



GAME DEVELOPER MAGAZINE

2010 FRONT LINE AWARD WINNERS A UNCEB

# Game Developer



# SEAPINE AGILE EXPEDITION™

## eXplore Agile in an eBook Adventure

The Seapine Agile Expedition is a **free** eBook-based learning series that provides a fun and informative overview of **Agile development**.

**New to Agile?** Discover how Agile boosts the performance of your entire team—developers, quality assurance, product owners, and ultimately, your customers—through timely communication and collaboration.

**Already using Agile?** Learn how to meet deadlines, respond to change, reduce risk, and deliver what you promise through Agile development powered by Seapine ALM.

Join the Adventure Today!



[www.seapine.com/gamebook](http://www.seapine.com/gamebook)

# gd

GAME DEVELOPER MAGAZINE

## CONTENTS.0111

VOLUME 18 NUMBER 1

### POSTMORTEM

- 22 HARMONIX' DANCE CENTRAL**  
Developing a launch title for a new piece of hardware is never easy, but simultaneously working with a new user input paradigm makes the task exponentially more difficult. Harmonix rose to the occasion, creating new UI and a new way to play. *By Kasson Crooker, Marc Flury, Matt Boch, Dean Tate, and Ryan Challinor*
- 38 PROPE'S IVY THE KIWII?**  
In this bonus postmortem, the IVY THE KIWII? team outlines how its newest game went from a minigame spawned by a new hire initiation, to a full-fledged product for Wii and DS. *By Kohei Takase and Yusaku Yamanaka*

### FEATURES

- 7 GDC PREVIEW GUIDE**  
Our GDC preview guide should help you get a handle on some of the early talks and summits coming out of the 25th anniversary of the biggest gathering of game developers anywhere. (GDC is run by the UBM Techweb Game Group, as is this magazine.)
- 15 FRONT LINE AWARDS**  
The 13th annual Front Line Awards are here, presenting the best in game tools as voted by you, the readers of *Game Developer* and the game development community at large. Categories include art, audio, middleware, engine, programming, and networking tools. *By Jeffrey Fleming, Chris Griffith, Tom Carroll, Alexander Brandon, Mark Cooke, Joe Allen, Adam Hatch, Nathan Fouts, and Amitt Mahajan*
- 31 YOUR MOVE**  
As hardware evolves, so too must each established genre adapt itself to new platforms. Here, an original COMMAND AND CONQUER designer discusses bringing the strategy genre to touch devices. *By Rade Stojasavljevic*

### DEPARTMENTS

- 2 GAME PLAN** *By Brandon Sheffield* [EDITORIAL]  
Developers Versus the Press
- 4 HEADS UP DISPLAY** [NEWS]  
ULTIMATE NEWCOMER, IGF China winners, and Propeller Awards.
- 42 TOOL BOX** *By Kenneth Lammers* [REVIEW]  
Allegorithmic's Substance Designer
- 46 PIXEL PUSHER** *By Steve Theodore* [ART]  
Scoped and Groped
- 48 THE INNER PRODUCT** *By Dave Mark* [PROGRAMMING]  
Getting More Out of Numbers
- 52 DESIGN OF THE TIMES** *By Damion Schubert* [DESIGN]  
The Many Forms of Game Narrative
- 54 EYE ON GDC** [GDC]  
Meggan Scavio and Brian Reynolds interviewed.
- 55 THE BUSINESS** *By John Graham* [BUSINESS]  
A Singular Journey
- 56 AURAL FIXATION** *By Jesse Harlin* [SOUND]  
Let's Do The Time Warp Again
- 57 GOOD JOB!** [CAREER]  
Ian Adams Q&A, who went where, and new studios.
- 58 EDUCATED PLAY** *By Jeffrey Fleming* [EDUCATION]  
SHADOWS OF ABIGAIL from the FIEA.
- 64 ARRESTED DEVELOPMENT** *By Matthew Wasteland* [HUMOR]  
Studio Pravda



# DEVELOPERS VERSUS THE PRESS

WHEN A STORY GOES WRONG, WHO'S TO BLAME?

## THERE'S BEEN A LOT OF TALK LATELY ABOUT

journalists twisting the words of developers, or representing them in a manner other than what the developer intended. In the past few months, Cliff Bleszinski, Ted Price, and others have complained that their words have been twisted, and that better standards of journalism need to be followed by the games press at large.

Most recently, Chris Hecker wrote about how he was lambasted for joking during a GDC rant that the Wii is a "piece of shit." His actual point was more nuanced than that, but that quote became the title of countless stories. The result was that readers got up in arms about Hecker's comment, and that caused them to reflect that upon his current work. Hecker got upset and felt the media took it the wrong way. The media in turn got defensive, saying "Well, he said it!" This cycle repeats itself frequently.

## WHO'S TO BLAME?

» No one party is at fault, here. It has always been in the nature of journalism to sum things up in headlines. Historically, the town crier was going to scream, "Troy falls!" even if the story is a lot deeper. It's only when the crowd gathers that the full story is told. Similarly, the headline of a story draws the reader in, and the body text should provide context. The trouble comes when the body text doesn't give enough context, or when the developer feels as though their point was actually quite different from what was finally presented.

There's also the issue of our current fast-paced news environment. Blogs are reporting things a mile per minute. Even the Associated Press is running "quick sketch" versions of stories, replete with inaccuracies, bad journalism, and typos, in order to get something—anything—up before the blogs get to it. About 15 minutes later, you'll see that same story a bit more refined, with better journalism and better grammar.

This necessity for speed has both created and been inspired by a new generation of information enthusiasts who feel they have grasped the entirety of a concept based on a couple of paragraphs. This has led to a whole lot of people not really reading articles very far past the headline, or first few sentences, before coming to a conclusion about a piece.

And I think most people know that if they say something particularly controversial, it's going to get reported by someone, somewhere. The illusion of privacy or closed circles has been shattered. Even Big Brother has had to wake up to this reality with the advent of WikiLeaks. In a world that includes all the above, developers might be inclined to stop talking to the press entirely, or to view them as the enemy. I'm in both camps, working for *Game Developer* magazine and *Gamasutra* while also developing games—and I really don't think that's the answer.

## BOREDOM AND WHOREDOM

» One solution that people have come to is to only give the press very select, very boring information, or info that does little more than talk up the product. "Here are our features. This is a new screenshot. Yes, we're very excited." This rarely translates into good stories, if stories are even written at all. This is not because journalists are voraciously going after hits, but because writing about something you're not interested in is a drag. It's with this in mind that journalists ask probing questions and try to dig something up. There may be some interest in scandal and controversy on the part of some—but in large part, they're just trying to make their own jobs interesting for themselves. And if they don't get good information, you don't get a good story.

The press is great for getting the word out about your game, and if you give them the right information, they respond well. But no matter what, if you say something like "Yeah, we really hate our publisher, but we've got to work with them, oh well," that's probably going to wind up being more interesting than a statement about how the game now has 25 guns instead of 20.

Of course, those controversial statements the press has gleaned can turn into precisely the scenario Hecker found himself in—feeling like his intent was removed from the final stories.

The worst possible outcome is that developers will feel like they can't say anything interesting for fear that it may be twisted into something other than what they intended. I think transparency and honesty is good for our industry, and I would hate for people to live in fear of getting information out. If developers were afraid to share information publicly, where would our industry be? Certainly a magazine like this one would not be able to exist.

## SAY WHAT YOU MEAN, MEAN WHAT YOU SAY

» In Hecker's case, he was giving a talk at GDC, which a lot of journalists attend. His rant was planned, and he knew what he was saying. He had already had experience with inflammatory statements (specifically about the Wii) being taken poorly, but made the decision to joke about the Wii being sub-par once more. I think the outcome of that should be pretty clear to anyone. It's not "wrong" to say the Wii is a piece of shit, if that's what you think. But you have to own it.

My conclusion is that developers should of course be able to say whatever they want—but choose your words carefully, as you might when speaking to your mother-in-law when she starts asking probing questions. It's not self-censorship, it's knowing your audience. And if you do make a strong statement, embrace it! If you truly believe in what you say, how can your detractors harm you?

—Brandon Sheffield, twitter: @necrosofty



United Business Media, 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](mailto:gamedeveloper@halldata.com)

FOR DIGITAL SUBSCRIPTION INFORMATION  
[www.gdmag.com/digital](http://www.gdmag.com/digital)

## EDITORIAL

**PUBLISHER**  
Simon Carless | [scarless@gdmag.com](mailto:scarless@gdmag.com)  
**EDITOR-IN-CHIEF**  
Brandon Sheffield | [bsheffield@gdmag.com](mailto:bsheffield@gdmag.com)  
**PRODUCTION EDITOR**  
Jeffrey Fleming | [jffleming@gdmag.com](mailto:jffleming@gdmag.com)  
**ART DIRECTOR**  
Joseph Mitch | [jmitch@gdmag.com](mailto:jmitch@gdmag.com)  
**PRODUCTION INTERN**  
Tom Curtis  
**CONTRIBUTING EDITORS**  
Jesse Harlin  
Steve Theodore  
John Graham  
Dave Mark  
Soren Johnson  
Damion Schubert

## ADVISORY BOARD

Hal Barwood Designer-at-Large  
Mick West Independent  
Brad Bulkley Neversoft  
Clinton Keith Independent  
Brenda Brathwaite Lolapps  
Bijan Forutanpour Sony Online Entertainment  
Mark DeLoura THQ  
Carey Chico Independent

## ADVERTISING SALES

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

## ADVERTISING PRODUCTION

**PRODUCTION MANAGER**  
Pete C. Scibilia e: [peter.scibilia@ubm.com](mailto:peter.scibilia@ubm.com)  
t: 516-562-5134

## REPRINTS

**WRIGHT'S MEDIA**  
Ryan Pratt e: [rpratt@wrightsreprints.com](mailto:rpratt@wrightsreprints.com)  
t: 877.652.5295

## AUDIENCE DEVELOPMENT

**TYSON ASSOCIATES** Elaine Tyson  
e: [Elaine@Tysonassociates.com](mailto:Elaine@Tysonassociates.com)  
**LIST RENTAL** Merit Direct LLC  
t: 914.368.1000

## MARKETING

**MARKETING COORDINATOR** Nahal Agahi  
e: [nahal.agahi@ubm.com](mailto:nahal.agahi@ubm.com)



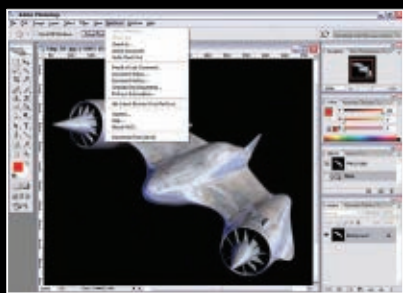


## Introducing P4GT, a productivity feature of Perforce SCM.

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

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

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



P4GT

**PERFORCE**  
SOFTWARE

**Download a free copy of Perforce, no questions asked, from [www.perforce.com](http://www.perforce.com). Free technical support is available throughout your evaluation.**



# newcomer arrives

/// TWENTY YEARS IN THE MAKING, THE EPIC COMMODORE 64 RPG NEWCOMER (WWW.NEWCOMER.HU) IS SET TO ARRIVE THIS SPRING, AND CAN BE ENJOYED ON MODERN HARDWARE RUNNING THE VICE EMULATOR. A SMALL TEAM WORKING OUT OF BUDAPEST BEGAN THE PROJECT BACK IN 1990 AND THE AMBITIOUS GAME HAS SEEN A NUMBER OF RELEASES AND REVISIONS OVER THE PAST DECADES, AT LONG LAST REACHING ITS FINAL ITERATION AS ULTIMATE NEWCOMER, WHICH WILL BE AVAILABLE IN BOTH ENGLISH AND HUNGARIAN. WE SPOKE WITH PROJECT LEAD ISTVAN BELANSZKY, AND ZOLTAN GONDA, NEWCOMER'S LEAD DESIGNER, WRITER, AND MUSIC COMPOSER TO FIND OUT HOW THIS ULTRA LONG-TERM PROJECT CAME TO BE.

**Jeffrey Fleming:** *Why the Commodore 64? Why after all this time?*

**Istvan Belanszky:** The question was raised quite often, and it took some ingenuity to find a legit excuse. Our team managed to create a highly complex game for a strongly limited and resource-constrained platform. And we did it for the hell of it!

**JF:** *Can you tell me more about your scripting system?*

**Zoltan Gonda:** Due to the demands of NEWCOMER's complex storyline,

conditional branch length limit on the 65xx CPU architecture. Even though 65xx assemblers display conditional branching instructions with an unsigned word absolute address operand, the operand is actually a signed 8-bit integer relative address.

How did it affect our scripting? 65xx conditional ops relative addressing means one byte can be shaved off per address operand, similar to a `jmp` op. So Andras implemented a `goto LABEL` script command by using a conditional branching op that always evaluates

limitations and inconveniences like these actually kept the script from bloating out-of-hand. As proof, this hack consistently decreased our compiled script size.

**JF:** *How are you organizing the English and Hungarian translations?*

**Istvan Belanszky:** Right now we are doing a final linguistic QA phase for the English version. For the ULTIMATE revision that we have been working on for the last seven years, the toughest linguistic task has been the Hungarian localization and keeping both language versions in sync while making a huge amount of in-game text changes.

The original version was released in Hungarian, back in 1994. Then we made a much expanded, more non-linear version in English-only called ENHANCED NEWCOMER and released it in 2001. One of the project goals for the current ULTIMATE revision is to provide Hungarian players with a localized version of the expanded and enhanced game.

Writing and linguistic QA for a non-profit, English/Hungarian multi-language game, made by a devteam with no native English speaker has been an enormous job for Zoltan Gonda who did most of the writing. During the last ten years there have been a number of native English speakers who have provided assistance for us, but it was in their free time, so we could not fully depend on them, and had to do most of the work by ourselves.

For the last seven and a half years I have been providing commercial game localization

services to various distributors and publishers in the Hungarian market, and I have involved Zoltan Gonda in commercial localization projects, so the professional experience that we've gained has been a great help to us when working on NEWCOMER.

**JF:** *How many people have worked on NEWCOMER, past and present?*

**Istvan Belanszky:** The original, Hungarian only version of NEWCOMER was made in 1990–1994 by a team of three (game design by Zoltan Gonda, graphics by Csaba Foris, and programming by Andras Lay). I was involved as a play tester and a RPG consultant, because I had a regular magazine column about tabletop role-playing and wrote some play aid articles in various Hungarian magazines back then. There were a couple more play testers as well. Back then the devs and testers were all based in Budapest.

The four of us made ENHANCED NEWCOMER, an English-only version, in 1997–2001. I joined the original devteam as a co-designer, and also organized part of the QA. The devteam was still Budapest-based, but play testing was done by an assortment of ESL folks from Germany and the Hungarian countryside, as well as a native English speaker from Canada.

For the ULTIMATE version, made both in Hungarian and in English 2003–2010, I took over the project lead role from Zoltan Gonda and involved a new programmer and graphic artist. So it is a team of three again, with me as a common link between the old and the new team. Members from the old team



the scripting system had to be flexible, expandable, easily compilable, and executable. It was mandatory to support as many of these things as possible at the design level without needing a programmer around.

**Istvan Belanszky:** Our “high-level, dedicated scripting language” is in fact a nice use of a macro assembler. However, there is a

true. As our script packages grew, Zoltan had to insert the occasional extra `goto LABEL` midway in a script package, and in a few instances had to build a chain of `goto` commands in order to reach the other part of the script beyond a single signed byte relative range.

Now, a good assembler or compiler would take care of that transparently by generating a jump table, but I can tell you that

help out with some stuff occasionally, though. The new team is an online one, because the new graphics guy is from the Hungarian countryside, and the new programmer has a daily job that requires him to travel. Most of the play testers from the previous revision are still helping us, and a couple of new testers from Germany as well.

**JF: Were there any software tools or hardware that you found particularly useful for developing NEWCOMER?**

**Istvan Belanzsny:** For the older versions we used the AsmOne Amiga IDE, running on our Amiga 2000 and Amiga 4000 workstations, along with a custom-built Commodore64 MMU cartridge for setting up software traps and breakpoints.

Our old game engine fully utilized the 64KB onboard RAM of the C64—so much so that Andras had to model the worst case CPU stack utilization scenario to find out how many bytes of the stack area would be safe to use as “normal memory.” And sure enough, those bytes ended up being used, storing sound tables. However our new, more feature rich game engine, programmed by Zsolt Kajtar, is smaller, so no such hacking is needed anymore.

For the ULTIMATE revision we use mostly free Unix software like bash, cc65, python, textutils, and the VICE emulator suite.

**JF: Now that there are C64 emulators on the Nintendo Wii and iPhone do you see yourself creating a version for those devices?**

**Istvan Belanzsny:** Ergonomics must be taken into consideration. NEWCOMER was designed for mid-to-large screen devices with a full keyboard. The GUI and text-heaviness makes it a bad fit for small screen devices, and the game’s complex keyboard controls would translate badly to a pointing-based device.

I’m not familiar with the C64 emulator solution used in some WiiWare titles, but I would guess that the emulation level is not cycle-exact and that those titles were modified to run on it. I might be wrong, but the C64 emulator for iOS by manomio does not seem to aim at a cycle-exact, down-to-the-silicon level of emulation, either.

The PC, Mac, and Linux installer bundles of our game use the VICE suite for C64 emulation, and even though we found VICE to be the best C64 emulator for our purposes, it is still has a way to go before it can provide an uncompromised down-to-the-silicon level emulation. Zsolt Kajtar, our programmer for the ULTIMATE revision, has contributed a new feature and some patches to VICE over the years, and a couple of those were specifically made to fix emulation flaws that we discovered while testing our game.

—Jeffrey Fleming

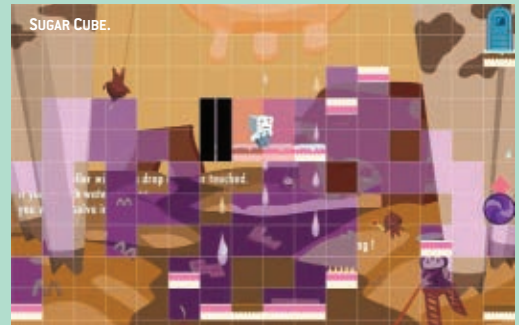
## winners of igf china 2010 announced

//// Organizers of the second annual Independent Games Festival China have announced winners for the Asian and Australasian indie games showcase in Shanghai, with South Korean developer Turtle Cream’s 2D tile-flipping platformer SUGAR CUBE getting the Best Game prize, among a host of other notable winners.

Following the announcement of the finalists last month, the teams attended a special awards show at the Shanghai International Convention Center during GDC China last night, where the winners of each category were revealed. (The jury elected not to select an Excellence In Technology award this year.)

**The winners of the 2010 IGF China awards are:**

- **BEST GAME**  
SUGAR CUBE (Turtle Cream, South Korea)
- **BEST MOBILE GAME**  
TRAIN CONDUCTOR 2 (The Voxel Agents, Australia)
- **EXCELLENCE IN AUDIO**  
SKILLZ: THE DJ GAME (Playpen Studios, Hong Kong)
- **EXCELLENCE IN VISUAL ARTS**  
BUTAVX: JUSTICE FIGHTER (Nekomura Games, Singapore)
- **BEST STUDENT GAME**  
THE WHITE LABORATORY (Huazhong University of Science & Technology, China)
- **EXCELLENT STUDENT AWARD**  
DEAD STEEL (Media Design School, Auckland, New Zealand)
- **EXCELLENT STUDENT AWARD**  
PONLAI (National Yunlin University of Science and Technology, Taiwan)



- **EXCELLENCE IN VISUAL ARTS**  
BUTAVX: JUSTICE FIGHTER (Nekomura Games, Singapore)
- **BEST STUDENT GAME**  
THE WHITE LABORATORY (Huazhong University of Science & Technology, China)
- **EXCELLENT STUDENT AWARD**  
DEAD STEEL (Media Design School, Auckland, New Zealand)
- **EXCELLENT STUDENT AWARD**  
PONLAI (National Yunlin University of Science and Technology, Taiwan)

Along the way, high-quality submissions for the second iteration of the event—a

newly formed sister competition to the main yearly Independent Games Festival in San Francisco—were received from multiple Chinese provinces, Hong Kong, Taiwan, Singapore, South Korea, Australia, New Zealand, Iran, India, and beyond.

Winners were chosen by a panel of distinguished local judges, including representatives from Shanda Games, Tencent, IGDA Shanghai, TipCat Interactive, and more. IGF China finalists were invited to Shanghai for the Game Developers Conference China event from December 5th to 7th, where they showed their games at a special Pavilion on the Expo Floor.

—Staff

## INDIEPUB, SXSW PARTNER FOR INDEPENDENT PROPELLER AWARDS



//// Publisher Zoo Games revealed that its online indie gaming community IndiePub Games will host its first Independent Propeller Awards competition—the winners will be announced on March 13 at SXSW ScreenBurn, the gaming portion of the SXSW Interactive Festival.

IndiePub will accept new and original PC/Mac/browser-based submissions for the contest starting today through February 18th. A panel of judges will select the finalists in several categories: Best Art, Best Audio, Best Design, and Technical Excellence.

Organizers will pay for finalists’ flights and accommodations so they can attend

SXSW and demo their games during the event. IndiePub will offer a possible publishing deal with Zoo and \$150,000 in total prizes, including \$50,000 for the grand prize and \$25,000 for subcategory winners.

“Like SXSW, IndiePub Games was founded to create a channel for artists where they can share their work and their ideas,” says Zoo Games CEO Mark Seremet. “We are thrilled to partner with SXSW Interactive to showcase the inspiring talent and innovation found within the independent gaming community through the inaugural Independent Propeller Awards.”

—Staff

# UNREAL TECHNOLOGY NEWS

BY Mark Rein  
Epic Games, Inc.



## FROM MOBILE TO MOTION, UNREAL ENGINE 3 POWERS NEW EXPERIENCES

Over the past year, Unreal Engine 3 has continued to power some of the best titles in the industry and produce success stories for the developers behind them across a wide range of genres and sizes. In addition, we've released Unreal Engine 3 for mobile, showing it onstage at the Apple Special Event in September with an introduction by Steve Jobs himself. Steve called it "remarkable," and one journalist's reaction was "everything's just changed," which I think sums it up pretty well.

With more than 1.5 million downloads in the iTunes App Store, the reaction to our Epic Citadel demo shows that people are enthralled by the kind of experiences that UE3 will soon be powering on mobile platforms.

Our recent release of Unreal Development Kit with iOS support opens up free access to our toolset to anyone looking to create iOS games, and ChAIR's recently launched *Infinity Blade* is just one more example of what can be achieved with UE3 on handheld devices.

But mobile isn't the whole story.

We're powering motion gaming. Microsoft's Kinect is already enjoying wide acclaim and early success. *Kinect Adventures*, the game that comes with every Kinect sensor, is powered by UE3.

We're on PlayStation Move as well. In their E3 keynote, Sony revealed *Sorcery*, a stunning first-party Move title coming in the New Year. Additional ground-breaking motion titles are in development using UE3 as I write this.

This year, we also added Scaleform GFx, the world's leading user interface technology, and the full package comes free with UE3. Scaleform has an awesome UE3-based in-the-world 3D interface demo you've got to see for yourself.

Speaking of 3D, at various tradeshows we have been showing stereoscopic 3D on PC, Xbox 360 and PS3, and even auto-stereoscopic 3D on a prototype Android mobile device powered by NVIDIA's Tegra 2 hardware. *Batman: Arkham Asylum™ Game of the Year Edition* shipped in stereoscopic 3D on Xbox 360, PS3 and PC with support from Darkworks' TriOviz technology, which has now joined our Integrated Partners Program. Look for several big 3D titles powered by UE3, including Disney's *TRON: Evolution*

and the next *Mortal Kombat* title, both of which had great 3D showings at E3.

There's so much more to talk about, but the bottom line is that regardless of your genre or budget, if you're looking for the best game engine technology on PC, console and mobile platforms, then we should talk. Our door is always open.

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 along with entry into the Hall of Fame. UE3 has won three consecutive *Develop Industry Excellence Awards*.

Epic is the creator of mega-hit "Unreal" series of games and the blockbuster "Gears of War" franchise.

Follow @MarkRein on Twitter.

## UPCOMING EPIC ATTENDED EVENTS

DICE Summit  
Las Vegas  
Feb. 9-11, 2011

GDC  
San Francisco  
Feb. 28-Mar. 4, 2011

Please email: [mrein@epicgames.com](mailto:mrein@epicgames.com) for appointments.

WWW.EPICGAMES.COM







# GDC

## 2011 PREVIEW

////////// Ah, GDC. It's that time of year when every young boy and girl's thoughts turn to flights of fancy—by which I mean big game contracts, career moves, camaraderie, and the art and science of video games, of course.

Truth be told, it seems as though the importance of the “big game contract” has diminished significantly over the last few years. We used to talk about games getting back to the old 80s idea of the bedroom programmer—someone who can make a living developing and releasing games on their own. But I don't think anyone imagined it would actually go so far.

With tiny, agile social game companies still doing well, intelligent iOS developers cleaning up, and certainly not forgetting the MINECRAFTS of the world, we are looking at a game industry that is increasingly about micro-studios exploring and expanding niches, rather than a culture where the \$60

million blockbuster is the only path to success. But far from being fragmented, these smaller niches seem to be moving toward convergence, with a new sort of definition of “multiplatform” emerging—one that bridges consoles, PC, and phones in ways that previously sounded like marketing jargon.

This year's GDC reflects that shift in its summits especially, with social games, smartphones, indie games, and other specific areas covered. The main conference hasn't been immune to industry shifts either, with talks discussing network play and monetization alongside more traditional art and tech talks.

As the industry changes, so too does the conference. And in these pages we've provided an early look at some of the best talks and summits, though more will be available on [gdconf.com](http://gdconf.com) as this issue ships. We look forward to seeing you there!

## 15 GAMES IN 15 YEARS

Stone Librande (EA/Maxis)

/// Maxis creative director Librande has been making a board or card game per year to entertain his children, as they grew from age 3 to 18. What began as a simple exercise to teach and engage with his kids wound up teaching him important lessons about game design. In this talk, Lebrande will outline how you don't need a big budget, expensive tools, or even a studio to make an engaging game. All you need is inspiration, and the drive to create.

## A REQUIREMENTS SPECIFICATION TEMPLATE FOR INTERACTIVE STORYTELLING

Ernest Adams (International Hobo)

/// The always-entertaining Ernest Adams will here discuss his vision for a requirements specification template as regards storytelling in games. That is to say, a game writer should outline what their story needs to achieve, which includes but is not limited to game design requirements, player desires, and overall goals of narrative communication. One might be skeptical that a template would work for every game, but using it as a skeletal framework, or even simply a guide for your thinking could prove very useful.

## FAULTY THINKING: BECOMING A BETTER PRODUCER BY UNDERSTANDING FAULTS IN HUMAN THOUGHT

Chuck Hoover (Schell Games)

/// There's always someone on the team you just can't get along with, and you may be at a loss as to why. But as a producer, you've got to be able to efficiently interact with—and mediate between—everyone you're working with. In this deep-level production talk, Chuck Hoover of Schell Games will discuss how errors in perception, both the team's and your own, can limit your ability to interact well with others. But understanding those perception faults can be a great tool for facilitating communication.

## HOW TO CONSTRUCT A GAME JAM IN A BOX

Foaad Khosmood (IGDA/Global Game Jam/UC Santa Cruz)

/// Game jams are increasing in popularity, growing alongside the explosion of indie game development. But managing game jams is hard, and nobody really enjoys doing it. Khosmood suggests an online resource with automated tools for game jam creation and management, which could handle everything from user

registration to sponsorship, to storing the actual finished games. With this proposed system, could game jams approach some semblance of standardization?

## I SHOT YOU FIRST: NETWORKING THE GAMEPLAY OF HALO: REACH

David Aldridge (Bungie)

/// Network accuracy and reliability is perhaps the most important aspect of any real-time multiplayer game. But as everyone who's tried to manage this knows, it's not so simple, when you have a lot of players all expecting the game to be "fair" to them. Bungie's David Aldridge will share insight into the company's network inspection tools for HALO: REACH, while also outlining opportunities for optimization and removal of lag. Aldridge says that after this talk, "attendees will never again refer to game networking as sockets programming."

## MEGA MESHES MODELING, RENDERING AND LIGHTING A WORLD MADE OF 100 BILLION POLYGONS

Ben Sugden (Lionhead Studios) and Michal Iwanicki (Lionhead Studios)

/// As Lionhead's early Kinect demo turns into the full-fledged product MILO AND KATE, the company is looking to innovate in the visual pipeline space as much as in the interactive AI storytelling space. For this title, Lionhead has moved away from traditional implementation, and toward building the entire world using only mesh sculpting tools. This talk brings virtual texture streaming, complex mesh algorithms, and billion-poly meshes into one unified process. Lionhead's Sugden and Iwanicki will share common implementation pitfalls, and also illuminate how to set up such a massive structure for proper lighting.

## MORE PIRATES ON A BURNING SHIP AND OTHER LEADERSHIP CHALLENGES

Laura Fryer (WB Games Seattle)

/// Continuing on from her excellent talk at the IGDA Leadership Forum, Fryer will discuss good management, and how it extends from the top to the bottom of your team structure. This is not just hand-waving. Through her extensive experience at big companies, such as shipping Xbox 360 and GEARS OF WAR at Microsoft, and her current VP position at WB Games, Fryer has actual processes that you can implement to get your team talking more, communicating better, and even trusting you as a manager or producer more. That said, you can't always make them like you!

