UBM

SPECIAL FEATURE: TOP DEVELOPERS OF 2009

VOL 17 NO 6 JUNE/JULY 2010

gamedeveloper

THE LEADING GAME INDUSTRY MAGAZINE

DARKSIDERS

POSTMORTEM

TOOLS FOR HUMANS

CREATING A FRIENDLY STATE-DRIVEN GAMEPLAY SYSTEM

EYE OF THE BEHOLDER

APPLY THESE CLASSIC ART TECHNIQUES FOR BETTER DESIGN AND PACING

Image of Unreal Tournament® 3 courtesy of Epic Games

3Di is here.

Create animated 3D interfaces using Scaleform GFx 3.2

www.scaleform.com/3di

Scaleform®

gamedeveloper

CONTENTS.0610
VOLUME 17 NUMBER 6

POSTMORTEM

FEATURES

DEPARTMENTS

COVER ART: DARKSIDERS ART TEAM UNDER JOE MADUREIRA

PLAYERS VS. USERS

SOCIAL AND "TRADITIONAL" GAME DEVELOPERS BUTT HEADS

THE MAIN BUZZ AT THIS YEAR'S GDC CANADA surrounded social games. That's really not too surprising when you look at where all the venture capital is going these days, and social games like FARMVILLE and Pet SOCIETY have done a lot to broaden the market, bringing in groups that never thought they would play video games. I'm always in favor of broadening the market, but I worry that some of these companies are more interested in players as revenue streams than as people to entertain.

Intent is very important to me in any media product—just as important as the execution. The much-discussed "No Russian" scene in MODERN WARFARE 2, during which the player can gun down innocent people in an airport, changes a lot depending on the intent of the developer. If the intent is for players to have fun slaughtering innocents, the scene becomes pretty negative on a meta level. If the intent is for players to uncomfortable and question their actions as a participant in electronic slaughter, that may be more progressive.

CAN MONEY BUY HAPPINESS?

>> Games are a business, but ultimately they must entertain. This is rather universal. What I am concerned by is the disconnect I'm seeing between the potential of this broadened market and the attitude of the major principals. There is a lot of talk about money in social games—one might argue too much. Nintendo makes money hand over fist through market expansion, but the company talks about players, and about bringing the fun of games to new people. It may simply be rhetoric or a question of semantics, but most social game companies are talking money first, and player enjoyment second.

As an example, in Zynga's early days, as CEO Mark Pincus discussed in a well-publicized lecture, "... we gave our users poker chips if they downloaded this Zwinky toolbar which was like, I don't know, I downloaded it once and couldn't get rid of it. We did anything possible just to get revenues so that we could grow and be a real business."

Pincus refocused this in a subsequent blog post, saying, "The primary reason I pointed to [that] is that entrepreneurs should push to control their own destinies, and being profitable was the best path."

He recanted to some degree, but that drive for money above all else is what bothers me, and I don't think it's viewed as a bad thing in the space. I want to hear a whole lot more talk about "players" of social games, and a whole lot less about "users." I don't want to single out Zynga here, and in fact the company is among our top 30 developers this month (see Pg. 6), primarily for growing the market and proving a new platform. Everyone from Playfish to SOE knows that online games are a service, and that "average revenue per user" is the most important thing to keeping afloat. But the faster these companies grow, the more I worry they are becoming detached from the idea that the players on the other side of their games are real people, not number generators.

There is a clear gap here between the social and traditional game markets, and I would love to feel like that gap had more to do with market expansion and low barriers to entry, and less to do with money, but every discussion I get into about the subject with social game execs points me to the contrary. The folks I talk to recognize that they are expanding the market, and that low barriers to entry and stickiness are the key, but to them it's the key to more money, not necessarily to user happiness.

It's a complicated issue, because after all, 80 million people aren't playing FARMVILLE because they hate it. The same metrics that executives use to drive revenue are also used to increase ease of use for players. Well, to some degree anyway—it's part user enjoyment, and part revenue generation. And I think that's where my appreciation breaks down. As a player first and a developer second, I want more decisions to be driven by entertainment value and risk, and fewer to be driven expressly by revenue.

US VERSUS THEM

>> Where does the antagonism between traditional and social games come from? I think it might be coming from the intent on both sides. The developers of console games have to think constantly about what might be fun—money comes second, and is the purview of the business guys. The developers of social games must necessarily think about money and fun together, whether they be executive or otherwise. They have access to insanely detailed information about what players do, like, and want to pay for. Why not use it? What would console developers do with the same information? Not ignore is surely. Is it simply a question of information, then?

It may also be a matter of ego. Console developers often view social games as lesser, in spite of how complex they have become (again, just look at the current depth of FARMVILLE), because with accessibility comes simplicity. Social game execs and developers, for all their money, still seem to feel they're the younger sibling of console games, and must rail against them whenever possible.

There is room for all of us in this industry, though the dollars may flow one way or another depending on the climate. It's important to learn what each side has to offer the other—and then eventually there may not be sides at all. But for my part, I want to hear a lot less talk of users, and start hearing a much greater focus on player enjoyment.

—Brandon Sheffield

gamedeveloper
www.gdmag.com

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

SUBSCRIPTION SERVICES

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

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
Jeffrey Fleming | *jfleming@gdmag.com*
ART DIRECTOR
Joseph Mitch | *jmitch@gdmag.com*
CONTRIBUTING EDITORS
Jesse Harlin
Steve Theodore
Daniel Nelson
Soren Johnson
Damion Schubert
ADVISORY BOARD
Hal Barwood **Designer-at-Large**
Mick West **Independent**
Brad Bulkley **Neversoft**
Clinton Keith **Independent**
Brenda Brathwaite **Independent**
Bijan Forutanpour **Sony Online Entertainment**
Mark DeLoura **Google**
Carey Chico **Independent**

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: *tysonassoc@aol.com*
LIST RENTAL Merit Direct LLC t: 914.368.1000

MARKETING

MARKETING SPECIALIST Mellisa Andrade
e: *mandrade@think-services.com*

UBM
WWW.UBM.COM

FIRST QUARTER SALES REVEAL CONSOLE TRENDS

USING EXCLUSIVE NPD DATA, MATT MATTHEWS LOOKS AT THE TOP SELLING U.S. RETAIL GAMES FOR EACH PLATFORM DURING THE FIRST QUARTER OF 2010, FROM MODERN WARFARE 2'S CONTINUED SALES THROUGH JUST DANCE'S SURPRISING WII STRENGTH.

FIRST-PARTY WII DOMINATION

» Nintendo continues to dominate software on its Wii platform with three of its 2009 titles topping the chart. NEW SUPER MARIO BROS. for the Wii launched in November 2009 and sold over 4.2 million units by the end of December, making it the third best-selling title for that year.

Sales so far in 2010 have added approximately 1.7 million more units bringing its lifetime total to just shy of 6 million units as of the end of March.

With April sales, NEW SUPER MARIO BROS. has no doubt already surpassed the 6 million mark on the Wii. We believe the Nintendo DS version of NEW SUPER MARIO BROS. now has sales of around 7 million units, based on available data.

Nintendo's update to its fitness package, WII FIT PLUS, is the second best-selling title on the Wii so far in 2010 with sales of around 1.1 million units. Total sales now reach nearly 4.7 million since it launched in October 2009.

With unit sales over 800,000 units, WII SPORTS RESORT is the number three title on the Wii so far this year. Combined with its total in 2009, after its July launch, WII SPORTS RESORT has life-to-date sales of well over 5 million units.

Sneaking in at the number four position is Ubisoft's hit, JUST DANCE, the only third-party to make this list. Just in 2010 we estimate that it has sold around 650,000 units, and according to Broadpoint AmTech analyst Ben Schachter, it has now reached 1 million units since its November 2009 launch in the U.S.

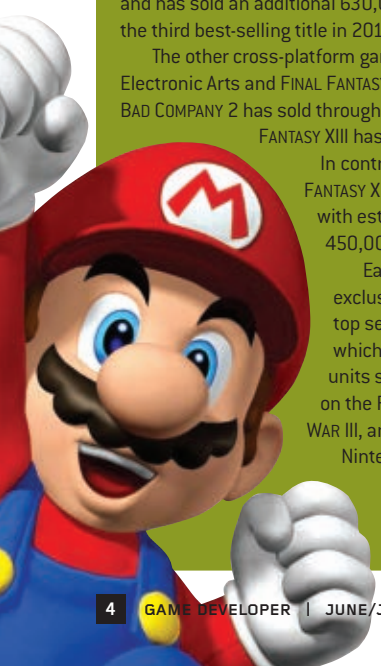
MARIO KART for the Wii is the final title in the top 5 for 2010, and it is one of Nintendo's greenest evergreen titles.

COMBAT GAMES DOMINATE XBOX 360 AND PS3

» Three cross-platform games appear on the top-selling software lists for both Microsoft's Xbox 360 and Sony's PlayStation 3.

The number two game on the Xbox 360 so far in 2010 is just such a title, CALL OF DUTY: MODERN WARFARE 2, published by Activision Blizzard. On that platform we estimate it has already garnered sales of 840,000 units, and has sold an additional 630,000 units on the PlayStation 3, where it is the third best-selling title in 2010.

The other cross-platform games are BATTLEFIELD: BAD COMPANY 2 from Electronic Arts and FINAL FANTASY XIII from Square-Enix. On the Xbox 360 BAD COMPANY 2 has sold through an estimated 830,000 units while FINAL FANTASY XIII has shifted only 500,000.



In contrast, the PlayStation 3 version of FINAL FANTASY XIII outsold BAD COMPANY 2 on that platform with estimated unit sales of 830,000 and 450,000, respectively.

Each platform's chart is headed by a title exclusive to that system. In particular, the top seller on the Xbox 360 was MASS EFFECT 2, which has nearly reached sales of 900,000 units since its launch in January. The top title on the PS3 so far in 2010 is Sony's own GOD OF WAR III, and it is the only million-seller on any non-Nintendo platform this year.

BIOSHOCK 2 appears at the number four spot on the Xbox 360 chart with estimated sales of 600,000 units.

Top 5 Retail Wii Games
First Quarter 2010, U.S. Only

SOURCE: NDP GROUP, M. MATTHEWS

RANK	TITLE	PUBLISHER	EST. UNITS
1	NEW SUPER MARIO BROS. Wii	Nintendo	1,700K
2	WII FIT PLUS	Nintendo	1,100K
3	WII SPORTS RESORT	Nintendo	830K
4	JUST DANCE	Ubisoft	670K
5	MARIO KART w/Wheel	Nintendo	600K

Top 5 Retail XBox 360 Games
First Quarter 2010, U.S. Only

SOURCE: NDP GROUP, M. MATTHEWS

RANK	TITLE	PUBLISHER	EST. UNITS
1	MASS EFFECT 2	EA	1,700K
2	CALL OF DUTY: MODERN WARFARE 2	Activision Blizzard	840K
3	BATTLEFIELD: BAD COMPANY 2	EA	830K
4	BIOSHOCK 2	2K Games	600K
5	FINAL FANTASY XIII	Square Enix	600K

Top 5 Retail PlayStation 3 Games
First Quarter 2010, U.S. Only

SOURCE: NDP GROUP, M. MATTHEWS

RANK	TITLE	PUBLISHER	EST. UNITS
1	GOD OF WAR III	Sony	1,100K
2	FINAL FANTASY XIII	Square Enix	830K
3	CALL OF DUTY: MODERN WARFARE 2	Activision Blizzard	630K
4	BATTLEFIELD: BAD COMPANY 2	EA	450K
5	MAG	Sony	400K

Top 10 Retail Videogames
First Quarter 2010, U.S. Only

SOURCE: NDP GROUP, M. MATTHEWS

RANK	TITLE	PLATFORM	EST. UNITS
1	NEW SUPER MARIO BROS. Wii	Nintendo Wii	1,700K
2	WII FIT PLUS	Nintendo Wii	1,100K
3	GOD OF WAR III	PlayStation 3	1,100K
4	POKÉMON SOULSILVER	Nintendo DS	1,020K
5	MASS EFFECT 2	XBox 360	890K
6	CALL OF DUTY: MODERN WARFARE 2	XBox 360	840K
7	WII SPORTS RESORT	Nintendo Wii	830K
8	FINAL FANTASY XIII	PlayStation 3	830K
9	BATTLEFIELD: BAD COMPANY 2	XBox 360	830K
10	POKÉMON HEARTGOLD	Nintendo DS	760K

While it is not exclusive to that system, the original game was a console exclusive on the Xbox 360 for a year, a fact that probably has helped propel sales of the sequel on that platform.

Rounding out the PS3 list is MAG, another Sony exclusive, and yet another shooter. We note that one can get a sense of the strong contrast between the Wii and the HD consoles using these lists. Whereas Nintendo's list could be characterized as a platformer, a fitness package, a minigame collection, a music game, and a driving games, the Xbox 360 and PS3 lists consist of RPGs, third-person action games, and shooters.

—Matt Matthews

Dragon Age: Origins

BioWare's Dragon Age: Origins Crowned in First Canadian Video Game Awards

VANCOUVER INAUGURATED ITS VERY FIRST CANADIAN VIDEO GAME Awards by crowning BioWare's Dragon Age: Origins game of the year, one of two awards the RPG received.

At the Vancouver Convention Center ceremony, held ahead of GDC Canada, which took place at the same location, the game beat out Assassin's Creed II, Dawn of War II, FIFA 10 and Might and Magic: Clash of Heroes for the award, and also won in the best writing category.

Ubisoft Montreal's Assassin's Creed II took home three awards, in the best console, best game design and best visual arts categories. It also shared the best technology award with Radical Entertainment's Prototype.

The Canadian Video Game Awards are produced by the Digital Media and Wireless Association of British Columbia and its partners, including Reboot Communications (which also co-produced GDC Canada) and Greedy Productions.

"This first event was an excellent showcase of the amazing talent of our video game developers in Canada," said DigiBC president Michael Bidu.

The winners in each category are as follows:

Game of the Year Dragon Age: Origins (Electronic Arts/BioWare)
Best Console Assassin's Creed II (Ubisoft Montreal)
Best Handheld Game Might and Magic: Clash of Heroes (Capybara Games)
Best Downloadable Game Critter Crunch (Capybara Games)
Best Audio Dawn of War II (THQ/Relic)
Best Game Design Assassin's Creed II (Ubisoft Montreal)
Best Technology Prototype (Radical Entertainment) and Assassin's Creed II (Ubisoft Montreal)
Best Visual Arts Assassin's Creed II (Ubisoft Montreal)
Best Writing: Dragon Age: Origins (Electronic Arts/BioWare)
In-Game Cinematic Ghostbusters: The Video Game Cinematics (Rainmaker)
Most Promising Game ModNation Racers (United Front Games)

"Interactive entertainment is one of our country's leading industries where we are clearly world-renowned," added Bidu, "and this world-class event was a fitting way to showcase to the globe the great work done in Canadian studios coast to coast." *—Leigh Alexander*

Independent Games Festival Names Boyer as Chairman

THE UBM TECHWEB GAME NETWORK, ORGANIZERS OF THE YEARLY Independent Games Festival and Independent Games Summit, announced that scene notable Brandon Boyer has been named Chairman of the IGF.

In his new role, Boyer will oversee submission and judging operations, provide community outreach and support, and help shape the structure and continued growth of the IGF—the longest-running and largest event relating to independent games worldwide.

This follows the event's all-time record 607 game submissions in 2010 across the IGF Main Competition, Student Showcase and IGF Mobile competitions, including high-profile titles like Monaco, Limbo, and Super Meat Boy.

The 2010 Independent Games Festival saw thousands of visitors to its Pavilion and more than 3,000 attend the IGF Awards Show in March 2010. The associated Independent Games Summit had nearly 1,000 attendees for its 2010 keynote session on the Indie Fund.

Both events are part of the larger Game Developers Conference, which is returning to San Francisco's Moscone Convention Center Monday, February 28 to Friday, March 4, 2011.

Boyer has previously co-founded and served as editor of Offworld.com, an independent-game-focused site operated by seminal weblog Boing Boing, where he currently serves as contributing editor. Boyer has also served as judge for the IGF since 2007, and was previously an advisor at multiple Independent Games Summit events. He has also contributed to various games publications including Gamasutra and *Edge Magazine*, and brings a wealth of knowledge on the independent games scene to the position.

Boyer (center, with Mac) at GDC 2010.

Boyer is taking over the role from former Chairman Simon Carless, who is now Global Brand Director for the entire UBM TechWeb Game Network, including the GDC shows, Gamasutra, *Game Developer* magazine, and new acquisition Game Advertising Online.

Carless will continue to contribute to the IGF as Chairman Emeritus, and as part of a Festival/Summit organizing committee that includes Boyer, Carless, and vital continuing IGF contributors Matthew Wegner (Flashbang Studios) and Steve Swink (Enemy Airship).

"The Independent Games Festival has consistently been my highlight of the Game Developers Conference for as many years as I have been attending," said Boyer. "I'm extremely honored to help shape the future of the festival and bring ever-wider attention to the indie games community, a group that is truly defining the future of video games as an artistic medium."

The Independent Games Festival was established in 1998 by UBM TechWeb Game Network to encourage innovation in game development and to recognize the best independent game developers. Previous honorees have included many of the pioneering independent games of the last decade, including World of Goo, Gish, Everyday Shooter, Crayon Physics, Braid, Castle Crashers, and Audiosurf. *—Staff*


TOP
30
DEVELOPERS

2009 was a big year for game development. The newest round of consoles had all been on the market for over two years, PC development was going through a renaissance period surrounding digital distribution, and the free-to-play model became nearly ubiquitous among online experiences, as social and iPhone games rose in prominence.

As studios become more tight lipped about the sales of their games and their financials, and as co-development convolutes the landscape, we decided to avoid empirical lists this year, instead partnering with our advisory board to choose what we think were 30 of the best developers in the industry for calendar year 2009.

Those companies we included had to have released at least one game in 2009 (or give extensive support to an existing one), and reasons for inclusion ranged from conscientious development practices, to forward-thinking design or business models, and occasionally simply making a lot of money. Ideally though, the companies we chose did all of these things, while creating games that move our industry forward.

THATGAMECOMPANY

Los Angeles, CA

In 2009, ThatGameCompany released FLOWER, the second game in its three-title PlayStation Network deal with Sony. The game has been widely acclaimed for its visuals, its sound, and perhaps most importantly, its sentiment. FLOWER is something you can point to when someone says "All games are X," or "Games are just made for Y age group." FLOWER is definitely not X, nor is it made specifically for Y, and the idea of experiencing the dream of a flower, as the game asks you to, is uncommon in any media.


FLOWER

This small company is headed up by Jenova Chen on the creative side, and Kellee Santiago on the business side, who together founded the company right out of college. The company's dedication to creating non-combative games and unique experiences is inspiring, which is likely why both Chen and Santiago are frequently requested at game, technology, and future-oriented conferences.

TGC won the "best downloadable game" award for FLOWER at the Game Developers Choice Awards, as well as "best interactive score" from the game audio network guild. As the company hints at its next very ambitious project, one can only hope this small team of 10 can pull out another miracle.

BIOWARE EDMONTON

Edmonton, Alberta

BioWare's Edmonton studio is responsible for the company's major offline RPGs, and this year we honor the release of DRAGON AGE: ORIGINS. The company's commitment to interactive storytelling puts this team at the forefront of the medium (alongside a handful of others). DRAGON AGE did well for BioWare, even selling through to some non-traditional gamers, and persons who wouldn't have otherwise picked up an RPG, due to publisher EA's marketing push. At the first annual Canadian Video Game Awards, DRAGON AGE was named the best Canadian-developed game of 2009.

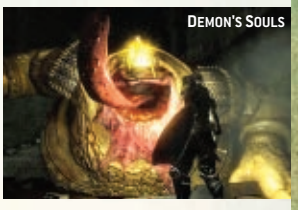

DRAGON AGE: ORIGINS

BioWare is a large studio, so not without its issues, but the company does seem to take time with its games, releasing them when they're ready, and tending heavily toward original IP. Looking into 2010, the studio extended DRAGON AGE with a traditional expansion pack, and released MASS EFFECT 2 to great acclaim. Clearly, BioWare Edmonton's commitment to the single-player RPG is unwavering.

FROM SOFTWARE

Tokyo, Japan

From Software knows its audience. As the creator of complex simulations such as ARMORED CORE and KING'S FIELD, From Software has always played to the refined tastes of veteran gamers. But who could have guessed that the old guard still represented such a sizeable customer base? Certainly not Sony Computer Entertainment America, which passed on the opportunity to publish From Software's dark and exacting DEMON'S SOULS, instead leaving it to niche publisher Atlus to bring to North America. Since the game's release, positive word-of-mouth has helped it sell more than a quarter of a million copies in the U.S. alone, proving that many players still crave a challenge despite the conventional wisdom that the future is casual.


DEMON'S SOULS

With its novel online implementation, high level of polish, and remarkably accessible design for what is essentially an old-school dungeon crawl, DEMON'S SOULS represents a new level of achievement for the developers at From Software. We're looking forward to seeing how the experience influences the development of the studio's next flagship title ARMORED CORE 5.

5TH CELL

Bellevue, WA

5th Cell is nothing if not innovative. This year the company makes the list for its hyper-inventive SCRIBBLENAUTS for Nintendo DS, which allows players to create game assets by writing or typing in almost any word.


SCRIBBLENAUTS

Ambitious concepts seem to be 5th Cell's primary modus operandi, and each of the company's major products has been an original IP. It takes determination and conviction to continue making original games as a smaller company. 5th Cell likely gained this scrappiness as it climbed up from the hardships of mobile phone development some years ago.

Led by creative director Jeremiah Slaczka and producer/cofounder Joseph Tringali, the studio encourages its employees' outside projects, and incentivizes them with royalties, even as it grows above 30 people. Looking forward, the company continues to develop original IP, and we propose that the company will climb the ranks even further once its innovative ideas are fully matched by the ability to execute them. This comes with the added time and respect afforded to more established developers, which 5th Cell is well on its way to becoming. Speaking of respect, SCRIBBLENAUTS wound up taking the "'best handheld game'" and "'innovation'" awards at the 2010 Game Developers Choice awards.

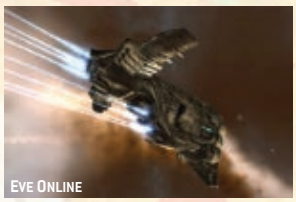
CCP GAMES

Reykjavik, Iceland

CCP Games very smartly understands that the most compelling aspects of an online game are its players. While the studio has rolled out fresh content for EVE ONLINE on a regular basis (2009 saw the release of

the Apocrypha and Dominion expansions), it is the game's Byzantine economic and political ecosystem that keeps players coming back month after month to the tune of 300,000 subscribers, six years after the game's release date.

Also in 2009 CCP met the real money trading challenge head on with a series of creative initiatives, including the massive "Unholy Rage" operation that rooted out over 6,200 RMTers, as well as incentivizing players to keep their money within EVE Online by allowing them to buy extra game time with in-game currency.

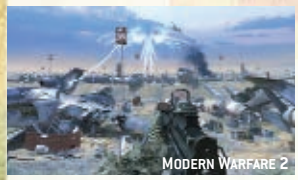

EVE Online

Looking to the future, CCP is spreading its wings into new arenas, with a Chinese-developed FPS (Dust 514), which will interact with EVE in interesting ways, and its long-in-development World of Darkness MMO. All these titles were under development in 2009.

INFINITY WARD

Encino, CA

Known for its Call of Duty series, Infinity Ward has consistently delivered quality blockbuster games to an increasingly rabid audience. From visuals to sound design to moment-to-moment set pieces, IW has raised the bar for the entire FPS genre, especially when it comes to those based in reality.


Modern Warfare 2

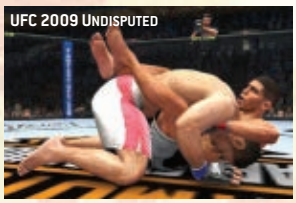
Though 2009's Modern Warfare 2 couldn't quite match the revolution that was Call of Duty 4, the game sold phenomenally well, dwarfing even the sales of its predecessor. And it was this success, coupled with the desire of its founders to control the company's destiny, that resulted in the disbursement of a number of Infinity Ward's key creators in 2010. In 2009 though, the studio was firing on all cylinders, bringing success both to Activision, its parent company, and to many of the developers themselves.

As the company moves into a new phase in 2010, after the exodus of its founders and much of the key staff, there is an opportunity for new blood to come forward—it's our hope that Infinity Ward won't become "just another Call of Duty studio."

YUKE'S

Osaka, Japan

This past year was a good one for Japanese independent developer Yuke's. The company tends to stay a bit behind the scenes, but in 2009 released three titles in the fighting/wrestling genre—WWE Legends of WrestleMania, WWE SmackDown vs. Raw 2010, and most importantly, UFC 2009 Undisputed. This reboot of the UFC series hit at the right time, and wound up being one of publisher THQ's better sellers of the year. What's more, the game was praised for its dedication to the sport, as it turned out to be quite close to the actual movesets of players, rather than changing the combat to become more gamey.


