vectorization of the original 3D scene. Take a look at Figure 1, which is a close-up of a dark triangle (already rasterized) over a white background.

Green lines mark the discontinuities that create the edges, which in turn define the pattern type. The basic pattern types are U-shaped, Z-shaped, and L-shaped, but there are many possible variations given that each pattern can be found flipped and/or rotated. The blue line is the estimated re-vectorization for that pattern. So, if this blue line represents a perfectly anti-aliased edge, we'd just have to bleed the black pixels into the areas of the upper part of the pattern, and the white pixels into the areas of the lower part. Thus, we can calculate the green and yellow areas and use them as weights for the final blending, like so:

$$c_{new} = (1 - a) \bullet c_{old} + a \bullet c_{opp}$$

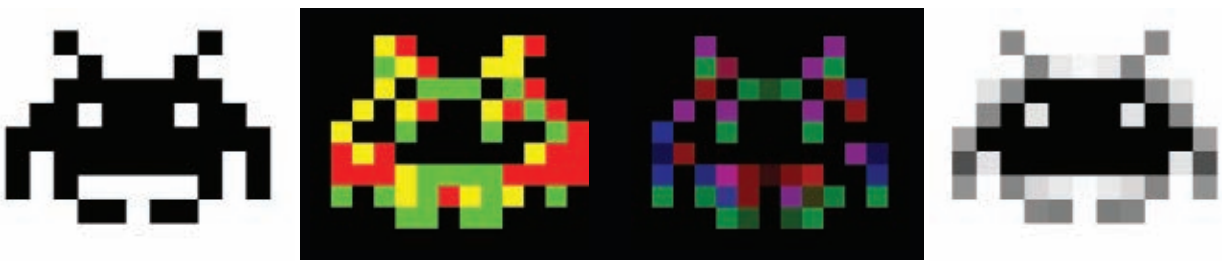
Here, c_{old} is the original color of the pixel, c_{opp} is the color of the pixel on the other side of the edge, and c_{new} is the new color of the pixel resulting from the bleeding of c_{opp} into c_{old} amount a (the area shown in yellow). The value of a approximates the coverage of the re-vectorized polygon, and is a function of both the pattern type of the edge and the position of the edgel (edge pixel) within the edge. The pattern type is defined by the crossing edges of the current edge, that is to say edges which are perpendicular to the current one, and thus define its ends (represented by the vertical green lines in Figure 1). In order to save processing time, this area a can be pre-computed and stored as a two-channel texture, as we will explain later on.

FROM LISTS TO TEXTURES

/// Intel's CPU-based implementation searches for specific patterns (U-shaped, Z-shaped, and L-shaped) that are then decomposed into simpler ones, an approach which would be impractical on current-generation GPU architectures , as this would involve complex dynamic branches. We have observed that the pattern type, and thus the anti-aliasing to be performed, only depends on four values (the four possible crossing edges), which can be obtained for each edgel with only two memory accesses. This way, Intel's algorithm is transformed in such a way that it uses texture structures instead of lists. Furthermore, our approach allows handling of all pattern types in a symmetric way, thus avoiding the need to decompose them into simpler ones and the use of a complex, branchy pattern-matching algorithm at run-time.

Our algorithm consists of three passes, the complete pipeline of which is shown in Figure 2. Starting from an original source image (with aliasing present—in Figure 2 left), we perform edge detection in the first pass. This yields a texture containing edgels (Figure 2, center-left). In the second pass, we process each edgel in the edges texture, generated in the previous pass, obtaining the corresponding blending weights of each pixel adjacent to the edgel being smoothed. To do this, we first calculate the distances from each edgel to the end of the lines to which it may belong (of which there are two: horizontal and vertical). Then, the crossing edges are fetched and used, together with the distances, to query the precomputed area texture (we can think of that as a look-up table), which returns the corresponding blending weights (Figure 2, center-right). The third and final pass involves blending each pixel with its 4-neighborhood using the blending weights from the previous pass to obtain the final anti-aliased image (Figure 2, right).

FIGURE 2 COMPLETE MLAA PIPELINE



ORIGINAL IMAGE | EDGES TEXTURE | BLENDING WEIGHTS TEXTURE | ANTI-ALIASED IMAGE

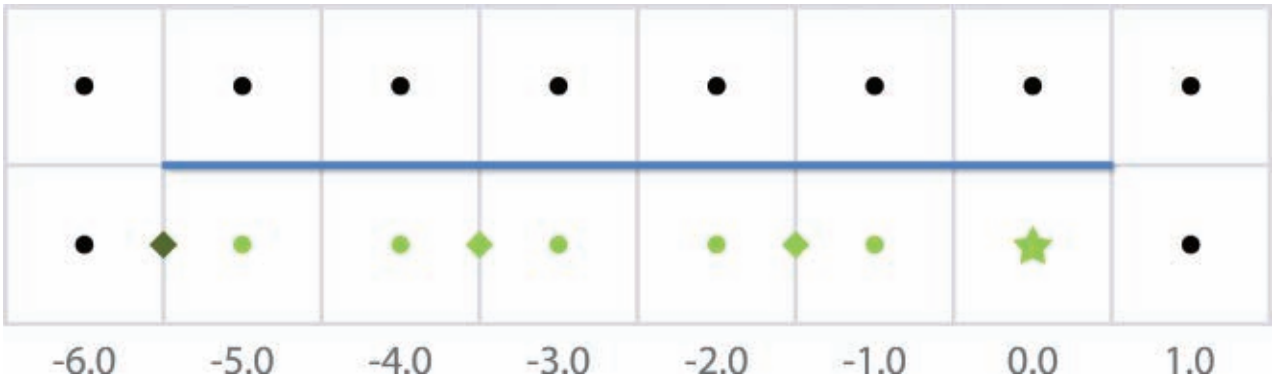


FIGURE 3 An example of a bilinear filter trick for distances search.

You may be thinking, why not just merge the last two passes into one? The reason is simple: Doing them separately allows you to spare calculations, taking advantage of the fact that two adjacent pixels share the same edgel. Put another way, you can think of the second pass as performing the pattern detection and the subsequent area calculation on a per-edgel basis, instead of on a per-pixel basis (which would be the naïve approach). This way, in the third pass, the two adjacent pixels (which share an edgel) will fetch the same information.

Since only a few pixels will need to be smoothed, we can add the stencil buffer into our recipe to apply the second and third passes only on the pixels which contain an edge, considerably reducing processing time. In the first pass, we store "1" in the stencil buffer for all pixels that contain an edge. Then, in the following passes, we just process pixels that have a stencil value of "1." As the destination buffer will not likely contain the color image, we probably won't want to just copy the processed edges into it in the last pass, as that would give us a black image with only anti-aliased edges on it. Thus, a full copy must be performed before executing this last pass. This is still faster than just disabling the stencil, as a hardware copy is relatively fast and only a few pixels are usually processed by our technique.

Now it's time to dive into the meaty nuts and bolts of our MLAA implementation!

STEP 1 LOOKING FOR EDGES

/// Edge detection is a critical step for the quality of the final image. Each undetected edge will remain aliased in the final image, so we need to detect as many edges as possible. Robustness in this step is also desirable, given that good edge detection enhances temporal stability. However, this is not as easy as it may sound. Optimally, we just want to detect edges that are visible to the human eye (no need to spare time anti-aliasing edges which won't be seen, right?). And not only that, we need clean edges as well, in order to detect their patterns properly.

In this article, we focus on a color-based edge detection, which is the most straightforward option. Depth, normals, or object IDs could also be used, since they are better estimators for geometrical edges; however, they are sometimes tricky and require extra information in the form of maps. Working with color also provides seamless handling of shading aliasing, which may improve quality in some scenarios.

First, we calculate luma values following the ITU-R Recommendation BT. 709:

$$Y' = 0.2126 \times R' + 0.7152 \times G' + 0.0722 \times B'$$

Note that R', G', B' are gamma-corrected values; this is crucial to performing accurate edge detection, so pay attention to your SRGBTexture flags and `DXGI_FORMAT_R8G8B8A8_UNORM_SRGB` texture flags, in DirectX 9 and 10 respectively.

Armed with this info, we calculate and threshold the luma differences between the current pixel and its top and left neighbors in order to obtain a binary value indicating whether there is a sharp edge in between. The result of this pass is stored as a two-channel edges texture (as we saw in Figure 2, center-left). The color of each pixel codes the location of the edges: green pixels have an edge at the top, red pixels have them at their left, and yellow pixels have edges at both boundaries. We actually oversimplified a little bit here: Since we need to create the stencil mask in this first pass and because it needs to tag every pixel that has an edge (in either the top, right, bottom, or left boundaries), we actually calculate and threshold the differences between the pixel and the full 4-neighborhood. Listing 1 shows the shader code of this luma-based edge detection.

In low-end cards, these dot products can introduce an important

LISTING 1 COLOR-BASED EDGE-DETECTION SHADER

```
float4 ColorEdgeDetectionPS(float4 position : SV_POSITION,
                            float2 texcoord : TEXCOORD0) :
SV_TARGET {
float3 weights = float3(0.2126,0.7152, 0.0722);

/**
   * Luma calculation requires gamma-corrected colors, and thus
´colorTex´ should
   * be a non-sRGB texture.
   */
   float L = dot(colorTex.SampleLevel(PointSampler, texcoord,
0).rgb, weights);
   float Lleft = dot(colorTex.SampleLevel(PointSampler, texcoord,
0, -int2(1, 0)).rgb, weights);
   float Ltop  = dot(colorTex.SampleLevel(PointSampler, texcoord,
0, -int2(0, 1)).rgb, weights);
   float Lright = dot(colorTex.SampleLevel(PointSampler, texcoord,
0, int2(1, 0)).rgb, weights);
   float Lbottom  = dot(colorTex.SampleLevel(PointSampler,
texcoord, 0, int2(0, 1)).rgb, weights);

   float4 delta = abs(L.xxxx - float4(Lleft, Ltop, Lright,
Lbottom));
  float4 edges = step(threshold.xxxx, delta);

   if (dot(edges, 1.0) == 0.0)
    discard;

   return edges;
}
```

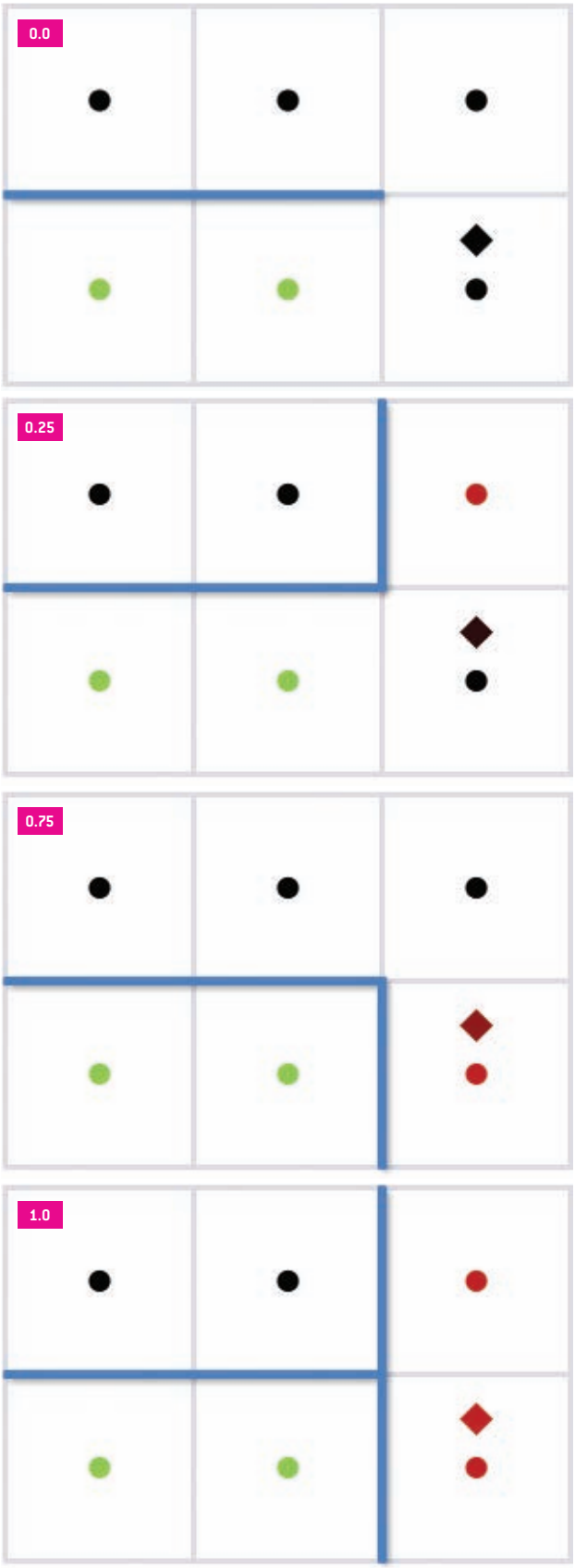


FIGURE 4 Crossing edges and their corresponding value using bilinear filter fetch.

overhead (given the fact that this pixel shader is executed for all the pixels in the framebuffer); thus, it can be wise to calculate lumas in the main render pass, and store them in the alpha channel of the color render target. If a z pre-pass is performed by the engine, this reduces the five dot products required per pixel to only one. Furthermore, in a DirectX 9 implementation, since offsets cannot be directly specified in texture functions, further cycles can be saved by offloading their calculations to the vertex shader.

STEP 2 OBTAINING BLENDING WEIGHTS

/// We're now getting to the most complex step of our crusade against the jaggies, so please keep Figure 1 in mind for what comes next: blending weights computation. To do this, we need to obtain the distances to the ends of the line segment that each edgel belongs to, using the edges textures from the previous pass. Once we know these distances, we will use them to fetch the crossing edges at both ends of the line. These crossing edges indicate the type of pattern we are dealing with. Both the distances to the ends of the line segment and the type of pattern are used to access the pre-calculated area texture used for blending in the final pass.

As you may have already noticed, two adjacent pixels share the same boundary. This allows sharing of calculations between adjacent pixels—you can perform an area calculation on a per-edgel basis. However, even though two adjacent pixels share the same calculation, the resulting a value is different for each of them: only one has a blending weight a, whereas for the opposite one a equals zero (pixels (1,2) and (1,1) in Figure 1, respectively). The one exception to this is the case in which the pixel lies at the middle of a line of odd length (as in pixel (2,1) in Figure 1); in this case, both the actual pixel and its opposite have a non-zero value for a. As a consequence, the output of this pass is a texture that, for each pixel, stores the areas at each side of its corresponding edgels (by the areas at each side, we mean those of the actual pixel and its opposite). This yields two values for north edgels and two values for west edgels in the final blending weights texture, perfectly fitting in the allocated RGBA storage. These weights will be used in the third pass to perform the final blending. Listing 2 shows the source code of this pass, while Figure 2, center-right, shows the resulting blending weights texture.

SEARCHING FOR DISTANCES

/// The search for the distances to the ends of the line is done using an iterative algorithm. In each iteration, it checks whether the end of the line has been reached. To accelerate this search, we leverage the fact that the information stored in the edges texture is binary (as it simply encodes whether an edgel exists), and query at positions between pixels using bilinear filtering for fetching two pixels at a time, thus advancing two pixels per iteration. The result of the query can be:

A. 0.0, WHICH MEANS THAT NEITHER PIXEL CONTAINS AN EDGEL,
B. 1.0, WHICH IMPLIES AN EDGEL EXISTS IN BOTH PIXELS, OR
C. 0.5, WHICH IS RETURNED WHEN JUST ONE OF THE TWO PIXELS CONTAINS AN EDGEL.

Stop the search if the returned value is lower than one (in practice, we use 0.9 due to bilinear filtering precision issues). By using a simple approach like this, we are introducing two sources of inaccuracy. First, we do not stop the search when encountering an edgel perpendicular to the line we are following, but when the line comes to an end instead. Second, when the returned value is 0.5, we cannot distinguish which of the two pixels contains an edgel. While these inaccuracies may introduce errors in some cases, we found them not be noticeable in practice. Moreover, the speed-up resulting from jumping two pixels per iteration is considerable.

Figure 3 shows an example where the color of the dot at the center of each pixel represents its value in the edges texture. The distance search for the left end of the line is performed for the pixel marked with a star. Positions where the edges texture is accessed, fetching pairs of pixels, are

LISTING 2 BLENDING WEIGHTS CALCULATION SHADER

```
float4 BlendingWeightCalculationPS(float4 position : SV_POSITION,
                                   float2 texcoord : TEXCOORD0) :
SV_TARGET {
   float4 weights = 0.0;

   float2 e = edgesTex.SampleLevel(PointSampler, texcoord, 0).rg;

   [branch]
   if (e.g) { // Edge at north

   // Search distances to the left and to the right:
   float2 d = float2(SearchXLeft(texcoord), SearchXRight(texcoord));

   // Now fetch the crossing edges. Instead of sampling between
edgels, we
   // sample at -0.25, to be able to discern what value each edgel
has:
   float4 coords = mad(float4(d.x, -0.25, d.y + 1.0, -0.25),
                       PIXEL_SIZE.xyxy, texcoord.xyxy);
   float e1 = edgesTex.SampleLevel(LinearSampler, coords.xy, 0).r;
    float e2 = edgesTex.SampleLevel(LinearSampler, coords.zw, 0).r;

   // Ok, we know how this pattern looks; now it is time for getting
   // the actual area:
     weights.rg = Area(abs(d), e1, e2);
  }

   [branch]
   if (e.r) { // Edge at west

   // Search distances to the top and to the bottom:
   float2 d = float2(SearchYUp(texcoord), SearchYDown(texcoord));

   // Now fetch the crossing edges (yet again):
   float4 coords = mad(float4(-0.25, d.x, -0.25, d.y + 1.0),
                       PIXEL_SIZE.xyxy, texcoord.xyxy);
   float e1 = edgesTex.SampleLevel(LinearSampler, coords.xy, 0).g;
   float e2 = edgesTex.SampleLevel(LinearSampler, coords.zw, 0).g;

   // Get the area for this direction:
   weights.ba = Area(abs(d), e1, e2);
   }

   return weights;
}
```

marked with rhombuses (their color represents the fetched value).

In order to keep execution times practical, we limit the search to a certain distance. As expected, the greater the maximum distance, the better the quality of the anti-aliasing. However, we have found that for the majority of cases, distance values between 8 and 16 pixels offer a good trade-off between quality and performance. Listing 3 shows one of the distance search functions.

In the particular case of the Xbox 360 implementation, we make use of the `tfetch2D` assembler instruction, which allows us to specify an offset in pixel units with respect to the original texture coordinates of the query. This instruction is limited to offsets of –8 and 7.5, which constrains the maximum distance that can be searched. When searching for distances greater than eight pixels we cannot use the hardware as efficiently, and performance takes a hit.

