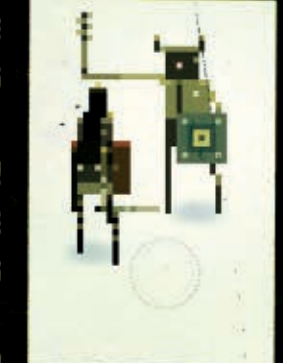


# gd

THE LEADING  
GAME INDUSTRY  
MAGAZINE VOL 10  
NO 8 SEPTEMBER  
2011 INSIDE:  
INTELLIGENT  
CAMERAS



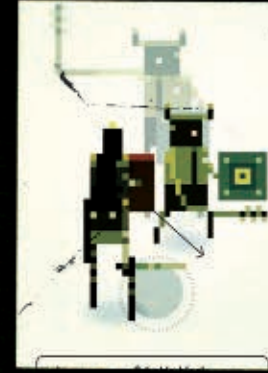
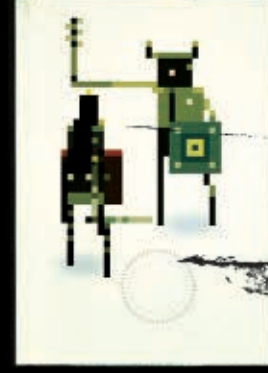
18

19

20

21

22



9 5 6 4

12

13

14

15

16

17



9 5 6 4

6

7

8

9

10

11



Woz strikes!

IF THE PLAYER BLOCKS THICE...  
-set ursula's powerstamina to 0  
-bar of rest, repeat attackPattern4  
-bring up infohelp button

IF THE PLAYER DODGES  
-woz goes to attackTBvulnerable  
-ursula gets a +3 to powerstamina  
-woz is vulnerable in LB & RT sector



## The best ideas evolve.

Great ideas don't just happen. They evolve. Your own development teams think and work fast.

**Don't miss a breakthrough.**  
**Version *everything* with Perforce.**

Software and firmware. Digital assets and games. Websites and documents. More than 5,000 organizations and 350,000 users trust Perforce SCM to version work enterprise-wide.

**Try it now. Download the free 2-user, non-expiring Perforce Server from [perforce.com](https://perforce.com)**

Or request an evaluation license for any number of users.





### DEPARTMENTS

- 2 **GAME PLAN** *By Brandon Sheffield* [EDITORIAL]  
The Portable Future
- 4 **HEADS UP DISPLAY** [NEWS]  
The Difference Initiative, HALO 2600 Source Code Released, and Flashback, Recipes, and Zombies

### POST MORTEM

**14 SECTION 8: PREJUDICE**  
With SECTION 8: PREJUDICE, TimeGate took what was planned as a console game and presented it as a \$15 downloadable title. Not only did the team have to change its way of thinking about pricing and presentation, it also had to work out how to shrink assets and scale the design appropriately to fit within the download limit. As this sort of story grows more common, the lessons learned here become more important. *By Adel Chaveleh*

**22 SWORD & SWORCERY**  
Much to everyone's surprise, not least the developers, a tiny iOS adventure game known as SWORD & SWORCERY was featured twice in Apple's shop and went on to sell over 250,000 units. This game, with its odd visuals, odd design, and odd music, was in turn an odd collaboration by three different creatives. This went both as well and as poorly as you can imagine. *By Craig D. Adams, Kris Piotrowski, and Jim Guthrie*

### FEATURES

**7 THE STICKY COLLISION BEAM**  
Creating intelligent third-person cameras is rife with difficulty. There are enemies to keep in view, obstacles to avoid, and important objects to highlight. Here, engine programmer Eric Undersander proposes a camera with a sticky collision beam to bypass many of the major issues faced by AI-controlled cameras. *By Eric Undersander*

**40 BEAUTIFUL CODE: AN INTERVIEW WITH BILL BUDGE**  
Bill Budge paved the way for user-generated content with his early games PINBALL CONSTRUCTION SET and RASTER BLASTER. But that wasn't really his intention—he just likes making software tools. This interview tried to get at the heart of his love for code. *By Brandon Sheffield*

**31 TOOL BOX** *By Bradley Meyer* [REVIEW]  
Soundminer HD

**35 THE INNER PRODUCT** *By Nick Darnell* [PROGRAMMING]  
Automated Occluders for GPU Culling

**42 THE BUSINESS** *By David Ederly* [BUSINESS]  
Realm of the Mad Developers

**44 PIXEL PUSHER** *By Steve Theodore* [ART]  
Conan the Illustrator

**47 GOOD JOB!** *By Brandon Sheffield* [CAREER]  
Q&A with Scott Brodie, Who Went Where, and New Studios

**48 DESIGN OF THE TIMES** *By Damion Schubert* [DESIGN]  
Good Grief!

**50 GDC NEWS** *By Staff* [NEWS]  
Game Developers Choice Online Awards Honor Kesmai Founders, SOE's EVERQUEST, GDC China Registration Now Open, First Talks Revealed

**51 AURAL FIXATION** *By Jesse Harlin* [SOUND]  
Playing The Field

**53 EDUCATED PLAY** *By Tom Curtis* [EDUCATION]  
WOMP!

**56 ARRESTED DEVELOPMENT** *By Matthew Wasteland* [HUMOR]  
Outsourcing Life



# THE PORTABLE FUTURE

IN THE HANDHELD MARKET, SONY AND NINTENDO'S GREATEST ENEMIES MAY BE THEMSELVES

**BY THE TIME YOU READ THIS, WE'LL** already know the preliminary results of Nintendo's 3DS price drop. As of this writing, we can only speculate, but we do know that this is one of the most drastic price drops that has ever been made so close to a system's launch. The decision-makers at Nintendo did it because they had to. But why did they have to?

One of the big problems is differentiation. Without the 3D, the 3DS is more or less a DS with nicer graphics. Without a big software push, that's not nearly enough to warrant a purchase if you have a DS already, especially if you've already replaced your older DS with a DS Lite, DSi, or XL. Consumers are clearly taking a wait and see approach to the 3DS, but while they wait, DS sales are dropping significantly. It seems that people are waiting to see whether the 3DS will become relevant to them, and they are already mostly done buying DS units and games, as those numbers have trailed off significantly. In effect, the 3DS is competing with the DS, because that's what it supplants. And if the 3DS isn't currently compelling enough to buy, and it replaces the older model, it makes both systems look bad.

And on the consumer side, with so many handheld gaming options available, including the PSP and iOS devices, any game player with more than one handheld has a lot of options right now. If people aren't buying many DS and 3DS games, they're probably buying something else. It seems to me that Nintendo has a mindshare problem right now, and consumers won't wait forever.

This is what I think the price drop addresses. But of course, the ultimate change will come with software people want to buy. This, more than anything, can change the course of the 3DS.

## VITA

» The Vita has a slightly different problem. Sony doesn't have to worry about the Vita killing off the

PSP, because that has essentially happened in Western markets already. The PSP still does well in Japan, where, popularity-wise, piracy is less, and MONSTER HUNTER is decidedly more. But even there, the PSP is definitely ready for a successor.

The Vita has a touch screen on the front, and a touchpad on the back (and if that winds up being a fantastic gameplay enabler I'll eat my hat). That's a bit of differentiation, but in a game like UNCHARTED: GOLDEN ABYSS, how central will the touch screen really be to core gameplay? Everything that's been announced for the Vita so far has been pretty hardcore-oriented. The console is essentially a PSP with nicer graphics—but in opposition to Nintendo, I don't think this is a problem for Sony. The PSP was always pitched as a console experience in a handheld, not a disruptive device. The DS was meant to change the way we played games—the PSP was just supposed to bring our consoles with us. Replacing a machine with nice graphics with a new machine with nicer graphics makes more sense for Sony than it does for Nintendo, because that's how the console has always been marketed.

The Vita is currently being targeted more like a portable home console—it's not trying to reclaim the PSP audience, it wants to win over the PS3 audience. This brings memories of the PSP and the PS2, which had a similar issue.

The Vita and the 3DS seem to have slightly different issues to tackle. One is the big brother of another handheld, and the other is the little brother of a home console.

## DIGITAL CORE

» As we continue to make the rocky transition from packaged to digital goods, it becomes increasingly obvious that an online store is the key to a handheld's success. Smaller games are generally associated with downloads nowadays, and handhelds are by

and large living in that "smaller" arena. While the platform holders insist that they offer deeper experiences than the smartphones, that has to be backed up not only by the software, but also by the delivery of that software. If my Droid can easily download games on the train, shouldn't my Vita be able to do it better, if it's the deeper, more dedicated console? Downloadable stores are going to be an incredibly important factor in the next generation of handhelds' success, and both Nintendo and Sony have a long way to go here.

If Nintendo and Sony are going to keep saying that their handhelds offer deeper experiences for core players than iOS and Android, they should back that up with the biggest differentiator between the two system styles—buttons and control. If you're going to say you're better than Apple, stop trying to shoehorn Apple's ideas into your product, and emphasize games that control well with actual pads and buttons. That speaks to the core player more than a touchpad. In this regard, I think Sony's announcements of Vita titles is more compelling, with UNCHARTED, RESISTANCE, and others. Nintendo continues to release software that could potentially be on an iPhone alongside its youth-core offerings like KID ICARUS. (Sony's Xperia phone could be a contender here, if they were making a bigger push with it.)

The 3DS and Vita are not only competing against the smartphones, they're also competing against each other (a bit), and older products from their respective companies (much more). The approaches Nintendo and Sony take to these predicaments in the next few years will determine the future of dedicated handhelds, as consumers seem to often feel that smartphones are "good enough."

—Brandon Sheffield  
twitter: @necrosofty

United Business Media  
303 Second Street, Suite 900, South Tower  
San Francisco, CA 94107  
t: 415.947.6000 f: 415.947.6090

## SUBSCRIPTION SERVICES

FOR INFORMATION, ORDER QUESTIONS, AND ADDRESS CHANGES  
t: 800.250.2429 f: 847.763.9606  
e: [gamedeveloper@halldata.com](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**  
Jade Kraus | [jkraus@gdmag.com](mailto:jkraus@gdmag.com)  
**ART DIRECTOR**  
Joseph Mitch | [jmitch@gdmag.com](mailto:jmitch@gdmag.com)  
**DESIGNER**  
Jessica Chan  
**CONTRIBUTING WRITERS**

Tom Curtis  
Jesse Harlin  
Bradley Meyer  
Steve Theodore  
Nick Darnell  
David Edery  
Damion Schubert  
Matthew Wasteland  
**ADVISORY BOARD**  
Hal Barwood Designer-at-Large  
Mick West Independent  
Brad Bulkley Microsoft  
Clinton Keith Independent  
Brenda Brathwaite Lolapps  
Bijan Forutanpour Sony Online Entertainment  
Mark DeLaura THQ  
Carey Chico Independent  
Mike Acton Insomniac

## 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



United Business Media  
WWW.UBM.COM

# around the Arab world there are more than **180 million** people under the age of 25\*.



## yet Arabic game development is still in its infancy.

**your opportunity is right here, right now at twofour54° in the heart of Abu Dhabi.**

The Arab world is one of the world's fastest growing media markets. With a young population of 180 million under the age of 25, more than 80% with mobile phones\*, strong broadband take-up and new gaming innovations, it's a prime opportunity for Arabic gaming businesses.

We empower businesses across all media platforms from television, film, radio, gaming, digital, animation and publishing – with world-class training from **twofour54° tadreeb**, state-of-the-art production facilities with **twofour54° intaj** and venture funding and support for Arab creative entrepreneurs from **twofour54° ibtikar** – to seize every media opportunity the region has to offer.

It's all part of our vision at **twofour54°**, creating a centre of excellence for Arabic content creation in Abu Dhabi.

**we are twofour54°. are you?  
find us. join us. create with us.**

+971 2 401 2454 [twofour54.com](http://twofour54.com)

follow us on:



**twofour54°**

Abu Dhabi

content creation community

\*Sources: Arab Media Outlook 2010. Media on the Move 2009. A.T. Kearney. Introduction to Gaming. Michael Moore. Screen Digest. IDC.

twofour54° is an initiative of the Abu Dhabi Government.



# The Difference Initiative

### Mare Sheppard's drive to bring more women to game development

\\ There's a theory that's been going around about game development for years: Games can't meaningfully evolve because the population creating them never changes. Kids grow up playing certain types of games, become adults, and then make games for other people like them.

A creative group of Toronto independent game developers believes that the game industry needs a little more hybrid vigor. Encouraged by what emerges when different kinds of artists—including those not necessarily from the world of games—collaborate, these game developers are fast-tracking initiatives to bring even more different kinds of people into the field.

Toronto's Artsy Games Incubator has been working to this end since 2007, with the support of Jonathan Mak's Queasy Games (EVERYDAY SHOOTER) and Raigan Burns and Mare Sheppard's Metanet Software (N/N+). The goal of the incubator—and of the development networking group that sprung from it, the Hand Eye Society—is to welcome creative people from all kinds of media who may be interested in making games, but who lack the tools or knowledge to actually engage in development of their ideas. Using accessible, relatively simple development tools like GameMaker and GameSalad, Toronto's indie game developers gather in support of newcomers to help them realize their ideas.

Recently, the Hand Eye Society teamed up with the Toronto Independent Film Festival (TIFF) for an event called the Arcadian Renaissance, an arts festival where custom-built arcade cabinets played host to indie games from the community. The groups applied for funding with the Ontario Media Development Corporation, known for its generous support of local artists.

The Ontario Media Development Corporation is "the backbone of so many Toronto indies that it's unreal," says Metanet's Mare Sheppard,

whose company also receives support from the organization.

The new funding will support four major projects between TIFF, the Hand Eye Society and Ryerson University, focused on mixing things up in the game industry. One is an event that matches hardware hackers with game developers interested in new physical inputs, and another encourages comics creators and game makers to collaborate. There is also a youth-oriented initiative.

First up, though, is the Difference Engine Initiative, a project coordinated by Sheppard. The Difference Engine is focused, at least at first, on bringing more women into the game industry, in the continuing spirit of the idea that more diversity means better projects.

"There's this huge, homogenous, very insular, established set of developers right now in the game industry, and it happens to be mostly white and mostly male," Sheppard says. "From that, you can really only get a certain amount of innovation."

Just like cross-disciplinary collaborations in the art community have resulted in more interesting projects, the Difference Engine Initiative believes that more diversity on teams can only mean better games. "If we had more voices and more opinions and more people coming in, then we would be able to take bigger steps in releasing games that represent different people, because they're involved in the development process," Sheppard says.

"Indies in particular are usually making huge progress in terms of innovation ... if there were even more diversity in the industry, I think we'd be seeing unbelievable things," she adds. "The collaboration is great, because it brings in people who aren't limited by the structure of the game industry. They have no preconceptions about what they should be making."



Mare Sheppard.

One barrier to entry for people who wouldn't traditionally be attracted to game development is that it appears so complex that it's hard to know where to begin. The Difference Engine wants to start there.

"There's a lot of resistance in the current game industry—that's the thing about homogenous groups. They really repel outsiders," suggests Sheppard. "There's a ton that we have to do; this is all a cultural problem. We actually have to change how people think to make these environments more appealing and welcoming to outsider groups."

One way to achieve this is to have visible role models in the field, so that women curious about game development can observe a number of successful figures already working therein. Difference Engine participants will have the opportunity to meet notables like thatgamecompany's Kellee Santiago and Robin Hunicke, and Kokoromi's Heather Kelley, among others, as an avenue to see what's possible.

"We'll bring in local women game makers and some international developers to chat with people, give them feedback, and help them along," Sheppard explains.

Six participants will be selected to attend six weeks of sessions, split across three hours per week, along with coordinators who will help them make games. "It's like a crafter's circle," Sheppard describes.

"It's loose and low-key, and it's about peer mentorship. You get feedback from people, you show your progress so people can play your game, and it's a supportive atmosphere, which will ideally help people take ideas they have and bring them into fully formed games."

It's a place to start, she says.

"One of the main things we want to note is we know this is not going to be a perfect solution ... but if we sit around and keep talking and waiting for this opportunity to come, we're going to miss times like this when we can actually learn from people."

Sheppard herself was slightly concerned at the initial conception of the Difference Engine—a low-pressure entry-level game development group for women. "I was concerned about the idea that this is going to set people up for easy sexist criticism," she admits. "I really don't like emphasizing the difference in gender. It's irrelevant; there are nurses and there are male nurses—I don't want there to be 'developers' and 'female developers.' It's ridiculous."

"And then I thought ... some people really respond to this, and it's totally worth it to help those people," she reflects. "I think, in light of that, this could work."

For more information about this initiative, visit <http://handeyesociety.com/project/the-difference-engine-initiative>.

—Eric Caoli

# Halo 2600 Source Code Released



\\ Over a year ago, ex-Microsoft Very Important Guy Ed Fries wanted to test whether he could still code in assembly, and decided the 2600 would be a great place to start. So it was that HALO 2600 was born—a top-down, multi-screen shooter with a high degree of sophistication for the console.

For those interested in taking a similar path, Fries has taken the next step. “I thought I’d celebrate the one-year anniversary of the release of HALO 2600 at the Classic Gaming Expo by making the source code available,” said Fries in an AtariAge forum post. “It’s not particularly cleaned up or well documented but I put it here,



as is, with the hopes that it will help some future 2600 homebrew programmers, just as I was helped by others who posted their code.”

The code should be useful for those looking to learn, as are the AtariAge forums themselves. Fries adds, “Indentation is (more or less)

consistent if you view it in the editor it was created in. Which is ... wait for it ... Notepad.exe!”

Interested persons can find the code attached to the AtariAge thread here: [www.atariage.com/forums/topic/185693-halo-2600-source](http://www.atariage.com/forums/topic/185693-halo-2600-source).

—Brandon Sheffield

# Flashback, Recipes, and Zombies

How a cooking blog led to the upcoming downloadable console game AMY



\\ In the modern world, many friendships begin on the internet. So it was with the wife of Eric Viennot, creative director of Lexis numerique (RED JOHNSON’S CHRONICLES). Mrs. Viennot runs a cooking blog of some renown, and one day, one of her fans asked for a favor. He wanted a certain recipe, so they met up to discuss it. They talked about their lives, and Mrs. Viennot related that her husband was into video games. “Oh, so am I!” said the mystery

fan. By happenstance, he turned out to be Paul Cuisset, designer of FLASHBACK for Amiga, among other platforms.