## REMAKING A CLASSIC: THE GAME DESIGN OF STARCRAFT 2

Dustin Browder (Blizzard Entertainment)

/// STARCRAFT II was in development for a very long time. It seems simple enough to expand upon the success of one of the greatest real-time strategy games of all time, but when consumer expectations are so high, a lot is at stake. Blizzard kept poking at it until they really had something, not only in terms of the core game, but a strategy for expansion going forward. This talk from Blizzard's Browder will lift the curtain on the company's game design techniques, from real-time strategy design to multiplayer concerns—all the way through to unique aspects such as designing for e-sports.

## SEVEN WAYS A VIDEO GAME CAN BE MORAL

Richard Rouse III (Ubisoft Montreal)

/// Moral choice in games has been a hot topic since the early days of interactive simulation. How do you make moral or humanistic choices matter to players in a medium where dispatching hundreds of foes in a gameplay session is the norm? Richard Rouse III will dissect examples of successful moral situations from other written fiction, television, and beyond, then discussing how these ideas can be applied to game design. This talk follows up his successful discussion last year of whether games can make us cry.

## THE ONE-HOUR VIDEO GAME MBA

Charlie Cleveland (Unknown Worlds Entertainment)

/// All too often developers ignore the equally challenging game of business. As a result, they can end up ceding control of their companies to fresh-faced MBAs who by nature steer toward far less interesting waters. Take matters into your own hands (or at least understand what the bean counters are talking about) by learning business administration on your own. You figured out inverse kinematics. A profit/loss statement will be a breeze.

## BEYOND FACADE: PATTERN MATCHING FOR NATURAL LANGUAGE APPLICATIONS

Bruce Wilcox (TellTale Games)

/// Unique games such as FACADE and SCRIBBLENAUTS have shown the creative possibilities found in natural language interfaces. Here Bruce Wilcox from Tell Tale Games will explore various approaches to natural language handling including AIML, FACADE's ABL, and ChatScript. The goal is to

create characters that can have believable conversations with the player as well as interpret commands delivered in natural language. A renaissance in adventure games may be just around the corner.

### DATA ANALYSIS ON PLAYER BEHAVIOR IN EVE ONLINE

Brynjlfur Erlingsson (CCP Games)

/// CCP Games has been using data mining techniques to great effect in EVE ONLINE. From defeating Real Money Traders to improving the game experience for players, the efficient collection and rapid analysis of metrics has allowed CCP Games to build a responsive, living online world. This session will detail CCP games' best practices for data collection and the most cost effective means for storing and interpreting the collected data.

### DESIGNING GAMES FOR THE 43-YEAR-OLD WOMAN

Chris Trottier (Zynga)

/// In the past, most game developers never really considered how to reach a middle-aged, female demographic. Moms just didn't seem all that interested in shooting giant killer robots. Well, guess what? They're still not looking to kill robots, but moms are increasingly interested in games that fit their lifestyle. Chris Trottier calls them accidental players, and savvy developers who are willing to throw out some long-held design standards to create games that better match the expectations of a forty-something female audience can find real success.

### APPLYING AUTOMATED METRICS TO GAME PRODUCTION

Larry Mellon

/// Collecting data is easy. Shaping it into something coherent and actionable is the hard part. In this roundtable session Larry Mellon will discuss the design of a flexible metrics infrastructure and how to apply it to player, production, and engine performance analysis. Effectively integrating metrics into your team is key, and the session will also examine how using metrics can fundamentally change how people work.

### BEYOND HORROR: ART DIRECTING DEAD SPACE 2

Ian Milham (Electronic Arts Redwood Shores)

/// In a hit-driven market, big budget games require mass appeal. Gameplay and visuals need to be carefully tuned to reach the widest possible audience. Here, Ian Milham, art director for the DEAD SPACE franchise will explore the



evolution of DEAD SPACE's art design toward higher production values and wider appeal. Recognizing and incorporating lessons learned from the first title are key, as well as managing art direction on a large scale.

### DIVERSITY WAIT, WHAT HAPPENED?

Sheri Graner Ray (Schell Games)

/// A decade ago the game industry fully recognized the need for diversity in its ranks. However, ten years later the sad truth is that little has changed, and minorities currently make up less than 10 percent of the workforce. Here Sheri Graner Ray from Schell Games will look at the history of the diversity movement in the games industry and where it lost its way. Rather than just focus on what went wrong, Ray will articulate a strategy for reinvigorating the call for diversity and how it can become a real asset for business and creativity.

### SOUND SYNTHESIS IN CRACKDOWN 2 AND WAVE ACOUSTICS FOR GAMES

Nikunj Raghuvanshi (Microsoft Research)

/// Ruffian Games' CRACKDOWN 2 was the showcase for a number of unique sound design techniques. In this session, Nikunj Raghuvanshi from Microsoft Research will explain two of the physics-driven audio techniques that helped bring texture to the game's world. First, he will detail the procedural sound synthesis process that the game utilized in which widely varied permutations of a single sound sample could be generated in real time. The procedural method can be used to great effect for impact sound effects, such as object strikes and footsteps. He will also cover the wave-based

sound propagation technique that was used to pre-compute the game's environment-based acoustic effects such as reverberation, scattering, and sound obstruction; a process that can reduce CPU workload while bringing increased richness to the player's aural experience.

### FAST AND EFFICIENT FACIAL RIGGING IN GEARS OF WAR 3

Jeremy Ernst (Epic Games)

/// Here, Jeremy Ernst from Epic Games will detail the facial rigging pipeline used in the development of GEARS OF WAR 3. High quality rigs that can be utilized quickly are the key to triple-A production values. He promises to go deep into both the technical aspects as well as the theory behind Epic's methods so that attendees can apply the knowledge regardless of which software package they work in.

### IDENTIFYING NEW TALENT IN TURBULENT TIMES

Alex Evans (Media Molecule), Ian Livingstone (Eidos), Mark Morris (Introversion Software), Adam Russell (University of Derby), and John Sear (We Are Colin)

/// The British are coming! In this fast-paced panel session some of the UK's leading game developers (including the co-founder of Games Workshop, Ian Livingstone) will debate the issues surrounding the incorporation of new ideas in a business environment that is constantly in flux. Are the lessons of the past meaningless in today's game industry? Or, are the entrepreneurs and iconoclasts reliving the same mistakes of their forebears? Attendees can join in the debate live via Twitter and vote on each motion. Our vote is on the old guys.



# CCP

## USES MORPHEME

“Morpheme has become an indispensable tool in delivering vivid expressiveness to the avatars populating EVE and beyond.”

Thor Gunnarsson, VP of Business Development at CCP



**morpheme**  
advanced animation system



[www.naturalmotion.com](http://www.naturalmotion.com)

## SUMMIT'S

GDC is home to a variety of summits, which adapt to the changing face of the games market, and its multiple facets. This year, summits encompass artificial intelligence, education, smartphone development, indie games, serious games, localization, and social games.

Below, we've outlined each summit, as well as a sampling of the key talks within each. More talks will be added after press time, so be sure to visit [www.gdconf.com](http://www.gdconf.com) for more.

### AI SUMMIT

The AI Summit at GDC is organized by the AI Game Programmers Guild, and features lectures from some of the industry's top AI programmers, who will discuss the key architectures and problems with modern games, as well as where AI will go moving forward. These lectures are directed at intermediate to advanced programmers who hope to gain a better understanding of AI, though most anyone interested in the subject will surely gain crucial knowledge from any of the numerous sessions.

#### PEOPLE IN YOUR POCKET: HIGH-QUALITY AI ON MOBILE DEVICES

Richard Evans (Little Text People), Emily Short (Freelance Writer and Consultant), and Andrew Stern (ngmoco)

/// While simulating human behavior on PC and consoles can require huge amounts of animation and graphic resources, developing on mobile devices allows developers to focus less on appearance, and more on behavior. This lecture focuses on the unique challenges of mobile development as regards AI design; with lower expectations for high-quality visuals, developers can focus on generating high quality behavior, even under strict technical limitations.

#### USING RANDOMNESS IN AI: BOTH SIDES OF THE COIN

Dave Mark (Intrinsic Algorithm), Brian Schwab (Blizzard Entertainment)

/// When players begin to notice AI becoming repetitive or predictable, it can severely harm a game's longevity. To alleviate this risk, adding elements of randomness can keep a game interesting, just so long as it continues react in a believable way to the player's actions. This lecture examines how players interpret randomness as more complicated game systems, as well as how to avoid some of the problems that random elements can generate.

#### LAY OF THE LAND: SMARTER AI THROUGH INFLUENCE MAPS

Damián Isla (Moonshot Games), Alex Chamandard (AiGameDev.com), Kevin Dill (Lockheed Martin)

/// When placing an AI-controlled character into a game environment, it needs to be able to successfully navigate its surroundings in believable and interesting ways. This lecture

will explain how to create more environmentally aware AI using influence maps, detailing their use for spatial analysis and position evaluation for strategy and action games.

### GAME EDUCATION SUMMIT

The GDC Education Summit aims to bring together educators and working professionals to share their experiences and ideas to benefit game education in the classroom, and to give advice to current and former students. Lectures will highlight experimental and inventive educational approaches that conference attendees can bring back to school, whether they are educators or students. Other sessions will focus on collaboration and how students should aim to succeed in the industry after graduation.

#### BRINGING GAMES USER RESEARCH INTO OUR EDUCATIONAL PRACTICE

Katherine Isbister (NYU's Polytechnic Institute)

/// In order to understand audiences and users, developers must understand how to conduct effective research to aid design decisions. This talk will examine games education programs at four different universities, and how research techniques can be woven into the curriculum and also into the general atmosphere of a program.

#### EDU TO INDIE: TEACHING STUDENTS SELF-RELIANCE

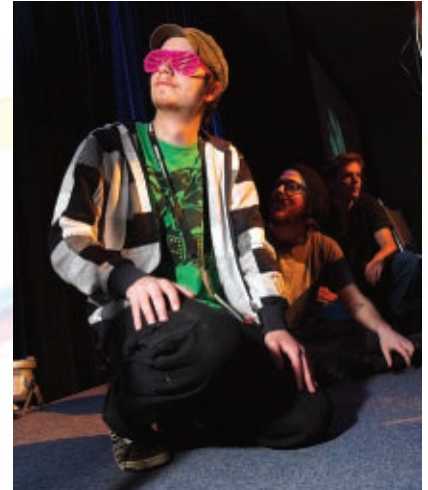
Jeremy Gibson (USC - Interactive Media Division)

/// While many students may dream of starting their own company after graduation, achieving such a goal can prove incredibly difficult. In this talk, USC faculty member Jeremy Gibson discusses how students should prepare for a successful indie future, citing examples of past student success.

#### COLLABORATION ACROSS DISCIPLINES AND PROGRAMS

Eric Zimmerman (Independent), Michael Sweet (Berklee College of Music), Matthew Weise (Singapore-MIT GAMBIT Game Lab), Tracy Fullerton (USC School of Cinematic Arts) and Colleen Macklin (Parsons the New School for Design)

/// Any development team knows collaboration is crucial for any project, and this panel will examine how an emphasis on coordination and communication can be used in education



programs to teach students how to bring together the multi-dimensional aspects of game development.

### GDC SMARTPHONE SUMMIT

In previous incarnations, GDC has featured a mobile summit, and later an iPhone summit. This year will mark the first annual GDC Smartphone Summit, which looks more broadly at the iPhone and iPad, Android OS devices, BlackBerry, and other similar handheld platforms. The mobile space's top developers will share their ideas and practices for developing mobile games, focusing on emerging business and marketing strategies, as well as design tips for taking advantage of these successful platforms.

#### CHANGING THE WAY WE MAKE GAMES: LESSONS LEARNED IN THE TRANSITION FROM AAA TO MOBILE DEVELOPMENT

Vijay Thakkar (Newtoy)

/// With the rise of mobile and social gaming has come a host of changes to the way we make games. Based on experience from working with AAA titles and social games, Vijay Thakkar will examine how to improve development for new or unexplored platforms.

#### THE NGMOCO LIVE TEAM PLAYBOOK: LIFE AFTER LAUNCH

Caryl Shaw (ngmoco)

/// With a slew of online-enabled iOS and Android games under its belt, developer ngmoco

certainly has experience running live services on mobile platforms. This lecture will reveal the company's lessons learned and essential strategies for managing in-game economies, attracting lapsed players, and more.

## CREATING A SOCIAL, LOCATION-BASED SMARTPHONE GAME IN 5 WEEKS

Jennie Lees (Google)

/// In order to keep up with the short release cycles of mobile games, developers need to be quick on their feet, with agile development strategies and practices. This session takes a look at how a team of four strangers took a location-based game from concept to prototype in just five weeks.

## INDEPENDENT GAMES SUMMIT

Serving as the voice of the independent developer at GDC, the Independent Games Summit features lectures, postmortems, and roundtables from some of the most influential indie developers, including many former and current Independent Games Festival finalists and winners. The summit lifts the curtain on the wide range of knowledge and experience within the indie community, with sessions focusing on design, production, philosophy, business, marketing, and more.

## THE HUMBLE INDIE BUNDLE

John Graham (Wolfire)

/// The Humble Indie Bundle, a pay-what-you-want charity promotion organized by a handful of indie developers, took in over 1.3 million dollars in just 11 days. This lecture provides an inside look at the inspiration and process that led to the bundle's creation, as well as a breakdown of user contribution data and tips for operating a successful promotion.

## THE FULL SPELUNKY ON SPELUNKY XBLA

Derek Yu (Independent) and Andy Hull (Story Fort)

/// In this panel, the creators of the roguelike SPELUNKY will detail the process and challenges of bringing the celebrated PC title to Xbox Live Arcade, while also discussing the benefits of freeware titles and procedurally-generated content.

## GAME DESIGN BY ACCIDENTS

Steph Thirion

/// Programming is an interesting discipline; a single change in a line of code can drastically affect a game in its entirety. This lecture will demonstrate how these changes can be beneficial, and how coding can inspire unexpected design ideas.

## LOCALIZATION SUMMIT

With new markets emerging all over the world, high quality localization is more important than ever for publishers who wish to take advantage of these untapped territories. The Game Localization Summit at GDC is supported and organized by the IGDA Game Localization SIG, and features lectures, panels, and postmortems intended to help developers and publishers understand how to effectively localize titles in terms of both planning and efficiency. As of press time only one talk is announced, but there will be several more at the show itself.

## GAME MARKETS IN THE MIDDLE EAST: OPPORTUNITIES AND CHALLENGES

Oded Sharon (Corbomite Games) and Mahmoud Khasawneh (Quirkat)

/// The Middle East is not often considered fertile ground for the industry, but its game market has seen substantial growth over the last five years, and this lecture will examine its untapped potential and present examples of the most successfully localized games for the region.

## SERIOUS GAMES SUMMIT

The eighth annual Serious Games Summit focuses on the evolution and growth of projects using video game technology for purposes other than entertainment. This summit features sessions on using game resources for training, health, education, behavior change, science, advertising, and more. This year's summit is built around changing behavior and participatory action, with day one focusing on health and healthcare, while day two will focus on the rising trends of "gamification," the process of building game-like incentives into non-game applications.

## HYPERLOCAL GAME DESIGN: CONNECTING SOCIAL CURRENCY TO REAL WORLD CURRENCY

Kati London (Area/Code)

/// This talk examines the community-specific social game MACON MONEY, which is structured based on player locations and real purchases within the town of Macon, GA. Its limited scope and unique constraints helped the team find new ways to innovate and link real world currency to the game's social infrastructure.

## VIDEO GAME PLAY AS NIGHTMARE PROTECTION

Jayne Gackenbach (Grant MacEwan University)

/// In this lecture, Jayne Gackenbach of Grant MacEwan University will explain how video games might help prevent nightmares, as they train us in handling threatening situations, and may help us gain control over the circumstances of our dreams.

## GAMIFICATION 201—30 TACTICS IN 30 MINUTES

Rajat Paharia (Bunchball)

/// In order to parse the most effective ways to implement gamification solution for companies of all sizes, this panel will answer some of the most common questions about gamification and how to execute it properly.

## SOCIAL AND ONLINE GAMES SUMMIT

Social and online enabled have matured rapidly over the last few years, with Facebook titles, web-based games, and downloadable persistent MMOs among the most prevalent examples. The Social and Online Games Summit will take a close look at the ever-evolving space and examine what does and doesn't work there, as well as where these types of games are headed in the future. Sessions will feature the industry's leading professions as well as up-and-coming success stories in lectures and panels on design, technology, business, and more.

## BEHAVIORAL ECONOMICS AND SOCIAL GAMES

Peter Fishman (Playdom)

/// In order to examine how and why we make certain decisions, Playdom economist Peter Fishman will explore behavioral biases in decision making, and how these innate biases influence the mechanics and features designers put into social games.

## USING YOUR FRIENDS: IDENTIFYING THE TOP INTERACTION MECHANICS IN CURRENT SOCIAL GAMES & MEDIA

Mia Consalvo (MIT)

/// While social gaming has become a prominent part of the modern industry, some of the most popular social games feature very limited player interaction. This panel will take a look at the various social mechanics found in these titles, and will point to the myriad interactions these games can support.

## RETAIN YOUR CUSTOMERS BY LETTING GO: FIXING THE SHORT TERM TACTICS THAT THREATEN PLAYER LOYALTY

Scott Rigby (Immersyve)

/// While some social games can attract a large number of players with relative ease, encouraging them to keep playing has become the real challenge. This lecture focuses on developing long term motivation in social games, and how to collect and examine behavioral and motivational metrics to better gauge player retention.



## THE 11<sup>TH</sup> ANNUAL

# game DEVELOPERS CHOICE awards

The **Game Developers Choice Awards** are the premier accolades for peer-recognition in the digital games industry. Every year at GDC, the Choice Awards recognize and celebrate the creativity, artistry and technological genius of the finest developers and games created in the last year.

**AWARD FINALISTS WILL BE ANNOUNCED IN FEBRUARY.**  
Stay updated at: [www.gamechoiceawards.com](http://www.gamechoiceawards.com).

### AWARDS ARE PRESENTED IN THE FOLLOWING CATEGORIES:

#### 2010 AWARD CATEGORIES

- Best Audio
- Best Debut
- Best Downloadable Game
- Best Game Design
- Best Handheld Game
- Best Technology
- Best Visual Arts
- Best Writing
- Best New Social/Online Game
- Innovation
- Game of the Year

#### SPECIAL AWARDS CATEGORIES

- Lifetime Achievement
- Pioneer
- Ambassador

PRESENTED BY

**Game Developers**  
Conference

PRODUCED AND HOSTED BY



**gamedeveloper**



# BLIZZARD

ENTERTAINMENT



## 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](http://jobs.blizzard.com)

Visit us in the GDC Career Pavilion at Booth #2432.

©2011 Blizzard Entertainment, Inc. All rights reserved. World of Warcraft, Diablo, StarCraft and Blizzard Entertainment are trademarks or registered trademarks of Blizzard Entertainment, Inc., in the US and/or other countries.





# FRONT LINE AWARDS 2010

//////////////////// Game Developer consistently strives to provide developers with information, news, and articles that pertain directly to them. The Front Line Awards are an official way for us to honor a specific aspect of the industry: the tools that developers need to do their jobs.

Each year, we look at the lineup of new products and new releases of important tools in the categories of game engines, art tools, middleware, audio tools, and programming/production tools. Recognizing the evolving nature of game development, we also added a Networking tools category to the awards that reflects the growing importance of connected gaming.

The Front Line Award winners on display here represent the most innovative, user-friendly, and useful products from behind the scenes of the world's best video games. In addition to the Front Line Award winners, we also want to pay special tribute with our Hall of Fame Award to a product that has made a lasting impact on the game industry year after year (and which was also not eligible to win in its specific category).

Nominations for this year's Front Line Awards were open to all new software products and new versions of software products related to game development released between September 1, 2009 and August 31, 2010. In determining the winners of the 2010 awards, we went through a multistep process. Open nominations were held in October, and from that list, we consulted with our advisory board to narrow down the results to five entries in each category. We then handed the finalists over to you, the readers of *Game Developer*, via an online survey in November, so that you could have a voice in picking the recipients of the Front Line Awards. After tabulating the survey responses, we are proud to present the winners along with commentary by developers from throughout the game industry. A special thanks goes out to everyone who contributed to this year's Front Line Awards and congratulations to all the finalists and winners.

—Jeffrey Fleming

## HALL OF FAME

# ADOBE FLASH

//// Adobe Systems | www.adobe.com



The first piece of Flash I ever created was a 30 second animated holiday card for clients of the ad agency where I worked. It took me a couple of frustrating days to produce, and if I were even able to locate a copy of it today, I'd probably laugh at myself in embarrassment. Flash's journey from an obscure animation tool to a sophisticated rich media and game development platform could probably not have been predicted by even a single individual—least of all myself from 10+ years ago, slaving away into the night to make a gingerbread man's arms wave. Today, Flash is such an integral, everyday part of my job that I really can't imagine how different my career would have been without it.

Part of Flash's continued strength over the course of its 11 iterations has been to re-invent and re-focus itself to meet the needs of the people who use it. It can be used for so many different tasks and industries that it has almost become a genre of software unto itself. I have used it and seen it used for cartooning and storytelling, post-production animation and effects, illustration, advertising (both TV and online), museum displays, video playback, e-commerce solutions, mobile applications, and game development—even as middleware for commercial AAA games using technology like Scaleform, and as its own platform for rich web entertainment. This last scenario is where I spend most of my time with Flash: creating games on the Internet for people to enjoy.

At Blockdot, we build games and applications for clients' brands, including household names like LEGO, Microsoft, American Airlines, General Motors, and Kraft. The scope of our projects can range from something as simple as a jigsaw puzzle to a full-blown, character-customizable, DIABLO-style action RPG. We've built multiplayer racing games and titles that support driving wheel and custom joystick peripherals. We created the gaming and matchmaking ecosystem for Microsoft's Club Bing community using Flash's companion Media Server as the backbone. We even developed one of the first Flash-authored iPhone games to appear in the iTunes App Store. While there are arguably

other solutions that could have been suitable for each of these needs, none of them allow for delivery on a tight timeline and enable us to leverage a re-usable library of code across all of them. Flash allowed us to do every one of the projects I just mentioned (sometimes with the help of some middleware) and finish them by our clients' deadlines.

One project in particular for Degree V12 Racing required us to create a driving game

Though not geared specifically for game development, the Flash platform's speed, graphical prowess, and easy development language make it a breeze to start building games quickly. Because the Flash Professional IDE is also very extensible, it's entirely possible to write custom panels, controls, and scripts to automate repetitive tasks during development. Combined with Adobe AIR, Flash can access most points



in a full 3D environment ... but inside of Flash's 2D space. It also needed to include a number of FMV cutscenes, interspersed between rounds. While there are 3D libraries such as Papervision and Away3D available for such tasks, the level of detail we needed to achieve just wouldn't have been possible while maintaining performance on lower-end machines. By compositing pre-rendered video loops of the environment, multiple angle clips of the player's car (complete with lighting and shadows), randomly-placed road obstacles and debris, and a special effects layer on top, we were able to create what felt like a true 3D experience that ran on a wide variety of computers. We were even able to apply polish like real-time blurring and distortion effects for when the player picks up power-ups and it had a minimal impact on performance.

of interest on the desktop, such as the file system. We've written "helper" apps for creating puzzle content for word games, parsing and exporting data, and other tasks that might traditionally be written as command line applets in other languages.

With Adobe's recent announcement that a future version of the Flash Player will feature direct-GPU access for implementing real-time 3D graphics and native support for joysticks and other controllers, the future of Flash in the games industry looks bright. I am extremely privileged to be a part of the community surrounding this innovative platform.

*Chris Griffith is the lead game developer at Blockdot and the author of Real-World Flash Game Development (www.flashgamebook.com).*

## ART TOOL

# ADOBE PHOTOSHOP

//// Adobe Systems | [www.adobe.com](http://www.adobe.com)

In reference to Adobe Photoshop CS5, the 2010 Front Line Award winner for art, I'll paraphrase a tag line used by a popular company, "We don't make the art; we make the art better."

Photoshop has been making the art better since PS 1.0 was introduced in 1990. I know this personally. When I broke into video games at Cinematronics in 1983, there was no Photoshop. I developed 2D sprites using a Z-80-based art station cobbled together by hardware wizard Alex McKay. What I wouldn't have given to simply lasso an object, or to pick a color with a color picker.

Today Photoshop CS5 is still the tool of choice for digital artists throughout video game development. It's certainly my tool of choice. In fact you can't swing a digitally rendered dead cat without running into a huge gaggle of professionals using it. And that won't change any time soon.

Photoshop CS5 is simply killin' it with amazing new features. Anyone using Flash or After Effects has probably passed out already over Refine and Content Aware. Manipulating photographic or photo-

real assets? Photoshop CS5 has a new Refine Tool that allows anyone to better refine an object's hazier edges. Use it to grab much more of the edge information around any object so it can be separated from its environment within seconds. And once you've put this object on its own layer, you can fill in the area behind the object with Content Aware information. Formerly, you had very limited options for filling in the area occupied by the object—usually just a solid color. How they did it, I don't know, but the geniuses behind Photoshop CS5 make it possible for the software to fill that area with its best guess on what should be there, be it trees, windows, oil derricks, or whatever. The results may not be perfect, but it saves a massive amount of time. And Content Aware is also present in other features, such as the Healing Brush.

Puppet Warp is another feature I now can't live without. It enables you to move various parts of a photo or a render to get appropriate results without messing with other parts of the render. Imagine that you've rendered out a cover illustration of an alien attacking a space marine, but the art director wants the alien's arm to be cocked back with a more menacing angle. You only have five minutes to send the image and a new render will take at least an hour. In Photoshop CS5, you just click on Puppet Warp, attach control points to the arm joints delineated within the alien figure, and then move them to set the arm at the new angle. Deadline met, and you and the art director are out drinking a pint together instead of trading menacing looks (or worse).

So sad there is limited space here to go on for, on I could. Suffice to say, almost the only way Photoshop CS5 could get better is if you hit a button and it actually did make the art.

—Tom Carroll is an independent game artist.



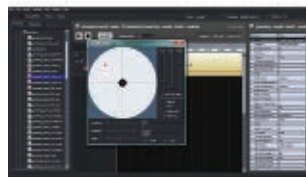
## ART TOOL FINALISTS

- ¶ Autodesk 3ds Max 2011
- ¶ Autodesk Softimage 2011
- ¶ Allegorithmic Substance Designer
- ¶ Pixologic ZBrush 4.0

## AUDIO TOOL

# FMOD DESIGNER

//// Firelight Technologies | [www.fmod.org](http://www.fmod.org)



In the mid-90s Brett Paterson had been writing and releasing MOD files, a file format for music used on titles such as SHADOW OF THE BEAST and UNREAL. Little did his fellow composers

suspect he was also preparing a super-fast code base for what would later become FMOD, one of the most widely-used audio middleware engines. FMOD brought speed and efficiency to the table where other engines were too legacy oriented, too expensive, or too bloated and still getting on their feet. Now, FMOD provides a wide range of features and a huge bang for the buck along with its streamlined performance on just about any platform. FMOD has been used on such recent titles as STARCRAFT II: WINGS OF LIBERTY and LEGO UNIVERSE, and has an integration scheme for most popular middleware game engines such as UDK and CryEngine.

FMOD has a Designer tool as its user interface for the sound designer, voice over integrator, and music integrator. It allows you to set up your audio files and organize them however you wish for loading, streaming and playback, as well as many properties that can also be manipulated at runtime. It also allows you to connect them to in game events. But what makes FMOD unique (apart

from its speed) is its sandbox and its specific interactive music integration pipeline. The sandbox gives a 3D space that developers can use to audition sounds in proper context rather than the age-old method of previewing in a sound bank manager or other auditioning tool.

Bringing audio designers that much closer to going from sound design to integration in the same session is a big step forward. For interactive music, one can create something similar to a flowchart

of a dynamic music flow and preview it in the Designer tool, adding events and layers to create a rich score with easy testing of transitions and crossfades without the hassle of doing so in a standard digital audio workstation environment.

Another useful feature of FMOD is its abstracted Event Layer. The ability to readily link events to the audio engine without time consuming hard coding practices allows for a much more sound designer-controlled connection between game events of any kind and audio objects, whether they be sound, voice, or music.

Finally, one of the best features of FMOD is its wide support of platforms. It supports all the major platforms (PC, Mac, Xbox 360, Wii, PlayStation 3) and several ones you might not expect (iPhone, Linux, Solaris). With these features and near ubiquitous platform and major middleware integration, FMOD deserves a serious look from anyone serious about their game audio.

Alexander Brandon was a founder of the Game Audio Network Guild and is the president of Funky Rustic ([www.funkyrustic.net](http://www.funkyrustic.net)).

## MIDDLEWARE

# HAVOK PHYSICS

//// Havok | [www.havok.com](http://www.havok.com)



When a product wins the *Game Developer* Front Line Award for middleware three times in a row, you know it is doing something really, really well. Havok Physics is once again the winner of the award for middleware and it truly is the premiere solution for bringing high quality physics simulation to your game.

From an engineering point of view, Havok Physics is straightforward and can be integrated into almost any existing engine with minimal changes to the underlying architecture. Once

### MIDDLEWARE FINALISTS

- ❏ Box 2D Box 2D 2.1.0
- ❏ Scaleform GFx 3.2
- ❏ Donya Labs Simplygon 2.9
- ❏ Xaitment XaitControl 3.1

up and running, the included tools, such as the Havok Visual Debugger, allow for real-time information and verification that saved days, and sometimes weeks of time when tracking hard-to-reproduce physical interaction bugs. Havok Physics includes multiple ways to interact with any data allowing for easy use of legacy code and allows the team to dictate how the physics system will be used rather than having the physics system define the game. Havok's powerful and easy to use API and tools help your team stay focused on finding the fun of your game without having to compromise to meet the requirements of the middleware.