LISTING 3 DISTANCE SEARCH FUNCTION (LEFT DIRECTION CASE)

```
float SearchXLeft(float2 texcoord) {
   texcoord -= float2(1.5, 0.0) * PIXEL_SIZE;
   float e = 0.0;
   // We offset by 0.5 to sample between edgels, thus fetching two
in a row
   for (int i = 0; i < maxSearchSteps; i++) {
      e = edgesTex.SampleLevel(LinearSampler, texcoord, 0).g;
   // We compare with 0.9 to prevent bilinear access precision
problems
   [flatten] if (e < 0.9) break;
   texcoord -= float2(2.0, 0.0) * PIXEL_SIZE;
   }
   // When we exit the loop without finding the end, we want to
return
   // -2 * maxSearchSteps
   return max(-2.0 * i - 2.0 * e, -2.0 * maxSearchSteps);
}
```

LISTING 4 PRECOMPUTED AREA TEXTURE ACCESS FUNCTION

```
#define MAX_DISTANCE 32

float2 Area(float2 distance, float e1, float e2) {
   // * By dividing by areaSize - 1.0 below we are implicitly
offsetting to
   //   always fall inside a pixel
   // * Rounding prevents bilinear access precision problems
   float areaSize = MAX_DISTANCE * 5.0;
   float2 pixcoord = MAX_DISTANCE * round(4.0 * float2(e1, e2)) +
distance;
   float2 texcoord = pixcoord / (areaSize - 1.0);
   return areaTex.SampleLevel(PointSampler, texcoord, 0).rg;
}
```

FETCHING CROSSING EDGES

/// Once we have the distances to the ends of the line, we use them to obtain the crossing edges. A naïve approach for fetching the crossing edge of an end of a line would imply querying two edgels. A more efficient approach is to use bilinear filtering for fetching both edgels at a time, similar to the distance search. However, in this case, we must be able to distinguish the actual value of each edgel, so we query with an offset of 0.25, allowing us to distinguish which edgel is equal to 1.0 when only one of the edgels is present. Figure 4 shows the crossing edge corresponding to each of the different values returned by the bilinear query. The color of the dot at the center of each pixel represents the value of that pixel in the edges texture. The rhombuses indicate the sampling position, while their color represents the value returned by the bilinear access.

THE PRECOMPUTED AREA TEXTURE

/// With distances and crossing edges at hand, we now have all the ingredients for calculating the area which corresponds to the current pixel. Since this is an expensive operation, we pre-compute it in a 4D table, stored as a conventional 2D texture (see Figure 5, right). This texture is divided into subtextures of size 9x9, each of them corresponding to a pattern type coded by the fetched crossing edges e1 and e2 at each end of the line. Figure 5 (left) shows you the 16 different patterns we handle (each one with direct correspondence to a subtexture); the orange lines indicate the perceptual re-vectorization to be performed in each case.

LISTING 5 4-NEIGHBORHOOD BLENDING SHADER

```
float4 NeighborhoodBlendingPS(float4 position : SV_POSITION,
                              float2 texcoord : TEXCOORD0) :
SV_TARGET {
   // Fetch the blending weights for current pixel:
   float4 topLeft = blendTex.SampleLevel(PointSampler, texcoord,
0);
   float bottom = blendTex.SampleLevel(PointSampler, texcoord, 0,
int2(0, 1)).g;
   float right = blendTex.SampleLevel(PointSampler, texcoord, 0,
int2(1, 0)).a;
   float4 a = float4(topLeft.r, bottom, topLeft.b, right);

   // Up to 4 lines can be crossing a pixel (one in each edge).
Thus, we perform
   // a weighted average, where the weight of each line is 'a'
cubed, which
   // favors blending and works well in practice.
   float4 w = a * a * a;

   // Is there any blending weight with a value greater than 0.0?
   float sum = dot(w, 1.0);
   [branch]
   if (sum > 0.0) {
      float4 o = a * PIXEL_SIZE.yyxx;
      float4 color = 0.0;

   // Add the contributions of the 4 possible lines that can cross
this
   // pixel:
   color = mad(colorTex.SampleLevel(LinearSampler, texcoord +
float2( 0.0, -o.r), 0), w.r, color);
   color = mad(colorTex.SampleLevel(LinearSampler, texcoord +
float2( 0.0,  o.g), 0), w.g, color);
   color = mad(colorTex.SampleLevel(LinearSampler, texcoord +
float2(-o.b,  0.0), 0), w.b, color);
   color = mad(colorTex.SampleLevel(LinearSampler, texcoord +
float2( o.a,  0.0), 0), w.a, color);

     // Normalize the resulting color and we are finished!
     return color / sum;
   } else {
     return colorTex.SampleLevel(LinearSampler, texcoord, 0);
   }
}
```

Inside each of these subtextures, the (u, v) coordinates correspond to distances to the ends of the line, eight being the maximum. Resolution can be increased if a higher maximum distance is required.

Maybe at this point you are wondering why, according to Figure 5, patterns 5, 10, and 15 do not perform any anti-aliasing. That's because we found objects like rails (patterns 5 and 10) and contiguous quads (pattern 15) to be better preserved this way. Compound patterns like 7, 11, 13, and 14 are also special cases. In these cases, in order to obtain the best results, you must choose a main pattern in order to reduce artifacts. We chose Z-shapes, but U-shapes could also be a valid choice.

Following the same reasoning, in which we store area values for two adjacent pixels in the same pixel of the final blending weights texture, the precomputed area texture needs to be built on a per-edgel basis. Thus, each pixel of the texture stores two *a* values, both for a pixel and its opposite (again, *a* will be zero for one of them in all cases except those of pixels at the center of lines of odd length).

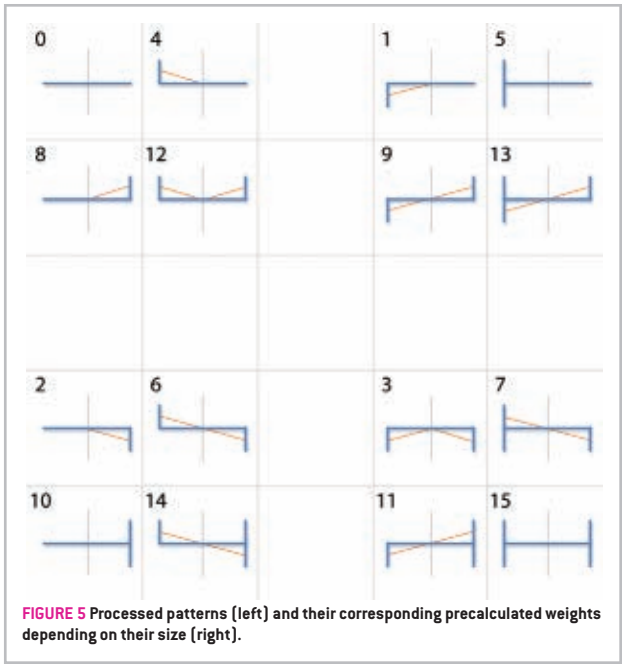


FIGURE 5 Processed patterns (left) and their corresponding precalculated weights depending on their size (right).

Listing 4 gives details on how the precomputed area texture is accessed. To query the texture, we first convert the bilinear filtered values e1 and e2 to an integer value in the range [0..4] Value 2 (which would correspond to value 0.5 for e1 or e2) cannot occur in practice, which is why the corresponding row and column in the texture are empty. Maintaining those empty spaces in the texture allows for a simpler and faster indexing. The round instruction is used to avoid possible precision problems caused by the bilinear filtering.

STEP 3 BLENDING WITH THE NEIGHBORHOOD

/// We already have the edges where anti-aliasing needs to be performed, plus the blending weights. In this last pass, we will obtain the final color of each pixel by blending the current color with its four neighbors according to the area values stored in the weights texture. To do this, we have to access three positions in the blending weights texture:

A. THE CURRENT PIXEL, WHICH GIVES US THE NORTH AND WEST BLENDING WEIGHTS;
B. THE PIXEL AT THE SOUTH; AND
C. THE PIXEL AT THE EAST.

This yields the blending weights with the complete 4-neighborhood. Once more, to exploit hardware capabilities, we use four bilinear filtered accesses to blend the current pixel with each of its four neighbors. Finally, given that one pixel can belong to four different lines, we perform a weighted average between the contributing lines. The cubed blending weight (*a* cubed) of each possible line is used as the weight of this average, which favors blending and works well in practice. Listing 5 shows the source code of this pass, while Figure 2, right, shows the resulting anti-aliased image.

It's important to note that if you want all this blending calculated properly, you have to ensure you are working in linear space. Using bilinear filtering and `DXGI_FORMAT_R8G8B8A8_UNORM_SRGB` textures for calculating this step in DirectX 10 enforces linear blending. In DirectX 10 hardware running DirectX 9 code, this bilinear filtering blending will be performed, again, in linear space. However, DirectX 9 hardware running in DirectX 9 will perform the blending in gamma space. Thus, in this case, manual blending using lerps is advised.

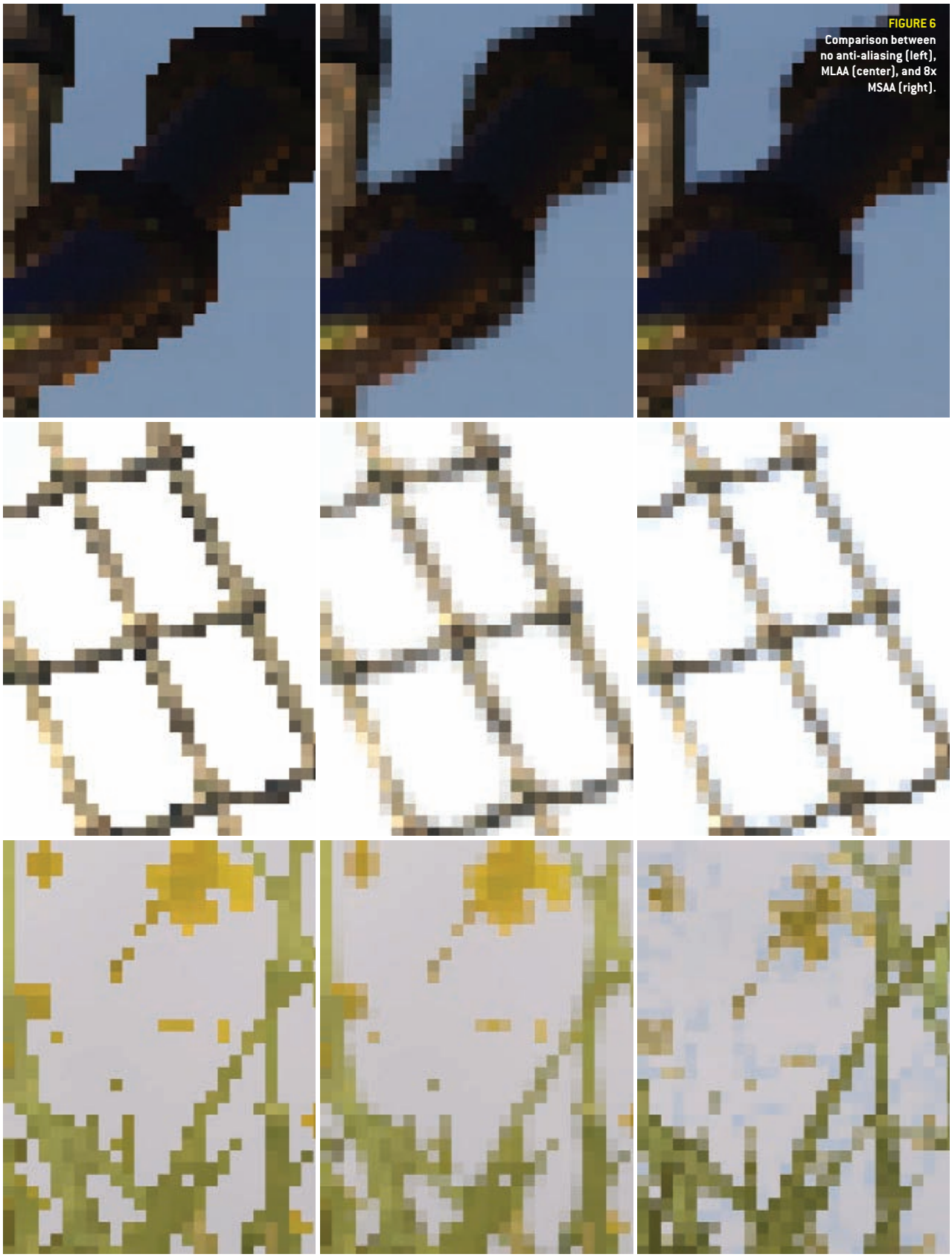


FIGURE 6
Comparison between no anti-aliasing (left), MLAA (center), and 8x MSAA (right).

Figure 8. More examples of our technique applied over images from Unigine's Heaven demo. Insets show no anti-aliasing (left) and MLAA (right).

FIGURE 7 Examples of our technique applied over images from Unigine's Heaven Demo. Insets show no anti-aliasing (left) and MLAA (right).

WRAP-UP

/// So, what are the benefits of this MLAA implementation for your graphics engine? Well, you can run this technique as a regular pixel shader in DirectX 9 and above (including the Xbox 360). In comparison to proprietary techniques, it has the additional advantage that it can be triggered at any desired step along the pipeline (usually somewhere after tone mapping and before GUI rendering), providing you with great flexibility. Furthermore, now that you know what is under the hood, you can tweak it to fit your game like a glove.

When edge detection fails, our technique can be as bad as 1x (in fact, in these cases, it is 1x). In the presence of sub-pixel features MSAA can be superior (although proper care in art direction can solve some cases). However, our implementation is comparable, in general, to 16x MSAA, while only requiring a memory consumption of 1.5x the size of the backbuffer on DirectX 10, and of 2x on DirectX 9-based implementations. Figure 6 shows a comparison between the algorithm, 8x MSAA, and no anti-aliasing at all on images from the Unigine Heaven benchmark. More results of our technique are shown in Figures 7 and 8. Take a look at our project page for additional information about the technique, including an exhaustive performance analysis, an image gallery, a movie, and implementations for both DirectX 9 and 10: www.iryoku.com/mlaa.

FIGURE 8 More examples of our technique applied over images from Unigine's Heaven demo. Insets show no anti-aliasing (left) and MLAA (right).

Typical execution times are 2.60ms on the Xbox 360 and 0.44 ms on an NVIDIA GeForce 9800 GTX+ for a resolution of 720p (tested in DirectX 10 and XNA, respectively). According to our measurements, 8x MSAA takes an average of 5 ms per image on the same GPU at the same resolution—that is, our algorithm is 11.80x faster.

The method presented has a minimal impact on existing rendering pipelines, and is entirely implemented as an image post-process. Resulting images can be on par with 16x MSAA in terms of quality, while requiring a fraction of their time and memory consumption. Furthermore, it can anti-alias transparent textures such as the ones used in alpha testing for rendering vegetation, whereas MSAA can only smooth vegetation when using alpha to coverage. We believe that the quality of the images produced by our algorithm, its speed, efficiency, and pluggability, make it an attractive choice for rendering high-quality images in today's game architectures, including platforms where benefiting from anti-aliasing together with outstanding techniques like deferred shading was previously difficult to achieve.

Some of the information presented in this article has been adapted by the original authors from the GPU Pro 2 chapter "Practical Morphological Anti-Aliasing."

AUTHORS: JORGE JIMENEZ *Universidad de Zaragoza,* JOSE I. ECHEVARRIA *Universidad de Zaragoza,* BELEN MASIA *Universidad de Zaragoza,* FERNANDO NAVARRO *Lionhead Studios*, NATALYA TATARCHUK *Bungie*, DIEGO GUTIERREZ *Universidad de Zaragoza*

19-21 July 2011
develop in brighton

Early Bird
Book before 15 June Save 20%

BROADEN YOUR SKILLS, YOUR NETWORK, YOUR HORIZONS

Join 1,300 international game developers and choose from over 80 top-notch sessions across 3 inspiring days.

19 July 2011
evolve in brighton

Tuesday 19 July

The forward-looking Evolve conference focuses on all that is new in game development - new platforms, new technologies and new markets.

Evolve will help developers tackle the issues arising from emerging platforms and digital marketplaces, connected gaming, user-generated content, and the crossover between games and Internet services.

20-21 July 2011
develop in brighton

Develop Expo runs alongside the conference and is free to attend!

Wednesday 20 & Thursday 21 July

Main conference tackling the issues, tools, tricks and techniques of game development and offering practical advice and solutions.

ART · BUSINESS · CODING · DESIGN · PRODUCTION · THE DEN · AUDIO

21 July 2011
indie dev day

NEW!

Thursday 21 July

The new **Indie Dev Day** designed specifically for independent developers offers tons of sessions, a chance to meet publishers and mix with other indies. Plus, **Indie Showcase:** a dedicated area within the Develop Expo (20-21 July) featuring some of the most exciting and new indie projects currently under development.

WHO'S SPEAKING?

EVOLVE — Keynote ☆
Lessons Learned Building Moshi Monsters to 50m Users
Michael Acton Smith, CEO, Mind Candy

EVOLVE — Keynote ☆
Moving the Goalposts – Bringing Social Experiences to All Game Platforms
Andrew Wilson, SVP of Worldwide Development for EA SPORTS

ART — Keynote ☆
Showtime: Learning to be Creative on Demand
Iain McCaig, Artist and Conceptual Designer

CODING
From Phones to High End PCs: How to Scale Your Technology
Aras Pranckevicius, Unity Technologies

DEVELOP — Keynote ☆
The Stereoscopic 3D Experience - The First Year of 3D on PlayStation 3
Mick Hocking, Senior Director, SCEE

BUSINESS — Keynote ☆
Managing the Mystery
Laura Fryer, VP and General Manager, WB Games Seattle

DESIGN
Innovation - A State of Mind
Nick Baynes, Black Rock Studio

PRODUCTION
Being Agile in a Waterfall World
Brynley Gibson, Zoe Mode

Plus many more... see website for more information.

www.developconference.com

International Media Sponsor: develop
Media Sponsor: gamesindustry.biz
Media Sponsor: EDGE
Media Sponsor: GAMASUTRA
Media Sponsor: casualgaming.biz
Media Sponsor: IMAGINE

Media Sponsor: AUDIO MEDIA
Media Sponsor: gd Game Developer Magazine
Member Discount: GAMEREPUBLIC
Member Discount: ukie
Member Discount: TIGA
Member Discount: igda
Organised by: tandem events

SCOTT HARTSMAN

postmortem

/// I'm writing this in the time between RIFT's first and second major live updates, six weeks after our official launch. The team is cranking through feature and bug lists, making last-minute adjustments to fine-tune both for flow and performance. All that comes just after our first major update to the game—RIFT 1.1: RIVER OF SOULS—where 30 days after launch, we released new features, new content, and plenty of adjustments from both player feedback and metrics along the way.

As if that weren't enough excitement, in parallel with the above, we're hard at work on RIFT 1.2 and additional releases further into the future.

For some background, RIFT is an MMORPG set in a world that is at a nexus of six different planes of reality, and as Ascended heroes, players must fight these invasions back. A lot of effort goes into finding the right balance between "How massive can we go?" versus "How many people in the same place is too many to be fun?"

The action here in Trion's Redwood Shores studio has never been more intense than it is right now. In that light, it seems a little strange to be writing something called a "Postmortem." Following our recent launch, I am happy to report that there is very little mortem-ing occurring around here at all (aside from the 4,831,100,815 untimely creature and player deaths that have occurred in the world of RIFT as of the moment I'm typing this sentence).

One thing you learn very rapidly once you ship an MMO is that, the day you ship it, the work only ramps up from there. And when I say "ramps up," I of course mean, "Shoots up to the sky, not entirely unlike the sudden appearance of a cliff wall covered in poison-tipped metal spikes."

Being able to iterate with your customers in real time is a truly addictive feeling for developers who love games, right alongside the mutual sense of urgency it creates. (On the Internet, remember, a day is "a long time," a week is "intolerable," and a month is "forever.") While that kind of time compression can be nerve-wracking, it can also make developers feel like we're all in this same crazy world together with our players, going through possibly the greatest shared experience you can have in game development. I think that's a huge part of what keeps a lot of us doing this year after year.

Instead of a traditional postmortem, think of this as a momentary pause for reflection on the journey that took us from Trion and RIFT's beginnings through the end of the period we'll call "Pre-Live."