Viennot set up a meeting between Cuisset and her husband, who frankly presumed this was just another guy off the street with pie-in-the-sky ideas. Cuisset was pitching, naturally, a cooking game. To hear Djamil Kemal (marketing director of Lexis Numerique) tell it, “The guy was trying to present

us with a cooking video game and it was like ‘Oh my god you’re a guy who is doing small games in the background and we don’t want to do that!’”


But the company was impressed by the video presentation. “We asked who did the video,” adds Kemal, “and he said, ‘Oh it was me.’ And who did the interface? He said, ‘It was me too.’ And we were like ‘Whoa, my god, you’re good!’ Finally we asked what it’s coded with and

he said, ‘Oh that’s C, but I have to redo the engine soon.’ And we were like, ‘Oh my god, and what’s your name again?’”

“Oh, my name is Paul Cuisset.” Cuisset hadn’t been very prominent in the game industry recently, so it was understandably after some working of the mental gears that Lexis Numerique realized who they had at hand. Together, Lexis Numerique and Cuisset have gone on to create a game called AMY, which could be pitched as ICO meets RESIDENT EVIL. You play as a young woman who has been bitten by someone infected with a plague that has spread through a city, and all infected persons are being shot on sight by the military. But she has with her a little girl who has the power to stave off the infection. The dynamic

between these two drives what the team hopes will be an emotionally driven high-end downloadable console experience.

“So we decided to create a company together,” said Kemal. “It’s his studio, called Vector Cell. We invested in it, and we were really, really happy. It was like serendipity. It was really out of luck like this and we ended up doing a great game together, so we’re really giving him full control in terms of creativity, and we’re taking charge of the production part and the business side and he is the head of development.”

So, just remember. The next time your husband or wife offers to introduce you to their friend who “really likes games,” don’t discount it outright. You could be this close to rediscovering the next Paul Cuisset. 

—Brandon Sheffield

# BLIZZARD

ENTERTAINMENT



WORLD  
WARCRAFT



STARCRAFT



DIABLO

## 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)



# the sticky collision beam



## coding RECKONING's third-person camera for obstacle avoidance

ERIC UNDERSANDER

Third-person cameras are a staple of 3D games, from SUPER MARIO 64 and TOMB RAIDER to more recent franchises like ASSASSIN'S CREED and UNCHARTED. KINGDOMS OF AMALUR: RECKONING is an action-RPG developed at Big Huge Games, and our camera system presented challenges that are likely familiar to many. In this programming-centric article, I'll give a brief overview of our third-person camera and then delve into three of our biggest technical hurdles: smoothing camera motion, framing enemies, and avoiding obstacles. >>>





## Basics

/// We define our camera in 3D by a focal position and an offset to the camera. The camera simply looks at the focal position. We often consider the offset vector in spherical coordinates, namely offset distance, yaw, and pitch, as shown in Figure 1. I'll refer to offset distance adjustments as zooming in or out.

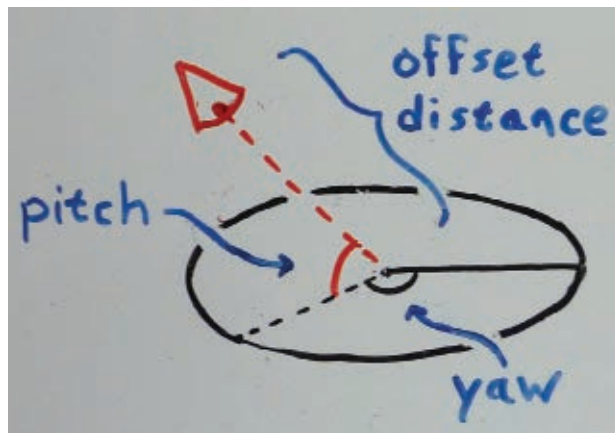


Figure 1: The camera offset vector.

The focal position is generally the player character, about chest high. A typical offset distance of 4 meters and a pitch of 25 degrees above the horizon produces a somewhat-overhead view of the character and his surroundings.

The player can take full manual control of yaw and pitch at any time, using a single controller analog stick (or the PC mouse) to rotate the camera about the character. Whenever it appears that the player is not opting for manual control, we make several intelligent, automatic adjustments to yaw and pitch:

- When the player character is running through the world, we gradually yaw the camera to stay directly behind him as he changes direction, so the player can see where he's going.
- When the character is on sloped ground, we pitch the camera to look uphill or downhill.
- If the player has left camera pitch at an extreme angle (perhaps accidentally), we gradually restore it to within a designer-tuned safe range.

With that brief introduction to our third-person camera out of the way, I'll jump into the first of our three main topics: smoothing camera motion.

## Smoothing Camera Motion

/// In order to follow the player character through the world, we could simply set the camera focal position to the character position every frame. This would keep the character exactly centered on-screen regardless of his movement.

However, the character can start, stop, and change direction very quickly, especially during combat. As a general rule, our camera should avoid abrupt acceleration, if only to prevent player motion sickness.

There's also a deeper motivation. I worked closely with our lead combat designer Joe Quadara on all aspects of RECKONING's camera, and he explained that, "Our perception of player movements is dampened when the camera tracks the player precisely." In other words, by smoothing camera motion, we'll introduce camera lag and draw more attention to the player character's movements. For example, a character's quick dash to the left will visually jerk him toward the left edge of the screen.

Our first attempt at achieving camera lag was to simulate our camera focal position as a real-world cameraman chasing the player character through the world. Our cameraman could accelerate to a run and slow to a halt. He could "turn on a dime" when moving slowly, but could only turn in a wide arc when running. This approach proved to be a total failure! The focal position often took an absurd, looping route in response to a sudden halt or sharp direction change by the character.

## Moving Average

/// Our next approach was a smoothing technique called a moving average. Here, we keep a list of character positions from all previous frames within the last few seconds (I'll use the variable  $d$  for this duration). For each camera update, we set the focal position to be the average of these recent character positions. The average changes smoothly over time as we add new character positions to the list and discard old ones (older than  $d$  seconds).

Figure 2 shows our moving average in action. Regardless of any doubts you may have about this smoothing technique, my stick-figure player character is adorable! You need to just agree with me on that. What's he doing there, a little sword attack? Later, he performs a quick dash, covering several meters in just a few frames. Nice work, tiny hero!

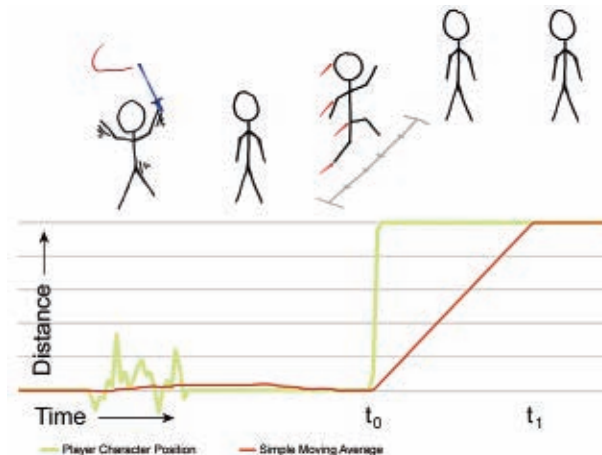


Figure 2: Smoothing character movement using a moving average.

Although the character's sword attack involves many small, rapid movements, the camera focal position remains fairly still. This is good! However, the response to the quick dash is a problem. At times  $t_0$  and  $t_1$ , the camera exhibits exactly the kind of abrupt acceleration we're trying to avoid.

The sensation would be similar to riding an escalator—although the escalator track moves slowly, the acceleration is rather harsh as you step on and step off. In math-speak, the first derivative (velocity) is discontinuous at these points, so the camera position function has only  $C_0$  continuity. (In non-math-speak, this means "not very smooth.")

## Smoother Weight Function

/// Recall that when computing our average, we give equal weight to all recent samples and that there's an abrupt cutoff after  $d$  seconds. This is a simple moving average. We can improve our smoothing behavior by switching to a weighted moving average and carefully choosing our weights.

Figure 3 shows weight as a function of sample age. Our new double-s weight function is shown alongside the weight function of our simple moving average.

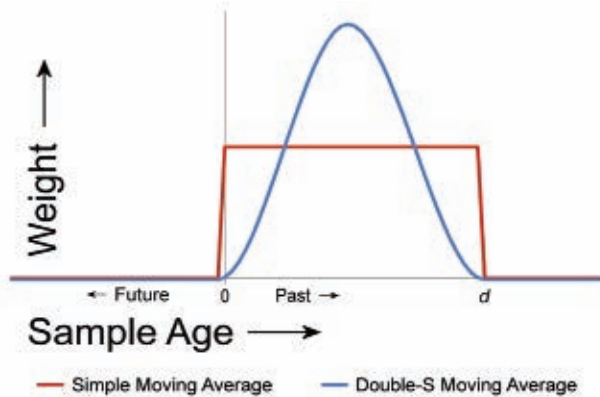


Figure 3: Weight functions for moving averages.

Notice how the X-axis shows both positive and negative values, corresponding to past and future samples. [Of course, when smoothing in real time, we can't know future samples, so their weights must be zero.]

Looking at this graph, the simple moving average's weight function is clearly not very smooth. It makes large vertical jumps at age=0 and age=d. Although a rigorous analysis is beyond the scope of this article, it should be evident that our earlier velocity discontinuities at  $t_0$  and  $t_1$  are due to these discontinuities in the weight function.

Our new double-s weight function is much smoother, and the result is shown in Figure 4. In response to the player character's quick dash, our focal position smoothly accelerates and decelerates. Success!

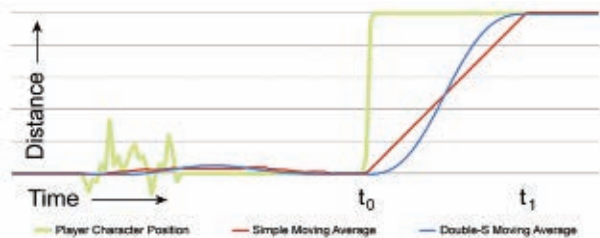


Figure 4: Simple versus Double-s moving average.

Our double-s weight function is constructed using the function  $3x^2 - 2x^3$ , which remaps values between 0 and 1 to an s shape. Crucially, our new weight function's derivative is continuous everywhere, including the ends of the s pieces at age=0 and age=d. This means our weight function has  $C^1$  continuity, and it's possible to further tweak this function as long as  $C^1$  continuity is maintained. For example, by stretching the falling s-curve along the x-axis and also raising it to a power, we get an asymmetric double-s curve with a long, thin tail on the right side. This subtly affects camera acceleration and deceleration in a pleasing way.

Now that we've smoothed out the jitters and jumps in our focal position movement, I'll shift to our next major challenge: framing combat.

## Framing Combat

/// A player's typical combat encounter in RECKONING might include three enemies closing in for melee attacks while two more hang back and use ranged attacks from a distance of 5 or 10 meters. Similarly, the player may rush in for her own melee attacks or focus on evasion and ranged attacks.

During all this action, the player should have a good view of her character and all nearby enemies so that she can see incoming attacks, plan who to attack next, and so on. To achieve this, we continuously adjust our focal position to stay centered on the combatants, and we

continuously adjust our offset distance, zooming out as necessary to keep all combatants inside the view frustum.

Figure 5 illustrates our method. In pane 1, the player has encountered a little gremlin and a snake-like critter. Pane 2 shows how we adjust focal position, and pane 3 shows focal position adjustment combined with zooming out. I'll explore the details of this method in the next two sections.

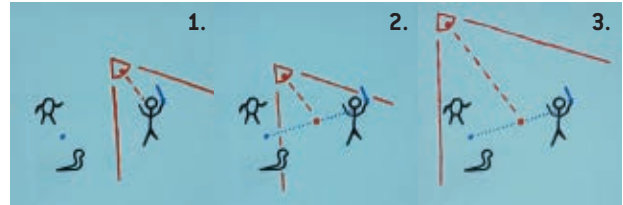


Figure 5: Framing combat.

## Centering Combatants

/// Previously, I stated that the focal position generally follows the player character. When not in combat, his position is the input to our earlier smoothing function. In this sense, the player character is our target focal position.

During combat, we adjust this target focal position to center on the combatants. First we compute a threat position; an average of all these enemies' positions (shown as a blue dot in Figure 5). The target focal position is then a blend between the player character position and the threat position, using fixed weights, perhaps 60/40 toward the player character.

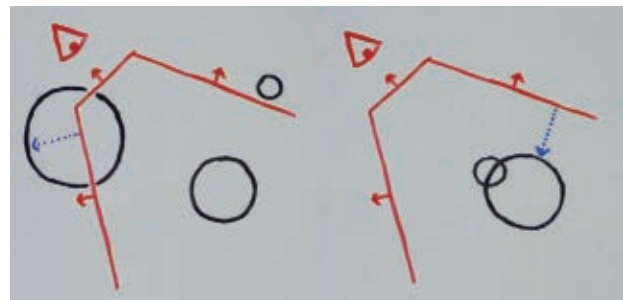
To avoid moving the threat position abruptly when new enemies appear, we assign each enemy a fade weight. The threat position is then a weighted average of enemy positions that changes gradually as enemies enter (fade in) and leave (fade out).

The threat position itself is also given a fade weight, equal to the highest fade weight among all enemies. So, as the first enemy appears and combat begins, the target focal position moves gradually from the player character position to the 60/40 blended position (rather than snapping).

## Keeping Everyone On-screen

/// Although our focal position stays centered on combatants, we may still need to zoom out to fit them all in the view frustum. Every combatant has a set of bounding spheres approximating his shape. Each frame, we compare all spheres to the near and four side planes of the camera frustum. We find the outermost distance in world space.

Figure 6 shows the camera frustum's near and side planes in red (including plane normals). Three spheres are scattered inside and outside the frustum. The outermost distance (represented by a dotted blue line) is between the largest sphere and one of the side planes. The distance is positive (in the same direction as the plane normal), so we should respond by zooming out. In Figure 7, two spheres are grouped inside the frustum and the outer-



Figures 6 & 7: Frustum and combatant bounding spheres.



most distance is negative. This indicates it's safe to zoom in to frame the combatants more tightly.

We prefer not to perfectly bound all spheres each frame, as this zooming behavior isn't smooth. We instead zoom in or out steadily over time. For outermost distances near zero, we make no adjustment at all, to avoid oscillation.

Our framing requirements can change quickly and drastically. An enemy can enter combat at a position far outside the frustum; or, the player can manually yaw and pitch the camera to change his view of the battle, so that previously onscreen enemies now lie outside the frustum. These cases are handled well because we always zoom gradually.

Thus far I've discussed framing combat and smoothing camera motion, but I've definitely saved the best for last. I'll devote the remainder of this article to our hardest problem: obstacle avoidance.

## Obstacle Avoidance

/// After our various camera subsystems have updated focal position, yaw, pitch, and offset distance, each current frame will finally have a new, desired camera position. However, this new position may be inside a boulder, or the camera may have lost sight of the player character around a corner or through a doorway. How can we coax our camera around these obstacles and maintain line of sight?

As a programmer, there are problems that make you want to bang your head against your whiteboard, and it turns out this is one of those. In the course of developing a viable obstacle-avoidance-algorithm, there were pitfalls, and we certainly toiled in those dark places! These next sections will retrace our path from the initial problem to our eventual solution.

For obstacle avoidance, we're concerned with that crucial sightline between the camera and the player character. The previously defined focal position and offset vector are less relevant. This distinction is most meaningful during combat, when our focal position tends to stray from the player character due to our combat framing. In these next sections, I'll still be using terms like yawing and zooming, but these will now refer to character-centric camera movements.

## Collision Zoom

/// RECKONING's fantasy world is filled with magic and danger, but one thing you don't have to worry about is getting teleported inside a boulder. Indeed, our camera code can safely assume that the player character is never inside any obstacles. So one solution for obstacle avoidance is to just zoom in toward the character, past any obstacles.

Using our physics library, we cast a ray backward from the character to the desired camera position. There may be multiple obstacles, but this cast's first hit will be the obstacle nearest to the character. We move the

camera position to the hit point to resolve all obstructions.

This simple collision zoom method is useful, but handles cases like Figure 8 poorly.  $P_0$  and  $P_1$  indicate the player character's movement over time. As he walks past the corner of a building, the camera, shown as a red eye, trails behind and abruptly pops to the nearest building wall. In this case, we'd prefer that the camera yaw about the character to maintain line of sight—I'll call this collision yaw.

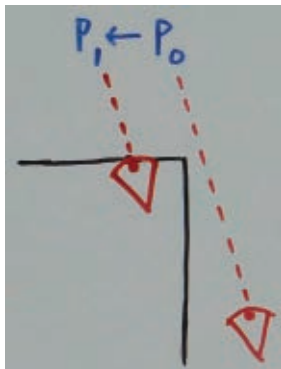


Figure 8: A collision zoom pop.

## Collision Yaw: Early Failures

/// Unfortunately, a single raycast doesn't give us enough information to properly yaw away from nearby obstacles. One early approach was to binary search for a clear yaw angle using many raycasts, but this was too inefficient.

Most physics libraries support a variety of queries beyond raycasts, including shape casts, queries to find penetrating shapes, and queries to find closest points. We experimented with these as well, but ultimately we couldn't get all the information we needed about nearby collision geometry.

We even tried simulating a rigid body using our physics library. We constructed a long, thin collision beam to represent our sight line. Constraining the end of the beam to the player character, we hoped it would robustly yaw and pitch about the character to avoid obstacles. Despite much tuning, this approach was unsuccessful.

## Caching Collision Geometry as a Point Cloud

/// Eventually, we stumbled upon a solution that would prove very powerful and versatile for collision processing: a point cloud. This is simply a flat list of 3D points that approximates the collision geometry near the sight line. In essence we're caching nearby collision geometry and converting it from the physics library's internal representation into something more convenient.