Keeping your engineers happy and empowering your team is great, but let's not forget that Havok Physics is good business too. Cross platform development continues to be important for third party publishers and their development partners, and Havok makes that easy. The middleware is near ubiquitous on gaming platforms and the physics engine is highly optimized on all the platforms it runs on. Taking advantage of unique platform features, such as the SPUs on the PS3, allows Havok Physics to perform at a lightning pace. If your game

is targeting multiple platforms, there really is no better choice.

Many of the most well-known and successful games of this generation have been built on Havok Physics technology. Those games would not have been able to deliver as compelling an experience without the benefit of the middleware's extensive



capabilities, ease of integration, and fantastic tool support. If that isn't reason enough to win this year, I don't know what is.

*Mark Coake is a lead systems designer and Joe Allen is a senior software engineer at Nihilistic ([www.nihilistic.com](http://www.nihilistic.com)).*

## ENGINE

# UNREAL ENGINE 3

//// Epic Games | [www.unrealtechnology.com](http://www.unrealtechnology.com)

In 2005, the RED ORCHESTRA mod team won the Make Something Unreal Contest. This helped jump-start Tripwire Interactive and gave us access to the Unreal Engine 3 license as the grand prize.

As a result we are currently using that license to create RED ORCHESTRA 2: HEROES OF STALINGRAD. This has empowered our small team to create something on par with AAA titles and companies with much larger team sizes. UE3 has a great foundation to work from and a smooth pipeline that is facilitated by its toolset.

### GAME ENGINE FINALISTS

- ¶ Crytek CryEngine 3
- ¶ Emergent Game Technologies Gamebryo Lightspeed 3.1.1
- ¶ Unity Technologies Unity 3
- ¶ Trinigy Vision Engine 8

Unreal Engine 3 is an extremely powerful asset for Tripwire. It is easy to work with because of its streamlined tools and features that have been refined over the years and through many AAA titles. We've had a lot of luck hiring people who have experience with the UE3 because it is so widely used (especially with the arrival of the Unreal Developer Kit).

Epic Games supplies hundreds, if not thousands of pages of documentation for various elements of the engine through its Unreal Developer Network (UDN). The documentation and support provided is unprecedented for any software I've seen. Articles are kept up to date and are constantly being translated

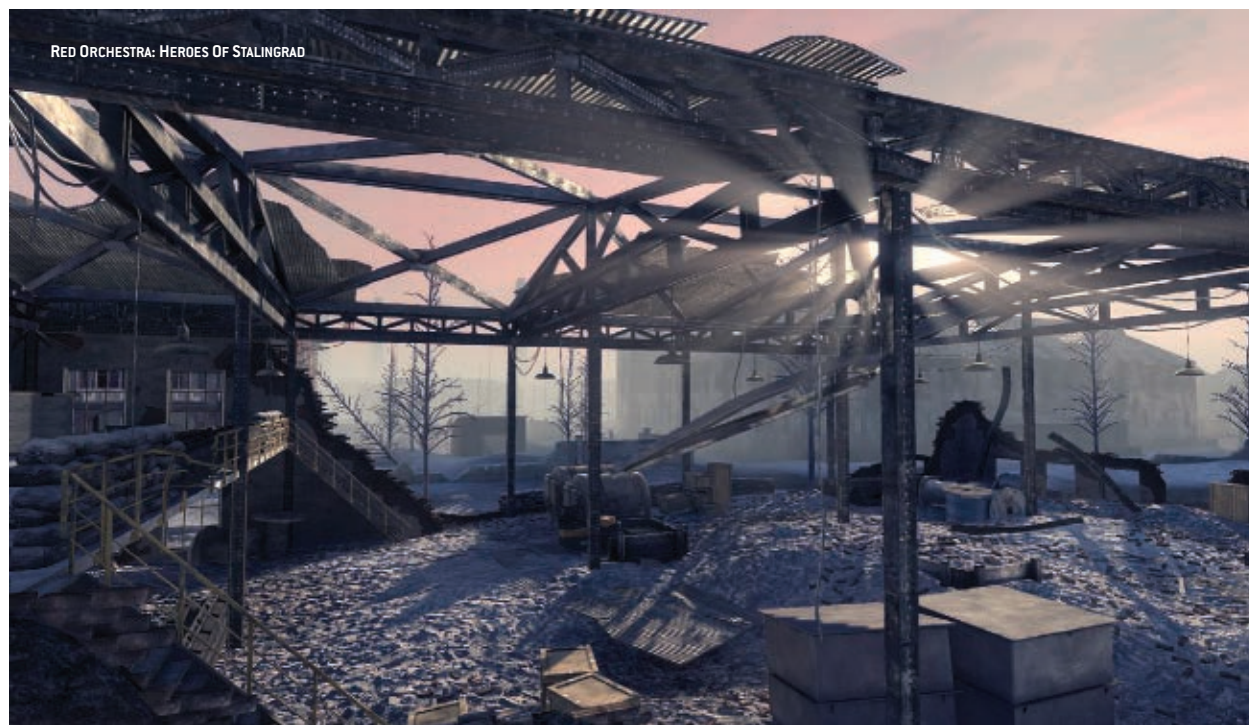
into additional languages such as Japanese and Korean. UDN is an incredible resource for training new people and keeping the team up to date with new features.

We currently take advantage of the Integrated Partners Program (IPP) that Epic and its partners offer. Some of the middleware Tripwire has purchased through this program are Umbra and HumanIK. Middleware provided by these companies add additional features or functionality that Epic does not offer itself, but compliment the engine. The companies that are a part of the IPP provide high-level support the same way Epic does for their licensees.

In addition to 3rd-Party companies adding features to UE3, Epic itself has been continuously adding competitive features and supporting the engine since the early days. Epic provides monthly QA-tested builds with various updates including new rendering features, tool functionality, optimizations, and bug fixes. In addition, full integration of Scaleform (Flash based UI) and Valve's Steamworks have been added to the base engine and are available to all licensees.

The Unreal Engine's extensibility has proved to be incredibly valuable for us, allowing us to create a triple-A title with a solid tool set. It means our small (25-man) team has the ability to compete with big production houses, without the big production costs.

*Adam Hatch is the lead level designer at Tripwire Interactive ([www.tripwireinteractive.com](http://www.tripwireinteractive.com)).*



## PROGRAMMING TOOL

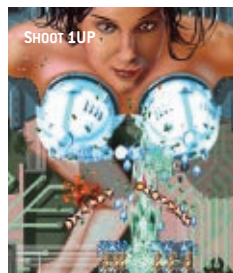
# XNA GAME STUDIO

//// Microsoft | <http://create.msdn.com>

I started out helping develop PC games such as *POSTAL 2*, but my dream had always been to create games for home consoles. A few years later I was able to help develop *RESISTANCE: FALL OF MAN* for the PlayStation 3. I had achieved my goal of console development, but I was still left wanting more. I wanted to put my own crazy game ideas on a home console, but knew the technical and logistical hurdles would be nearly insurmountable.

That's where XNA Game Studio came in. It gives us well-designed ways to render 3D models and 2D sprites, play sounds, use pixel and vertex shaders, poll controllers and the touch screen, save and load to memory storage, build off a solid game class, utilize networking for gameplay, integrate social connections—the list goes on!

All of those intense game engine issues are already taken care of, leaving you to get down and dirty programming your cool game for the Xbox 360 or even Windows Phone 7. For my game *SHOOT 1UP*, I thought it would be interesting if there were a boss enemy who was a floating, robotic, womanly torso with bullet spewing bosoms. She and everything else in *SHOOT 1UP* were perfectly facilitated by XNA Game Studio all the way from my brain to the Xbox 360 marketplace with nary an "evil publisher" restriction or serious technical hurdle.



## PROGRAMMING AND PRODUCTION FINALISTS

- ¶ FlashDevelop 3.2.2 RTM
- ¶ Intel Graphics Performance Analyzers 3.0
- ¶ Hansoft 6.1
- ¶ Perforce 2010.1

sprite starter kit as the stepping-stone to get my core concept for *SHOOT 1UP* together and running as soon as possible.

Once I had a playable demo, the enormous help from XNA forums was equally important as the powerful code libraries. The feedback in the forums from playtesting the early versions of *SHOOT 1UP* was invaluable in giving me a wide spectrum of gamer skill perspectives. Even Microsoft employees sometimes show up to offer help on the more difficult programming issues developers encounter.

If XNA Game Studio merely gave you the ability to sell games on the Xbox 360 and Windows Phone 7, it would be great. But in addition, it's a powerful and robust development suite that allows your team to create ambitious, full games or simply realize new gameplay prototypes in zero time. And with the heart and devotion the XNA team has shown, I can't wait to see how amazing the next version will be. Congratulations XNA Team and all XNA developers for working together to make an awesome package!

*Nathan Fouts is the president of Mommy's Best Games (www.MommysBestGames.com).*

## NETWORKING

# FACEBOOK SDK

//// Facebook | <http://developers.facebook.com>

It's hard to describe the enormous impact that the Facebook platform has had, both on our business and how we socialize with our friends, but I'll attempt to summarize it here. With the release of its platform SDK, Facebook allowed developers to leverage social data in their applications and easily share content with the user's network of friends. The idea was that everything is better with your friends, so why not give folks the ability to add their friends to their own apps?



At Zynga, we believe that games should be played with your friends, so we used the Facebook SDK to create games like *FARMVILLE* and *MAFIA WARS* that were meant to provide players with a fun, social way to connect. We are probably one of the heaviest users of the SDK, utilizing numerous features to make our games more social and engaging for players. We primarily use the Facebook SDK to retrieve information about the player and their friends, and then let the player share game actions or ask their friends to help them out. Both information retrieval and the sharing of content are facilitated easily and quickly using the Facebook SDK.

## NETWORKING FINALISTS

- ¶ GameSpy Technology
- ¶ Aurora Feint Open Feint 2.6
- ¶ ngmoco Plus+
- ¶ RightScale Cloud Management Platform

By leveraging the features provided by the Facebook SDK, we've created new types of gameplay and social interactions between people. For example, there are several accounts of families connecting with each other by playing *FARMVILLE* or friends keeping in touch through a game of *ZYNGA POKER*. By allowing us to make games truly social and enabling play with your real life friends, the Facebook platform has been incredibly powerful for us.

It bears mentioning that this wouldn't have been possible without the commitment Facebook has made to improving their product. The Facebook SDK, unlike traditional tools used by game developers, is a living and constantly evolving service. Facebook realized this early on and has strived to provide updated documentation and tight communication channels with developers. Throughout our relationship with the company, Facebook has constantly improved the quality of its platform for both the users of the site, and also the developers. This dedication to quality and to the developer experience has allowed us to move quickly, adapt to the ever-changing application ecosystem, and avoid service downtime.

Without a doubt, we are only scratching the surface of how the Facebook platform will change the Internet. With tighter integration into products, more opportunities and ways to connect with friends, and an ever growing set of integrated websites and users, the Facebook platform is definitely a one-of-a-kind product that is shaping the future of the web and how we connect with one another.

*Amitt Mahajan is the director of engineering at Zynga (www.zynga.com).*

# GIVE SPREADSHEETS THE BOOT!



## DevTest Studio

The game industry's #1 choice for test management and defect tracking.

### DevTrack

Use DevTrack to track defects/issues

- Track each issue through a definable workflow
- SCM integration-track fixes against their source code deliverables
- Deploy a resolution across multiple releases, versions and products
- Reporting and metrics to illustrate the entire defect lifecycle

### DevTest

Use DevTest to manage your testing

- Create a central repository for your test cases, Knowledge items and automation scripts
- Schedule releases and test cycles using a wizard-driven interface
- Execute test assignments and submit defects from the same interface
- Track results with real-time dashboards and reports

### TestLink

Use TestLink to automate your testing

- Add automated tests to the DevTest test library
- Schedule automated tests along with manual tests
- Launch automated tests from the DevTest interface
- Track automation results with real-time dashboards and reports

Try DevTrack and DevTest live. Download free evaluation software. Watch a recorded overview demo.

[www.techexcel.com](http://www.techexcel.com)



# dance





# post mortem

# central

////////// Over 5 years ago, Harmonix developers were brainstorming about all the great ways that people interact with music and wondering how we might transform those interactions into authentic gameplay experiences. Dancing, with its visceral connection to rhythm, was the most potent and promising new idea. We knew that some day the time and tech would be right for us to make a fully immersive, authentic dance game. The opportunity to offer non-dancers the same type of approachable experience that exposed millions of non-musicians to the joy of rock music was a huge incentive to make this dream a reality. Serendipity struck when Microsoft showed Kinect to Harmonix, providing us with the perfect opportunity to develop an authentic dance game for a population fascinated with dancing shows and itching to get off the couch and join in. >>>

# dancecentral

For the dev team, this was the perfect opportunity to break from the ROCK BAND paradigm and craft a new universe of characters and venues. We were confident that the controller-free, body tracking, and fully immersive capabilities of Kinect made it the right technology for our game. Further, the opportunity to work with choreographers and dancers on a daily basis injected a new energy into an experienced and seasoned team. However, like any team working on a new IP, we made mistakes along the way, many of which were important learning experiences for us. The following is a selection of our most notable successes and missteps.

## WHAT WENT RIGHT

**1) KEEPING IT REAL.** DANCE CENTRAL's design started with a lofty aim - to create a game that will teach players real dance moves. This goal served as a compass during early prototyping, leading the team toward designs that gave prominence to dancing above all else. We utilized the instantly recognizable choreography from Soulja Boy's "Crank Dat" as a litmus test for potential mechanics, throwing out a number of pose matching and gem hitting prototypes when they didn't stand up to the challenge of communicating moves like the "Lean n' Rock" or the "Supaman." After a few months and several iterations, we settled on our move names and flashcards approach, which handily communicated the entirety of "Crank Dat."

Although we had identified core gameplay mechanics that would teach real dance, we were still unsure of the makeup of our development team. In working on the ROCK BAND series, we've always considered our staff's intimate knowledge of rock music a vital element of the franchise. Our developers' experience touring, playing, and writing for their bands imbues Rock Band with an authenticity that we believe sets it apart from other music games. In order to inject the same type of authenticity into a dance game, we felt we had to assemble a team with a love for hip-hop, pop, and, most of all, dance.

At the dawn of the development of DANCE CENTRAL, we held company-wide tryouts to determine who would mocop the first prototype routine. We asked each prospective choreographer to both dance the routine to "Crank Dat" and develop a routine of his or her own design. While most Harmonix developers ran away screaming, a handful of Harmonix artists proved to be outstanding dancers and were promptly sent to mocop the first routines. This choreography proved essential during prototyping, but we quickly realized it would be critical to add professional dancers to the staff, both to create original professional-level choreography for the game and to teach the rest of the team how to dance.

We recruited local choreographers, eventually hiring Marcos Aguirre and Francisca "Frenchy" Hernandez as our internal choreography team. Marcos and Frenchy occupied various roles during development: creating choreography for game levels, holding dance classes for the team, and consulting on everything from voice over to song list. Their knowledge and excitement were a huge boost for the team. Through the team-wide dance classes, our choreographers imparted vital lessons about dance instruction, while exposing the team to the music, language, and culture of hip-hop dance.

As we tasked our choreographers with developing the first routines for the game, we refrained from imposing gameplay-driven choreographic restrictions so we could better understand what made up an authentic routine. This somewhat open approach was risky, but led to a refined focus on the game's core mission, as we shaped our mechanics around actual dances rather than limiting or shoehorning routines into a rigid structure.

**2) BREAK IT DOWN!** Given the complexity of the choreographed routines, it became clear that DANCE CENTRAL would require an in-depth learning mode where players could spend hours learning more complicated combinations. We set out to create an experience that was not only effective at delivering instruction, but also fun to play. We knew that if it weren't fun, no one would play it, and therefore no one would learn.



<b>DEVELOPER</b>	Harmonix Music Systems
<b>PUBLISHER</b>	MTV Games/Microsoft
<b>RELEASE DATE</b>	11/4/10
<b>PLATFORM</b>	Kinect for Xbox 360
<b>NUMBER OF DEVELOPERS</b>	60 Devs
<b>LENGTH OF DEVELOPMENT</b>	15 Months
<b>LINES OF CODE</b>	63,000 lines of script in 310 files, 515,000 lines of code in 1860 files
<b>DEVELOPMENT TOOLS</b>	Typical development machine: dual core Intel Xeon 3GHz processor 4GB of RAM nVidia Quadro FX3700 512MB of RAM 320GB hard drive
<b>SOFTWARE</b>	Visual Studio 2008 Motion Builder 3D Studio Max Cubase Bink
<b>TECHNOLOGIES</b>	Motion capture Harmonix custom engine & tools Kinect full-body tracking

# hiring

"It's all work, work, work, work, work..."

- **Work** with the best game teams in the world
- **Work** in a stimulating challenging team environment
- **Work** on all of the very latest hardware platforms
- **Work** on the cutting edge of games technology
- **Work** in a range of locations around the world

Havok has vacancies in

**Dublin**   **Munich**   **San Francisco**   **Tokyo**

See [www.havok.com/careers](http://www.havok.com/careers) for more details



**havok**  
[www.havok.com](http://www.havok.com)



Our first step in designing what is now known as “Break it Down” was to observe Marcos and Frenchy. Their weekly dance lessons provided insight into how people master dance moves and routines. By observing their natural method of instruction, we narrowed the possibility space for BiD’s design, quickly discarding less fruitful ideas. We recognized that our play structure should emulate their teaching style, building on established elements such as the use of “recaps,” “slow-downs” (allowing the player to choose to slow down the action to give them a better look at the current move), and “verb barks” (shorts bursts of VO that step players through new moves, such as “Cross! Back! Step! Together!”). These elements formed the foundation of the shipping version of BiD and were critical to its success.

The same cross-discipline team responsible for developing BiD was also responsible for developing all other key gameplay features. This could have been an unmanageable amount of

work, but we constructed a development process to mitigate the impact of the numerous iterations BiD required. Instead of continuous iteration, the development team worked with our internal playtesting department in a leapfrog fashion. A few key BiD features would be implemented, and while awaiting playtest feedback, the team would switch to development on something unrelated to BiD. Once playtest results came back, the team would refocus on BiD, with iteration driven by the playtest results. Other new non-BiD features, implemented in the meantime, were then sent to playtest. This alternating cycle continued throughout development.

We’re very satisfied with the final design of Break it Down. Its dynamic and detailed instruction methodology is unique to DANCE CENTRAL, and is a critical component of the overall experience we wanted to construct. While there is plenty of room for improvement, the mode’s variable pacing successfully pairs game-like motivation with

effective teaching methods. This would not have been possible without the inspiration provided by the choreographers paired with an effective process of playtest-driven iteration.

**3) ADAPTING WORKSPACES AND TOOLS FOR KINECT.** Kinect doesn’t just enable new ways of playing games, it also demands new approaches to making games. At the onset of development, it was immediately clear that our existing workspaces lacked the physical space for developers to easily play and test the game. After reorganizing our office layout, we were able to give everyone on the team the room they needed to stand up, get down, and move around.

Next, we realized that it was difficult for coders, designers, and artists to maintain an efficient workflow if they frequently needed to stand up, wait for their skeleton to be tracked, and then sit back down at their workstation. Our first solution was decidedly low-tech: Because Kinect is also capable of tracking skeletons for non-human dummies, we constructed makeshift mannequins to stand in for real human developers. Our UI coder even went as far as building a functional mannequin with a movable right arm for testing our “swipe” shell gesture (it became affectionately known as “Swipey”).

Of course, a mannequin can’t perform a dance routine, so testing efficiency remained an issue for many on the team. To remedy this, we made an important investment in a recording and playback system for Kinect’s skeletal data. We developed a custom solution within our internal engine, making it easy for anyone on the team to record, play, loop, and analyze clips of skeletal data. We used our graphics engine to create debug visualizations, using geometric shapes and color. Because each recording was stored within our standard object files, it was straightforward to organize the hundreds of recordings into a reasonable database-like structure by annotating each clip with metadata.

Our recording tools proved invaluable for coders, artists, and designers working on any system reliant on skeletal data, including our dance detection, shell navigation, and freestyle “visualizer.”

**4) HITTING A MOVING SKELETON.** As a Kinect launch title, we were faced with the challenge of building a game for a completely new input device while that device itself was still in development. The challenges were similar to developing for a new console, but in some ways the process was even more fluid as the Kinect software platform evolved alongside the hardware.

Given DANCE CENTRAL’s reliance on dance detection, Kinect’s skeleton tracking was the key technology we needed to harness. Preliminary versions of the tracking pipeline were less precise and reliable than the final shipping system. As we

designed our user interface and dance choreography, we uncovered gestures and poses that were difficult for the early tracking to reliably detect. It was impossible for us to predict exactly how robust tracking would be by launch, and there was some anxiety as to whether we would need to simplify or remove moves and gestures that were undetectable.

Rather than putting development off while the tracking improved, we accepted the situation and tried, whenever possible, to handle noisy skeletal data and cases of skeleton “crumpling” gracefully. We also began investing in sophisticated in-game detection technology that could better handle the possibility of imperfect tracking. The payoff was a more consistently playable and fun game throughout development. Even as the Kinect skeletal tracking was refined to its more robust shipping state, we realized that this wasn’t wasted work. Every Kinect game needs to handle cases where skeletal data isn’t reliable (e.g. if the player drifts out of the sensor’s field of view). Dealing with the design ramifications of noisy and unreliable data early on helped us better understand the capabilities of the technology and mitigate risk.

Early on, Microsoft provided us with detailed resource costs for using Kinect, including the CPU, GPU, and memory costs for skeletal tracking and other features of the platform. These costs are non-trivial, but because we had plenty of advance warning, we were able to plan and budget accordingly. The Kinect platform team also provided great resources for tackling some of the technical challenges unique to Kinect, such as smoothing algorithms, image processing, and latency reduction.

**5) THE RIGHT SCOPE, THE RIGHT MOVES.** Creating new IP isn’t easy, especially when your studio is well known for one type of genre-defining game. We knew that the DANCE CENTRAL team would need the drive to do something completely new while also quickly reaching consensus on feature scope. We looked back to the work our studio had done developing the original GUITAR HERO as a reference point. At that time, Harmonix was a much smaller developer. We had to be prudent about our ambitions while remaining flexible and innovative. Our imperative for GH was first and foremost to completely nail the core fun experience. Naturally, it felt like DANCE CENTRAL should share that approach.

We made the decision to keep the team small, composed of key members who had proven themselves on other Harmonix titles over the years. This was no easy task, given that we were simultaneously developing ROCK BAND 3! This core DANCE CENTRAL group was able to break into agile sub-teams that rapidly iterated on gameplay prototypes. There was a conscious effort to focus the design on strengthening the core dance experience rather than adding breadth and complexity. Adding a character creator, in-depth single player campaign, or other ancillary feature would have detracted from achieving our core goals. Instead, we spent most of our time perfecting the dance gameplay – learning how to best handle difficulty, building a teaching mode, and making sure our flashcard and spotlight HUD were conveying the right information at the right time, all in the service of keeping the dancing as fun and entertaining as possible. Our strike teams consisted of a designer, coder, artists, sound designer, QA tester, and a producer, each empowered to scope, design, and prototype the main gameplay modes.

As noted earlier, we also made sure the team had a genuine interest in dancing and dance culture. That dedication shows in the final product. Having a common background, knowledge of club music, and passion for dancing meant we were able to approach all design choices knowing that they were grounded in the world of dance. By “speaking the same language,” we moved rapidly through iteration, since we didn’t suffer from off-the-mark decisions. We felt validated in our approach when we showed the game to professional dancers who were impressed with the authenticity of both the dancing and approach to teaching.

The combination of a small, powerful team, tasked with the goal of implementing a core feature set unhindered by feature bloat enabled us to execute a sophisticated game based on Microsoft’s brand new motion tracking tech in just 12 months.



## Ryan Challinor on UI/Nav

/// When developing a Kinect title, you also need to create a way to navigate menus using gestural input. Microsoft hadn’t yet released any UI libraries or guidelines, so despite the fact that none of our team members had any expertise in this field, we set out to design a solution from scratch.

Our approach to working in this unfamiliar and evolving field was to iterate quickly through rapid prototyping. This allowed us to fail quickly, identify what didn’t work, and eventually whittle down to what did. We formed a team to focus on the UI that met twice a week. The principal workers in this team were a programmer and a UI artist; the remainder of this team included the project director, producer, lead programmer, lead designer, and other senior members of the project. Generally these meetings had about 8 attendees total.

Here’s how we structured the team meetings: We’d begin by taking a look at the latest prototype, noting its deficiencies and limitations, and brainstorming ways to improve it. If someone had an idea, they would have to sell it to the rest of the team. Eventually, we’d reach a consensus and leave with a list of action items. Those action items alone determined the subsequent work for the programmer and UI artist. With this method we were able to allow a large group of individuals to have creative input while removing dependencies or changes in direction for the principal workers in between meetings.

As we ran through many prototypes, we learned some important lessons about full-

body gestural input. One major hurdle that separated this type of input from, say, a touchscreen, was the lack of an obvious way to signal engagement/disengagement. With an iPhone, if you want to press a button, flick between pages, or scroll a list, you can just touch your finger to the screen. When navigating a UI with Kinect, it’s essentially like your finger is always on the screen. Kinect doesn’t detect whether your hands are open or closed, so we couldn’t determine engagement that way. We tried using the player’s hand position in the depth axis and arm extension to determine engagement, and while it seemed like it would be an intuitive way to interact with this sort of UI, in practice it was unsatisfying and strange.

We decided to ditch using the depth axis, electing to use only the hand’s horizontal and vertical position to navigate. We prototyped a UI using this concept: a vertical list of buttons where a specific button was highlighted based on your hand height, and that highlighted button could be slid across the screen in order to select it. After the prototype concept was initially dismissed because it didn’t feel right, our UI artist had an idea for a treatment and created a video in Maya to mock up how he envisioned us using this navigation paradigm. He altered the look and animation of these buttons, while still retaining the same underlying user interaction. This added aesthetic polish elucidated the concept to the team, and the shipping version of our UI matches that mockup video very closely.

## WHAT WENT WRONG

**1) MOVING FROM "GOOD" TO "FLAWLESS" DETECTION.** Our move detection system relies on Kinect's skeletal tracking to score hundreds of unique dance moves across our 32 songs. Delivering immediate and accurate feedback to the player about how well they are dancing is at the heart of our game. As a result, we knew that the quality of our detection would make or break the experience. We also knew it would be one of our biggest technical challenges.

Eager to start, we began prototyping our move detection with modest expectations in terms of how well it would work. It didn't take us long to develop reasonably good detection. While our prototype wasn't at shipping-quality, it worked well enough to facilitate gameplay and choreography testing while our engineering team continued to research better methods.

Unfortunately, getting from reasonably good to shipping-quality move detection proved much harder than our initial efforts led us to believe. Our two subsequent approaches demonstrated incremental improvements over our early prototype, but were ultimately deemed unacceptable and scrapped. We had learned a lot from our efforts, but we were running out of time. With only a few months left, we settled on what we believed was a solid technical solution.

To prove that this system was capable, we focused on authoring and tuning detection for one example song with the goal of getting it all the way to shipping-quality. The positive outcome was that the detection worked well and we finally had a system we were confident would do our dance moves justice. That said, we had an experience that remained unpolished until the final throes of development. This made it more challenging to evaluate our overall progress and led to other production woes as we closed in on GM.

**2) BEHIND THE BEAT.** After we got detection we were happy with for one of our songs, we began the process of applying that method to the entire title. It was immediately clear that we had greatly underestimated how much time it would take to author, tune, and validate this new detection for each song. With 31 more songs to go and rapidly approaching deadlines, this miscalculation created an arduous three-week "detection crunch" at the end of the project.

This "detection crunch" bled over into a time when we were expecting to be focused on bug fixing. It was an immense challenge for the design team to revisit all the material in the game and to painstakingly prepare every routine for review and testing while we integrated other last minute additions like voice-over.

Furthermore, as each song was tuned, our QA team had to learn, master, and continually repeat demanding expert-level choreography so they could confirm the detection was working as intended. Dancing for hours a day for several weeks, our testers became physically and mentally exhausted. In order to acquire all the necessary test data, we enlisted dozens of additional staff from Harmonix to learn and perform routines. As our deadline loomed, it took a Herculean effort from our design, QA, and playtesting teams, with an assist from other studio volunteers over the final stretch, to make it to shipping quality detection for all 32 of our routines.

**3) GETTING LOW.** DANCE CENTRAL's choreography covers a substantial range of difficulty, from simple two-steps to challenging toprock moves. We knew that having a broad range of choreography was necessary for the game to appeal to both novices and experts, but as the project unfolded we were unsure about the qualities that define the difficulty of a given move. We knew that delivering an accessible experience on easy would be vital to the title's broad appeal, but throughout development, we tried and failed to establish an appropriate "low bar" numerous times.

Our first attempt began with our choreographers developing a few complex routines and presenting them to the design team. The design team, a group with a range of dance skills, tried out each move and discussed which were easy, medium, and hard. Using those ratings, we derived easy and medium combinations and videotaped the choreographers performing them. We presented these videos to various playtesters and had each try to dance



along, rating the difficulty of the moves and the routines. Unfortunately, our playtesters weren't good judges of their own skill level or performance. Once the songs were integrated into the game and players were scored, we found playtesters struggling with moves they had previously rated as easy. This problem was compounded by the fact that we had already motion captured the routines and couldn't reshoot, given the release schedule. We were stuck with some very challenging hard routines.