WHAT WENT RIGHT

1 /// OFF-THE-SHELF SOFTWARE. Almost every postmortem I've read or been a part of has talked about tools. Either they were an afterthought, there wasn't enough investment, or the tool-users' time wasn't valued as highly as that of the teams' more technical members—and the less technical were expected to make do.

Our team went into this project assuming that the sooner we got usable tools, the better off we'd be in the long run. Beyond that, the

less time the team had to spend on library elements that are fairly common, the more time we'd have to focus on the core uniqueness that was important to get right. One thing that let us maintain that focus was wise selection of off the-shelf software, which we iterated up (replacing pieces as needed).

Gamebryo (though it has almost all been replaced by now with custom tools and rendering code) was key to kicking off the project. It was the team's window into the world, and gave the worldbuilding team the ability to start iterating. Scaleform was a great tool for creating a UI that was both functional and attractive, following an earlier iteration that didn't quite work as well.

The list goes on: Havok helped us out a lot with physics, Wwise gave our audio team great ways to prototype and the ability to mod the audio experience in real time, rsyslog was used for logging, NavPower for pathfinding, and so on.

With this foundation, we could go back and address the places where existing tools just didn't scale to production, or where making them fit our game was not a reasonable expectation.

Our tools engineers were able to focus the design and CS tools we needed to launch with in a way where they could extend however our game needed. There's a lot of value in being able to focus on the things that are really specific to our game and platform.

2 /// HIRED A SMALL, TALENTED TEAM AT THE START AND BUILT SLOWLY AND CAREFULLY. Not counting the fact that an entire company was being built simultaneously, the development team knew that the game was going to be a massive undertaking (pun intended), and that by the time we were at a stage to ramp toward alpha and beta, there would be 100 or more people needed in order to make sure everything was going to get done.

The early core team did a lot of work on policies and best practices. When much of the later hiring occurred, there was already significant institutional knowledge, or "here's how we do that," to be passed along. Ongoing iterations could also be passed to everyone using a common base language, which is something you only realize is hard to create when you're having to do it from scratch.

Additionally, although recruitment was hard for a new startup with low visibility, we used word of mouth and personal networks to get great dev team members early on, which started a positive hiring cycle. That paid off tremendously later. The fact that so many great people brought all their experience to the team meant that we were a huge step ahead of many online game startups in that things like stability and data integrity were already being developed as though we were a live service. That's really the big thing that enabled our incredibly rapid iteration cycle over the final year of development, which was truly make or break for us as all the features were coming together in their final form.

3 /// ELEPHANTS DON'T BELONG IN ROOMS. This team is more aggressive with playing and commenting on their own work, and each other's, than any I've ever seen. Sometimes feathers are ruffled, but in the end, the culture is really one where we understand that getting to quality is the best thing for everyone, and that blunt opinions are generally appreciated.

As a result, we had to grow into a culture where we're unafraid to ditch something if it isn't working, otherwise all those insights would be for naught. We weren't necessarily happy to do it, but we redid pieces of content many, many times. Even if they stuck for a while (or a year), we would start over if we realized they were not working. This was a critical part of getting our content to where it is now, because at several points, we had to make very painful decisions with zones, characters, animations, and practically every other aspect of the game.

In line with both having a sane hiring ramp of solid devs, and making a point of having good tools, we have been able to be nimble when we change gameplay, add gameplay, or alter the order in which features/content items are delivered. This has also allowed us to, more often than not, create and release content very quickly with confidence that the quality and stability will be high. This was always a core value from day one.

4 /// SCOPE: CONTROL IT, OR BE CONTROLLED. Fantasy MMO RPGs are interesting to make in a lot of ways. Take, for example, feature sets and what those features look like. If your game is new, you're going to get compared to whoever is number one at the time. It's been happening for over a decade.

If you try to innovate in every single area just to ensure you are different, you're never going to ship on time and at a quality level that people will accept. You'll hit exactly one or the other, which unfortunately means an untimely death for your game.

Further, there's a price to pay for innovation in a fantasy MMORPG. If you don't have enough similar elements to others in the space presented in a way that is easily understood by experienced players, your MMO will be viewed as "broken" even if it all works. (What this really means is, "too inaccessible to attract and retain a sufficient audience.") Unfortunately, that too means you die. Is that the way the world should work? I think a lot of us wish it were a little more forgiving, but that's what the last few years of MMO releases have reinforced to developers.

With that in mind, there is a balance to hit between "new" and "what you're used to." We landed where we landed intentionally. Our goal was to iterate up while making sure to focus our innovation time on things that made us unique. One of those unique features was our rift system, where tears from the planes could open any time, and players would have to stop what they're doing and band together

GAME DATA

PUBLISHER
Trion Worlds
DEVELOPER
Trion Worlds
NUMBER OF DEVELOPERS
110 at launch
LENGTH OF DEVELOPMENT
4.5 years
RELEASE DATE
March 3, 2011 USA
March 4, 2011 Europe
BUDGET
$50 million
PLATFORM PC

and fight back invaders. (In an MMO, that is an incredibly important differentiator: what is fun to do with a full server is frequently an entirely different thing from what is fun while you're the only person on an internal world.)

That leads us up to about a year and a half before launch. We had a pretty good idea of how much we were going to be able to get done in the remaining time. We locked down what would become the final feature set, and made sure to leave time at the end for the inevitable learning of new systems that would only occur to us once we got the game into alpha testing and saw how it played in front of a real audience.

We made some painful cuts at that point, instead of waiting for features to be partially developed only to be deferred until later.

5 /// SAME GREEK ALPHABET, BRAND NEW FLAVOR. Our method of testing RIFT brought some interesting new twists to the typical Greek Alphabet Progression™ method of MMO tests, where Alpha is Alpha, Beta is Beta, and Gamma doesn't exist. In our case, they overlapped in parallel.

Alpha. For us, this meant semi-private testing on one external set of servers that ran 24/7 beginning about a year before launch. This was where we practiced running a 24/7 service. This was where we worked on stability and integrity and rapid iteration in front of (immensely patient) outside users. Because we had a dedicated place to focus on this, we were able to make it through Alpha with no unplanned wipes, and only one database restore.

Beta. This was public testing on an entirely different set of external servers. These were time-limited events. What these let us do was check out how everything played out "for real." Doing our beta this had two distinct advantages from the development side.

First, we went through a full company-wide "launch" each time we ran a beta. I cannot stress how important this was to us. By the time we got to the game's actual launch—a thing that most MMOs only get to do once—we had practiced so many times that it was old hat. When it came time for the big show, there were no surprises. We had just done it eight times in three months.

Second, it let us go dark for a week or two in between most events, where we could be sheltered from the constant frenzy of running the real service at scale. You know how you might get a random brilliant idea in the shower when you're not thinking about anything in particular? It's like that. Once you're running 24/7 at scale, getting those quiet times is more and more difficult. It was during many of those quiet times that we had some great revelations for improvements and could also follow up on focusing on them to the exclusion of everything else.

WHAT WENT WRONG

1 /// SAME CODE, TWO TEAMS, TWO MISSIONS. From its inception, our company was built around the idea that it's possible to start with absolutely nothing, then build an MMO framework and publishing platform, as well as multiple products, all at the same time. It's an incredibly ambitious (some would say insane) goal.

The lines of "is this a part of the product or is this a part of the platform?" had to be redrawn multiple times in order to get us to success, as the day-to-day realities of progress set in. Everything that is good for an MMORPG is not necessarily what is good for an MMORTS, and is also not necessarily good for an MMO action game. It sounds like common sense when you phrase it that way, but it's fraught with nuance. Where do you draw the line? Is it the comm. library? Is it the database storage layer? The graphics engine? The UI library? What about patching? Authentication? All of it? None of it? There are dozens of technology pieces. You can't make them all unique, and they can't all be common.

One of the earliest iterations had developers in one studio working on technologies that were to be used as both time-savers and known-good solutions by all of our games, while developers in the studios such as ours were trying to get fun games off of the ground by providing our own teams with the servers, clients, tools, and gameplay systems that would let us build toward greatness from the customer point of view, which in the end, is the only point of view that matters. We have three studios in Austin, San Diego and Redwood Shores. Austin is primarily responsible for the tech and game platform development for our games. Redwood City is our main company headquarters, and where the RIFT team is. San Diego is working on SYFY and END OF NATIONS.

Suffice it to say, it was a frustrating experience for smart developers on both sides of the issue. Each studio is working on a totally different game, developing its own technology and best-practices that worked for that specific title. As time passed we learned to communicate more effectively and honed in on the right amount of technology that would be smart to consider "shared," and which parts should exist per-game, to maximize our development effectiveness.

In the end, our guide became the general overriding sentiment that "If the product isn't awesome, nothing else matters." As we focused the entire company on that saying, more pieces began to fall into place, and more answers became self-evident.

2 /// IDENTITY: ARE WE A PLATFORM, A PUBLISHER, A DEVELOPER, OR ALL THREE? Beyond the technology, what you're trying to become when you grow up shapes a lot of how you logically build out the company. Which departments are where? Who's responsible for what?

We originally focused much more around the concept of being a platform. With a second game already in development (END OF NATIONS, being developed by Petroglyph Games), that meant we rapidly had to become ready to build up a milestone-evaluation type of quality assurance to make sure games were getting done on time. That kind of QA is more about periodic certification.

The internal developers, such as the RIFT team, needed another kind of QA—one that was built purely around validation. It doesn't help a RIFT developer to hear about bugs weeks later at the end of a milestone when they've already long since moved on. In order to stand any chance of obtaining quality, developers should check something in one day and have QA feedback the next, while their heads are in the same place.

We realized that we needed both kinds of QA in order to fulfill all the company's goals (periodic certification as well as daily validation). Acting on that was a key part of becoming a rapid iteration shop. It was more expensive than we had bargained for, but it's more than paid for itself since.

3 /// CREATING A NEW IP WAS HARDER THAN WE THOUGHT. While trying to build a company up from nothing, and building out multiple studios and products in parallel, there was also this little problem of having to create a new IP.

I'm not sure if it works the same way in your studios, but at least at ours—and this may shock some of you—not everyone has the same taste in fictional direction. As such, it took far longer than anyone expected to really get everyone on the same page and excited about the world we were making. What would it look like? What would it sound like? Who are the notable characters? What are their motivations? How can we express those? Ad infinitum.

When so much is being created and iterated every day, trying to keep a rapidly growing team of people, much less the general public, up to date on the current state of everything is a hell of a challenge, and one that was definitely underestimated. Creating the IP and surfacing the story wasn't really something that we began getting good at until fairly late in the process, though still early enough to have a successful launch. Most MMORPGS have long PR and marketing cycles—years in most cases. With RIFT, we announced officially in April 2010 and launched in March 2011, which was not a ton of time for the PR/Marketing machine to get all the information about the game out to fans and build a groundswell. But I'm happy to say that we did a pretty good job with that on a limited time budget.

4 /// LOCALIZATION: WE CAN DO THAT ALL AT THE END, RIGHT? Localization was a big challenge for us. We reached a really good level of quality, but it had a far more significant cost than we assumed it would, both in terms of money and emergency engineering time forging the pipeline.

Our game launched with over one million words, plus select voiced characters and movies. We knew that we had to launch it at quality in each of the launch languages (English, French, and German). There was also a lot of iteration, right up through launch. We didn't treat it as an afterthought as thinking wasn't the issue—doing was the issue. We should have begun forging the localization pipeline earlier than we did. Only if we had ramped up our localization efforts earlier, there would not have been such a glut of translation work at the end of the project before launch.

Lesson learned: don't be afraid to start early with small samples of unfinished text—out of the game, out to translation, back into the game. It's always easy to say (as we did): "All this text is going to change (or be edited, or be rewritten) anyway. It's too soon!" Any time that you can spend at the outset to ensure a functional pipeline is in place will help negate a potentially huge cost explosion at the end.

Beyond technology, finding people with relevant experience to run the operation and produce the languages was a far greater challenge than we had imagined. We did find them, though, and it's a huge testament to their talent that we still managed to pull it all off in time, given that we started as late as we did. We already had a solid localization department, but we really fleshed it out about a year before launch.

5 /// DYNAMIC CONTENT—PRESENTATION IS EVERYTHING. A large goal of the game was to introduce the concept of a world where there is something interesting going on, a world under active invasion, which is entirely outside the boundaries of what years (for some, over a decade) of MMO experience has trained people to think of as "MMO content."

The architecture was set up to handle real-time swapping of assets and gameplay data, as well as handle the inevitable flash mobs that would gather during in-game events. Once the technical challenges were solved, problems of messaging and visibility appeared. Early in the project, players had a hard time distinguishing what was static content and what was dynamic. Any difference in challenge or new/adaptive behavior was viewed as a bug, even when it was explained in real time.

The answer, and our eventual name, was in our world's lore all along! RIFTS explained visually and fictionally why some content wasn't always there, and they were immediately recognizable as something that players would know to treat differently than the previous decade of fantasy MMO gameplay has trained them to behave around "all game content."

The hallmark of our Dynamic Content system is our rifts, which are events that tears the fabric of reality that spawn planar invaders. These can happen anytime and anywhere in the world, and players must band together to seal the tears.

Later in the development cycle, once we re-introduced the dynamic content as elemental rifts, invasions, colossi, and world takeovers with their own very clear appearances, we crystallized the way we could highlight which content was dynamic in an instantly recognizable way.

By providing this very obvious distinction, players who had previously complained about bugs or hadn't understood content that was unexpected could now understand why it was there, and thus happily throw themselves against a giant colossus with smiles on their faces.

FINAL THOUGHT

/// As I mentioned at the start, this is less of a postmortem and more of a wrap-up of everything that occurred on the way to going live, for real. For us, our journey is just beginning. We couldn't be any happier to be here for the ride.

SCOTT HARTSMAN *is the general manager of Redwood Studio and CCO of Trion Worlds.*

GDC Europe

Game Developers Conference Europe™ 2011

August 15th - 17th, 2011

Congress-Centrum Ost Koelnmesse | Cologne, Germany

www.GDCEurope.com

Summits
(Monday - Wednesday)

- Independent Games
- Smartphone & Tablet Games
- Social Games
- Community Management

Main Conference Sessions
(Wednesday - Friday)

- Business & Marketing
- Game Design
- Production
- Programming
- Visual Arts

Expo Floor
(Monday - Tuesday)

Explore the latest in game innovations and talk with industry experts.

GDC Europe VIP Lounge
(Monday - Wednesday)

Broaden your networks and connect with fellow game developers and professionals from the European games industry.

Game Developers Conference Europe™ 2011 (GDC Europe) is the essential platform for learning, networking, and opportunity for the creators of computer, console, handheld, smartphone, social, and online games in Europe.

Join us at the **Game Developers Conference Europe™ 2011**, August 15-17, for three days of over 100 sessions, as we cover a comprehensive selection of European game development topics taught by leading domestic and international industry experts.

Register before July 20 and save up to €100!

Supported by

European Games Developer Federation · BIU · gamescom

For more information please visit
www.GDCEurope.com

UBM

MULTI-PLATFORM GROWTH

How to cost-effectively reach high-growth gaming markets using enablement technologies: a look at TransGaming's multi-platform approach

Developing games in a world with growing competition and an increasing focus on self-publishing is a daunting task. How do you ensure that your game sells its maximum number of units, or at least enough to fund your next title? One particularly effective strategy is to ensure broad distribution by developing your game for multiple platforms from the outset. Once the realm of large publishers, enablement technologies now allow smaller studios and independent developers to achieve additional revenue streams by taking advantage of emerging platform trends in order to extend your reach, while limiting your risk.

TransGaming Inc. (transgaming.com), an internationally recognized leader in cross-platform enablement software, has recently applied their own tools in creating an original game. "Garage Inc." is a time-management game set in 1920's Chicago, developed by TransGaming's Studio Division in partnership with Breakthrough Entertainment, based on an original concept from Global Fun! and made possible with the support of the Ontario Media Development Corporation. Players take on the role of Angelo Marito as he tries to balance running his own automobile garage: fixing cars, managing employees and avoiding the local mob, all in a day's work. From the outset, the game was targeted for multiple platforms including iOS, Windows, Mac and GameTree TV (Smart TV).

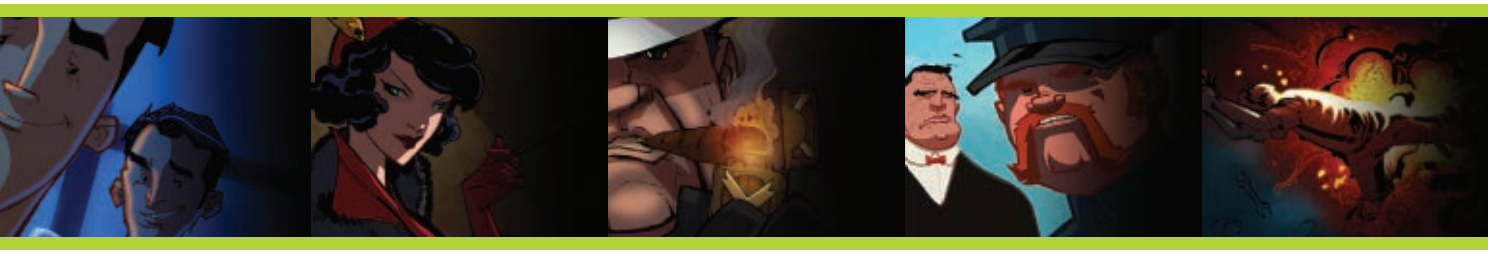
In this article, the development process will be explored in depth to highlight the challenges for developers and introduce procedures for cross-platform game enablement into a project. Beginning with the initial planning, moving into the code base, and then graphic considerations, Garage Inc will be used as a case study to illustrate the principles being discussed. The TransGaming technologies that were incorporated include Cider for Mac enablement, and the GameTree TV SDK for GameTree TV enablement.

Initial Planning and Comprehension

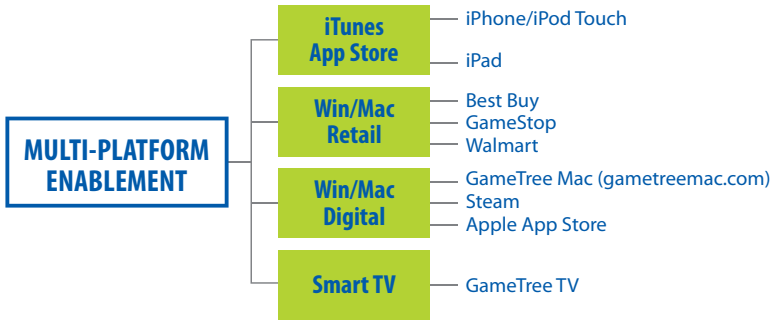
The first aspect to consider is the limitations of the platforms that you choose to target. Because the platforms sometimes vary significantly, there will be adjustments required for each. Consider the following questions: will the game be touch-screen friendly? Is it small enough to run on both a PC and a phone, or will significant memory and cutting edge graphics processing be required to support it? Can the game be packaged and distributed on its own or will the user need to connect to the developer's server? Next, consider all of the facets that go into the gameplay design, identifying features which cannot be supported on all platforms. The remaining features will constitute the core feature set that can be implemented

on all platforms. The platform-specific features will need to be evaluated to determine their additional cost and the corresponding benefit that they provide. When you are done, you will have established the scope of development for your multi-platform title.