UFC 2009 Undisputed

The game is a true test of precise collision, and holds up better than most. In fact, a member of our advisory board praised the game for having some of the best body interpenetration of the year. The company is currently hard at work on a UFC sequel, and doubtlessly additional WWE games besides.

PLAYFISH

London, UK

Playfish is one of the earlier major players in the social games space, alongside Zynga. The company started in 2007, and wound up getting some $17 million in funding the next year. In 2009, the company was purchased by Electronic Arts for $275 million, as part of the publishing giant's steps toward a larger online presence.

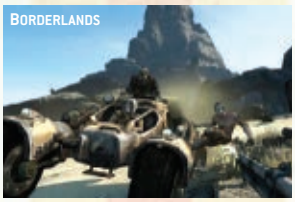

Pet Society

Playfish is known primarily for its games Who Has the Biggest Brain? and Pet Society, and throughout 2009 remained one of the largest players in the space. Going forward it will be interesting to see how this web developer evolves under the wing of EA. In 2010, the developer has already begun work on an EA property (Fifa), and one can imagine even more cross-pollination with the parent company's properties.

GEARBOX

Plano, TX

Borderlands is Gearbox's first original IP since Brothers in Arms, as the company had traditionally chosen to work more on expansions of other companies' titles rather than release its own properties. With Borderlands' release in 2009, Gearbox achieved a new level of success. The game released to critical acclaim and financial reward, which the company says it shares with all of its employees, which is certainly an admirable claim.

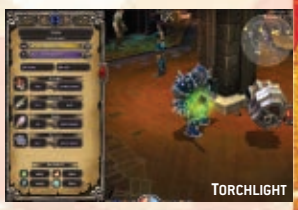

Borderlands

Gearbox has invested heavily in both internal and external focus testing, and has formed a team called "Truth" to handle the customer expectation side of the business. The company continues to tweak, balance, and support Borderlands, releasing downloadable content within 2009 and beyond. Gearbox clearly intends to position this title as a banner for its studio, and shows no indication of slowing its support, whether that be through further DLC or a rumored sequel.

RUNIC GAMES

Seattle, WA

If there is such a thing as sustainable game development, Runic Games may be the example to follow. Formed in 2008 by veterans from the Fate, Diablo, and Mythos teams, Runic delivered the warmly received Torchlight to players after slightly less than a year of development. The game even won the "best debut game" award for Runic in the 2009 Game Developers Choice Awards.


Torchlight

The studio was able to execute on such a tight schedule thanks to a design that had strong roots within an established genre, a small team size, use of the open source OGRE 3D rendering engine, and an art direction that accommodated lower-spec hardware. Now that Torchlight is a proven single-player success, the team is currently leveraging it into a full MMO experience.

CAPCOM OSAKA

Osaka, Japan

Capcom has really been coming into its own again lately, as one of the few major Japanese developers that can resonate with a Western audience with original IP. In 2009, Capcom released a slew of titles in multiple venerable series, including Resident Evil 5, which excelled visually and financially, and Street Fighter IV, co-developed with Dimps.


Street Fighter IV

Street Fighter IV in particular is a marvel of game design and execution, basically single-handedly revitalizing the popularity of the fighting genre. The company is pushing fighting games forward in 2010 and beyond, but it was this initial spark of guts that

allowed an entire industry to reap the rewards. After all, more viable genres means more potential for new games.

2009 was also the year that Capcom released Monster Hunter Tri for Wii, at least in Japan, and the game went on to be the best-selling third party Wii title in the country. Capcom has consistently delivered high quality content, and fingers are crossed both on the company side and the game playing side that this quality will continue.

VALVE CORPORATION

Bellevue, WA

Valve represents the developer ideal. Resolutely independent, with an almost contrarian streak, the studio works on the games that it wants to play and releases them when it's ready. Valve's work is also instantly identifiable; showing an attention to visual detail, smart writing, and carefully tuned design that has earned the studio many committed fans. The name Valve means something to its audience and the studio has been careful to maintain that trust. 2009 saw the arrival of Left 4 Dead 2 and coming as it did so soon after the original Left 4 Dead, it might have been tempting to treat the release as a simple expansion pack. Instead, the studio produced a fully realized game of astonishing graphic elegance and fearsome gameplay.


Left 4 Dead 2

In addition to its great games, a fertile development ecology has grown up around Valve. With more than 25 million PC users of Steam and a just released Mac version, Valve's digital distribution service is providing a stable and lucrative platform for developers and publishers both large and small. Valve's Source engine technology is also the center of a vibrant mod scene.

FIREMINT

Melbourne, Australia

Firemint is one of those few independent iPhone developers that has managed to achieve success in both the high end and low end of the pricing scale, with the ubiquitous Flight Control in the cheaper category, and the 3D racing game Real Racing in the more full priced camp.

The company has been making games for mobile platforms since its formation in 1999, but never achieved breakout worldwide success or recognition until the iPhone platform's launch. The company correctly positioned both titles in terms of design and price point, catering to players who want something lighter (but addictive) and something a bit deeper, respectively.

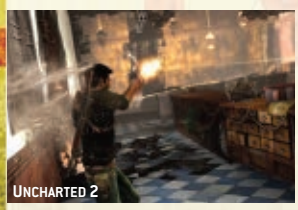

Flight Control

Looking forward for Firemint, the company is bringing both Flight Control and Real Racing to the iPad, looking to capitalize on the titles' existing success.

NAUGHTY DOG

Santa Monica, CA

Naughty Dog has become the jewel in Sony's first-party crown, and with Uncharted 2: Among Thieves the studio delivered a widescreen action adventure that could easily match Hollywood's recent entertainment output in terms of both visuals and narrative. That Uncharted 2 was a video game rather than a film didn't seem particularly unusual to anyone—a fact that speaks volumes about the cultural acceptance of games as a medium for entertaining adults.


Uncharted 2

On the development side, the company has admitted that it's a bit of a controlled chaos at times, but there are a lot of progressive practices at work here, such as the simultaneous voice and motion capturing led by creative director Amy Hennig, which winds up creating a more cinematic feel than previous techniques. Combined with intelligent use of outsourcing and solid visual design, Uncharted 2 wound up becoming one of the smoothest game experiences in recent memory for a large portion of the game population.

On top of that, Naughty Dog swept awards shows internationally, winning "best visual arts," "best technology," and "game of the year" at the Game Developers Choice Awards.

INSOMNIAC

Burbank, CA

Insomniac makes our list for the newest installment in its Ratchet & Clank series, Ratchet & Clank Future: A Crack in Time. Though often glossed over by more adult gaming press, the game was widely considered by fans to be one of the best in the series, showing the power of retaining ownership of your property, compared to series like Spyro and Crash Bandicoot, which floundered under unfamiliar studios.

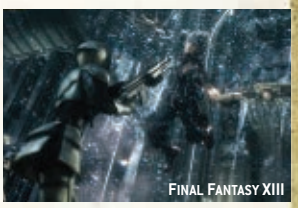

Ratchet & Clank Future

Insomniac also continues its community-facing practices, from the Full Moon Show podcast, which updates gamers about the goings-on at the studio, to its Nocturnal Initiative, which shared the company's code and ideas with other developers (that project seems to have slowed since 2009). Perhaps most importantly, Insomniac employees seem generally happy where they work, and the company has a relatively low turnaround in spite of occasional crunch.

SQUARE ENIX

Tokyo, Japan

Although Square Enix has been very active as a publisher over the last few years, its work as a developer has been relatively low key, limited primarily to remakes of classic titles and updates to ancillary franchises. That all changed with the release of the much-anticipated Final Fantasy XIII in Japan last fall. Now that we can see the results of more than four years of the studio's work first hand, we are reminded again of Square Enix's raison d'être: creating tremendously ambitious RPGs that blur the line between game and cinema.


Final Fantasy XIII

While the resulting design decisions will be hotly debated in the coming months and perhaps years, there is no doubt that the studio has made a bold statement on accessibility and cross-gender appeal with the release of Final Fantasy XIII, distilling the JRPG genre down to its most critical elements.

Outside of the console game world, Square Enix has been making moves in the iPhone space, aggressively developing new content for the system at multiple different scales, eschewing the 99 cent price point to which most developers default. 2009 was also the year that Square Enix purchased Eidos Interactive and its stable of developers.

RELIC ENTERTAINMENT

Vancouver, British Columbia

Relic has become one of the masters of real-time strategy game design, cementing this with the release of 2009's Warhammer 40,000: Dawn of War II.

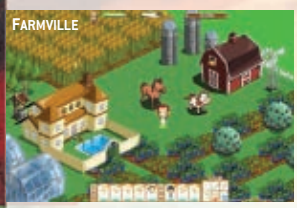

Warhammer 40,000: Dawn of War II

With nearly a decade of Company of Heroes and Warhammer titles under its belt, Relic has been steadily refining its unique approach to RTS design that emphasizes operational mobility over resource farming in a manner that more closely reflects the dynamic give and take of real-world warfare. We can only hope that Relic will one day turn its attention back to the Homeworld franchise that gave the studio its auspicious start.

ZYNGA

San Francisco, CA

Zynga claims 100 million players every month and was one of the first companies to claim profitability in the social space. With development studios in the U.S. Europe, and India, the company is quickly becoming a mega-developer on the social platform, with titles including FARMVILLE and MAFIA WARS.


FARMVILLE

Increasing friction between the developer and Facebook over the revenue split on Facebook Credits may lead to Zynga forming its own independent social gaming network, a move that would have interesting historical parallels to the disruption of the console gaming industry seen in the late 70s that was brought about by the rise of independent publishers.

Zynga's developer count numbers over 800, and the company is clearly forward looking in its approach to the web and games in general. We can only hope that its development practices will rise to match that level of progressiveness. FARMVILLE won "best new online/social game" at the Choice Awards, so our confidence has been placed.

SUMO DIGITAL

Sheffield, UK

Sumo has been toiling away making extremely competent games since its founding in 2003, notably reviving older arcade-oriented titles for their parent companies. In 2009, Sumo (now owned by independent conglomerate Foundation 9 Entertainment) released several notable games that make it worthy of this list.


OUTRUN ONLINE ARCADE

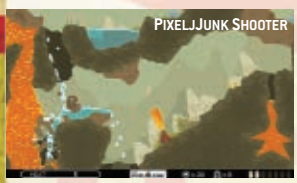
On the downloadable side, the company released rally game GTI CLUB+ for Konami on PSN, as well as OUTRUN ONLINE ARCADE for PSN and XBLA, both to positive critical review. Jumping genres, the company also brought VIRTUA TENNIS 2009 to the table, the first since VIRTUA TENNIS 3 several years prior.

This UK developer, composed of Gremlin Interactive alums, quietly makes very good games, but tends to fly under the radar by virtue of the company name being subsumed by the larger logos of the games' license holders.

Q-GAMES

Kyoto, Japan

Q-Games is uniquely positioned to bridge the space between Eastern and Western game development. Founded by a Briton but based in Japan, Q-Games works closely with both Nintendo and Sony, even working on the cross media bar visualization for the PlayStation 3. The downloadable games arena has been its focus for a number of years and 2009 saw releases from the studio on both DSiWare and the PlayStation Network.


PIXELJUNK SHOOTER

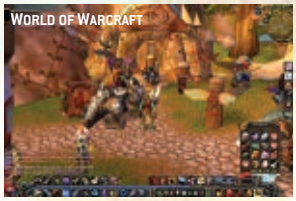
Perhaps best known for the PIXELJUNK games, Q-Games most recently released PIXELJUNK SHOOTER. Part art game, part scrolling shooter, PIXELJUNK SHOOTER features an intriguing mix of action, exploration, and environmental puzzle solving, all wrapped up in elegant 2D art and a pulsing back beat from the The Orb's Alex Paterson. 2009 also saw the release of STARSHIP DEFENSE, a DSiWare tower defense title.

BLIZZARD

Irvine, CA

While Blizzard did not release a major expansion pack to its seminal WORLD OF WARCRAFT in 2009, nor any other game for that matter, Blizzard makes our list for sheer retention of millions of players. WoW is the only subscription-based RPG to retain these kinds of numbers. A few free-to-play games can boast a higher player base than Blizzard's flagship MMO, but those titles generally have a lot of player turnover, and the vast majority of players never pay a cent.

In Blizzard's case, the company has had to make constant patches, smooth out the UI, integrate microtransactions, and simply make sure there's enough content for 11.5 million paying subscribers (as of December 2008) of all levels of experience. This is no simple task, and attests to why the company claims over 4,700 employees.

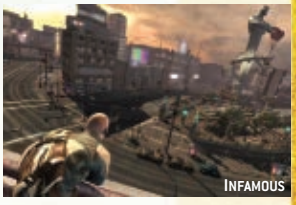

WORLD OF WARCRAFT

In 2009 Blizzard revamped its Battle.net service, moving toward getting all players to use the online client to launch their games. Its intention to streamline the user experience is ultimately being seen by fans as a good thing, despite complaints over STARCRAFT II's requirements to use the system for multiplayer.

SUCKER PUNCH PRODUCTIONS

Bellevue, WA

Comic books and video games have always gone together, but it is rare to see an original title like Sucker Punch's INFAMOUS that so successfully conveys the energy and spirit of comic book heroics (and villainy). Employing a go-anywhere design, INFAMOUS turns the city into giant romper room for super hero antics.


INFAMOUS

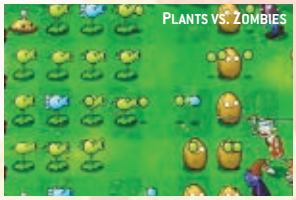
Sucker Punch was previously known for its kid-friendly series of SLY COOPER games and the studio's winning transition to the more mature style of INFAMOUS is a good example of playing to your strengths as a developer while reaching new audiences. The PlayStation 3 exclusivity of the title arguably limited the audience, but on the flip side may have provided a marketing boost.

POPCAP

Seattle, WA

As is common for casual game companies, PopCap did a lot of work in 2009 to bring its existing games to new platforms. The company released over two dozen titles across almost every platform.

Lately though, the company has really been coming into its own with original IP. We started taking more notice of the company in this regard when it released PEGGLE in 2007, and since then the hits have kept coming. 2009 was no exception, with the release of the indomitable PLANTS VS. ZOMBIES, created by George Fan.

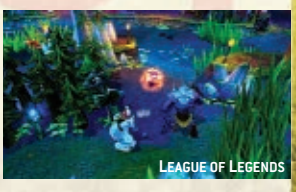

PLANTS VS. ZOMBIES

PLANTS VS. ZOMBIES has taken all downloadable markets by storm, instantly rocketing to the top of any platform it touches (though for the period discussed, that's only PC). The game shows, alongside several others, that dedication to polish can make what would otherwise be a traditional tower defense game into something much stickier.

RIOT

Culver City, CA

This independent developer released its first game, LEAGUE OF LEGENDS, in 2009. The game is based on the exceedingly popular DEFENSE OF THE ANCIENTS map in WARCRAFT III, which wound up spawning an entire genre. Riot's game, for its part, included input from some of the original map's creators.


LEAGUE OF LEGENDS

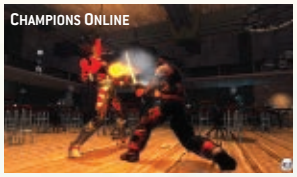
LEAGUE OF LEGENDS follows the now-pervasive free to play, pay for items business model, and has launched in

the West, but additionally in China, where this particular model is already fully accepted as the de facto payment scheme. The company prides itself on working with fan input to create its content, boasting a "suggestion engine" through which players can request changes. The company requested input all through initial development as well, which when combined with its solid matchmaking service makes for a very player-friendly experience.

CRYPTIC

Los Gatos, CA

In its first full year under Atari ownership, this small company managed to release two full-fledged MMOs within six months of each other—Champions Online and Star Trek Online. Only Champions was released during calendar 2009, but the company's ability to work on both with under 200 employees is impressive. Cryptic focuses on player-versus-player games specifically, and excels in creating compelling sandboxes in which players can test their might.

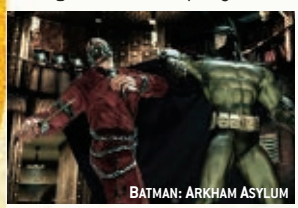

Champions Online

Looking to the future for Cryptic, it seems that while the company can launch successful IP to critical acclaim, it needs to reinforce its infrastructure to support these titles for the long term. The company suffers from the predicament of many mid-size online game developers—its games follow the typical player excitement curve of many MMOs, where much of the content is front-loaded, and the game is developed as the players play. 2010 will be a critical year for Cryptic, as it supports STO and plans new projects.

ROCKSTEADY

London, UK

Rocksteady has rocketed itself into the elites of the UK game industry with the studio's sophomore effort, Batman: Arkham Asylum. With that game, the company was suddenly the jewel of the country, and cofounder/game director Sefton Hill's name was everywhere, up from comparative obscurity.


Batman: Arkham Asylum

This isn't without reason—the game was the best use of a license in recent memory, and one of the better-designed games of the year. Love for the franchise combined with solid execution meant the game was universally lauded by the press, and praised by the industry, winning the "best game design" award at the game developers choice awards. It's been a big year for the company on all counts, as it was acquired by Time Warner (Eidos/Square Enix retains a 25 percent stake), and won "best studio" in the Spike TV Video Game Awards for 2009.

UBISOFT MONTREAL

Montreal, Quebec

Out of all the mega-studios, Ubisoft Montreal strikes us as one of the most creative and artistically successful. With well over a thousand developers on staff and partial or whole responsibility for high-profile franchises like Assassin's Creed, Tom Clancy, Prince of Persia, and Far Cry, Ubisoft Montreal is a behemoth.

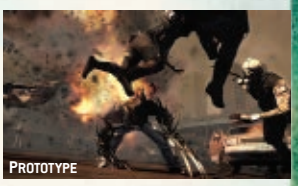

Assassin's Creed 2

But with that size comes deep artistic and technical resources that allow it to recreate detailed game worlds such as the ornate architecture of Renaissance-era Venice in Assassin's Creed 2, the lush forest world of Pandora in James Cameron's Avatar: The Game, and the grim, desaturated architecture of Tom Clancy's Splinter Cell: Conviction (which the studio worked on for much of 2009).

RADICAL ENTERTAINMENT

Vancouver, British Columbia

Radical Entertainment, owned by Activision, had been doing yeoman work on licensed properties for most of the past decade but with Prototype the studio stepped out with an original title of its own. The result was a smoothly-playable exploration of the sandbox school of game design. Weirdly violent in places but with a smart storyline that plays on modern fears of infection and control, Prototype was the confident work of a studio with near decades of development know-how.


Prototype

Though Radical was hit with layoffs, they didn't occur until 2010, and did not seem to reflect at all on the quality of Prototype or the company's recent offerings.

VOLITION INC.

Champaign, IL

In Volition's Saint's Row series the developer seemed to be walking a shaky line, sometimes serious but often veering into pure video game ridiculousness. With Red Faction: Guerrilla the game's tone is altogether more somber but with plenty of opportunities for emergent fun thanks to the destructible play environments afforded by the studio's Geo-Mod engine.


Red Faction: Guerilla

As a studio, Volition has always had an eye toward the unique play opportunities that its technology enables, going all the way back its first title Descent: FreeSpace which featured enormous destructible capital ships. The ability to deal out physical chaos on a massive scale is one of the more unique experiences to be found in video games and it is a pleasure that Volition continues to refine.

New Super Mario Bros. Wii

NINTENDO ENTERTAINMENT ANALYSIS AND DEVELOPMENT (NINTENDO EAD)

Kyoto, Japan

Despite Nintendo's prominence in the game industry the company has always seemed a bit removed from the business as a whole. This is because Nintendo is essentially a toy company: both historically and in its institutional mindset. While the game industry goes about its business, Nintendo has been happy to follow its own path, successfully selling its toys to both children and adults while paying little heed to changes in game fashion.

While the space marine set may look askance at Nintendo's brightly colored, general audience offerings, tightly designed titles like Wii Fit Plus, Wii Sports Resort, and New Super Mario Bros. Wii have gone on to sell millions of copies this past year. Special note must also be made of Nintendo's innovative "Super Guide" player assistance system, first seen in New Super Mario Bros. Wii, which aims to help less skilled players without ruining the game's creative presentation.

Nintendo continues to innovate in the console game space, broadening the market to new consumers that only Facebook could hope to touch, and it's the internal Kyoto studio that leads the way in this regard.

VVoh!\I[)3r

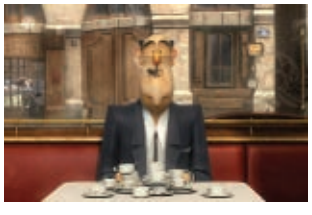
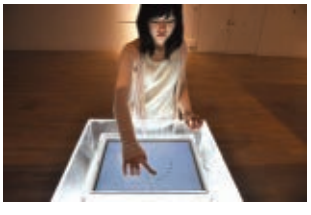
Join the people behind the pixels in the world capital of digital imagination.

The People Behind the Pixels

SIGGRAPH2010 Los Angeles

share ideas build relationships advance your career be inspired

French Roast © 2009 Fabrice O. Joubert, The Pumpkin Factory; Touch the Invisibles © 2009 Junji Watanabe, PRESTO Japan Science & Technology Agency


The 37th International Conference and Exhibition on Computer Graphics and Interactive Techniques

Conference 25-29 July 2010 Exhibition 27-29 July 2010

Los Angeles Convention Center Los Angeles, California USA

Art Gallery • Art Papers • Birds of a Feather • Computer Animation Festival • Courses • Emerging Technologies Exhibition • Exhibitor Tech Talks • Featured Speakers • Game Papers • International Resources • Job Fair • Panels Posters • Research Challenge • SIGGRAPH Dailies! • Talks • Technical Papers • The Studio

FOR COMPLETE DETAILS:
www.siggraph.org/s2010

SPONSORED BY ACM SIGGRAPH

FOLLOW US ON:

AN ARTIST'S EYE

APPLYING ART TECHNIQUES TO GAME DESIGN

ROB KAY AND ANDY TUDOR

SOME PEOPLE WILL TELL YOU THAT HAVING A GOOD EYE IS an artist's gift—that no sooner have artists left the womb than they're able to see the world differently than regular folk, and hence have a natural tendency toward brilliant art. These people couldn't be more wrong. The notion that artists are "born, not made" is complete nonsense. The simple truth is that artists learn to hone their eyes over many years of education and practice.

The two authors of this article both took this long journey to obtain good eyes, and then journeyed past, to the brave frontier of game design. In making the switch we found that visual arts and game design share a lot in common. The principles we'd learned as we became artists were useful in game design too. In this article we'll share three art principles that we've found particularly useful as game designers: thumbnail sketches, the Golden Ratio, and anticipation.

THUMBNAIL SKETCHES

» Thumbnail sketches are small abbreviated drawings, just an inch or two in size, which artists use to explore ideas quickly. A graphic designer might do thumbnail sketches of possible page layouts, a fashion designer might sketch interesting new silhouettes, and a product designer could explore new form factors. These rapidly sketched ideas give them multiple directions for attacking their problem. They help artists find strong ideas and rule out weak ones before precious time and resources are committed. A common practice is to explore a wide range of possible solutions with thumbnail sketches before committing to any single idea. This approach works, is widely practiced, and

becomes second nature to most artists.

I first had thumbnail sketches explained to me in my high school graphic design class, and have used them non-stop since. Jumping into game design, the habit stuck and I've found thumbnail sketches really useful here as well. These "sketches" don't have to be in perfect visual form either, if you're not an artist. Ideas can be sketched out in text, diagram, diorama, or stick figures.

It's worth noting that the major difference between thumbnails and prototypes is that thumbnails are far faster to create. Creating them quickly means a range of possible options can be explored in minutes, not days. The key use of thumbnails then is a rapid exploration of the solution space prior to prototyping. The end result is one well-defined problem, and multiple ideas for solving it. This early exploration via thumbnails increases the chances of revealing the "obvious solution," which of course only becomes obvious once it's found. Here are three examples, two from my time at Harmonix and one from AiLive.

Rock Band, drum controller design. Early on, I did some thumbnails of various drum pad layouts, mostly to help me think visually about the various alternatives we were discussing. This varied from things like the number of pads, asymmetrical vs symmetrical layouts, and different mappings of drums and cymbals to pads. By doing quick drawings of the layouts which bubbled up in conversation, I was forced to identify more important criteria (for instance, pads should be readable as one row to aid usability). This in turn fed into more design discussion. The solutions we gravitated toward at this early stage (four pads in a row, kick pedal mounted to the frame) shaped our first physical prototype, became requirements for the industrial design phase, and can be seen in the shipping drum controllers.