We tried again, this time encouraging our choreographers to come up with a few very simple routines. This time, our easy and medium levels turned out much easier. Some of our more talented playtesters were able to pick up hard levels without much effort. We thought we had reached an acceptable easy, but then tried presenting these levels to some key high-level staff who struggled, unable to perform the majority of the moves. With important members of the Harmonix brain trust unable to comment on the mechanics of the game, we knew we had yet to find universally accessible choreography.

With a few months to go, we finally figured out how to use staff members with minimal dance skills to our advantage. We asked our choreographers to generate a number of very easy moves and set up a dance class to teach



these moves to the self-described “bad dancers” at Harmonix. The choreographers went through each move, asking the novices to follow along as the designers watched and noted which moves they picked up quickly. Using this information, we crafted four new easy songs, which make up the first tier of DANCE CENTRAL. Although we succeeded in making these first songs very approachable, the difficulty ramp across all songs is not as smooth as we would have liked.

#### 4) BRAND NEW ROUTINE – UNDERESTIMATING THE NEW PROCESS.

For many years, the process of licensing and integrating songs into our music games has been a well-oiled machine. The turnaround time from the moment a song is licensed to when it’s playable and bug-free in ROCK BAND is relatively quick and well-understood internally. For DANCE CENTRAL, we initially thought we knew which production practices would work and which new processes would be needed. We quickly found out how easily it was to derail an unproven process and how complicated and dependent all the steps of bringing a song to completion were.

Unlike ROCK BAND’s process of licensing, stem prep, authoring, and testing, DANCE CENTRAL’s process became licensing, choreographing and vetting, song editing, difficulty creation, mocap shooting and cleanup, animation integration, clip authoring, filter tuning, testing, and a handful of other steps along the way. Our two-to-three week-long processes became a two-to-three month-long process. This new process was much more fragile than we initially thought. Early missteps, however small, would ripple through the months-long pipeline. A single delay in song licensing or a mocap dancer being out sick for a day could end up putting an on-disc song at risk of being cut. With this high wire act performed by a small team with finite deadlines, we had an extreme production balancing act on our hands. Our producers became very adept at shifting schedules to keep things on track.

It was also easy to underestimate the physical demands of making a dance game. Both our choreographers and our QA testers were constantly being pushed to their physical limits in order to get their jobs done and stay on schedule. This physical burden on the testing and development teams was something we had never considered when developing previous games.

#### 5) IT’S GETTING LATE; LET’S ADD VO AND NARRATIVE!

Nearly all of Harmonix’s games over the past decade have placed a heavy focus on creating a deep connection between the player and the music. Elements that distract from this have always been given less attention; we’ve devoted little focus to developing complex narratives or fleshing out character backgrounds. Only one of our games, ANTIGRAV, featured characters that spoke, and while GUITAR HERO

1 and 2 had selectable prefab characters, they never spoke, allowing the game to hint at a narrative context rather than inhabit a developed story. The ROCK BAND series has a very loose “rise to fame” narrative backbone focused on player-created band members, who start as unknowns and finish as superstars. However, all of this is presented in a lightweight fashion: characters don’t speak and are essentially blank slates onto which players can project their own personality.

For DANCE CENTRAL, the decision was made early on to create a world inhabited by an ensemble cast of unique characters. As with our previous rock-focused games, we would keep things simple; any narrative would be loosely implied and supported by a lightweight text based player ranking system. Our characters would have unique personalities, but these would be communicated through their dance styles, body language, and fashion sense. They would not speak.

Quite late in development it became clear that the decision to mute our characters had been the wrong one. We reached a point where the animated mocap sequences that bookend each song were fleshed out, and featured our characters doing crazy dance moves and strutting their stuff. The moves looked killer! However, with no voice, our characters were coming across as lifeless puppets. So we made the decision to add character VO to the game. We felt that this would bring our characters to life in ways that the visual cues and animation had not. In doing this, we underestimated both the risks and amount of work involved, in large part due to our inexperience in this area of development.


We kickstarted the process of giving each of our characters a voice and immediately encountered a slew of execution hindrances further compounded by our tight deadlines. Our staff writer was able to flesh out characters and turn around quality dialog quickly, but was not afforded the luxury of time to iterate. We located a company to secure and record VO talent, but due to time constraints our choice of available actors was limited and our opportunity

for pickup recordings and line tweaks was almost non-existent. We had to roll new code in to support lip synch animation, and as a result our already strapped animation department now had more work thrown onto the pile. Then there was localization to deal with ... the list of complications went on and on. The mad scramble that ensued to resolve these issues and meet our already aggressive deadline added stress and distractions to an already taxed team.

In the end, some of the characters came out fun and relatable and others missed the mark completely. We’ve learned that creating fully-fledged characters is not an undertaking that should be taken lightly or added late to the process; the risks in doing so cannot be underestimated.

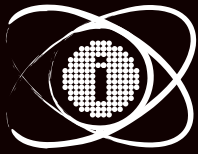
#### GET READY TO GET DOWN

It wasn’t until Dance Central’s reveal at E3 that we truly understood how our decisions would pay off. It was an amazing moment to see people engage with our game for the first time, cast off their inhibitions and realize that anyone could step up and shake it. We always knew that this title would push the Kinect technology to its limits and we’re incredibly proud to have made a game that is getting people up off their couches, moving around and feeling good. In true Harmonix style, we took on risky work in an unfamiliar space with a short amount of time, and delivered a game that fits perfectly within the Harmonix universe.

Lastly, a sincere and heartfelt thank you to Microsoft and the teams there that worked closely with us, all of who stepped up and delivered resources and advice throughout development that kept us on track and on schedule. Beyond that, we thank them for creating Kinect itself, through which we’re bringing new and authentic experiences to gamers and non-gamers, dancers and non-dancers! 

*Authors include project lead Kasson Crooker, lead programmer Marc Flury, designer Matt Boch, lead designer Dean Tate, and programmer Ryan Challinor.*





THE 13TH ANNUAL  
INDEPENDENT  
GAMES FESTIVAL

# AWARDS

WEDNESDAY, MARCH 2, 2011 • 6:30PM – 8:30PM

SAN FRANCISCO MOSCONE CONVENTION CENTER, HALL D



## IGF AWARD CATEGORIES

### IGF MAIN COMPETITION

- » Seamus McNally Grand Prize
- » Excellence In Design
- » Excellence In Visual Art
- » Excellence In Audio
- » Technical Excellence
- » Audience Award
- » Best Mobile Game

### IGF STUDENT SHOWCASE

- » Student Showcase Finalist
- » Best Student Game

### NUOVO AWARD



CELEBRATING OVER 600 INNOVATIVE GAMES ACROSS THIS  
YEAR'S MAIN, STUDENT, AND NUOVO AWARD COMPETITIONS

VIEW THIS YEAR'S SUBMISSIONS AT [WWW.IGF.COM](http://WWW.IGF.COM)

PLAY THE FINALISTS AT THE IGF PAVILION ON THE GDC 2011 EXPO FLOOR, MARCH 2-4, 2011

PLATINUM SPONSOR



GOLD SPONSOR



PLATFORM SPONSOR





# strategy games

the next step for

R A D E S T O J S A V L J E V I C

I've been a fan of strategy games since I began playing them as a kid. It is incredibly fun to analyze your enemy, plan your strategy, manage resources, plot your moves, and then see all your hard work pay off when you successfully attack your opponent. Over the course of my game development career I've been fortunate enough to work on some of the biggest strategy franchises in gaming, which is why it was really painful to watch the genre fall, going from being one of the biggest in gaming to almost a footnote.

There have been many attempts to revive the genre by bringing strategy games onto different formats such as consoles, mobile devices, and the massively multiplayer online space, but so far none of these attempts have produced a breakout hit. In this article, I'll examine the roots of the most successful strategy sub-genre, real-time strategy, to find out why such games are so fun to play, and why they can work well on mobile platforms.

## THE GENRE

» Let's start out by discussing the three core elements that make a strategy game work. Resource management (money or raw materials for production), production (base building, unit creation, technology upgrades), and combat are the fundamental components of most strategy games. Modern games also require a multiplayer mode because no matter how much time you spend building an AI system, it will always be more challenging and fun to play against a real player. It's also critical to offer the player multiple ways to accomplish a particular goal since, by definition, there are no strategic choices to be made if there is only one viable option. Finally, strategy games should prioritize and reward long-term strategic thinking over immediate actions or tactical gameplay.

Turn-based strategy games have their roots in board game design. They made the leap onto PCs in the early 1980s, and over time, there have been several innovations in the strategy genre that have spawned new types of gameplay and, in some cases, birthed new sub-genres.

The first big evolutionary step was the implementation of real-time play in Westwood Studios' 1992 title, *DUNE II*. My business partner at Jet Set Games, Brett Sperry (who was also Westwood Studios' co-founder), was an avid player of turn-based strategy games and postulated that you could reach a much broader audience of gamers if you streamlined gameplay to the point where it would be possible to play in real time using a mouse and keyboard interface. The success of *DUNE II* ushered in a brand new genre and its follow-up, *COMMAND & CONQUER*, became one of the best selling franchises in video game history.

Another evolution took place when Nintendo released the hit title *ADVANCE WARS* on the Game Boy Advance in 2001. Although not the first turn-based strategy game on a handheld, it was one of the best selling, and numerous strategy games for mobile devices followed in its wake. *ADVANCE WARS'* innovation was to simplify player interactions with the user interface in order to make it more playable

# it's your move

# it's your move

the next step for strategy games

on handheld devices. ADVANCE WARS' speedy pace also emphasized combat over resource gathering and unit production although these elements were still present in streamlined form. Critically, ADVANCE WARS went back to the turn-based model, which broke the game into smaller chunks that were ideal for mobile gaming and also had the benefit of making the game easier to play on a small screen. It's mobility increased its accessibility dramatically.

The most recent evolution has been the popularization of web browser-based tower defense games, which are now finding a new home on mobile platforms like the iPhone. Tower defense games have simplified play by removing direct user control of combat but have retained the immediacy of real-time gameplay. This makes them perfect for shorter gaming sessions, and has the benefit of making the games more accessible to a new group of casual gamers who otherwise wouldn't touch a strategy game.

Born out of the mod community, DEFENSE OF THE ANCIENTS is an offshoot of the hero-based gameplay in WARCRAFT 3, and has been quite successful on the PC. This style of game does away with base management and minimizes resource gathering, allowing for an even faster-paced game which is well suited to real-time multiplayer combat. DOTA games have become a hit in competitive multiplayer circles and are evolving at a rapid rate, gaining the support of publishers such as Riot and Valve.

## PLATFORM CHOICE AND DESIGN

» Whenever I begin a new project, one of the first things I think about is the platform. This is especially critical for strategy games. Certain platforms lend themselves to certain genres, and you can get in a lot of trouble when you try to force a game mechanic onto a platform when it isn't an ideal fit. Before we look into strategy games on multiple platforms, let's spend a little time on what differentiates platforms.

**Consoles:** Action focused, multiple players looking at the same screen, chat via voice, player controls something directly, sometimes online. Well suited for games like first-person shooters, role-playing games, action games, driving games, and sports games.

**PCs:** Strategy focused, one player per screen, chat via text or voice, player controls something directly or it can be abstracted, always online. Ideal for massively multiplayer role-playing games, strategy games, RPGs, web games, card games, and Facebook games.

**Mobile:** Casual focused with short play sessions, one player per screen, chat via text, direct or abstract control, display screen is the controller, and unreliable online connection. A good fit for puzzle, word/trivia, arcade action, and strategy games.

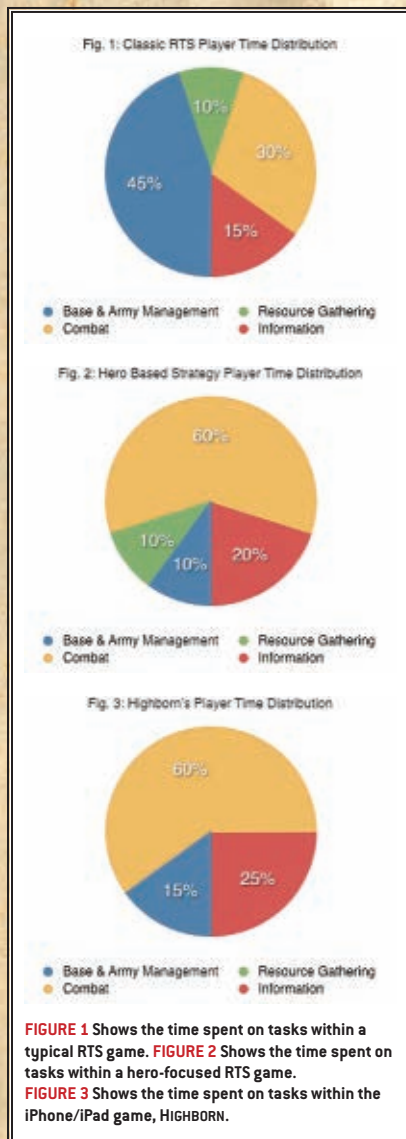


FIGURE 1 Shows the time spent on tasks within a typical RTS game. FIGURE 2 Shows the time spent on tasks within a hero-focused RTS game. FIGURE 3 Shows the time spent on tasks within the iPhone/iPad game, HIGBORN.

As these platforms have evolved, some genres have successfully transitioned to newer platforms and have flourished. Others came into existence because of technical innovations or improvements. Driving games, for example, were once best experienced on a PC due to higher graphical fidelity and frame rate. However, larger screens, more powerful consoles, and analog controls made the experience so much better on a console that developers have stopped making PC driving simulators. As another example, there are fundamental considerations such as size and play environment that preclude a WORLD OF WARCRAFT-style MMO from ever working right on an iPhone.

## STRATEGY GAMES ON CONSOLES

» Let's consider the example of real-time strategy games and designing one for consoles. The entire core of these games is based on

micromanagement, and they are designed around the mouse-keyboard input combination. Players control multiple units, scroll around the map, select units, and use the other mouse button to perform context-specific actions. There have been many attempts to bring RTS games onto consoles where the design focus has been to remap the core functions to the console controller. Unfortunately, experience has shown that these kinds of interface fixes just don't work. There are a few fundamental problems that you can't design your way around when working on a console RTS.

First, as mentioned micromanagement is at the core of RTS gameplay. Spend some time around hardcore multiplayer STARCRAFT users and you'll hear comparisons about APMs, or actions per minute, with the idea being that all other things being equal, the player with a greater APM number will win a battle. The mouse allows players to rapidly select units, issue orders, and move on to the next unit. Console controllers don't make this easy as they are designed for the exact opposite: direct unit control, such as driving a tank around and shooting at targets.

Take a look at Figure 1; it illustrates approximately how a user spends his time in a classic RTS game. Combat and resource gathering are self-explanatory. Base and army management includes things such as building production and layout, unit production, technology research or upgrades, diplomacy, and trade. Information is scouting, sending out spies, or anything else in the game that gives players an idea of what their enemy is doing.

There have been some notable attempts to circumvent the micromanagement problem on consoles, most notably by giving the player hero-type units that are under direct user control with an array of other units that follow the hero. Nintendo's PIKMIN is a great example of this. This approach puts the game's focus squarely on combat, as illustrated in Figure 2. While this isn't necessarily a bad thing, games of this type generally involve far less strategic thinking and instead rely on tactics in combat for fun.

The second problem is figuring out how the player will navigate and interact with a complex UI using a game controller. How do you deal with less exciting tasks such as unit production, upgrades, and building placement on a platform where interacting with UI in general can be tedious? There have been several clever attempts to solve this problem, but they've often been either too simplistic, which limits gameplay, or require users to learn convoluted new control schemes. One of the more successful solutions to this problem was the voice command system used in Ubisoft's ENDWAR. The voice command approach worked quite well with a few exceptions: you lost the ability for social chat in multiplayer games, and generally had to make sure you were alone with the console unless you wanted to look really silly barking out orders in front of your housemates.

TOM CLANCY'S ENDWAR.



## HIGHBORN - STRATEGY GAME ON IPHONE

» When Jet Set Games set out to build Highborn for the iPhone and iPad, we looked at other strategy games on the App Store, discussed what we'd like to play as gamers, and spent a lot of time thinking about how players use the device. We came away with some insight and worked to design around the device's shortfalls and play up to its strengths.

The first issue that drove us crazy in numerous games was what we called the "fat fingers" problem on a touch screen interface. When a device's display is also its primary interface input, you are immediately forced to simplify the UI so the display is obscured as little as possible. Tilt and voice controls were ruled out immediately, and the only viable option left for us was to separate the user's input (fingers on the screen) from a resulting action. We did this by making the primary gameplay in Highborn turn based, which conveniently solved several other issues as well.

Second, we realized that users were likely to get interrupted frequently while playing a game, either by a phone call, text message, or something in their environment. Gameplay needed to handle pauses in such a way that the action was easy to get back into without the player being confused because they stopped in the middle of an attack and couldn't recover their battlefield memory to figure out what was going on and what to do next. Again, the turn-based design solved this problem by breaking the game into numerous smaller bits that waited for user input before continuing.

We felt the game needed a multiplayer component, but we didn't want to build a real-time system since, as any iPhone user knows, the network connection isn't completely reliable, nor did a real-time system mesh with our belief that the game needed to gracefully handle frequent interruptions. The three options we considered were a proximity based connection via Bluetooth or ad hoc WiFi, a pass-and-play system where two users take turns playing on a single device, or an asynchronous system that could be played online.

We decided on an asynchronous system, and built a lightweight server



# it's your move

the next step for strategy games

HIGHBORN.

that stores and passes game states between players. Since launch, there's been user demand for a pass-and-play system and it's something we're looking into building for future revisions.

We also wanted a lighter game that was easy for people to pick up and play, to put down when they were done, yet have it be compelling enough for them to come back to again and again. With the *COMMAND & CONQUER* games we built, it was remarkably easy for users to lose track of time and not realize that they had spent over an hour playing a particular mission. We took this as a sign of a well-designed mission since one of our goals was to create as immersive of an experience as possible.

However, this isn't something we wanted in a game that players can pick up and play anywhere at any time. Sure, we wanted players to get engaged with the fiction and feel compelled to continue because they were having fun, but we felt it would be irresponsible to punish them for putting the game down when something more important happens, such as their subway train arriving.

We sacrificed some of the traditional strategy game elements such as unit production, the ability to carry over your forces from one mission to the next, and direct control of units during combat. Resource gathering was discarded in favor of a simple approach that granted players units upon capturing certain structures. Base building was also removed; instead, players can capture and hold certain structures which provide combat support for any battles that take place within the structure's support range. Figure 3 illustrates where players spend time in *HIGHBORN*, with combat taking the majority of time followed by information (checking for hidden units, studying enemy structure and unit support ranges) and base and army management.

While we took some hits from traditional game reviewers who felt *HIGHBORN* was too simplistic, we feel we were very successful in reaching our target audience as reviewers of mobile games and thousands of our customers gave *HIGHBORN* consistently high marks.

## WHERE IS IT GOING NEXT?

» To figure out where strategy games are going next we need to take a look at where the genre exploded in popularity, specifically the birth of real-time strategy games. Before the RTS, turn-based games had become an increasingly niche market in no small part due to its ever-increasing complexity. This is exactly the same problem that's crept into RTS games over the past decade, to the point that some of today's RTS games rival the complexity of a turn-based game. The worst part about this is that most of these features have done little or nothing to make the games more fun and have distracted players and designers alike from focusing on core gameplay. There are a lot of reasons for this, not the least of which is a desire to stand apart from the competition or to add a marketing bullet point on the back of the box, but it has to stop.



Merriam-Webster defines strategy as “the science and art of military command exercised to meet the enemy in combat under advantageous conditions.” Why is it that so many strategy games focus so heavily on tactics? When you have a ton of units, more structures than you can remember, and so many factions that any sort of meaningful differentiation between sides is basically impossible, how can you possibly engage in any real strategic decision-making? Given all this, what will the future bring for strategy games?

I think we'll see strategy games take a few different paths. There are several titles that have forgone radical innovations and instead focused on finely polishing tried and true strategy gameplay. Blizzard has done this with *STARCRAFT 2* to great success, and Firaxis also stuck with proven and successful gameplay for its recently-released *CIVILIZATION V*. Yes, both of these titles are sequels from highly successful franchises, but I don't think that should diminish the point that you don't have to reinvent the wheel with every title to make it good.

Developers have just scratched the surface of what's possible with strategy games on mobile platforms. There will be more tower defense and turn-based strategy games. Our company, for example, is working to extend our *HIGHBORN* app by bringing it to additional platforms and adding new features. There is so much room for innovative game design on mobile platforms. Shorter development schedules and relatively lower budgets compared to PC and console games allow designers to take more risks. Imagine a game based on Foursquare's location-based check-in model, where you capture and hold real territories for strategic gain in a version of an alternate reality game. How cool would it be if the entire world, via Google Maps, were your battlefield?

We'll also see strategy games hit big on consoles, but we shouldn't expect them to look anything like their PC counterparts. Trying to

shoehorn PC RTS mechanics onto a console by coming up with a clever control scheme just isn't going to work. Designers need to go back to the fundamentals and rethink what a strategy game means, what is the absolute core gameplay mechanic, and what is the minimal feature set required to make the game fun. Compounding the problem of console strategy games is the lack of confidence in the genre that most publishers have, given the less-than-stellar sales of existing releases. I think the best bet is to continue development of strategy console games for digital distribution via XBLA and PSN. *TOY SOLDIERS* on XBLA and *FAT PRINCESS* on PSN are two great examples of strategy games on consoles that work quite well.

Finally, I think there are a lot of great opportunities for a social strategy game either via Facebook or perhaps a MMO-type mechanic. The few existing web-based strategy games I've played, while fun, have been overly focused on hardcore players and require near around-the-clock attention in order for you to reach the higher ranks. A *CIVILIZATION*-type mechanic running online via a persistent world could work very well.

Strategy games may no longer be one of the biggest genres in gaming but I'm optimistic that they can shine again. There's a generation of gamers who grew up playing *COMMAND & CONQUER* and *WARCRAFT* and are now working in the game industry. New technologies, interfaces, and platforms are being developed that enable brand new ways to interact with a virtual battlefield and create fun new gameplay in the same way the mouse-and-keyboard combination did with *DUNE II* almost two decades ago. I'm looking forward to seeing what comes next. ☐

**RADE STOJAVLJEVIC** has worked on major franchises at companies such as Westwood Studios, Electronic Arts, Activision, and Sony Computer Entertainment America. He is the co-founder and president of Jet Set Games.




## TAKE THE STAGE

SHOWCASE AT GDC | CELEBRITY JUDGES | OVER \$50,000 IN PRIZES



Enrol now at <http://gametree.tv/competition>





# Game Developers Conference<sup>®</sup>

**February 28-March 4, 2011**

**Moscone Center | San Francisco, CA**

**Register before January 24 and save up to 35% off!**

Visit [www.GDCConf.com](http://www.GDCConf.com) for more information.

# GDC



## Celebrating 25 Editions

### Learn

#### Summits and Tutorials (Monday - Tuesday)

- AI
- GDC Education
- Game Localization
- GDC Smartphone
- Independent Games
- Social & Online Games
- Serious Games

#### Main Conference Sessions (Wednesday - Friday)

- Audio
- Business/Management
- Game Design
- Production
- Programming
- Visual Arts
- Monetization

#### Game Career Seminar (Friday)

### Network

#### Business Center (Wednesday - Friday)



@GDC

#### Game Connection@GDC (Tuesday - Thursday)

#### Career Pavilion (Wednesday - Friday)

### Inspire



#### Independent Games Festival (Monday - Friday)



#### Game Developers Choice Awards (Wednesday)

#### Expo Floor (Wednesday - Friday)

For almost three decades, the **Game Developers Conference®** has delivered pertinent industry updates in the trends, techniques, and technologies of game development.

Join us at the **2011 Game Developers Conference®**, February 28, 2011 - March 4, 2011, for five days of over 400 sessions, as we cover a comprehensive selection of game development topics taught by leading industry experts.

# ivy the kiwi



K O H E I T A K A S E A N D Y U S A K U Y A M A N A K A

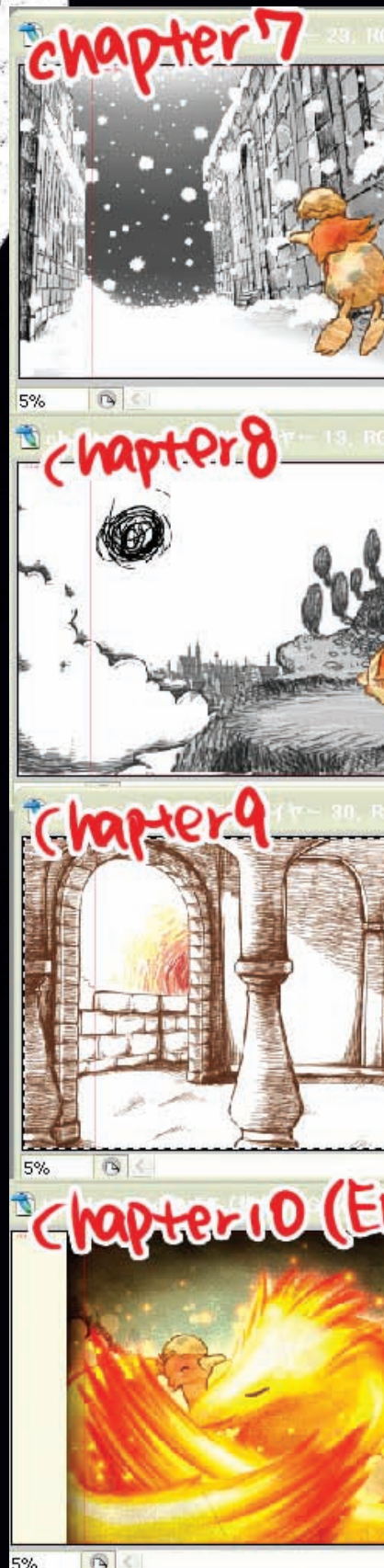
IVY THE KIWI? is a platform game in which the player doesn't actually control the character on screen, but instead guides them to their destination via the DS stylus or Wii remote, drawing lines and catapults to help protagonist Ivy reach her destination. The game's development was very different from the traditional process. Here, we present to you how Yuji Naka's studio PROPE Ltd. created this title, as well as how we announced it to the world.

## WHAT WENT RIGHT

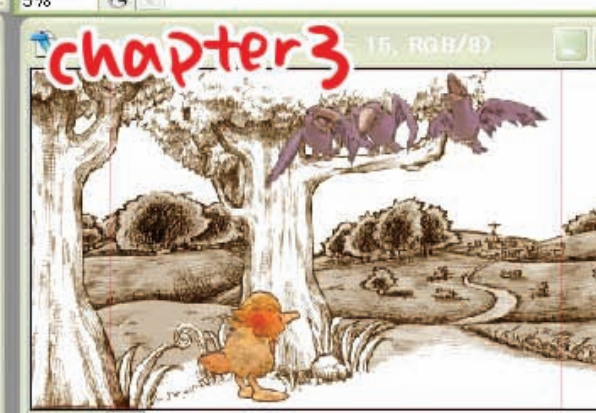
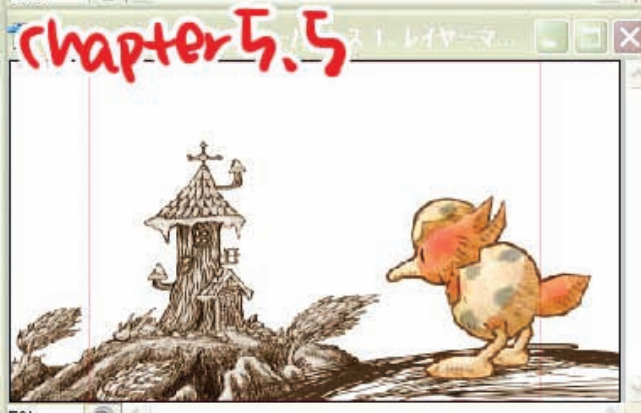
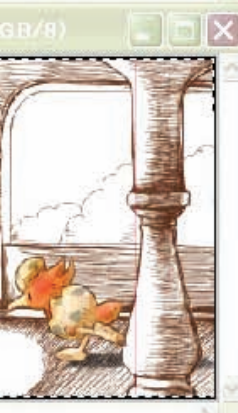
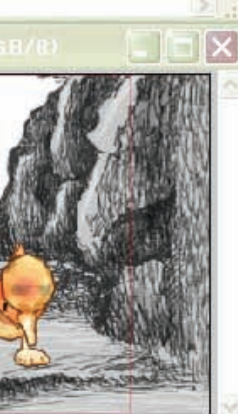
**1} GAME CREATED IN "NEW HIRES" TRAINING SESSION.** IVY THE KIWI? was created by three newly employed staff members who had to make a mini-game in one of their training sessions.