Garage Inc. was designed specifically to be a multi-platform game. Since one of the primary platforms for the



game would be iOS which includes devices like the iPhone and iPad, gameplay mechanics that use multiple buttons and complex input methods were ruled out. The iOS devices have much smaller screens and significantly less processing power than most desktop computers. As such, the iOS devices provided a good point of reference for the base set of features that the game could support across all platforms, while GameTree TV with its emphasis on the much larger television screen provided a good target for



the creative team when determining the resolution at which art assets needed to be prepared.

This stage in the process is perhaps the most daunting and time-intensive. However, with proper research and understanding of the available tools, the following steps are made vastly less difficult.

Common Code and the Game Engine

A major advantage of using enablement technologies such as Cider and the GameTree TV SDK from TransGaming, is the ability to easily support multiple platforms using existing development tools and technologies. The Mac OS X and GameTree TV versions of Garage Inc. were developed on Windows using Visual Studio and the existing tool chain and build scripts used for the Windows version of the game. The ultimate differences between the various versions of the game are all minor, and mostly concern issues relating to UI behavior on the different platforms. For example, messages on iOS refer to "tapping" the screen while other platforms refer to "clicking" buttons. This use of a common code base significantly reduces the amount of time required to bring a game to market on multiple platforms.

Codebases are typically forked near milestones in game development. This provides an isolated base from which to focus upon stability rather than feature development and an opportunity to address platform-specific issues with minimal disruption to other development efforts. However, this is not always necessary. In the case of Garage Inc., release builds for all platforms were made from the same branch. By keeping an eye on the various platforms during development, the developers were able to avoid fragmentation amongst the various versions. As a result, it was not necessary to invest a significant amount of time in porting fixes between different versions of the same game and the multiple branches that would typically be made never came into existence.

Another benefit of concurrent development on multiple platforms was the earlier detection of bugs in the game. Many of the issues discovered on Mac OS X and GameTree TV in particular turned out to be issues that could potentially affect the other platforms but which were not immediately visible on the Windows or iOS build.

In the development of a multi-platform game, the game engine is crucial in limiting the amount of work required to bring a game to each platform. Ensuring that gameplay details are separate from system-specific details makes it easier to deal with the idiosyncrasies of each platform, while minimizing the changes that are made to how the game plays. The features that your game design requires and the set of platforms that you wish to support will be the major considerations in choosing the engine that you will use. There are, however, other things to keep in mind. You need to ensure that the engine you choose will be compatible with your team's workflow. Commercial engines may provide a more robust and mature set of capabilities as well as readily available support, but also typically have a greater monetary cost and licensing restrictions.

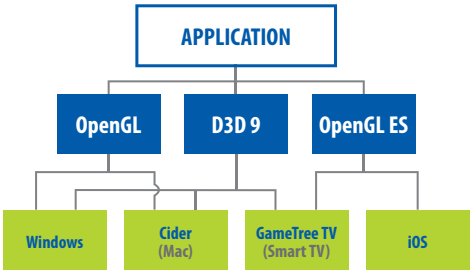
Garage Inc. was built on a proprietary engine that was developed in-house. This engine provides features that are particularly useful for cross-platform development, including automatic configuration of screen layouts for different screen sizes and aspect ratios, as well as automatic selection of different assets based upon the features that the platform supports. For example, different compressed texture packs are selected at run-time based upon the hardware support that is available. While a proprietary engine is not a necessity for multi-platform release or for using TransGaming's enablement technologies, it was deemed the best route in this particular case. Most of the features were implemented to help manage the design issues related to cross-platform development.

In the first month of development, early prototypes of Garage Inc. were running on both iOS and Windows. As development progressed, developers were working simultaneously on both platforms, not just to address platform-specific issues but to implement general gameplay mechanics as well.

Graphics Quality Across Platforms

One of the most crucial facets of the process of developing a multi platform game is determining graphics specifications. The first step is to make an inventory of the available technologies on the platforms you will be targeting. On Windows, the primary graphics API for games is DirectX. Most casual games use various revisions of the DirectX APIs. Higher-end games will typically use Direct3D 9 or newer APIs, while many casual games will still use DirectDraw or Direct3D 8. Mac OS X provides support for OpenGL, while most mobile platforms and set-top boxes provide support for either OpenGL ES 1.1 or OpenGL ES 2.0. Some of these platforms with provide support for both versions of OpenGL ES. This means that there is a potential for a lot of overlap in implementing graphics on these platforms.

TransGaming's Cider and GameTree TV SDKs provide implementations of numerous graphics APIs. The Cider SDK for Mac OS X provides implementations of Direct3D 9.0c including Shader Model 3 and OpenGL 2.1 including numerous vendor extensions. The GameTree TV SDK provides implementations of Direct3D 9.0, OpenGL ES



1.1 and OpenGL ES 2.0. The goal is to provide developers with the opportunity to choose the best tool for the task. If they wish to maximize performance or take advantage of platform-specific features, they can use a graphics API that is native to the platform that they are targeting. However, many games will run very well on other platforms using TransGaming's implementation of Direct3D, allowing developers to minimize the number of different graphics APIs that they need to use.

Even with TransGaming's portability technology bearing a large share of the development burden for multiple platforms, there is room to improve performance on specific platforms, both in terms of speed and resource usage. Various platforms provide different techniques for optimizing a game. For example, DirectX provides support for DXT texture compression, while iOS and GameTree TV provide support for PVRTC texture compression. TransGaming's technologies provide support for these texture compressions through the various graphics APIs. For example, an extension is provided in the Direct3D implementation that allows the use of PVRTC textures

on hardware that supports it. This approach provides the convenience of a known programming interface while still allowing for platform-specific optimization.

Considering that Garage Inc. was initially designed for mobile devices and desktop platforms, there are graphics paths in the game for Direct3D 9 and OpenGL ES 1.1. Garage Inc. uses only fixed-function graphics, which is common for casual games.

The OpenGL ES 1.1 path was developed primarily for iOS as that is the commonly available graphics API on that platform. During development, a desktop OpenGL 1.5 path was also developed for testing. For GameTree TV, the option was available to use OpenGL ES 1.1 or Direct3D 9 for graphics. For Mac OS X there was the option of using OpenGL 1.5 or Direct3D 9 for graphics. In both cases, TransGaming's enablement technology provided enough performance that the Direct3D 9 path was used for all non-iOS implementations. This allowed development to focus on a single graphics path outside of the mobile space and still provide good performance on a variety of platforms.

Input Devices

Another significant challenge is how to deal with the different input devices that are present on different platforms. Subtle differences between input devices on different platforms can have a profound impact on how the game plays. Desktop and notebook computers as well as Smart TV platforms such as GameTree TV provide various devices for controlling a cursor. Consoles typically provide some variant of a gamepad with the addition of motion controls and body and gesture tracking in recent years. Mobile platforms sometimes offer inputs that mimic gamepad controls, but platforms like iOS have been successful with just a touchscreen and some motion controls.

In the case of Garage Inc., the game was designed from an early stage such that its controls mapped easily to both a standard mouse-like control as well as a touchscreen.

The technical programming details of interfacing with input on Mac and GameTree TV was handled by TransGaming's Cider and GameTree TV SDKs, however, a significant amount of effort was put into play-testing and balancing the game on each of the various platforms. Variables such as time limits, game speed, response times and achievement levels were all tweaked. For all of these, the different input controls were the biggest factors in determining how these variables needed to be adjusted. TransGaming has also setup a Khronos working group to identify and incorporate common input standards that can be leveraged across platforms.

While the biggest concern for developing on multiple platforms is often the risk in the allocation of time and budget, software and enablement technologies such as those offered by TransGaming's Cider and GameTree TV SDKs have made this process significantly easier, as can be seen through the development of Garage Inc. Smaller development studios can now benefit greatly from enablement technologies that continue to provide a way of making multi-platform game releases more accessible, and making the distribution process much simpler. For more information on TransGaming's enablement technologies, visit www.transgaming.com.

MULTI-PLATFORM DISTRIBUTION

A great game that can thrive on a wide range of platforms is an asset, however, the risk tends to outweigh the "multi-platform" appeal. What developers must take into account is the end value that their game will have. There is considerable untapped value concerning this issue, for example, when developers think "multi platform" they think of the added stress of multiple technical considerations and issues, however, when an audience hears "multi platform" they get excited for a product that will likely be available for them. The marketing benefits of multi platform enablement are invaluable:

Increase in market size - Games can immediately be available to multiple markets, significantly expanding the number of potential customers.

Cost-efficient marketing - Having a product available on multiple platforms allows developers to efficiently build upon existing marketing, rather than increasing spending.

To get started with multi-platform enablement, developers can incorporate *Cider*, TransGaming's portability engine that allows PC games to be enabled on Apple's Intel Macs, eliminating the need to re-develop a game from scratch. Similarly, the GameTree TV SDK allows developers to create and distribute their interactive, gesture-controlled content through TransGaming's GameTree TV Platform.

REVIEW BY TOM CARROLL

ADOBE
Creative Suite 5

I'M GUESSING THAT MOST OF THE readers of *Game Developer* have seen Pixar's delightful superhero romp *The Incredibles* at least once. I myself have seen it numerous times, and that's why I remember a little tossed-off line from the beginning, where the young Mr. Incredible says, "No matter how many times you save the world, it always manages to get back in jeopardy again. Sometimes I just want it to stay saved, you know, for a little bit? I feel like the maid; I just cleaned up this mess! Can we keep it clean for... for ten minutes!?!"

That has to be how the people who work at Adobe on its Creative Suite products must feel. They ship CS4, but then before the blush is off the bloom, it's time to gear up for CS5. Their behind-the-scenes quotes might be something like, "No matter how many times we make Photoshop better, there's always some sort of cockamamie new feature set to include. Maybe people are still okay with it the way it was! You know, can't we just leave it alone for a year?"

But of course they can't, can they? What was good can always be better, and what was better really needs to be excellent. If you're not forging ahead, then you're falling behind.

It's been a year or so since Adobe CS5 was released, so we can now observe it through the lens of time to determine how well it has served the game development community as we wait for the inevitable upgrade.

I'm glad to report that, for most people's money, the number on the box could have been a symbol, and with that symbol, the product would have been called Adobe CS∞. Yes, that's an infinity symbol.

With all the goodies in the latest CS version that can help game developers do their jobs, Adobe really doesn't need to do another release.

YOU CAN SHOP THE 'SHOP

)))) Let's hit Photoshop first, because it is the sine qua non for texture artists, matte painters, and anyone else that produces high-quality 2D art. Here are the main points:

Better Subject-Isolating Tools

Over the years, I have probably spent the most inordinately unproductive moments of my life trying to isolate objects that I want to move to a new layer or remove so that I can use them in another application, such as Flash or After Effects. While the old selection tools were fine, they took time to use well, and for most artists, time is in short supply. The new advanced selection tool in CS5 lets you isolate subjects from their backgrounds with much greater speed and accuracy, even with the most nefariously complicated subject ... hair. Not only does the new tool work great, but it is so much better now that it banishes any thought of downloading someone's 3rd party tool that accomplishes the same thing. Take that, third-party tool!!

Content Aware Fill

Content Aware Fill is undisputedly the most remarkable new feature in Photoshop. Think of it as a merging of the Patch Tool and the Spot Healing Brush—but that doesn't quite do it justice. Create a selection with the Lasso, Marquee, or any other selection method and hit the Delete key. Photoshop displays a dialog box, which prompts you to choose Content Aware Fill. Hit OK and Photoshop automatically detects the content surrounding your selection and produces a fill based on what it finds there. The ultimate graphics smackdown. You'll inevitably have to check the results and look for seams, but you're way ahead of the curve using this feature.

Puppet Warp Transformation

This feature absolutely knocks it out of the park by allowing for the addition of control points to an image so it can be transformed using the points as guides. For one of my projects, I incorporated a savage samurai into the menu screen I was producing, but my art lead wanted the figure's blade raised instead of pointing forward. Using the Puppet Warp, I was able to quickly map the arm and shoulder, put control points on the arm segments, and then rotate the arm upward. The results needed a little tinkering, but (as with Content Aware Fill) the ability to move the arm so easily gives you plenty of time to modify the results to make it perfect. My boss ended up happy, and I was off to the next task.

HDR Toning

HDR stands for High Dynamic Range, a photographic technique that is characterized by intense light and shadows. Think drama! Use the new HDR Toning feature to replicate that effect using just one photo. While I haven't used this too much in my own work, it is really fun to fiddle around with at 6:30 pm while you're waiting for the company-sponsored pizza dinner to arrive.

Now, let's talk about some of the other programs in CS5 that benefit game development.

AFTER EFFECTS

)))) Rotoscoping is a repetitive function that almost everyone nowadays uses After Effects to accomplish with varying degrees of success and difficulty. With CS5, the Roto Brush tool enables people who never thought that they could do rotoscoping (because it would take too long or be too hard) to separate various foreground elements from their backgrounds. It is crazy simple. The most unfortunate part is that, unlike most of the other software applications, After Effects for CS5 requires a 64-bit OS (and scads of memory) to take advantage of the new features. Boo hoo.

ADOBE
CREATIVE SUITE 5
http://adobe.com

PRICE

- $129 Subscription
- $549 Upgrade
- $2,599 Full

SYSTEM REQUIREMENTS

- See website

PROS

1. More intelligent help features that deliver the goods with a lot less time and effort
2. Content Aware Fill is simple the "to die for" feature of the century
3. It's cheaper to upgrade if you're already using a Creative Suite version

CONS

1. After Effects requires a 64-bit processor (and a huge section of humanity is still trying to bleed along on their old XP-powered PCs)
2. No supports for upgrading a single application to CS5 status
3. Transferring vital settings, presents, Dreamweaver Snippets, Bridge Favorites, and so on continues to be a major pain in the asterisk

NO FLASH IN THE PAN

)))) As for Flash CS5, well, if you are a programmer using Flash in your daily work, you should absolutely get this release. There's no doubt about it, the enhancements to the workflow and code editor alone will pay for the upgrade. If you're a designer, then maybe not. The new brushes for the Deco tool and the improved IK are hardly solid selling points. If you need better control of text, the TLF integration will save you a lot of time. However, if you're a designer that wants to get into programming ActionScript, then this is the release for you; the new Snippets function can give you a good starting point for learning.

Speaking of learning, I've found an excellent book chock full of useful, well-laid-out information titled *How to Cheat in Adobe Flash CS5: The Art of Design and Animation* by Chris Georgenes, 2010, Focal Press. It will help you to become a superstar in the time it takes most people to simply go supernova (from the stress of picking up Flash animation skills, of course).

NEGATIVE TALK: MIGRATION

)))) It used to be that migrating from package to package was no big deal. Now that we have suites of software, each more complex than and different from the last, migration, while not the biggest deal, has at least become a "pretty big deal." Amazingly, Adobe doesn't allow someone to simply upgrade their existing installation. That's okay for people who want to keep their CS3 (and CS4) while also running their shiny new CS5, but for the rest of us, well ... we don't want the hard disk clutter that keeping multiple versions requires. Add to this the fact that there is no easy way to transfer your presets, settings, Dreamweaver Snippets, and so on ... you get the picture. Would it be too much to ask to streamline this process? I realize it's impossible to ask for it to be fun, but at least make it less arduous than filing the average long form tax return.

PRICE IS THE SPICE

)))) Now that you've kicked the tires, what is this going to cost? Prices for Adobe CS5 range between $1,300 and $2,600, depending on which package you buy, with upgrades priced between $500 and $1,500. Photoshop CS5 alone will cost $700, or $200 for an upgrade. Photoshop CS5 Extended, which has some additional tools, will cost $1,000, or $350 for an upgrade.

KICKIN' IT

)))) With all the goodies in CS5 that can help game developers do their jobs, Adobe really doesn't need to do another release. But they will, of course. And it will be really good, even bordering on excellent because it will provide new tools that kick it in ways we never thought anyone could kick it.

And if that were to happen (and it will), we'll all be just like the little boy in "*The Incredibles*" ... the one on the tricycle watching in awe, completely slack-jawed as our beloved hero does his heroic best to make our lives more productive.

And, like him, we'll probably say, "That is totally wicked!"

TOM CARROLL *has worked long and hard for such industry leaders as BlueSky Software, Interplay Entertainment, and Rockstar San Diego. He continues producing art for games and comics, and is learning the Lua programming language. He is currently developing Space Baby (iPad / iPhone / Android) for Retro Activ Games.*

AND NOW FOR SOMETHING COMPLETELY DIFFERENT...

THERE'S MORE THAN TWO WAYS TO DO 3D

IN THE MIGHTY ARSENAL OF GAME INDUSTRY CLICHÉS, ESPECIALLY AMONG artists, you can't find a trustier weapon than the old saw "it's the artist, not the tools." It's hard to argue with, especially if you've been around long enough to remember the days when merely knowing your way around the Max command panel or the Maya graph editor made you an "artist." No amount of tech can substitute for an observant eye and an unpredictable perspective.

That said, it's rare to meet professionals in any field who don't actually care about their tools. Mechanics have strong opinions regarding the pros and cons of different kinds of air wrench. Pro golfers spend lavishly on clubs, and even on cleats and gloves. Painters search endlessly for the perfect brush.

Of course, the same impulse comes naturally to us too, even though the realities of studio life mean we don't always have free reign to choose our tools. The official Pixel Pusher position has always been that freeing yourself from slavish dependency on a single art package is a good idea, both for your art and your career. That said, it's a rare individual who can hop back and forth between Max, Maya, XSI, and what have you with no personal feelings.

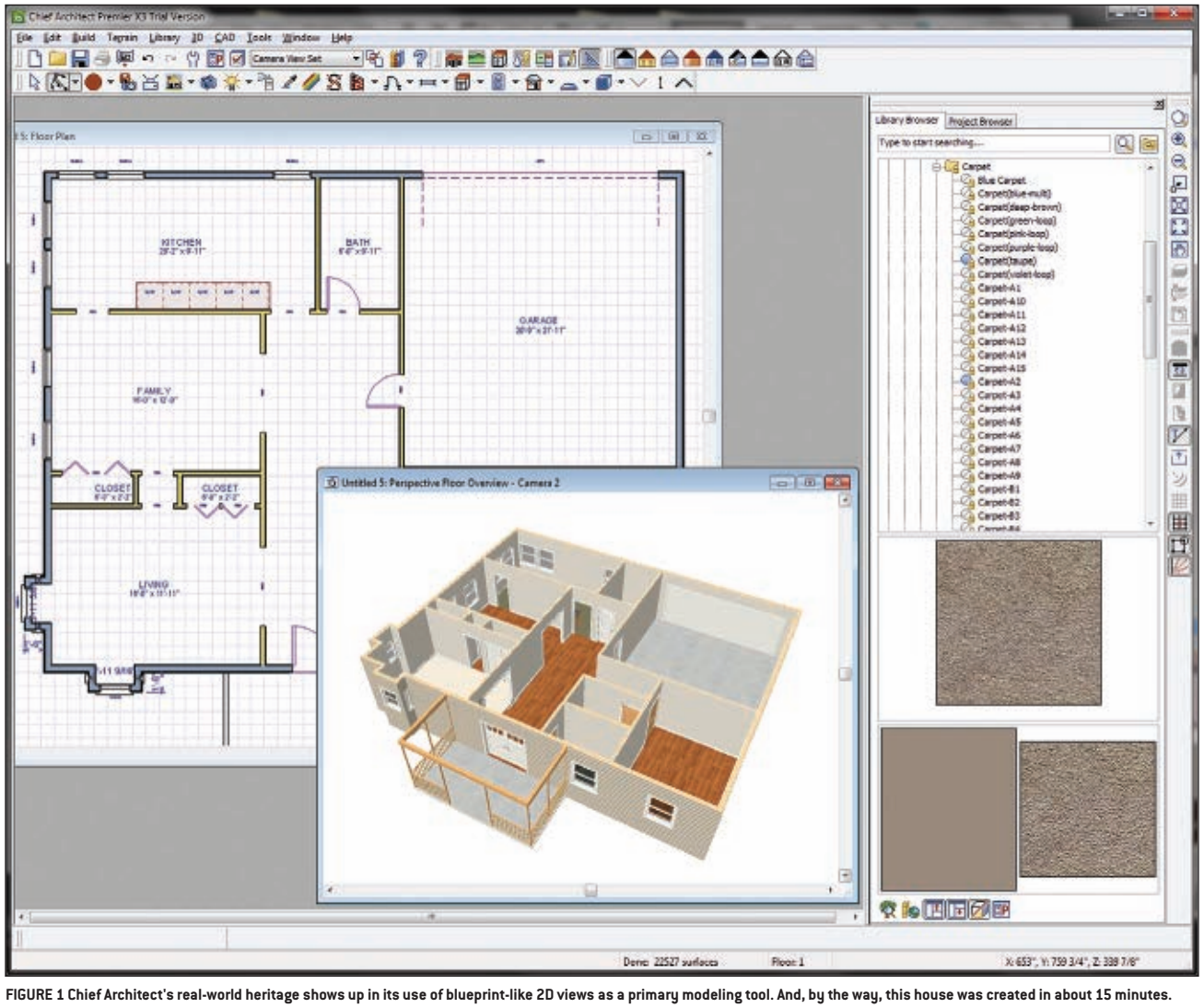


FIGURE 1 Chief Architect's real-world heritage shows up in its use of blueprint-like 2D views as a primary modeling tool. And, by the way, this house was created in about 15 minutes.