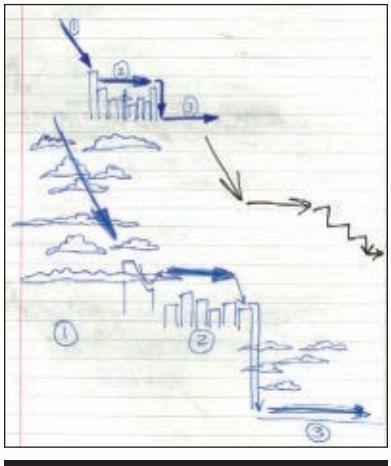
EyeToy: AntiGrav, level design thumbnails. On this project, quick sketches helped us explore big-picture ideas for each level and the main sections within. Ideas came from the whole team through Harmonix's internal newsgroups, and were incorporated into thumbnail sketches and resulting level designs. The ideas came together first in sketch form, informing the digital 2D maps and 3D level creation that followed. The sketches focused on the main lines through a level, dramatic changes of environment and direction, and most of all, created a sense of journey. Time pressure limited the amount of exploration, but there was just enough for solid ideas to come together on paper, and the resulting levels were well received.

AiLive, game concepting. The thumbnail approach has helped us generate and evaluate new game concepts at AiLive. In this case, the thumbnail sketches weren't literally sketches, but words: a collection of very short game descriptions in the spirit of thumbnail sketches. Each game thumbnail attacked our new project's brief from a different angle. I'm happy to say a clear winner emerged, and while I can't say much about it yet, I will say it wasn't the first interesting idea we had; an important point we'll return to in a moment.

When creating and reviewing thumbnails, there are a couple of tricks to avoid pitfalls. First, you should never go with the first idea. This is easier said than done. It can be very tempting to develop the first half-decent idea that emerges. Jumping straight into a design or prototyping phase like this is a risky move. No amount of detailed thinking or iteration makes up for a bad early design call. Iteration is like walking up a mountain. You need to do it to reach the summit, but it helps if you choose the right mountain first. Or put more simply, you can't polish a turd.

Avoiding this pitfall is simple. Be diligent. Explore different ideas in thumbnail sketches before developing any single idea too far. Force

The shipping Rock Band controller, the pad layout of which first found form in thumbnail sketches.



ABOVE: A thumbnail sketch for EyeToy: AntiGrav's skyway level. Sections 1, 2, and 3 rough out the big picture of a journey which descends though clouds, rooftops, and down the side of a skyscraper into the city. RIGHT: A moment from the rooftops section in the final version of EyeToy: AntiGrav.

yourself (or your team) to come up with several approaches to your problem, thumbnail them, and do a little analysis of the options on the table before making big design calls.

Second, make the criteria explicit. It's much easier to evaluate an idea objectively as strong or weak with a list of criteria. What qualities are required in the solution? What's desirable? What isn't desired? Exploring options via thumbnails gets you thinking about the pros and cons, and forces you to get a clear picture of your own criteria. The more explicit the needs, the more defined the solution space is, and the easier it is to generate good ideas. A lack of criteria can lead to idea churn, and can force you to run with an idea you're not sure about.

If there are enough criteria, it can make sense to throw together a quick table to make evaluation easier. For example, list your idea thumbnails (one per row) and your criteria (one per column). Find a simple metric by which to rate each column and complete the table. This helps clarify the big picture, and makes a good conversation point for collaborative decision-making.

In GEARS OF WAR 2, guided by the Golden Ratio, the player's eye and Marcus' path are channeled between the two buildings, with the main action area staying in the bottom third of the screen.

THE GOLDEN RATIO

» Since 500 BCE, the arts and sciences have studied and utilized the Golden Ratio in their work. Either consciously or subconsciously, the usage and recurrence of the mathematical constant 1.618 and the proportion of thirds in nature, architecture, and biology is one that has kept boffins and people who wear bow ties intrigued for years.

At the risk of comparing BAYONETTA's currency progression system to the works of Pythagoras and Da Vinci, the Golden Ratio is also in active use in video games, as a modern day medium where art and science collaborate.

For the purposes of simplicity, though there are many equations and diagrams that can aid the definition of the Golden Ratio, we'll consider an approximation—the relationship between one-

third and two-thirds. This ratio creates an interesting asymmetrical balance, and is often used as a compositional device. In this guise, the Golden Ratio can be extensively seen in game interface design, camera positioning, and environment layout. If we take a broader look at the mathematics around gameplay length, the Golden Ratio can also bear some interesting analysis there too.

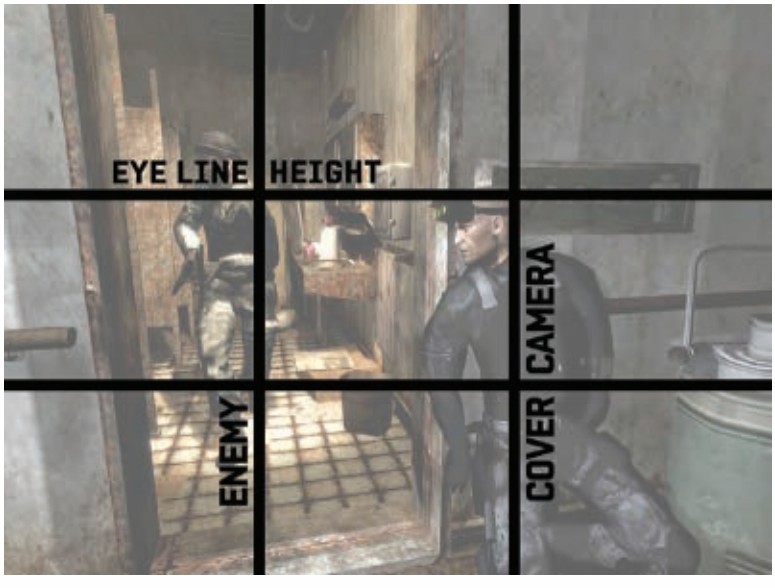
Dependent on genre or platform, one statistic that remains generally constant is that a third of purchasers actually finish games, according to research we did at sony. In today's world of "butterfly media" (where users move rapidly between different media sources), traditional games now fight for spare time with many other electronic activities, such as TV, social networking, and mobile gaming/Internet browsing. It's worth considering, then, that the chances of players actually getting that final Weapon of Epicness or reaching that Ultimate Boss Battle are generally quite slim.

Therefore, one good usage of the Rule of Thirds is to keep splitting your total game length into thirds until you reach a "gameplay bite" length of around half an hour (the average time spent by players in a session).

If you ensure that some new experience or gameplay change or substantial reward is synchronized with these half-hour session, you can be more confident that the player is going to keep returning to the game because to them, regular and meaningful "carrots on sticks" provide compelling gameplay rewards.

Evidence of this can be seen in some of the more memorable and successful games of the past few years, such as RESIDENT EVIL 4, UNCHARTED 2, SHADOW COMPLEX, and BATMAN: ARKHAM ASYLUM, all of which continue to throw brand new experiences at the player. Whether that's with a new mission type, weapon, location change, story twist, character introduction, or power-up, the regularity of these incentives may keep the player involved beyond their traditional cutoff point. Subconsciously, players look forward to the next reveal, since they know they won't have to wait very long for it.

On NEED FOR SPEED: SHIFT we achieved this by layering multiple reward mechanisms of Cash, Badges, Stars, and Driver Levels in order to create a persistent urge of "just one more race" within the player. Where possible, these were timed to occur within half hour bites, and at a higher level, we used the concept of thirds to structure the introduction of game modes, Tier unlocking, track and car reveals, and communication of the roadmap for the player's final entry into the NFS World Championship. We also provided sneak peeks of future events in Career in order to give the player a chance to experience our equivalents of the Weapons of Epicness ahead of time. Telemetry showed that over 55 percent of players reached the penultimate tier in the game due to these choices, significantly higher than the baseline statistic.



When in cover, the camera in SPLINTER CELL naturally moves into position so that Sam Fisher takes up a vertical third of the screen, with his eyeline hitting the top horizontal third.

A simple interface in MASS EFFECT 2 allows the player to reconstruct Commander Shepard's face. This is made all the more aesthetically pleasing and user-friendly through the lens of thirds.

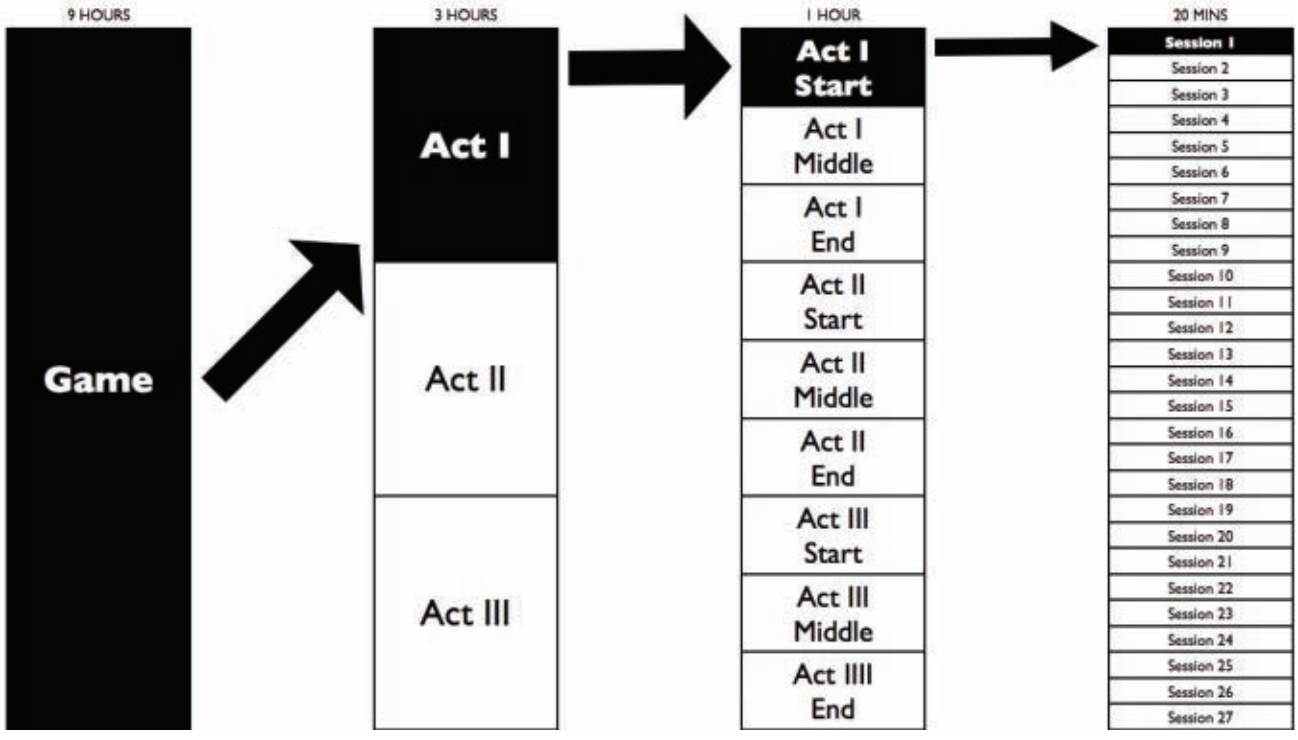
ANTICIPATION

» Back in the 1930s, Disney's "nine old men" coined 12 basic principles for animation. One of the most insightful is anticipation—the preparation for an action. It turns out that when animating, the individual actions themselves are relatively unimportant. The key is selling the audience on the idea that something is about to happen (the anticipation) and then a moment later that it did happen (the reaction). For example: a boxer pulls his arm back to prepare for a punch (anticipation), strikes his opponent (action), who recoils from the hit (reaction). Every action has three distinct phases: anticipation, action, and reaction. To ensure their animation can be read by audiences, animators the world over use this three-step approach, which always starts with anticipation.

So, how can this principle be used in game design? It turns out that the anticipation-action-reaction model is a great aid for crafting clear and rewarding player interactions.

For a moment, let's re-frame games as interactive animation. Games are much richer than this of course, but the simplification is useful for breaking down the magic of a single interactive moment. So, in an interactive animation, the player co-animates the action with the game. The game serves up opportunities for the player to trigger action animations using button presses, and does it so expertly that it feels to players that they, rather than the game, are driving events forward.

It's an illusion of course. In fact, each and



Divide gameplay length into thirds until you reach a gameplay bite of around a half hour, and then plan a new experience or reward at these markers to ensure a steady and meaningful progression.

every interaction moment is an illusion which can be analyzed with our anticipation-action-reaction model. Break it down, and it's the game that animates the anticipation, which cues the player to trigger the action animation, which the game swiftly follows up with a suitable reaction animation. What the game is responsible for animating here is critical. It's creating both the anticipation and the reaction, telling the player what to do and letting them know that they did it. Each and every moment in a game is forged between game and player in this manner.

Let's look at some examples using an animator's eye for spotting the anticipation, action, and reaction.

First, let's discuss the so-called quick time events in the GOD OF WAR series. The player is asked to perform a sequence of moves, one at a time. Breaking down one move using our model, a monstrous enemy is stunned with a large circle icon floating above its head. The stun animation and the circle icon is the game's form of anticipation. It cues the player to press the circle button quickly, triggering the attack move. This button press is the action. In response to the button press, the game animates Kratos attacking the enemy who recoils from the attack, often with some animated gore as payoff. This is the reaction.

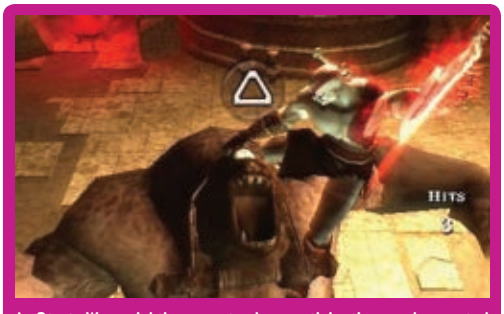
Now, let's look at another GOD OF WAR moment, this time a block move in combat. An enemy within attack range of Kratos pulls his sword back. This is the anticipation, exaggerated and held in place just long enough for the player to spot it. This "tell" cues the player to hold down the block button. In response to this button press, the game animates Kratos into a block stance, and the incoming enemy attack is greeted by a ricochet reaction which includes a gratifying shower of metal-on-metal sparks. This is the reaction.

The final example is almost devoid of animation, showing that the principle is a universal one. Consider Star Power activation in GUITAR HERO. A "Star Power Ready" message flashes center screen as the Star Power meter begins pulsing to draw attention. This is the anticipation. The player lifts the guitar neck upwards providing the action. The game sends electricity FX rushing down the play surface, notes turn electric blue, and the crowd roars. This is the reaction.

It's worth noting that the anticipation-action-reaction model has a parallel in interface design, known as "cue, action, feedback," about which much has been written (see Resources). No matter the language you use to describe it, it's a powerful tool for analyzing and creating interaction moments in games.

Here are three rules of thumb that may prove helpful when sandwiching player actions with anticipation and reaction:

Stage it. In animation, staging is the clear presentation of an idea. This usually means communicating one visual element of the game at a time, leading the viewer's eye to that one element, exaggerating its size or



As GOD OF WAR quicktime events show, anticipation can be created anywhere in the game's user interface. In this case, it's created on the HUD as well as the enemy character. It could just as easily come from the environment, the lighting, visual effects, the sound effects, the music, the controller rumble, or any other channel capable of communicating with the player.

Star Power in GUITAR HERO carries a theme into the UI for maximum impact.

duration. Staging can be particularly tricky in games, as we often deal with an interplay of systems which, by their nature, throw up near simultaneous interactions. The more actions you anticipate at any given moment, the less likely it is that the player will see or hear them, and the less well-staged they'll be. A cluttered HUD is an obvious symptom of poor staging. Some tips to help staging: anticipate only when you need the action (not all the time), remember to use audio and tactile channels, and always look for opportunities to simplify.

Theme it. Ideally, every element of a game reinforces the intended player experience. Theming can be a great way to do this. For example, to help the player feel like a GUITAR HERO, we themed our secondary game mechanics in that game around Star Power. The idea was to boost the showmanship side of being a rock star knowing we already had musicianship covered in our core beat-matching gameplay. We knew the user interface would need to be clear and themed around our idea. The question became what does Star Power, which is a bit of an abstract concept, look and sound like? The electricity theme we used felt right for a game about rock 'n roll, and crucially gave us a bold color (electric blue) to aid with staging.

Playtest it. It's impossible to know if anticipation-action-reaction solutions are working until you actually see people "get it," as they're co-animating your game with you. Clear anticipation is particularly hard to get right. Fortunately, it's very easy to tell when you're failing. If your playtesters don't perform the action you're expecting at the moment you want them to, chances are your anticipation isn't clear enough (which usually means it's too subtle).

resources

ILLUSION OF LIFE by Frank Thomas and Ollie Johnston

THE ANIMATORS SURVIVAL KIT by Richard Williams

A PAINTER'S GUIDE TO DESIGN AND COMPOSITION by Margot Schulzke

THE POWER OF LIMITS: PROPORTIONAL HARMONIES IN NATURE, ART, AND ARCHITECTURE by Gyorgy Doczi

THE DESIGN OF EVERYDAY THINGS by Donald Norman

UNIVERSAL PRINCIPLES OF DESIGN by William Lidwell, Kritina Holden, Jill Butler

ART FOR ALL

>> That concludes our whistlestop tour of the proven art principles we've found useful in game design. We started off with the thumbnail method, proven to help avoid common pitfalls in idea development. Andy showed you how the Golden Ratio can be used to stage a game's rewards and content. The tour finished up by showing how animators sell individual actions using anticipation, and how designers can harness the anticipation-action-reaction model to inspect and craft every moment of interaction with the player. Knowledge and practice of these principles can help you inspect any game with an artist's eye.

Applying proven art principles to your game design will pay dividends, and what we've covered here is just the tip of the iceberg. I'd encourage game designers from any background who want to improve their eye to break out some art books (see Resources for our recommendations). Let's not forget that many revered game designers began their careers as artists—Fumito Ueda (ICO) and Shigeru Miyamoto (MARIO) to name just two. So you'll certainly be in good company!

ROB KAY *has been making games for a decade, most notably as lead designer of GUITAR HERO and ROCK BAND. He began his career animating Looney Tunes characters, and is now creative director at Wii MotionPlus creator AiLive, where he's leading a new game project.*

ANDY TUDOR *has worked in the game industry for ten years, creating environments and cinematics for titles like PRIMAL, 24, and EYETOY. Most recently, he was lead designer on NEED FOR SPEED: SHIFT at London-based developer Slightly Mad Studios.*

ANTICIPATION

ACTION

REACTION

Unreal Technology News

by Mark Rein, Epic Games, Inc.

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

Epic's Unreal Engine 3 has won Game Developer magazine's Best Engine Front Line Award four times and is also one of the few Hall of Fame inductees.

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

Upcoming Epic Attended Events:

E3 2010
Los Angeles, CA
June 15-17, 2010

Develop
Brighton, UK
July 13-15, 2010

Gamescom
Cologne, Germany
August 18-22, 2010

Please email:
mrein@epicgames.com
for appointments.

ZOMBIE GOES DOWNLOAD-ONLY WITH UNREAL ENGINE 3 FOR "BLACKLIGHT: TANGO DOWN"

Independent developer Zombie Studios has been working with Unreal Engine technology for almost a decade, shipping games like "America's Army: Special Forces," "Shadow Ops: Red Mercury," and "Saw: The Video Game."

The studio's latest project, "Blacklight: Tango Down," is a fully featured, downloadable multiplayer shooter for Xbox LIVE Arcade, PlayStation Network and PC slated for release this summer.

Zombie Studios' *Blacklight: Tango Down*

Set 25 years in the future, "Blacklight: Tango Down" is a covert military action epic based more on science-fact than science fiction.

Early buzz has emphasized the game's high-quality visuals, complete leveling system, Black Ops co-op mode, generous customization options, equipment like Hyper Reality Vision (HRV) visors and weapons like digi grenades. Multiplayer modes support up to four players with the ability to seamlessly jump in and jump out of the action at any time.

Zombie has found certain Unreal Engine 3 features to be indispensable for quickly achieving the team's creative vision.

"I think Epic's Lightmass and Scaleform GFx features are my favorites in 'Tango Down,'" said Mark Long, CEO of Zombie Studios.

"Ambient occlusion takes scene realism to a new level. Our environments look stunning after just a default pass on lighting.

"What used to take weeks before can be done in hours with Lightmass. The level of control the environmental artists and level designers have allows us to get under the hood to optimize every element, so these visual improvements don't impact framerate."

"GFx is Scaleform's Flash transcoder for UI components," Long added. "Authoring in Flash and exporting directly into Unreal allows our interface artists — and increasingly, our level designers — to rapidly iterate UI designs and to add state-of-the-art motion graphic effects to our menus, HUD and even mini-games. We're using almost every feature Scaleform offers in 'Tango Down.'"

Long explained how Epic's direct licensee support in tandem with the prolific community of Unreal Engine developers is crucial to his team's productivity.

"The Unreal Developer Network (UDN) is Epic's secret weapon," said Long. "There are literally hundreds of developers sharing information in real time on the UDN.

"The support from Epic is great, but nothing beats the support of a peer trying to solve a similar problem. Epic is way ahead of the curve in community tools and communication."

Zombie is pushing the envelope even further with its "Blacklight" property, expanding the military action adventure franchise to include movies and comic books from Fox Atomic Studios. And it's using Unreal Engine 3 technology in some new and exciting ways.

NOT JUST GAMES: ZOMBIE TO MAKE FEATURE FILMS WITH UNREAL ENGINE 3 AS WELL

"We're doing something really cool with Unreal Engine 3 that no one has tried before," added Long. "Through our partners in Blacklight Transmedia, we're using Unreal to produce a live-action feature film titled 'Samurai.' We just completed a proof-of-concept that composites live-action actors shot on a green screen with props and environments created in Unreal. The goal is to share movie and game productions, and it is amazing looking!"

Also on the horizon at Zombie is the Unreal Engine 3-powered, third-person survival horror game "Saw II" for PlayStation 3 and Xbox 360. Published by Konami, "Saw II" is scheduled for a simultaneous release with the "Saw VII" motion picture this fall.

POWERED BY UNREAL TECHNOLOGY

EPIC GAMES

For UE3 licensing inquiries email:
licensing@epicgames.com

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

WWW.EPICGAMES.COM

DARKS

TIMOTHY BELL

IN THE SPRING OF 2005, FOUR DEVELOPERS WENT TO E3 TO PITCH A GAME ABOUT THE FOUR Horsemen of the Apocalypse to any publisher who would listen. Most publishers were polite, but clearly not interested in backing a new IP from a team of four guys with no track record. THQ was one of the last publishers to hear the pitch that year and by the time they saw the demo on a cracked monitor, the Vigil Games team had all but thrown in the towel. Nine months after that pitch, THQ acquired Vigil Games and work began in earnest on DARKSIDERS.

Now that production on DARKSIDERS has wrapped, the story of how the game began has been romanticized into a tale of "four guys, a cracked monitor, and a dream." But the reality is that the challenges our studio faced over the four years that followed were enormous. We had to create a new IP, a new proprietary engine, new tools, grow the team from those four people to over 124 people, and find a way to balance a second mammoth project.

WHAT WENT RIGHT

1) STRONG VISUAL DIRECTION. Joe Madureira was one of the four founders of Vigil Games. Joe is one of the most well-known comic book artists around, and he has a very distinctive art style. Our visual goal was to make every character and environment in DARKSIDERS reflect his visual style.

The character side of things proved easier than the environment side. Joe drew the vast majority of our creature and character concepts. There were a few issues initially because Joe likes to draw his characters in dynamic poses. This approach presented a challenge to the character artists because there weren't a lot of traditional T-poses or character turnarounds to give the artists a clear direction on a given character from all angles. In those cases, the character artists were able to work directly with Joe to resolve any issues.

When you look at Joe Mad's comic work, you'll notice a distinct lack of environment detail in most panels. The focus is always on the action and the characters—not on the backdrops. We had to find ways to dissect Joe's character style and then apply those principles to the environment art. A lot of next-gen console games use photo reference for the bulk of their concept work and texture detail. We couldn't use most of those libraries as a base because Joe's art style is anything but photo-realistic. Moreover, there simply isn't a real world reference for many of the more fantastic locations that we wanted to create in DARKSIDERS. A ton of iteration was involved in finding the right mix of "painterly" style with texture detail.

SIDERS

2) STRIKE TEAMS. We tried multiple variations on the strike team concept over the course of DARKSIDERS' development. Our early strike team efforts would definitely belong on the "what went wrong" side of the ledger.

Right after we first entered full production mode, we decided to divide up the work based on the major points of interest in the game. The "hub and spoke" nature of the game world made it easy to find division lines and break the world into what we believed to be manageable chunks. We then assigned members from each discipline to these strike teams and empowered them to work as a group to complete each game area. We tried to divide the experience and talent pool equally across each team to ensure no group would fall behind. The strike teams were full of developers waiting to unleash their creative talents. What could go wrong?