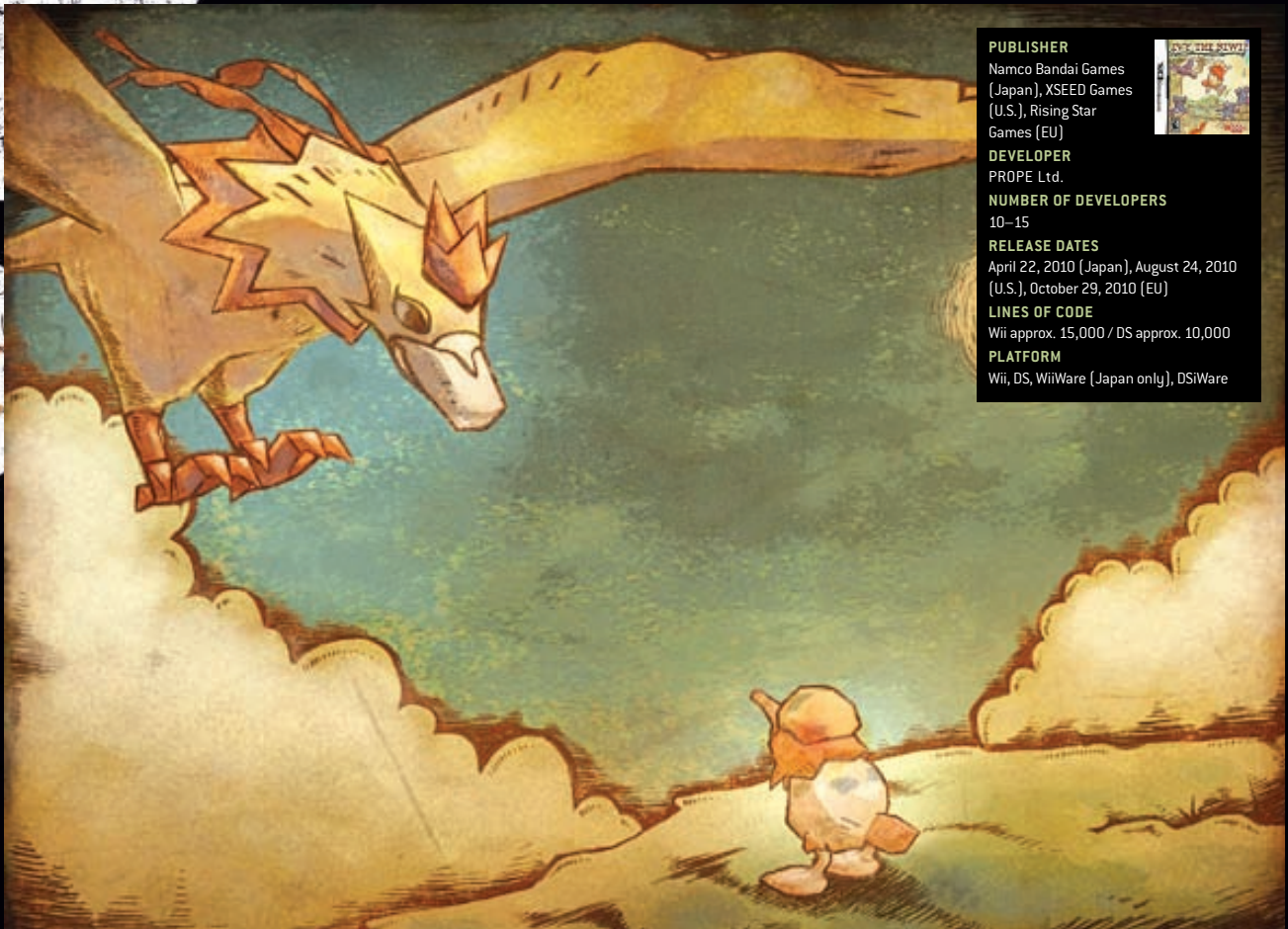
At PROPE, every year, when we hire new employees we customarily divide them into teams and have them create a mini-game as one of their training exercises. The development period is one month and we set the platform as Wii, but other than that, each team is free to create any type of game they would like.

The planner at the time (who would become the director for IVY THE KIWI?) had an idea to "create a game with a soft and squishy creature that you pinch and extend." On the other hand, the designer's idea was to create "a character that keeps running while the player creates clouds to prevent the character from falling." Both staff members felt strongly about their ideas and didn't want to back down, so when they almost got into an argument, the programmer said, "What about a game that incorporates the two together?" So he created a program that formed soft clouds, which later became the vines. It was his way of mediating the disagreement, but the program became the prototype for IVY THE KIWI?









**PUBLISHER**  
 Namco Bandai Games (Japan), XSEED Games (U.S.), Rising Star Games (EU)

**DEVELOPER**  
 PROPE Ltd.

**NUMBER OF DEVELOPERS**  
 10–15

**RELEASE DATES**  
 April 22, 2010 (Japan), August 24, 2010 (U.S.), October 29, 2010 (EU)

**LINES OF CODE**  
 Wii approx. 15,000 / DS approx. 10,000

**PLATFORM**  
 Wii, DS, WiiWare (Japan only), DSiWare

**2) FUN IS JUSTICE! (PROPE MOTTO)** At the initial concept review for the mini-game, which the three new recruits called “Pitch” (this was the main character’s name, which was a baby chick at the time), the gameplay was more puzzle oriented. After this presentation, the programmer thought that the game would be a lot more fun if the player could launch the character rather than just placing pathways. However, there wasn’t much time until the final presentation and changing the specs at that point would have been very risky. After talking it through with the team, they decided that “fun justifies everything” and decided to go through with the change. They wanted to emphasize the concept and changed the character from a baby chick to a flightless kiwi bird, and changed the clouds to vines. At this point though, it was a game where the player just bounced a helpless kiwi into the air.

The map was just a simple infinitely scrolling platform with floating islands. The final presentation was only a week away and the planner came to the conclusion that there was no way they would be able to create a game with multiple stages while incorporating puzzle elements into them. He drastically changed the gameplay from puzzle to action-based and created blocks so there would be no “death by falling” elements. They created several dozen stages and selected the ten best for the presentation.

The initial prototype was finished on time and Kiwi got a lot of praise at the presentation. Then the president of PROPE, Yuji Naka, said, “Let’s sell this as a WiiWare title.”

**3) EVERYONE IN THE COMPANY PARTICIPATED IN CREATING THE TITLE.** When people join PROPE, they’re all surprised at our method of game creation. We really try to help one another despite being assigned to different projects. This is because we look at our titles as if they are our children, and we want to help one another like family. This is similar to the culture that Yuji Naka utilized during his days at Sega, when he was leading Sonic Team.

Now that Ivy was moving forward as a WiiWare game, there were a lot of ideas sprouting from both inside and outside the project team.

Some examples:

- /// Maybe Ivy’s speed is too fast. Let’s drastically cut it and make it 1/3 of the current speed.
- /// Can Ivy not perform any attacks? We can launch Ivy using the vines to slingshot her.
- /// Shouldn’t the gameplay be more interactive? Let’s create special blocks that can be destroyed and place a massive amount of them in the stages.
- /// The gameplay is very simple. We should add special items to find and collect.
- /// Would like to make reaching the goal more fun. We’ll make the goal into a podium and depending on where you land, you get a different amount of points.

There were especially a lot of ideas from Yuji Naka. Other than the original three members, Yuji Naka probably played the largest role out of the whole company. He discussed his thoughts with the director multiple times as the game slowly evolved into its current form.

**4) GRAPHICS AND THE STORYBOOK-LIKE STORY.** The story surrounds a helpless Kiwi running with all her might to find her mother.

The art director wanted the game to be unique by creating graphics that “look like they are from a heartwarming storybook.” He thought this would express the game’s emotion and sentiment. He went through a lot of concept art and decided to try out line art, which he happened to be researching at the time.

There were some difficulties, but with adjustments, the Kiwi eventually looked like it was from a hand-drawn storybook, which you see in the final version. After seeing this graphical style, Yuji Naka thought that the game’s story no longer fit the visuals.

During this time, the game was very simple and only a single end screen. However, during the development process, the stage count grew to 100 stages, and as players progressed through the game, the difficulty level ramped up significantly like an action game, so we needed a sufficiently satisfying ending after clearing such a challenging game.

Naka proposed the idea, "What if the player gets to read the storybook when they finish the game?" We then had a few staff members draft up a story explaining why Ivy can't fly, the meaning of the shell around her, why she's heading toward the sky, the meaning of her red feathers, et cetera, and we ended up with a story about a baby Kiwi being born without her mother in sight.

As a side note, PROPE turned the ending into an actual storybook, which is unfortunately only available in Japan.

**5} WORLDWIDE SALES.** At this point, *IVY THE KIWI?* had enough content to not only be a downloadable game, but a packaged title as well. Naka decided to create the title for Wii, DS, and DSiWare. We were able to find publishers for various territories: Namco Bandai Games for Japan, XSEED Games for North America, and Rising Star Games for Europe.

So at last, the baby bird that started out as a new employee training program, loved by all of us at PROPE, was able to fly off to the rest of the world.

#### WHAT WENT WRONG

**1} THE SCHEDULE KEPT CHANGING.** As mentioned before, this project started off as a training exercise, and was originally planned as a WiiWare title. Since Naka constantly has new ideas on his mind, he kept on adding new content even after the beta version was created, saying, "Fun is the justification, right?"

From there, even though we didn't have any publishers lined up, we created the Wii and DS versions, and needed to revise the schedule for all the versions. When we finally finished everything, it was months past our original schedule.

I think it took the biggest toll on the quality assurance staff. They probably thought, "How many beta versions do we need to play for this title?"

In the end, it was all worth the trouble, but for the next project, everyone at PROPE sincerely hopes we know the platforms we're planning to make the title available on from the beginning.

**2} SUPER-PLAY FOOTAGE.** The producer, Yuji Naka, was deep in thought after the game was created. "What would be the best way to show what type of game this is and how fun it can be?" he asked.

It was determined that the best way to make the game attractive for action game fans was by showing gameplay videos. With this in mind, Naka asked the development members to come up with a super-play video and to distribute it on the web.

It was hell from there. If you made a small mistake, you needed to reshoot the entire stage. After playing the game multiple times for these super-play videos, one member of the team became so good that no one else in the company could compete with him. Others tried to mimic what he could do but couldn't pull it off the same way. In the end, he became the go-to guy for any super-play videos, even if he was in the midst of another project. Sometimes being too good can be a double-edged sword.

**3} CHANGING THE COLORS.** The graphics in *IVY THE KIWI?* differ between the Japanese version and the U.S./EU version.

In the Japanese version, it has more of a sepia tone; however, the U.S. and EU versions are more colorful due to the publisher's request. Since the schedule for the U.S. version was coming up, it was dangerous to attempt to change the coloring of the entire game without making it look weird.

In the end, the graphic adjustment went well and the coloring process was a success, but we had to re-take hundreds of screenshots and re-record the super-play video.

**4} SYNCHRONIZING THE NETWORK.** Unlike the split-screen multiplayer mode on the Wii, the DS version utilizes network communication for multiplayer. This network communication was the toughest challenge in creating *IVY THE KIWI?* For a fast-paced game that rapidly creates and

swings vines, network communication did not synch well with the gameplay, especially with four players at once.

In the end, we needed to bring on programmers from different projects, and were finally able to create the network multiplayer mode, but it took a lot longer than expected. The DS director swore to himself that the next time he has a DS networking game, it will definitely have much slower gameplay than *IVY*.

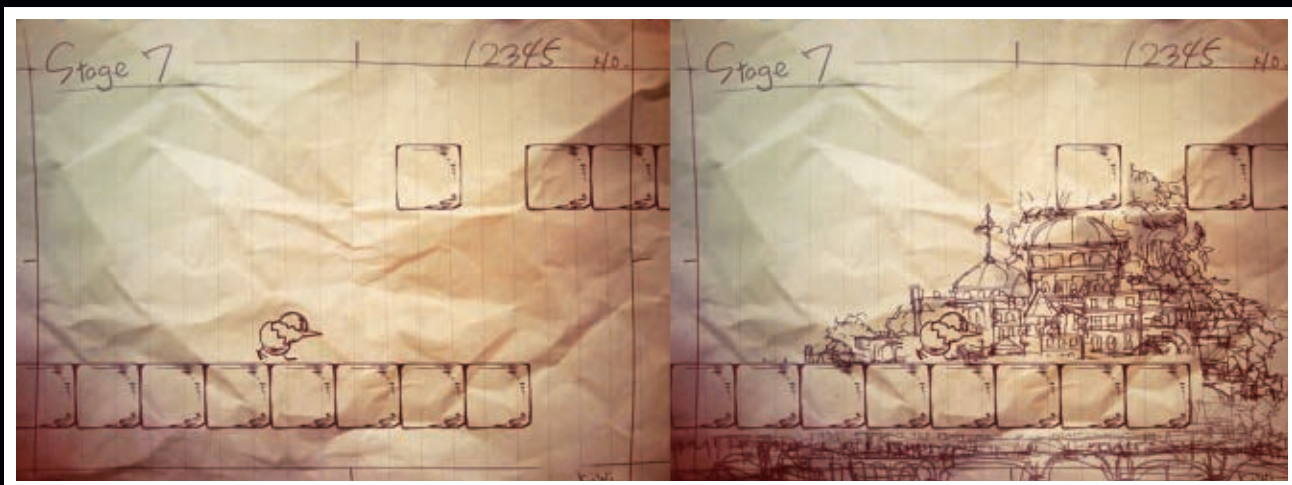
**5} DEBUGGING.** At the end of the project, a new element, the "medal system," was introduced to the game. The medal system awards players various medals for collecting all of the red feathers in a particular stage depending on their finishing time. The gold medal, which is the hardest to obtain, made the QA team cry. Also, when you obtain a gold medal in every stage, a feature which shows what medal you have for the entire game is unlocked.

Of course, this meant that only a select few testers could debug this function, so every time there was a new build, these elite in-house players (including the super-play staff member) needed to get all the gold medals to make sure everything was okay.

#### FROM MINIGAME TO BOXED PRODUCT

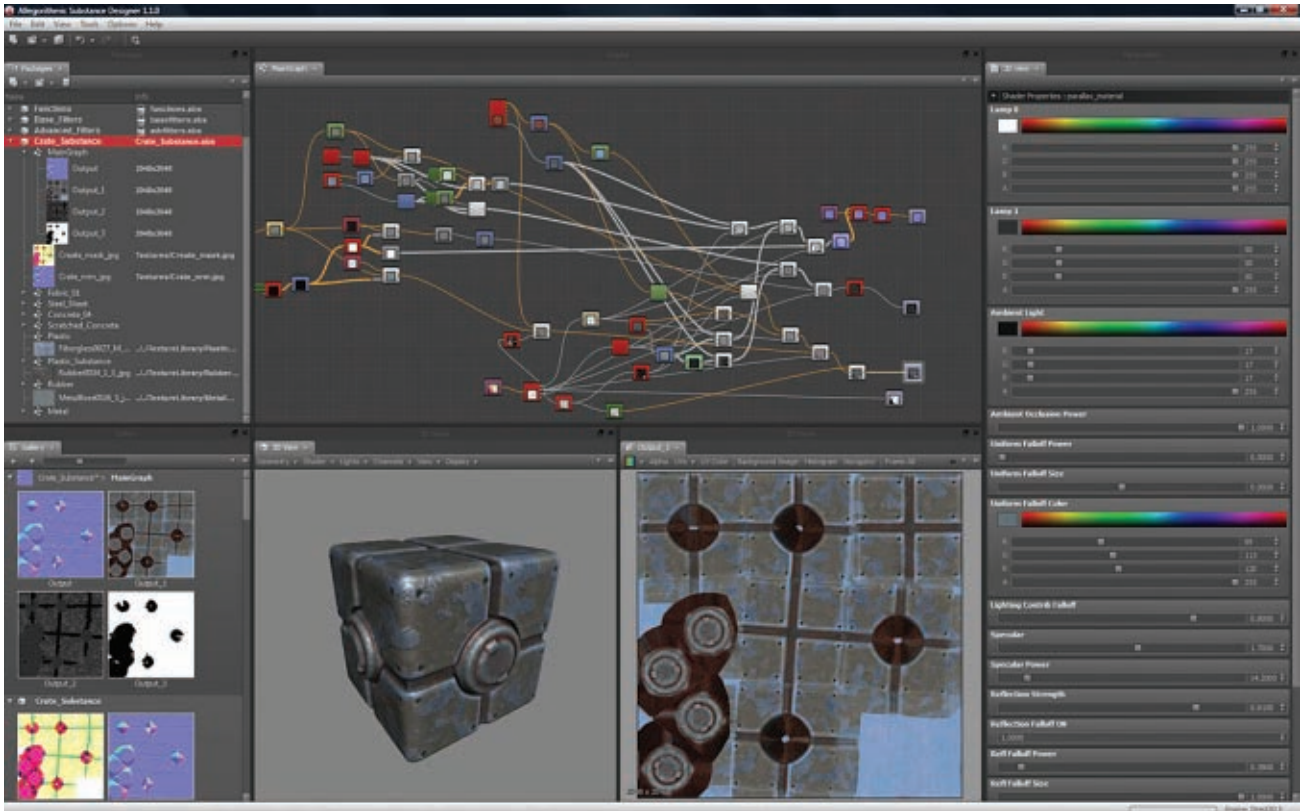
In the beginning, we didn't think it was possible to have an action game where the player controls the character with the Wii pointing system without becoming frustrated by not being able to perform sophisticated movements. However, our three new employees presented their fresh new idea, and in *IVY THE KIWI?*, you will be surprised at how precisely you're able to control the character and how freely you are able to control Ivy once you get used to the mechanics. We all got to discover a new type of gameplay by thinking outside the box, and I think that was the greatest achievement for all of us. 🐣

**KOHEI TAKASE**, director (*Wii*) and **YUSAKU YAMANAKA**, director (*DS*). Translation provided by **JIMMY SOGA** at XSEED Games.





# ALLEGORITHMIC SUBSTANCE DESIGNER



WE ARE WITNESSING THE BIRTH OF a new generation of content creation tools, and “procedural content” is definitely on the minds of many game creators. As projects become more visually compelling and require more time to make, artists are forced to find ways to be more efficient in their creative process while still maintaining the level of quality required by next-generation games.

The word procedural in itself may still sound a bit scary to some, as it sounds like one’s talents might be given over completely to the computer. But using procedural tools for asset creation can give you more data to work with, more time to iterate assets, and jump-start your creative process by automatically building data that used to be

mundane and time-intensive to generate. Adopting more procedural tools into your workflow can make your pipelines more efficient and allows for more flexibility in the visual fidelity of your assets.

Procedural applications are gaining in popularity and Allegorithmic has just added a set of very powerful, easy-to-use procedural tools to the game developer’s toolkit. Substance Designer, a stand-alone application for texture authoring, and Substance Player, for texture editing and rendering, are interconnected tools that give game developers a way to produce “intelligent” texture information that can be generated and modified on-the-fly. No longer do we have to rely on vast libraries

to texture our environments and characters. When it comes time to polish a level, we don’t have to worry about storing tons of large PSD files in case we need to edit a texture.

Substance Designer is on the front end of this new texturing pipeline. It is the application that generates all the data needed to create a procedural texture, whether it’s based off of preset procedural noises and patterns, imported bitmaps, or custom filters that your technical art team generates.

Using a visual node-building process, you can quickly network together bricks of logic in order to generate the basic element of a procedural texture, called a substance. An artist or technical artist can build up huge libraries

**The Substance Designer interface is shown, along with a typical project setup view.**

of customizable substances that represent elements such as dirt, rust, rivets, moss, and rock, which can then be used by the rest of the art team in their higher-level substances. This is one of the key concepts behind using the Substance Designer—break down your textures into reusable components that can be used in other substance textures. Once these components are created, they can be reassembled into larger graphs that ultimately create your final tiling texture.

The benefit to this component approach is that textures created in

Allegorithmic's Substance Designer are resolution independent. This means if your texture requires more resolution, an artist doesn't need to use a combination of bitmaps and filters to re-author a new texture in Photoshop. They can just go back into a created substance and increase or decrease the resolution, and the changes will be propagated down to all components of the substance, i.e. diffuse map, specular map, height map, normal map, and so on. This feature alone makes Substance Designer one of the most powerful software applications for generating texture data, whether for prototyping or full production. However, there did appear to be some software slowdown when working with very high-resolution textures.

There have been a few applications that boast procedural texture creation, such as Genetica and Dark Tree, but none of them have offered the end-to-end solution that Allegorithmic has provided with Substance Designer and Player. Each of these applications provides its part in the creation, compression, and streaming of textures in a real-time environment.

### THE GENESIS PROJECT

» So, what is the process of creating a new substance in Substance Designer? It can actually start in a few different ways. An artist can utilize a set of pre-built noises, patterns, and vector graphics, or import bitmaps of rocks, dirt, and rust to start creating their substance. A technical artist can spend some time pre-building a library of procedural dirt, rust, and rock using their own custom filters and FX maps that can then be composited by another artist. Or you can just use Substance Designer's Bitmap to Material nodes to generate automatic tiling textures with accompanying diffuse, specular, ambient occlusion, and normal maps.

This flexible approach allows you to pick the method that best fits your project's needs. Allegorithmic has also produced a large set of pre-built substances, available through Turbo Squid, which you can

use as the starting point for your substance library.

Once these base substances are assembled, an artist utilizes instances of them to composite together their own new texture using a set of robust Photoshop-like filters, adjustments, and transformations. This is my favorite aspect of Substance Designer, as I can create large texture libraries and not have to rebuild my texture information every time I start a new asset. I can start by creating substances that are just general rock, dirt, and skins, and then use them to composite together new substances very quickly using things like HSL adjustments, scaling/rotating/moving, blurring, drop shadows, and blending modes. The list of available filters is almost endless, and Substance Designer gives you all the Math nodes you would need to produce your own custom filters if the pre-built ones don't meet the needs of your project.

Normally, I'm a big fan of visual node networks to build up logic, but there are times when the networks can get quite large in the Substance Designer interface. However, if you bundle up the networks you create using a component mindset, you can avoid the resulting tangle of nodes and node connections that often results. This node-based approach to generating texture data creates an environment that is both collaborative and iterative, making your textures very flexible so they can change along with the needs of your project.

At the end of constructing all this texture data, you simply drop down Output Nodes to output textures such as diffuse maps, ambient occlusion maps, cavity maps, specular maps, normal maps, and height maps. It's very similar to what Crazy Bump and Xnormal produce, but with flexibility and re-usability.

With a substance completed and saved, an artist or art director can access the texture through the Substance Player. The Substance Player environment provides a very intuitive interface for iterating and customizing the substance

to fit the needs of the project, without having to learn how to construct substances. Using exposed parameters which are set up in Substance Designer by the technical artist, an art director can change everything from the color of a texture and normal intensity, to the amount of specular and ambient occlusion, through to how much moss is growing over the details in the texture. The list is endless, as just about every slider, integer value, and color swatch can be propagated up to the Substance Player as well as changed on the fly in Maya and 3ds Max, or even in your game engine directly.

Also, the latest releases of Maya and 3ds Max provide a new shader that can import the Substance format directly. This allows an artist to use their substance while they are building

a level or character and change the elements of the substance as needed. This makes the process of texturing very iterative, as your textures evolve along with your environments and characters.

Maybe you'll decide that you need more sand on the tops of your rock textures. Instead of having to go back into Photoshop to re-author your texture sets, you simply pull in a pre-built sand substance and composite it with your rock substance, propagate the necessary tweak-able parameters, and save out your new substance. Load it back into Maya or 3ds Max and you now have sand on your rocks. You can also adjust the amount of sand and the softness of the falloff from sand to rock. Once the final look has been found, you can render your textures to final bitmaps using the Substance Player.

I CAN'T EMPHASISE ENOUGH HOW IMPORTANT THAT DEVELOPMENT DEADLINE IS

IT'S A TEAM EFFORT AND WE ALL NEED TO DO OUR PART

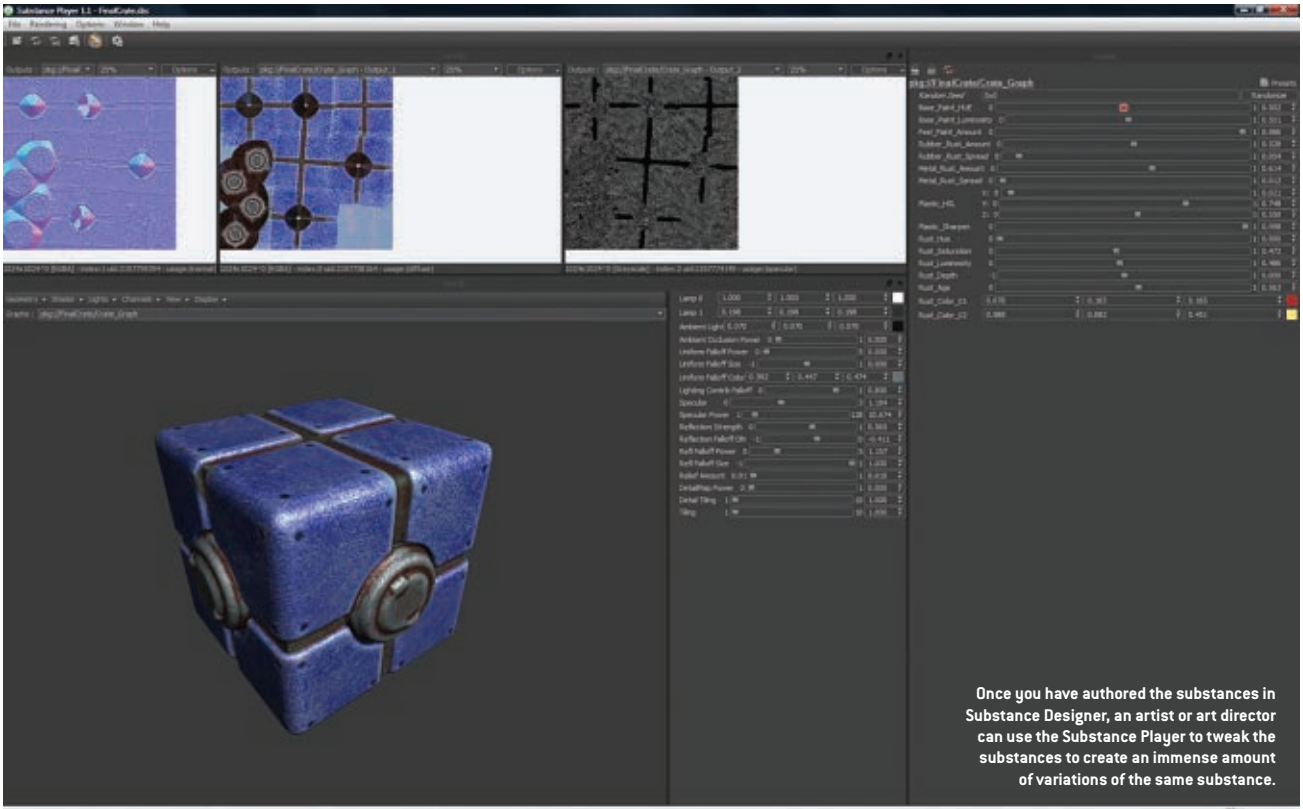
YES, I'M RUNNING THE BUILD RIGHT NOW

!?!

I AUTOMATED THE BUILD WITH FINALBUILDER

**FINALBUILDER**

[www.finalbuilder.com/game](http://www.finalbuilder.com/game)



Once you have authored the substances in Substance Designer, an artist or art director can use the Substance Player to tweak the substances to create an immense amount of variations of the same substance.

## GETTING UP TO SPEED

» At the end of the day, having a toolset like Substance Designer greatly speeds up the process of creating textures as opposed to creating them from bitmaps in Photoshop. It also creates an environment that promotes iteration

and exploration of variety in your textures, since you don't have to worry about re-authoring a texture just to change a color or add more rust to some metal.

It does, however, shift the texture generation pipeline and will require training, and setup within

your team. This shift is definitely one of the toughest items that your art team will face. Most artists are very familiar with using Photoshop to generate textures from their large libraries of bitmap images. Understanding how to use noises to generate a stone that you see in a reference image isn't something that the industry is used to.

Allegorithmic has the base training in place and does offer on-site training with your team, but I feel that they could produce some more pipeline-centric training videos. This would help teams become more familiar with the Substance Designer workflow as well as how to generate texture information from noises in general.

Ideally, a team composed of technical artists would take some time to learn the ins and outs of Substance Designer to create libraries of general material types, such as moss, dirt, rust, metals, and masonry. The artists on the team responsible for content creation would also need to get trained on how to use Substance Designer and Substance Player, but

only enough so they can modify and do the compositing of the pre-built substances created by the technical artists. These composited substances could then be used by character artists and level designers to populate scenes and then tweaked to achieve the final look. This would be the ideal setup, but every project is different, so this pipeline might be set up a bit different from studio to studio.

## SPONTANEOUS GENERATION

» We are moving into a new era of content creation where procedural tools are going to be a staple of our everyday content creation. Allegorithmic has produced a world-class set of software that allows content creators the ability to take advantage of procedural technology without feeling like the machine is doing the creative part for them. It still leaves you in control creatively, while giving you the flexibility needed for next generation game development.

**KENNETH LAMMERS** is a technical art director at Microsoft Game Studios.

## ALLEGORITHMIC SUBSTANCE DESIGNER

Allegorithmic SAS, 31 rue Gonod, Etage B1, 63000 Clermont-Ferrand, France  
[www.allegorithmic.com](http://www.allegorithmic.com)

### PRICE

> \$990. Volume discount granted for multiple seats. Substance Player is included with Substance Designer and can also be downloaded as a free standalone from [www.allegorithmic.com](http://www.allegorithmic.com).

### SYSTEM REQUIREMENTS

> PC only.  
Windows 98, XP, Vista, Windows 7.  
X86 CPU with SSE2.  
GPU that supports DX9, DX10, shader model 3.0 minimum.

### PROS

- 1 Textures are resolution independent.
- 2 Textures can be easily edited by non-technical artists using Substance Player.
- 3 Integration with Maya and 3ds Max.

### CONS

- 1 Software slows down at higher resolutions.
- 2 Needs more organizational tools for complex node networks.
- 3 Needs more training materials.

## product news

### Hansoft 6.5 Released

[WWW.HANSOFT.SE](http://WWW.HANSOFT.SE)

Hansoft has launched Hansoft 6.5, the latest version of its project development toolset that adds features to help simplify global collaboration in game development.

New features in version 6.5 include localization for Japanese and Chinese users, better support for outsourcing and distributed collaboration.