The emotional tug of working the way you want is so powerful that a very substantial minority of artists find ways to sneak their preferred tools into the pipeline by any means necessary. Have you ever worked at a studio where there *weren't* a few people secretly FBX-ing or OBJ-ing their work back and forth on the sly so they could stay in their comfort zone? You can't find a better illustration of the emotional bond we have with our tools than seeing somebody spend three or four thousand bucks of their own instead of using the tools the boss provides for free.

Though the boss may say one thing, it's good to remind yourself that Max, Maya, and other familiar tools aren't the only games in town. We should resist trying to shoehorn them into every job that comes our way just because they're so familiar. Despite their power and flexibility, they can't do everything. Plenty of teams have found out the hard way that building a big open world inside a package originally designed for modeling and animation of human-sized characters is barely possible, and rarely pleasant. Very few of us would seriously consider sculpting a multi-million poly character in an animation package rather than a subdivision modeler. No matter how much you love your familiar hotkeys, that muscle memory isn't much of a consolation if you can count to twenty every time the screen redraws.

Conversely, a tool that is custom made for one task can be a huge productivity booster. As Aristotle said, the measure of a tool is its fitness for the job at hand. The most familiar illustration of this idea is Zbrush, a program which provided so much power to frustrated character modelers that they embraced it in spite of an awkward interface and daunting learning curve. In a very different genre, SpeedTree and similar vegetation modeling packages are highly productive alternatives to hand modeling. Many of us secretly believe we could do better than the knobs and sliders, but few of us can spare the time to hand-sculpt every twig.

Even though moving beyond the comfort zone of the familiar packages can be a little nerve-wracking, software packages like Zbrush and SpeedTree illustrate how big the payoff can be. With that in mind, this month we're going to spotlight a couple off-the-beaten-track tools that can be helpful for specialized tasks, which can give you a fresh perspective on your old favorites. Even if it isn't true that the grass is always greener on the other side, it's still good to take a peek over the fence once in a while.

CHIEF ARCHITECT

» Chief Architect ($2,195 MSRP, www.chiefarchitect.com) is a great example of the benefits of doing one thing very well rather than trying to cover all possible 3D tasks. It's a popular architectural modeler used by architects and contractors for residential and small-scale commercial construction. It falls midway in price and complexity between a full-scale architectural drafting system like AutoCad and the sort of DIY home improvement packages you see in Best Buy.

Since the software is designed for working architects, its primary working metaphor is a floor plan (see Figure 1). While 3D views are also available, the core workflow involves drawing 2D plan views, almost like an old-fashioned blueprint drafter. Max and Maya veterans might expect to find this constricting, and it does take some getting used to. But for most real world architectural tasks, it's perfectly natural and highly efficient, although it's not going to be much help building a spaceship interior or twisty series of fantasy caverns.

Since Chief Architect is a practical tool for real world builders and designers, it encodes a huge amount of knowledge about real-world architectural construction. You place construction elements like walls, doors, and windows instead of managing individual vertices or polygons. The actual 3D geometry is generated automatically for you. When you decide to move a door or window, you're not responsible for managing the seams or stitching up the geometry. Just grab the window or doorframe and move it to the new location, and the program will rebuild the underlying 3D meshes.

This structured approach is very powerful if you're working on realistic buildings. Chief Architect can generate not only the visible surfaces (walls, floors, and so on) but also substructures and framing, which can be very handy if you need to make destroyed or half-built buildings. On the other hand, you'll need to put in some management time to exclude that extra geometry in most game applications.

The Chief Architect narrow focus on conventional modern architecture makes the program highly efficient. Parametric walls and features are easy to adjust as you evolve your own ideas or respond to feedback from designers. Operations that most modelers would shy away from—like moving a staircase or adding new windows—are fast and easy. Moreover, a lot of minor modeling details are handled for you: You can get conventional pieces like window frames, doors, and even cabinets or appliances from the program's extensive library. (See Figure 2.)

After you've designed a building in Chief Architect, you can get it into a traditional 3D package using DXF export. Unfortunately, this step typically involves manual cleanup. The DXF files tend to include a few faces with randomly flipped normals, and many objects include stray vertices sitting at the world origin or degenerate faces. Naturally, you'll also have to add game-read materials and textures. You can't expect to send geometry straight from Chief Architect into your game without some handholding on your part.

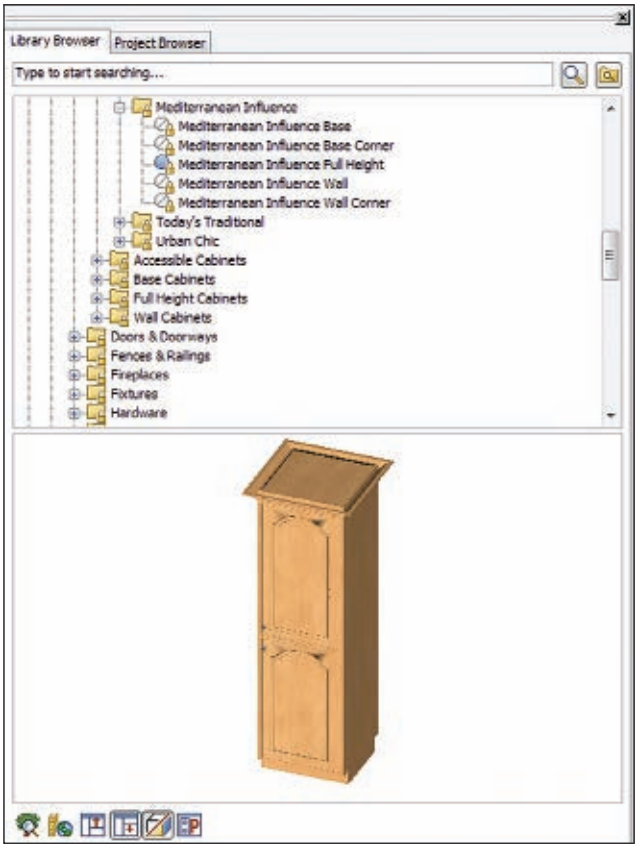


FIGURE 2 CA also includes an extensive library of ready-made assets you can use—this cabinet, for example.

Despite these problems—and of course the cost of climbing another complex software learning curve—the efficiency of a dedicated architectural package is enormous compared to hand-modeling buildings in conventional animation software. This makes up for the geometry cleanup and texture work. If your game involves a modern environment, Chief Architect and competing architectural packages are definitely worth investigating.

HOUDINI

>> If Chief Architect showcases the power of specialization, Houdini ($1995—$6695, www.sidefx.com) excels at tackling unusual kinds of content creation. Houdini has a long history in the 3D world. It's been around as long as Max and Maya, debuting in 1996. Despite this pedigree, it's rarely found in game studios. Houdini is a fixture in high-end effects houses and film studios as a way to create complex, procedural solutions to baffling special effects and simulation problems.

Houdini can also be used as a general-purpose animation and modeling tool in the familiar Max/Maya mold, and it shares many interface conventions with those packages. Nonetheless, it's a very different beast. Houdini is heavily dedicated to a procedural workflow instead of conventional hand-built models and animations. That procedural emphasis makes it appealing to artists who have to handle complex effect setups—for example the DEAD ISLAND trailer that made the rounds a couple months back showcased lighting, gore effects, and procedural glass shattering all done in Houdini.

The core of the package is a node-based workflow similar to, but much more extensive than the Maya HyperGraph, or compositing packages like Nuke. Artists wire up networks of nodes or "OPs" to create complex effects, models, or animations (see Figure 3). These OPs represent everything from familiar operations such as booleans, UV projections, and deformations, to exotica like metaballs, voxels, and L-Systems.

As you can imagine, Houdini takes a very left-brained approach to artwork. The company pitches the node-based workflow as "construction history you'll never want to delete"—but given the testy relationship a lot of working artists have with construction history (see Pixel Pusher, December 2006, "The History Channel: How I Learned to Stop Worrying and Love Construction History") it's a tough sell for artists who prefer to tackle their problems vertex-by-vertex. Even enthusiastic Houdini fans admit that the learning curve is pretty steep... This is one of the reasons the package has found few adherents in the games business.

On the other hand, technical artists and effects specialists start salivating when they see what can be done with OPs. The extensive library of nodes really amounts to a kind of graphics programming language for art tasks. If you're the kind of artist who dabbles in muscle systems or dreams of building entire cities procedurally, Houdini is hard to beat. It's not just for rendering fancy offline effects, either—there are a lot of geometry processing tasks that can be automated using a combination of procedural networks and scripting. Houdini is a powerful tool for tackling unpleasant tasks like generating low-res collision geometry or debris models.

Another plus is the fact that Houdini's procedural tools handle 2D image compositing and shading. The ability to connect geometric OPs and image OPs provides a powerful toolkit for developing shaders, especially the kind that require very abstract, non-representational textures like vector flow fields or anisotropy maps. For a good example of what can be done, you might check out the paper presented by Valve's Alex Vlachos at SIGGRAPH last year (see References), which touches on some of the ways they used Houdini to create the water flow maps for PORTAL, LEFT4DEAD 2, and PORTAL 2, a task that would have been mind-bendingly difficult for artists armed only with polygon and paint tools.

Despite Houdini's impressive strengths, it's not likely to supplant Max or Maya in the games industry, thanks to its unusual paradigm and steep learning curve. Nonetheless, it's an amazingly powerful tool in the hands of an artist who's willing to grapple with the node-based workflow and really embrace proceduralism. If you think that describes you, there's a free, watermarked version to play with, which you can find in resources.

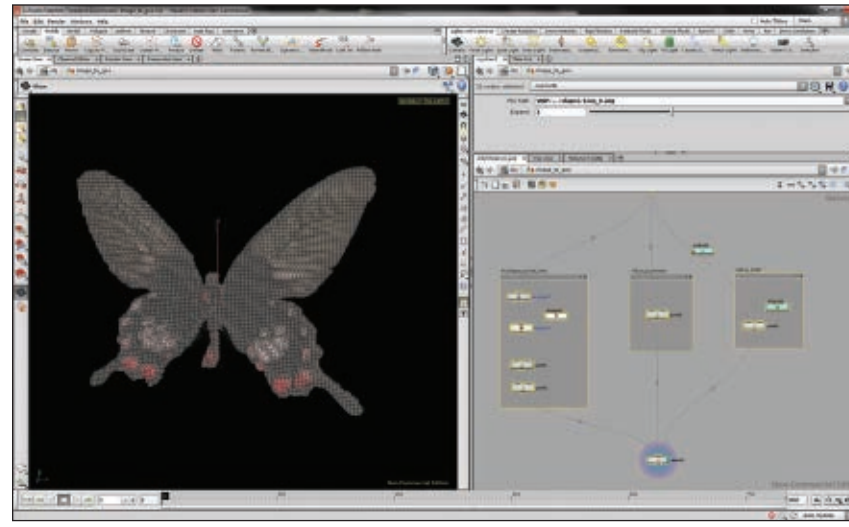


FIGURE 3 Houdini relies on complex procedural networks to create sophisticated effects. Here, a procedural network generates displaced geometry from a compositing image.

THE FAR REACHES

>> It's pretty unlikely that many of us will jump ship to a new package simply because of a cool feature. The history of XSI shows that even really hot products have a tough time converting dedicated users. Our tools are part of us, literally, considering how many of those hotkey combinations have stolen into our hindbrains and rewired our fingers.

However, even though that bonding makes us more productive and lets us get past the purely digital side of the work to focus on our art, getting too close to any particular piece of software can lead us to miss out on other opportunities. It's good professional discipline to keep up with more than just the next version of Max or Maya, and even if you don't think these (or other packages) are right for you, just being familiar with alternatives will make you a more informed user of any tool.

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 the technical art director at Seattle's Undead Labs.*

REFERENCES

ALEX VLACHOS DISCUSSES HOUDINI:
www.valvesoftware.com/publications/2010/siggraph2010_vlachos_waterflow.pdf

HOUDINI'S TRIAL VERSION:
www.sidefx.com/index.php?option=com_content&task=view&id=589&Itemid=221

PORTABLE PERFORMANCE

A MOBILE GRAPHICS API WISHLIST

ILOMILO, a puzzle game from SouthEnd Interactive, was made using OpenGL ES2.0.

////////////////// Most mobile platforms are currently based on OpenGL ES 2.0. While it is *much* better than traditional desktop OpenGL, there are places where it limits performance or does not expose the most interesting hardware features. It's the developers that influence the technology, and with that in mind, here's a small wishlist for the future of GLES2.0.

A pipe dream would be starting from scratch, getting rid of all of OpenGL's baggage, and coming up with a much cleaner, leaner, and better API, especially if it's designed to only support a particular platform. Being more realistic, I'll be focusing on, in my limited understanding, the short-term low-hanging fruit that could extend or patch the existing GLES2.0 API.

Another caveat for my feature wishes is that there already are GPUs that could do all this; I just want them to be available through the API. Completely new GPU architectures could bring way more features, but that's a story for another day.

Paraphrasing one tweet: "*400 draw calls saturate the CPU?! I could do ten times more on the PSP!*" It is certainly true that GLES2.0 adds non-trivial overhead. How much depends on the OS and drivers used, and of course, some of that overhead might be optimized away with future OS/driver releases.

I won't focus on raw throughput of the graphics API, though. Instead, let's talk about things that cause hiccups or unexpected performance problems, or are just plain annoying.

NO INDICATIONS WHEN SOMETHING EXPENSIVE MIGHT HAPPEN

>> Due to some flexibility in GLES2.0, there might be expensive things happening at almost any point in your frame. For example, binding a texture with a different format might cause a driver to recompile a shader at draw-call time. I've seen 60 milliseconds on an iPhone 3Gs at first draw call with a relatively simple shader, all spent inside the shader compiler backend. Sixty milliseconds is a lot! There are a lot of things that can cause such performance hiccups; texture formats, blending modes, vertex layout, non-power of two textures, and so on.

Suggestion: **Work with GPU vendors and agree on an API that would try to make guarantees as far as when the expensive resource creation/patching work can happen, and when it can't. For example, somehow guarantee that a draw call or a state set will not cause any object recreation or shader patching in the driver.**

This might not be possible in all cases, and that's fine. But try to move as many expensive "resource creation" operations into separate API calls as possible. I quite like what Direct3D 10/11 does there; API calls that create objects

are very separate from APIs that set state or issue drawing commands.

OFFLINE SHADER COMPILATION

>> GLES2.0 has the functionality to load binary shaders, but it's not mandatory. Some of the big platforms (iOS, I'm looking at you) just don't support it.

Now, of course, a single platform (like iOS or Android) can have many different GPUs. Therefore, you can't fully compile a shader offline into final optimized GPU microcode, and some runtime states might warrant shader patching or recompilation, which is fine. But some of the full compilation cost could very well be done offline, without being specific to any particular GPU or runtime states.

Suggestion: **Come up with a platform-independent binary shader format. Something like the D3D9 shader assembly is probably too low level (it assumes a vector4-based GPU, a limited number of registers, and so on), but something higher level should be possible. All the shader lexing, parsing, and common optimizations, such as constant folding, arithmetic simplifications, and dead code removal, can be done offline. It won't speed up shader loading by an order of magnitude, but even if it's possible to cut it by 20%, it's worth it. And it would remove a very large bug surface area too!**

Additional suggestion: **Provide more control when runtime shader compilation happens. Currently the most robust approach is to set the state/textures/shaders that you would use and draw a zero-pixel-area triangle. That's both stupid and wasteful, if you think about it! I'd take something like `glWarmupCurrentShaders()`, or even better, a way to do that asynchronously on the second CPU core in your iPad 2.**

TEXTURE LOADING

>> A lot of (all?) mobile platforms have unified CPU and GPU memories, but to actually load the texture, we have to read or memory map it from the disk, and then copy into OpenGL via `glTexImage2D` and similar functions. Then, depending on the format, the driver would internally do swizzling and alignment of texture data.

Suggestion: **Can't most of this cost be removed? If for some formats it's perfectly, statically known what layout and swizzling the GPU expects, then can't we just point the API to the data we already loaded or memory mapped? We would still need to implement the `glTexImage2D` case in the event that a totally new strange GPU comes along that needs the data in a different order, but why not provide a faster path for current GPUs?**

VERTEX DECLARATIONS

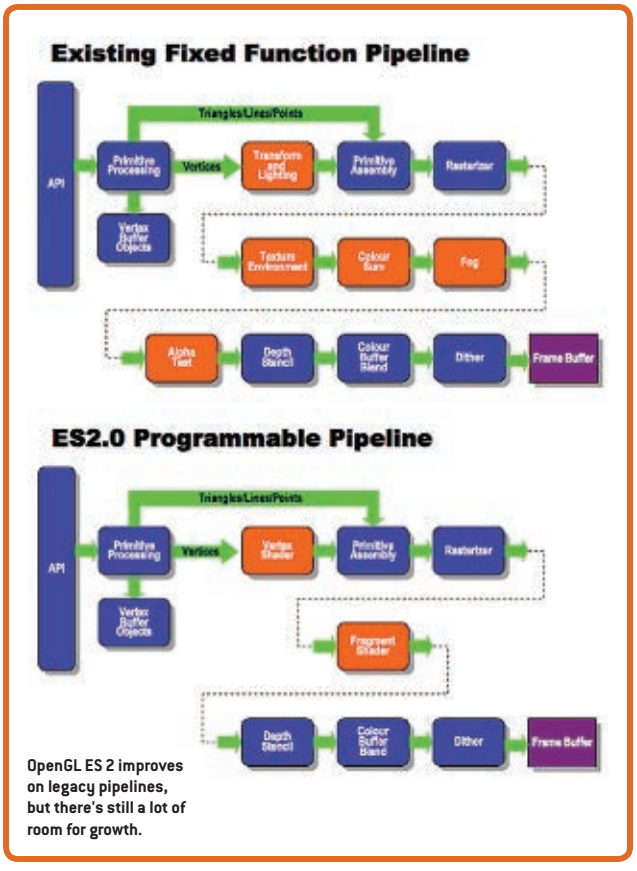
>> In unextended GLES2.0, you have to do a ton of calls just to set up vertex data. `OES_vertex_array_object` is a step in the right direction (see References), providing the ability to create sets of vertex data bindings ("vertex declarations" in D3D speak). However, it builds upon an existing API, resulting in something that feels quite messy. It feels that by starting from scratch we could have a much cleaner API. Just take a look at the vertex declarations in D3D!