Everything.

The milestones for each team were several months apart, so we didn't discover until too late that most teams had gone off into the weeds. They weren't just a little off track—they were so far off the track that most of the work had to be completely redone. This situation infuriated the team as a whole and wreaked havoc with our dates. It would be a while before we tried using strike teams again.

We revisited the strike team concept toward the end of production. This time around, we used significantly fewer people on each team, gave them extremely clear goals, and required daily updates on their progress toward those goals. Each week ended with a demonstration of their progress to senior management. We knew that the increased oversight would help keep asset production on track, but we were concerned that the removal of the creative aspect of the strike teams might adversely affect team morale. We were wrong. The strike teams were able to solve problems quickly and efficiently, and the demonstrable progress that they made each week provided a ton of positive reinforcement for everyone involved. As a studio, we plan to continue using strike teams in this way across all projects in the future.

3) FIRST PARTY-RELATIONSHIP WITH THQ. Vigil Games was integrated into the THQ studio system following the acquisition in February 2006. Our studio was entirely focused on the Herculean task of building the game engine, tools, content, and team all at the same time. THQ provided all the HR, finance, and other infrastructure support necessary for us to function as a studio during that first year of development. They continued to provide support as we moved some of that infrastructure in-house, and support ramped up again towards the end of the project when we worked closely with the THQ QA, CQA, and PD teams to get the game through the submission processes for Sony and Microsoft.

One of the high-level goals of the THQ studio system is to share as many of the tools and as much of the tech as possible between each of the studios. Not a lot happened on that front for the first two years after we joined THQ because of the incredible variety of game genres and platforms that were supported across the company. In the last 16 months, we've started to make inroads on that front. The engine we created for DARKSIDERS is now being used for other titles within the THQ studio system, and more tech-sharing initiatives are planned for the future.

Production art by Joe Madureira.

4) USABILITY TESTING. Our first attempt at usability testing was a disaster. The goal was to see if the gameplay would resonate with our target audience. Unfortunately, the people included in the test were largely first-person shooter fans who had little interest in third-person action adventure games like ours. The usability lab itself was still under construction and had never done a focus test prior to DARKSIDERS. Not surprisingly, the feedback from those focus testers was all over the map. The hardest of the hardcore shooter fans didn't like it because it was "too easy," and it clearly wasn't a first-person shooter. Many testers didn't have much to say about anything in the game. Ironically, our most enthusiastic support came from a tester who never made it out of the starting crater.

All our subsequent focus tests were handled by a professional usability test lab. We worked closely with the head of that lab to determine what we should focus

on for each test. He then spent time with each build to make sure that what we wanted to test could be fairly evaluated using the build we provided. Those usability tests were an invaluable tool in helping us balance the pace of the game (puzzle solving versus combat), discover areas that needed additional tutorial help, and revealed features of the game that players simply didn't use on a regular basis. These tests were so useful that we decided to build a small usability test lab within our new facility.

5) CONSTANT ITERATION. Absolutely nothing in DARKSIDERS made it into the game on the first try. In the early days of development, much of that iteration was focused on trying to hit the desired art style for the first time or making the core game mechanics "feel right." These iterations were extremely painful at times, particularly during the early production days when our milestones were several months long and we didn't have weekly feedback sessions. As our production process matured, those iterations happened more frequently and were smaller in scope, resulting in the desired level of quality but with less frustration for the team.

The Choking Grounds area in DARKSIDERS is a good example of how the iterative process greatly improved the final result. In the original version of the Choking Grounds, the player used the Mask of Shadows to find Caster enemies that were scattered around the level. The Mask also revealed hidden geometry in the area that the player had to use to get to the Casters. The center of the Choking Grounds was made up of narrow paths and steep cliffs.

Early on, the area was visually where we wanted it to be, but our usability tests revealed that players didn't like it. In fact, they hated it. Riding Ruin (the main character's horse) around the narrow passages was difficult at best, and killing the Casters to complete the Tortured Gate quest just wasn't satisfying. We also noticed that players continued to ignore Focus Mode in combat, and that they didn't have a good grasp of all the moves available in the game. Based on that feedback, we completely flattened the Choking Grounds and turned it into a giant graveyard. We then added several arenas to the area in place of the Caster quest. The arenas served as both a showcase for combat and as mini-tutorials to show players some of the depth of the combat system.

Once those changes were made, we put the Choking Grounds back through usability testing. The Choking Grounds went from the worst-reviewed section of the game to the players' favorite area.

WHAT WENT WRONG

1) SCOPE CREEP. The original design called for four-player co-op featuring all four Horsemen of the Apocalypse. The game was to be divided into "Overworld" and "Underworld" sections, and each of those sections was larger than the ultimate size of the final game. Clearly, our reach far exceeded

our grasp. Some features, like four-player co-op, were abandoned early in the project. Others, like the Underworld, remained a part of the game design until late in the production cycle.

Many of our scope problems can be traced back to the fact that we were a first-time studio that didn't clearly understand both the amount of work involved and the rate at which we would be able to hire people onto the project. We should have adjusted our scope more aggressively during pre-production. We didn't. Instead, we worked on multiple areas of the game at the same time trying to demonstrate our vision for each major section of the game. Much of that work was wasted when we ultimately had to abandon those areas of the game.

The scope changes adversely affected the team in a number of ways. Many of our artists and designers were frustrated and demoralized by having spent so much time on areas of the game that would never be seen by players. Moreover, quite a few people on the team were left with the impression that we didn't have a solid plan on how we were going to finish the game with the time and resources that we had remaining. The scope changes were necessary to complete the game on time and at the level of quality we wanted, but our management team learned a valuable lesson about making scope decisions as early in the process as possible.

2) WE WORKED IN THE WRONG ORDER. In our rush to demonstrate a first playable area of the game, we focused too much on the short term of how things looked in line reviews and demos and not on the long-term needs of the project. As a result, we were well into art production on a number of areas that didn't mesh with core metrics, like how far our main character War could jump or what gear he would have at that point in the game to help solve puzzle elements.

In hindsight, we should have spent a lot more time in pre-production working on all the gear items and abilities for War. We spent a lot of time retrofitting and iterating on levels to make them match the gear items and player abilities as those systems were finalized. On future projects, we plan to work on gear items and abilities first so we can use those metrics to drive the level creation process and polish every combat encounter.

Another major priority misstep we made was on the script. Joe Mad was both the lead character concept artist as well as the creative force behind the story for the game. We tried offloading some of that work by having another writer work on the script. The script required a number of rewrites because the story didn't translate well into a video game environment. The dialogue didn't reflect the personality of the characters the way we wanted it to, and the major plot points relied on gameplay elements that had not been fully fleshed out and areas of the game that were ultimately cut. The script wasn't finalized until January of 2009, less than six months before we were scheduled to submit the game to Sony and Microsoft. This caused no small number of problems with everything from the vendor we were using to help animate the cutscenes to the voice recording sessions we had planned for early 2009.

GAME DATA

PUBLISHER THQ, Inc.

DEVELOPER Vigil Games

PLATFORMS PlayStation 3, Xbox 360

RELEASE DATE North America, January 5, 2010

NUMBER OF DEVELOPERS 124

3) OUTSOURCING BEGAN TOO EARLY. We had a number of different scheduled ship dates for DARKSIDERS over the course of the project. Some of those dates were simply not realistic, but we had to do everything we could as a team to try to hit them.

One way that we tried to complete our content load on time was through outsourcing. We only had three concept artists on the team. Those three artists spent all their time creating assets for internal use. We decided to outsource the concepts in order to feed the outsourcing of the in-game assets we needed. When you combine that with our highly iterative art processes, you have the perfect storm.

Some of that work actually made it into the final game. Many of the church statues, for example, came from those vendors. But no one could effectively argue that those assets were worth either the money they cost or the time invested it took to bring them to fruition. We had to train every vendor on our art pipeline and our style guide. We used forums as our primary communication tool with each vendor. The written word is rarely an effective feedback tool for artists, and that problem is

exacerbated when you have a major language barrier between the studio and the vendor.

Our solution to the communication problem was to have our art director spend time every day painting over screenshots of the delivered assets to give feedback. That process improved the overall quality of assets that we received from the vendor, but it came at the expense of the art director spending less time with the in-house artists.

4) PRE-PRODUCTION SHORT-CIRCUITED. We did not have an effective pre-production period on DARKSIDERS. Our first few months on the project were focused on getting through the technical design documentation process. Once that milestone was out of the way, we had to quickly put together our first playable level to help demonstrate the core mechanics of the game. That effort was followed by a drive to show "something" at E3 of 2007. Instead of using the pre-production schedule to manage internal and external expectations, we allowed ourselves to be driven by perceived external needs.

To make things worse, getting our key hires in place took much longer than anticipated. By the time the core leads were finally assembled and on the project, we were already well past our appointed pre-production period.

5) NOT ENOUGH BENCH STRENGTH. A small studio working on a brand new next-gen IP without a proven track record is not exactly a recruiter's dream come true. We wanted to attract seasoned talent from every discipline to our studio. Unfortunately, our convictions about the type of studio that we would have in the future and the caliber of game DARKSIDERS would one day become did not always mesh with the realities of our budget limitations and the perception of the studio externally. The truth is that most of the people that we wanted to hire were firmly entrenched in other studios. Getting them would require money that we didn't have and the stability of a studio with a proven track record.

The good news is that we were eventually able to hire an incredibly talented group of leads and other key team members. The bad news is that we lacked depth at almost every position. For example, we didn't have a second-string UI artist because there was only one UI artist in the whole studio. We had two VFX artists on the project (one lead and one senior). We have talented modelers and texture artists on the team, but no one stood out as a clear potential candidate for art director or lead environment/character artist should something happen to one of those positions.

Now that we have started pre-production on our next title, we're making a concerted effort to improve our bench strength. We've started mentoring programs within the art department to help improve the overall quality of all the art we produce and raise the skill level of every artist on the team. We've changed some of our strike team processes to give more people the opportunity to demonstrate their leadership skills. We've also completed several internal postmortems and audits to evaluate those areas where we need to improve the most going forward, and we've developed strategic plans based on that information to help the studio continue to grow and improve over the next few years.

LET SLIP THE DOGS OF WAR

>> The final version of DARKSIDERS reflects everything we set out to achieve with the game. The world is smaller than we thought it would be, but all the core game mechanics are intact, and the story that we wanted to tell is fundamentally unchanged. We made mistakes along the way in every facet of game production, but we were able to overcome those mistakes because of the extraordinary commitment of the DARKSIDERS team and the willingness of THQ to grant us more time to help realize the game's full potential. The last 12 months of the project were brutal, but we've learned invaluable lessons as a team about how to hit our desired quality bar, recognize our limitations as a team, and how to communicate effectively internally and with our corporate counterparts. The engine, tools, pipelines, and team we've forged over the last four years of development are a great platform from which we plan to launch future successes.

TIMOTHY BELL *is the director of production at Vigil Games.*

DARKSIDERS FUN FACTS

War was almost a kid with a robotic arm, affectionately referred to as "Robot-arm-kid." Horseman of the Apocalypse ended up being cooler.

The original pitch was for all four horsemen featuring multiplayer co-op for up to four players. Then we realized we were crazy.

The original scope of the game was so large that three dungeons and an entire Underworld area were cut. Several key characters were cut from the story as well, including "The Hunter," a human who somehow managed to survive, becoming the last living human being. Even after making it through the Apocalypse, he never made it into the final game.

Vigil Games' very first office was 2,000 square feet with only one phone. There were 22 people in this space for a year before we moved to a larger 18,000 square foot office. After completing DARKSIDERS, Vigil moved into a state-of-the art 38,000 square foot office, which houses 125 employees.

Vigil spent an average of $11,500 per month on meals for the DARKSIDERS team during the last six months of game development.

DARKSIDERS' lead environment artist consumed 486 energy drinks in the last eight months. Enough to kill a large elephant. Or three small ones.

A life-sized cardboard Fabio was purchased during the development cycle. The extremely lifelike cutout was used in countless gags and pranks, but after enduring months of torment, cardboard Fabio would not make it to see DARKSIDERS launch. He was found beheaded in the Vigil parking lot shortly before the game launched. R.I.P. Fabio—you live on in our memories.

At one point, War's arsenal included a thrown grenade-like bomb, which let you fully control its trajectory, deflecting off walls and other surfaces. It was cut in favor of focusing on War's other weapons.

War's hair was one of the biggest artistic and technical challenges on the entire project. We considered completely removing his hair during the last weeks of development, until we were able to finally get it right.

The Demon Crowd noise in the Ashlands Arena is actually a group of devs cheering on the car parking garage roof run through filters.

Vigil co-owner David Adams chewed the corners off of 2,367 sheets of paper, including one programmer's resume. The programmer later spotted his half-eaten resume in Dave's trash can while at Vigil for his interview.

Four walls were busted in with feet and knees from late night shenanigans, including several attempts to wall run.

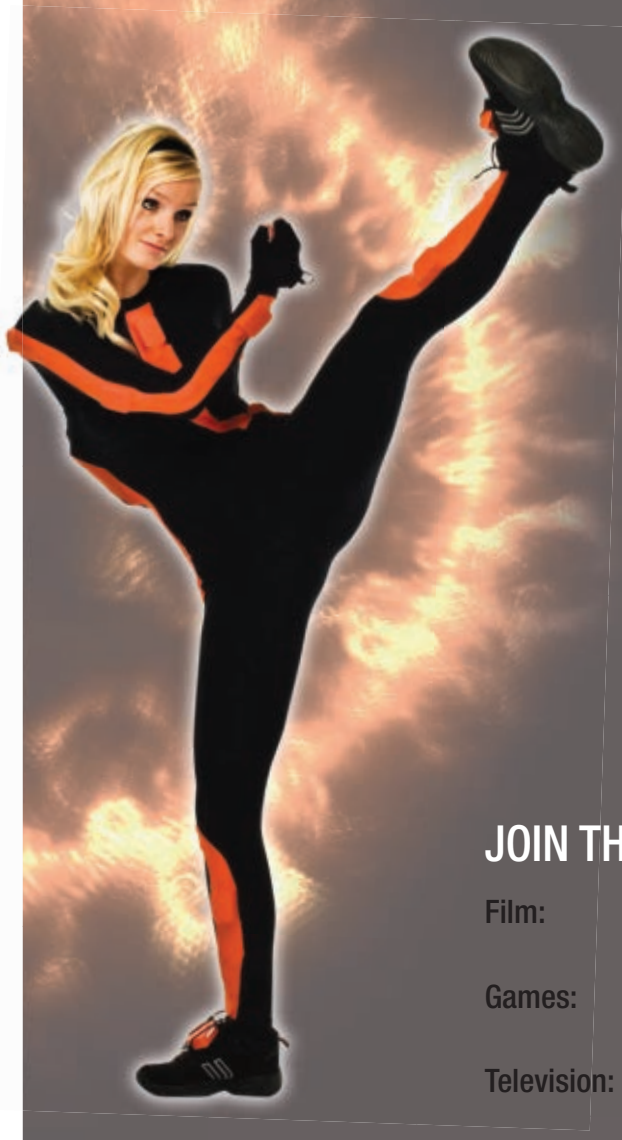
The script went through 17 revisions, including three versions by comics writer Joe Kelly, who was contracted to help out midway through development.

Making good on a promise, Joe Madureira got a DARKSIDERS tattoo from one of the members of the DARKSIDERS art team when the game shipped. The novice tattoo artist had only done a total of six tattoos, not including ones he had done on pig's feet and orange peels. Remarkably, no limbs were lost.

Freedom of movement

Previs and virtual moviemaking the quick and easy way

xsens



Xsens MVN
Cameraless Motion Capture

Free demo & workshop available
See www.xsens.com/demo

- Flexibility and quick turnaround times
- Unlimited (outdoor) capture volume
- Real-time, on-set previs
- Saves up to 80% post-processing time

SIGGRAPH2010

Xsens will launch breakthrough technology at Siggraph, come and see!

JOIN THE MOVEMENT!

Film:	Industrial Light and Magic, Sony Pictures Imageworks, The Third Floor, Image Movers Digital, 20th Century Fox
Games:	Electronic Arts, THQ, Sony Computer Entertainment, 2K, Insomniac, Gearbox Software
Television:	Animated Storyboards, Pipedream 3D, Simaestudio

Xsens Technologies B.V. / +31 (0)88 97367 00 / info@xsens.com / **www.xsens.com**

CREATING A STATE DRIVEN GAMEPLAY SYSTEM

MACHINA EX

LUIS SEMPÉ

IN THE PAST DECADE, GAMES HAVE STEADILY INCREASED in complexity, length and profundity. Game designs are increasingly ambitious—and rightly so—as the expectations and standards of gamers also continue to rise. Great games keep raising the bar, not only technologically, but also in terms of game design, mechanics, storytelling, and other creative areas. Achieving excellence means getting the right combination of one or several of these elements. To get there, we as programmers need to create the tools that will give content creators the power, flexibility, and freedom to achieve their lofty ambitions in a way that maximizes their efficiency.

For many years, complex interactions in games have been developed through the use of scripting languages. More recently, visual scripting systems have gained popularity, particularly since they are available in commercial game engines. These visual scripting systems are usually not gameplay systems, but general-purpose scripting languages in a more intuitive or familiar UI. Still, they do not address the problem of reducing complexity and maximizing efficiency for content creators.

In this article, I present the system I developed for DEUS EX: HUMAN REVOLUTION, which was designed to make it easier for our content team to move their designs to in-game prototypes, then iterate on those prototypes to develop final game content. The best lesson I learned in creating this system is that when we're making game creation tools, we need to stop thinking like programmers and really walk in our users' shoes.

Early in the pre-production phase, I designed the first iteration of the system. The core of the system was a powerful state machine (see Listing 1). The system's main flaw was in the data entry—it was overly technical and required a fair deal of hand editing to get right. The goal was to improve the content team's workflow, so the data entry was replaced by a graphical tool that allowed the team to visualize and edit complex gameplay interactions in a straightforward and consistent manner.

AN OVERVIEW OF THE SYSTEM

There are two components to the system: data and runtime. The data part refers to how data for the state graph is authored and how it is stored. The first operation we need is a translation of this data from its stored representation to a set of structures in memory from which we can create the runtime component or the state graph. To generate the data, we have a graphical UI tool that uses vector shapes to represent states and transitions (see Figure 1). This tool creates an XML file that contains all the information about the states, their data, and how they are connected to one another. We need to parse this XML and bring it into memory for the runtime component.

With the data in memory, we can begin following the graph as described by the data, creating runtime states that correspond to those in the data, and processing them to perform some action. Finally, they can be released when we're done with them.

At the core of the runtime component, there is a state machine whose job is to process the state graph one state at a time until the graph is complete or, if the graph loops, infinitely. The state machine itself does not have any knowledge of what each state is doing, it simply processes the states in the order it is told. The order may be sequential or driven by the states themselves. This means that states may tell the state graph which state the graph should go to next; if the state does not provide the graph with a destination, then we will need to look at what that state is connected to and follow one of its transitions. If no transitions are available, the graph will stop running.

The state machine will have a pointer to the graph that is running, and in each frame it can run three basic operations on states in the graph: `OnEntry`, `Process`, and `OnExit`. These operations are implemented by each state by deriving from `BaseState` (see Listing 2). `OnEntry` is called the moment the state is activated; it may validate some entry conditions or perform any necessary initializations. `Process` will be called every frame thereafter until it returns one of two possible values: `Processing` or `Done` (see Listing 3 for an example). When it returns `Processing`, the state machine will continue to the next frame, staying on the current state. If it returns `Done`, it will call `OnExit` on the current state and switch to the next state. On the next frame, this process repeats itself on the new state. If ever the current state is `NULL`, the state machine will know it's done and will close itself.

The state machine needs to be updated every frame in a non-blocking way. This is a particularly good setup because we can take advantage of it to perform some asynchronous processing within the state graph. For example, some of the conversations in our dialog trees require large amounts of data (such as animation, sound, or models) to be streamed before the conversation starts. To solve this, I create a state internally that loads all the data asynchronously, and while the data is loading, the state will just return `Processing`. Once the streaming is complete, the state will return `Done`. The conversation's state graph will then continue to run and I will be guaranteed that all the data I need is in memory and ready to be used.

LISTING 1
The heart of the system, this function needs to be called each frame and it will in turn process the graph's currently active state.

```
StateGraph::Status StateGraph::Process()
{
        // If the state has changed, enter the new state
        if ( m_previousState != m_currentState )
        {
                m_currentState->OnEntry();
                m_previousSate = m_currentState;
        }

        if ( m_currentState->Process() == Done )
        {
                // The current state has finished
                m_currentState->OnExit();

                State* nextState = GetNextState();

                delete m_currentState;
                m_currentState = nextState;

        }

        // If we don´t have a current state at this point, the state graph is done
        if ( !m_currentState )
                return Done;

        return Processing;

}
```

THE GRAPH DATA

There are several structures that hold different parts of the data required to generate the complete state graph, of which `StateGraphData` is the main definition (see Listing 4). It consists of two arrays: the first contains the states themselves, and the second, called a transitions array, describes how they are interconnected. A transition consists of the index to the source state and an index to the destination state. Optionally, it may also have some custom data that can tell states whether they may use those transitions at runtime.

Ideally, the data for this structure will be generated using a graphical editor such as the one in Figure 1. The editor saves this data into XML format, assigns each state a unique ID, and saves a list of transitions in which it binds the unique ID of the source state with the ID of the destination state. At runtime, when you open a state graph you can parse the XML and generate all the data the state graph will need to run.

The algorithm to parse the XML is fairly straightforward. You begin by counting how many states there are, and how many transitions. Knowing how many states and transitions you have, you can allocate them in memory. You should iterate first over all the transitions and store them into the transitions array—it's easier to do these first because we need to reference them from each state in the next step. Iterate then through every state, check the type of state, and allocate a data structure for that type. Next, copy the data into the newly allocated memory. Finally, count how many transitions that state has, allocate memory for its list of transition indices, and set the indices to the respective entries in the transition array.

I mentioned that you need to create a state's data based on the type—this can be done in many ways. For simplicity, I created a very straightforward factory that receives XML data that describes the state that needs to be created; the factory then returns an instance of the appropriate state initialized with the XML data received. (See code samples at see www.gdmag.com/code.)

TRANSITION DATA

The same idea can be used in the transitions between states themselves. We can create

LISTING 2

Class definition of a state. Some examples of states can be animation, sounds, and queries. You can add even more power by creating a script state which will execute game script.

```
class BaseState
{
public:

        enum Status {
                Done,
                Processing
        }

        BaseState(StateGraph* graph,
        StateData* data);
        virtual ~BaseState();

        virtual void   OnEntry();
        virtual void   OnExit();
        virtual Status Process();

        virtual bool HandleEvent(const
                GraphEvent& event);

protected:

        StateGraph*    m_stateGraph;
        StateData*     m_stateData;

};
```

LISTING 3

This is an example implementation of a state that gives the player a choice. It uses data stored in the state's transitions to determine which state the graph will go to next.

```
State::Status StateChoice::Process()
{
        // . . .
        bool playerChoice = GetPlayerSelection();
        for ( int i = 0; i < m_data.NumTransitions; ++i )
        {
                TransitionData_Choice* transitionData = static_cast
                        < TransitionData_Choice*>( m_data.Transitions[i].data);
                if ( transitionData->option == playerChoice )
                {
                        m_graph->SetNextState( transitionData->DestState );
                        return Done;
                }
        }
        // . . .
        return Processing;
}
```

some data structures for certain transitions, allowing states to examine the data for each transition and use it as a way to evaluate whether the state should use that transition. Imagine creating a state that pops up a small UI in which the player is prompted to make a choice, and he can choose "yes" or "no." Once the player makes his choice, the state can iterate over all the transitions it has in its list and check the data within. If any transition matches the choice the player made, the state will take the destination node for that transition and call `m_graph->SetNextNode(destinationNodeIndex)`, then return Done (see Listing 3).

THE STATE GRAPH

>> At the heart of the system is the StateGraph class, the state machine that will process each state as it receives it. The logic is fairly straightforward. We start by calling the Start function, which will set and initialize the current state. If the current state has changed since the previous frame, it means that this is a new state and we should call OnEntry. We continue by calling Process on the current state, and we will continue to do this each frame until the state returns Done, at which point we will call OnExit on it. We will then use a helper function to determine which state the graph should go to next. Finally, we can destroy the current state and set it to the next state (returned by GetNextState). If the current state is NULL, then the graph no longer has anything to do and needs to close down, and so would return Done. Otherwise, a new frame would start and the process would repeat itself with the new state (see Listing 1).

The helper function GetNextState is responsible for instantiating the new state. What's interesting is where the next state comes from. In the normal case, each state has outbound transitions that lead into other states. If we only have one transition, then the choice is obvious. We get the current state, find the transition, and create the new state from the destination index in the transition. If we have multiple transitions, and there is no logical way to choose one, in my system, one of the available transitions will be selected and used. This is a design decision that you should discuss with your content creators, though.