Figure 9 shows the player character in a t-shaped hallway. The cloud's borders are a large capsule (blue) that bounds the sight line, with 3 or 4 meters of padding. The cloud already contains several points (green) and we're attempting to add three more via random raycasts (orange). The first cast hits a wall and a new point is added. The second cast reaches the edge of the capsule without hitting anything. The third cast hits, but would generate a new point too close to an existing point, so we discard the result. Our point spacing (currently 25 cm) is a tradeoff between collision accuracy and performance.

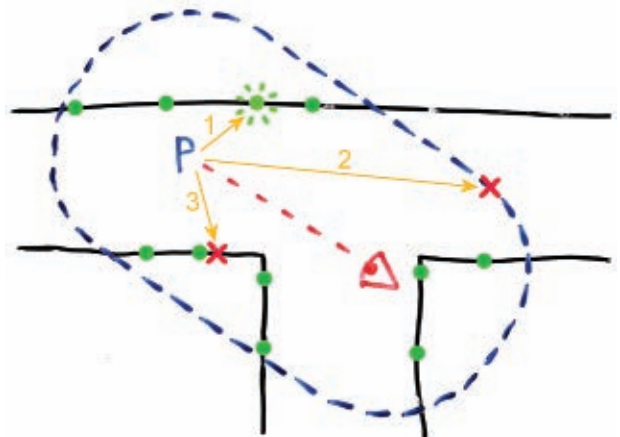


Figure 9: Building a point cloud.

We don't re-create the cloud from scratch every frame. Because RECKONING's collision geometry is primarily static, it's safe to keep and reuse old points found in previous frames. However, the bounding capsule follows the player character through the world, so we'll eventually discard old points when the capsule moves away from them. To refill the capsule, we cast seven rays per frame. These can be batched and performed on a background thread for efficiency.

## A Collision Beam

/// With our point cloud now available for collision queries, we return to the idea of a collision beam—a long, thin box with a square cross-section that extends from the character to the new desired camera position. We can easily find all the points penetrating our collision beam: we simply loop through all points in the cloud and perform point-versus-box intersection tests.

If we only query for penetrations once per frame, we'll miss collisions when the camera is moving too fast. To avoid this we use sub-steps. We iterate from the previous frame's beam position to the current frame's new desired beam position using a small step size. To resolve penetrating points at each time step, we'll work in the 2D ground plane, treating the collision beam as a rectangle.

An early naïve approach of ours was to completely eliminate penetration by yawing. Figure 10 shows our collision beam (red) and several collision points (green) approximating the corner of a building. Although the approximation slightly misses the building corner, this small inaccuracy won't be a problem.

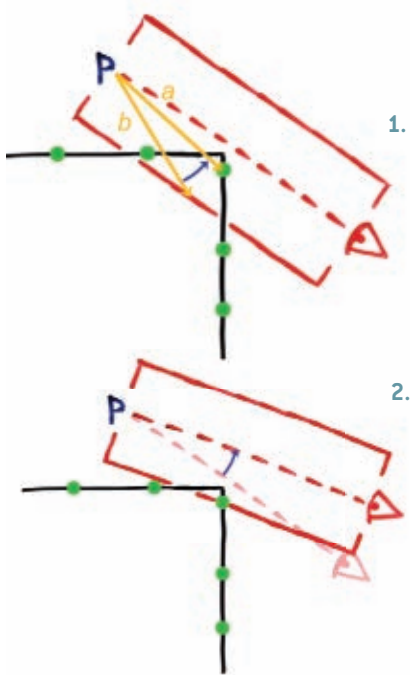


Figure 10 [Pane 1 & 2]: Resolving penetration of our collision beam.

In pane 1, we start with vector *a* extending from the character position to the deepest collision point. We construct vector *b* of equal length; it extends to the side of the beam. Between *a* and *b* is our penetration angle (blue). In pane 2, we yaw the camera around the character by this angle to eliminate the penetration. Listing 1 shows our penetration angle calculation.

### LISTING 1 CALCULATING THE PENETRATION ANGLE OF A COLLISION POINT

```
// this function computes the z component of the cross-product
// between
// these two vectors as if they were 3D.
float vector2_cross ( const Vector2& src_a, const Vector2& src_b )
{
    return ( src_a.x * src_b.y ) - ( src_a.y * src_b.x );
}

float calc_penetration_angle( const Vector2& point,
    const Vector2& character_pos, const Vector2& beam_axis )
{
    // note: beam_axis should be normalized
    Vector2 character_to_point = point - character_pos;
    // the signed distance from the point to the beam axis
    float point_dist_from_axis = vector2_cross( character_to_point,
        beam_axis );
    float hypotenuse_len = length( character_to_point );
    float beam_edge_dist_from_axis = ( point_dist_from_axis > 0.f )
        ? COLLISION_BEAM_HALF_WIDTH : -COLLISION_BEAM_HALF_WIDTH;
    // our arcsine calls are based on two right triangles, each with
    // character_to_point as the hypotenuse.
    float angle_from_curr_beam_axis = asinf( point_dist_from_axis /
        hypotenuse_len );
    float angle_from_corrected_beam_axis = asinf( beam_edge_dist_from_
        axis /
        hypotenuse_len );
    return angle_from_curr_beam_axis - angle_from_corrected_beam_axis;
}
```

### LISTING 2 THE SLIDE ANGLE OF A COLLISION POINT IN ONE TIME STEP

```
float calc_slide_angle( const Vector2& point,
    const Vector2& character_pos_t0, const Vector2& character_pos_t1,
    const Vector2& beam_axis )
{
    // note: beam_axis should be normalized
    Vector2 character_to_point_t0 = point - character_pos_t0;
    Vector2 character_to_point_t1 = point - character_pos_t1;
    // the signed distance from the point to the beam axis at t0
    float point_dist_from_axis_t0 = vector2_cross( character_to_point_
        t0,
        beam_axis );
    // clamp our signed distance from the beam axis to be no greater
    // than
    // COLLISION_BEAM_HALF_WIDTH in magnitude.
    point_dist_from_axis_t0 = bhmin( COLLISION_BEAM_HALF_WIDTH, bhmax(
        -COLLISION_BEAM_HALF_WIDTH, point_dist_from_axis_t0 ) );
    float point_dist_from_axis_t1 = vector2_cross( character_to_point_
        t1,
        beam_axis );
    float hypotenuse_length = length( character_to_point_t1 );
    // our arcsine calls are based on two right triangles, each with
    // character_to_point as the hypotenuse.
    float angle_from_curr_beam_axis = asinf( point_dist_from_axis_t1 /
        hypotenuse_length );
    float angle_from_corrected_beam_axis = asinf( point_dist_from_
        axis_t0 /
        hypotenuse_length );
    return angle_from_curr_beam_axis - angle_from_corrected_beam_axis;
}
```



How can this approach resolve more complex cases involving multiple penetrating points, you may ask? In Figure 11, the player character will back into a wall, forcing the camera into a corner. In Figure 12, the player character has passed through a doorway and will now move right, trapping the camera in the doorway.

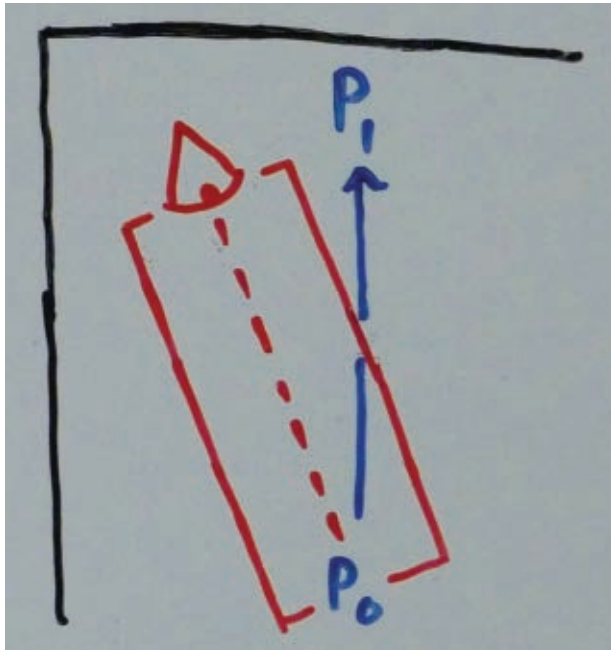


Figure 11: Backing the camera into a corner.

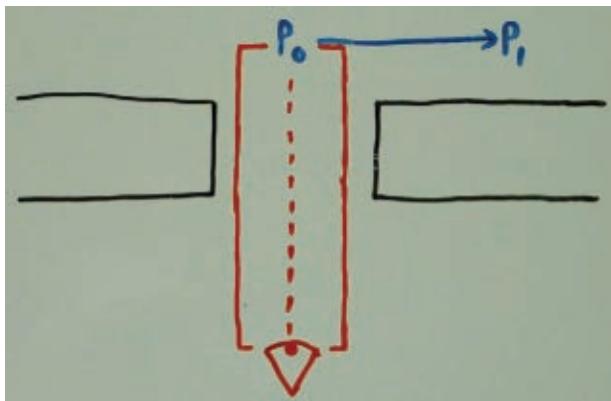


Figure 12: Trapping the camera in a doorway.

In each of these cases, our point cloud contains many collision points approximating the walls of these environments. As these points begin to penetrate the collision beam, our naïve approach attempts to eliminate all penetrations by yawing, but this is clearly impossible. Whether trapped in a corner or trapped in a doorway, the camera is likely to jitter wildly left and right or perhaps make a huge, disorienting jump to escape the confining space.

### A Better, Stickier Collision Beam

/// To avoid overreacting in cases like Figures 11 and 12, we revise our original naïve approach. For a collision point that was already penetrating the beam on the previous step, we don't try to completely eliminate its penetration. We only try to prevent its penetration depth from increasing.

The resulting behavior is almost as if the beam is sticky—collision points inside it become stuck, and the beam tends to rotate about any stuck point as the character moves.

Figure 13 illustrates this approach for a single collision point. Pane 1 shows player character movement over a single time step. The collision point, already penetrating the beam at  $t_0$ , is now deeper at  $t_1$ . Two orange arrows highlight these depths. Pane 2 shows the same time step in the collision beam's frame of reference (local space). Here, it's easier to see how the collision point is moving deeper into the beam.

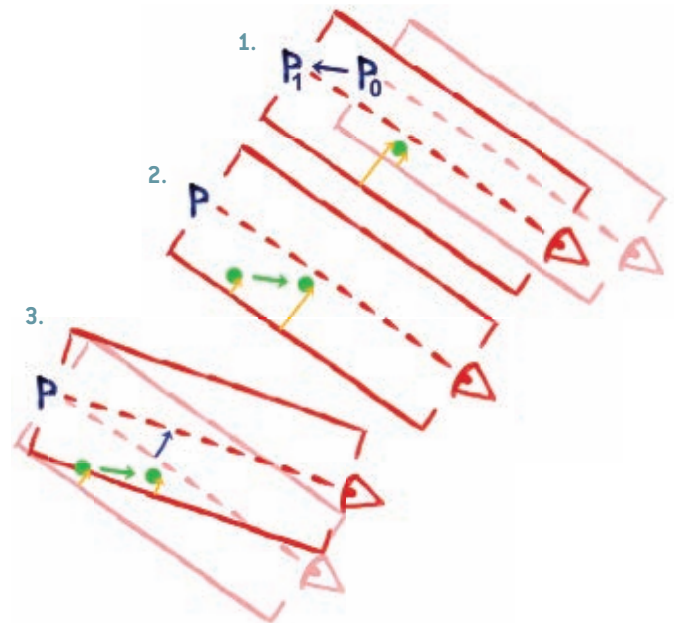


Figure 13 (Panels 1, 2, and 3): Our sticky collision beam rotates about a stuck point.

Finally, in Pane 3, we yaw the camera counterclockwise by a certain angle to restore the collision point's original penetration depth from  $t_0$ . I call this the slide angle, and the calculation is shown in Listing 2.

For a collision point that wasn't already penetrating on the previous step, the slide angle is simply equal to the penetration angle. In the general case, our algorithm considers all penetrating points so that we find the largest clockwise (CW) and counterclockwise (CCW) slide angles. For more difficult cases like Figure 11 and Figure 12, there are conflicting CW and CCW slide angles. Our compromise is to let the smaller angle partially cancel out the larger one; that is to say, a large CW slide angle conflicting with a small CCW slide angle will be resolved by a small CW yaw.

With this method of collision yaw, we aren't guaranteed to completely eliminate penetrations. Thus, we also do collision zoom immediately after doing collision yaw. In addition, collision yaw shouldn't interfere when the player is actively controlling yaw. At these times, we disable collision yaw but still employ collision zoom.

### Obstacle Avoidance in Action

/// Enough with the drawings and theory, it's time to check out the behavior of our algorithm in some actual environments! The next few screenshots are from Havok's Visual Debugger. They show RECKONING's world collision geometry (gray and black) and the player character (a blue circle). The character's recent movement is shown as a light blue path. Each screenshot includes a time sequence of collision beams (dark red). For each time step,

the final camera position (pink) usually lays at the end of the collision beam. However, recall that we do collision zoom after collision yaw and that sometimes zooms the final camera position toward the character.

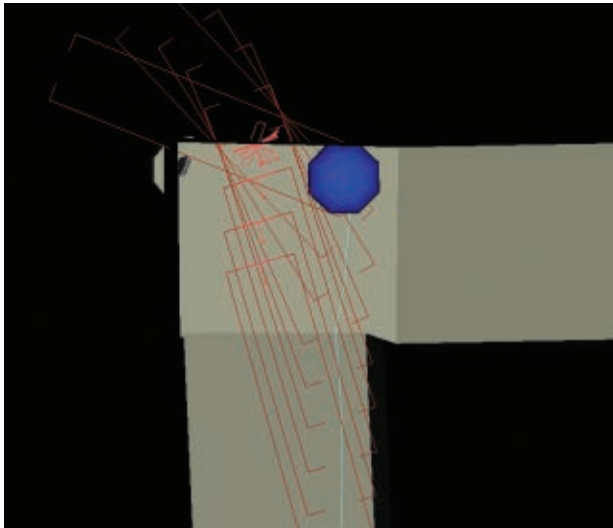


Figure 14: Backing the camera into a wall.

Figure 14 shows our first hard case from earlier. As the player character backs into a wall, our collision beam sticks to the wall and rotates.

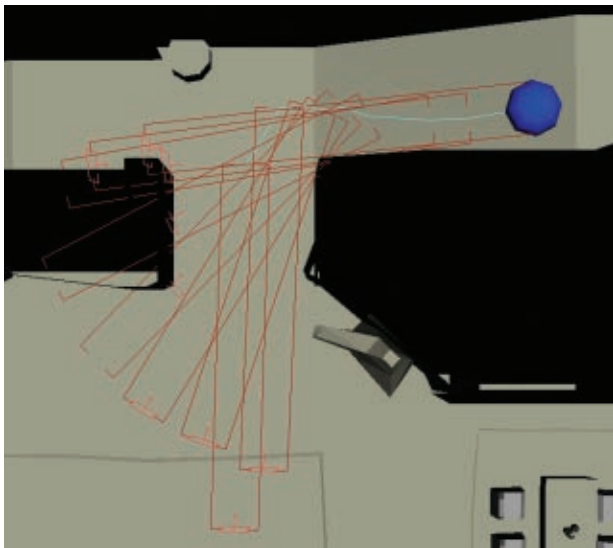


Figure 15: Moving through a doorway.

Figure 15 shows our second hard case. The player character moves down a hallway and our camera yaws and zooms to get through the doorway. As the collision beam becomes squeezed by both sides of the doorway, our compromise behavior for the conflicting slide angles works well. Note the zoom pop as our collision zoom method's raycast catches the left side of the doorway. This is not ideal, and is a subject for possible future work.

Finally, Figure 16 shows a more complex situation. As the player character walks from one room to another, our camera first sticks to a wall. Then it rotates free to escape the corner and follow through the doorway.

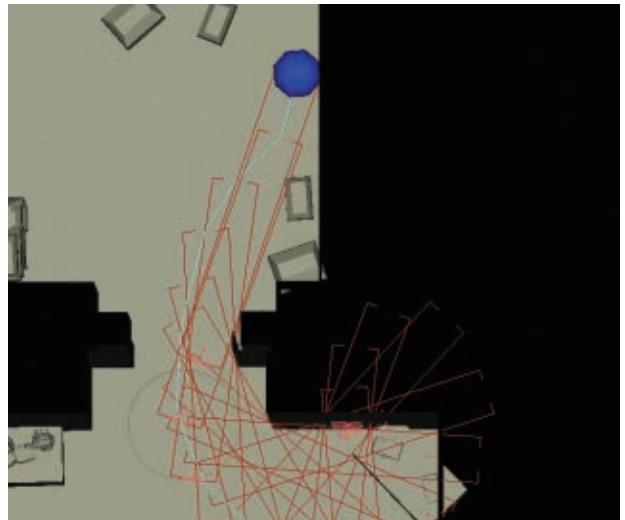


Figure 16: Moving between rooms.

## Future Work for Obstacle Avoidance

/// Going back to the undesirable zoom pop in Figure 15, we've had some success with an approach I'll call predictive zoom. As collision points approach our sight line from just one side, we can rely on collision yaw to avoid them. But when they approach from both sides as in Figure 15, we need to zoom in gradually and preemptively to avoid a zoom pop.

Using a second, much wider collision volume (perhaps several meters), we can indeed detect this situation and respond. The ideal result is that the camera gradually zooms in as the player character moves into confining spaces. We'll continue investigating this in the future.

Another avenue for future work is 3D obstacle avoidance—yaw and pitch. Imagine that the camera is pitched high above the character as he passes under a low ceiling rafter. As the rafter penetrates the collision beam from above, collision yaw ignores it and we get an undesirable collision zoom pop. In the future, it should be feasible for us to implement a collision pitch algorithm that computes vertical slide angles and pitches the camera down to avoid the rafter.

## A Few Last Words

/// RECKONING's third-person camera was a collaborative effort at Big Huge, with Joe Quadara scripting, play testing, and generally doing the things designers do, while I focused on implementing his ideas (and occasionally mine) in code. I hope fellow coders out there will find my discussion of moving averages, point clouds, and sticky collision beams useful. ☺

ERIC UNDERSANDER is an engine programmer who most recently worked for Big Huge Games on KINGDOMS OF AMALUR: RECKONING. He welcomes your comments at [undersander@gmail.com](mailto:undersander@gmail.com).