Suggestion: **Clean that up! It would probably need to be tied to a vertex-shader-input signature (just like in D3D10/11) to minimize shader patching, but I'd be fine with that.**

BIND-TO-EDIT AND MUTABLE OBJECTS

>> The need to "bind" an object you'll be operating on, instead of passing it directly to the needed functions, is just annoying for the developer. I could very well imagine it does nothing good for the driver writers either. `EXT_direct_state_access` for GLES2.0, please! (See References.)

Somewhat related to that, most objects aren't immutable in OpenGL (ES). I could create a 32x32, 8-bit/channel RGBA texture, use it a bit, and then make the same texture become a 400x300, 24-bit depth + 8-bit stencil texture. Flexible? Of course. Useful? Not so much. Does it complicate the driver and the GLES runtime? Very likely.



OpenGL ES 2 improves on legacy pipelines, but there's still a lot of room for growth.

SHADER UNIFORMS ARE PER SHADER PROGRAM

>> Shader uniforms ("constants" in D3D speak) are not global; they are tied to a specific shader program. I don't quite understand why, and I don't think any GPU works that way. This causes complexities and/or performance loss in the driver (it either has to save and restore all uniform values on each shader change, or have dirty tracking on which uniforms have changed). It also causes unneeded uniform sets on the client side; for example, instead of having `view*projection` matrix set just once per frame, it has to be set for each shader program that we use.

Suggestion: **Just get rid of that! If you need to not break the existing spec, how about adding an extension to make all uniforms global? I propose `glCanHaz(GL_OES_GLOBAL_UNIFORMS_PLZ)`.**

PROGRAMMABLE BLENDING

>> At least two out of three big current mobile GPU families (PowerVR SGX and NVIDIA Tegra 2) support programmable blending in the hardware. Maybe all of them do this, and I just don't have enough data. By "support it in the hardware," I mean the GPU has no blending hardware, and the drivers add "read current pixel and blend" instructions to the shaders, or the GPU has blending hardware for commonly used modes, but fancier modes use shader patching with no severe performance penalties.

Programmable blending is useful for various things, from deferred-style decals (blending normals is hard in fixed function), to fancier Photoshop-like

blend modes, to potentially faster single-pixel image post-processing effects (like color correction).

Currently, only NVIDIA exposes this capability through the `NV_shader_framebuffer_fetch` extension (See References).

Suggestion: **Expose programmable blending on other hardware that can do this! It's fine to not handle hard-edge cases (for example, what happens when multisampling is used?). We can live with the limitations.**

DIRECT, FAST ACCESS TO FRAME BUFFER ON THE CPU

» Most (all?) mobile platforms use a unified memory approach, where there's no physical distinction between "system memory" and "video memory." Some of those platforms are slightly unbalanced, such as a strong GPU coupled with a weak CPU or vice versa. More and more of those systems will have multicore CPUs. It might make sense to take a similar approach to what Sony's doing with the PS3 these days and offload some of the GPU work to the CPU(s).

Image processing, deferred lighting, and similar tasks could be done more efficiently on a general purpose CPU, where you aren't limited to a "one pixel at a time" model of programmability.

Suggestion: **Can we get a pointer to framebuffer memory? Of course this is grossly oversimplifying all the synchronization and security issues, but there must be something that can be done to exploit the unified memory model. Right now it just sits there largely unused, with GLES2.0 still pretending the CPU is talking to a GPU over a 10-meter-high concrete wall.**

EXPOSE TILE BASED GPU CAPABILITIES

» The PowerVR GPUs found in all iOS and some Android devices are so-called "tile-based" architectures. So is, to some extent, the Qualcomm Adreno family.

Currently, this capability is mostly sitting behind a black box. On PowerVR GPUs, the programmer knows that "overdraw of opaque objects does not matter" or "alpha testing is really slow," but that's about it. There's no control over the whole rendering process, even if some tasks could benefit from having more control over the whole tiling thing.

Take, for example, deferred lighting and shading. The cool folks are already doing it tile-based on DirectX 11 or PS3 (DICE, for example—see References).

On a tile-based GPU, all rendering is already happening in tiles, so what if we could say, "Now we'll work on tile A. Render some objects. Now switch to tile B. Render some other objects." Maybe that way we could achieve two things at once. First, we'd have better light culling because it's at tile level, and second, most of the data could stay on this super-fast on-chip memory, without having to be written into system memory and later read again. Memory bandwidth is very often a limiting factor in mobile graphics performance, and the ability to keep deferred lighting buffers on-chip through the whole process could cut down bandwidth requirements a lot.

Suggestion: **Somehow (I'm feeling very hand-wavy today) expose more control over tiled rendering. For example, explicitly say that rendering will only happen to the given tiles, and (that these textures are very likely to be read just after they are rendered into, so don't resolve to main memory if they fit into the on-chip memory.**

There's already a Qualcomm extension that's moving toward that area (`QCOM_tiled_rendering`—see References), though it seems to be more concerned with where rendering happens. More control is needed in terms of how to mark FBO textures as "keep in on-chip memory for sampling as a texture please."

OPENCL

» Current mobile GPUs already are, or very soon will be, OpenCL capable, and OpenCL can be implemented on the CPU, nicely SIMDified via NEON, and can use multicore. *DO WANT!* (And while you're at it, let's do everything possible to make interop between CL and GL faster.)

This can be used for a ton of things: skinning, culling, particles, procedural animations, image post-processing, and so on. And with a much less restrictive programming model, it's easier to reuse computation results across draw calls or frames.

Couple this with "direct access to memory on the CPU" and OpenCL could be used for more things than graphics. Of course, I'm grossly oversimplifying here and ignoring the whole synchronization/latency/security elephant!

ARAS PRANCKEVICIUS *is a lead graphics programmer (which is just another word for troublemaker) at Unity Technologies. Each day at work, he's making the Unity game engine capable of producing prettier pictures or doing something faster. He's been involved in almost all aspects of Unity's graphics code, and has seen the company grow from 3 to 100+ people over several crazy years. Before that, he was involved in small game productions, computer vision, and the demoscene, and has published articles in the ShaderX book series.*

RAGING THUNDER 2 by TriplePoint also made with OpenGL ES2.0.

references

OPENGL / OPENGL ES EXTENSIONS

OES_vertex_array_object
www.khronos.org/registry/gles/extensions/OES/OES_vertex_array_object.txt

EXT_direct_state_access
www.opengl.org/registry/specs/EXT/direct_state_access.txt

QCOM_tiled_rendering
www.khronos.org/registry/gles/extensions/QCOM/QCOM_tiled_rendering.txt

NV_shader_framebuffer_fetch
http://developer.download.nvidia.com/tegra/docs/tegra_gles2_development.pdf

TILE-BASED DEFERRED SHADING BY DICE

BF3 DX11 Rendering
www.slideshare.net/DICEStudio/directx-11-rendering-in-battlefield-3

BF3 SP3 Rendering
www.slideshare.net/DICEStudio/spubased-deferred-shading-in-battlefield-3-for-playstation-3

SWEET VINDICATION

BUILDING MOMENTS OF TRIUMPH INTO YOUR DESIGN

FOR YEARS NOW, THE STONE TEMPLE PILOTS have been my great white whale.

Ever since the original ROCK BAND came out in 2007, I've been banging on the skins, taking drum skills that started as shamefully comical (I believe I caught the dog laughing at me once) and slowly improving them through earnest practice. Improvement was relatively quick: I slogged all the way from Easy up to Expert, and one by one songs that seemed unplayable were conquered. But a few of those songs continued to elude me. One such song was "Vaseline."

At first, I could barely finish it, and once I did, my scores were pathetic. The song was a chore for me to play. I liked the song—it has a certain nostalgia factor that takes me right back to my unkempt days—but as I progressed through the game, I stopped playing it. I didn't stop playing the drums, though; I kept playing, downloading DLC, buying expansions, and eventually got the pro set. But I kept avoiding "Vasoline" until one day, three years later, it was thrown in a random set list by happenchance. As the familiar drum beat kicked in, I approached the song with a certain level of trepidation. But then a funny thing happened.

I destroyed it.

Apparently, somewhere along the way, I'd picked up enough drumming skills to not only skate by the song, but to utterly conquer it—gold star, top score, you name it. What once seemed borderline impossible now seemed shockingly simple, and the sense of victory that arose was well beyond that of beating your average song. It was the taste of sweet, sweet vindication.

Few moments in gaming are more powerful than that moment in which you completely own something that previously flummoxed you. Fortunately for us designers, this is a feeling that we can manipulate and inspire.

SKILL AND VINDICATION

>> No matter what the genre, you can see this effect in great games. It can take many forms, based largely on the kind of advancement that drives the game. In ROCK BAND, advancement is almost entirely based on skill, and as such, an increase in skill is what results in our little Eureka moment.

These moments can come to pass in almost any hobby, of course. Just ask the guitar player who realizes he can now play "Crazy Train" without looking down at the fretboard. Or the knitter who can now whip out a well-knit cardigan in a week instead of a hobbled-together disaster in three months. This sense and feeling of growth is one of the reasons these hobbies are so compelling and rewarding in the long run, and these quick and easy victories help spur the hobbyist on to harder challenges.

ROCK BAND.

It should come as no surprise, then, that similar feelings occur in almost any game with a strong skill element, such as casually clearing a board in seconds in MINESWEEPER or topping your friends list in BEJEWELED BLITZ. Or when you realize that you're building railroads while the AI is building chariots in CIVILIZATION. While going on a rampage in UNREAL: TOURNAMENT. After executing a flawless victory against a boss you could previously barely defeat in SOUL CALIBER.

As sweet as all of these little moments are, they do present a problem to the designer: only the player who invests the time to gain the skill is going to taste them. The trick is ushering the player along that path, in the hope that he doesn't get discouraged.

FAKING SKILL

>> This feeling can be manufactured. Progression in action games like DANTE'S INFERNO or GOD OF WAR is more based on stats than skill, as the player's capabilities are increased by the unlocking of more powerful moves and weapons that make them feel more godlike. The lack of a skill component doesn't dampen the feeling of sweet vindication that occurs later on, when monsters that previously stumped them as bosses or minibosses start showing up as easily dispatchable trash.

Nicole Lazzaro, president of the game consulting group XEODesign, often speaks of the various kinds of fun found in games. In particular, she points out that the best games on the market have a vicarious, immersive play loop that curves between easy fun and hard fun. Easy fun is soothing and enjoyable, whereas hard fun is challenging and interesting. The balance is tricky because if there is too much easy fun, then the game becomes a snoozefest; if there is too much hard fun, then the game becomes tedious and frustrating.

What GOD OF WAR does, then, is interesting. It takes moments that are clearly hard fun early in the game and turns them into easy fun later on; for example, a single minotaur is thrown at the player in the beginning, which takes time, skill, and a couple deaths to defeat. Later on, three minotaurs at a time are thrown at them, with the developers knowing full well that the player's abilities have been upgraded enough to make the task trivial. Beyond feeling awesome, this approach helps serve a couple of clear design goals: it provides a real sense of character growth and helps support the narrative that Kratos is slowly becoming powerful enough to challenge the gods themselves. It's also a handy reuse of assets!

VARYING FORMS OF VINDICATION

>> Skill based vs. stat based isn't the only way that you can see this. Boss fights such as those in EverQuest or World of Warcraft are highly skill-based fights, but the stat inflation that earned gear provides means that those fights get easier as time progresses, allowing players who lack the skill to taste that endgame content, while making it easy, sometimes trivial, for those who are more skilled.

Some see this as a weakness of the design. On the other hand, it does allow for more players to see the content while still cementing the hardcore reputation of the guilds that manage to defeat the bosses first. Even more interesting, the slow inflation creates a communal sense of growth as well. Once upon a time, WoW kingdoms shuddered at the name of Gruul, but now that epic 25-man boss can be defeated by an evading rogue and a healer.

The other interesting factor is how time figures in. In my Rock Band example, it took years to feel the moment of supreme triumph (although there were certainly smaller victories along the way). At the other extreme, the time lapse from frustration to conquest takes mere minutes in a game like Portal, not by skill or stats, but by fooling the player, ever so briefly, into thinking he was very clever. Indeed, games such as these are practically built around Eureka moments.

A THEORY OF VINDICATION

>> In his book *A Theory of Fun*, Raph Koster describes the cycle of fun found in games as one where the player learns a pattern, conquers the pattern, and then moves on to the next pattern, slowly expanding his mastery of the game. In this model, sweet vindication holds an important role: one of validation.

There is a perverse joy to be found in massively multiplayer games, in taking a fully maxed-out character back to the starting village and destroying the elite level-10 critter that was such a problem as a newbie, just as there is joy to be found in discovering that it's now easy to tackle a once-seemingly impossible song in Guitar Hero. Whether progression is based on skill, stats, or just time and labor, such moments act to validate the player's time and emotional investment in your game.

The careful designer can manufacture these moments by finding reasons to send players back to these earlier challenges, and to show him how far he's progressed. But be wary, as this moment is fleeting. An occasional revisit is fine, but mentally, the player has already mastered this pattern. Any long stays in this content will only result in annoyance and tedium.

THE DOWNWARD SLOPE

>> My favorite example of sweet vindication in action was the delicate game balance derived by the creators of Vampire: The Masquerade—Redemption. This Diablo-style RPG set in the World of Darkness universe was a tightly balanced game for the early going. It was by no means difficult, but still, one had to be somewhat careful and tactical when picking fights. And yes, occasionally you would die.

In the last act, however, the balance totally shifted. The end was in sight, and as good a time as I was having playing, I was ready to finish my experience and move on with my life. Most games throw their toughest challenges here. Vampire did the opposite; suddenly, the game threw waves of hordes in front of me, all of which were dispatched with superheroic ease. The rest of the game had conspired to set up this feeling, to make me feel like I had advanced and grown, and that I was now ready for the final confrontation. The phrase "it's all downhill from here" came to mind.

And then I encountered the final boss. Far from being the cream puff that his immediately adjacent minions were, he put up one of the toughest fights I faced in the game, utterly crushing my first couple of attempts; which was, perhaps to some degree, because I'd grown sloppy on the way in. And if the previous hallway had elevated me and my own sense of character growth—I went in feeling like one of the most powerful lords of darkness the world had ever seen—it elevated him as well. I became immediately aware that I was, at best, number two. This only served to make the final showdown more epic, and the sense of gratification all that much greater when I pulled off the kill.

VALIDATION

>> Some players play games to exhibit skill, while others do it to simply kill time in the drudgery that is their life. In both cases, players are devoting a sizable chunk of their lives to their hobby, embracing it and improving their skill at it. Making them feel good about this investment is never a bad thing.

In most cases, players are playing for the journey, not the destination. But the journey can be long, sometimes difficult, and at times tedious or overstay its welcome. During these moments, a little spot of sweet vindication can go a long way. After all, anytime someone travels anywhere inch by inch, it's useful to provide them with a mountaintop so they can realize they've traveled miles.

DAMION SCHUBERT *is the lead systems 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 Meridian59 and Shadowbane as well as other virtual worlds. Damion also is responsible for Zen of Design, a blog devoted to game design issues. Email him at* ***dschubert@gdmag.com****.*

WHEREFORE ART THOU, WIKILEAKS?

A CALL FOR TRANSPARENCY IN DIGITAL STOREFRONTS

IN MY MARCH COLUMN, I WROTE ABOUT WHAT I viewed as a growing element in the competition between digital storefronts (app stores). My assertion was that the future ability of platforms and storefronts to differentiate and compete would be determined by two things: first, their willingness to provide developers with flexible policies in order to let them innovate; and second, the e-commerce tools and information they provide to let developers guide that innovation in a timely fashion.

After the issue dropped at this year's Game Developers Conference, I had a number of developers reach out to me to further discuss the subject. Most agreed my points were relevant, but also noted that these elements are secondary to the fundamental factors of the platform installed base, the propensity of a platform's customers to buy games, and the price those games sell for.

These conversations also brought up some interesting issues related to transparency. One factor is the visibility (or lack thereof) into overall game sales on a platform or store, meaning the platform's ecosystem as a whole. Most platforms and stores have healthy levels of disclosure about how an individual developer's game has performed on the service. From that point though, if a developer wants to compare how their game is performing against others, information is less forthcoming.

VISIBILITY

» Information about the overall ecosystem for a platform or store can come from three places:

- The platform owner can release it (such as we see in limited form with Apple's top selling/top grossing lists on its AppStore). Generally, such disclosures are limited and self-serving. In the worst cases, they can be outright falsehoods meant to seed "top 10" lists with titles serving the platform owner's interests.
- It can be reverse engineered and/or modeled from public data and services. This is a non-trivial effort, and is usually taken on by market research firms who can then sell the information. As a result, such market research usually isn't cheap. (Analyses of the XBLA leaderboard stats to infer sales numbers are a good example here.)
- In the case of a publisher, distributor, or a large developer with many titles, they can sometimes have enough sample points on the sales curve to infer their own stats.

The problem for any small, independent developer is that two of the above list items are out of reach, leaving them at the mercy of whatever info the platform owner chooses to disclose.

There is, of course, another way to get information about how games are doing on different platforms and of overall platform marketplace health, and that is by talking to one another. Better-connected developers share sales figures and other data in conversation or via mail lists. While it beats being in the dark, it doesn't scale, and it doesn't offer nearly as complete a picture as the above methods.

TERMS OF SERVICE

» Another topic that came up surrounds deal terms around distribution arrangements. These are somewhat more accessible, since developers have to sign an agreement at some point, but they are viewed as opaque. For some, the terms aren't made available until late in the submission process, which may be after development has taken place. Then there's the fact that the implications of deal terms and their use is often left to the developer's interpretation. For many small developers, this means sifting through the legalese themselves.

In both cases, developers fall back on word of mouth when trying to navigate these waters, and once again it doesn't scale.

Recently, though we've seen other interesting developments. In April, the International Game Developers Association published an open letter regarding the deal terms offered by Amazon's Android app store, claiming developers were at risk of Amazon pulling the rug out from under their feet in terms of pricing should Amazon feel like drawing traffic to its store at any one developer's expense. At the time of this writing, Amazon and the IGDA are still exchanging barbs about this.

Daniel Cook, in his 2011 GDC lecture (available in the GDC Vault, and on his web site), discussed what he viewed as an inevitable part of a platform's lifecycle, where developers will be squeezed of profits to sustain platform revenue growth. He points out that developers can use the press as a tactic in that platform owners tend to shun controversy and bad developer or user community PR. Using the press to bring an issue into the spotlight can work to everyone's advantage.

FORCED TRANSPARENCY

» One can't discuss transparency today and not eventually get to the topic of Wikileaks. There has been no shortage of controversy surrounding Wikileaks, especially since the site's massive leak of US government military and diplomatic information. Controversy aside, we can still step back and view Wikileaks for what it is: an anonymous, crowdsourced vehicle of transparency.

And I guess what I'm asking is ... where is the game industry's Wikileaks?

I should be clear that I'm not asking people to start breaking NDAs and uploading their companies' revenue numbers. I do, however, think it inevitable that people will start taking the information that is available publicly, but which is hard to gather and sift through, and start crowdsourcing it in some way.