A state can also dictate the next state that should be used on its own. For example, you may have a state in the graph that gives the player three choices, and the choice the player makes will determine which state the graph continues to. In this scenario, states have access to the graph and they can call `m_graph->SetNextState( targetStateIndex );`. This sets a member variable in the graph to the index of the target state, and the GetNextState function I mentioned before checks to see whether this has been set, and returns this state if it has.

CREATING STATES—THE STATE FACTORY

>> To create the states at runtime from our data, we need a mechanism that will instantiate and initialize the specified states in code. For this, we have a StateFactory—a static class that takes the state's data as input. The type of state provided (in the data) is used to determine the appropriate C++ object to instantiate. Listing 5 shows an example that uses an enum value to specify the type of the state to create.

You have two choices for state creation. The first is to iterate over all the states in the array and create them, keeping a list of all states ready in memory. The other is to call StateFactory::Create as needed, essentially creating states right before you need them, and they'd be released right after you're done. The first approach has the benefit that you can create the complete state graph while parsing the XML data saved by the editor. The second approach requires you to keep a copy of the data in memory so it's available when we need to create a state. I've been working with the second approach; states are created by the StateGraph when it starts and then when it's time to change

FIGURE 1 Designers create complex gameplay interactions by placing states in a graph editor and connecting states with transitions. Each state represents an action or a query and allows for logic to be created within the state graph when used in conjunction with the data in the transitions.

LISTING 4
This data structure will hold the entire definition of the state graph and needs to be populated from a data source at runtime.

```
struct StateGraphData
{
        unsigned int numStates;
        State* stateList;
        unsigned int numTransitions;
        Transition* transitionList;
};
```

states, the data for the state graph is already loaded in memory.

STATES

States are the runtime objects in which the actual work is done. They are created using the state type and the data that comes from the graph editor tool. States all derive from a BaseState class which may or may not be an abstract class. I chose not to make it abstract and to instead provide some logging functionality at the base class, allowing me to record the order in which a state graph is traversed.

You can see in Listing 2 that states have four main functions; OnEntry will be called when a state is first activated, followed by Process, which will be called each frame as long as it returns Processing. Finally when a state returns Done, OnExit will be called. The last function is HandleEvent, but we'll talk more about this a little bit later. Refer to Listing 6 for an example of an event.

The states you create depend entirely on your goals and what you need to achieve, and also the features that you wish to expose to your content team. Some useful states are actions (setting game variables, rewarding the player, mission objectives), queries (inventory queries, game variables, objective progress), script (allows game scripts to be executed directly from within the state graph), and choice (displays a UI giving the player a choice). Each state has a custom data structure—see Figure 2 for an example of the type of settings that are available for an animation state.

EVENTS

Sometimes it is useful to let the state machine tell the graph that some event has occurred. This allows states to react in a certain way when events occur. For example, if the player aborts the action, or the NPC using this state machine is destroyed, then the state machine will need to be aborted or shut down.

To support this, the StateGraph class will have a public function Notify to which you can pass an event. A straightforward approach is to have an enum of events, and pass in the enum you need.

When m_runningGraph->Notify() is called, it will call the virtual function HandleEvent for the state that is currently being processed. States can then choose to handle only the events they are interested in. For example, say we have a state that is playing a sound, and the event "Abort" is sent by the state machine. The Process function in StatePlaySound may detect that the sound has stopped playing and return Done at that point. The userData parameter can be used in case you need to pass information associated with that state.

LISTING 5
The idea of the state factory is to bind the data side representation of a state with the object in code that will do the actual work.

```
State* StateFactory::Create(StateGraph* graph, StateData* stateData)
{
        switch (stateData->mType )
        {
        case StateType_Action:
                return new StateAction(stateData, graph);
        case StateType_Query:
                return new StateQuery(stateData, graph);
        }

        assert_msg(0, "StateFactory cannot create unknown state: %d", stateData->mType);

        return 0;
}
```

LISTING 6
Example of the PlaySound state handling an abort event.

```
bool StatePlaySound::HandleEvent(StateGraphEvents event, void* userData)
{
    if ( event == StateGraphEvents::Abort )
    {
            if ( m_sound.IsPlaying() )
            {
                    m_sound.Stop();
                    return true;
            }
    }

    return false;
}
```

SUBGRAPHS

I mentioned before that the state machine doesn't have any knowledge of what the graph is currently doing. We can take advantage of this to swap out the graph that is currently running with another one altogether. Once this "subgraph" is done, we can restore the graph that was previously running and the state machine will continue processing it as if nothing happened.

To implement this, remember that all states have a pointer to the state machine. Any state or dedicated subgraph state may call a function on the state machine, in which it sends a pointer to the graph you need the state machine to process. Listing 7 shows SetSubGraph which may be called from within any state. When the state machine receives the data for the subgraph, it will initialize a new graph m_subGraph with it, keep a pointer to the graph that is currently being processed, then replace that graph with a pointer to m_subGraph. On the next frame the state machine will do the

LISTING 7
Switching the running state graph to a subgraph means initializing the subgraph to the data of the desired state graph, then switching the pointer of the running graph.

```
SetSubGraph(StateGraphData* data)
{
        m_subGraph.Start( data, this );
        m_previousGraph = m_runningGraph;
        m_runningGraph = &m_subGraph;
}
```

LISTING 8
The modified StateGraph with subgraph support.

```
StateGraph::Status StateGraph::Process()
{
        if ( m_runningGraph )
        {
        StateGraph::Status status = m_runningGraph->Process();

        if ( status == StateGraph::Done )
        {
                if ( m_previousGraph )
                {
                        m_runningGraph = m_previousGraph;
                        m_previousGraph = 0;
                        m_runningGraph->Notify(SubGraphDone);
                                return StateGraph::Processing;
                }

                return StateGraph::Done;
        }

        return StateGraph::Processing;
        }
}
```

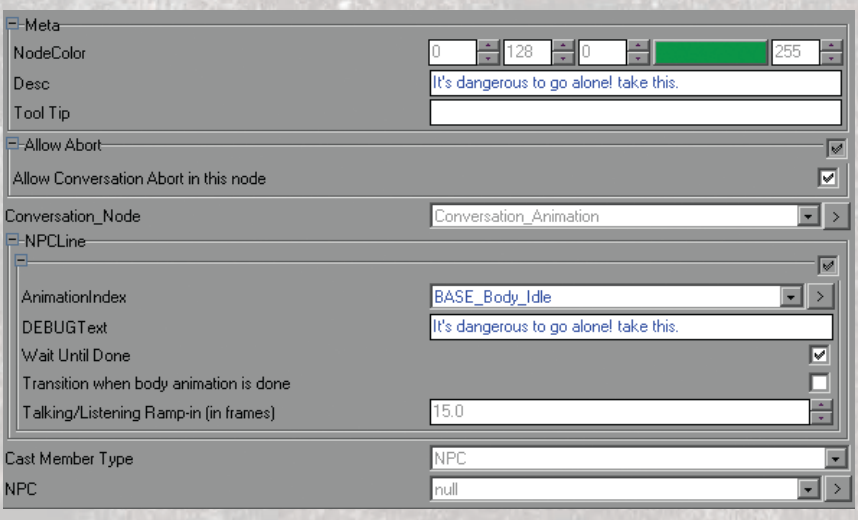


FIGURE 2 An example of the data entry for an animation node within a conversation graph.

same thing it was doing before, except that the graph will now be different.

We keep a subgraph object `m_subGraph` which we initialize to the subgraph data we just received, and keep a pointer to the graph that was previously running so we can return to it and keep processing it when we're done. Finally, we switch the graph that is currently running.

There is only one thing that we need to change in the state machine—when a graph returns Done, we need to verify whether we had a previous graph. If we did, this means we just finished processing a subgraph, and need to go back to processing the previous graph. This is shown in Listing 8. To restore the original graph, we can assign `m_previousGraph` to `m_runningGraph` to make it the current running graph again, then clear `m_previousGraph`. One nice thing to do after restoring a graph is to send it an event notification `m_runningGraph->Notify( SubGraph_Done );`. This can be useful in allowing the state from which the subgraph was launched to continue processing as it was doing before the graph changed. Most likely, this means the state will return Done and allow this graph to proceed. Listing 8 shows the state graph Process function that supports subgraphs.

resources

Gregory, Jason Game Engine Architecture

Richter, Jeffrey, "Concurrent Affairs" series, http://msdn.microsoft.com/en-us/magazine/cc501041.aspx

USE CASE—CONVERSATIONS

>> By far, the largest feature in DEUS EX: HUMAN REVOLUTION to take advantage of this state graph-driven approach are the conversations. This approach worked remarkably well to bring all the elements of a conversation together. Designers could play animations and sounds on characters or the player, perform branching based on choices the player makes, perform actions such as taking items from a character's inventory to give to the player and vice versa, rewarding the player, giving the player mission objectives, and much more.

One of the main features of this system is that it can be easily extended or customized, so in the case of conversations I derived `ConversationStateGraph` from `StateGraph` and also created a few custom states that would only be visible in the context of a conversation. We wanted to be able to support conversations between the player and multiple characters, or conversations between several characters without the player. This can be done by adding a "cast" of characters to the state graph data, and then adding a field to all the states where you can choose which actor you wish that state to be executed for.

The workflow for creating conversations was also quite efficient; given that the tool saves all the data using XML, this meant that level designers were able to place a rough prototype of a conversation in the game using text provided by the writers. Animators could go in, find the animation states, and plug in which animations should be used. Sound designers could simultaneously set the correct sounds that need to play in any sound states, and at the end, all their changes would be merged and a conversation would come to life.

DEUS EX MACHINA

>> I believe that as programmers, we should strive to treat our content creators as first-class citizens. They are the creative power at the front lines, often working long hours struggling with complex procedures and difficult-to-use tools. Making them more efficient in their daily work will increase the quality of our games, and that makes everyone happy!

LUIS SEMPÉ *is extremely proud to play a part in the history of DEUS EX and to work alongside the team at Eidos Montreal. He got his start in game programming working on several MEDAL OF HONOR titles at Electronic Arts.*

PIXOLOGIC ZBRUSH 3.5 R3

REVIEW BY ADAM MAY

ZBRUSH HAS BEEN AROUND FOR A while now and has become a staple in many an artist's workflow facilitating creation of fast and efficient high-poly organic models. With the recent release of Zbrush 3.5 r3, Pixologic once again shows its dedication to continually building up this amazing tool, expanding its scope to enable good artists to make great art.

DIFFICULT TO LEARN, EASY TO MASTER

» There are a few things about Zbrush that make it very different from most 3D programs. For starters, it wasn't originally designed to be a 3D program. Zbrush started out as what Pixologic coined 2.5D, which was basically 2D with faked depth, lighting, and material information. Because of this, the functionality and workflow can at times be awkward or frustrating to people who have a long history with other 3D programs. For instance, your workspace isn't a 3D view port, it's actually a resolution-dependant image, so if you zoom in, your work becomes pixelated. To zoom in on your model, you must scale it up instead. Following this example, a lot of its tools and controls are more akin to 2D software than 3D.

There are other little nuances throughout that seem odd at first, but soon become natural parts of your workflow. Thankfully, there are massive amounts of resources available to help you as you learn. Pixologic has done an amazing job of building and supporting a community around its software and constantly providing updates with tutorials and videos. The company's web site, www.pixologic.com, provides tutorials to help new users get started, and also shows new features and innovative techniques.

Another great source for finding resources and inspiration is www.zbrushcentral.com. The forums there have a great, frequently unnoticed feature called "thread gallery," which quickly and conveniently sets all images submitted to the thread on a single page for easy browsing. Zbrush tutorial videos can also be purchased from Gnomon at www.thegnomonworkshop.com. There are tons of other great sites as well, and with a little searching, you can find the answer to any question or problem you might run into.

Zbrush can be pretty confusing at first. Once you get over that initial hump and have the system down, it's amazing how streamlined and efficient the whole thing works, and how quickly you'll be blasting through your day doing things you hadn't even thought were possible with any other program.

THE INTERFACE

» Zbrush's interface is designed to be extremely customizable. Tools and palettes can be docked on top, bottom, or either side of the workspace and almost everything has been assigned a shortcut key. The best part is, with the release of Zbrush 3, Pixologic gave us the ability to quickly and easily assign our own shortcuts by simply holding ctrl and clicking on a tool or function. You will probably want to assign your own shortcuts because there are so many tools and functions that Pixologic is running out of easy shortcut combinations and some of them can be a little awkward. Pixologic also added the ability to view a description of any button by simply hovering the cursor over it and holding the ctrl key.

Zbrush software was developed to be used with a pen and tablet and not doing so would really be a disservice to yourself and the program. The setup is so perfect that the only time you'll ever find yourself needing to put down the pen is to name your files. Every brush is set up to account for pressure sensitivity as well, so a good tablet and pen really are a must.

The one real gripe I frequently hear about the interface, which I myself share, is the confusing naming used in title menus. They don't always seem to fit the functions they hold. For instance, the "document" menu on the title bar controls all your view port adjustments, but the perspective and 3D grid controls are all inside the "draw" menu. And the fact that several menus inside of different palettes have the same name can be confusing as well, such as having a "Layers" menu inside both the "Document" panel and the "Tool" panel. Once again, these all stem from Zbrush being a 2D program handling 3D objects. But the reality is, with all the features, tools, and functions Pixologic has jammed into this one program, there's bound to be some confusion.

TOOLS AND FEATURES

» Pixologic likes to boast that Zbrush has been developed by artists for artists, so the company tends to laugh in the face of convention. At times, it seems as though Pixologic refuses to even acknowledge how other software works. Zbrush developers try to have you approach modeling, in general, from a whole different direction.

Pixologic enables and encourages artists to approach 3D modeling like clay sculptures. Most mesh creation in Zbrush is either done using Zspheres (the equivalent of blocking out models with clay) or using its re-topology tools to recreate a cleaner mesh around your already existing model. Zbrush also fully supports the use of models created in external programs, which can be imported as an .obj.

After you have a model in Zbrush and it's time to start sculpting, this is where the fun starts and Zbrush really shines. As you work with the mesh, you create new, higher subdivisions as you need more and more detail. The sheer number

Pixologic ZBRUSH 3.5 R3

PIXOLOGIC
336 W 31st St
Los Angeles, CA
90007-3806
www.pixologic.com

PRICE $595

SYSTEM REQUIREMENTS
Windows XP SP2 or newer, 32 bits or 64 bits (ZBrush 3.5 is a 32-bit application, but can use up to 4 GB of system RAM). Pentium D or newer (or equivalent, such as AMD Athlon 64 X2 or newer) with optional multithreading or hyper-threading capabilities; 2 GB RAM recommended for working with multi-million polys; 1280x1024 monitor resolution set to Millions of Colors; Wacom or Wacom compatible pen tablet.

PROS
1 Incredibly powerful tool for sculpting and texturing millions of polygons in 3D.
2 Great technical support and lots of tutorials and reference backed by a huge user community.
3 Amazing price for such a powerful tool that's continually being updated and improved.

CONS
1 Steep initial learning curve can be frustrating to new users.
2 Occasional stability issues cause unexpected and painful crashes.
3 Some scaling and translation issues cause some problems jumping back and forth between 3D applications.

of polygons you can have in your workspace without slowdown still amazes me. I will frequently brush detail on a 6–8 million polygon model without even a stutter in my stroke. Zbrush unfortunately doesn't support normal smoothing, so you have to get accustomed to your models looking faceted in the lower subdivisions, but as you subdivide higher and higher, the polygons get so small you won't even notice.

People generally tend to think of Zbrush as only an organic high-poly modeling tool, but it really is much more. Zbrush also has a lot of great 2D brushes and tools that work well for making spectacular art on their own. When used with the 3D tools though, they really allow you to do amazing things, such as paint diffuse, specular, bump maps, and materials on separate channels—directly on your model. The 2D brushes can even be used to paint deformations and details on your model through the use of a tool called Projection Master.

Pixologic also includes tools for baking out normal, cavity, and displacement maps. And Zbrush can now even bake out ambient occlusion maps and be used for masking, which has several great applications for texture creation and modeling.

WHAT'S NEW

>> For those of you already using Zbrush, 3.5 r3 features a lot of cool new and updated tools to work with. One of my favorites is a new set of brushes called Polish Brushes. They're basically a flatten brush and smooth brush combined. The brush smooths normally, but the more pressure you put on a brush stroke the more chiseled and flattened it makes the surface. This works especially well for getting out those small waves and imperfections you often get while working in the higher subdivisions.

The 3.5 version adds even more new brushes and extra settings for controlling the way they work, including a set specifically designed to aid in making hard surface modeling much more feasible in Zbrush. Since it's still more akin to carving surfaces, it works great for creating those difficult-to-model curved and flowing edges, but it's still generally easier and quicker to use an outside package for building your standard hard-surface models.

Probably the most significant addition is in Zsketch. This big step forward comes in the form of an update to Zspheres simply called Zspheres II. The feature further emulates the feeling of working with digital clay. You use traditional Zspheres to create a posable armature for your model and then build up form and volume by laying down more Zspheres over the top. The Zspheres are then converted into a sculptable mesh with Zbrush 3.5's new and improved Adaptive and Unified Skin tools. It's a dream come true for anyone who's ever wanted to be able to sketch in 3D.

Pixologic also integrated many of its most useful plug-ins, such as Subtool Master and my personal favorite, Decimation Master, which is by far the best optimization tool I've used. It can break your six million polygon models down to a few hundred kilobytes without you even being able to see the difference.

There's still more though. Pixologic has added ambient occlusion tools, and updated Polygroups and masking. There's also a great new Quick Sketch tool for quickly sketching out your models in 2D, and the added ability to export HD Geometry in your normal or displacement maps. This is on top of a bunch of new tweaks and updates to the already huge list of great tools.

Basically, Pixologic took everything it had and made it better, adding a bunch of new features that were badly needed and a couple of crazy ones that no one knew they needed but now can't do without.

RAINBOWS AND ROSES

>> Zbrush is undoubtedly one of the most innovative and powerful tools on the market. Without it, I wouldn't be able to do my day-to-day work, and I frequently find myself trying to quickly rush through the other steps to get back to Zbrush. But all that innovation doesn't come without its price.

Besides Zbrush's initial learning curve, it also has some stability problems. The program occasionally crashes unexpectedly for no apparent reason, so save frequently and incrementally. There are also a few quirks when working between Zbrush and other 3D programs, namely with scale. Zbrush doesn't have a standard unit size, so scale is completely subjective per mesh. There are workarounds for the scale issues and things should improve with the promised GoZ plug-in for future versions of Zbrush, which will allow you to seamlessly jump back and forth with other 3D applications. But until then, you will frequently find yourself adjusting the scale and placement of your meshes once you return to an external application.

ART BY ADAM MAY

PRICING

>> Pixologic has set what I wish would be a new precedent in pricing and upgrades. Pixologic has been giving free upgrades and updates to all Zbrush users since version 2. So for one initial purchase price of under $600, Zbrush users have been getting all the benefits of years of Pixologic's hard work developing and supporting their software for free. Compared to the enormous cost of most 3D applications and the yearly updates that cost more than the full purchase price of Zbrush, it's hard to understand how Pixologic keeps afloat.

WORTH IT

>> So is Zbrush worth it? I have used other 3D sculpting tools, and they have generally been easier to just pick up and jump into, but none of them have come anywhere near the power and versatility of Zbrush. It truly allows the artist to create anything he or she can imagine. And when you take into consideration the price, Zbrush is hands down one of the best deals out there.

ADAM MAY *is employed by Gearbox Software and has been involved in creating characters for several next-gen games, and was lead character artist on titles such as* BORDERLANDS *and the upcoming* ALIENS: COLONIAL MARINES. *Email him at* **adam.may@gearboxsoftware.com**.

13-15 JULY 2010
develop in Brighton

Early Bird Rates until 7 June

be inspired

Excellent Speakers. Great Networking. Top Location.

Develop in Brighton – the main event for European developers – is back with a vengeance! In celebration of our fifth year we can promise you a stellar line up of speakers, a choice of over 80 high quality sessions and fantastic networking opportunities with more than 1,200 international developers over the three days.

Opening Keynote

KEYNOTE

Ray Muzyka
Group General Manager, BioWare Group and Senior Vice President, Electronic Arts, and Co-founder and CEO, BioWare

Greg Zeschuk
Group Creative Officer, BioWare Studio Group, Electronic Arts Vice President, and Co-founder of BioWare

The superstar developers and founders of **BioWare** the studio responsible for the hugely successful titles Mass Effect and Dragon Age, will be making a rare appearance in Europe as they give this year's opening keynote.

Other Keynote Sessions

KEYNOTE EVOLVE

Traditional Games Breaking into Social Networks: A View from the Frontline
Louis Castle, CEO, Instant Action Inc.

KEYNOTE AUDIO

The Highly Anticipated Sound Revolution
Adam Levenson, Director, Central Audio and Talent, Activision Publishing

KEYNOTE EVOLVE

Gamification: How Games are Everywhere
David Helgason, CEO, Unity Technologies

KEYNOTE DESIGN

The Future IS Controller-Free Games and Entertainment
George Andreas, Creative Director, Rare

Develop sessions include:

Tim Schafer
Developer legend Tim Schafer is President of Double Fine Productions, creators of Psychonauts and Brütal Legend.

- **3DTV: The Next Dimension in Game Graphics**
  Mick Hocking, SCEE
- **All the Right Moves: Grasping the Possibilities of Motion Controllers**
  Andrew Oliver, Blitz Games Studios
- **Enslaved to the Story: When Ninja Theory Met Alex Garland**
  Tameem Antoniades, Ninja Theory

EVOLVE sessions include:

- **Casual Game in Every Pocket**
  Dave Bishop, Pop Cap
- **An Android Game Post-Mortem**
  Chris Pruett, google
- **Hats, Loops and Levers: Experiences of a Small Studio**
  Simon Oliver, Hand Circus

Expo

Running alongside the Conference is the Develop Expo, which is **FREE** for visitors to attend and brings together some of Europe's most innovative companies from every sector of games development.

The Main Event for European Developers

www.developconference.com

International Media Sponsor: develop
Media Sponsor: casualgaming.biz
Media Sponsor: AUDIO MEDIA
Media Sponsor: gamesindustry.biz
Media Sponsor: gamedeveloper
Media Sponsor: IMAGINE
Media Sponsor: GAMASUTRA The Art & Business of Making Games

Member Discounts: TIGA, igda
Partner: develop Industry Excellence Awards
Organised by: tandem events

DECAL TESSELLATION

REALISTIC DAMAGE ON THE FLY

ONE THING THAT MAKES GAMES FEEL REALISTIC IS THE ABILITY TO INTERACT WITH THE ENVIRONMENT. If a player hits or shoots something, they expect it to change in some way. Most games will at least have some sort of visual feedback, such as a decal texture or a particle system showing smoke or dust. Some games may have a few breakable objects, like glass bottles. Ideally, you would like everything in the scene to change its shape based on the damage the player does to it. Games rarely do this though, primarily because of the complexity of modifying geometry in real time and the extra preparation necessary by artists to create damage geometry. Still, a player can easily lose the sense of immersion when his RPG is able to destroy a heavily armored tank, but only leaves a scorch mark when it's shot at a wall.

Decal tessellation is a simple solution to this problem. The idea is that you can take a displacement map texture, project it onto geometry—just like a decal—and tessellate the geometry in real time so that the displaced geometry looks like physical damage to the object. Figure 1 shows an example of decal tessellation. The ability to tessellate in real time is what enables this technique. Both Direct3D 11 and OpenGL 4.0 have APIs for hardware tessellation, and at least two of the current graphics hardware vendors have GPUs that are capable of doing tessellation. Since tessellation is a required feature in Direct3D 11, you can expect most, if not all, future GPUs to be suitable for this technique.

HARDWARE TESSELLATION

>> Before going into too much detail about the implementation, I should first go over some of the basics of hardware tessellation. Figure 2 shows the Direct3D 11 pipeline which includes the tessellator stages. At the top of the pipeline is the vertex shader. Usually the vertex shader is responsible for all vertex transformations, including the transformation to clip space prior to rasterization. With tessellation, the input mesh is called a control cage or control mesh, since it's a mesh of patches made up of control points. The simplest patches are just triangles. The vertex shader can be used for transforming the control mesh, as in the case of animation. Otherwise, it just passes the control mesh vertices onto the next shader stage. After the vertex shader comes the hull shader. The hull shader has two parts that run in different phases. The control point phase runs once per control point, and the patch-constant phase runs once per patch. The control point phase can be used for doing further transformations of the control points, or it can just pass the control mesh vertices to the next stage. The patch-constant phase hull shader is responsible for setting tessellation factors. These factors are set for each edge of the patch and also for the inside of the patch. They determine how much to tessellate (break up into triangles) each part of the patch. Tessellation factors can have a huge impact on the rendering performance. After the hull shader comes the tessellator stage. The tessellator is a fixed function stage that generates the new vertices that make up the tessellated patch. These new vertices are then passed to the domain shader which gets called once per tessellated vertex. The domain shader is where the vertices of the patch get evaluated based on the type of patch. For example, if the patch uses displacement mapping, the domain shader fetches the height values from the displacement map and translates the vertices accordingly. If the patch is a parametric surface patch, such as a Bézier patch, the domain shader calculates the necessary mathematical function to transform the vertices to their position in world space. After evaluating the patch surface, the domain shader transforms the vertices into clip space, as the vertex shader would do for non-tessellated rendering. The rest of the pipeline executes the same as non-tessellated rendering, with vertices passed to the geometry shader followed by the pixel shader which gets executed for each pixel.