## references

- Moving Average [http://en.wikipedia.org/wiki/Moving\\_average](http://en.wikipedia.org/wiki/Moving_average).
- Smooth function [http://en.wikipedia.org/wiki/Smooth\\_function](http://en.wikipedia.org/wiki/Smooth_function).
- "Real-Time Cameras," Mark Haigh-Hutchinson, 2009.
- "Iterating on a Dynamic Camera System" Phil Wilkins, GDC 2011.
- "An AI Approach to Creating an Intelligent Camera System" Phil Carlisle, AI Game Programming Wisdom 2, 2003.
- "Practical Game Design Blog" Joe Quadara and Patrick Connor, [www.practicalgamedesign.com](http://www.practicalgamedesign.com).



**PUBLISHER** TimeGate Studios  
**DEVELOPER** TimeGate Studios  
**RELEASE DATE**  
April 20, 2011 (XBLA), May 4,  
2011 (PC), July 26, 2011 (PSN)  
**NUMBER OF DEVELOPERS** 90  
**LENGTH OF DEVELOPMENT**  
1 year, 10 months  
**LINES OF CODE** Over 2 million  
**PLATFORM** Xbox LIVE Arcade, PC,  
PlayStation Network  
**NUMBER OF TIMES PEOPLE  
YELLED "THAT'S WHAT SHE  
SAID":** 24,681





# section 8

# prejudice

A D E L C H A V E L E H

////////// After wrapping up SECTION 8 in October 2009, the team at TimeGate felt there was a lot of potential left to explore in both the gameplay and universe. With so many great ideas still buzzing about, we knew we couldn't let them all go to waste. We also had a seasoned FPS dev team that had spent the better part of two years building the technology for the largest game our studio had ever made. This is where SECTION 8: PREJUDICE was born.

During pre-production, we took a look at old and new ideas, as well as feedback from our developers, community members, and reviewers, to figure out how we were going to up the ante on the next installment. With both the experience and the tools in place to make a better game than its predecessor, we set out to add more modes, maps, weapons, and vehicles, as well as update the mechanics to make a far superior sequel.

We knew we had a huge challenge in front of us, because not only were we going to develop the game, we were going to publish it on our own. As the developer and publisher, we had control over the game as well as the messaging. In the end, we took a game that had the same amount of content you would find in a \$60 retail title, dropped the price to \$14.99, and sold

it through digital outlets such as Xbox LIVE Arcade, Steam, and PlayStation Network.

We knew it was going to be a challenge to change players' perceptions of what to expect from an arcade title, but in the words of Barney Stinson, "Challenge accepted."

## W H A T      W E N T      R I G H T

### 1 focus testing

/// Immediately following the release of the original SECTION 8, the team outlined an internal postmortem that defined what we needed to do differently with SECTION 8: PREJUDICE. One thing we wanted to do was integrate focus testing earlier. Thanks to the benefits of the technology and team being carried over from SECTION 8, we knew we'd have something testable much earlier, and we had to leverage that advantage to its fullest.

During the focus-testing sessions, we looked at how players interacted with bots, how they used vehicles, what weapons they used, and every other possible feature. Our tutorial level was tested heavily to figure out if players were

>>>

quickly picking up on our core mechanics, such as jetpacks. This resulted in us finding several key usability problems that might have otherwise been completely missed. In one important but subtle example, we found players were having usability problems with grenades. The core flaw was that grenades spawned in the player's left hand, and whenever a player tried to peek around the right corner of a wall to throw a grenade, the grenade would often bounce off the wall and right back into their face. To fix this problem, we moved the spot where the actual grenade projectile spawned closer to the center of the player's body, which gave it much better corner clearance.

Campaign difficulty benefited significantly from focus testing, as we noticed that players died frequently during the first few levels of the campaign, but less so as the campaign continued. This information allowed us to make targeted tweaks to difficulty, lowering the challenge in the first few missions, while ensuring that the last few levels didn't turn into cakewalks.

We even focus tested the name SECTION 8: PREJUDICE to gauge interest in the title. We were happy to discover the majority of testers liked the boldness of the name. Ultimately, it was focus testing, too, that led us to believe that there was a strong untapped opportunity looming in the digital space. This was a catalyst for us to seriously consider distributing the title exclusively digitally.

## 2 tech art

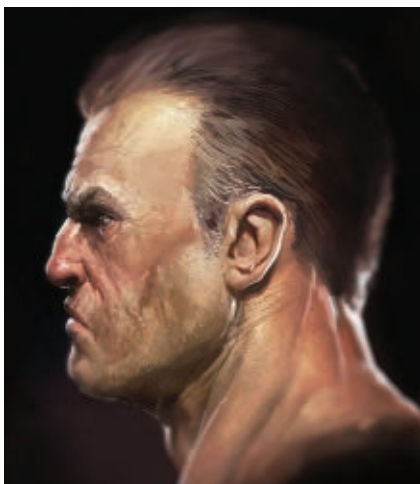
/// One of the big takeaways from SECTION 8's development was that there was a communication gap between the programming, level design, and art departments on the visual design of new features. In addition to deciding how we wanted a feature to appear in-game, we needed a better process for how we would accomplish that target visual quality without breaking our performance and memory budgets.

As we ramped up to make SECTION 8: PREJUDICE, we empowered an internal technical art team to help bridge this gap and streamline the tech-art pipeline. This involved meeting with different department leads to determine the art requirements for new game features, such as weapon variants, third-person dropping, and vehicle damage feedback. The tech art team worked closely with the gameplay, engine, and graphics programmers to prototype new features like global illumination, ambient occlusion, and texture streaming. They then trained the artists to leverage these new features in the most efficient way.

Once an art asset is imported into the game engine, there are many settings the content developer has to tweak. Many of those settings can dramatically affect the game's performance or memory in non-obvious ways, such as flagging a light as a dynamic shadow caster or flagging a weapon effect as looping. The tech art team

identified many of these high-risk content settings, and worked with the tools programmers to implement a validation tool that would review all assets and report anything questionable. This helped us quickly catch new issues and prevent many obscure and subtle bugs.

We incorporated the tech art team into the level pipeline, and had them leverage QA for more aggressive performance and memory tracking. QA would stress test maps on a weekly basis to find framerate and memory hotspots, documenting any new issues in our task tracking software. Tech art would then diagnose those hotspots and work closely with the art and level design departments to implement the optimizations. Ultimately, all this enabled us to have significantly improved visuals without harming our framerate.



## 3 usability

One of the biggest points of negative feedback from the original SECTION 8 was that the game's learning curve was much steeper than that of a typical shooter. We knew that we would need to address as many usability issues as possible in SECTION 8: PREJUDICE, and focus on smoothing out that first hour of gameplay for new players.

While the SECTION 8 single-player campaign offered a competent basic tutorial, it was not particularly entertaining, and many players skipped it entirely. For SECTION 8: PREJUDICE, we revamped our tutorial system and rewrote nearly every tutorial hint from scratch. We also put a much bigger emphasis on creating an entertaining tutorial in the first mission of the campaign, something that would really teach players how to use most of the game's core mechanics. We made constant iterations on the tutorial portion of the level, churning through many different layouts in an attempt to keep the tutorial experience as short and effective, as possible. We knew we didn't want to laboriously teach every possible aspect of the game, so we focused on a small

subset that would enable the players to fight and move effectively through any level. The goal of the tutorial was ultimately realized in an obstacle course that required players to use all of our core mechanics, especially the jetpack and overdrive function.

Interface was another sore point from SECTION 8. Although we felt that our interfaces conveyed a lot of useful information, it was information overload for new players. Not only did we improve usability in the sequel by moving some of the more complex elements to subscreens, but we also made some hard decisions to revise gameplay mechanics that had obtuse or complex displays. In particular, the Dynamic Combat Mission system (DCM) underwent several revisions that improved its interface usability. The most obvious example is in pacing; in SECTION 8, you would sometimes have four active DCMs in larger games. This resulted in an explosion of objective icons on your HUD, and this was very intimidating to new players. We altered DCM pacing in SECTION 8: PREJUDICE such that there is normally only one, or sometimes two, DCMs active at the same time. This removed a lot of HUD icons from our worst-case interface scenarios. In the end, our efforts to improve the usability had a positive impact on the new-user experience.

## 4 unlock system

/// One of the significant new features we included in SECTION 8: PREJUDICE was a substantial weapon and equipment unlock system. This presented a large number of technical, memory budget, and gameplay challenges, but was one of the biggest threads left over from SECTION 8 that we wanted to pursue in our new game.

The largest and perhaps most intimidating hurdle to developing unlocks was the memory implications of all of the new content. With SECTION 8, we had pushed the technical limits of the consoles in a number of ways, and deciding to cram in a bunch of new weapons and equipment was going to make our tech team cry. But we knew we wanted it, and room had to be found, so we set our engine and tech art teams scouring through the code to find how to wiggle out enough room to add a much larger number of weapon and equipment unlocks into the game. The teams scrimped and saved at every possible opportunity to pull this off, reclaiming memory through optimizations, better compressions, and simply redesigning content to use less memory. One big area of savings came from redesigning a few of our weapon models. We found that a number of weapon models, especially the first-person ones, were too mechanically complex to be very memory efficient. If any fans wondered why some of the assault rifles and machine guns changed their reload mechanics between games, it's because it allowed us to cut their vertex memory cost in half!

# MAKE YOUR GAMES MORE ENGAGING, ADDICTIVE AND PROFITABLE



2011  
**STORYWORLD**  
conference + expo  
san francisco • oct 31-nov 2

Learn directly from the game developers and other media pros who are using transmedia techniques to get their fans fully hooked! At StoryWorld Conference + Expo, major creative minds like ARG pioneer **Steve Peters of Fourth Wall Studios** (*Six Minutes to Midnight; Why So Serious; Year Zero; and Dead Man's Tale*) will show you how to avoid potential pitfalls and succeed at cross-platform entertainment.

Use your exclusive Game Developer discount code GDSW11 to save \$50 off full conference registration. **Combine your discount code with Early-Bird savings by August 31 to save \$250!**



**SIGN UP NOW AT [STORYWORLDCONFERENCE.COM](http://STORYWORLDCONFERENCE.COM)**

**THE EDUCATION EXPERTS IN  
GAMES, 3D & VFX**

**NEW US CAMPUSES  
OPENING AUGUST 2011!**

**[www.theaie.us](http://www.theaie.us)**

As one of the first educators in the world to offer games-specific qualifications way back in 1996, the Academy of Interactive Entertainment knows how to get graduates into games, animation and VFX.

More Info: 206-428-6350 OR 225-288-5221



**AIE** ACADEMY OF  
INTERACTIVE ENTERTAINMENT



A second major issue was the sheer amount of new art and audio content we'd need for the unlocks. In order to both minimize the cost of the new content and minimize its memory footprint, we decided against doing completely new models for each of the unlocks. Instead, we would portray the weapon variants as "round" or "ammunition" variants. These variants could still have different gameplay implications (damage, accuracy, special effects), but would only require new firing and impact effects instead of entirely new models. Game designers worked diligently with visual FX, audio FX, and gameplay programming teams to create a content toolset that provided a robust variety of potential unlocks.

In the end, we created four different variants for each of the weapons, and three to four variants for each piece of equipment. Gameplay effects for each of these variants varied from rather simple changes in firing patterns (e.g., burst fire) to complex changes that would leave semi-persistent fires on the map. We had a number of unique effects, ranging from EMP energy drains to concussive slowing effects, and this left us with a large number of viable loadout combinations.

### 5 swarm

/// SECTION 8 was primarily a multiplayer experience, firmly focused on our competitive Conquest game mode. We support bots in all our multiplayer game modes, which allows us to

keep servers filled to capacity with competent AI opponents. This led us to develop several cooperative Conquest variants where players could team up against a larger team of bots. We named the mode Swarm, and it turned out to be an instant hit amongst the team and fans alike. During our post-launch support for SECTION 8, our community manager hosted a "Swarmament Day" tournament for the community. We had community teams competing against one another, playing on some of the hardest maps against super-stacked teams filled with the hardest-difficulty bots. It was a great experience, and it further cemented that Swarm needed to play a larger role in our next game.

So when it came time to introduce a new game mode in SECTION 8: PREJUDICE, the first thing we set out to include was Swarm as an official game mode (rather than a Conquest-derived mode). This would allow us to do official matchmaking for the mode, as well as implement far more custom rules for it than we had access to in SECTION 8. To win Swarm, players would have to fight against waves of enemies that would ramp up in difficulty and complexity as the game progressed. While we had to leverage our existing multiplayer maps for the mode, we found ourselves making a number of map-specific updates to make the Swarm variant play out better. We added elevators for enemy waves, customized each wave's spawn location, and more, making it better and better with each iteration. We already

had a system for dynamically deploying turrets on the map from SECTION 8, and this gave our Swarm mode a fun tower-defense subtext.

By the end of the project, Swarm felt very different from Conquest and our campaign. It played to the strength of our deployable system, and it leveraged much of the map and enemy content we had created for our other modes. Most importantly, it was just plain fun, and testers and developers alike were having a blast battling against the swarms of Arm of Orion enemies.

## WHAT WENT WRONG

### 1 game thread performance

/// While we've just talked about how smoothly our tech art process went, we did have one major technical issue that threw a wrench into much of our development. As we added more complex code and script to our variant of Unreal Engine 3, we were inadvertently adding a lot more overhead to a key performance metric: the game thread. Simplified, the game thread is a measure of CPU horsepower, and while it's a non-issue on any modern PC, it became a significant issue on consoles.

We eventually caught the problem, but not early enough. Our tech and engine teams were initially much more focused on graphic performance bottlenecks, which they did a great job solving, but the growing game thread performance issues flew under our radar for some



time. About halfway into development, we noticed that the consoles were struggling whenever we had what we felt was an average number of bots active. After some diagnosis, the problem was obvious, but the solutions were not. We were going to have to rewrite portions of our engine, as well as optimize several very core systems to get back the necessary milliseconds to hit our target framerate.

It was a slow, expensive process that weighed on the team during development. Campaign developers were given estimates for the number of bots they would eventually be able to leverage in the campaign, but it was rough going. Testing on consoles was very difficult for several months, as the game dragged whenever we had much action on the screen. This made it hard to gauge the true experience, and we had to revert to PC testing in many cases to temporarily work around the issue.

Near the end of the project, most of the core problems were worked through, but we still had to make several design changes in order to hit our target performance numbers. The campaign and Swarm were hit the hardest, as we had to lower our target number of simultaneously active bots to fit under the performance cap. While we didn't feel this had much of a noticeable impact on Swarm, it resulted in some of our battles in the campaign feeling less "epic" than the designers had initially targeted.

Credit to our developers is owed, though, as they stuck it through and managed to reclaim vast portions of game thread performance. Our biases from SECTION 8 helped us focus on solving some key issues early, but it also may have hindered us from noticing the new threat looming on the horizon. We've learned that we need to properly monitor all potential performance bottlenecks as the project evolves, not just the current hot-button items.

## 2 server stress testing

/// After SECTION 8: PREJUDICE came out of the beta phase, our QA team began the process of stress testing how the game ran on servers for both PC and consoles. Unfortunately, we ramped down our QA staff too soon, which left us with insufficient testers to internally test 32-player matches. With our smaller QA team in hand, we weren't sure how the game would perform in a live environment because we weren't testing at maximum capacity.

As a result, we didn't get much real stress testing going on our servers until later in development. We could still run bots to fill in for the missing players, but what we were not seeing were issues caused by networking and server stress. Games that would run smoothly with six human players and 10 bots would run poorly with 16 human players. Complicating this was that we were running many of our servers locally at TimeGate. This made maintenance and monitoring of the servers easy, and we were still

playing online during playtests, so we felt we had our bases covered. The base we did not have covered is that most server hosting companies have a lot more per-machine stress than what we were simulating at TimeGate. A hosting company doesn't just host one server per box; they're often hosting multiples, and this affects how your servers perform in a number of difficult-to-simulate ways. As we started playing on external servers (around beta), we saw a new breed of latency and server quality issues arise that had not been seen during internal playtests.

Once again, the tech triage team was called upon and identified several issues. Where we had expected a small delta between the cost of human player and the cost of a bot player, a much larger gap was discovered. The issues were solvable, but when you're near the end of a project and you suddenly discover more tech performance issues, it hurts. This of course led to some late-project stress, but we eventually plowed through the issues.

Both SECTION 8 and SECTION 8: PREJUDICE had significant online functionality that most shooters do not: a proprietary stats portal, 32-player servers across all platforms (40 on the PC), the ability for users to host dedicated servers for all platforms, and a web-based clan management system. Getting this proprietary tech to this point

which caused it to completely freeze up. Because the crash was infrequent, and we had already passed certification with Microsoft, it didn't seem like a problem to worry about; plus we would fix the issue with our first title update.

Everything was going smoothly as we neared launch, and two weeks before the game released we set up play sessions with various press outlets to simulate a live 32-player online multiplayer experience with our team. Unfortunately, we saw the crash rear its ugly head several times unexpectedly.

Our programmers and QA team worked round-the-clock trying to figure out what caused the crash, and unfortunately, we weren't able to figure out a solution before the game released. After poring through lines of code, it turned out the cause of the crash was a fix we had put into the engine to solve a relatively minor foliage-rendering issue. There was a rare flickering issue with some of our foliage that would happen at certain view angles, and our engine team had identified a solution and implemented it relatively late into development. The low-priority foliage problem was fixed, but we had inadvertently introduced a highly erratic hardlock on the Xbox 360. No reasonable amount of testing by the programmers or QA who vetted the initial fix would have spotted the crash, and it was a devastating stroke of bad luck.



has made us realize the necessity of testing the entire multiplayer component with a full human-player roster in a realistic server environment. Local testing with bots certainly helps speed up testing iterations, but it's no replacement for testing with humans in a real-world environment.