Hansoft is a solution for team management that includes features for scheduling, real-time reporting, bug-tracking, and other components of software development.

### Havok Strike Program

[WWW.HAVOK.COM](http://WWW.HAVOK.COM)

Middleware maker Havok announced the upcoming launch of the Havok Strike Program, which will offer basic access to the company's suite of products for a reduced cost.

The new offering is targeted at developers making smaller-budget and casual titles pegged at a lower price point. The company said Strike Program participants

will be able to purchase access to any or all of Havok's full suite of middleware products, including AI, animation, physics, cloth behavior and destruction modeling tools. Participation in the program will also include access to Havok's technical support infrastructure.

### Mixamo and DAZ 3D Partner Up

[WWW.MIXAMO.COM](http://WWW.MIXAMO.COM)

3D and Mixamo announced an alliance that brings Mixamo's high-quality animations and real-time customization capabilities to the DAZ 3D character model portfolio.

Users of DAZ 3D's latest generation human figures now have access to the entire Mixamo motion collection at [www.mixamo.com](http://www.mixamo.com). Every Mixamo motion is guaranteed to work with any character created from the DAZ 3D human figures within DAZ Studio.

DAZ 3D community members can visit [www.mixamo.com](http://www.mixamo.com) and see any motion applied to the DAZ 3D standard characters, Victoria and Michael (with the Kids forthcoming), in the Mixamo 3D

viewer. They can also customize these motions, then download and merge them back into their scenes in minutes. Users can also purchase Mixamo motions applied to the DAZ 3D characters by visiting the DAZ 3D storefront at [www.daz3d.com](http://www.daz3d.com), or with special distribution licensing in the DAZ 3D developer storefront at [developer.daz3d.com](http://developer.daz3d.com). Workflows and tutorial videos are available that outline the entire end-to-end pipeline between Mixamo and DAZ 3D.

### Scaleform GfX 4.0

[WWW.SCALEFORM.COM](http://WWW.SCALEFORM.COM)

Scaleform Corporation, a provider of Flash-based middleware and user interface solutions for the video game and consumer electronic industries, announced that GfX 4.0 will support Flash 10 ActionScript 3.0, as well as boost performance by at least 2x over previous versions.

Full support for the faster and more robust ActionScript 3 gives game developers the ability to create more impressive menus and interfaces, while also opening up new doors for casual game developers to port their web games to native apps on mobiles and consoles.

### NeoReel's MoClip

[WWW.MOCLIP.COM](http://WWW.MOCLIP.COM)

To better serve the 3D needs of the game, film, and television industries, NeoReel unveiled its latest MoClip tools and services, which aim to make archiving, sharing, and blending 3D character animations easier.

NeoReel has developed a unique cloud computing technology which allows individuals and production facilities to convert character animations from past, present, and future projects into a unified library accessible through a standalone application called MoClip.

MoClip allows for quick access to a private animation library or an ever-growing public animation library through a specialized search

engine, in addition to providing efficient animation blending tools. MoClip can drastically cut the cost and speed up projects like pre-visualization, game cinematic, crowd simulation, and the like, without many changes to an existing pipeline.

Users may also choose to make parts or entire animation libraries public to generate royalties from sales to the community.

The latest version of MoClip features strong security measures and corporate accounts with multiple users and privileges. It has improved intelligent blending, a stronger search engine, and custom animation categories. Autodesk MotionBuilder and Autodesk Maya are supported. MoClip allows animation uploads to a private clip library, a public request and royalty program, and also offers detailed help documentation and video tutorials.

The MoClip tools are immediately and freely available on the [www.moclip.com](http://www.moclip.com) web site. The MoClip service fees and royalties depend on the usage of the tools and are described in registered accounts.

### FMOD Bundles with Vision Engine

[WWW.TRINIGY.NET](http://WWW.TRINIGY.NET)

Trinigy, the company behind the Vision 3D game engine, announced it would add Firelight Technologies' FMOD audio suite to Vision Engine 8.

FMOD will be included for free to Vision Engine licensees. The engine, used in games including Ubisoft and Blue Byte's THE SETTLERS 7: PATHS TO A KINGDOM, will still support other third-party audio solutions in addition to the FMOD Ex suite.

Both companies agreed on joint developer support for Vision and FMOD. The audio solution supports Windows, Linux, iOS, Xbox 360, PlayStation 3, Wii, and web browsers.

Vision and FMOD had previously been integrated, but the new agreement means developers can access all of FMOD's features directly through the Vision Engine SDK.



DAZ 3D's Victoria standard character model.



# SCOPED AND GROPED

## FROM TSA TO FBX

### AS A TEENAGER, I ONCE TRAVELED

through what was still, at the time, Communist Yugoslavia. The border crossing was a straight John LeCarre rip-off: wispy snow blowing through an unheated train station, a giant red star on the wall, lit by a single, unshaded lightbulb, and enormous border guards in fur hats and greatcoats. If you didn't mind the intimidation factor (or standing on a Serbian railway platform in the snow for an hour), it was a very ... informative ... look at the machinery of authoritarianism. It was also, alas, less intrusive and demeaning than catching a flight to Newark today. And, amazingly enough, it actually relates to game development—really.

You can't run anything—a communist dictatorship, an airline, or an innocent game project—without well-policed boundaries. In our case, that means keeping bad data out of the game: making sure that our little errors can't mess up performance, create bugs, or blow budgets. If you've been responsible for an embarrassing howler (I shudder to recall a character with 4 identical meshes overlaid on top of each other, who looked fine but dragged the game down to 6 fps whenever he appeared on screen), you want your data to be thoroughly scoped and groped on its way into the game. Enhanced productivity is great, but keeping your teammates from knowing what a dork you are is priceless.

Of course, as we know a bit too well these days, good security involves a balancing act. A process that's too intrusive or too complex becomes so discouraging that the whole enterprise (game development or airline travel) suffers. So this month we're going to look at the TSA screener of game development: the humble exporter plugin.

### PLEASE PLACE YOUR LAPTOP IN A SEPARATE BIN

» Exporters are nearly universal in game development. There are a few notable exceptions (for example, the popular Unity middleware engine reads Maya files directly) but most studios opt for some kind of plugin that reformats art data for use in game.

There are good reasons why exporters are more common than file importers. Partly it's because nobody wants to re-write their pipeline whenever Max or Maya revise their file formats. But more importantly, the act of exporting makes a useful distinction between simply saving your work and updating the game. For old Max and Maya hands, who habitually tiddle control-S after every vertex push or stack operation, that's an important point.

There's a lot of information in our files that artists need, but which the game doesn't want to know about. There's construction history, layers, groups, reference material, experimental variants, and all the other gadgetry that keeps us productive but has no place in the game. Keeping the debris of the creative process out of the way of the engine is good for both sides: the engine doesn't need to learn your workflow to read your files, and you don't need to give up on tools you want in order to get your data into the game. Hence our friend, the exporter plugin.

### HAVE YOUR DOCUMENTS READY

» As a metaphorical "border crossing," exporting offers a great opportunity to do a little pat down on your data and make sure it's not smuggling something dangerous into your runtime. Every engine has a unique (and perpetually evolving) list of taboos. From zero-

area faces to untextured geometry to textures located in the wrong folder, there are a lot of things which could bring the game down if they made it through security.

Every exporter tries to enforce these rules to some extent. A good exporter enforces these rules smartly to make artists more productive by heading off predictable errors. Unfortunately, smart exporters are a distinct minority.

Most exporters provide feedback in the form of error messages or log files. Unfortunately, most of the time this information isn't intended for us: it's stuff the coders need during development. I suppose it's nice that the exporter tells me "WARN 0x004BC5> invalid texcoord @ 45?" instead of, say, crashing the game, but my gratitude has pretty strict limits for this kind of feedback. Often, when you don't get cryptic technobabble, you get so many trivial warnings that your eyes glaze over. I once knew a team that had to update their exporter because it generated so many error messages that the warning dialog ran out of memory. The fix, alas, was to write the spew out to a text file, instead of suppressing the useless messages or replacing them with something informative.

Verbose debug information can be very useful if you're a plugin developer. For the rest of us, what's needed are clear error messages that tell you what's wrong and how to fix it. A good exporter should act like an airline pilot's pre-flight checklist, making sure that all the necessary bases are covered and providing clear warnings when serious problems arise.

Embracing the checklist metaphor in the UI (see Figure 1) is especially useful. For one thing, it forces you to keep the list of possible problems short. If the technical requirements won't fit in a decent-sized dialog, the system

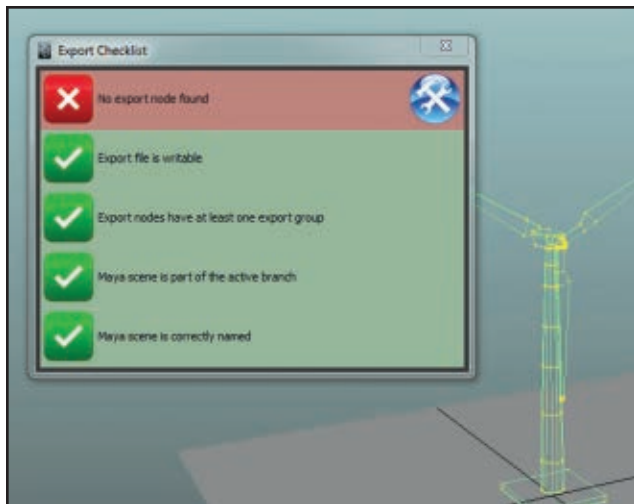
is way too cumbersome already. More importantly, a checklist-style dialog that pops up on each export gradually educates users to make the fixes by habit, gently nudging us onto the right path the way a good game design would.

The UI should also integrate tools that can fix problems as well as messages that point them out. Problems get squashed quickly when the exporter error dialog includes a link to the right tool, advanced documentation, or maybe a chance to IM the tech art team for help. It's important, though, not to try to be too clever. Exporting, like saving, should never be destructive, which means any fixup tools have to be opt-in, rather than automatic, and they should allow the user to save before altering the scene. You can't let artists lose confidence in the safety of hitting the export button. "Hey! I exported my scene and it popped up this error message, and now all the lights I just placed are gone!" is a recipe for mutiny.

### PROHIBITED ITEMS

» Of course, when most of us think of exporter UI, we're really thinking of the infamous settings dialog. Modern engines typically demand a lot of very exacting control over how data is laid out: geometry, animations, and textures have to be correctly arranged to make the most efficient use of the GPU and engine features. For example, most engines want triangles carefully arranged to maximize the reuse of vertices, or bone weights normalized to a particular tolerance. Animations typically need to be baked down from whatever system of IKs, constraints, and simulations the animator uses to a simple set of bone rotations. On top of all that, there are compression settings, optimization strategies, and all





**FIGURE 1** An exporter dialog that clearly tells you what has to happen before you can export correctly, and provides a tool for fixing the problem right when you need it.

sorts of other techno-minutia that matter a lot to the game, but rarely make sense to the person who's poking around in the dialog at export time.

When it comes to options, less is definitely more. It's usually possible to figure out the right export settings from context: if the file is in the animations folder, export it as an animation, but if it's not, then don't make the user uncheck "export animations" every time. Doing this smartly seems like a lot of work to tools coders, who'd rather just make some sliders and buttons and go home.

In the long run, though, complex options are very costly. Whatever runtime benefits you get from tightly controlled export settings have to be weighed against the productivity costs of user error and sheer bafflement. The person who wrote the animation compressor may know that the "angular epsilon" slider is supposed to say how many radians of slop are allowable in the compressed data, but the poor animator doing the exporting doesn't know a radian from a full-body radioscope. Asking users to type in all 16 numbers in an orientation matrix, or to choose the right byte-endian value—that's just asking for trouble.

Besides sparing us poor artists, keeping the export process

"settings-free" makes good sense for the pipeline as well. Pushing engine requirements all the way upstream into the organization of data in Max or Maya cuts down on technical flexibility. If changing the render pipeline involves manually re-exporting your existing assets, you can bet that production will shoot down a lot of possible improvements instead of sending the art team back to square one. A good exporter should insulate the art team as much as possible from the evolution of the engine, and vice versa. By all means, coders, swap from big-endian to little-endian or Y-up to Z-up as much as you like, as long as it doesn't mean I have to open the file and hit "export" again.

### PASSING THROUGH SECURITY

» The most common trick for keeping the source and game data at arm's length is to export the artwork to a neutral intermediate format, rather than exporting optimized game data directly out of the DCC tool. For example, CryEngine exports to the open-source Collada format and uses a dedicated resource compiler program to turn the Collada files into game assets. Ideally, all intermediate files would be standard formats like FBX or Collada, since free export plug-ins are already available for

those formats. Unfortunately, both formats suffer from inconsistent data handling and dodgy version control, so many engineers shy away from them and roll their own.

However they're formatted, intermediate formats can help rationalize the worst parts of exporting. The intermediate file imposes a useful barrier between the world of the artist and that of the game. Intermediate files don't need the tweaky optimizations that tie them to a particular console or engine version. They can be re-targeted to different platforms or new engine formats without involving artists. A cross-platform team can export each asset only once and reprocess it for use on different consoles. Best of all, the act of exporting isn't where the tweaky settings come in, so it may be possible to remove the dreaded exporter settings dialog altogether. No more epsilon radians for me, no sir!

Since the intermediate files don't depend on a particular tool, they are well suited for batch processing. An engineer doesn't need to open the Max or Maya files to change the formatting of the runtime data, which also means there's no need to call the art staff in over the weekend and manually re-export the contents of the game. Loss-producing steps, like animation compression, can be tried out quickly using data that's already been exported (few exporters can export animation without stepping through the frames one at a time, so long sequences can take a lot of time). It's the old Robert Frost adage: good fences make good neighbors.

Intermediate files do have one obvious drawback: they break the direct link between the asset we see in the game and the Max or Maya file that made it. This can be an expensive flaw if it creates confusion in the art review or debugging process. It's hard to tell somebody to fix something when they can't figure out what file created the glitch.

Naming conventions can help a little, but any solution based on the

clerical skills of artists is dubious at best. It's smarter to embed information in the intermediate file that provides an edit trail, such as a line in the file that says, "I was exported from blah\_blah\_blah.max." This will be far more reliable than an angry email from an overworked lead. Some teams even embed a copy of the original file right into the intermediate file. No matter what happens later, this allows them to see and fix the exact asset that's in the game—assuming, of course, that there's ample disk space and enough room in the source control server for all those big fat files. A related trick is to make a coordinated check-in that includes the source, intermediate, and game files in a single change list that can be retrieved later.

Whatever the means, the important goal is always the same: when a bug comes in, or when an angry art director sees something in need of a do-over, there should be no question about where the game asset came from and what has to be done to fix it.

### I'M GOING TO HAVE TO ASK YOU TO BEND OVER

» Most exporters enjoy about the same level of popularity as airport security, and for good reason. With confusing options and bad feedback, a mediocre exporter can definitely rub you (so to speak) the wrong way. But that's not how it has to be. Add some decent UI, informative error messages, and a little attention to artist productivity and the humble exporter will speed you safely on your way.

Of course, there's still the business of 20-inch-wide seats with no legroom. Some problems really are insoluble. ☹

---

**STEVE THEODORE** has been pushing pixels for more than a dozen years. His credits include *MECH COMMANDER*, *HALF-LIFE*, *TEAM FORTRESS*, *COUNTER-STRIKE*, and *HALO 3*. He's been a modeler, animator, and technical artist, as well as a frequent speaker at industry conferences. He's currently the technical art director at Seattle's *Undead Labs*.



# GETTING MORE OUT OF NUMBERS

## TECHNIQUES FOR MORE SOPHISTICATED AI

### WE HAVE LONG BEEN USED TO NUMBERS IN GAMES. THOUSANDS OF YEARS

ago, when people first started playing games, the simple act of keeping score was dealt with in terms of numbers. Even before games, when people had to simply scratch by for food, numbers were an integral part of their lives. From how many rabbits the hunter killed to how many sling stones he had left in his pouch, numbers have been a part of competitive activity for all of history.

Coming back to the present, numbers are everywhere for the game player. Some are concrete values that have analogs in real life: How much ammo do we have? How many resources will this take to build? What is the range of that weapon? Some are a little more nebulous, if not altogether contrived: What level am I? What condition are my gloves in? What armor bonus does this magic ring afford? And, although the medical community might be a bit startled by the simplicity, games often parade a single number in front of us to tell us how much “life” we have. (“He’s only *mostly* dead.”) Suffice to say that we have educated our gaming clientele to forgive this intentional shredding of the coveted suspension of disbelief. Even though some games have attempted to obscure this relationship by removing some of the observable references to this numeric fixation, gamers still recognize that the numbers are still there, churning away behind the wizard’s curtain.

### NUMBERS IN GAMES

» As programmers, our fixation with numbers is not coincidental. After all, along with logic, mathematics is the language of our medium. Computers excel in their capacity to crunch all these numbers. They’re in their element, so to speak. This capacity is a primary reason that the pen-and-paper RPGs of the 1970s and 1980s computers so energetically made the transition to the computer RPGs of the 1980s and beyond. Resolving combat in *ULTIMA* (1980) and *WIZARDRY* (1981) was far swifter than shuffling through charts, tables, and scribbles on scratch paper in *D&D*.

Numbers surround us, whether in overt use or under the hood. The interesting—and sometimes even disconcerting—thing is that they aren’t used more often. Even with all the advances and slick architectures that artificial intelligence programmers use, we too often fall back on the most simple of logical constructs to make our decision code. The most obvious example is the venerable if/then statement. Testing for the existence of a criterion is one of the core building blocks of computational logic. “If I see the player, attack the player.” Certainly, this sort of binary simplicity has its place. Left to itself, however, it can fall woefully short of even being adequate.

We could extend the above example by putting a number in the equation, such as, “If the player is < 30 [units of distance] away from me, attack him.” But really, what have we gained? We are still testing for the existence of a criterion, albeit one that is defined elsewhere in the statement or in the program. After all, “see the player” and “player < 30” are simply functions. There is still a razor edge separating the two potential states of “idle” and “attack.” All subtlety is lost.

So, how might we do things differently? One answer lies not in the existence of numbers in our world but what those numbers represent and, ultimately, how we use them.

### LOOKING INWARD

» Stop for a moment and do a self-inventory. Right now, as you sit reading this column, are you hungry? Sure, for the sake of simplicity, you may answer “yes” or “no.” But there is usually more to it than that. When my daughter was younger, she tended to cease using “hungry” when she was no longer empty. (This usually meant that she ate two or three bites only to come back wanting more about 20 minutes later.) I, on the other hand, could easily see myself defining “hungry” as “no longer full.” My wife has a penchant for answering somewhat cryptically, saying “I could be.” (This is usually less convenient than it sounds.)

All this makes sense to us on an intuitive level. “Hunger” is a continuum. We don’t just transition from “completely full” to “famished”—it is a gradient descent. What we do with that information may change, however, depending on where we are in that scale. For example, we can make judgment decisions, such as, “I’m not hungry enough to bother eating right now ... I want to finish writing this column.” We can also make comparative judgments, such as, “I’m a little hungry, but not as hungry as I am tired.” We can even go so far as use this information to make estimates on our future state: “I’m only a little hungry now, but if I don’t eat before I get on this flight, my abdomen will implode somewhere over Wyoming. Maybe I’d better grab a snack while I can.”

Compare this to how the AI for game characters is often written. The subtlety of the differences in value seems to be lost on them. Soldiers may only reload when their gun is completely *out* of ammunition, despite being in the proverbial calm before the storm. Sidekicks may elect to waste a large, valuable health kit on something that amounts to a cosmetically unfortunate skin abrasion. The coded rules that would guide the behaviors above are easy for us to infer:

```
if (MyAmmo == 0)
{
    Reload;
}
if (MyHealth < 100)
{
    UseHealthKit;
}
```

Certainly we could have changed the threshold for reloading to `MyAmmo <= 5`, but that only kicks the can down the road a bit. We could just as easily have found our agent in a situation where he had six remaining bullets in his magazine and, to co-opt a classic novel title, all was quiet on the western front. *Dude, seriously—you’re not doing anything else right now, might as well shove some bullets into that gun.* An agent built to only pay homage to the Single Guiding Rule of Reloading would still stubbornly wait until he had five before reaching for his ammo belt.

Additionally, there are other times when a rule like the above could backfire (so to speak) with the agent reloading too soon. If you are faced with one final enemy who needs one final shot to be dispatched, you don’t automatically reach for your reload when you have five bullets. You finish off your aggressor so as to get some peace and quiet for a change.

Needless to say, these are extraordinarily simplistic examples, and yet most of us have seen behavior similar to this in games. The fault doesn't rest in the *lack* of information—as discussed, often the information we need is already in the game engine. The problem is that AI developers don't leverage this information in creative ways that are more indicative of the way real people make decisions. Very rarely do we humans make a decision based solely on a single criteria. As reasonable facsimiles of the hypothetical *Homo economicus*, we are wired to compare and contrast the inputs from our environment in a complex dance of multivariate assessments leading us to conclusions and, ultimately, to the decisions we make. The trick, then, is to endow our AI creations with the ability to make these comparisons of relative merit on their own.

### LEVELING THE FIELD

» So, how do we do this? The first step is to homogenize our data in such a way as to make comparisons not only possible, but simple. Even when dealing with concrete numbers, it's difficult to align disparate scales.

Consider a sample agent that has maximums of 138 health points and 40 shots in a fully loaded gun. If at a particular moment he had 51 health and 23 bullets in the gun, we wouldn't necessarily know at first glance which of the two conditions is more dire. Most of us would instinctively convert this information to a percentage, even at a simple level. E.g., "He has less than half health but more than half ammo." Therein lays our first solution ... normalization of data.

My gentle readers should be familiar with the term normalization in principle, if not the exact usage in this case. Put simply, it is restating data as a percent—a value from 0 to 1. In the case above, our agent's health was 0.369 and his ammo status 0.575. Not only does viewing the data this way allow for more direct comparisons—e.g.,  $0.575 > 0.369$ —but it has built-in flexibility to handle changing conditions. If our agent levels up, for example, and now has 147 health points, we do not have to take this change into consideration as regards comparison formula. Our 51 health above is now 0.347 ( $51 \div 147$ ) rather than 0.369. We have detached the comparison code from any tweaking we do with the actual construction of the values themselves.

### BUT WHAT DOES IT MEAN?

» Normalization only sets the stage for the actual fun stuff. Simply comparing percentages between statuses like "health" and "ammo" usually isn't sufficient to determine the relative importance of their values. For example, I posit that being at 1 percent health is measurably more urgent than being down to 1 percent ammo. Enter the concept of *utility*.

Utility is generally a different measure than simple value. Value, such as our normalized examples above, expresses a concrete number. Utility, on the other hand, expresses a *concept*. In this case, the concept we are concerned with is "urgency." While it is related to the concrete values of health and ammo, urgency is its own animal.

The easiest way to do this is to create a "response curve." Think of passing the actual numbers through a filter of "what does this value mean to me?" That is what converting value to utility is like. This filter is usually some sort of formula that we use to massage the raw data. Unlike a lookup table of ranges (such as "ammo  $\leq 5$ "), we have the benefit of continuous conversion of data. We will see how this benefits us later.

The selection of the formula needs to take into consideration specific contour of the translation from value to utility. There are innumerable functions that we can use, but they are all built out of a few simple building blocks. Each of these blocks can be stretched and squished in and of themselves, and combining them together results in myriad combinations.

The first filter that we can run our numbers through is simply a linear conversion. (For these examples, I will use the standard  $x$  and  $y$  axis. I'll occasionally throw in an example of what they could represent.) Consider the formula:

$$y = 0.8x + .2$$

This results in a line running from our co-maximum values of  $(1.0, 1.0)$  which arrives at  $y = 0$  when  $x = .2$ . (See Figure 1.) Put another way, we want a steady descent in our utility ( $y$ ) at a somewhat quicker rate than the decent of the actual value ( $x$ ). We could have done something similar by changing the formula to:

$$y = 0.8x$$

As this point, the line extends from  $(1.0, 0.8)$  to  $(0, 0)$ .

Obviously, changing the slope of the line—in this case, 0.8—would change the rate that the utility changes along with the value ( $x$ ). If we were to change it to 1.2, for example, the rate of descent would increase significantly. (See Figure 2.)

$$y = 1.2x - .2$$

It's worth noting here that these formulas are best served by being combined with a clamping function that ensures  $0.0 \leq y \leq 1.0$ . When we take that into consideration, we have another feature to identify here: when  $x < 0.2$ ,  $y$  is always equal to 0.

On the other hand, consider the similar formula:

$$y = 1.2x$$

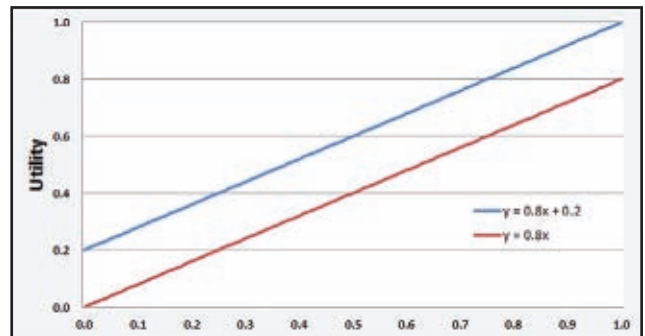


FIGURE 1

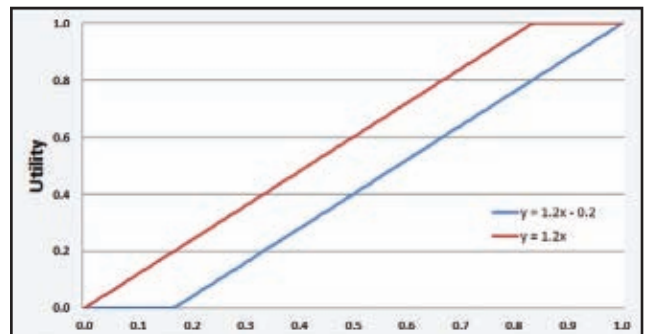


FIGURE 2



This exhibits the same effect with the exception that now the "no effect" zone is when  $x > 0.8$ . That is, the utility doesn't start changing until our value  $< 0.8$ .

These effects are useful for expressing situations where we simply do not care about changes in the utility at that point.

**ENTER THE EXPONENT**

» The simple formulas above merely set the stage for more advanced manipulations. For example, imagine a scenario where the meaning of something starts out as "no big deal," yet becomes important *at an increasing rate*. The state of the ammo in our gun that we illustrated above makes an excellent example. In this case, the value is simply the number of shots remaining, whereas the utility value is our urgency to reload.

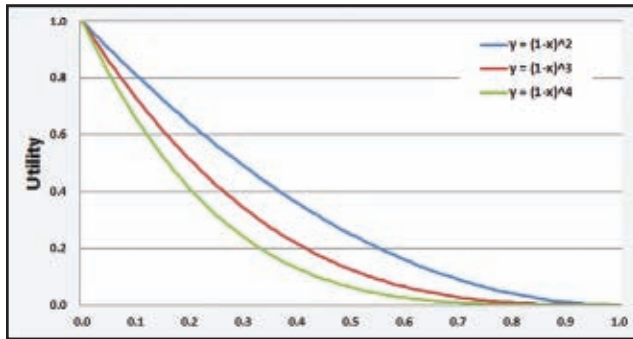


FIGURE 3

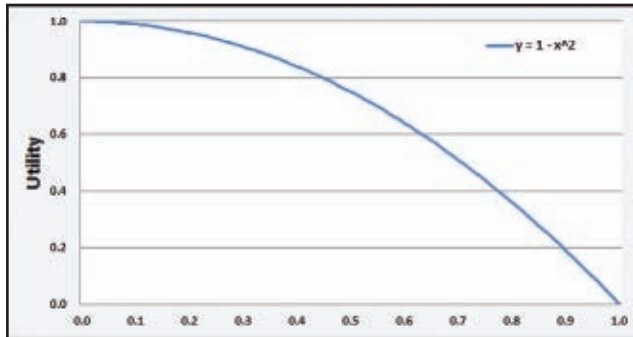


FIGURE 4

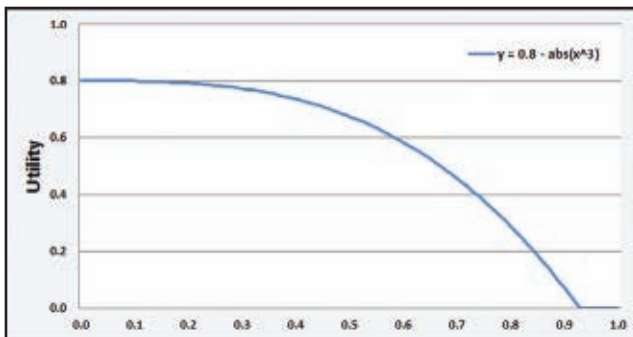


FIGURE 5

Analyzing this *outside* the realm of the math—that is, how would we behave—gives us clues as to how we should approach this. Imagine that our gun is full (let's assume 40 shots for convenience), and we fire a *single* shot. All other things being equal, we aren't likely to get too twitchy about reloading. However, firing the *last* shot in our gun is pretty alarming. After all, even having one shot left was a heckuva lot better than having none at all. At this point, it is helpful to start from those two endpoints and move toward the center. How would we feel about having 35 shots compared to 5? Thirty compared to 10? Eventually, we will start to see that we only really become concerned with reloading when our ammo drops down to around 20 shots—and below that, things get urgent very quickly!

In a simple manner, this can be represented by the following formula:

$$y = (x - 1)^2$$

As we use up the ammo in our gun ( $x$ ), there is still an increase in the utility of reloading, but the *rate* that the utility increases is accelerating. [See Figure 3.] This is even more apparent when we change the exponent to higher values such as 3 or 4. This serves to deepen the curve significantly. Note that a version of the formula with odd exponents would require an absolute value function so as to avoid negative values.