FIGURE 1 **Decal tessellation. The model on the left has no decals. The model on the right is the same model with tessellation decals. Notice how the silhouette is changed.**

CURRENT TECHNIQUES FOR RENDERING DAMAGE

>> Decals have traditionally been used to represent damage because they are simple to use, and can be applied anywhere. This is important because in order to give the feeling of a realistic environment, you don't want to limit damage only to locations that you think the player will try to destroy. Decals are typically implemented by finding the location of the damage and creating geometry that can be placed over a portion of an object, and texture mapping that geometry with a texture that looks like damage, such as a scorch mark. In its simplest form, the decal is a quad or single triangle that is placed very close and in front of a flat surface like a wall. Depth bias may be needed to prevent z fighting. The problem with standard decals is that they are flat; they simply add color to the area to represent the damage. This is particularly noticeable when viewing them from an angle where incorrect silhouettes and lack of parallax makes the flatness of the decals look unrealistic. An improvement on flat decals is parallax mapping and parallax occlusion mapping, which add parallax to the decal but have artifacts when viewed at low angles. Also, the silhouette of parallax mapping is still flat, since the geometry is unchanged. (See "Parallax Mapped Bullet Holes," May 2006.) A similar technique to decal tessellation is "detail tessellation," which is an improvement to parallax occlusion mapping because it uses the height field of parallax occlusion mapping to create a realistic silhouette. Decal tessellation is a more general form of this technique since it can be applied anywhere.

True destructible geometry either needs to be created by an artist by hand, or requires an engine capable of modifying existing geometry in real time. Damaged geometry can be created in advance by an artist to replace the original geometry when damage occurs. The swapping of original geometry with damaged geometry can be made less obvious by adding a smoke particle system. There are a couple of drawbacks to this kind of special casing. One problem is that the appearance of the damage is repeated because the artist can only create so many damage models. Unreal Engine 3 simplifies damage geometry by allowing artists to create destructible geometry with the Fracture tool in UnrealEd. These objects actually fracture into pieces, giving the appearance of damage that corresponds to the force being applied to the object. In practice, only a limited number of damaged objects can be created due to time and memory constraints. It may not be worth creating damaged geometry for every object in the scene if the player is not likely to ever shoot at them. But it only takes one time when a player decides to shoot at something that a game developer didn't predict for the reality of the simulation to break down. A more advanced destructible geometry is one where the engine calculates new geometry based on the situation. One example is the Geomod engine used in the RED FACTION games. This engine is able to calculate new geometry appropriate to the destruction using Constructive Solid Geometry (CSG) techniques. Decal tessellation doesn't go that far, but is much simpler to implement and can be added almost as an afterthought to the engine design.

DESIGN AND IMPLEMENTATION

>> There were a few important goals I had in mind when designing this technology. First, I wanted this to be something a programmer could add to the game without too much work from artists. With this technique, an artist is really only needed for the creation of displacement maps, normal maps, and any other textures associated with the decal. Another design goal was to be able to tessellate anywhere. The idea of decals fits well with this because they are placed onto or associated with geometry and don't need to be a property of the geometry in advance. Related to this is the requirement that few restrictions be placed on the existing geometry. For example, neither the format of the vertex data nor the parameterization of the texture mapping should be a limiting factor.

Meeting these design goals meant overcoming a few obstacles. Probably the most difficult of these was the mapping of the texture coordinates. Many 3D objects used in games don't use texture coordinates in a way that's well suited for displacement mapping. Consider the use of texture atlases. Texture atlases are good for saving memory and re-using textures because they combine many textures onto one texture map. The artist then maps parts of the texture map to parts of the geometry, rather than mapping the whole texture continuously to all of the geometry. This

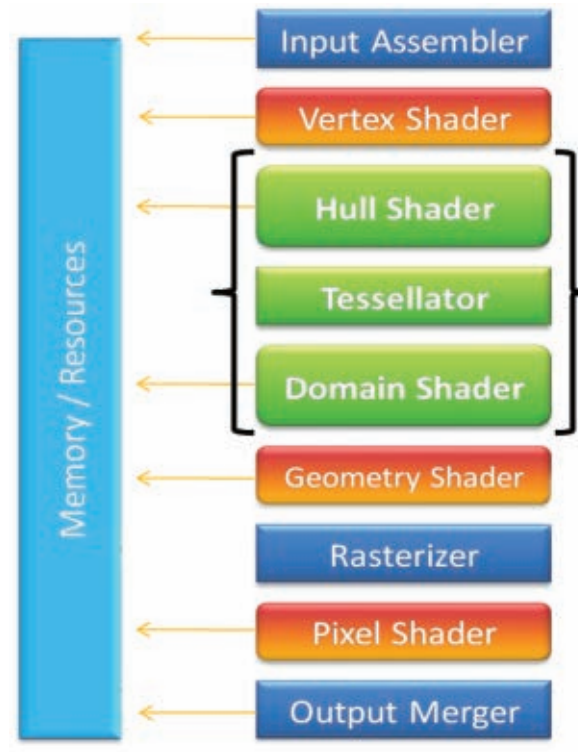


FIGURE 2 The Direct3D 11 pipeline. The blocks in green are the stages used for tessellation.

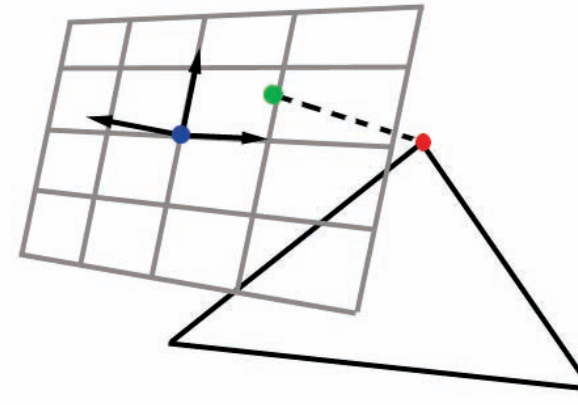


FIGURE 3 Domain shader projection of a vertex onto the decal coordinate plane is shown. The red dot is the tessellated vertex location. The blue dot is the location of the decal in world space. Note the orientation of the decal is specified by the 3 vectors. The green dot is the projection of the vertex onto the coordinate plane formed by the two vectors perpendicular to the normal of the decal.

is a problem for displacement maps because it introduces discontinuities in the mapping. To compensate for this, the displacement map would need to be laid out like the texture atlas, but even this would have problems because some areas of the atlas have a denser mapping than others. This isn't such a bad problem for colors, but when using displacements, it can introduce different displacements along triangle boundaries which can cause visible cracks. What we really want is a mapping that is both uniform and continuous for the whole displacement map decal. A uniform mapping could be created by pre-processing the geometry, but that would mean saving it out and changing the vertex format, because you would need to add a new set of texture coordinates. A more flexible approach, which allows you to use the existing the geometry, is to generate texture coordinates in real time during rendering.

To generate texture coordinates, we can use the decal location and the orientation of the decal. Decal locations and orientations are determined by the game code, executing on the CPU. For example, if the player shoots a gun, a ray can be cast from the gun in the direction that it's aimed to see what object it intersects with. Once the intersection is found, the location is stored in a constant buffer. We can also save the orientation of the decal by creating a coordinate space with the normal facing in the opposite direction of the ray. The orthogonal tangent vectors are then created based on this normal direction. The three coordinate space vectors are stored in a constant buffer along with the 3D world space location. Multiple decal position and orientation information can be stored in this constant buffer in a ring buffer fashion. The domain shader uses this constant buffer to generate the texture coordinates for the displacement map. For each of the decal locations, the domain shader can test to see if the tessellated vertex is within the radius of the decal. Decals can have a global radius or a unique radius for each decal passed to the domain shader through a constant buffer. If the vertex is within the decal radius, then texture coordinates are created by taking the vertex location and transforming it into the decal's local coordinate space as defined by the location and orientation vectors (see Figure 3). The u,v texture coordinates lie on the decal plane, and are the transformed x,y coordinates of the vertex. The z coordinate can be dropped since we are projecting it onto the plane. Since the z coordinate is unused, we can use it as a flag to indicate if this is a valid texture coordinate. This is important because the pixel shader will use this information to determine how to do the lighting. The pseudo code in Listing 1 shows the loop in the domain shader that checks for decals within the vicinity of the vertex.

Lighting the decal can be done any way you want, but using a normal map is an easy way to get good results. A normal map can be created by an artist that matches the bumps and contours of the displacement map. You can use this as a replacement for the normals of the displaced geometry. The original normals of the displaced geometry are not valid after the displacement of the vertices because the surface normal changes. Unless you re-orient the vertex normals, you won't get correct results. Normal mapping avoids having to calculate how the normals change with displacement because you use the normals from the normal map instead of the interpolated vertex normals. The pixel shader uses the z coordinate of the generated texture coordinates to determine if it needs to use the decal normal map, or if it needs to do lighting as it would without decal tessellation. The pseudo code in Listing 2 shows the pixel shader code that gets the normal for decal tessellation.

OPTIMIZATION

>> Objects that have decals need to be rendered with tessellation. With Direct3D 11, rendering with tessellation is as simple as binding the hull and domain shader stages to the pipeline, and specifying an input topology that is a patch. A naive implementation would just tessellate the entire object even if it had one small decal tessellation applied to it. This can severely impact performance, since the decal may have a lot of detail in it, and may require a high tessellation factor to look good. A necessary optimization is to adaptively tessellate only the patches that are affected by the decal's displacement map. Figure 4 shows a wireframe view of this optimization where you can see that only the affected patches are highly tessellated. To determine these areas, the patch-constant phase of the hull shader can be used. The algorithm for finding patches that need to be tessellated is very similar to the algorithm for generating the displacement map texture coordinates. The hull shader code runs through the positions of the decals, and calculates whether the current patch is within the radius of the decal.

Since there are separate tessellation factors for the edges and inside, the distance from each edge to the decal is compared. If any of the edges are within the radius of the decal, then the inside automatically needs to

LISTING 1

A pseudo code example of the domain shader that implements decal tessellation. The loop iterates over the decal locations to see if the current vertex is close enough to the center of a decal. If it is, the texture coordinates for the decal's displacement map are calculated, and used to fetch a height value for displacing the vertex.

```
for (each decal in the list)
{
        WorldToDecal Matrix = [DecalNormal, DecalBinormal, DecalTangent]

        if ( the vertex is within the decal´s radius )
        {
                // Generate the displacement map texture coordinate
                // by transforming it into the decal´s coordinate space.
                vTrans = vertex - decalPosition;
                TexCoord  =  WorldToDecalMatrix * vTrans;
                // scale and bias TexCoord to within 0 and 1
                TexCoord  /=  decalRadius * 2;
                TexCoord += 0.5;
                // set z coordinate to 1 to tell the pixel shader this is valid
                TexCoord.z  = 1;

                // Displace the vertex
                vertex  += sample (DisplacementMap, TexCoord) * -DecalNormal;


                // Transform the light for normal map lighting
                Binormal  = VertexNormal x DecalTangent;
                Tangent  = Binormal x VertexNormal;
                Tangent SpaceMatrix = [VertexNormal, Binormal, Tangent];
                LightVector  = TangentSpaceMatrix * LightVector;

                break;
        }
}
```

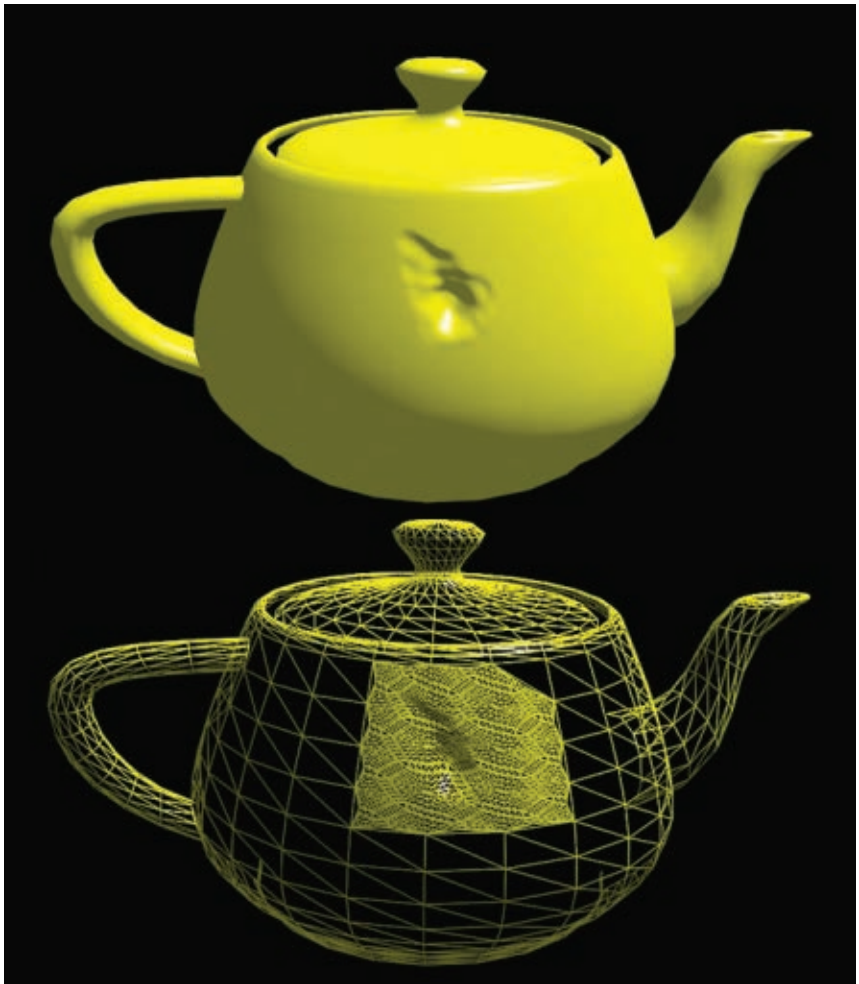


FIGURE 4 Displacement adaptive tessellation is shown. The bottom half of the figure is the wireframe view of the teapot. Tessellation is only done on patches that are part of the decal.

be tessellated. If none of the edges are within the radius of the decal, then we still need to check to see if the inside of the patch is within the radius of the decal. This can be done by calculating the distance from the decal location to the plane formed by the triangle patch. We also need to do a point in triangle test to make sure the distance being measured is from within the triangle patch, since a plane is infinite. Tessellation is "disabled" for patches that don't need tessellation by simply setting the tessellation factors to 1. There is still overhead from the hull and domain shaders even when the tessellation factor is set to 1, though. Therefore, it is advisable to render the mesh with a non-tessellated draw call if you know in advance that none of the patches need to be tessellated. Listing 3 shows pseudo code for the hull shader. Depending on the mesh and the level of tessellation used, adaptive tessellation can have an improvement of as much as one or two orders of magnitude.

Other types of adaptive tessellation that can improve performance include: distance-based adaptive, screen space adaptive, orientation-based adaptive, and density adaptive.

• Distance-based adaptive uses the distance from the camera to the object to determine the tessellation factor. This can be implemented by interpolating a minimum and maximum tessellation factor over a min and max distance. When the object exceeds the maximum distance, tessellation is unnecessary, but normal mapping can still be used to get correct lighting. The world space distance can be determined before the draw call, so the tessellation factor can be set prior to rendering. The remaining techniques are done on a per-patch basis, so they are implemented in the patch-constant phase hull shader.

• Screen space adaptive is similar to distance-based except that it uses the length of patch edges projected into screen space to determine the tessellation factor. With this technique, patches closer to the camera will get tessellated more.

• Orientation-based adaptive uses the patch face normal to determine how much to tessellate. When the normal is near perpendicular to the view vector, the silhouette is most visible, so tessellation should be the highest.

• Density adaptive analyzes the displacement map to see where there are areas of higher frequency of displacement that would require more tessellation. Areas of the displacement map that are smooth require less tessellation, so those areas can use lower tessellation factors.

In addition to adaptive tessellation, there are big performance gains from simple culling. Backface culling can be done in the hull shader by checking if the patch is facing away from the camera. If so, the tessellation factors are set to 0, which cause the patch not to be rendered. Frustum culling is especially useful if there are many objects with decal tessellation outside the viewing frustum. Whole objects can be culled, or frustum culling can be done in the hull shader for large objects, like decals placed on terrain. Usually, the best performance can be achieved by using a combination of various adaptive tessellation and culling techniques.

LIMITATIONS AND CONSIDERATIONS

» One of the limitations of decal tessellation is that it requires the geometry to be a triangle list. In Direct3D 11, tessellation is not allowed for triangle strips, but it's fairly easy to expand a strip into a list by creating a new index buffer. One solution would be to go through the geometry at level load time and create new index buffers for strips that you may want to use with decal tessellation.

Another issue to consider with decal tessellation is collision detection after tessellation. In most cases, collision detection is done on the CPU, and since tessellation and displacement is done on the GPU, the tests that check for collisions are done with

different geometry. This may not be too big of a problem for small displacements, but if large displacements and large decals are used, incorrect collisions could result. For example, suppose a tessellation decal is used to remove part of a wall that is being used for cover. If someone tries to shoot through that area, the game will still see an obstruction because the geometry on the CPU hasn't changed. As of now, it seems the only solution is to change the CPU collision geometry. This may not be so bad, since collision geometry is often simplified geometry, such as bounding boxes. Splitting the object up into multiple bounding boxes and reducing the size of the bounding box in the area where the decal was placed may be a solution.

Another thing to consider when using decal tessellation is the amount of displacement applied. If the size of the object is small and the displacement is large, then the displacement could go through the object. The scale of the displacement could be determined by the object's bounding box, or some literal displacement scale associated with each object.

Decal tessellation may need additional work to be used with animation. Currently, the decal locations are stored in world space. If the mesh is deformed by animation, the decal locations and orientations need to be transformed as well.

Overlapping decals can also have edge artifacts because of inconsistent texture coordinates generated for a triangle. One way to avoid overlaps is to resize a new decal before it's added so that it's small enough to fit without overlapping any existing decals.

FUTURE WORK

>> There are a few things that could be done to add value to the decal tessellation technique. One thing would be to make the decal displacements additive so that if one decal overlaps another, the resulting displacement is the sum of the two. This would be fairly easy to implement, but some care must be taken so that if many decals overlap, the displacement isn't too large, causing it to extend outside the opposite end of the object. Another valuable feature would be to make it possible to create holes or fractures. Since 3D models are basically hollow shells, not solid objects, a hole going through an object would need extra polygons to connect one end of the hole to the other. Fortunately, tessellation does add new polygons, so it may be possible to generate new polygons for the inside of the hole, and then remove the polygons at the opposite end to make an opening. Decals are meant to represent physical damage, so another good feature to add would be integration with a physical simulation of the damage. For example, Digital Molecular Matter (DMM), developed by Pixelux, is a physical simulation of soft body dynamics. This could be used to generate displacement maps for decal tessellation in real time to get realistic damage. Doing the physics simulation on the GPU would be an advantage because the displacement maps can be rendered directly into a texture, without having to be uploaded from CPU to GPU memory.

Decals have been an effective and simple way for games to represent damage to an object. With the capabilities of current-generation graphics hardware, it is now possible to improve on this technique by forcing the decal to change the geometry of the object. Decal tessellation can be an easy way to make the game environment a step closer to being fully destructible, and thereby adding to the realism of the experience.

LISTING 2
A pseudo code example of the portion of the pixel shader used for determining what normal and light vectors to use for lighting the pixel.

```
if ( TexCoord.z  = 1)
{
        vNormal  = sample (NormalMap, TexCoord);
        vLight  = tangent space light vector;
}
else
{
        vNormal  = interpolated vertex normal;
        vLight = world space light vector;
}

Calculate the lighting for the pixel using vNormal and vLight -
```

LISTING 3
Pseudo code for the patch-constant phase of the hull shader used for determining the tessellation factors.

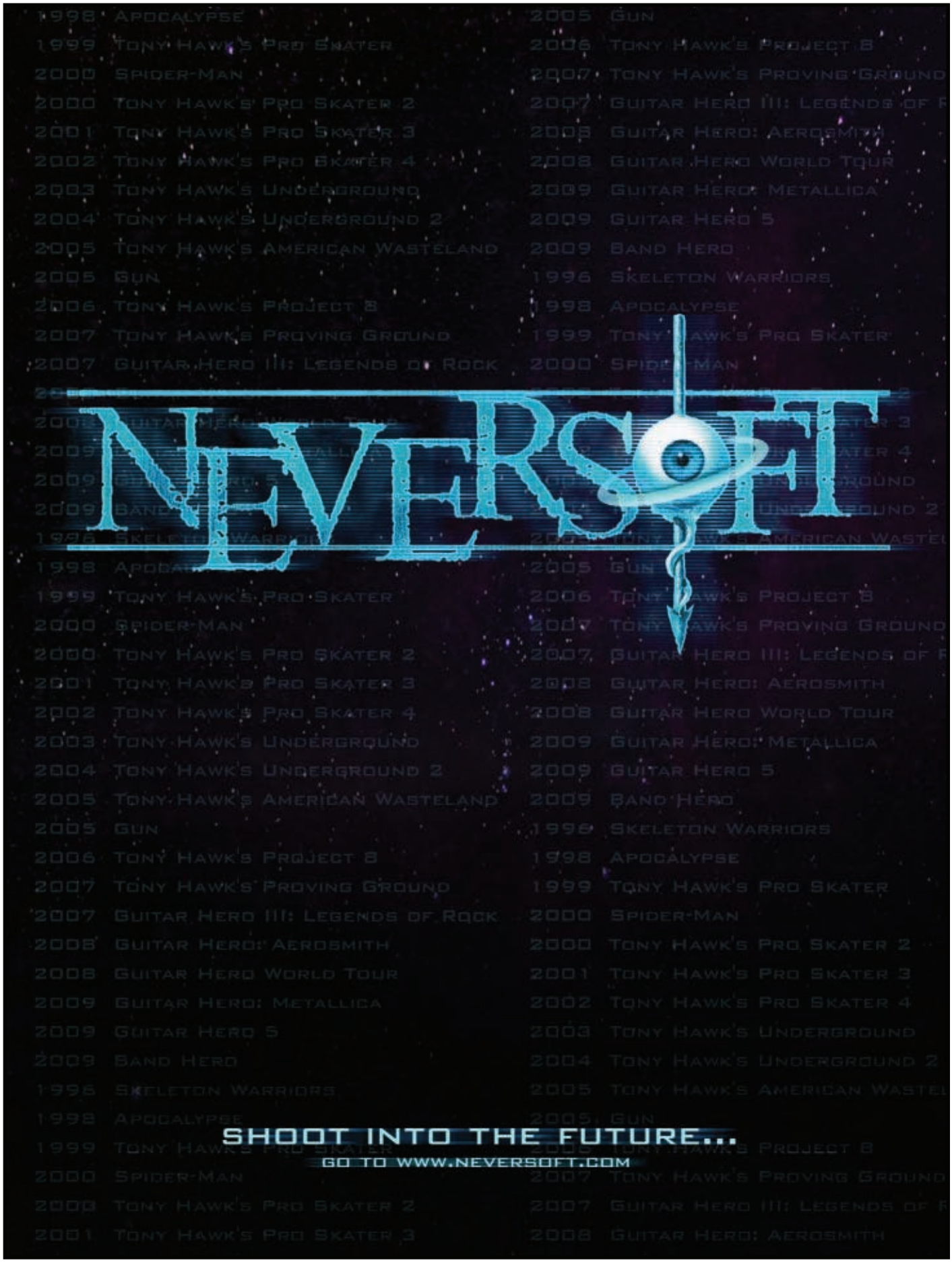
```
for (each decal in the list)
{
        if ( either vertex of Edge 0 is within the decal´s radius )
        {
                        tessellate this edge
        }
        else
        {
        if (  ( distance from the decal to Edge 0 is within the decal´s radius )
                                        && ( decal intersects the line segment )  )
                {
                        tessellate this edge
                }
        }

Do the same for Edge 1 and Edge 2 -

        if ( any edge is tessellated )
                || ( ( distance from decal to triangle plane is within the decal´s radius )
                && (decal projected onto the triangle plane is within triangle ) )
        {
                tessellate the inside
        }
}
```

BILL BILODEAU *is an engineer in the ISV Relations Group at AMD. He works with game developers to help them make their graphics run faster and look better. He has been in the game industry for over 10 years, working for hardware developers and game companies, with a specialization in real-time graphics.*

The complete working source code used for this article is available for download at www.gdmag.com/resources/code.htm.