## 3 xbox 360 hardlock

/// A few months before SECTION 8: PREJUDICE was released, our QA team reported a rare but pretty harsh crash that would occur on the Xbox 360

It took several weeks to fix the bug and release the patch with our second title update. We are very fortunate to have a patient and loyal community, so as soon as the patch went live, we saw a dramatic increase in our player numbers on the Xbox 360. While the crash could not have been spotted by the programmers implementing the original fix, we ultimately could have caught the destabilization sooner if we had been able to do a higher volume of testing on builds immediately after the crash was introduced.

#### 4 retail-to-digital switch development implications

/// When we initially set out to make SECTION 8: PREJUDICE, our plan was to develop the game as a full retail title. Roughly halfway through development, we started evaluating various distribution models and came to the conclusion that going exclusively digital and pricing the game at \$14.99 would provide us the best means to get SECTION 8: PREJUDICE into as many players' hands as possible. While this was a great business decision that ultimately worked in our favor, digital titles pose different requirements than their retail counterparts.

Specifically, we had to figure out how we were going to fit what was expected to be a 4GB game into 2GB, per the size limitations for Xbox LIVE Arcade titles. We did some serious content reviews internally, and came to the determination that we would have to save at every possible step. Videos and audio would need to be compressed to save on size, and we were not going to be able to ship as many maps as originally targeted. As a result, several maps that we had hoped to finish and ship with the game were shelved. We revived several of our original designs as DLC, but the content developers that had invested a lot of hard work into those maps were not pleased to hear that they would not ship with the original game.

Apart from content cuts, we also had to refactor several reward systems that had assumed we were shipping a retail title with more maps. XBLA games do not get as many achievements as retail titles, and as such, we had to split the XBLA achievement list from other versions of the

game. This normally wouldn't have been a huge issue, but we had tied several of our unlocks to achievements that no longer existed, so we had to refactor several portions of our unlock tree to accommodate for fewer achievement-based unlocks. Compounding the reward issue was the fact that our offline progress system had been based on a separate "stars" system that assumed more maps would ship with the game than actually did. Offline players could earn access only to a limited amount of the unlockable content, forcing them to wait for the arrival of more DLC maps to earn the rest of the unlocks.

While the move to digital was ultimately a great business decision, it came after key design and architectural decisions had been made and implemented. The lesson learned here is a simple "identify your distribution medium as early into development as possible!"

#### 5 interface

/// Part of the process of making any game is listing all the features you want to include, and then paring down to what is going to be the most feasible for development and most fun for players. One of the areas we felt needed a big boost in quality from SECTION 8 was our interface, both the front-end menus and the in-game HUD.

As the game kicked off, we made the switch to UI middleware to improve the quality of our interface. As a result, we were able to do much slicker interfaces than we had been previously capable of, but it came with a higher manpower cost than we had anticipated. Ramping up on the new interface was slow, and it required dedicated

art and programming staff to leverage effectively. Our UI programming and art teams quickly found themselves overwhelmed, and we had to rapidly hire additional developers in an attempt to shore up the deficiency. As any developer will tell you, throwing people at the problem is never the best solution, but at the same time, we were under tight deadlines and we couldn't fail to ship the game simply because the interface wasn't going to be finished in time.

This is when the axe came down on one of the features that hurt the most to cut: party match. From the beginning of SECTION 8: PREJUDICE, we wanted to include a party match system that would allow groups of friends to easily join a game together. We completed the entire design for the party match system, and went so far as to finish all technical designs and visual mockups for the feature. But as the interface schedule fell further behind, we knew that it would take a major cut to get the team back on track. The axe fell, and party match was severed from the project.

There isn't a great lesson to be learned here, other than "predict the staffing implications of your technology decisions as early as possible." The only way we could have salvaged the party match feature was to have our interface team onboard the project sooner, working more efficiently, and getting through the core features in time to get party match into the game. Now that we have a few shipped games with the tech and a much more experienced UI team, future games will benefit from these previous happenings.

#### SECTION 8: PREJUDICE

SECTION 8: PREJUDICE was a great opportunity to create the game that both our fans wanted to play and we wanted to make. Even though we had to make some substantial changes to the game once we made the digital switch, the process went much smoother, as we already had the tools and an experienced team in place. While iterating on certain features can be a tedious and frustrating process, it ultimately helped us craft a much better game. Of course, we could not have done this without our hardworking, dedicated, and loyal team, to whom I owe a huge "thank you."

Ultimately, we are very proud of what we accomplished. We succeeded in creating a sequel that scored 10 Metacritic points higher than its predecessor. Also, as a 12-year-old studio, TimeGate has proven that we are not slowing down at all. After self-financing, self-publishing, and fully developing this product internally, we are more ready than ever for our next big challenge. 🍳

**ADEL CHAVELEH** is the president & CEO at TimeGate and executive producer for SECTION 8: PREJUDICE. When not making pancakes for the TimeGate team, he is reading your emails at [adel@timegate.com](mailto:adel@timegate.com)



MONTREAL  
INTERNATIONAL  
GAME  
SUMMIT

Nov.  
1&2

Hilton  
Montreal  
Bonaventure

# MONTREAL SHINES

The biggest event dedicated  
to professionals in Canada  
and on the East Coast

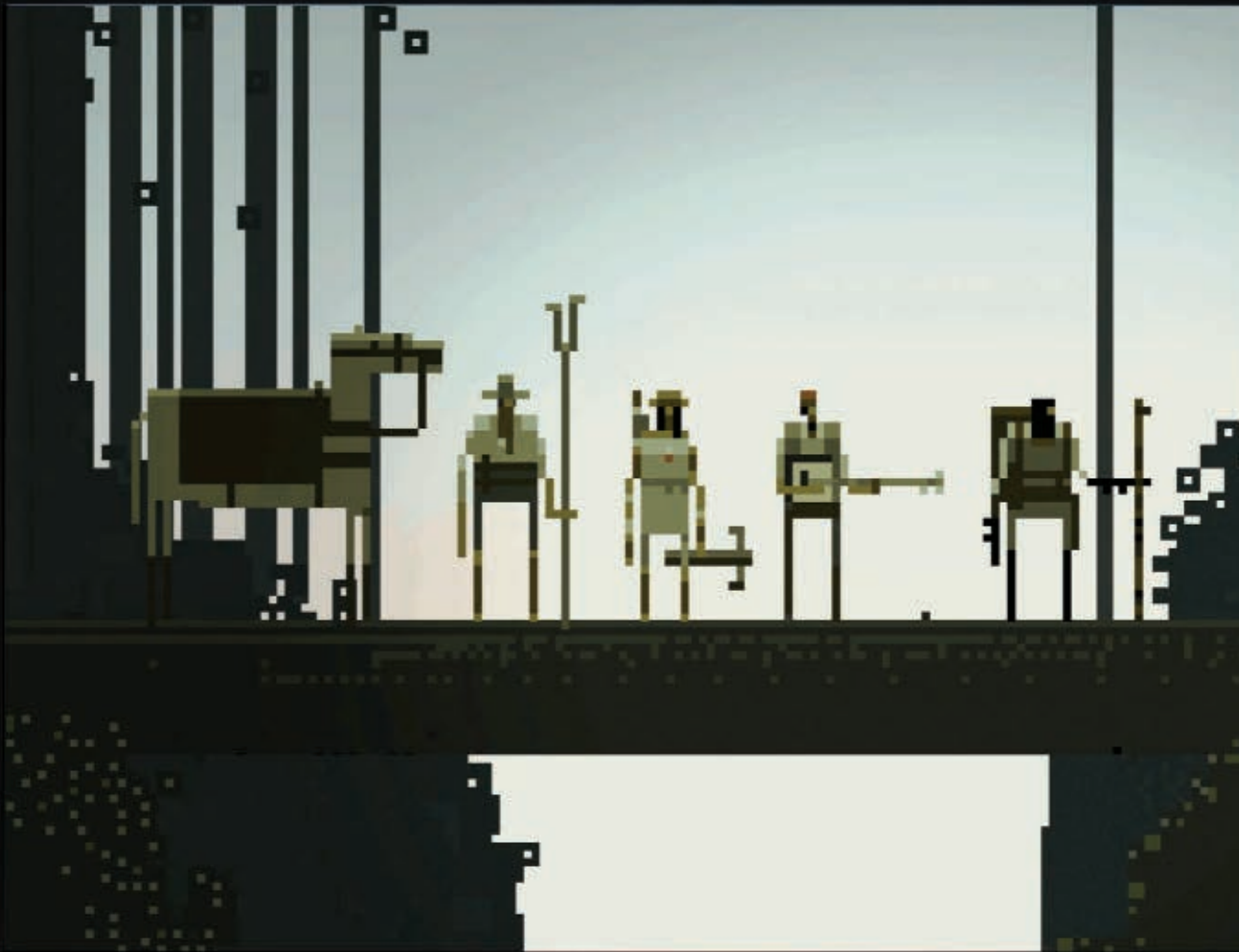
- > Over 70 internationally-renowned **speakers** and **keynotes**.
- > An **exhibition zone** to meet, discover and play.
- > A **Business Lounge** to deal and network.
- > Many **cocktails** and **activities**.
- > The first edition of the **Canadian Game Development Talent Awards**.

Register now and save up to 30% when you purchase  
your ticket before September 16<sup>th</sup>!

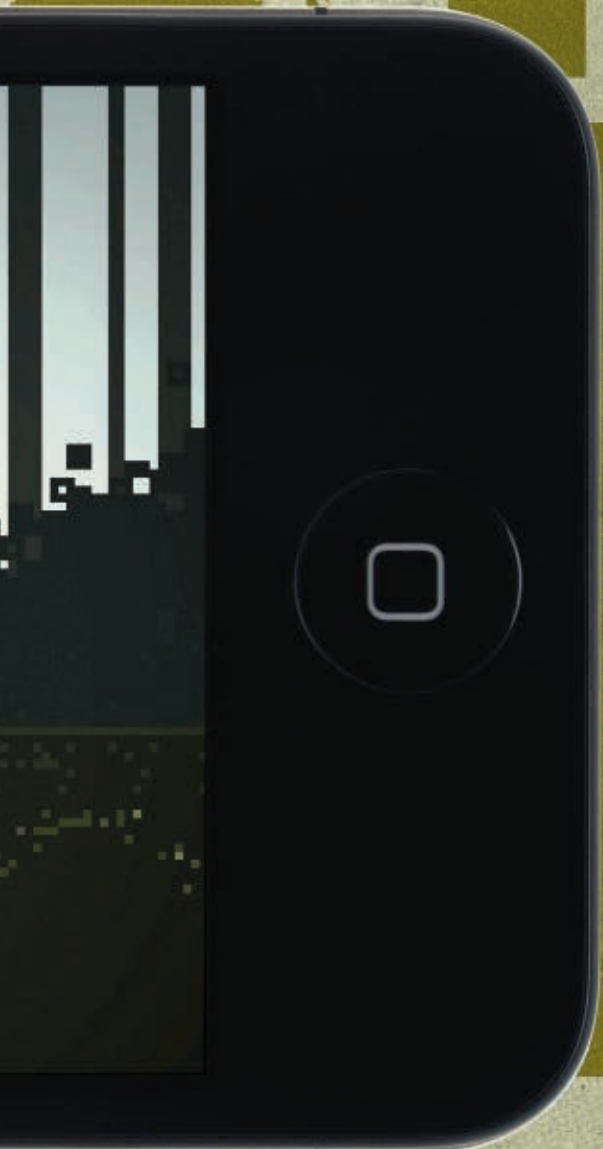
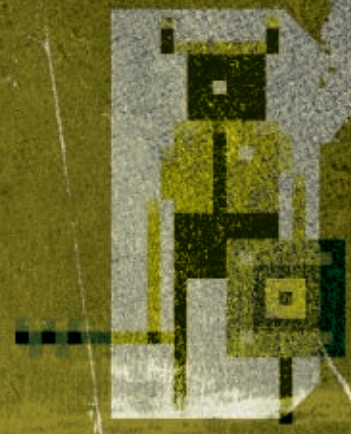


Visit [migs.ca](http://migs.ca)

postmortem







////////////////// SUPERBROTHERS: SWORD & SWORCERY EP (S:S&S EP) launched for iOS around the vernal equinox of 2011. It's an exploratory action-adventure with an emphasis on audiovisual style that tells a sparse story of a wandering warrior monk on a woeful errand. Even in the over-crowded App Store we feel it stands apart as an unusual offering, with an old-school design sensibility harkening back to video games of yore, and generously sprinkled with 21st-century ideas about what games could be. Despite (or maybe because) of its strangeness, S:S&S EP has found an enthusiastic audience, proving to be both a critical and commercial success in a sector of the market often dominated by fowl-themed time-wasters.

If S:S&S EP is an unusual video game, it's the result of an unusually collaborative process, in which three separate creative entities—the artist Superbrothers, the musician Jim Guthrie, and the dev studio Capy, aka Capybara Games—came together for the first time as co-creators.

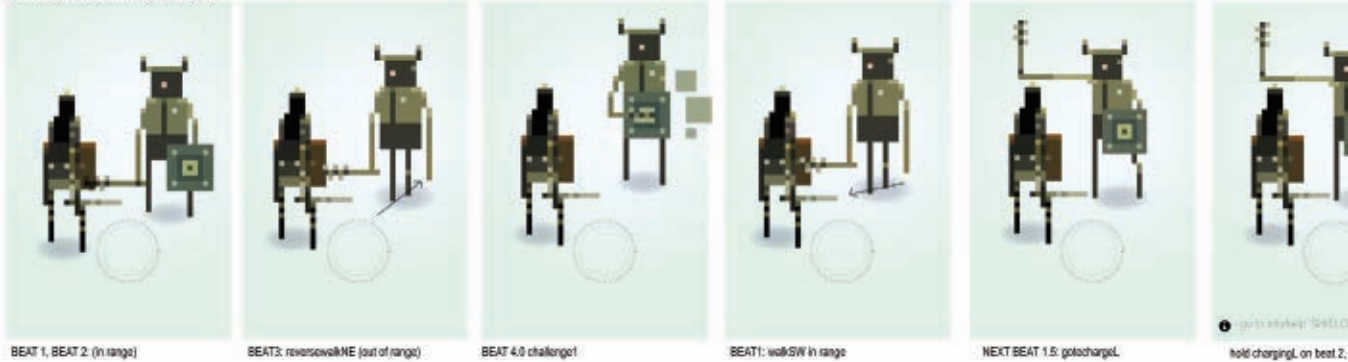
S:S&S EP is the very first game project undertaken by an ambiguously pluralized art and design organization known as Superbrothers Inc. Superbrothers—effectively a pseudonym for Craig D. Adams—started to hone a particular pixel art aesthetic known as “rustic 21st-century minimalism,” creating illustrations for magazines and newspapers as well as brief motion pictures. For S:S&S EP, Superbrothers supplied the core concepts, production design, art, and writing, and also defined the approach to sound and music for the project while acting as project co-lead.

S:S&S EP is also the first video game project undertaken by Jim Guthrie, the legendary composer and indie rock star, a veteran of bands including Royal City and Islands. Ten years ago Jim was in a peculiar groove while on tour, and started creating evocative chip-tune-type musical compositions on a PSOne using a rudimentary sequencer program called MTV Music Generator. Two of these strange songs, “Children of the Clone” and “Dot Matrix Revolution,” were interpreted into film by Superbrothers in 2005 and 2008, respectively, establishing a collaborative creative relationship that pre-dated the S:S&S EP project. Jim Guthrie signed on to S:S&S EP as a composer, sound designer, and co-creator along with Superbrothers and Capy, and a handful of Jim's pre-existing songs directly inspired locations, sequences, and story ideas. >>>

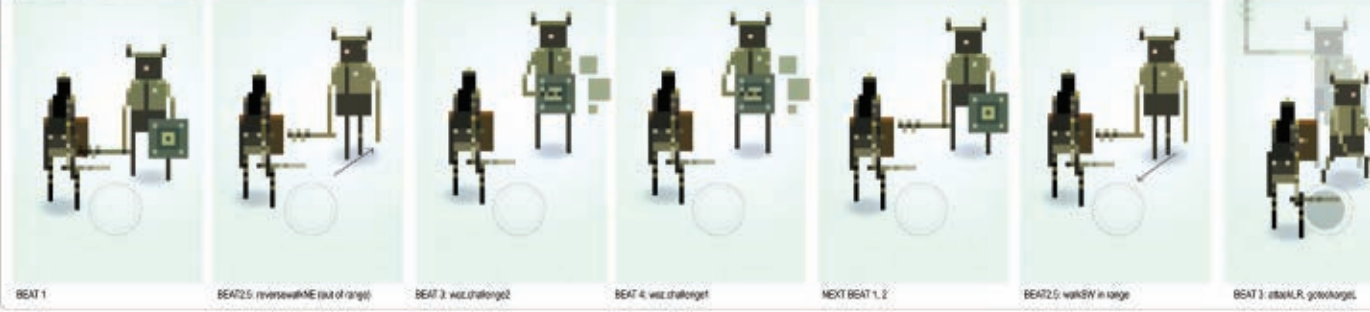


# SWORD & S

attackPattern=2 "SIMPLE BLOCK ATTACK"



attackPattern=3 "TRICKY RHYTHM ATTACK"



S:S&S EP was entirely made possible by Cappy, an independent game studio located in Toronto and creator of CRITTER CRUNCH and MIGHT AND MAGIC: CLASH OF HEROES. Cappy was founded in 2003, and has a long history in the pre-iOS mobile game space, and the iPhone edition of CRITTER CRUNCH was an early hit for the platform. S:S&S EP was a dramatic departure for the studio—an experimental, music-driven adventure video game instead of a mechanics-focused game like its earlier efforts—but with Cappy’s proven track record and extensive game-creation experience, a positive outcome seemed likely. Cappy staff made up the bulk of the core S:S&S EP team, with two programmers, Jon Maur and Frankie Leung, as well as Cappy co-founder and creative director Kris Piotrowski as a project co-lead. Cappy staff also assisted with administration, sound, quality assurance, and many other aspects.

**WHAT WENT RIGHT**

**1 a blank canvas**

/// As Nathan Vella, president of Cappy says, “Within five minutes of meeting Craig in person, I was yelling at him about how we need to make a game together. A mere couple months later we set out to develop a weird and style-driven game we were calling ‘Poopsock,’ which flew in the face of traditional iOS development wisdom.”