When and if that happens, it will mean several interesting things. For developers, it will serve as an equalizer, letting small guys make informed decisions using a depth of information that was previously only available to large developers and publishers. For platform and storefront owners, it will force them to consider how they'll feel about their distribution terms and performance statistics being aired in public and compared to competing platforms and services. This, too, will be good for developers, as well as for those platforms that do a better job serving them. In the meantime, it probably serves everybody well to consider how their current business practices would look if and when they do get "wikileaked."

KIM PALLISTER *works at Intel doing game industry forecasting and requirements planning. When not prepping the world for super-cool hardware, he blogs at www.kimpallister.com. His views in this column are his and do not reflect those of his employer.*

REWARDING AWARDS

GAME MUSIC MAKES GRAMMY INROADS

ART CRAVES ATTENTION, WHILE ATTENTION craves validation. The longer a form of art is in existence, the more it breeds established systems of recognition for its artists in the form of peer-recognized prizes and rewards. Whether Oscar or Emmy, Golden Globe or Tony, the entertainment industry is full of award shows, prizes, and statuettes. Each form of entertainment media has spawned its own trophies. In that regard, the game industry is no different with its Game Developers Choice Awards or Spike TV Video Game Awards; and while gaming audio has its annual Game Audio Network Guild Awards, game industry composers have been reaching for recognition from a larger group of peers.

Perhaps it's due to the large amount of crossover talent among composers who work on games, film, and television. Perhaps it's simply because of the wider name recognition of the award itself. Whatever the case, game composers have been petitioning the National Academy of Recording Arts and Sciences for years now to include video games among those eligible for the annual Grammy Awards.

GRABBING FOR GRAMMYS

>> The National Academy of Recording Arts and Sciences (or NARAS) is the organization that has been behind the annual Grammy Awards for over 50 years. For nearly a decade now, a dedicated group of game composers has been pleading the case for equal recognition among the Grammys for video game music alongside film and television scores. But that recognition has been slow to come.

Seven years ago, video games were first allowed for consideration of Grammy Awards, but they were not specifically called out by name. Instead, game scores were considered to be part of the "other visual media" section of the award for "Best Compilation Soundtrack Album for a Motion Picture, Television or Other Visual Media." However, since their inclusion in the category, no game scores have actually garnered a nomination.

According to NARAS, this is due largely to the scarcity of game scores submitted for consideration. The important thing to understand about the Grammy Awards is that they celebrate commercially released music, meaning albums and singles. The simple fact that a game's score

exists as part of its source game doesn't make it eligible for Grammy consideration, nor does freely streaming a game's soundtrack off a developer's web site, or making MP3s available for download to the game's community. NARAS has strict distribution guidelines that require eligible music be available from brick and mortar stores and/or via digital distribution from a third-party site like iTunes or Amazon.

There are very few commercially released game soundtracks each year. On Amazon, film and television scores have their own separate sales categories. Films are even broken down further into film scores and film soundtracks. Game scores, though, share a category with anime soundtracks, and Amazon only lists 14 video game soundtracks as having been released in 2010. For the sake of comparison, Amazon lists over 100 television soundtracks as having been released in the first quarter of 2011 alone.

GRAMMYS WITHIN GRASP

>> In April of this year, NARAS announced a handful of official changes to the line-up of awards for the 2012 Grammys. Among these changes includes specifically calling out video games as eligible among four different award categories:

Best Compilation Soundtrack for Visual Media Motion, Television, Video Game Music, or Other Visual Media

Best Score Soundtrack for Visual Media Motion, Television, Video Game Music, or Other Visual Media

Best Song Written for Visual Media Motion, Television, Video Game Music, or Other Visual Media

In order for a game to be nominated for these awards, a number of things must first take place. First of all, the game's score must get an official commercial release. Once the score is released, it is eligible to become what NARAS refers to as an "entry." Entries are the name given to all of the recordings submitted to NARAS for consideration of Grammy nominations. However, entries can only be submitted by NARAS members and registered media companies, meaning record labels and/or independent distributors. Once the entries have been received, NARAS convenes screening sessions where entries are reviewed to ensure that they meet the submission requirements and have been submitted to the correct categories.

Once reviewed, the list of entries is compiled and sent out as first-round ballots to NARAS members. Within the past year, NARAS has started to make streaming of entries available from a secure section of their web site so that voting members can listen to the bulk of category submissions from their PCs. Members then vote, return their ballots, and the tabulated results produce the final list of Grammy nominations. At last, final nomination ballots are sent out to voting NARAS members.

While the specific acknowledgement of game scores across these four Grammy categories is a big step forward for game music, it's still not the dedicated "Best Video Game Music" category that some game composers are petitioning NARAS to include. It's clear, however, that for games to demand the same level of validation from the Grammy Awards as film and television, game composers and publishers are going to have to significantly increase the number of game soundtracks made available each year to the general public.

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.***

ILLUSTRATION BY KELSEY KRAUS

Big Blue Bubble
220 Dundas Street, Suite 900
London, ON N6A 1H3
519.649.0071
contact@bigbluebubble.com
www.bigbluebubble.com

services + capabilities

- Cross platform Game Design and Development
- Publishing
- Licensing

Supported Platforms

- Smartphone: iOS, Android, Windows Phone 7, Symbian
- Console: Nintendo DS, 3DS and Wii, XBOX 360
- Virtual console: XBLA, WiiWare, DSiWare, PSN
- Online: Facebook, Flash, Silverlight, HTML5
- Windows PC, Mac OSX

////////////// Big Blue Bubble was founded by game industry veterans in February 2004 with one driving goal in mind – to become the leader in the video game industry. Its experienced team is dedicated to making the best game possible by combining innovative design with stunning graphics and leading-edge technology. The company philosophy is based on these simple guiding principles – to design addictive pick-up-and-pay multi-platform games, to develop them in the highest quality possible and to deliver them on time and within budget.

Big Blue Bubble has garnered numerous industry awards and accolades, but the true testament of their accomplishments can be seen through the success of their games. To date, several of their games have surpassed 1 million copies sold, and their own franchise, HOME SWEET HOME, was their second title to reach over 10 million customers. Whether it be their own ip or work-for-hire, they consistently obtain success as their latest two titles can attest to. In early 2011, their own BURN THE ROPE surpassed millions of apps to reach the top of the iPhone charts. A mere three months later, LEGO NINJAGO skyrocketed up to the top as well.

Their continued success can be attributed to their revolutionary research and development efforts in establishing proprietary tools and technologies which support their focus on cross-platform development. These tools have enhanced their ability to streamline developmental processes and leverage the maximum potential of each of their titles. Driving these multi-platform technologies forward and binding them together is a result of several key philosophical goals:

RAPID PROTOTYPING To support agile methodologies used within Big Blue Bubble, rapid prototyping is key to enabling both faster development of titles and quick adaptation to change.

DATA-DRIVEN DESIGN Artists and designers are empowered and have their efficiency greatly enhanced by allowing them to directly translate content and assets into gameplay with little or no programmer intervention or support required.

EASE OF USE For tools and engines to effectively optimize productivity, they must be easy to use by all members of the development team.

EXTENSIBILITY Recognizing the uniqueness of each operating system or platform, it is critical to allow new or custom functionality to be added to tools and engines with minimal effort.

With over 80 games in its portfolio and games distributed in over 100 countries, Big Blue Bubble has led the way in becoming a cornerstone of the gaming world.

CAPCOM®

////////////// **Capcom Game Studio Vancouver is an innovative, technically ambitious and rapidly expanding game development studio, focusing on third-person, action / adventure sand-box games.**

Capcom Game Studio Vancouver started humbly as Blue Castle Games in 2005 with only a dozen committed employees. But our vision and energy resulted in rapid growth, our own powerful game engine and a remarkable six games produced within our first five years, across six platforms. We joined the Capcom family in 2010, just as we released the eagerly anticipated DEAD RISING 2.

OUR VALUES We believe that happy people make great games. Even as we expand, we retain the feeling of family that has made our studio so successful. We believe that all voices deserve to be heard, and that an open-door, collaborative approach—as well as a strong spirit of play—enables all of our employees to feel the ownership and passion that feed great games.

OUR TECHNOLOGY At Capcom Game Studio Vancouver, we believe that custom written tools, pipelines and engine give the most flexibility and best results (and are way more fun to work on). Our engineers find innovative ways to improve the game development experience for everyone in the studio. Forge, our multiplatform in-house game engine, has been designed to push the limits of modern consoles while maintaining the flexibility necessary to allow our designers and content creators to let their imaginations run wild.

OUR DESIGN TEAM Imagination is the foundation of all great games. Our artists, designers, audio team and animators work together, in-house, to form their own strong creative vision and then actively shepherd that vision into being, from start to finish.

OUR GAMES

Dead Rising 2: Innovative weapons, a huge world to play in, and the most zombies on screen in any game, ever. Voted Gamespot Funniest Game of the Year. Rated 9.5 out of 10 by Game Informer.

CASE ZERO: This Xbox exclusive was the fastest-selling game on Xbox LIVE. A groundbreaking approach to DLC, CASE ZERO changed the way the industry looked at pre-release DLC.

CASE WEST: Novel co-op game play, increased AI, and a satisfying global conspiracy. The heroes of DEAD RISING and DEAD RISING 2 team up for some wisecracking co-op zombie slaughter.

DEAD RISING 2: OFF THE RECORD: This innovative alternate take on the infamous Fortune City outbreak brings back the beloved wiseguy hero of the original DEAD RISING, Frank West.

But our future holds much more than zombies, as we are actively working on other innovative projects, and developing original IP in conjunction with Capcom.

OUR LOCATION Capcom Game Studio Vancouver is located in beautiful Vancouver, British Columbia, Canada–one of the only places on the earth you could ski, golf and go sailing all on the same day. Vancouver has been repeatedly ranked as one of the world's most livable cities by *The Economist* and Mercer's Quality of Living survey, while *Conde Nast* travel magazine rated Vancouver the top destination city in the Americas.

Capcom Game Studio Vancouver, Inc.
4401 Still Creek Drive, Suite 200
Burnaby, BC V5C 6G9
Phone: 604.299.5626
Fax: 604.299.5653
cv.info@capcom.com

www.capcomvancouver.com

Capcom Game Studio Vancouver

- Founded in July 2005
- Joined Capcom in October 2010
- 185 employees and growing
- Our own proprietary multiplatform game engine
- Multiple projects in development
- Multiple IP
- Full service studio
  - In house concept team
  - In house audio team
  - In house cinematics team
  - In house development support
  - Everything a studio requires to make great games

FOCUS ON

InnoGames GmbH
Harburger Schlossstr. 28
21079 Hamburg
Phone: +49 40 788 9335 0
info@innogames.de
www.innogames.de

InnoGames

Who we are

With about 70 million players from nearly 200 countries, InnoGames is one of the leading worldwide developers and publishers of Massive Multiplayer Online Games. Our products (TRIBAL WARS, THE WEST, GREPOLIS, WESTWARS, SEVEN LANDS...) are available in more than 30 languages. Our headquarters is located in Hamburg, northern Germany. We also have a subsidiary company in Seoul, Korea.

The Business Principle

Our users have the option of playing InnoGames products completely free of charge and without restrictions for as long as they want. We also offer players the possibility of enjoying added benefits in the game by signing up for fee-based premium accounts. At the same time, InnoGames places a high priority on providing entertaining gameplay even without a premium membership.

This principle has great advantages to the user. In contrast to traditional PC games, the product features here are already well-known before the player makes the decision to pay for added benefits—or not. The specific advantages offered by the premium account are also transparent. "What you see is what is what you get" is the underlying principle.

In our games, we focus on a high long term motivation. Our game TRIBAL WARS is online for more than eight years now and it daily attracts some million users up to now.

Partner with us

We have a good working network with trustworthy partners on all important game markets. In Asia, we have founded a subsidiary company, InnoGames Korea Ltd. This makes it much easier for us to publish our games in this area and to find suitable Asian games, which we publish in western markets. As we are rapidly growing, we are looking for promising new partnerships. You have great games, which you would like to publish in Europe, North/South America or Asia? You have a big media competence and would like to cooperate with us? Or you are highly talented and want to boost our human resources? No matter what – we are eagerly waiting for you. Just contact us!

We are looking for

Business partners

- High quality online games (Client and Browser)
- Effective media cooperations
- Partners for our in-house affiliate program

Talents to boost our human resources

- Frontend-Developer
- PHP-Developer Games
- Ruby on Rails Developer
- Producer
- Gamedesigner

gamescom

gamescom 2011: The entire gaming world in one place

The concept of the world's largest trade fair and event highlight for interactive games and entertainment is unique: it networks the entire value-added chain, from development and publishing to retail and the consumer. As the largest event of its kind in the world and the leading trade fair, it provides discussion platforms on all levels. It covers the entire spectrum of the international gaming scene:

+ PC Games
+ Online Games
+ Browser Games
+ Video Games
+ Mobile Games
+ Gaming Hardware

The concept provides individual platforms for all target groups:

ENTERTAINMENT AREA For passionate gamers (public visitors)
BUSINESS AREA Exclusively for exhibitors, trade visitors and journalists
GDC EUROPE Largest European developer conference

gamescom started with a new key visual into the year 2011. The trade fair and event highlight for interactive games and entertainment demonstrates with its leitmotif what the games world can expect in Cologne from 17th to 21st August 2011: The international games community—developers, providers, trade visitors, media representatives, retailers and thousands of gamers—meets at gamescom 2011 in order to experience together spectacular innovations and to celebrate the games and entertainment event of the year.

Koelnmesse and its partners, headed by the BIU (the German Trade Association of Interactive Entertainment Software), are already working flat out to further develop gamescom as Europe's central business and entertainment platform. Thus, the event is not only accompanied by a new advertisement campaign; the gamescom awards are also further developed due to the great popularity and, for the first time, be awarded in the framework of an opening event in the evening. The BIU also expects exciting novelties and innovations at gamescom when it comes to hardware and software innovations.

We look forward to welcoming you to gamescom 2011!

Koelnmesse Inc.
8700 West Bryn Mawr Avenue
Suite 640 North, Chicago, Illinois,
60631
Tel. +1 773 3269920
Fax +1 773 7140063
info@koelnmessenafta.com

gamescom at a glance

+ business area, halls 4/5
+ entertainment area (halls 6-9) with extra level (hall 10)
+ Presentations of news and innovations of the entire industry
+ gamescom award
+ Games Developers Conference
+ gamescom festival, City of Cologne

gamescom 2010 was a complete success:

+ 505 exhibitors from 33 countries
+ 254,356 visitors in total
+ 235,413 public visitors
+ 18,943 trade visitors from 71 countries
+ 4,400 journalists from 49 countries
+ More than 100,000 additional visitors at City-Festival

FOCUS ON

Seapine Software
5412 Courseview Drive, Suite 200
Mason, OH 45040
Phone: 513.754.1655
Fax: 513.754.1660
info@seapine.com
Twitter: @Seapine
Facebook: https://www.facebook.com/SeapineSoftware

www.seapine.com

Seapine Customers

Seapine's game industry customers include:

+ 2K Games
+ Atari
+ Cheyenne Mountain Entertainment
+ Epic Games
+ Microsoft Game Studios
+ NCSoft Corporation
+ Red Storm Interactive
+ SEGA
+ Turbine
+ Video Gaming Technology (VGT)

Seapine Software™

Seapine Software *Ensuring Quality in Every Game*

Even as video games become more sophisticated and complex to develop, gamers continue to demand a high level of quality in the games they buy and play. Just one bug in a game can result in dismal earnings and damage to a company's reputation—as well as to the chance for future success.

At the same time, teams are under increasing pressure to develop these complex games, while meeting aggressive schedules and satisfying productivity and quality demands.

Over the past 15 years, Seapine Software has been making it easier for top game studios to release award-winning, high-quality games. Leading game development companies—including Atari, Epic Games, and 2K Games—rely on Seapine's ALM product suite to help deliver bug-free games on time and on budget.

Tracking designs, art, source code, scripts, issues, and bugs is a necessity throughout the entire game development lifecycle. Seapine ALM provides a solid foundation for managing all of a game's digital assets.

Seapine's comprehensive and flexible solutions offer:

+ Requirements management, impact analysis, and traceability
+ Centralized digital asset storage
+ Real-time, secure access to bug databases and source repositories
+ Scalability to manage dozens of game titles
+ Team-based collaboration and communication
+ Role-based security
+ Quality metrics and reporting for management

The cost of a game's development often exceeds 10 million dollars, so return on investment is critical. Teams that implement proper asset management benefit from shorter release cycles, increased productivity, and above all, higher quality.

The key to delivering quality games is a comprehensive approach to finding and fixing the defects that will inevitably arise in the coding process. And for that, you can rely on Seapine Software.

Seapine ALM Suite

+ TestTrack RM – Requirements Management
+ TestTrack TCM – Test Management
+ Surround SCM – Configuration and Change Management
+ QA Wizard Pro – Automated Functional and Load Testing
+ Seapine ALM RP – Cross-project and Cross-application Reporting

Seapine Agile Services

Seapine Consulting Services

"TestTrack saved time because it was easy to implement, but the real benefit has been the large ongoing ROI & Instead of changing our process to fit a tool, we were able to easily adapt TestTrack Pro to our way of doing business."

– Doug Doine, Director of Quality Assurance, ***Cheyenne Mountain Entertainment***

"TestTrack Pro is a winner for Epic Games. Developer productivity has increased markedly as crucial information is now located in one place. Since bug reports are accessed more easily, problems are fixed more quickly. TestTrack Pro fits into the organization so seamlessly that we rely on it almost as much as email—and that's saying a lot."

– Jeff Morris, Product Manager, ***Epic Games***

SANTA MONICA studio

Background

Santa Monica Studio was established in 1999, and is focused on creating the most immersive, enjoyable, compelling and playable experiences for the PlayStation® consumer. The studio is credited with perfecting the action/adventure genre, with the success of GOD OF WAR® and its sequels. The GOD OF WAR® series has won hundreds of awards around the world including: ten awards and thirteen nominations from the AIAS (The Academy of Interactive Arts & Sciences), and three awards and seven nominations at the BAFTAs (The British Academy of Film and Television Arts).

The studio pioneered the catalog for the PlayStation®Network by publishing many award-winning original games including FLOW, FLOWER, WARHAWK, BLASTFACTOR, EVERYDAY SHOOTER, FAT PRINCESS, and the PIXELJUNK series. Their development relationship with thatgamecompany has served as an industry model for incubating student teams to commercial success. The studio also collaborated in the development of the TWISTED METAL car combat games. Santa Monica Studio originated the HD (High Definition) remastering of PlayStation 2 classics by porting GOD OF WAR® and GOD OF WAR 2® to 1080p on the PlayStation 3.

Location

Santa Monica Studio is nestled in one of the most creative and inspiring locations in the U.S.A., two miles from the sunny beaches of Southern California and ten miles from the nightlife and entertainment of Beverly Hills and Hollywood. The studio is home to over 200 talented developers who work in an open and airy environment, with high ceilings and ample light. There is an outdoor patio area for gatherings and multiple lounge areas offering free snacks and beverages. Employees have access to state-of-the-art equipment and software, including a five station user-testing lab and an extensive games reference library with over 2,000 titles. It also has six custom sound design pods housing the latest in audio equipment.

Santa Monica Studio is a creative and integrated games production studio, employing the industry's most talented and dedicated professionals, designing, developing and delivering the highest quality AAA PlayStation® products. The studio inspires employees to do their best work in an environment of respect, collaboration, and support.