Another quick note about manipulating these formulas. We could turn the above curves "upside down" by arranging it as follows:

$$y = 1 - x^2$$

Looking at the chart in Figure 4 shows that this version provides a significantly different behavior—for example, an agent who has a very low tolerance for having an empty gun!

By manipulating how the function is arranged, we can achieve many different arrangements to suit our needs. We can shift the function on either axis much as we did the linear equations above, for example. [See Figure 5.]

We can specify where we want the maximum utility to occur—it doesn't have to be at either end of the scale. For example, we might want to express a utility for the optimal distance for an agent to stand away from an enemy based on our weapon choice. [See Figure 6.]

$$y = 2(1 - |x - 0.3|)^2$$

**SOFT THRESHOLDS**

» While we can certainly get a lot of mileage out of simple linear and exponential equations, one final class of formulas is very useful. Sigmoid functions, particularly the logistic function, can be used to define "soft thresholds" between values. In fact, logistic functions are often used as activation functions in neural networks. Their use here, however, is much less esoteric.

The base logistic function is:

$$y = \frac{1}{1 + e^{-x}}$$

While the base of the natural logarithm,  $e$ , is conspicuous in the denominator of the fraction, it is really optional. We can certainly use the approximation of 2.718 in that space, or any other number for that matter. In fact, by changing the value for  $e$ , we can achieve a variety of different slopes to the center portion of the resulting curve. As stated though, the formula graphs out as shown in Figure 7.

Notice that, unfortunately, the graph's natural range is not 0–1 as with our other examples. In fact, the range of the graph is infinite in that it asymptotically approaches both  $y = 0$  and  $y = 1$ . We can apply some shifting to get it to fit the 0–1 range though, so that we can use it with normalized values of  $x$ . We can also change the area of the graph where the threshold occurs by changing what we are adding to the exponent.

$$y = \frac{1}{1 + e^{-(10x-5)}}$$

### COMPARING AND CONTRASTING

» The end result of all of this is that we can create very sophisticated response curves that translate our raw values into meaningful utility values. Also, because these end products are normalized, we can now easily compare and contrast them with other results. Going back to the examples I cited early on, we can decide how hungry we are in relation to other feelings such as fatigue (or too busy finishing a last-minute column for a magazine). In fact, we can line up dozens—or even *hundreds*—of utilities for various feelings or potential actions and select from among them using techniques as simple as “pick the highest” to seeding weight-based randoms.

Compare this to what we would have to do were we not to use the normalized utility values. In our hungry/tired/busy example, we normally would have to construct a multi-part condition to define each portion of our decision. For example:

```
If ( (Hungry > 5) && (Tired < 3) && (Busy < 7) ) then
{
    Eat();
}
If ( (Hungry < 4) && (Tired > 6) && (Busy < 3) ) then
{
    Sleep();
}
If ( ...
```

Ya know what? Never mind ...

Even if the above values were normalized (i.e. between 0 and 1), the complexity explosion involved with comparing the different possible ranges and mapping them to the appropriate outcome would get out of hand quickly. And that's just with 3 inputs and 3 outputs! By converting from value to utility, we massage what the data “means to us” inside each response curve. We now can feel comfortable that a direct comparison of the utilities will yield which item is truly the most important to us.

The system is extensible to grand lengths as well. If we want to include a new piece of information or a new action to take into account, we simply need to add it to a list. Because all the potential actions are scored and sorted by their relative benefit, we will automatically take newcomers into stride without much (if any) adjustment to the existing items.

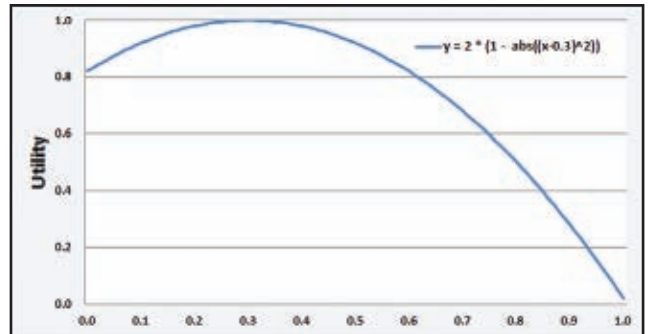


FIGURE 6

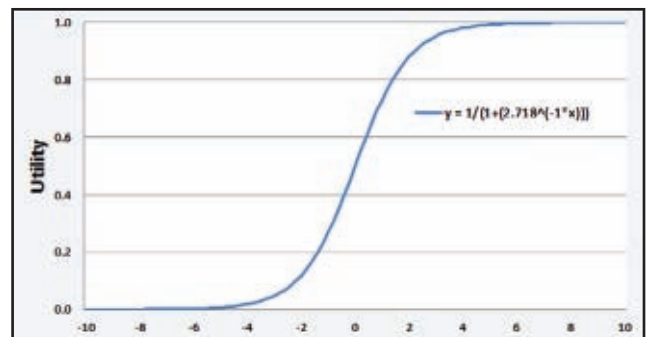


FIGURE 7

If calculating and measuring all of these feelings and desires is starting to sound a lot like *THE SIMS*, it is not a coincidence. *THE SIMS* is an excellent (but not the only) example of how complex utility-based functions can be used to simulate fairly reasonable, context-dependent decision processes in agents. Richard Evans has spoken numerous times at GDC on this very subject. I wholeheartedly recommend reading his papers and viewing his slides on the subject.

The uses of these methods aren't limited to that genre, of course. Strategy games, in particular, lend themselves to more nuanced calculation. Even in modern shooters and RPGs, agents are expected to make increasingly more believable decisions in environments that contain significantly more information. Our AI no longer has the luxury of simply leaning on “if I see the player, shoot him!” as its sole guideline, and building static rulesets that address all the possible permutations of world state gets brittle at an exponential pace.

As I've illustrated (ever so briefly), the inclusion of some very simple techniques lets us step away from these complicated, often contrived, and sometimes even contradictory rulesets. It also allows us, as AI designers, to think in familiar terms of “how much”—the same terms that we often use when we think of our own (human) states. The numbers we need are there already. The magic is in how we use them. ☺

**DAVE MARK** is the president and lead designer of *Intrinsic Algorithm*, an independent game studio and AI consulting company. He is also the author of *Behavioral Mathematics for Game AI* and a summit advisor of the annual *GDC AI Summit*. He continues to attend the *University of Life*. He has no plans to graduate anytime soon.



# THE MANY FORMS OF GAME NARRATIVE

## USING STORY AS A DESIGN TOOL

### UNLIKE WHAT VIDEO GAME

detractors might tell you, story and narrative are a huge part of the art and science of building interactive entertainment. Nearly all games have some level of storytelling in them. What is more fascinating to me as a designer is how wildly different the usage of it can be from genre to genre, and even within that genre from game to game.

This is, of course, quite different from most other media—most fiction, be it murder mysteries, cop shows, blockbuster movies, or even Saturday morning cartoons, are deeply and intuitively narrative driven (whether it's of high quality is another story). Not so with video games. Nearly every game leverages narrative in a way, to a different degree, to different results.

Some games (such as those from BioWare, my studio) make story central, whereas others use it as more of a backdrop. Most triple-A titles opt for a middle path—having a simple (but sometimes powerful) story that creates a sense of place and purpose. These designs never forget that gameplay is king, and story should only be pushed far enough to support those ends.

Story is perhaps the most flexible tool in the designer's toolbox, and as such, use of story in games can take wildly divergent approaches. This is one of the reasons that making rules about narrative in games can be so difficult—the approach and focus given to story is going to wildly adjust how the designer needs to approach it. Is the story a backdrop to the action, or is it core to the player's activity? Can the player adjust the flow of the story, or is he merely along for the ride? Does your design require that the player pay

attention to the story, or is story just there for color? All these things are central to how the story, and the player's interactions with the story, must be constructed.

This is all complicated by the fact that telling stories in games is hard for a lot of reasons. Designers don't have control over the flow or cadence of the experience. Games are long—so long that it can be hard for players to keep track of the narrative, especially if they walk away from the experience for a while. And despite the fact that players always claim to want more and better stories, un-artfully trying to cram narrative down their throats is more likely to bore or confuse them. Care needs to be taken to present the story in a manner and pace conducive to the rest of their game environment. What that manner is will vary wildly based on the game you're trying to make.

### NARRATIVE AS BACKDROP

» Most games have at least the minimum level of story, which is the narrative backdrop. For some games, especially multiplayer games, this might be all the story that a player encounters. The depth of these backdrops varies from game to game, and often only serve to give the game some grounding. Board games like *Dominion* and *Settlers of Catan* both have backstories in the rules that are only a couple of paragraphs long—these are meant to ground the player experience, and players will rarely give them a second thought.

Backdrops can be much deeper and more involved, though. The MMO *SHADOWBANE* had a backstory of astonishing depth. Reams and reams of lore, covering every available aspect of the fantasy world

were on the web site for players to discover. Little of it was to be found in-game, other than the handful of players who used this lore to roleplay—most players appreciated the depth, but were more interested in the game's siege mechanics. Using a backstory-centric approach to narrative is common in MMOs, as most MMOs are not well suited to presenting a sense of change and history, especially in a world with a few thousand protagonists all trying to change it.

The deepest backstories in games come from licensed ones. *ARKHAM ASYLUM* and *FORCE: UNLEASHED* have decades of material to draw from and, in both cases, designers worked hard to ensure that they drew on commonly recognized elements recognized by casual fans as well as Easter eggs to cater to the hardcore. In a licensed project, perhaps no application of story is more important, and when reviewers state *ARKHAM ASYLUM* really brings Batman's world to life, or that *FORCE UNLEASHED* captures the far reaches of the *Star Wars* universe, you have a good understanding why these games succeeded.

### NARRATIVE AS PLOT PROPELLANT

» Most games described as having great stories (such as *STARCRRAFT*, *HALF-LIFE*, and *UNCHARTED 2*) go beyond a backdrop and instead, provide a narrative for which the player is expected to play the role of the protagonist. These stories provide a framework for the missions of the game and an explanation for why the player is doing what he's doing, why he's visiting the locales he's visiting, and an explanation for why things are getting progressively harder and more urgent.

In this regard, stories also provide continual rewards for the player. Completing a map in *PSYCHONAUTS* or *UNCHARTED 2* gives the player a cutscene that progresses the narrative, but with charming characters and witty dialogue that makes the player want to progress to the next cutscene and get the next narrative *Scooby Snack*. Stories done in this manner with this goal can have well fleshed out characters who experience growth, use clever banter, and provide a sense of mystery and the occasional twist. Or the narrative can be whittled down to nothing but the reward, as it is in *ROCK BAND*: you've been rocking the Casbah, so now you get a private jet.

The narrative as a propellant concept is not just for the triple-A games. *DINER DASH* similarly does this, and *PUZZLE QUEST*'s narrative and RPG is a core reason why that game differentiated itself from other match-3 puzzlers. Story is also a key design focus of the Facebook game *FRONTIERVILLE*, and one of the reasons the game separated itself from the pack. Expect story to hit Facebook game design in a big way over the next year.

### NARRATIVE TO BUILD EMOTIONAL CONNECTION

» *BALDUR'S GATE* was released more than a decade ago, so you can forgive most players if they forget the finer details of the story. However, few players who played *BALDUR'S GATE* would soon forget Minsc and his space hamster, Boo. In not just games but TV and movies as well, the characters navigating the narrative are at least as important, if not more important than the twists and turns of the story itself.

HALF-LIFE provides a narrative for which the player is expected to play the protagonist.



In most video games, the main character is designed to be somewhat of a blank slate, so that the player can more readily see himself in that role, taking part in the narrative. This raises the importance of the auxiliary characters: Minsc and Boo, or Chloe in UNCHARTED 2, or even the squadmates in BAD COMPANY and the little sisters in BIOSHOCK. These characters and companions add personality and warmth to the game experience, give emotional cues as to how the player should feel, and perhaps most significantly, once you like them, give the designer something in the narrative that the player has an emotional investment in.

Most screenwriters accept as dogma that audiences are more biased toward saving a loved one than a world—in their hearts, they would choose saving Lois Lane over a nameless, faceless Metropolis every time. This is a triumph of heart over mind that audiences want in our escapist fantasy, but it is also something that can be manipulated by the designer in order to increase personal investment into the game. Rescuing the princess is one thing—rescuing your love interest is quite another. And spoilers alert, but would anyone be talking about the story of FINAL FANTASY VII if it weren't for the death of Aeris? Or talking about

the story in STARCRAFT if not for the capture of Kerrigan?

#### NARRATIVE AS EXPLANATION


» It turns out that the world of MIGHT AND MAGIC is not a fantasy world. Not to spoil the story for those who were really hoping to go back and play DARK SIDE OF XEEN someday, but according to the first MIGHT AND MAGIC, these worlds were all giant terrariums hurtling through the void for some cosmic purpose.

A nifty upside of this science-fiction backstory was that it allowed for robot bosses and laser rifles at the endgame, but the real reason was much simpler: the tech for the first

MIGHT AND MAGIC was grid-based at the core, and designer Jon Van Canegham wanted an explanation as to why the world was perfectly square and had an impassable barrier on all sides. In this instance, he saw story as design spackle, suited for covering up game elements that were nonsensical and inconvenient.

Story used in this fashion is an incredibly useful design tool, but the designer should be wary about going too far and instead drawing attention to what you had hoped to hide. When KNIGHTS OF THE OLD REPUBLIC was in the early design stages, they recognized early on that the walk back to your ship was killing their game flow. Their first pass design to solve the problem called for a "call speeder" button that summoned a speeder bike to where the player was, animated the player getting on it, and then teleported the player to his starship after a fade-to-black. The feature raised more questions than it answered—why isn't a player riding a speeder all the time? Where did it come from? Why can it be summoned anywhere? The designers killed the long, convoluted idea and changed the button to a simpler "return to ship." By hand-waving away the explanation, the designers drew less attention to what was an important but incongruent feature.

#### NARRATIVE AS PLAYER AGENCY

» Last, but not least, we have the game's narrative acting as the primary means of a player's agency—the player makes choices within the game that dramatically affect the outcome of the game's narrative. This is, of course, the bread and butter of the game design behind BioWare games like DRAGON AGE and MASS EFFECT. The intricacies of designing these are fascinating, and well worthy of their own discussion—so we'll talk about them next month. 

---

**DAMIAN SCHUBERT** is the lead systems designer of STAR WARS: THE OLD REPUBLIC at BioWare Austin. He has spent nearly a decade working on the design of games, with experience on MERIDIAN59 and SHADOWBANE as well as other virtual worlds. Damian also is responsible for Zen of Design, a blog devoted to game design issues.

## Interview: Meggan Scavio

MEGGAN SCAVIO, DIRECTOR OF THE GAME DEVELOPERS CONFERENCE, DISCUSSES THE UPCOMING 25TH EDITION, THE VERY FIRST GDC, AND THE SECRET TO HAVING YOUR TALK ACCEPTED

**Q: Meggan, this is a very special year for the Game Developers Conference—it's the 25th edition, correct?**

**Meggan Scavio:** It is! It started in 1988, and there were two that year. The first was in the living room of Chris Crawford, GDC's founder. Because this is a very special year for us, I want to honor Chris. So I think we're going to put him on a panel with others who have been in the industry for 25 years and have been making games for that long. We're still working on those names. But did you know that there are only three people who have been to every single GDC and have been active developers all that time? They are Chris; Gordon Walton, who is VP at BioWare Austin; and Tim Brengle, who is now an adjunct instructor at DeVry University and who has run GDC's conference associate program for as long as I can remember.

**Q: What continues to make GDC such an important event for developers?**

**MS:** Someone once called GDC a family reunion, and to me it's always been that, especially back in the San Jose days when we were a lot smaller. That's when you could walk down the hall and see everyone you knew. It's a little tougher now because there are so many more people ... but it is that one time of the year when, as a developer, you leave the confines of your office and you join like-minded people to share information and ideas and talk about what you're doing. We say that GDC is where you learn, network, and become inspired.

**Q: What special things can attendees expect when they come to the show this February?**


**MS:** This year there will be more focus on Smartphones, and so

we've consolidated our iPhone Summit and Mobile Summit into a Smartphone Summit. At the same time, we're rebranding our Serious Games Summit and dividing it into two days—the first will be about serious games in the health field; the second is all about "gamification," a trendy word that means using gameplay mechanics for non-game consumer technology applications.

**Q: What were some of the biggest announcements at GDC over the past 25 years?**

**MS:** Well, Bill Gates announced the Xbox at GDC in 2000, and Will Wright announced SPORE at GDC 2005. Because GDC is for developers, our show is where cutting-edge game technologies are unveiled. What you see at GDC is what the public is going to see in 3–5 years at E3. E3 is now; GDC is what's going to happen.

**Q: For companies looking to speak at GDC, what's the secret to getting your talk accepted?**

**MS:** GDC is all about integrity. We have a very strict advisory board which, when it is reviewing talks, if it senses any sort of a sales pitch, the talk gets declined. So my suggestion to vendors who want their products mentioned at GDC is to ask their customers to submit a talk that includes the technology and how they use it. It's the user who should be talking about it, not the vendor. That's because we want to hear the bad along with the good; we want to know what problems the user had and how they fixed them. And you're not going to hear that from the vendor. So there has to be takeaway ... that's what we look for when we're reviewing talks—takeaway, takeaway, takeaway. Is the attendee going to walk out of that room knowing something they didn't know when they walked in? 

## Interview: Brian Reynolds

BRIAN REYNOLDS, CHIEF GAME DESIGNER AT ZYNGA, TALKS DESIGNING INNOVATE GAMEPLAY, AND 3D IN SOCIAL GAMING

**Q: Brian, your keynote address at GDC Online looked at the launch of FRONTIERVILLE and the process of designing innovative gameplay.**

**For readers who may not have attended, what was the main takeaway?**

**Brian Reynolds:** I talked about the approach we took to pitching FRONTIERVILLE within Zynga, and then the process we used to design it. We had a fairly conservative pitch (FARMVILLE plus more game) and set out to take the best from FARMVILLE and the best from MAFIA WARS and put

them together on the frontier. Maybe the most interesting part was showing how many of our original pitch slides made it into the game, but then noticing that largely those original ideas weren't the things that people now identify as what makes FRONTIERVILLE a great social game. The real innovations in FRONTIERVILLE weren't in the pitch; they were things we found along the way, things we found by playing and improving and scrapping. So it was a talk about "here's a good process to design innovative gameplay."


**Q: Who or what inspired you to be a game designer? And what games—past or present—do you consider excellent from a game-design perspective?**

**BR:** I sold my first game when I was 13 years old—for one hundred dollars [gives Dr. Evil hand gesture]. So I've always been interested in game design and programming. The first game that made me want to do this for a living was probably ULTIMA VI. I remember thinking "Wow, this is a really big and open world" and, at the same time,

"Hey, I could totally have written this!" Some of the other games that most appealed to me as both a player and a designer over the years include CIVILIZATION [amazing open world plus lots of simple mechanics that interact in complex ways], STARCRAFT [great combo of sharp asymmetry and near-perfect game balance], HALF LIFE 2 [my favorite story ever combined with terrific game detail and balance], GEARS OF WAR [the most fun pure shooting experience and such innovative game craft], and BIOSHOCK

[great passive-listening story-telling technique].

**Q: What's your take on 3D social games? What are some of the ways you would use that technology to engage players?**

**BR:** I think it's still a bit early for 3D in the social space. Navigating in 3D makes for difficult UI decisions that are hard to teach to the mass market, and the quality of the pure graphics isn't high enough yet for the mass market to get attracted to the potential immersiveness of it. So I don't see it happening yet. 





# A SINGULAR JOURNEY

## SHOULD I BECOME AN INDEPENDENT GAME DEVELOPER?

**GIVEN THE EXPLOSION OF** successful independent games like *WORLD OF GOO*, *BRAID*, and more recently *MINECRAFT*, I'm sure there are some professional game developers out there, sitting in their cubicles, wondering if they too should attempt to go indie. I've been on the independent game development team *Wolfire Games* for two years now, and because this is one of the questions we are emailed most frequently, I've wrestled with this topic a bit in my individual responses. This is my first stab at really compiling my thoughts in one place.

### THINKING OF FUN GAME IDEAS

» If you're going to go indie, you have to start with a fun idea—but don't overdo this initial academic thinking phase by writing out pages and pages of design documents. The truth is that everyone can think of fun game ideas, and when you go indie, your real problem won't be generating fun ideas, but implementing them. Thomas Edison said that "genius is 1 percent inspiration and 99 percent perspiration." For games this is especially true, because you often can't objectively determine how fun something is until you build it. I'm not sure how the big companies do it, but indie game development is often a messy iterative process that requires you to sculpt

as you go. Even Will Wright, one of the greatest game designers of all time, has used the phrase "accidents that worked" to describe parts of his development process.

### TESTING THE WATER

» To go indie, you have to anticipate the predicament of having no brand, no community, no budget, and a large risk of no sales. The good news is all these factors can be mitigated by fast prototyping. Build something fun as fast as you can and release it into the wild. This will show you quickly whether you are headed in the right direction. If the public reacts positively to your work, you should consider iterating on your idea. Several indie games like *CORTEX COMMAND*, *NATURAL SELECTION 2*, and especially *MINECRAFT* have been able to raise awareness, goodwill, and decent money for themselves in early development by following this model. If it turns out gamers aren't interested in your released prototype, you'll be able to scrap your idea and try something new before wasting a lot of time and money.

### PLANNING YOUR DEVELOPMENT CYCLE

» Game development adheres to Hofstadter's law: "It always takes longer than you expect, even when you take Hofstadter's Law into account." As such, you need to have a plan

that gives you more than enough time to finish your game. Are you prepared to eat ramen for an extended period of time, live with your parents, or get a part-time job flipping burgers to see your game through to the end? Or maybe you've got some severance from your big studio job. If so, you probably have just enough of a runway to complete your game. On the other side of the equation, narrow the scope of your game whenever possible. Every motivated developer feels compelled to add more and more to his game but you must fight the urge to bite off more than you can chew. The dreaded feature creep is even more damaging when you go indie.

### BEING YOUR OWN BOSS

» It's great to have creative control over your project. However, when no one is looking over your shoulder and scheduling milestones and Scrum meetings for you, you'd better be able to manage your own time. Having hard-working team mates can foster an ecosystem of high motivation—so can the stress of a continually shrinking bank account. A more unconventional tactic we've been trying at *Wolfire* has been to openly share our development process with the public from day one. This has enabled the world to look over our shoulder as we craft *OVERGROWTH*, essentially making the community our boss. There are lots of ways

to approach the motivation issue for an indie team, but the big thing to keep in mind is that making your own game should be fun. If this is not the case, you might as well have kept your corporate job.

### WEARING MANY HATS

» If you've only worked on games at big companies, you've probably ended up specializing in just a few areas. Indie developers on the other hand, have to sweep through a full gamut of roles on a daily basis (programmer, artist, game designer, story writer, Foley technician, composer, marketer, community manager, et cetera.). If you don't have all the necessary hats for your project in your wardrobe already, you'll need a plan. You either have to be ready to learn new skills, or assemble a team that can fill in the gaps for you. Some people hold onto the notion that they can just be game designers and come up with ideas that others will magically want to implement. Unless you can afford to pay a team to work on your idea or you have already proven yourself to be a game design rock star, best of luck to you. Some hats are more important than others (e.g. programming and making art) but it's best to be competent in several areas so if all else fails you can singlehandedly get your project sailing. As it becomes clear you are headed somewhere fun, that's when others

will magically appear to join you on your development journey.

### DON'T BE A F\*\*\*ING PUSSY!

» At *Fantastic Fest*, I had the good fortune to hear *Jon Blow* explain how people's assessment of risk is not always kept in perspective. If you're even considering becoming an independent game developer, you're probably not in one of the many parts of the world where you have to worry about getting shot or bitten by a snake every time you walk outside. How frightening can it really be to risk not making very much money for a year or two (or three)? As *Ron Carmel* once wisely told us, when you're 80 years old on your death bed, you won't think to yourself, "Man I wish I had never tried to make that game."

### KNOW THYSELF

» My hope is that this article will help readers who may be on the fence to think critically and honestly about how they would fare as independent developers. If you've contemplated all these points and deep down you can see yourself having the motivation, skills, discipline, and resources to make a game on your own, then going indie might definitely be the right choice for you. ☺

*JOHN GRAHAM* is the COO (Coffee Operations Officer) at *Wolfire Games* and has been known to grow beards.



# LET'S DO THE TIME WARP AGAIN



## REVISITING THE CHIPTUNE

**EVERYTHING OLD IS NEW AGAIN AND**, from MEGA MAN 9 and 10 to SCOTT PILGRIM VS. THE WORLD, demakes are the new remakes. The 8-bit canvas and the chiptune soundtrack are going through a nostalgic renaissance, where what once was technical limitation is now quirky fun with a retro flair. For many of the audio professionals working on these titles, this means a return to audio tools that they grew up listening to, tools that were out of date and passé by the time they cut their first loops.

### NES-ESSITIES

» Without a doubt, the largest number of demakes are done in the seminal style of the original Nintendo Entertainment System (NES). Thankfully for those new to chiptune music and sound, the NES and Game Boy have been a favorite of remix artists and electronic acts for years, so a wide variety of options exist when it comes to emulating the sounds of these classic systems.

The first step to understanding chiptune music is to recognize the classic palette that gives it the unique sound a generation of gamers consider the soundtrack to their childhoods. The original NES console had the ability to generate four types of simultaneous audio: two pulse waves, one triangle wave, and a white noise generator. The pulse waves were capable of pitch bend, 16 different volume settings, and a frequency range of essentially A1 to F7. The triangle wave was a fixed volume and had a range essentially from A0 to well beyond the C8 range of a MIDI keyboard. The white noise generator had 16 different volume

settings and was the source for all percussion. Complicating matters further, each of these four voices was monophonic.

More than 25 years on from the NES's debut, there are a vast number of ways to achieve the same sound palette synonymous with classic gaming. These options basically break down into two categories: software and hardware. Perhaps the most common software solution for chiptune music is Propellerhead's Reason. Whether working within Reason's own sequencer or via Rewire into your DAW of choice, Reason's Subtractor Polyphonic Synthesizer is an excellent solution for the pulse wave, triangle wave, and white noise source sounds, and is one of the most popular go-to programs among chiptune enthusiasts. Working with Reason or a similar software-based monophonic synth within a DAW environment means that the technical limitation of only four available voices is removed. Composers and sound designers will need to decide how authentic they want to be and whether they want to stick to the four-voice limit imposed by the old hardware.

Aside from the soft synths common to pro audio production, there are also a number of VST plugins dedicated entirely to chiptune audio production. Tweakbench's peach is a sample-based NES chiptune tool and one of the most highly regarded for its authentic sound. Additionally, Tweakbench also offers triforme, a soft synth-based tool, and toad, which is an NES-inspired drum kit. All are free downloads.

Once you have the palette, you'll need to understand some of the

compositional tricks of the trade. Thankfully, the Internet is full of chiptune tutorials, and there are some great instructional videos on YouTube. In particular, check out the "Fake N Bake" chiptune series by Judson "Tettix" Cowan ([www.youtube.com/user/cicadacom](http://www.youtube.com/user/cicadacom)) or Michael "Skitch" Schiciano's "Famicom Friday" series ([www.youtube.com/user/skitchmusic](http://www.youtube.com/user/skitchmusic)).

For those looking for a hardcore level of authenticity, there are also hardware solutions for NES chiptune music. The MidiNES is one of those ubiquitous gray NES cartridges with a MIDI cable connected directly to the circuit board. By inserting the cartridge into an original NES console, it allows for direct MIDI control of the NES hardware. Nanoloop is a German homebrew creation that turns the classic Game Boy into a step sequencer via a specific Nanoloop cartridge. iPhone, iPad, and iPod touch users may be interested in checking out Nanoloop for the iOS.

### EXTRAVAGAN-C64

» The NES isn't the only game in town though, and chiptune emulation tools exist for a number of other classic platforms. Some of the most popular are those centered on creating music akin to that found on the Commodore 64. basic64 by British software developer de la Mancha is a VST plugin based on the SID chip from the Commodore 64. Another well-regarded C64 option, Odosynth's Unknown 64, is currently unavailable as the Odosynth web site claims to be on a brief hiatus.

For composers looking to emulate the 16-bit sounds of the Super Nintendo Entertainment

System (or SNES), a fan-driven community of SoundFonts exists. These SoundFonts have been sampled directly from classic SNES games, and each bank tends to be named after the originally sampled source game. Composers and sound designers looking for SNES SoundFonts can find them with names like CHRONO TRIGGER, LEGEND OF ZELDA: A LINK TO THE PAST, SECRET OF MANA, or F-ZERO. Each of these SoundFonts will open directly within Kontakt as would any other Kontakt instrument.

The technological options available to composers and sound designers today have made achieving a retro sound a fast and easy process. What now takes only minutes to set up and program can help to create a flood of nostalgia that warps your listeners backwards in time to the blips and bleeps of simpler days. 🎮

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



## touched in the head

### IAN ADAMS MOVES FROM PC CASUAL DEVELOPER GAMEHOUSE TO IOS GAME MAKER Z2LIVE

When GameHouse laid off most of its development staff, Ian Adams was looking for his next gig. With experience in QA and design, he wasn't sure which way to push. The answer turned out to be the company that offered the greatest opportunity in the most intimate setting—bite-sized iOS developer Z2Live.

**Brandon Sheffield:** You moved from a medium-sized company to a much smaller one—how different was the hiring and interviewing process?