With S:S&S EP, we really wanted to create an experience that felt fresh, something that

would feel genuinely at home with the specific capabilities of iPhone and iPod touch (iPad had not yet been announced). To do this we basically ignored all existing apps and all the prevailing common sense of “what an app should be,” and we approached the platform as though it were something completely new. We considered the particular form factor of the machine, how it is most comfortably played, and how the machines typically are used outside of playing video games. We challenged ourselves to imagine “the perfect iPhone/iPod touch experience,” and we would often ask ourselves in the early days, “What would Shigeru Miyamoto do if he were tasked with creating for this particular machine?” Core concepts like Twitter integration, the ability to tilt the machine to unsheath a sword (switching from a laid back adventure style of play into a disruptive action-heavy combat encounter) and our whole approach to mechanics arose from these early thought experiments.

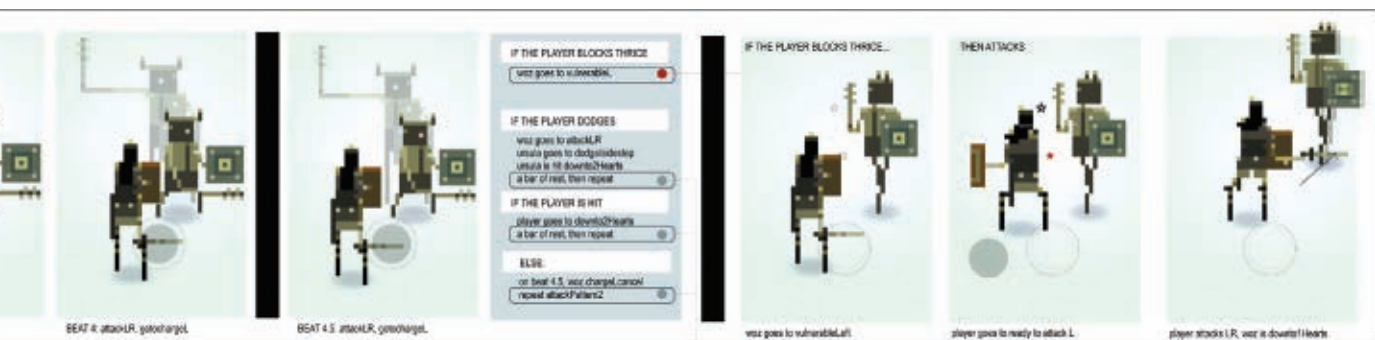
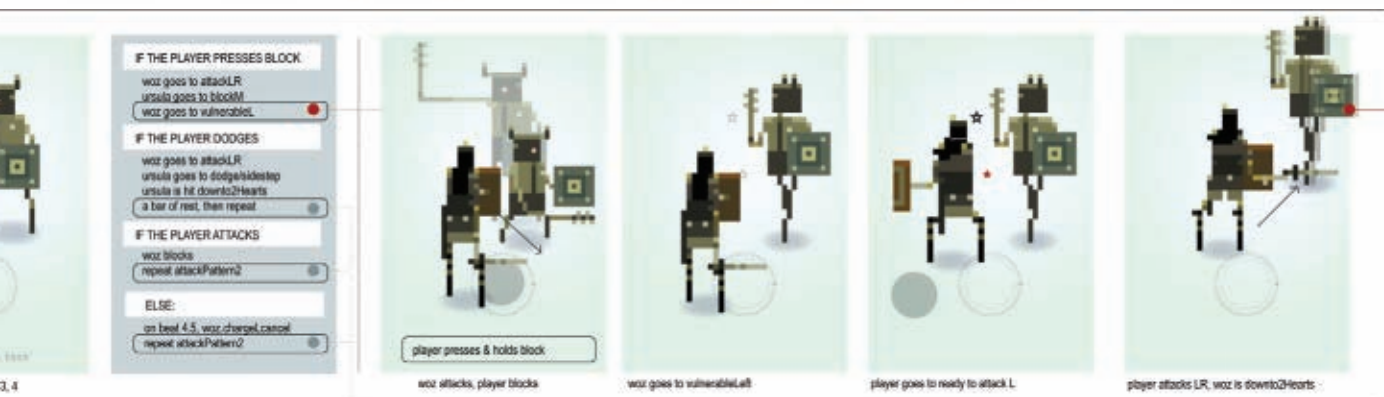
Additionally, the concept of Input Output Cinema (I/O Cinema) helped define many of the initial explorations and prototypes for S:S&S EP. I/O Cinema is an intentionally amusing phrase intended to describe a theoretical category of electronic entertainment that would be built using the craft of video games, presented with the language of cinema, with a focus on creating and sustaining a subtle conversation with the player. Rather than telling the audience “this is how the game works, this is how you overcome obstacles

and achieve a score,” with I/O Cinema, the audience would be free to poke, prod, and attempt other inputs. The system would be devised in such a way that the outputs would be amusing, informative, and intermittently meaningful so as to encourage further exploration and progress. In S:S&S EP, we feel that much of the enjoyment comes from slowly figuring out the rules of the world, discovering the story as it unfolds, and stumbling toward a hazily understood objective; so while it might not be a great “video game,” we can proudly proclaim that it is a brave experiment in I/O Cinema.

As Cappy creative director Kris Piotrowski remembers, “We let ourselves explore and discover new concepts, intentionally avoiding obvious solutions to standard design problems. This approach helped us end up with something we feel is unique and interesting.”

**2 trust**

/// “One of the things that went right was that we could trust in everyone’s ability and intent” says Jim Guthrie. Through it all, there was a solid foundation of trust that kept everything on an even keel. We knew Jim would be able to find the right song or sound for any occasion, we knew Superbrothers would eventually find the art or narrative concept, and we knew that the Cappy team had their hearts in the right place and that they’d figure out how to move us forward to get us across the finish line. Even through the darker,



more uncertain times on this project, this basic trust remained intact.

This high degree of trust allowed for an improvisatory, highly iterative design process between collaborators: Jim's song might inspire a scene from *Superbrothers*, which might inspire a design idea from *Capy*, which would influence Jim's song and suggest new ideas for everyone. Even the strangest of suggestions—a naked dancing bear, unknowable cosmic geometry, a landscape inspired by Al Jaffe *Mad* fold-ins—would eventually come together with the right art, sound, and design ideas. This trusting, respectful relationship allowed each collaborator a high degree of authorship. People were able to more or less "own" aspects of the project, giving it a personal, handcrafted feeling.

This same trusting approach extended to other contributors, as *Superbrothers*'s Craig D. Adams relates, "Robert Ashley's hilarious improv voiceover recordings for *Logfella*, Clive Holden's authoritative voiceover for *The Archetype*, and *Scientific American*'s otherworldly sound design for *The Moon Grotto*—these were all created with minimal creative direction and next to no interference. We could simply trust that whatever each contributor dreamed up would be a perfect fit for the strange, sprawling *S:S&S EP* project."

In a strange twist of fate, *Capy* programmer Jon Maur had worked with Craig almost 10 years prior, and this coincidence (or fate) added to the trust on the project. Through Jon and Craig's

shared history came the immediate mutual understanding between creative and technical that usually takes significant time to build.

As Kris Piotrowski says, "Everyone on the team brought a different skill and a different sensibility to the project, and the end result feels mildly schizophrenic in every regard, but beautifully so."

### 3 the creative

/// Music composition, art production, writing, trailer making, and even crazy high-concept idea-generating seemed to be the smoothest elements of the project, which is fitting considering the make up of the team.

*S:S&S EP* was initially conceived of as "a record you can hang out in," and it was primarily inspired by a collection of previously unreleased musical compositions by Jim Guthrie. As the project progressed, Jim was called upon to score an absurd number of hard-to-describe moments, all of which he handled with aplomb. With the musical voice of the project as a starting point and the artistic style of *Superbrothers* already established before the project began, the next step was to decide on a time, a place, and some characters to wrap around the vague ideas we had in mind.

"Because we knew going in that our approach would be unusual, we wanted to choose an aesthetic that would resonate with a broad audience, including video game enthusiasts, so we decided on a concept intended to evoke the

phrase 'the archetypal adventure video game,'" remembers Craig Adams. "The name *SWORD & SWORCERY* is in line with this creative direction: sword and sorcery, the most generic descriptor for arguably the most universal genre of storytelling, amusingly misspelled, seemed to be a perfect fit for what we had in mind."

The setting and themes of *S:S&S EP* are original, but they were conceived with the imagery and tone of Robert E. Howard's genre-defining sword-and-sorcery stories as a template. Shigeru Miyamoto's *THE LEGEND OF ZELDA* was also foundational. The *Trigon Trifecta* in our game is an obvious reference to another iconic set of three triangles, and there were many other cases where we relied upon the audience's video game literacy to sustain a joke, an idea, or a puzzle. To invite a more intellectual contemplation of all these ideas, *S:S&S EP* was constructed with an intentionally sparse narrative, bookended by strange pronouncements from a Rod Steiger-like figure known as *The Archetype*, who presents the adventure as a kind of psychological evaluation, referencing influential thinkers like Carl G. Jung, whose thoughts about archetypes and the collective unconscious were very much on our minds. We sprinkled in ideas from thinkers further down the rabbit hole like Timothy Leary, Terrance McKenna, and crackpot conspiracy theorists like David Icke, whose 12-foot-tall reptilian shape-shifters were woven into *S:S&S EP*'s obscure backstory. There are also not-so-subtle



connections to Davids Cronenberg and Lynch. All of these persons offered common reference points as we tried to find the vibe of S:S&S EP.

While many of these ideas grew out of the initial research and sketches on the Superbrothers side, the ideas grew and were shaped in collaboration with everyone else on the project. "It helped that we were all on about the same creative wavelength and were all super excited about making something kinda interesting, kinda crazy. We didn't worry too much about people taking all these ideas too seriously, though, because of course we were going to let the music do most of the talking, and we were going to include naked dancing bears and other jokes to take the edge off."

#### 4 perseverance

/// Jim Guthrie notes that one of our strengths was "our ability to push forward even when things seemed to be at a complete standstill."

Craig remembers, "Capy is a special studio, not just because the team was ready to take a risk on an experimental adventure game led by a couple of relative neophytes, but also because their gameography is rock solid. Clearly they knew what it would take to get a job done, and get it done right. I had confidence in their abilities

"Capy has enough history making games to be able to trust in a creative direction for a video game, even while it's basically unplayable. As Kris said multiple times on this project, a game sucks and sucks hard for a long, long, long time before it gets good, and it only ever gets good right at the end, after enough stuff is fixed. So you've got to have faith that it'll get there, to put one foot in front of the other, fix one thing at a time until enough stuff is fixed that the intended experience begins to shine through. I think this was especially true for S:S&S EP, a scripted adventure video game with limited replay value that has almost nothing to offer on the thousandth playthrough." Craig remembers. "Even in the darker moments, when I was so deep in the trenches and so exhausted that I couldn't imagine a positive outcome for the project, I had faith in Kris's belief that it would all work out in the end."

Kris adds, "Loose, high-level goals were our guiding lights, and the rest was just faith that the game would actually come together at some point."

#### 5 buzz

/// With the Jim Guthrie, Superbrothers, and Capy collaboration as a starting point, we knew

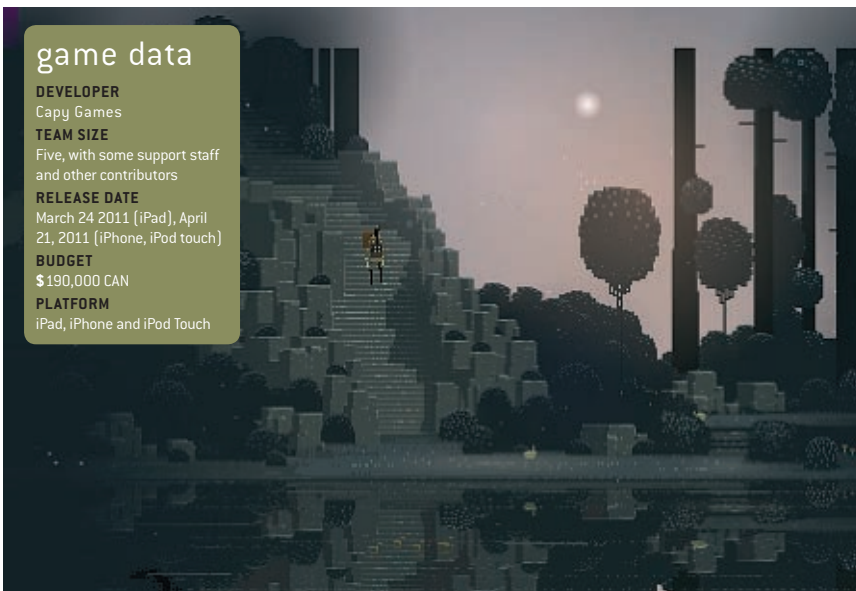
heard, or watched it was clear that this voice would continue in the game.

Our announcement teaser—a tiny slice of art and song—in December 2009, made after only 30 days of prototyping, created a minor stir in some quarters. Our first public reveal, just two months later in an absolutely jam-packed bar in Toronto at a Hand Eye Society social, stirred up quite a bit more interest and intrigue. The stakes were raised like crazy when, a month after the public reveal, our S:S&S EP playable prototype struck a chord with the folks who played it at our IGF booth on the GDC 2010 Expo show floor. This early appearance evoked a strong positive response from just about everyone who saw and played it, creating a crazy amount of positive buzz around a project that barely existed. As much as the hype added the pressure of heightened expectation to the team, it was also a vote of confidence for the overall creative direction. We had the feeling that so long as we made something decent, there would be an audience waiting for it. Shortly after GDC, the project "went dark" for months and months while the team tried to focus on the task at hand ... other than vague guesses at potential launch dates, all of them woefully inaccurate, almost nothing was heard from the project until almost a year later.

Thankfully, all was forgiven when, on the eve of GDC 2011, a clip described as the "S:S&S EP Audience Calibration Procedure" was posted, re-introducing the target audience to the strange vibe of the finished project and firmly announcing an imminent release date "around the vernal equinox." We were super proud of this clip—a crazy concept from Capy, storyboarded by Superbrothers, scored by Jim Guthrie, anchored by some solid voiceover. The clip presented a project that was now fully formed, from a crew of now-veteran collaborators, put together skillfully by Capy trailer-master and president, Nathan Vella. After a year's worth of speculation and semi-obscure, this clip put the project back into center stage, attracting more than 200,000 views on Vimeo alone in a matter of days, and re-awakening the enthusiast press who had been long starved for details.

In the weeks leading up to launch we were blessed with extremely positive posts and write-ups at major video game enthusiast sites and magazines, and when the project finally launched, it was covered just about everywhere, from Mashable.com to The Huffington Post, as well as in national newspapers in Canada and the UK, and the weeklies in Toronto. Even large USA news outlets like ABC News, MSNBC, CNN, and Fox News found a slot to feature S:S&S EP. Interestingly, much of this coverage was on the arts, music, and culture side of things, rather than in the gadgets and technology section.

The #sworcery takeover of Twitter, covered in the next section, certainly helped keep the project's visibility immediately post launch, but we think



from the start, but I had no idea what it would actually take to get a project done, and when the going got tough, I relied on them to figure out what to do.

"There were plenty of cases on S:S&S EP where I'd get lost, not knowing what to prioritize, and usually Kris would parachute in from another project, take stock, think things over, identify the gaps, and then get us going again on a path toward completion ... usually by tackling the hardest problems first.

we had a project with the potential to make a real splash in the worlds of music, art, and video games. With this in mind, right from the start we treated the project as something worth getting excited about, well before we knew what the project would actually be, and so we treated every point of contact with the audience—from the swordandsworcery.com web site, to announcements and news, to trailers—as something special. All these communications intentionally had a singular voice, so when read,

what people seemed to respond to most at the beginning was our respectful, earnest, unusual, and occasionally enigmatic approach to the project, as well as our respectful attitude to the audience.

## WHAT WENT WRONG

### 1 lack of core mechanics

/// S:S&S EP's blank canvas approach and I/O Cinema concept is listed among the project's strengths, and while we feel our investigations into this theoretical non-genre represent much of what is interesting about our effort, the flipside is that firming up even the most basic mechanics was actually super challenging.

As by far the most experienced dev on the team, Kris remembers, "Trying to make a game with no real idea of what it should be is pretty hard, and this directly resulted in some of our most difficult moments. It took us a long time to find the game people are playing now, and there were moments when the project had no end in sight and was a total mess, with a million loose ends and massive design, game flow, and narrative problems. Although we managed to find our way out of the extremely horrifying middle portion of development, we came out of it absolutely drained and kind of half crazy."

Jim Guthrie recalls it similarly: "We always had more of an idea about the vibe and atmosphere of the game than we did about the actual gameplay. How the game made you feel was always more important than how it was actually played, and that caused a lot of problems for obvious reasons. We were also plagued with knowing more about what kind of game we didn't want it to be than what kind of game it was going to be."

As the originator of the project's core concepts, Craig remembers, "I wanted to be very careful that the gameplay mechanics were narratively appropriate and expressive, and supported the style of experience I had in mind. So early on I would end up resisting a lot of straightforward ideas that probably would've made the experience more fun and understandable, because I felt they would come across as too traditional and video game-like. I was always seeking a more unusual, organic solution."

"I had no solid prior experience as an actual game designer, and when the project was being conceived and prototyped I was kind of going it alone, flailing around trying to create something coherent for the adventure and combat sections, and despite my hazy imaginings, I simply wasn't able to firm up anything mechanics-wise in those challenging early months. This is partly why the GDC 2010 playable prototype ended up being necessarily content-focused, relying primarily on the audiovisual experience ... with an extreme time pressure of a few months and at a time when there was only minimal project support available, I felt Jim's songs and my artwork were

much more dependable than a game system that didn't exist yet.

"When Kris eventually moved from a supervisory role to more of a co-creator role following the creation of the GDC prototype, he helped shape and grow the project as a content-focused, scripted, linear storybook experience. This was ultimately a very appropriate and positive thing, but during production the content demands seemed so heavy—so many pixels to paint, so many sounds to place, so many moments to orchestrate with awkward tools. Worrying about this was the source of many a sleepless night, and toward the end of the project I became so burned out that I could barely playtest, so Kris and Christian, Capy's QA specialist, had to carry the project to the finish line."