SEND RESUMES TO:
smstudio@playstation.sony.com

SONY COMPUTER ENTERTAINMENT ®

Santa Monica Studio

AVAILABLE POSITIONS

Fulltime

+ Sr. Combat Designer
+ Sr. Level Designer
+ Sr. Gameplay Programmer
+ Sr. Online Programmer
+ Sr. Tools Programmer
+ Sr. Graphics Programmer
+ Sr. Animator
+ Sr. Environment Artist

Project Duration Contracts

+ Camera Designer
+ Technical Designer
+ Associate Audio Implementer
+ Concept Artist
+ SFX Artist
+ Technical Artist
+ UI Graphic Designer/Artist
+ UI Implementer
+ Associate Producer
+ QA Testers

who went where

Will Kassoy, a 13-year veteran Activision executive, has joined social entertainment and media company Jirbo, Inc., where he will oversee the firm's app gaming division and other operations as CEO.

Sony Computer Entertainment UK has announced that managing director Ray Maguire has parted ways with the company after 17 years.

DigiBC, the Digital Media and Wireless Association of British Columbia, has appointed former VP of Disney Interactive Studios and co-founder of Propaganda Games Howard Donaldson as its new president.

Five months after its former chief executive resigned, social game developer RockYou (ZOO WORLD) has announced that its board of directors promoted Lisa Marino, previously chief operating officer, as its new CEO.

new studios

Former LucasArts and EA veterans have founded the Bay Area-based November Software, which aims to bring streaming 3D content to the web and mobile platforms.

Several former employees from LAIR and STAR WARS: ROGUE SQUADRON developer Factor 5 have formed TouchFactor, a new studio dedicated to building social games, located in the San Francisco Bay Area.

Seattle-based BEJEWELED house PopCap Games announced the new experimental games label and design center 4th & Battery, which the company says will create "smaller, simpler and sometimes edgier" games.

BEJEWELED.

No Paranoid Android

CHRIS PRUETT LEAVES GOOGLE TO GO INDIE

A few years ago, Chris Pruett stopped being a traditional game developer, and moved over to the developer advocacy side at Google. Many would find Google the ultimate destination, but Pruett wanted to get back to development again. To that end, he's formed the indie team Robot Invader, along with Casey Richardson. The company is focused on console-style games for mobile platforms.

BRANDON SHEFFIELD: What made you decide to leave Google and go full-time as a developer?

CHRIS PRUETT: I worked in the game industry for a bunch of years before I joined Google, and the itch to create games full time again was just too strong to ignore. I also felt that I could safely leave Android in the hands of my fellow game industry advocates; the platform is now quite unstoppable.

BS: And why indie, versus another large company?

CP: With a couple of exceptions, large game companies are not able to make the games I want to make. They move too slowly, have too much overhead, and their hands are too often tied by fear of failure. I think the console game industry has gotten very good at making high-tech, highly polished versions of the same five games over and over again (there are certainly exceptions, just not very many). The indie scene, on the other hand, is all about experimentation, but often at the cost of gameplay depth. I want to be able to experiment with all aspects of a game without throwing away the thirty years of design knowledge the industry has developed. I don't think many large game companies are prepared to work that way.

BS: Many have tried to bring console-style design to mobile platforms—how will you be different?

CP: Many have tried to simply port console games to mobile platforms. The audience and interface is so different that it rarely works. A few developers have tried to bring the "console experience" (which usually means high-end graphics) to mobile with some success. That approach seems counter productive to me; it's clear that ultra high-end graphics technology is not a key selling point for mobile games, and the cost to develop that kind of content is so great that it limits what the developer is able to do in other areas.

Our angle is different. We are building mobile games, which means they must be designed for the mobile user—touch interface, on-the-go, easy-to-learn mechanics. But we want to infuse those games with design philosophy from the console space. There are many ways to do this, but the key goal is to provide a layered, complex experience to the user. In the Android game I built at Google, REPLICA ISLAND, I tried to accomplish this with story and characters. In another game, it might be dynamic difficulty adjustment and achievements. We want to pair simple mechanics with content along other axes to produce something more challenging to the player than your average color match or physics sandbox game. This approach is thoroughly informed by console precedent.

BS: Given your interest in horror games, do you think a horror experience could work on mobile, with the frequent distractions and tendency for players to mute sound?

CP: Horror can absolutely work on a handheld device. There are already mobile horror games that prove it. The Japanese NAMELESS GAME series for DS, for example, succeeds in being scary on a small screen despite a lot of gameplay problems and ultra basic graphics. That said, sound is an important component; NAMELESS GAME aggressively advises the user to play with headphones, which is smart. Games like PAPA SANGRE make me think that sound-based horror can work on phones as well.

BS: Do you think people actually want depth out of their mobile games? Are these devices becoming more than just brief distractions?

CP: Yes, I think people do want additional depth. Mobile games live and die by their playability, and users won't stick around to see the additional complexity if the game isn't immediately fun. But once they are hooked, I think that content along other axes will give players a reason to continue above and beyond the basic mechanics. It's important not to shove this additional depth down the player's throat, but making it available for those who are interested increases the value of the play experience. REPLICA ISLAND provides a whole lot of narrative to those players who are interested in it, but the game can be completed without ever reading a line of dialog. The idea is to provide depth for those who want it, but to do so in such a way that those who do not are not at a disadvantage.

SHOOT INTO THE FUTURE...

LEAD MULTIPLAYER DESIGNER • MULTIPLAYER DESIGN SCRIPTER
SR. LEVEL DESIGNER • LEVEL BUILDERS • CONCEPT ARTIST
SR. ENVIRONMENT ARTIST • SR. TECHNICAL ARTIST
LIGHTING ARTIST • PC PLATFORM LEAD ENGINEER

APPLY @ HTTP://WWW.NEVERSOFT.COM/SITE/HIRING.HTML

TINY AND BIG

www.tinyandbig.com

IGF STUDENT FINALIST TINY AND BIG COMBINES 3D PLATFORMING WITH PHYSICS-BASED SANDBOX DESTRUCTION, ALLOWING PLAYERS TO SLICE AND RE-SHAPE THE LANDSCAPE USING TOOLS SUCH AS A GRAPPLING HOOK AND A HIGH-POWERED LASER. THE KASSEL, GERMANY-BASED TEAM RELEASED A PROTOTYPE FOR THE GAME IN 2010, AND IS NOW WORKING ON A NEW, FULL-FEATURED VERSION, TITLED TINY AND BIG: GRANDPA'S LEFTOVERS.

***TOM CURTIS:* Tell me about the team's approach to design for TINY AND BIG. What process did you use? Prototyping? Sketches?**

FLORIAN GROLIG AND SEBASTIAN STAMM (game designer and artist): Designing the game mechanics was an iterative process; it usually started with an idea roughly sketched on paper to illustrate it to the team. Based on that sketch, we created a prototype to check whether the basic idea actually worked. Keeping the concept consistent turned out to be the major challenge. Many features we were excited about or that were requested by people who played the demo didn't make it into the game because they would have weakened the core mechanic.

To catch the mood and structure of our levels, we created concept artwork of the scenery and the key elements of each level. We simultaneously started to create sketches of the challenges and puzzles. During the whole process of making the game, we allowed ourselves a lot of room for testing and time to rearrange the environment and puzzles to eventually get to a point where every level had its own theme and feeling.

***TC:* Why did you choose to develop your own engine rather than use an existing one like Torque or Unity?**

JOHANNES SPOHR (engine lead): The three programmers among us have been enthusiastic about game development since back when we started our CS degrees, or even earlier. The Scape Engine was born out of a few projects in computer graphics classes, which we attended mainly because

of their relevance to 3D games. With time, more and more code was added as more ambitious projects were pursued. A few successful courses and some unfinished games later, we had a nice game programming toolset which allowed us to rapidly develop 3D games. The driving force behind our DIY effort was very basic, yet irresistible: It was fun!

***TC:* Did the slicing mechanic present any particular design challenges? It seems to allow a lot of room for players to accidentally destroy their path through the environment.**

SS: You bet it does! One of the biggest challenges we've encountered was the almost unlimited freedom given to the player. As nearly everything in the game can be cut and modified, every piece of the environment is a potential way to stray from the level's predefined path. On one hand, we wanted to maintain a certain linearity in the game to prevent the player from getting lost or overwhelmed by possibilities. On the other hand, we didn't want to destroy the whole experience by taking away too many options, so we were pretty careful in creating boundaries. We limited the range of the laser and other tools to keep the player's focus on the area he is standing in and to prevent him from accidentally cutting apart pieces in the far, far distance. This makes sense gameplay-wise and gives us the opportunity to make use of it in level design.

Furthermore, we use objects that are simply too big to be cut as massive level borders, or just put Tiny on platforms high in the air. One thing the player has to become aware of as the game proceeds are his powerful abilities. They give him a dozen ways to solve puzzles, but can also put him in a self-inflicted dead end. It's a peril of sandbox gameplay.

***TC:* The game certainly seems to revel in its comic-book aesthetics. What were your sources of inspiration for the art style?**

SS: Ever since I began playing video games, I liked games like DAY OF THE TENTACLE, SAM AND MAX, WOODRUFF, HOT WHEELS (C64), and EARTHWORM JIM. They all had a certain style that made them feel special and self-contained. What made them different from a lot of games was their own visual language. When developing the visual style and characters for our game, I always tried to bear in mind what made me excited about those games.

Our general approach was to get as close as possible to the original, analog-drawn artwork. We didn't want to produce a mash-up of different styles and end up with trade-offs and a common visual appearance. This led to the pop-up words for sound effects, the flat and painted shading, the speech bubbles, and the crosshatched shadows on every object.

***TC:* How did you all come to work together on this project?**

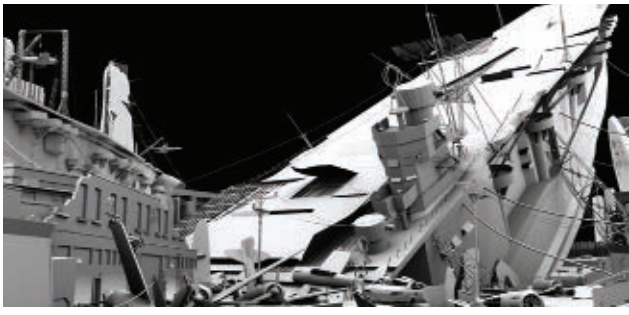
CHRISTIAN NIEMAND (engine programmer): In early 2004, Johannes and I met at the University of Kassel. In 2008, Sebastian Schulz joined our team, and together we took a shot at creating a game. Besides coding, we had to build content like models, textures, sounds, and music ourselves, or grab it from the internet. This resulted in a couple of prototypes with programmer art that were miles away from a seamlessly designed game. So we decided to look for talented designers at Kassel's School of Arts and Design.

Florian Grolig, Sebastian Stamm, and Philip Gutjahr were hooked after our first meeting. They brought amazing skills to our team, skills which we couldn't have dreamed of covering before. During the year that followed, we ventured off into the art school's dark dungeons to work on the TINY AND BIG prototype. The feedback we received after its release largely factored into our motivation to make a full game with the same concept. Our only problem was that we had less than no money at all to do it. So we concocted two plans: One was to develop TINY AND BIG in our spare time while still having a day job. The other was to apply for a grant for our enterprise. Writing a business plan took us eight weeks, but in the end, it was worth it; the grant was approved. We now had money to secure our existence for 12 months. While we're still working in yet another dark dungeon, this one is our own office at the university. Our goal is to release TINY AND BIG: GRANDPA'S LEFTOVERS this year through our own company: Black Pants Game Studio.

—Tom Curtis

ACADEMY of ART UNIVERSITY®

FOUNDED IN SAN FRANCISCO 1929 BY ARTISTS FOR ARTISTS



TAKE CLASSES ONLINE OR IN SAN FRANCISCO

Advertising
Animation & Visual Effects
Architecture*
Art Education
Fashion
Fine Art
Game Design
Graphic Design
Illustration
Industrial Design
Interior Architecture & Design
Landscape Architecture*
Motion Pictures & Television
Multimedia Communications
Music Production & Sound Design for Visual Media
Photography
Web Design & New Media

ENROLL NOW

EARN
YOUR AA, BA, BFA, MA, MFA OR M-ARCH ACCREDITED DEGREE

ENGAGE
IN CONTINUING ART EDUCATION COURSES

EXPLORE
PRE-COLLEGE SCHOLARSHIP PROGRAMS

WWW.ACADEMYART.EDU
800.544.2787 (U.S. Only) or 415.274.2200

79 NEW MONTGOMERY ST, SAN FRANCISCO, CA 94105

Accredited member WASC, NASAD, CIDA (BFA-IAD), NAAB (M-ARCH)

**Landscape Architecture and Architecture BFA degree programs not currently available online.*

Photo credit: Sungho Lee, Texture & Light Game Design Course

GDC EUROPE 2011 REVEALS SUMMIT ADVISORY BOARDS, MAIN BOARD ADDITIONS

\\\ The organizers of this August's Game Developers Conference Europe 2011 have revealed the line-up for the brand-new Summit advisory boards, as well as an addition to the main GDC Europe board.

The industry luminaries joining the advisory boards represent companies including Playfish, Sony Online Entertainment, THQ, Copenhagen Game Collective, DDM, and more, and will be responsible for helping craft the content of the event, which now includes four major Summits and five Main Conference tracks.

For the first time, the 2011 GDC Europe advisory boards are now split into Summit-specific boards and a Main Conference board. Along with these changes, organizers made the following announcements:

>> Johan Sjoberg, the Swedish-based lead agent at game representation agency DDM, is joining GDC Europe's main advisory board. Sjoberg works closely with the agency's game developer clients, including firms like Ninja Theory and Vatra Games, on business development and corporate strategy, and his connections within the European game industry should prove invaluable to the event.

>> The inaugural Social Games Summit has announced board members from the forefront of the European social games scene. These members include UK-based Playfish/Electronic Arts VP and co-founder Kristian Segerstrale, as well as the Finnish founder of Rocket Pack, Jiri Kupiainen whose social game company was recently acquired by Disney.

>> The first-ever Smartphone & Tablet Summit at GDC Europe has added as a board member Germany-based Fishlabs CEO and co-founder Michael Schade, whose GALAXY ON FIRE series has found success on multiple mobile platforms. Other confirmed advisors include Secret Exit's Jetro Lauha—the Finnish firm's ZEN BOUND 2 and STAIR DISMOUNT have seen multiple millions of downloads.

>> The Independent Games Summit at GDC now includes advisors such as Copenhagen Game Collective co-founder Lau Korsgaard, whose Danish collective is behind innovative indie party game B.U.T.T.O.N., recently showcased at this year's Independent Games Festival at GDC 2011. Also added is Remote Control Productions' Hendrik Lesser, whose firm works with notable European independent creators such as Brightside Games (Zeit2).

>> Finally, the Community Management Summit has added to its core co-organizer, German-based Two Pi Team CEO Thomas Lagemann, with the appointment of Linda Carlson, director of global community relations for Sony Online Entertainment, as well as THQ's director of community management Chris Mancil, a veteran of studios including Trion Worlds and Vivendi Games.

In addition, returning advisors to the GDC Europe main board include Zynga's Bob Bates, pioneering game designer Don Daglow, Remedy Entertainment's Matias Myllyrinne, Avni Yerli of Crytek, Strategic Alternatives' Robert Wallace, Alexander Fernandez of Streamline Studios, International Digital Entertainment Agency's Sean Kauppinen, Harald Riegler of Sproing, and Frank Sliwka, Vice President European Business Development and Event Director GDC Europe.

GDC Europe will take place August 15–17, 2011 at the Congress-Centrum Ost Koelnmesse in Cologne, Germany.

ADVERTISER INDEX

gd Game Developer (ISSN 1073-922X) is published monthly by United Business Media LLC, 303 Second Street, Suite 900 South, South Tower, 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 *gd Game Developer*, P.O. Box 1274, Skokie, IL 60076-8274. 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 *gd Game Developer* on any correspondence.

ASK A VINYL CHARACTER FIGURE

MORE WISDOM FROM GAME DEVELOPMENT'S UNSUNG HEROES

IN A CONTINUING SERIES OF interviews with people or things key to game development, *Game Developer* presents this discussion with a video game character figurine that we found sitting on a shelf in an artist's office.

Tell us about yourself.
Well, where should I start? I was cast in polyvinyl chloride a few years back and distributed as a bonus inside the Collector's Edition of the hit game FUTURO-BATTLE 2: HUNTERS RISING. I'm sure you've seen me around, even if you don't know recognize me: I'm the girl in the game who whispers in the player's ear and tells him what to do. Like, "That door should be unlocked now! You can go through it! Go on; go through that door you just unlocked!!" ... that sort of thing.

Since the game came out a couple years ago, I've been taken out of that stuffy Collector's Edition box, thank goodness, and have been chilling on the "action figure shelf" of someone's office—a senior 3D artist at another well-respected video game studio, in fact.

So, you were a Collector's Edition bonus. Care to share your thoughts on what makes a good–
A good Collector's Edition? I've seen some really rough ones out there, you know? Like, wow, a soundtrack CD. Nothing like some unremarkable, generic orchestral bombast encoded on a dead medium to increase your asking price by 10 bucks or more. Am I right? Okay, what about that limited edition "poster"? Seriously, what are you going to do with that? Try to flatten out those creases and frame it? Don't make me laugh.

You sound pretty cynical about–
About Collector's Edition pack-in items? Look, I know where I came from—a factory a couple dozen miles to the west of Shanghai. If you saw the manufacturing process for bonus items like me yourself, you'd be just as amazed that people are willing to pay an extra 40 dollars to have them. To be fair, without a Collector's Edition, I might not be alive today. We are our own worst critics, I suppose.

So, if you were to design a Collector's Edition of a major game release for this fall, what would you include?
A big signboard that says "get a life"—no, just kidding. I think people respond to items that evoke the world of the game, like those old Infocom "feelies."

Wait, you know about feelies and Infocom games? Aren't you just a couple years old?
I sit next to a vintage King Gidorah figure, who got here via eBay a few months ago. He's been around various nerd havens, and has seen a lot in his time.

Speaking of which, what's your day-to-day like as an office figurine in the game industry?
It's not a bad life. I definitely get an interesting perspective of how games are made these days. The only thing I'd complain about is the other figures up here on this shelf with me; it's pretty crowded with giant robots, anime girls, and superheroes. For a while, I sat next to a $200 Lara Croft cold-cast statue, but I think the artist figured out it was pushing him over the line into that "creepy figure guy" territory. I mean, who owns something like that? After a while he took it down and told people that he sold it, but I'm not sure. I'm willing to wager she keeps him company at home.

You mentioned having an interesting view into game development—what are some of the more memorable things you've seen up there on that shelf?
There was this one time... It was around two in the morning, and nobody had been around for hours. All of a sudden, the senior 3D artist staggered into the office. He was super drunk and sort of fell into his chair like he was going to do some work. He poked at the computer for about 30 seconds—I think he logged in and launched Maya– then crawled under his desk and passed out. He woke up again around five and left before anyone else came in. It was really weird! Oh, and I forgot to mention, he was in his underwear the whole time.

ILLUSTRATION BY JUAN RAMIREZ

That definitely sounds like a view on game development you don't get through the normal channels.
Of course, I see the usual things too. Grown men totally flipping out about how a laser gun looks—that sort of thing.

It Looks like we're almost out of time. Any closing thoughts?
The last thing I'd like to impart to any game developers reading this is: I know you're busy making your thing, and we're just little trophies for you, but don't forget to dust us once in a while. Keep us looking alright. I don't mean to sound, like, threatening or anything, but we remember those who mistreat us.

Oh, yes. We remember.

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