1998 Apocalypse
1999 Tony Hawk's Pro Skater
2000 Spider-Man
2000 Tony Hawk's Pro Skater 2
2001 Tony Hawk's Pro Skater 3
2002 Tony Hawk's Pro Skater 4
2003 Tony Hawk's Underground
2004 Tony Hawk's Underground 2
2005 Tony Hawk's American Wasteland
2005 Gun
2006 Tony Hawk's Project 8
2007 Tony Hawk's Proving Ground
2007 Guitar Hero III: Legends of Rock
2008 Guitar Hero: Aerosmith
2008 Guitar Hero World Tour
2009 Guitar Hero: Metallica
2009 Guitar Hero 5
2009 Band Hero
1996 Skeleton Warriors
NEVERSOFT
SHOOT INTO THE FUTURE...
GO TO WWW.NEVERSOFT.COM

iPad, 3D, iPhone, Game Narrative Summits at GDC Online

ORGANIZERS OF THIS OCTOBER'S GDC ONLINE game conference in Austin have confirmed summits to run parallel with the major event, including one and two-day events dedicated to game development on Apple's iPad, 3D stereoscopic games, iPhone games and video game narrative.

The summits, to be held on the first two days (October 5th and 6th) of the four-day event, will present key in-depth business and technical advice in these major up-and-coming facets of the game industry, ahead of the online game-centric Main Conference content.

IPAD GAMING SUMMIT

Debuting at GDC Online, the one-day iPad Gaming Summit, paired with the iPhone Games Summit, will provide an abundance of key facts and advice for all those working on the iPhone and iPad platforms.

The iPad Summit provides a springboard into the world of iPhone/iPad game development, sharing key business learnings and top technical takeaways from developers already surging to major success on this already extremely competitive platform.

3D STEREOSCOPIC GAMING SUMMIT

In another debut, the one-day 3D Stereoscopic Gaming Summit will be the first GDC event to explore the world of 3D stereoscopic gaming in depth.

With 3D-capable movies such as *Avatar* and *Alice In Wonderland* topping the worldwide box office for multiple weeks, and graphics engine companies and console manufacturers alike touting the rise of 3D in gaming, the 3D Stereoscopic Gaming Summit will gather industry experts to explain and demonstrate the latest advances in the area.

Sessions will cover how to build 3D support into future games, potential technical barriers and solutions, and an analysis of key trends surrounding hardware adoption and penetration.

IPHONE GAMES SUMMIT

Returning to Austin after its successful debut in 2009, the third iPhone Games Summit will once again bring together top game developers from around the world to share ideas, best practices and discuss the future of this revolutionary platform.

This one-day summit, paired with the iPad Summit on the following day, will focus on driving business and marketing strategies and how to truly succeed through micro-transactions through the iPhone 4.0 operating system and beyond.

With advice from successful iPhone game companies, leading speakers and advisors in the space this event will be a must-attend for those in the space, or just looking to learn more about this space.

GAME NARRATIVE SUMMIT

The 2010 Game Narrative Summit, evolving in name and intent from the long-running Game Writers Summit, is dedicated to exploring the state and future of storytelling in video games—all the way from emerging independent and social game experiences to the biggest AAA game titles.

The two-day event will span all facets of the game story-making process and will bring together experts from the writing, design and creative game communities to showcase what interactive entertainment can do with storytelling—and why it's so vital to the future of video games.

Sessions will address the art of writing scripts that seamlessly integrate into major console and online game titles with the new breed of story experiences, centered in anything from ARGs to Facebook games.

"With this new batch of specialized summits, GDC Online proves once again that this event is at the forefront of the emerging technologies and platforms of the future," says Izora De Lillard, event manager of GDC Online. "Alongside the key online game event this October, we're proud to present these summits on the new technologies that make a difference in today's game business."

In addition to these Summits, the three days of GDC Online main conference sessions will offer a host of business, marketing, design and technical content vital to those making social network titles, free-to-play web games, kid-friendly online titles, larger-scale MMOs, or experimenting with other online gaming genres.

GDC Online 2010 takes place at the Austin Convention Center in Austin, Texas from October 5th to the 8th. The call for submissions for both GDC Online summits and main conference sessions will open in the near future, and more information will be available at the GDC Online web site at **www.gdconline.com** in the near future.

GDC Europe Debuts Keynote From BigPoint CEO Hubertz

GDC EUROPE ORGANIZERS HAVE ANNOUNCED the first keynote for the August event in Cologne, with Heiko Hubertz, CEO of dominant German online game firm BigPoint discussing elements likely to include the company's NBC backing, the BATTLESTAR GALACTICA MMO, and pointers for business success.

BigPoint, which was founded in 2002, now employs 340 employees in Hamburg, Germany. Its runaway financial success in the browser game marketplace led to NBC taking a 35 percent stake in the firm in 2008.

As a result, the company recently opened a U.S. office, announcing a major partnership with engine company Unity and an MMO based on NBC Universal's *Battlestar Galactica* property along the way.

Company founder Hubertz is likely to discuss the firm's history, trends in the industry, and tips to succeed in the competitive online game business as a major keynote for GDC Europe, which is ramping up announcements for its second event.

The conference, taking place Monday through Wednesday August 16–18, 2010 at the Cologne Congress Center East in Cologne, Germany, will once again run alongside the major GamesCom event to present the leading game industry event for developers, consumers, publishers and trade professionals.

By again pairing GDC Europe with GamesCom, Europe's leading consumer and industry show, the conference can offer content to address the development community at a central location in the heart of Europe and command the critical mass of the European games sector. In 2009, its first year, GDC Europe saw more than 1,500 participants, including 130 international speakers, 40 exhibitors and 240 media representatives.

For more information on GDC Europe, including location and how to purchase passes—for which registration is now open—interested parties can visit the official Game Developers Conference Europe web site at **www.gdceurope.com**.

THE REAL WORLD

THE UPS AND DOWNS OF USING REAL-WORLD REFERENCE MATERIAL

LIFE IN THE GAMER DAD DEMOGRAPHIC CAN BE VERY STRESSFUL, particularly if you've spent a lot of your career working on games that aren't pitched at youngsters. It's tough to explain for the millionth time why junior can't see what you've been doing at the office, or why you have a shelf in the game closet full of disks they can't touch. Of course, the filters you try to maintain are always a bit leaky. After all, if you didn't obsess over games, you wouldn't make them; plus, it's hard to keep your obsessions under wraps until the tots are safely tucked in for the night.

So, it wasn't entirely surprising when my fourth-grader snuck up on me as I slogged my way through the trenches in the (T-rated) TOY SOLDIERS. Ordinarily, those you-shouldn't-watch-this-yet conversations are driven by dutiful parenting discipline of the "yes, we have to brush teeth every night!" variety, but this time the lecture went down an entirely different route. Rather than the toy mayhem on-screen, my son focused intently on the unfamiliar setting and period style of TOY SOLDIERS. He's a child of the 21st century; for him, biplanes and pancake helmets are more exotic than elves and dragons.

TOY SOLDIERS does a marvelous job of channeling the aesthetics of the Great War era. From the waxed mustachios and pointy pickelhauben of the Prussian officers to the flowery Belle Epoque typography, the atmospherics of the game are a brilliant introduction to a world that's far more alien to most of us than the nominally "fantastic" fantasies we see on a regular basis. That thorough job of art direction turned my routine bit of parenting into a long detour through the history of the first World War. The kid, who is so tuned in to the clichés of modern action that he finds the original *Star Wars* "too slow" and "boring," was entranced by a scratchy gramophone recording of the classic war tune "Goodbye, Dolly Grey." He ended the evening watching the 1927 silent movie *Wings* (still the best WWI movie ever made) with intense interest and questions. The next day, I caught him trying out his newfound knowledge on his little sister, gravely explaining the fate of Archduke Franz Ferdinand and how taxicab drivers saved France in The Miracle of the Marne. Seeing the power of our medium laid out so clearly was even more satisfying than finally blasting that damn armored train boss off the tracks.

The design of TOY SOLDIERS from Signal Studios elegantly combines the aesthetics of the first World War with a whimsical childhood setting.

REAL AND UNREAL

>> Longtime readers may know that in a previous life, your humble columnist dropped out of a PhD program in history, so perhaps I have a weak spot for arcane details and period knowledge. Still, this little story illustrates an incredibly powerful truth that we often forget in the games business—namely how much power comes from engaging with the real world. Swap out the gas-masked Tommies and Maxim guns of TOY SOLDIERS for laser blasters and aliens, and you might still have an entertaining tower defense game, but you would lose the strange combination of antique charm and tragic pathos which makes the game such a gem.

As game artists, we've been granted an amazing privilege. We can bring to life almost anything we can imagine. Unfortunately, it's all too easy to get lazy and rely on recycled genre conventions instead of accepting the challenge that comes with limitless power. There is more genuine strangeness in a well-researched and well-presented real-world setting than in 99 percent of the fantasy or science fiction games ever made. Who would dare dream up the pastel-colored favelas of MODERN WARFARE 2? How many concept teams would dare dress an army in big fat stripes, unless they had an ASSASSIN'S CREED 2-sized folder of reference pictures? Who would cast a short, balding, chubby fellow like Napoleon as an all-conquering dictator? The real world provides an infinite wealth of looks, styles, and icons that can spark ideas and challenge our tendency to stick to safe mass-market choices.

Being artists, of course, some of us would rather create everything from scratch. Artistic masochism is a common syndrome. I once overheard a teammate dissing a stunningly realistic character because it was made using photo reference, as though working from pictures were cheating! Sure, grabbing a reference off of the Internet is less laborious than creating a brand-new concept drawing from scratch. Still, that doesn't mean working from reference isn't "artistic" or creative. Sifting, sorting, and hunting down evocative real-world material demands careful selection, judicious edits, and an artist's eye. Film and television prize location scouts

and costume and set dressers who can find exactly the right vintage doodad to set the mood. Why shouldn't we value the same skills?

In any case, it's wrong to imply that concept design and research are opposites. Most successful concept artists are constantly seeking out new images and inspirations from real world sources. Great designers are voracious omnivores, devouring images and ideas from every available source. The same World War I tanks that creep across the battlefields of TOY SOLDIERS find a new life as Imperial armor in the highly stylized world of WARHAMMER 40K, and Darth Vader got his signature look from the mask of a 17th century samurai. Good reference material has all of the warts and wrinkles of the real world, which are great sources of inspiration, even when the nominal subject is something fantastic.

REALITY BITES

>> Unfortunately, the resonance and power that comes from smart use of real-world subject matter comes with a price tag. Working with real-world subjects also raises interesting ethical and practical questions. Comedian Hari Kondabalou has a great routine about his mixed feelings on tearing open his eagerly awaited copy of MODERN WARFARE 2 and discovering a multiplayer map set in his home town of Karachi. Not only was he offended, he didn't even gain any tactical advantage for fighting on his home turf!

This neatly illustrates two truths about realistic material: first, you are inevitably going to get involved with real-world attitudes and controversies. Second, you're never going to be able to satisfy all the disparate expectations raised when people see familiar settings or characters in digital form. When you work with recognizable locations, characters, or icons, everybody's a critic. Whether it's coping with that guy in the forums who knows you completely fudged the suspension on the T34-85 or fielding complaints from an angry chamber of commerce, the use of real-world material can be a thankless job. Grabbing designs off the Internet might seem like it's easier than making stuff up from whole cloth, but it's got hidden costs.

I once worked on a game involving an episode set in Bangladesh. Although the player's role was to support the government of Bangladesh, and (so we thought, at any rate) the treatment of the country was positive, advance word of the game leaked into the local press. This generated a huge furor, with threats of lawsuits and rumors of real-world violence. Our publisher quickly decided to excise all references to Bangladesh from the game, and the existing levels were suddenly airlifted to the fictional "Republic of South Asia." This is far from the only case where politics and game design have collided. Québécois politicians pressured Sony into canceling a SYPHON FILTER mission about a separatist attack in a Canadian shopping mall. The South Korean government has banned several games for negative portrayals of North Korea (while the North Korean government said of GHOST RECON 2: "This may be just a game to them now, but it will not be a game for them later. In war, they will only face miserable defeat and gruesome deaths!").

It's obvious that portraying the government or people of a foreign country as dastardly villains will produce some ill will (apparently, the major publishers have entirely written off the former Soviet republics of central Asia ... Here's a tip, guys: Buy more games and we'll set up the fictional dictators somewhere else!). Worse, you can stir up a hornet's nest without even trying: When RESISTANCE: FALL OF MAN let players fight inside the historic cathedral in Manchester, England, church officials threatening legal action generated a lot of negative comments in the press and politics. More famously, RESIDENT EVIL 5 raised accusations of racist imagery: Set in Africa, the game featured light-skinned protagonists blowing away hordes of dark-skinned zombies.

Real-world subject matter inevitably involves some controversy. It's not easy to balance creative freedom against the sensitivities of the people and places you depict. It is important, though, to be a bit self-consciousness about what you're creating. It helps to have a lot of different backgrounds on your art team: as N'Gai Croal wrote, "I looked at the RESIDENT EVIL 5 trailer and I was like, 'Wow, clearly no one black worked on this game.'"

It's also critical to have more than just a visual appreciation of your material. If you just troll through Google Images looking for striking pictures, you might miss out on important pieces of background: is that cool building a sacred site for somebody else's religion? Is the impressive leader in that statue an infamous ethnic cleanser? Understanding the context of the art and environments you're looking at is critical, both for good art direction and for navigating the tricky currents of real-world sensibilities.

REALITY CHECK

>> Of course, we're not always free to choose our subject matter. It's possible that the world is hungering for a game about a fascinating subject like the Dahomey Amazons, a female military regiment in the 19th century, but the pitch meeting would probably be a short one. Publishers are obsessively interested in the years 1939–1945, selected bits of the European and Japanese middle ages, and little nuggets of Greece, Rome, and possibly Egypt. Huge swathes of time and whole countries, even continents, get glossed over. Even when the subject matter is not deemed too controversial, it may still be dinged for failing to be commercial. It's a vicious cycle in which timid choices and me-too decisions perpetuate themselves. In our era of graphics bloat and $30 million budgets, it's no surprise that publishers tend to err on the side of caution.

The Mark IV tank—created in the 20th century, and revamped for the 40Oth. DAWN OF WAR shows how Games Workshop's designers found real-world inspiration for sci-fi armor.

If you're an idealist, you might point out that our narrow choice of palettes is based on a lazy reading of the market. "Commercial" isn't a fixed condition; it's always being re-negotiated. Not too long ago, games treated World War 2 as a niche market for strategy games and flight sims, until the 50th anniversary of D-Day and *Saving Private Ryan* made it mainstream (Note to publishers: we're only one product cycle away from the centennial of World War I !). Meanwhile, settings that are mainstream in other media—say, the Western—haven't caught on in the games business due largely to the lack of a standard-bearing franchise. A good enough game can make any subject appealing, where a me-too product is doomed to extinction even in a popular genre. The world is full of unexplored territory that will make great games, and only adventurous developers will be able to claim it.

If you're a cynic, you'll probably counter that it's hard to get even our own teammates fired up about obscure cultural and historical references—which is true, but only if all you can get from rummaging through the whole of human history and culture is a bunch of "references." If you use real-world material only for its own sake, you'll have a hard time finding an audience.

It's also true, though, that even mass market audiences are remarkably tolerant of unfamiliar material, as long as the basics of storytelling, gameplay, and of course, artistic presentation are well handled. There was a time when subjects like samurai, kung fu, and Bollywood were too exotic for the mainstream, but now they're familiar to every high school kid. Tastes change and grow, and what makes our job so awesome is that we can make that happen.

Of course, sometimes the best way is to dial down the erudition and let the material speak for itself. "Let's go see a movie about an ascetic order, devoted to Mahayana Buddhism and historical preservation, caught up in the dynastic politics of the Tang Dynasty" doesn't have quite the zing of "Hey, it's *The Shaolin Temple* with Jet Li!"

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

THE SOCIAL REVOLUTION

GAME DESIGNERS OR SOCIAL ENGINEERS?

FOR THE FIRST TIME IN RECENT MEMORY, THE dominant topic at this year's GDC was neither the current console generation nor the one coming over the horizon. Instead, the industry obsessed over the astonishing explosion of Facebook gaming over the last year. In fact, the poster child for the social network's success—Zynga's FARMVILLE, with its 82 million monthly active users—had been out for just a mere nine months.

FARMVILLE's scale is difficult to compare with other games. Before its plateau in March, the online game had grown by the size of the entire WOW user base every single month. Certainly, FARMVILLE must also be the first online game that can claim to be actively played by over 1 percent of the world's population.

Many traditional game developers have mixed feelings about the rapid growth of these social games. While the huge audience signifies a massive broadening of the worldwide gaming demographic, the games themselves are often simplistic affairs, emphasizing time investment over interesting decisions. Further, certain practices have given the format a bad name, such as monetization through dubious lead generation offers and viral growth from wall post spam or in-game pyramid schemes.

Nonetheless, Facebook gaming does represent a real breakthrough for the industry because the social network combines an enormous audience with four advantages that promise great things for gamers and designers alike:

TRUE SOCIAL PLAY. Gaming can now happen with the given assumption that you can play within your existing group of friends. Multiplayer games no longer suffer from the Catch-22 of requiring friends to be fun while new players always start the game without friends.

PERSISTENT ASYNCHRONOUS PLAY. Finding time to play with one's real friends is difficult, especially for working adult gamers. Asynchronous mechanics let gamers play at their own pace and with specific friends, not strangers who happen to be online at the same time.

FREE-TO-PLAY BUSINESS MODEL. New players need not shell out $60 to join the crowd. Consumers don't like buying multiplayer games unless they know their friends are all going to buy the game as well (and for the same platform). The free-to-play model removes that friction, and being web-based solves the platform issue.

METRICS-BASED ITERATION. Retail games are developed in a vacuum with designers working by gut instinct. Games get only one launch—a single chance to succeed. Many developers would love, instead, to iterate quickly on genuine, live feedback.

Developers who master these four characteristics of Facebook gaming stand the best chance of breaking away from the pack in an increasingly crowded field. To help me understand these dynamics, I interviewed two Zynga developers, senior designer Paul Stephanouk and director of product management Siqi Chen, and asked them to describe their experiences in this new field. (The full interviews will be available on Gamasutra.com.)

SOCIAL FIRST, GAMEPLAY SECOND

>> Brian Reynolds, Zynga's chief designer, often points out that successful social games need to be social first and games second. But just because a feature is social first doesn't mean that it won't be interesting, Chen explains. "In a certain context, to certain people, deciding whether to plant grain or grapes, whether to buy a chicken or a fruit tree, are all interesting decisions," he said. "Letting players gift things to their friends is a good way to build up social currency, but it becomes an interesting choice when the gift itself is scarce, or it costs the players something real. There's a place for skill and challenge, but skill and challenge in a social game might not be as obviously punishing or competitive as a traditional game. Making your farm in FARMVILLE look impressive is a real skill, and making it look more impressive than your neighbor's can be a real challenge."

Playing within the context of one's actual friends brings new emotions to the table: pride, obligation, gratitude, desire, and even shame. FARMVILLE's wither mechanic—in which crops die out and shrivel if not harvested in time—is an example of a social mechanic designed to shame players into caring for their virtual farm. "What will my friends think of me if my poor farm is full of dried-up strawberries?"

In fact, some social games have incorrectly copied this dynamic by taking the wither mechanic while ignoring the social factor. In PONZI, a social game set in the corporate world, the reward for finishing jobs drops to zero if the player does not return in time to pick up the check. Although this mechanic does encourage players to return regularly, it lacks the social pressure found in FARMVILLE because the decaying jobs are invisible to one's friends.

ASYNCHRONOUS INNOVATIONS

>> Although the social factors are paramount, Facebook titles do pose new, interesting design challenges. For example, two distinct mechanics are currently evolving on Facebook to handle offline progress—the energy system and the appointment mechanic. Under the energy system, each action costs a certain amount of energy which regenerates in real time. Eventually, the player must wait for her energy bar to refill to some degree before resuming play. In contrast, appointment mechanics are free to start, but they lock the player out for a specific period of time, for example after planting strawberries in FARMVILLE, the player must return in four hours to harvest them and collect the sale.

FARMVILLE's wither mechanic—in which crops die out and shrivel if not harvested in time—is an example of a social mechanic designed to shame players into caring for their virtual farm. "What will my friends think of me if my poor farm is full of dried-up strawberries?"

Stephanouk explains the pros and cons: "Progress-oriented players tend to respond better to the energy approach—MAFIA WARS is an excellent example of this," he says. "Appointment, or 'return' mechanics, are perceived as a softer approach. Return works well in games like FARMVILLE where players are, as a whole, less competitive and more focused on the social and building components. Overall, it really depends on

the game. Both methods can be successful and can even be combined. FARMVILLE has examples of both models; tractor fuel is an example of energy."

The energy system has the advantage of being a more natural match for profitable virtual items; a booster pack, for example, can allow players to refill their energy and continue playing the game. Appointment mechanics, on the other hand, allow players to strategize around their real-life schedule. Tasks that take 15 minutes to complete are useful for players that know they'll be staying online; they can tab over to the game at any time. Tasks that take two or eight hours, on the other hand, are great for players going to dinner or heading off to bed.

MEET THE MAINSTREAM

>> One big difference between social and core games is the subject matter. Instead of the niche themes usually found in retail games—fantasy, sci-fi, racing, WWII, zombies, and so forth—successful social developers choose very mainstream topics. Facebook's top 10 games include subject matters such as farms, restaurants, pets, and aquariums. The format developed so differently because, unlike with consoles, handhelds, or high-end PCs, the audience started out mainstream, without having to grow from early adopters with more niche tastes.

In many ways, Facebook is the industry's first "TV of gaming." The site allows users to flip from game to game in a safe, standardized environment, with the expectation that there will be no barriers to entry and that players' friends will be playing the same games. By allowing players to advertise their accomplishments and invite their own personal network to play, the site goes beyond TV by letting players exert direct social influence on each other.

The mainstream audience affects not just the distribution or themes of social games but their underlying mechanics as well. Stephanouk describes what he had to unlearn when transitioning from real-time games like RISE OF NATIONS to social gaming.

"One of the things I had to come around on was the importance of zero-sum conflict," said Stephanouk. "Coming from strategy games as I did, I was very focused on the competitive aspect of games. I was aware of players wanting to build or explore, but I always saw that as serving a conflict-driven goal. I have learned that, for many people, the conflict-driven nature of traditional games is a major detraction. I'm not saying that overall conflict is bad or that you can't have conflict-driven action in social games—both of these things are very much not the case. What I am saying is that there are a lot of players out there, far more than I understood, that really want a game experience that isn't driven by the need to compete against another person."

Zero-sum conflict is indeed one mechanic that core game developers tend to take for granted. Although cooperative gaming has grown in popularity in recent years, judging by the acclaim for LEFT 4 DEAD or the auto-grouping feature in WORLD OF WARCRAFT, competitive play usually means that one side triumphs and another is destroyed. Social games, on the other hand, can still be competitive without being destructive. The answer is parallel competition. For instance, the race to grow and improve one's restaurant faster than one's friends.

The agrarian ideal in FARMVILLE.

WHO IS THE DESIGNER?

>> Another important area in which social games differ from traditional games is the pervasive use of metrics to inform rapid iteration, often on a weekly or even daily schedule. The ability to test design hypotheses by split testing can revolutionize development. Chen provides one simple example: "Back when I was running Serious Business [a social game company Chen founded which was later bought by Zynga], Facebook allowed applications to access the notification channel, and we wanted to find out whether longer notifications performed better, or shorter ones. I guessed that it was probably a wash—the shorter ones are more concise, but the longer ones were probably more noticeable since they were physically larger. We ran a 30-way split test where we asked our team to come up with a bunch of different copies. As it turns out, there was a roughly linear correlation between how short the notification was and how often players would click through it. The shorter it was, the better the performance. The difference in performance was up to 300 percent. That's a huge impact for basically writing a few lines of copy."

The question on many designers' minds is what their role might be in this new environment with virtually real-time feedback for development decisions. Is the designer still the primary "author" of the game experience, or do designers now fill a new role, surfing the incoming data while sitting in the murky middle ground between the community and the company? Indeed, Reynolds admits that his role as Zynga's "chief designer" is not nearly as important as one might imagine.