### 2 the grind

/// Craig relates, "It took an awfully long time for the project to take shape, and it took an equally long time for the team to find any kind of groove. Even when we did, it was a lopsided and awkward groove. Admittedly, it was a weird project from the start—very high concept, driven by mood, emotion, and gut feelings—and while some core concepts remained firm from the start, everything else was constantly in flux as we discovered together what S:S&S EP wanted to be.

"Menu systems were developed and necessarily cast aside, inventory systems were redesigned and re-redesigned, and combat encounters were perpetually underdeveloped. Our first attempt at a complex puzzle-solving-sequence—where The Scythian crosses a chasm in the very first session—took a whole month to implement, but it didn't feel right—and I led the charge to dismantle it and start fresh. All these iterations and revisions were improvements, new 'best guesses', but when every new best guess took weeks to implement and evaluate, the feedback loop seemed very wasteful."

The two programmers on the project, Capy's Jon Maur and Frankie Leung, went above and beyond on many occasions, capably handling various challenges and methodically overcoming every obstacle, but their role was usually limited to patiently serving the ever-changing, often strange requests from the creative side. While there was ample collaborative chemistry between art, music, and design, there wasn't much room for collaborative chemistry between creative and technical for much of the project.

Craig adds: "The communication from the art and design side to the code side needed to be painstakingly specific, and then the implementation of almost every feature required intense iteration, collaboration, and revision ... so with a team as tiny as ours, this meant that our pace seemed agonizingly slow at times. Consequently, the feedback between collaborators was not as tight as I had hoped it might be. For example: we'd receive a

new set of loops from Jim and would be looking to get them in-game to evaluate them and refine our ideas, but the team would be busy struggling with another set of urgent challenges, and so it might be weeks or even as long as a month before Jim would get his ears on a new build with the new loops in place, and in many cases we would could only afford the time for a few tweaks and some clean-up on the sound side. Thankfully, in many cases Jim's first best guess for sound and music would be pretty much perfect."

### 3 a woeful errand

/// Every project has its challenges, and it's not uncommon for a tough project to tax the physical and mental health of team members. With S:S&S EP there were pressures to deliver from the start, starting with a 30-day prototype sprint to create an IGF submission, followed by a two-month sprint to create an audiovisually polished playable prototype for the GDC 2010 show floor. These pressures escalated after GDC 2010, once the project had attracted a significant amount of positive attention, and there was an expectation for the team to deliver on the creative potential suggested by the prototype in a reasonably timely manner. This would prove to be a tall order for a tiny team with hazy, ambitious goals, and at least one woefully naive project lead.

As Craig relates, "S:S&S EP represented my first time as a project lead, and my first time as an entrepreneur, sharing the risk/reward with Capy. This was also my first time as primary content creator for a game project, and I was responsible for every pixel, every animation, every word. The pressure was on from the start, as I felt I had to try to prove myself to Capy and the S:S&S EP team, while simultaneously trying to build an experimental project that I could only hazily imagine, and couldn't coherently explain.

"With the exposure resulting from our GDC 2010 showing, it was tough to keep everything in proportion. The project would eventually become a true collaboration, but in the early stages, it was very much on Superbrothers' shoulders. I had set the project on its path, determining the core concepts and driving the prototype process, and I had even been permitted to put the name Superbrothers—a brand I had spent years defining—right there in the title. This was a huge blessing for me, and one that I'm immensely grateful for now, but in the darker times it hung heavy with the weight of a curse. If the project failed to live up to its creative potential, it would have been a lasting shame for a personal brand I had spent years shaping; and if the project failed to perform commercially, I would be deep in debt. As the project wore on in the summer of 2010, overshooting a series of budget-driven deadlines, the workload got heavier. For years and years I had been working toward a project exactly like this, and now that this dream project had come about



and was deep into production, it had become an unending, inescapable nightmare.

“Thankfully, everyone involved with the project was committed to seeing the project through. The folks at Capy and Jim Guthrie were class acts throughout, and I was lucky to have a very supportive family to assist me financially when my savings evaporated partway through. Nevertheless, this stressful situation led to a few crashes in the latter half of the project. I was still able to be productive and paint—animating and designing through doubt, anxiety, insomnia, and whatever else—but my ability to problem-solve and stay positive was adversely affected, and I relied heavily on Kris to push things forward. Elsewhere I’ve written about the deteriorating health of The Scythian as she persists on her woeful errand in the narrative of *S:S&S EP*. Both Kris and I were very aware that this grim trajectory reflects the experience of all different types of creators who set out to do something tough, something that they believe in. I’m very conscious that there are many creators and teams who have had it much worse, who have struggled for longer, who have suffered more, and I know there are some who are still struggling with sacrifices and unintended consequences of a prolonged ordeal ... but knowing this doesn’t totally flush away the memory of months of constant panic. Of course, if I had known what I was in for I probably wouldn’t have signed on, and if I hadn’t signed on, we would never have gotten it done ... so really, all’s well that ends well.”

#### 4 troublesome tools

/// *S:S&S EP*’s production was complicated by a few troublesome tools. Our homebrew spline editor and entity placement tools were bare bones but functional, but with the ever-changing and highly iterative nature of the game, we could never afford the time to create much of an event or animation editor. For each event we were left with the prospect of creating and maintaining a set of interoperating XML and LUA files set up by the programmers and edited, hastily and in a limited way, by Craig.

*S:S&S EP*’s music and sound were made possible by the iOS version of FMOD, a common audio middleware package. We also made heavy use of the Designer tool bundled in with FMOD that allows a non-programmer to set up logic for adaptive music, or set very specific parameters for sounds, even on iOS. Capy had positive experiences with the tool on previous projects, so it was a natural choice. Further, Superbrothers’ Craig D. Adams had prior experience on a previous project as a sound design coordinator, so Craig took the helm of the project file for most of the project.

We wanted the audio experience to be as rich and involving as possible, with evocative environment sounds and fancy music behaviors. While our first explorations were creatively interesting, we were quickly grappling with issues related to the technical constraints of the iOS



machines and, frustratingly, the iOS version of FMOD’s occasionally unpredictable quirks.

As Jim remembers, “On the music side we had all these plans of breaking down the score into a million little adaptable loops that would seamlessly change depending on what the player would do. It proved too much for iOS and caused a lot of problems with loops not triggering for long periods of time, or not at all. We learned that the less we tried to control this ‘adaptable score’ and let the tunes just play out, the richer the game experience seemed.” Put another way, Kris Piotrowski remembers, “Wrangling the sound design, which is actually very complex, was a nightmare.”

Craig adds, “We absolutely needed an audio tool for this project, and it was essential for me to roll up my sleeves and get involved as a sound designer, to be able to express my ideas without too much interference, and FMOD Designer gave me an incredible amount of control over every sound and every loop. In many ways it was a pleasure to use, but because this tool is always in development, there were some frustratingly mysterious issues that hurt production. Further, with my novice status in the sound design world, many of the obstacles I had to overcome were of my own creation. Toward the end of the project I was able to hand off my messy audio project file to Capy co-founder, audio specialist, and FMOD veteran Sean Lohrisch, allowing

me to focus on writing and other critical aspects of *S:S&S EP*. Sean was indefatigable, systematically solving all of our audio issues and working through the middleware’s quirks. Looking back at the tools situation I can’t really find a lesson to learn in this. It’s almost as if it had to be this way given the project’s constraints and Capy’s familiarity with this tool, but dealing with these tools was no picnic.”

#### 5 the twitter shitstorm

/// One of *S:S&S EP*’s core concepts was to make a game that felt perfectly at home on Apple’s touchtronic machinery, and because these machines are at least partially communication devices, we wanted to establish a system that would allow an individual to optionally share little tidbits of their experience with like-minded friends.

Craig remembers, “A year or two before social games became a big deal, I attended the Social Games Summit at GDC. While I was disturbed by much of what I heard there, it got me thinking about how much I’ve really enjoyed video games with a social dimension. I had enjoyed *ANIMAL CROSSING* years before, but even more than playing the game, I enjoyed talking about it with friends, sharing esoteric knowledge and commiserating about the slumlord raccoon Tom Nook. I remembered how the knowledge of obscure secrets in 8-bit adventure video games like *The Legend of Zelda* or *Super Mario Bros.* would pass from brother to sister, from classmate to classmate. Discovering these secrets and sharing them with friends definitely added to those experiences for me.”

With this in mind, all the text in *S:S&S EP* was written with Twitter’s 140-character limit in mind, and we put in a feature that allowed people to optionally sign in to their account so players could rebroadcast a tidbit of out-of-context text to their followers. We took the Twitter-centric concept a step further, with an in-game logbook called “The Megatome” that collected all the text in a scrollable format resembling a Twitter feed, and as the adventure progressed, the unspoken thoughts of non-player characters would pop up in this logbook like tweets, offering clues or context about what was going on in the world. We felt really good about these ideas creatively, and on the design side, the ability for players to discover and propagate knowledge allowed us to commit to some really mysterious ideas with the confidence that the audience, collectively, would be able to make sense of them. We imagined a player, during a particular moon phase, in a particular location, discovering something unexpected and then broadcasting something to that effect ... and we felt that these occasional glimpses inside the world of *S:S&S EP* would be informative enough to intrigue other players and amusing enough to entertain anyone who wasn’t playing. We also felt the Twitter integration was background enough that it wouldn’t negatively

# SWORDCERY

affect the experience, but at the eleventh hour, following a few playtests where the feature had gone unnoticed, we chose to have a handful of the text tidbits suggest themselves for rebroadcast with a little “tweet this?” UI animation, and a week before submission we changed the name of the App to #sworcery. We were happy with the way everything fit. It sounds naive now, but as the S:S&S EP buzz built leading up to the moment of launch on the App Store, we had absolutely no idea we were about to unleash a monster.

As Kris puts it, “S:S&S EP launched and Twitter exploded. Every game player who was on Twitter had #sworcery in their feed.”

In the first few hours and days, as overly excited early adopters gleefully chose to rebroadcast every tidbit of text, many people’s feeds were filled with a series of bizarre pronouncements, all identified with the #sworcery hashtag. There’s little doubt that this outpouring of enthusiasm improved S:S&S EP’s visibility in that critical time, but we had poured our hearts and soul into creating something beautiful, working hard to respect our audience at every step of the way, and in a matter of hours we were at risk of rubbing a lot of people the wrong way—specifically people in the game industry, our friends and peers. For a few moments there, we worried that this unplanned social media takeover would hijack the narrative of the project and eclipse our efforts, and efforts were made—on Twitter, of course—to apologize to those inconvenienced and explain our intentions.

Thankfully, the shitstorm subsided in a matter of days as the novelty dissipated. The #sworcery hashtag lapsed back into its intended pattern, as an occasional, informative, and amusing dispatch

from a handcrafted world. Even the harshest critics of this #sworcery debacle have since relented after experiencing S:S&S EP themselves and understanding the feature in context. What’s more, some of the folks at Twitter dug our approach, and many of the players who participated in this way seemed to have a really good time, playing a solitary adventure alongside their friends.

## WE CONSTRUCTED A CONSTRUCTION, AND WE FELT PRETTY GOOD ABOUT IT

/// On June 30, a few months after the project’s launch, more than 200 people gathered in a cinema at TIFF Bell Lightbox in downtown Toronto for something called a “Midsummer Rockshowcase,” a showcase of short films and trailers intended to shine a light on the city’s DIY video game crowd, followed by a rock show from Jim Guthrie and his seven-piece band performing a set of material from the SWORD & SWORCERY project, live for the first time. This was accompanied by a SWORD & SWORCERY-themed visual presentation from Superbrothers.

This event, as well as the vinyl release of *Jim Guthrie’s Sword & Sworcery LP: The Ballad of the Space Babies* (which has now sold out of its second pressing!) get at the heart of the unusually collaborative SWORD & SWORCERY project. As Jim says, “We were all trying very hard to make something different, and it showed in how hard we all worked on it.”

We set out to blur the lines between a video game and a record, between a record and storybook, and we tried to make something with style and a little bit of soul ... and to some extent,

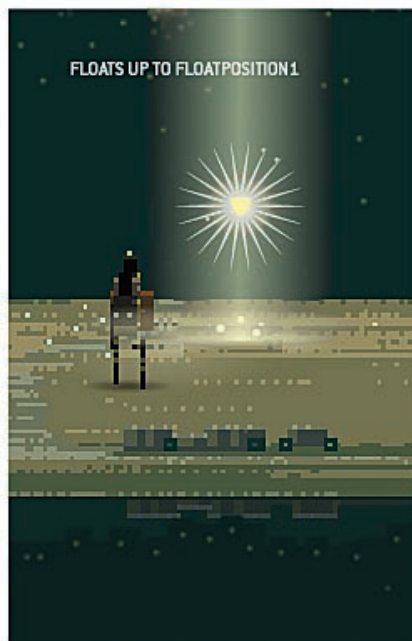
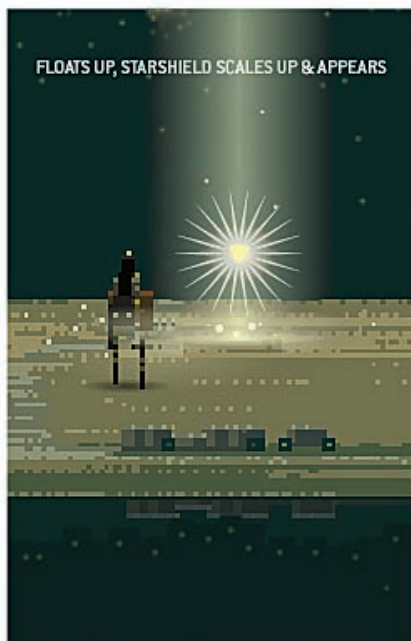
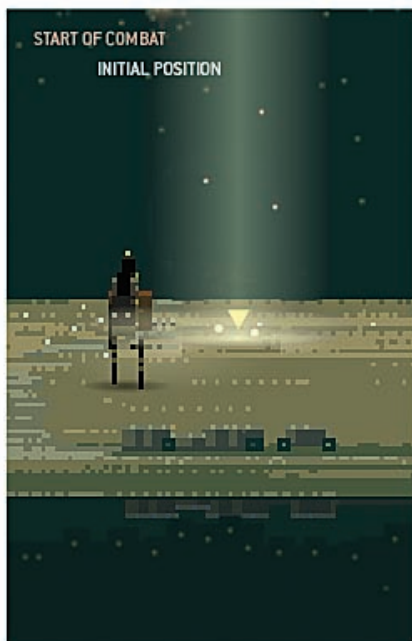
it seems we’ve succeeded. It was a tough project to make, but we ended up with something we’re immensely proud of, and in the process we’ve managed to reach over 250,000 people, many of whom have taken the time to leave a five-star review and download Jim Guthrie’s record.

To many, S:S&S EP is a game that defies the common sense of the iOS App marketplace—it’s an occasionally obscure, relatively long-form adventure on a platform usually associated with fun, bite-sized little games—and in only a few months it seems to have already become a bit of a cult classic in some quarters, helping to define what’s possible and what’s viable on Apple’s touch machinery. Launching on almost the exact same day as Nintendo’s 3DS system, it has been especially interesting to see S:S&S EP cited as a counterpoint to Nintendo President Satoru Iwata’s claims regarding the disposability of inexpensive iOS games.

While there are a few small surprises still in store with this project, it’s very likely that S:S&S EP will remain a one-off, the way it was intended. Capy, Jim Guthrie, and Superbrothers each have their own projects to attend to and any potential reunion on a new project is entirely unknown at this time, but we’re all very hopeful that more unusually collaborative video game projects like S:S&S EP will appear in the future. ☺



CRAIG D. ADAMS is a spokesman for Superbrothers Inc., KRIS PIOTROWSKI is creative director at Capy, and JIM GUTHRIE was the composer for S:S&S EP.





## THE 2<sup>ND</sup> ANNUAL

# game DEVELOPERS CHOICE ONLINE awards

The Game Developers Choice Online Awards is the premier award ceremony for peer-recognition in the connected games industry. Taking place annually during GDC Online, the Choice Online Awards recognize and celebrate the creativity, ingenuity and innovation of the finest online developers and games created in the last year.

### AWARDS ARE PRESENTED IN THE FOLLOWING CATEGORIES:

#### 2011 AWARD CATEGORIES

- Audience Award
- Best Audio for an Online Game
- Best Community Relations
- Best Live Game
- Best New Online Game
- Best Online Game Design
- Best Online Visual Arts

- Best Online Technology
- Best Social Network Game
- Online Innovation Award

#### SPECIAL AWARDS CATEGORIES

- Online Game Legend
- Hall of Fame

Award finalists will be announced in August.

Stay updated at [www.gdconlineawards.com](http://www.gdconlineawards.com)

PRESENTED BY

**GameDevelopers**  
Conference

PRODUCED AND HOSTED BY

**GAMASUTRA**  
The Art & Business of Making Games



**GAME DEVELOPER  
MAGAZINE**







# SOUNDMINER INC

## Soundminer HD

**MOST MAC-CENTRIC SOUND DESIGNERS, MIXERS, FOLEY ARTISTS, VIDEOGRAPHERS, MUSIC EDITORS,** and others entrenched in the audio-visual arts are intimately familiar with Soundminer, a sound effects and music database application. Soundminer, currently on version 4, has been the go-to tool for organizing, searching, and importing sound effects and music libraries on the Mac for years. Additional versions for other platforms have been released, including Soundminer XP for Windows and a web-based interface, but neither of these tools have matched the breadth, power, and flexibility of the flagship version of Soundminer. Enter a new product line in the Soundminer family—Soundminer HD—intended to create a true cross-platform solution with most of the power of the current Mac version (v4), expanded functionality, and some new features.

### WHAT IT ISN'T

» Before we dive into what Soundminer HD is, let's first look at what it lacks in comparison to V4. Fortunately, most of the core features are carried over. The interface retains much of its familiar layout, with a few changes intended to simplify workflow. For example, the command line interface has been removed. While you no longer have a DOS-like window to run tasks to clean up your database, embed metadata, automate metadata formatting, and so forth, most of the functionality is still available as menu items, rather than command line executions.