**Ian Adams:** The interview with Z2Live was really interesting. There was a little of the normal conversational interview, but the bulk of it was doing actual work: playing a game, and writing bugs. It was honestly kind of refreshing to have an interview that was less focused on my ability to interview well, and far more on my actual ability to do the job.



**BS:** Do you find personality and company culture to be a bigger part of hiring in this scenario?

**IA:** Company culture was very important to me. I was most recently at a small studio that was bought up by a much larger corporation. When I left my last job, one of the first questions I asked myself was what kind of place I wanted to work next. I decided I wanted to get back to a small, independent studio. I wanted to get back to a company where you can fit every employee in one room, and you actually know who everyone is and what they do. Landing at a small iOS publisher like Z2Live is perfect. It's nice to have everyone feel close to the product and be invested.

**BS:** You were previously in a more hierarchical structure—how do you find things are different at your new company?

**IA:** The hierarchy in QA is entirely flat. I'm lucky that in my previous job we would take turns being the Lead, so I'm equally used to being in charge and being directed. We have all our testers in one room, so rather than getting assignments top down, we just discuss what needs to be done, and each of us volunteers to take on a task. At my previous company, a lot of the time test cases were essentially created as a way of proving to producers that QA was doing something. More of a paper trail than a tool. Conversely, we're getting ready to ship DOGFIGHT: ONLINE, and while we're still creating test cases, we're creating them as we need them, making sure the testing is thorough instead of filling out spread sheets for the sake of having a lot of spread sheets filled out.

**BS:** How would you say QA changes when you move from a more traditional platform to a touch or gesture-based one?

**IA:** The biggest shift is that we're working on a very limited range of hardware. I've gone from a huge compatibility rack with dozens of permutations on OS versions, video cards, sound cards, resolutions, and on and on, to just working with iOS. There are 9 major permutations of hardware. We support two versions of the OS. Done.

Actually, the biggest challenge in terms of the technical side of testing is that all the games have an online component. With TRADE NATIONS, I'm testing the client updates, but I also have to make sure that the hundreds of thousands of people playing on the old client aren't suddenly cut off when we update the server. There are a lot more moving parts to keep track of.

**BS:** If someone feels stuck in QA, how would you say a lateral move to a smaller company would affect their career?

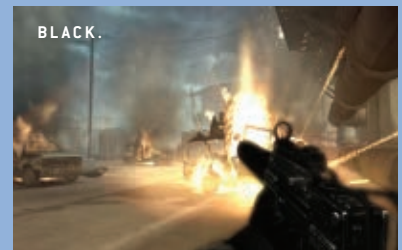
**IA:** While I'd say mobility improves at a small company, I don't think it's a sure thing. While a small company is more intimate, and creates more opportunities for you to stretch into different roles, and demonstrate your skill and versatility, there's a possibility that a small enough company just doesn't HAVE the position you're qualified to move into. Of course, that can be generalized to any company: There will be more opportunities at a company that is doing well than one that is struggling.

## new studios

Hamburg-headquartered free-to-play online game company Bigpoint has opened a new branch in Sao Paulo, Brazil, its fifth office and its first location in South America.

Viximo, a Cambridge, MA-based company that specializes in distributing social games beyond Facebook, has opened a new office in San Francisco and hired game industry veteran Sutton Trout to its management team.

## who went where



After departing from Codemasters earlier this year, game industry veteran Stuart Black, senior designer on the 2006 Criterion shooter BLACK, will head up a new London-based in-house development studio at City Interactive.

Long-serving general manager of Disney Interactive Studios Graham Hopper will be leaving the company as it continues to shift focus away from boxed products and toward digital releases.

Cloud-based game streaming service Gaikai has found a new chief strategy officer in former Jamdat Mobile and Electronic Arts exec Nanea Reeves.

Ed Bartlett, co-founder of in-game ad firm IGA Worldwide announced he has left the company to pursue "new challenges in gaming and beyond."

Former Disney Online exec Arthur Houtman will take the role of CEO at Vanguard Entertainment Group, placing him at the reins of the new company formed by Guerilla Games' original founders Martin de Ronde and Michiel Mol.

Tecmo Koei co-founder Yoichi Erikawa will take over as CEO and president of the company following the resignation of Kenji Matsubara.





# shadows of abigail

## FINDING EMERGENT FUN

AS PART OF THE UNIVERSITY OF CENTRAL FLORIDA IN ORLANDO, THE FLORIDA INTERACTIVE ENTERTAINMENT ACADEMY PROVIDES AN INTENSIVE TRAINING PROGRAM THAT AIMS TO REFLECT THE REALITIES OF COMMERCIAL GAME DEVELOPMENT. SHADOWS OF ABIGAIL IS A TEAM EFFORT FROM STUDENTS AT THE FIEA AND WE SPOKE WITH LEAD DESIGNER STEVE JULSON AND PROJECT LEAD/GAME PLAY PROGRAMMER JOSÉ LUIS LÓPEZ ZURITA TO FIND OUT HOW THE 3D PLATFORMER CAME TOGETHER IN ONLY SIX MONTHS.

**Jeffrey Fleming:** *What was your prototyping process like?*

**Steve Julson:** We first prototyped with the Unreal Development Kit to determine what types of platforming would fit the best with our shooting mechanic. It was after this that we realized how beneficial LITTLE BIG PLANET's level editor was for our prototyping. Once we discovered the METAL GEAR SOLID mod for LBP, which provided the player with a physics-based projectile weapon that would influence other physics-based objects, we immediately went to work putting all of our paper designs to test.

At FIEA, the importance of prototyping was stressed in order to successfully understand what specific mechanics needed to be hashed out and to know what was fun, intuitive, and potentially emergent. As lead designer, it was beneficial to make a convincing prototype in LBP because the other designers could rally around it and feel that we had a serious trajectory for our concepts.

**JF:** *I like the idea of finding emergent play during prototyping. What were some bits of emergent fun that you hadn't planned out ahead of time and instead discovered during prototyping?*

**SJ:** We found one aspect of emergent game play

while we were using the Metal Gear Solid mod for LBP. I had been thinking about how the projectile allowed us to move certain physics objects around the game space. After moving the objects around horizontally, I then built a tower where I shot the objects down vertically. I then thought to myself, "How can I move these up?" So using this same tower area, I placed little launchers that I could activate to throw the objects into the air, where I shot at them, moving them onto an ledge while in air. It was a lot of fun, and we returned to this concept a lot in SHADOWS.

We also created item spawners in the game for players to get objects out of in case the one they were using fell over a ledge or was lost. While testing the game, one of our designers just kept hitting the item spawner and object upon object started pouring out. Once we put a cap on the item spawners, we realized we could actually have them in the environment and that players could use them to fill large gaps with stuffed animals so they could cross, basically making a bridge out of the objects, which is actually one of the first things the player does in the game.

Our favorite bit of emergent game play actually did not make it in to the final build. We created a 4-wheeled



cart that Abigail would pull and push to catch falling objects inside. We made the cart a physics object so it would react to her appropriately. Well, one of the designers hopped inside the cart and began shooting the inside of the cart, which made it move forward. Repeatedly shooting the cart made it zoom 40 miles an hour through the level. We stood around the computer laughing for 30 minutes until we remembered we had to have a final build in the next 48 hours!

**JF:** *How did game play evolve over the course of developing the game?*

**José Luis López Zurita:** The game was pitched as a 2.5D platformer, but we really wanted to do something unique to it. We analyzed many games in this genre trying to find how to create something with a little twist. 2D arcade shooters are so

populate that it is difficult to innovate on them. The first prototype mixed an old school 2D platformer style with a 3D platformer/puzzling gameplay. Each enemy reacted differently when attacked, but all of them had a common design pattern and their reaction could trigger chains of reactions in other enemies.

As much as we liked the idea, soon we realized this project was beyond our scope. This problem was present during the whole development process, sometimes because the idea was too complex to be implemented and tested in the given time, and other times because we just weren't able to get a good prototype working in a reasonably short period of time. We knew that many game ideas that seem great don't play that great, so we needed a way to discover the good ones. We found in the LITTLE BIG PLANET editor a

great tool to accomplish that goal. It kept our level designers busy while the programmers were able to work on the core of the game and it gave us a lot of flexibility to test different ideas, although it also forced us focus exclusively on 2D gameplay.

**JF:** *Did you have a separate art team from level design, or did everyone have multiple responsibilities?*

**SJ:** Our art team carried our game for the first five weeks of development. We actually had such a strong art team that they were creating art assets before designers could put an official stamp on the design-required assets. This created a broken design pipeline where we were incorporating their assets into our design, rather than us telling them what we needed.

We designers were actually slowing down the artists who could pump

out beautiful assets in a matter of hours. Once we finally realized that innovative game play does not always start at a round table but from iteration on mechanics, we were able to move forward on the assets we needed from the art team.

**JF: What classes or areas of study were particularly helpful to you when it came time to create SHADOWS?**

**SJ:** Our curriculum at FIEA was built around us being able to create games with long development cycles. Our whole project was meant to be a simulation of working on an actual development team. For seven months we work on these games and control

our own development methodologies, so they train us continuously to be autonomous in these projects.

We are also put through varying design exercises and workshops to help us get to the core of a compelling gaming experience. FIEA also brings in industry veterans from all over the country to talk about game and level design, and we would always eagerly take notes, thinking of all the ways we could incorporate the lessons of our speakers—lessons they had to learn actually being paid to make games—into our own development cycle.

**JF: How has your interning experience been? How**

**closely did your game production experience in school match that of a commercial environment?**

**SJ:** I can absolutely say that my experience at FIEA made me a well-spoken designer in a studio full of industry veterans and tough guys. I am interning at EA Tiburon in Orlando, and I was lucky enough to work with the COO and two senior creative directors on developing new IPs for the studio. Thanks to a lot of the lessons I learned at FIEA, I was actually able to impress these guys on multiple occasions—or maybe I just said something that fit with an idea they already had? The interning experience has been incredible and a dream come true, because

I am actively being creative and being a designer, and having the opportunity to share my designs with a group of concept artists, where one day in the future these designs can turn into a shipped title.

I can actually see some semblance to how SHADOWS OF ABIGAIL did things—or how we meant to do things—in what the teams at EA are doing. They have weekly tasks, stand-up meetings, people constantly reviewing work, and a very precise pipeline and way of getting work done, which we definitely tried to accomplish during SHADOWS.

Best moment so far—I was discussing a design concept with the great Mark Turmell of NBA JAM and BLITZ fame because

I wanted to pitch the idea to Tiburon's senior management, and after explaining the idea to him, I could tell he was actually intrigued by my idea. Little Steve the design intern impressed Mark Turmell. It was a pretty great day.

**JZ:** I think the school does a great job of trying to push you to your limits. I had some industry experience before coming to FIEA and I can easily say this project was for me the most intense experience I ever had before as a game developer. The environment at FIEA was so close to a commercial one, that you would have a hard time trying to figure out that it is indeed a school.

—Jeffrey Fleming



## GET YOUR GAME ON! EVERYTHING YOU NEED TO KNOW TO GET INTO THE GAME INDUSTRY!

- NEWS and FEATURES FOR STUDENTS and EDUCATORS
- GETTING STARTED SECTION – an INVALUABLE HOW-TO GUIDE
- MESSAGE BOARDS

**DIGITAL COUNSELOR**

I'LL MATCH YOUR INTERESTS AND GOALS WITH THE RIGHT GAME RELATED PROGRAMS AND SCHOOLS FROM AROUND THE WORLD.

[www.gamecareerguide.com](http://www.gamecareerguide.com)

FIND OUT HOW THEY GOT THERE. > [gulfball.com.edu](http://gulfball.com.edu)

SMU QUITTING

Need Help Choosing the Right School?

**DIGITAL COUNSELOR**

GETTING STARTED

FEATURED

Download and Play the Latest Student Games!

Even your lunch break's dragging in three years.

**DigiP**

Game Career Guide

Download your FREE digital edition of the 2010 gamedeveloper Game Career Guide online at:  
[www.gamecareerguide.com](http://www.gamecareerguide.com)



Programmers & 3D Artists:  
Are you the trailblazer  
for our new frontier?

Friendly and small-feeling  
\$1B company (on NYSE)  
HIRING experienced  
team members full-time  
for our Naples, Florida studio

Apply online: [dreamtools.com](http://dreamtools.com)

... oh and the answer to your first question for us... Ninjas.

**DreamTools™**

If it doesn't look real, we didn't make it.

TURN YOUR  
PASSION FOR GAMING  
INTO A CAREER

**Game Art**

Bachelor's Degree Program  
Campus & Online

**Game Design**

Master's Degree Program  
Campus

**Game Development**

Bachelor's Degree Program  
Campus

**Game Design**

Bachelor's Degree Program  
Online



**Campus Degrees**

**Master's**  
Entertainment Business  
▶ Game Design

**Bachelor's**  
Computer Animation  
Digital Arts & Design  
Entertainment Business  
Film

▶ Game Art  
▶ Game Development  
Music Business  
Recording Arts  
Show Production  
Web Design & Development

**Associate's**  
Graphic Design  
Recording Engineering

**Online Degrees**

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

**Bachelor's**  
Computer Animation  
Entertainment Business

▶ Game Art  
▶ Game Design  
Graphic Design  
Internet Marketing  
Music Business  
Music Production  
Web Design & Development



**FULL SAIL**  
UNIVERSITY.

[fullsail.edu](http://fullsail.edu)

Winter Park, FL

800.226.7625 • 3300 University Boulevard

Financial aid available to those who qualify • Career development assistance  
Accredited University, ACCSC

## Game Design and Interactive Media in the heart of Boston

Our Creative Industries program strives to create, foster, and implement digital media innovation through research, education, and ground-breaking team-based interdisciplinary projects.

Collaborate and create with one of seven combined majors, or the Creative Industries minor.

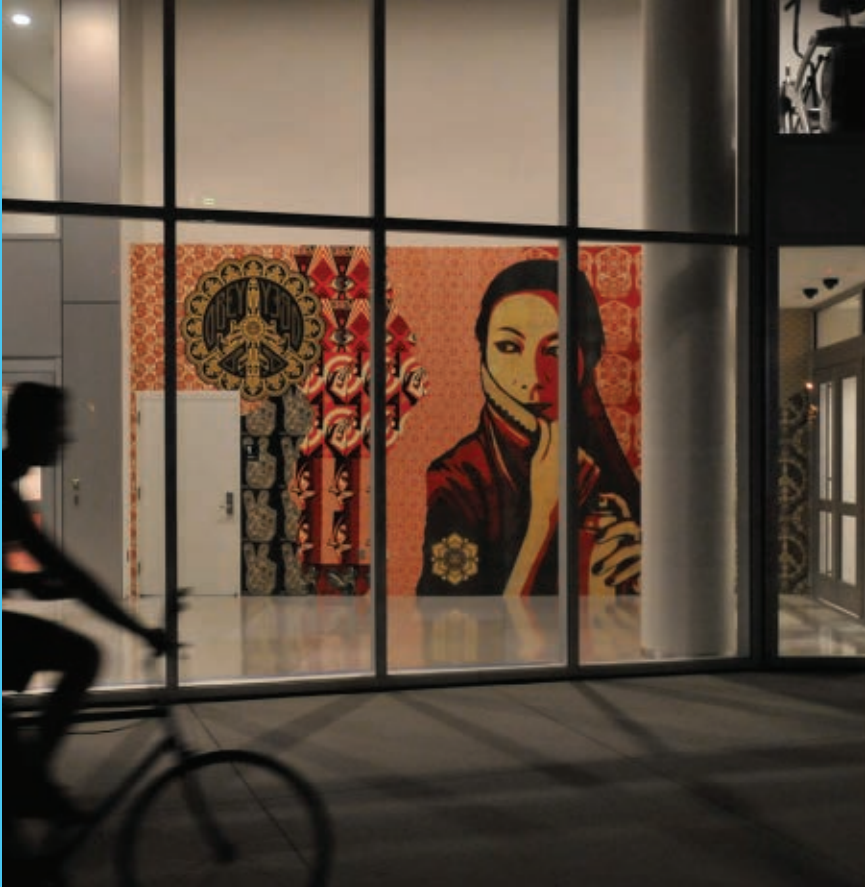
For more information, go to:  
[www.ci.neu.edu](http://www.ci.neu.edu)



### Northeastern

CREATIVE INDUSTRIES

Northeastern University  
 360 Huntington Avenue  
 Boston, MA 02115-5000

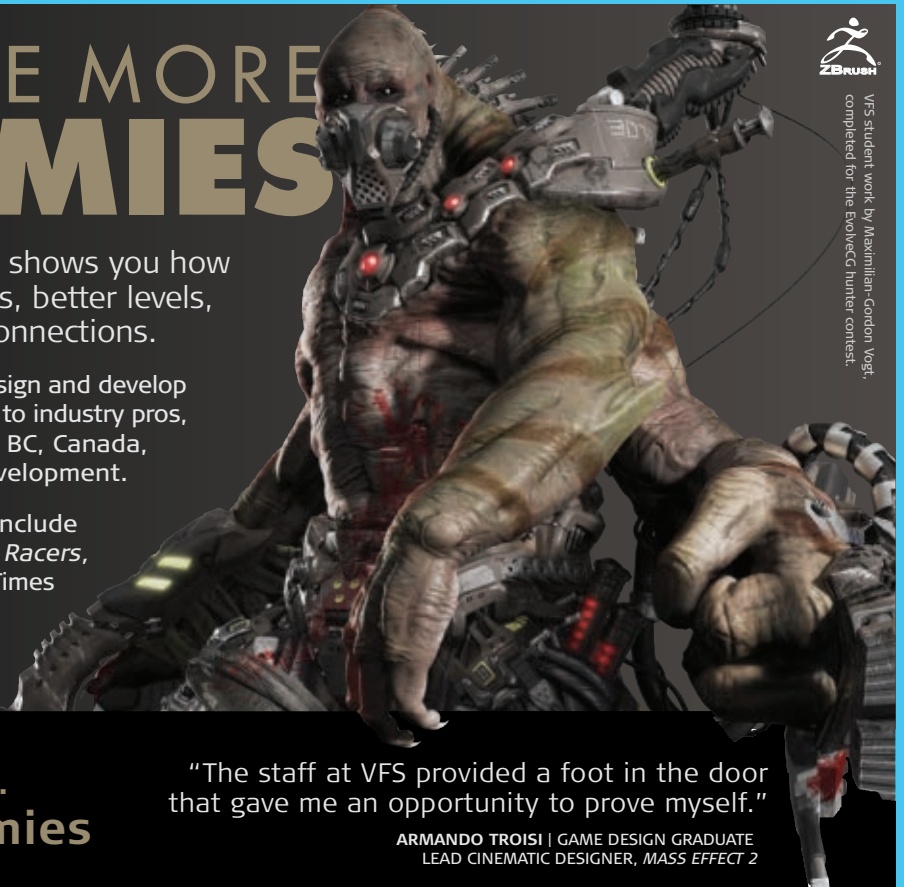


# MAKE MORE ENEMIES

**Game Design at VFS** shows you how to make more enemies, better levels, and tighter industry connections.

In one intense year, you design and develop great games, present them to industry pros, and do it all in Vancouver, BC, Canada, a leading hub of game development.

Our grads' recent credits include *Mass Effect 2*, *ModNation Racers*, and *Dead Rising 2*. The LA Times named VFS a top school "most favored by video game industry recruiters."



VFS student work by Maximilian-Gordon Vogt, completed for the EvolveCC hunter contest.

VFS

Find out more.  
[vfs.com/enemies](http://vfs.com/enemies)

"The staff at VFS provided a foot in the door that gave me an opportunity to prove myself."

ARMANDO TROISI | GAME DESIGN GRADUATE  
 LEAD CINEMATIC DESIGNER, *MASS EFFECT 2*

# le cnam enjmin

The graduate school of games and interactive media

The graduate school of game and interactive media welcomes you in France !

- Video Game is your passion ? Join us !
- Master degree in 6 speciality aeras
- Admission : applicants develop a project on a set theme

March the 22th  
Subject available online  
[www.enjmin.fr](http://www.enjmin.fr)

- ENJMIN
- 121, rue de Bordeaux
- 16 000 ANGOULEME Cedex FRANCE
- Phone : +33(0)5 45 38 65 68
- Fax : +33(0)5 45 35 65 66
- [contact@enjmin.fr](mailto:contact@enjmin.fr)



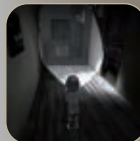
## Get THE Game Master Degree in Europe !

Come to see us at the GDC 2011, booth #2641, South Hall

Summer language training on the Atlantic coast



- Software design and development
- Sound and Music Design
- Project management
- Game Design
- Ergonomics
- Visual Design



## INVEST IN YOURSELF. BECOME A MASTER OF DIGITAL MEDIA.

We offer a 20-month Master's program in entertainment technology and digital media that combines industry-facing curriculum, real-world projects and a four-month internship. The MDM degree is jointly awarded by four leading Canadian post-secondary institutions: The University of British Columbia, Simon Fraser University, Emily Carr University of Art + Design, and the British Columbia Institute of Technology.

Our students come from around the globe, and from diverse undergraduate and professional backgrounds – including media and fine arts, computing science, natural and social sciences, business, and engineering.

Come work and play with us in Vancouver BC: Canada's video game capital.

For more information about the Masters of Digital Media (MDM) Program, go to [mdm.gnwc.ca](http://mdm.gnwc.ca)

VISIT US AT GDC IN SAN FRANCISCO | FEB 28 – MAR 4 2011

CENTRE FOR  
DIGITAL MEDIA  
Great Northern Way Campus





# Game Art & Animation

Associate's Degree

**Careers Include:**

- Animator
- Modeler
- Technical Animator
- Level Designer

**Program Highlights:**

- Motion Capture facilities
- Utilizing: Unreal III Engine, Maya, Motion Builder, Mudbox, Body Paint, & more

**Additional Emphasis:**

- Story development
- Performance
- Cinematography
- Traditional art
- Color theory

**Launch your career today!**

Madison: 800.236.4997 Minneapolis: 877.416.2783



MADISON: 2702 Agriculture Drive | Madison, WI | madisonmedia.edu MINNEAPOLIS: 4100 West 76th St. | Edina, MN | minneapolismediainstitute.com

## ADVERTISER INDEX

COMPANY NAME	PAGE	COMPANY NAME	PAGE
Blizzard Entertainment	14	Perforce Software	3
ENJMIN	62	Rad Game Tools	C4
Epic Games	6	Seapine Software	C2
Full Sail Real World Education	60	Simpson Strong-Tie	60
Madison Media Institute	63	TechExcel	21
Masters of Digital Media Program	62	TransGaming	35
Havok	25	Vancouver Film School	61
Natural Motion	10	VSoft Technologies	43
Northeastern University	61	Zoo Entertainment	C3

gd *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. All content, copyright *Game Developer* magazine/United Business Media LLC, unless otherwise indicated. Don't steal any of it.



# STUDIO PRAVDA

## THE PEOPLE'S GAME DEVELOPMENT STUDIO MARCHES ONWARD

### PROJECT PROCEEDS ON SCHEDULE TOWARDS CERTAIN VICTORY

» Thanks to the great wisdom and all-seeing fatherly guidance of the Great Studio Leader, our indomitable video game project advances relentlessly toward the long-sought Day of Ship. All subcommittees have completed their reports to the central office with the expected news that each division is producing to their quotas in lockstep with the flawless Six-Point Plan.

"Once the Day of Ship has been completed and our enemies are dashed to pieces upon the rocks of our unbending will, all of the righteous and fearless workers under the care of the Great Studio Leader are guaranteed a secure and prosperous existence," the Management Cadre announced in a statement. "We will all share in the riches to come!"

### ENEMY STUDIOS UNABLE TO COPE WITH OUR MIGHT

» Covert operators continue to keep us informed of their pathetic and hopeless attempts to create software titles to compete with our glorious future success. In a recent report to the Management Cadre and the Great Studio Leader, intelligence officials explained how every other game development studio is completely bungling their strategy and can barely manage to scrape along due to constant missteps and complete lack of talent, skill, knowledge, organization, and competence in any field.

"As we examined the results of our research, it became clear that nobody understands video games except for us," a Senior Intelligence Official reported. "At first, we found it difficult to believe that Valve, Blizzard, Rockstar, BioWare, Microsoft, Nintendo, and every other game company is so wrong about what makes a good game and what sells in the marketplace, but under the watchful guidance of the Great Studio Leader, we soon realized that this is simply the truth of the matter."

### GENERAL TSO'S CHICKEN BRINGS JOY TO GRATEFUL WORKERS

» A large tub of General Tso's chicken was placed in the break room on Saturday at 9 PM, where it warmed the hearts and stomachs of appreciative studio employees. An orderly queue was formed in front of the lavish feast where workers ate their fill of healthy and nutritious sustenance and wiped away tears of emotion.

One employee was heard to say, "I am very deeply moved. The Great Studio Leader thinks of us even on the weekends. Thanks to this outpouring of generosity, we shall certainly redouble our efforts with a fresh spirit."

The dinner ended with an impromptu and well-received performance of several oral histories related to the Great Studio Leader, including the popular account of his unmatched ability to consume large quantities of Sriracha brand hot sauce.

### OPINION: SELF-SACRIFICE IS THE KEY TO AN INVINCIBLE SPIRIT

» The other day, during the course of my official duties at an industry conference, I had the chance to communicate with employees of several other game development studios.

"I'm not so sure about our production schedule," said one.

"I have some concerns with our creative direction," said another.

Frankly, I was shocked to hear such careless thoughts being tossed around, so I asked them how they could enjoy their livelihoods if they did not fully agree with their leaders.

"Sometimes we disagree with each other," they said. "But the people we work with are mostly cool, so we have arguments, hash things out and make compromises."

This demonstrated the decrepitude of all studios outside of the care of the Great Studio Leader. For the first time, I saw with my own eyes that our enemies operate in a continual state of disorganization and chaos. It is little wonder that, come the Day of Ship, our inherent superiority will be clear for the entire world to see.

I came away from this educational and enlightening experience even more thankful that I have been granted the fortune to work here. I realized even more clearly than before that the smallness of my own insignificant life pales in comparison to the shining glory that shall soon be upon us!

—Contributed by Department Overseer, No. 7 Production Department

### BE EVER VIGILANT: REPORT TREASONOUS ACTIVITY

» Because our game studio is the envy of the entire world, enemy agents are often sent to our borders to discover our secrets and undermine our coming triumph. These spies may look and talk like us, but they deserve no mercy. If you suspect treason, report it to your division overseer immediately. Anyone who reports a traitor will be rewarded with a luxurious Logitech computer speaker system for their individual workstation!

Treasonous activity includes but is not limited to:

- ★ Questioning any aspect of the Six-Point Plan.
- ★ Asking why a superior is doing something in a particular way.
- ★ Claiming that any part of the project is behind schedule.
- ★ Thinking or believing that another game may be doing something better than ours.
- ★ Failure to publicly express constant and spirited enthusiasm for the project.

### EMPLOYEE REVIEW PROCESS AGAIN DETERMINES IRREFUTABLE TRUTHS

» Justice and truth have once again prevailed this year thanks to the Great Studio Leader and the Management Cadre's Annual Review Process, which continues to ensure that the studio treats all of its workers fairly and equally.

The contribution of each worker over the last year was precisely measured using a robust and unbiased scientific process. They were then assigned a classification into one of two groups. Generous rewards, including Big Grab-sized bags of Fritos® and hearty backslaps, were equitably distributed among those belonging to the first employee group known as Studio Patriots.

Employees in the second group, Future Potential Valued Employees, were judged to be in need of reeducation, and were compassionately given the opportunity to compose patriotic essays on a topic of their choosing before being exiled from the Studio, pending post-Mental Reassignment re-application. 🇺🇸

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



# INDEPENDENT PROPELLER AWARDS

\* at South by Southwest \*

indiePub is accepting indie game submissions for the First Annual Independent Propeller Awards Competition which will come to life at South by Southwest's ScreenBurn in Austin, Texas this March. Finalists will get the chance to demo their game at SXSW where \$150,000 in cash and prizes will be given away, and developers will get a shot at having their games published by indiePub through its sponsor, Zoo Publishing.

Check out [indiePubGames.com](http://indiePubGames.com) to see how your game could become the next big thing!

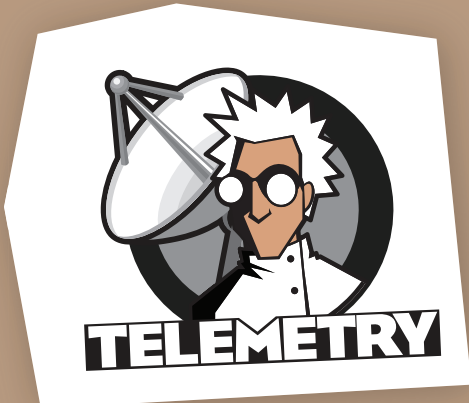
Register Today at  
[WWW.INDIEPUBGAMES.COM](http://WWW.INDIEPUBGAMES.COM)  
Submit by Feb 18th

THE NEWEST RAD TOOL IS

yes, it's **NEW**

**NOW SHIPPING**

**OUT**  
OF THIS  
**WORLD**



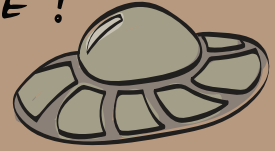
# TELEMETRY

42

is a programming library and set of tools for instrumenting, profiling, tuning and visualizing application **PERFORMANCE!**



VISUALIZE real-time game performance,



see **WHEN** things happen — not merely **WHAT** happened!



PROBE the hierarchical display —



see thread interactions, context switches, and mutex locking!

THIS ISN'T JUST ROCKET SCIENCE, THIS IS **rad!**



[www.radgametools.com/telemetry](http://www.radgametools.com/telemetry)  
(425) 893-4300