Stephanouk discussed with me the role of metrics in his current job. "Metrics are everything I thought they might be," he opined, "or at least what I hoped they would be every time I found myself sitting in a room of designers fighting over whether a player would rather press one button over another. Why would a designer want to remain in the dark on something that has a clear, knowable answer? Understanding how players play doesn't stifle creativity in game design any more than understanding how people live stifles creativity in architecture. I think it's the other way around: knowledge helps us understand constraints and constraints are usually the building blocks of good design. Do I feel less like an 'author?' That depends, fiction or non-fiction?"

The designer-as-auteur ideal is perhaps incompatible with this model, but the best game makers are usually the ones willing to "get dirty," to engage fully with the audience to discover which ideas actually work and which are wishful thinking. Social game development simply accelerates this process to new extremes.

SOREN JOHNSON *is a designer/programmer at EA, working on browser-based strategy games at www.strategystation.com. He was the lead designer of CIVILIZATION IV and the co-designer of CIVILIZATION III. Read more of his thoughts on game design at www.designer-notes.com.*

SETTING THE STAGE

WORKING WITH VOICE ACTORS

EVERY VOICE SESSION IS A DELICATE DANCE ACTED out by a number of key players. In the booth, the engineer and the voice director work together to capture the right performances. In front of the microphone the actor works to breathe life into the game's characters. Both sides of the glass are working with limited time and resources to achieve their goals of a convincing vocal performance.

For an actor's perspective on how developers can best prepare and run a voice session, I spoke with Nolan North. From SHADOW COMPLEX to ARMY OF TWO, from PRINCE OF PERSIA's Prince to UNCHARTED's Nathan Drake, Nolan has lent his talents to some of the most recognizable characters in this generation of gaming.

Nolan North and Emily Rose during rehearsals for UNCHARTED: DRAKE'S FORTUNE.

CLARIFYING THE CONTEXT

>> When we spoke, it became clear that much of the important work developers can do for a voice session occurs before even reaching the recording studio. "You have to have some context in order to give the proper performance," said North. As such, concept art and a character's backstory become crucial information for the voice talent. "I think that helps any actor find the character," he said. "If they have the concept art and it's a big troll or ogre, or a big Cockney gruff-sounding character, for someone walking into a new project, that would be invaluable."

There are additional ways to help give an actor context other than visuals. For North, the script itself is the other piece of the puzzle. "Even on a telephone book-sized stack of scripts, little contextual notes on the side always really help," he said. "It's gotten better in the last couple of years; in the past, we sometimes just got scripts with all our lines and had no idea what the character before is saying to [us], or how they're saying it."

Simply adding extra information to the script isn't enough, though. How the lines are presented can be just as important as the lines themselves. According to North, " ... because I started in theater, I like script format rather than Excel spreadsheets, but some games are just so dialogue heavy and so intense, it's just not plausible to do that." When he receives a script with his lines in bold text, or colored and highlighted differently than surrounding lines, "That's a help because I'm turning so many pages," he says.

Many developers don't have time to read through the script with all actors beforehand, but when possible, North says this is very helpful. "Table reads and rehearsals ahead of time, I think that can only help," he added. "A lot of developers hesitate to do that simply because it costs money. But at a certain point, you have to look at what the investment is and what the return is going to be. I think it gets everybody—in terms of the director, the actors, everybody—on the same page. We know where the story starts, where it's going, and where it ends. Again, it helps to give that overall idea of context. Ultimately, I would imagine that would save time. There [would just be] less mistakes."

TUNING THE TEAMWORK

>> With the script streamlined and the character rooted in the context of the game world, the task turns to the organic process of recording. One of the most important elements is to make sure you have the right people on both sides of the glass. "I don't believe there's any substitution for a good voice over (VO) director," North tells me. "Certain directors know how to talk to an actor whereas sometimes some developers just don't know how to speak actor-speak.

"Now, [if] you have a bad VO director, the developer may be better," North warns. The issue at hand comes down to the dynamic between the different personalities in the room. "A good game really depends on good chemistry," he says, adding by way of example that, "certain actors don't like to take a line reading. I've never minded one because, you know, if I'm missing the line and I'm not saying it the right way, maybe I don't know the full context of how you want me to say it. If you need it done a certain way, I'm here to please you. Some actors won't do that. They get very offended. A VO director will know a way to kind of nudge them in the right direction whereas a developer may just—innocently—say, 'Hey, can you say it like this?' and then there's tension. Never underestimate the size of an actor's ego."

When I asked North about the one thing he hopes to not run into when working with a new team, his answer was quick and simple: "An asshole. A big ego. You just want everyone to be proud of what they're doing, motivated, and you don't want that one rotten apple to spoil the whole barrel. If it's a new team, you just want them to be cool.

"What we do is a lot of fun," he summed up, "and I really take pride in the fact that I want to make it as fun as possible while we get the material we need in a timely manner. When you get somebody in the booth who is passionate about the project they're working on and they want to have a good time, it's golden."

JESSE HARLIN *has been composing music for games since 1999. He is currently the staff composer for LucasArts.*

sidhe
WE MAKE GAMES

RECRUITING NOW

We are looking for creative and talented developers who will contribute to innovative titles for console, PC and handheld. Challenge yourself, embrace the New Zealand lifestyle, and join a studio with a fresh perspective!

ARTISTS | PROGRAMMERS | ANIMATORS | GAME DESIGNERS | PRODUCERS

www.sidhe.co.nz/liveworkplaynz (ask about our ability to fast track your work visa)

NOW HIRING

PROGRAMMERS
DESIGNERS
ARTISTS

PRODUCERS
TESTERS
BRAND MANAGERS

For more info or to apply online visit:

Bethesda SOFTWORKS
www.bethsoft.com

id
www.idsoftware.com

BETHESDA GAME STUDIOS
bgs.bethsoft.com

ZeniMax ONLINE STUDIOS
www.zenimaxonline.com

GOOD JOB!

Hired someone interesting? Let us know at editors@gdmag.com!

SHAW GETS MOBILE

EX-MAXIS ONLINE LEAD JOINS NGMOCO AS LIVE PRODUCER

In March, longtime Maxis employee Caryl Shaw left the EA-owned company for Ngmoco. At Maxis she was most recently an online lead for SPORE, and she will take a similar role at Ngmoco—but as we discovered in our interview with Shaw, that title means different things at both companies.

BRANDON SHEFFIELD: How does your role differ at Ngmoco versus Maxis?
CARYL SHAW: It's different in some ways and not so different in others. My title is live producer, and that means I'm working with the games once they're live, which I also did when I was at Maxis as the lead of the online team. I live for launch days—there's nothing more exciting to me than shipping a game and watching customers play and talk about their experiences—and that's something that I loved at Maxis and already love at Ngmoco. WE RULE launched globally for the iPhone two days after I started at Ngmoco, so I got a good taste of launch right from the start.

SPORE was a classic PC boxed product, and the online features didn't generate any additional revenue after the game was purchased by a customer. At Ngmoco the games are free to download, and we generate revenue from in-App purchases. So now my job will really need to revolve around how to keep our customers happy with the game by creating new content and features that they can easily see the value in. I'll have a great opportunity to contribute directly to the success of the company now, and that can be really hard to experience at a big company like EA.

BS: What do you hope to be able to accomplish? I mean your personal goals within the company.
CS: I've been doing online development for the web and video games for 15 years, so one of my personal goals was to be able to step into a job and be able to hit the ground running. While I definitely have lots to learn about iPhone game development, I wanted to be able to really contribute from day one, which I think I have been doing. With a company that moves as fast as Ngmoco does, we need to have solid online systems to support almost instant scalability if a game takes off, so I'm hoping to be able to contribute some of the learning I have in that area. Oh, and I want to help the company make money! (laughs) The potential for success is pretty great, and I want to help contribute to that in any way I can.

BS: Is there any similarity between the prototype approach of Maxis on SPORE and the small game creation style of Ngmoco?
CS: The SPORE team made over 100 unique prototypes for SPORE and spent over a year doing them. I think it worked for that kind of game development—creating so much complicated tech from scratch just needed lots of time for trial and error and most of all for exploration. Ngmoco game development seems to move much faster—so far I haven't seen much in the way of prototyping —but what I do see people spending time thinking about, talking about and doing is live tuning of the games. Talking to the ELIMINATE guys about live tuning experiments feels a lot like talking to the SPORE engineers about using prototypes to understand player behavior. And with live tuning, I like the instant feedback you get from customers who are actually playing your game. It feels very real.

BS: What would you say to someone who has lost their job and is looking at the social or iPhone space?
CS: Well, I guess I'd tell them to not expect a lot of stability—these are platforms that are young and bound to have lots of ups and downs before they stabilize. But to me that's part of the excitement.

And do expect to work your ass off. You'll have to figure things out along the way and you'll have to try stuff, make mistakes and learn from them. And then try again. Like I said, part of the excitement.

new studios

Infinity Ward top execs Jason West and Vince Zampella, upon being fired from the Activision-owned company, have formed Respawn Studios and landed a deal with Electronic Arts, taking a number of IW employees with them.

Proper Games' design lead Andrew Smith has left the Dundee-based FLOCK developer to start his own studio, Spilt Milk, which will initially work on design consultancy and outsourcing projects to raise funds for the development of digital PC games.

The GRIN team that recreated BIONIC COMMANDO as BIONIC COMMANDO REARMED has recreated itself as Might and Delight, and plans to repeat its former success with new retro-inspired games.

who went where

Ubisoft's creative director Clint Hocking, who has shipped games including SPLINTER CELL, SPLINTER CELL: CHAOS THEORY, and FAR CRY 2 announced that he's leaving the company after spending more than nine years with the Montreal studio.

Two former senior-level employees from Sony Computer Entertainment Europe—Gareth Betts and Tony Buckley, who both worked on WIPEOUT—have joined Liverpool's Catalyst Outsourcing Limited as senior producers.

Former Sony Computer Entertainment Worldwide Studios vice president Jamie Macdonald has joined Codemasters in the role of senior VP of production, overseeing the company's production and engineering strategy.

Apple has reached into the world of video game journalism to head up editorial content for the game section of its iPhone and iPad App Store, hiring longtime IGN editor Matt Casamassina for the position.

Atari founder Nolan Bushnell has taken a position on the company's Board of Directors, along with online business veteran Tom Virden—as departed execs Phil Harrison and David Gardner resign their positions.

As Activision currently fields two separate lawsuits from former employees of its once-prized Infinity Ward studio, Activision Publishing CEO and president Mike Griffith is resigning his role.

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

Join us

Trion Worlds is the premier publisher and developer of connected video games. With groundbreaking new server platform capabilities, Trion delivers high-quality, fully dynamic and massively social content across video game genres, gaming platforms, and mass-market entertainment formats.

Headquartered in Redwood Shores, Calif., with development and technology studios in San Diego, Calif., and Austin, Texas, Trion is redefining interactive entertainment in the connected era.

For more information, or to apply, visit trionworlds.com/careers

Austin, Texas

QA Tester
Senior System Engineer

Redwood Shores, Calif.

Gameplay Programmer
Content Designer
Systems Designer
Vice President, Sales
Global Brand Director,
RIFT: PLANES OF TELARA™
3D/2D Artist (Marketing)

San Diego, Calif.

Art Outsource Manager
Lead Structure Environment Artist
Senior Terrain Artist

TRION WORLDS

radiator

WHAT PLAYING OUTSIDE CAN TEACH GAME DESIGNERS

RADIATOR VOL. 1
WWW.RADIATOR.DEBACLE.US
LUDOCITY
HTTP://LUDOCITY.ORG

Robert Yang's RADIATOR is an ongoing series of intensely personal, haiku-like Half-Life 2 mods that challenge the traditional FPS form. In subverting the shooter, Yang shows that games can present a far wider range of emotional experiences than they are typically given credit for.

Jeffrey Fleming: ***Can you tell me a bit about working with the Source engine?***
Robert Yang: Source is notorious for its convoluted 3D asset pipeline; importing a model from Maya into the engine, for example, is a crazy patchwork of plug-ins, .QC file edits, and batch files. Basic BSP-level sound playback control has been broken since HL1; you need hacks for the simplest function of stopping a sound that's already playing. The prop system is outdated—you can't scale or re-skin anything in the editor. Random entities don't work, the in-editor documentation is really insufficient, and the lack of accurate lighting previews in-editor is glaringly primitive.

That said, I would never switch engines. Source has the largest mod player-base of any of the major FPS engines. I mean, come on, who keeps UT3 or CRYSIS installed? No one. You'll see a lot of modders complain about the engine, how it won't let them do what you want—well, good design involves designing around constraints. If you want to make a Source mod that won't run on Source, don't blame the engine for it. Build around the problems and modify your design.

JF: ***It's interesting that you are using a FPS engine to create experiences outside of the shooter genre. What has been the biggest challenge in repurposing Source for the kinds of games that you want to make?***
RY: I don't want to do photorealism, but I'm stuck working roughly within Valve's style because a full non-photorealistic aesthetic would involve re-skinning and modeling new props, changing the weapon models, modifying particle effects, modifying NPC models ... it's a lot of work. Too much work. But I'm confident I'll come up with a style that'll look okay for my future projects and maybe compromise a bit less.

However, I firmly believe in staying with the first-person perspective for Source mods—that's what it was designed for and that's what it does best. Years ago, I saw an FPS mod for WARCRAFT III; it was really clever how they did it and they clearly put a lot of work into it, but it was really awful by any rational standard. There's still so much unexplored territory in the FPS genre, we should try to map out more of it as modders instead of trying to re-create our favorite genre in something else.

JF: ***Can you tell me more about your Video Game Level Design course?***
RY: At UC Berkeley, undergraduates can teach their own classes as part of a program called "DeCal." We're the biggest program of our size, with nearly 200 courses offered this spring.

I taught a DeCal for around three semesters about video game level design. For two semesters, I used COUNTER-STRIKE: SOURCE, but then I got frustrated with students who just ended up building cool-looking stuff without thinking about balance or level flow. The last time I taught it, it was a survey course that covered FPS deathmatch arenas, a TRACKMANIA track, and a STARCRAFT map—and it went really poorly because, again, it's hard to convince the average college student to think about theory instead of building a floating swimming pool or something.

So I decided that computers were the problem, and the best way to teach game design is to play outside. So I've been running a separate "playing outside" course for three semesters that's been an absolute blast. I think all game dev schools should have a "non-digital" component to their game design degrees, ideally a course for outdoor games and a course for board games.

JF: ***What types of games were being run outdoors? What design lessons did they teach?***
RY: I try to give a wide range of games for them to build a vocabulary with: we play team games that are kind of like leadership/trust exercises, we play capture the flag in libraries, we play scavenger hunts, we play rock-paper-scissors tag ... a lot of good stuff from ludocity.org. Then with that, the students have a common base to design their own outdoor games. So when they get into groups to workshop their own games, they can refer to this canon of sorts—"It's like that zombie game we played, except in teams of three."

I'd say the main lesson here is about the process of design: Games will have horrible problems that you won't realize until you playtest, and so it's okay to fail—that doesn't mean you're a bad designer. What makes a good designer is someone who is able to fix a broken game design to make it work.

The last unit of the "playing outside" class is always the most difficult because it's about how to embed a worldview into a game. We're familiar with novels and films having ideas and themes, but not games, so it takes a while to learn how to "read games." The example I always use in the class is a student game from the first semester called "Family Dinner."

In "Family Dinner," there are three teams: dogs, kids, and parents. Dogs and kids are allies—dogs circle along the periphery while kids try to sneak them lima beans from the table. Parents serve kids lima beans and try to catch kids in the act of sneaking them to the dogs. What usually ends up happening is the kids start finding loopholes in the rules and parents get really frustrated. Some parents start refusing to serve kids lima beans as a weird form of punishment. Some parents blame other parents for not being watchful enough. It brings out a lot of emotions and dynamics that are embedded in a family dinner—and obviously it represents a cynical worldview of a family dinner, but it's an interesting worldview that only emerges when you play the game.

And that, I think, is the highest form of a game, digital or non-digital: role-playing lets you imagine and understand some larger piece of the real world outside of yourself.

—Jeffrey Fleming

ACADEMY *of* ART UNIVERSITY

FOUNDED IN SAN FRANCISCO 1929 BY ARTISTS FOR ARTISTS

Gaming Artwork by Rob Arnold

Gaming Artwork Collaborative Student Project

Gaming Artwork Collaborative Student Project

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
Motion Pictures & Television
Multimedia Communications
Music 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

79 NEW MONTGOMERY ST, SAN FRANCISCO, CA 94105

Accredited member WASC, NASAD, Council for Interior Design Accreditation (BFA-IAD), NAAB (M-ARCH)

**Degree program not currently available online.*

ADVERTISER INDEX

Game Developer (ISSN 1073-922X) is published monthly by United Business Media LLC, 600 Harrison St., 6th Fl., San Francisco, CA 94107, (415) 947-6000. Please direct advertising and editorial inquiries to this address. Canadian Registered for GST as United Business Media LLC, GST No. R13288078, Customer No. 2116057, Agreement No. 40011901. **SUBSCRIPTION RATES:** Subscription rate for the U.S. is $49.95 for twelve issues. Countries outside the U.S. must be prepaid in U.S. funds drawn on a U.S. bank or via credit card. Canada/Mexico: $69.95; all other countries: $99.95 (issues shipped via air delivery). Periodical postage paid at San Francisco, CA and additional mailing offices. **POSTMASTER:** Send address changes to Game Developer, P.O. Box 1274, Skokie, IL 60076-8274. **CUSTOMER SERVICE:** For subscription orders and changes of address, call toll-free in the U.S. (800) 250-2429 or fax (847) 647-5972. All other countries call (1) (847) 647-5928 or fax (1) (847) 647-5972. Send payments to *Game Developer*, P.O. Box 1274, Skokie, IL 60076-8274. Call toll-free in the U.S./Canada (800) 444-4881 or fax (785) 838-7566. All other countries call (1) (785) 841-1631 or fax (1) (785) 841-2624. Please remember to indicate *Game Developer* on any correspondence.

KNIGHT TAKES QUEEN: THE ROOKENING

A PUBLISHER'S RESPONSE TO CHESS

HI FOLKS, WE'VE JUST FINISHED OUR EVALUATION OF YOUR GAME ENTITLED "Chess" a little while ago and we've got some feedback ready for you. It's clear that the core of a good game is here, but we think there's a ways to go in terms of its design. Don't take this the wrong way—we want this game to be a hit as much as you guys do, of course—but we're pretty concerned at the moment. Will people still be playing it a year from now? We aren't so sure.

I've broken out the areas we feel you should consider for improvement below.

STORY

>> While everyone here seemed to enjoy the complex strategic gameplay—for example, it seems like there are a lot of choices to make about what to do at any given moment—the game's total lack of narrative context really hurts it. Our focus group had no idea who the White King was, why they were fighting him, and so on.

Properly setting up a story at the beginning of the game will get players invested in the world of the game and want to win. If you don't feature this, there just doesn't seem to be any point to the game. To that effect, I asked our Hollywood screenwriter to come up with a unique idea that works with the design you guys already have, and I've reproduced his response here for your benefit:

"The Black King has had his empire stolen, his family killed, and was framed for a murder he did not commit. Now, he must claw his way back from the fiery pit of Hell itself to get his revenge upon the evil White King. Fueled by nothing but the fumes of his own seething rage, he will overcome impossible odds—and devious puzzles—to make them all pay for what they have done!"

This kind of unique storyline can really help to differentiate your title.

ACCESSIBILITY

>> Another problem is that the game doesn't teach itself to you as you play, something every good game should do. It's long past the time when people could be bothered to read manuals or get their friends to teach them how to play. A tutorial sequence would be good, introduced by the player's female sidekick over the radio. I'm sure you know the kind I mean. "Now, move that rook, recruit!" Et cetera.

REWARDS

>> Another problem that came to light in our focus group test was that the primary activity of the game—taking enemy pieces—just isn't very rewarding. Basically, the only real "action" is in getting your opponent's piece off the board, and it simply stops there. There's no real feedback to the player that this is a good thing, or that they should continue doing it.

How about adding points for every time you make a move? The climbing score will make players feel good about themselves, even if they are losing. The point accrual could tie into an unlock system of new pieces (some quick ideas—Wizard, Hydra, Slime Monster). These new pieces would be more powerful than the pieces you've got now, allowing players to feel as though they are getting more powerful with repeated play.

Related to that point, after a few intense games here at the office, everyone seems to concur that pawns are vastly underpowered. Even the name "pawn" makes them feel useless. Why would a player want to be in charge of pieces that are not powerful in any way? We'd suggest giving them some interesting abilities to compensate, such as the power to call in air strikes. Our focus groups really felt that ability in MODERN WARFARE 2 resonated with their core synergies.

MULTIPLAYER

>> In your GDD, you mention that the game is designed for a "sweet spot" of two players, but featuring only two-player versus seems really sparse. Games need to be more feature-rich than this to compete in the current market. Most multiplayer games feature up to sixteen players and four-player co-op. I don't think it would be too hard to implement into the rules you've already got.

There are more modes you could add as well, like some kind of "survival" mode where enemy chess pieces constantly spawn that you have to fend off. There are a lot of games doing that right now, and it seems like a good idea to increase the longevity of the experience. Without this sort of thing, people will just return the game to the store after they've played it once.

GRAPHICS

>> I hate to say this, but graphics do matter, and the look of this game just doesn't impress right now. Our focus group compared the game unfavorably to other games, such as *Hungry Hungry Hippos*, which has a lot of colors and a lot of things going on at once.

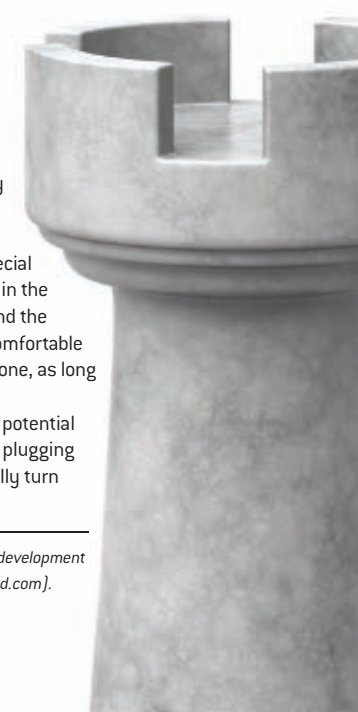
In contrast, the sparse black-and-white presentation of your game doesn't grab the eye at all. Especially since a great battle is supposedly going on—there's just no sense that it is occurring. We really need to up the ambience here, with raining arrows, fire, explosions in the distance, and so on.

Also, the pieces are really simple—very smooth with hardly any texture on them. I'd suggest getting some normal maps in there; take a look at games like GEARS OF WAR or UNCHARTED 2 and I think you'll see what I mean. You ought to be able to see the sweat dripping off the knight, the veins bulging out of the bishop's neck, and the nipples underneath the fabric of the queen's dress.

CONCLUSION

>> That about does it, I believe—oh yeah, one more thing. We'd also like to get you thinking about motion control possibilities, since platform holders are very interested in anything that features motion control right now. Games shouldn't be a sedentary activity after all. This is just a thought, but, related to the pawn problem noted above, perhaps pawns could move up to three spaces if a special bouncing target is double-punched precisely in the green zone, two spaces in the yellow zone, and the regular single space any other way? We're comfortable letting you guys work out the details on that one, as long as it's in.

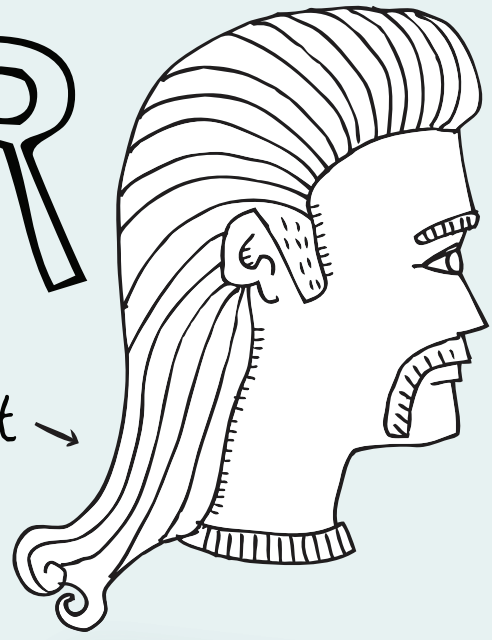
Anyway, all of this said, we really like the potential of what we're seeing in your game. Just keep plugging away on these issues, and I think it could really turn into something memorable!



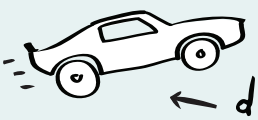
MATTHEW WASTELAND *writes about games and game development at his blog, Magical Wasteland (www.magicalwasteland.com).*

THERE ARE LOTS OF WAYS TO BE AN EVEN

COOLER GAME DEVELOPER



like grow a mullet →



← drive a sweet old Camaro

or you can use **BINK VIDEO**.

You get an amazing, super =FAST video and audio codec - all in a simple, clean API. And Bink Video runs on **EVERY PLATFORM**!

BINK VIDEO

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

RAD GAME TOOLS ©

www.radgametools.com
(425) 893-4300