VST support is not present in Soundminer HD, which in V4 was a nice way to audition effects on audio clips before importing them into your digital audio workplace

(DAW). However, the DAW has historically been the place to sculpt your sounds using plug-ins and other manipulation techniques, so I don't view this as a showstopper—it's more of a potential minor inconvenience. Also absent from HD is ReWire support, which is also not necessary for most people, but in V4 served as a means to get smaller segments of sounds into a DAW not yet supported by the application.

Soundminer HD, like its V4 big brother, supports spotting to track for certain applications. Spotting is a means to import directly to a specific area on the DAW timeline. Currently Pro Tools, Nuendo 5, Cubase 6, Logic, Pyramix, and Soundtrack Pro are all supported. There are inconsistencies inherent in some of the apps themselves, though. For example, Logic will only

spot to the first track in a project, while Nuendo users on Mac can spot with additional download. Final Cut and Avid Media Composer support the transfer of files with a single mouse click, which can simplify the task of grabbing clips and importing them into a project.

One way or another, Soundminer recognizes most DAWs. I tested Soundminer with REAPER, an unsupported DAW, and Soundminer recognized it immediately. On the Mac, I was able to grab a clip of a sound, tweak the pitch, and bring it into REAPER by using the Spot to Timeline button. Unfortunately, it opened the new file in a new project tab and not on the track/position where I wanted it. The PC version did not fare as well, as the files were not brought into REAPER at all. The issue here is not Soundminer per se, but the fact that REAPER needs to implement code to support Soundminer's features. As a final fallback, Soundminer HD supports drag-and-drop of samples into a DAW project. The drawback of drag-and-drop is that you can only drag-and-drop the entire audio sample, not a snippet thereof, and you cannot apply pitch or volume modifications to the sample before importing it. However, if you wish to access snippets of processed files for an unsupported application, you can easily do so in the Transfer History window and drag those into your project.

Soundminer HD also lacks the thesaurus of V4, which provides

### SOUNDMINER INC

#### Soundminer HD

157 Princess Street, 3rd floor  
Toronto, Canada M5A 4M4  
[www.soundminer.com](http://www.soundminer.com)

#### PRICE

> Basic: \$199  
Plus: \$399  
Universal: +\$100

#### SYSTEM REQUIREMENTS

> Mac OS X 10.5 or higher  
Windows XP or higher  
iLok or HASP USB dongle (sold separately)

#### PROS

- 1 Solid, fast search engine
- 2 Works with most popular DAWs
- 3 Importing new files and embedding metadata is smooth and easy

#### CONS

- 1 Only writes proprietary V4 metadata format
- 2 Spot to Timeline not supported for all DAWs
- 3 Very young product with some minor bugs still present

the engine with a dictionary of synonyms to supplement searches. For example, typing in "roar" would also return similar items such as "scream" or "growl." It's a useful feature when randomly perusing libraries, but once you know your library, maintain consistent naming and description methods, and use Boolean searches, it becomes less of a necessity.

Lastly, for now, there is no SQL server support in Soundminer HD. V4 contains a separate stand-alone product called Soundminer Server that allows for a centralized database for multiple designers to work from and modify. It's most beneficial to an organization with many designers working within Soundminer and constantly changing or uploading new content. You lose this



Figure 1: The main window of Soundminer HD.



# TOOLBOX

- ✓ Mark
- Artwork
- ✓ Filename
- ✓ Description
- ✓ Duration
- ✓ Library
- ✓ Channels
- ✓ Location
- ✓ Popularity
- ✓ Category
- BitDepth
- CreationDate
- Designer
- FilePath
- FXName
- Index
- Keywords
- LongID
- Manufacturer
- Microphone
- ModificationDate
- Notes
- Pathname
- Rating
- RecMedium
- RecType
- SampleRate
- Scene
- ShortID
- Show
- Source
- SubCategory
- Take
- Tape
- Track
- UserComments
- Volume

**Figure 2:** List of data fields supported by the application. The user can add data for any of these, and even request custom fields from Soundminer.

A	B	C	D	E	F	G	H
Filename	Category	Description	Manufacturer	Location	Library	LongID	ShortID
19 MARK_SOUND_EFFECTS_02.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-3	SOAMPFX	Mark Dr. Bradley D. Meyer	
20 MARK_SOUND_EFFECTS_03.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-2	SOAMPFX	Mark Dr. Bradley D. Meyer	
21 MARK_SOUND_EFFECTS_04.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-1	SOAMPFX	Mark Dr. Bradley D. Meyer	
22 MARK_SOUND_EFFECTS_05.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-2	SOAMPFX	Mark Dr. Bradley D. Meyer	
23 MARK_SOUND_EFFECTS_06.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-1	SOAMPFX	Mark Dr. Bradley D. Meyer	
24 MARK_SOUND_EFFECTS_07.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-2	SOAMPFX	Mark Dr. Bradley D. Meyer	
25 MARK_SOUND_EFFECTS_08.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-1	SOAMPFX	Mark Dr. Bradley D. Meyer	
26 MARK_SOUND_EFFECTS_09.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-2	SOAMPFX	Mark Dr. Bradley D. Meyer	
27 MARK_SOUND_EFFECTS_10.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-1	SOAMPFX	Mark Dr. Bradley D. Meyer	
28 MARK_SOUND_EFFECTS_11.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-2	SOAMPFX	Mark Dr. Bradley D. Meyer	
29 MARK_SOUND_EFFECTS_12.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-1	SOAMPFX	Mark Dr. Bradley D. Meyer	
30 MARK_SOUND_EFFECTS_13.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-2	SOAMPFX	Mark Dr. Bradley D. Meyer	
31 MARK_SOUND_EFFECTS_14.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-1	SOAMPFX	Mark Dr. Bradley D. Meyer	
32 MARK_SOUND_EFFECTS_15.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-2	SOAMPFX	Mark Dr. Bradley D. Meyer	
33 MARK_SOUND_EFFECTS_16.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-1	SOAMPFX	Mark Dr. Bradley D. Meyer	
34 MARK_SOUND_EFFECTS_17.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-2	SOAMPFX	Mark Dr. Bradley D. Meyer	
35 MARK_SOUND_EFFECTS_18.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-1	SOAMPFX	Mark Dr. Bradley D. Meyer	
36 MARK_SOUND_EFFECTS_19.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-2	SOAMPFX	Mark Dr. Bradley D. Meyer	
37 MARK_SOUND_EFFECTS_20.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-1	SOAMPFX	Mark Dr. Bradley D. Meyer	
38 MARK_SOUND_EFFECTS_21.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-2	SOAMPFX	Mark Dr. Bradley D. Meyer	
39 MARK_SOUND_EFFECTS_22.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-1	SOAMPFX	Mark Dr. Bradley D. Meyer	
40 MARK_SOUND_EFFECTS_23.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-2	SOAMPFX	Mark Dr. Bradley D. Meyer	
41 MARK_SOUND_EFFECTS_24.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-1	SOAMPFX	Mark Dr. Bradley D. Meyer	
42 MARK_SOUND_EFFECTS_25.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-2	SOAMPFX	Mark Dr. Bradley D. Meyer	
43 MARK_SOUND_EFFECTS_26.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-1	SOAMPFX	Mark Dr. Bradley D. Meyer	
44 MARK_SOUND_EFFECTS_27.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-2	SOAMPFX	Mark Dr. Bradley D. Meyer	
45 MARK_SOUND_EFFECTS_28.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-1	SOAMPFX	Mark Dr. Bradley D. Meyer	
46 MARK_SOUND_EFFECTS_29.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-2	SOAMPFX	Mark Dr. Bradley D. Meyer	
47 MARK_SOUND_EFFECTS_30.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-1	SOAMPFX	Mark Dr. Bradley D. Meyer	
48 MARK_SOUND_EFFECTS_31.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-2	SOAMPFX	Mark Dr. Bradley D. Meyer	
49 MARK_SOUND_EFFECTS_32.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-1	SOAMPFX	Mark Dr. Bradley D. Meyer	
50 MARK_SOUND_EFFECTS_33.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-2	SOAMPFX	Mark Dr. Bradley D. Meyer	
51 MARK_SOUND_EFFECTS_34.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-1	SOAMPFX	Mark Dr. Bradley D. Meyer	
52 MARK_SOUND_EFFECTS_35.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-2	SOAMPFX	Mark Dr. Bradley D. Meyer	
53 MARK_SOUND_EFFECTS_36.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-1	SOAMPFX	Mark Dr. Bradley D. Meyer	
54 MARK_SOUND_EFFECTS_37.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-2	SOAMPFX	Mark Dr. Bradley D. Meyer	
55 MARK_SOUND_EFFECTS_38.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-1	SOAMPFX	Mark Dr. Bradley D. Meyer	
56 MARK_SOUND_EFFECTS_39.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-2	SOAMPFX	Mark Dr. Bradley D. Meyer	
57 MARK_SOUND_EFFECTS_40.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-1	SOAMPFX	Mark Dr. Bradley D. Meyer	
58 MARK_SOUND_EFFECTS_41.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-2	SOAMPFX	Mark Dr. Bradley D. Meyer	
59 MARK_SOUND_EFFECTS_42.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-1	SOAMPFX	Mark Dr. Bradley D. Meyer	
60 MARK_SOUND_EFFECTS_43.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-2	SOAMPFX	Mark Dr. Bradley D. Meyer	
61 MARK_SOUND_EFFECTS_44.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-1	SOAMPFX	Mark Dr. Bradley D. Meyer	
62 MARK_SOUND_EFFECTS_45.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-2	SOAMPFX	Mark Dr. Bradley D. Meyer	
63 MARK_SOUND_EFFECTS_46.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-1	SOAMPFX	Mark Dr. Bradley D. Meyer	
64 MARK_SOUND_EFFECTS_47.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-2	SOAMPFX	Mark Dr. Bradley D. Meyer	
65 MARK_SOUND_EFFECTS_48.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-1	SOAMPFX	Mark Dr. Bradley D. Meyer	
66 MARK_SOUND_EFFECTS_49.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-2	SOAMPFX	Mark Dr. Bradley D. Meyer	
67 MARK_SOUND_EFFECTS_50.wav	RAIN	RAIN SHOWER PARALLEL BLOWING ON BRICK GARAGE. Occasional wind	Soundminer, CA	2 Storm H-1	SOAMPFX	Mark Dr. Bradley D. Meyer	

**Figure 3:** An example csv file used to import and easily assign metadata to sound files.

functionality with Soundminer HD, but save \$1,500 and can still re-import folders or entire directory structures. HD is intelligent enough to add only new or modified files using the Scan Sounds command, rather than duplicate your database every time you add new sounds. Furthermore, the company is considering Server functionality with HD in the future.

### WHAT IT IS

» Each section of Soundminer HD's interface can be resized and custom colorized to match your personal workflow. The main portion of the window displays your search returns in a series of customizable fields (see Figure 1). You have access to more than 35 fields of data (as seen in Figure 2), and can set up the main window to display whatever you deem most important. Soundminer will also, for a fee, create custom data fields specifically for your company or project. Below the main window is a waveform representation of the currently selected file. You can select a portion of the sound, change pitch and volume, audition it, and import it into your DAW. On the right side is a window with three tabs, including a metadata browser that displays more detail ascribed to a selected file. There are also Playlist and the aforementioned Transfer History tabs, so you can see what you have auditioned recently and what you have imported into your DAW. The toolbar at the top of the window features large, well-illustrated icons to cover most of the important tasks, from opening system preferences to engaging Soundminer's classic "roulette" mode to spotting a file to your DAW.

One of the most commendable features in Soundminer is the text import of metadata. As in V4, you can either enter metadata directly within the program or import a .csv file with all your metadata laid out in rows and columns (as in Figure 3). When executing a text import, the application asks for the .csv file and the location to start looking for sounds, and will then crawl through the folder structure until it finds all filenames referenced in the .csv. You can then embed the metadata back into the .wavs with a simple menu click. The search and embed times are fast and responsive, and make adding metadata to new sound effects easy and efficient. Since the multi-editing of metadata is absent except for a few fields like Category and Library, .csv files are the most effective means of editing large amounts of metadata. The drawback of Soundminer metadata is that the metadata itself is a proprietary format, so most other database software will not be able to read it.

Soundminer HD, like V4, requires a hardware dongle to use. Fortunately, it supports both HASP and iLok keys. I say fortunately only because many people in audio already use another piece of software that requires an iLok. The benefit is that there is no authorization per computer, so you can install Soundminer on all of your computers, and as long as you have your dongle plugged into the one you are using, it will work. The downside of course is that there is a dongle.

Soundminer HD is also truly cross-platform. While Soundminer has historically been a Mac-only tool, Windows had its own platform-

centric offering in BaseHead (see Damian Kastbauer's review of BaseHead 2.5 in the March 2011 edition of *Game Developer*). In the last year, BaseHead introduced a Mac version of its software as well. The difference is that the releases, interface, and features of the PC and Mac versions of Soundminer HD are identical, while in BaseHead, the interfaces differ slightly, and the Mac trails behind in both features and new versions (as of this writing, BaseHead 3's release is fast approaching for PC, while the Mac version 3 is still several months away).

### ONE PRODUCT, THREE FLAVORS

» While I have been speaking of the various features of Soundminer HD, I should be clear that there are three versions of the product, intended to allow basic use at a cheaper level and offer premium features for a premium price. The allure of being able to upgrade the product as you need is matched by the fact that it is significantly more affordable than V4 all the way up the upgrade chain. The upgrade chain is also flat in that the cost is always the same no matter what you start with or what you upgrade to. This method makes it fair whether you start with the biggest package or slowly work your way up.

Soundminer HD Basic (\$199), as the name and price point suggest, is the most barebones version of the software. With it you are limited to having no more than two databases, and can embed only Category and Description metadata to new files you add.

Soundminer HD Plus (\$399), which is the version I evaluated for this review, is most akin to Soundminer V4. With Plus you get all

the goodies: Spot to Timeline, 64-bit iZotope Sample Rate Conversion, batch processing, editing of all metadata fields, unlimited databases, and more.

And then there is Universal (+\$100 to either version), which offers only one additional feature: cross platform compatibility. You can get either Basic or Plus with Universal and can upgrade from one to the next until you own the holy mecca of the Soundminer HD series: HD Plus Universal. Anyone using both Mac and PC should embrace this concept, as up until this point a true feature-complete, cross-platform sound database has not been available.

### WHAT ELSE HAVE YOU GOT?

» The most important aspect of a sound database application is the search feature, and Soundminer offers myriad ways to sift through your sounds. There is the standard search field, in addition to wildcard searches and Boolean searches (a space is AND, a comma is OR, and a minus is NOT. You can also encapsulate multiple commands within parentheses for more specific searches). For further refinement, you can “lock” your current results by pressing the lock icon in the toolbar and continue to search down from just the current results. You can lock your search to smaller and smaller subsets by pressing the lock button again, and can also use the back button to step back to previous searches. A right-click on a field in the search window or the metadata pane allows you to easily search for similar files. For example, you can click on the category of a sound and Soundminer HD will return all records of that category in your database.

Soundminer HD also plays host to some new features, which the company intends to fold back into V4 in the future. These new features are perhaps a paradigm shift for most people used to searching through their libraries by typing out things like “explo\* (debris, boom, big) -glass.” The focus here is less on typing, more on clicking. The biggest new search feature is called LaunchPad, a

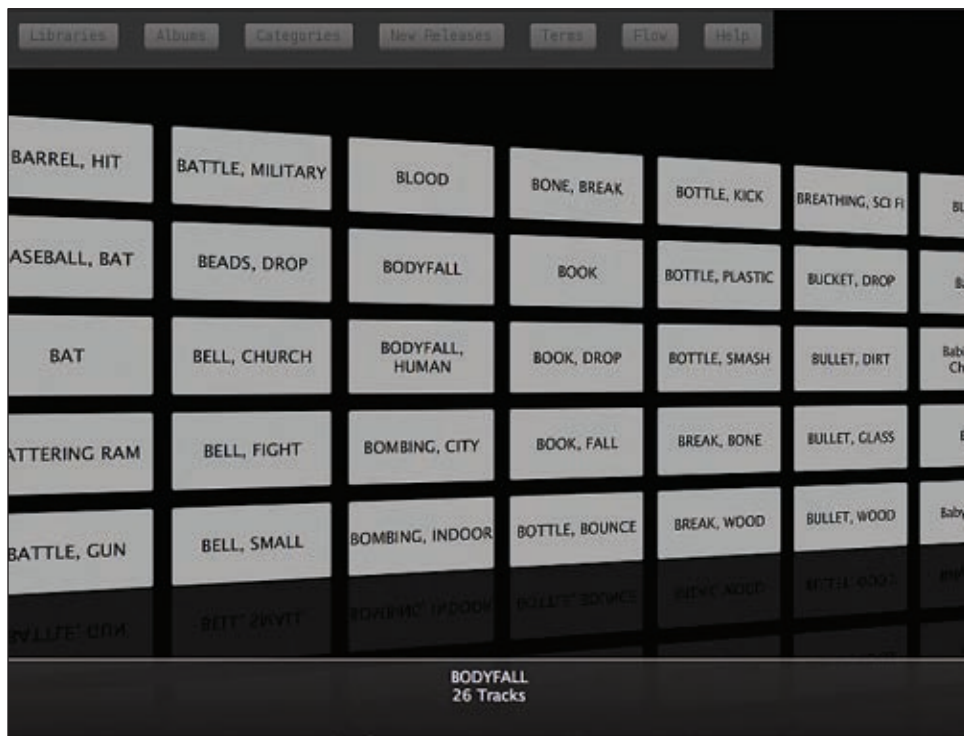


Figure 4: The new LaunchPad, a visual sound database browser.

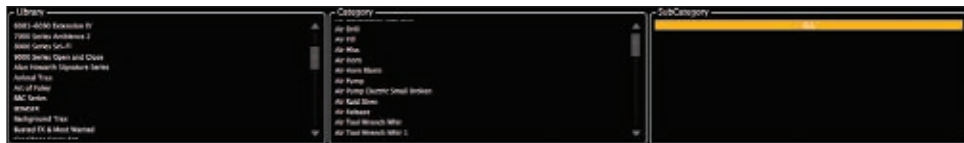


Figure 5: The Live Filter browser, another way to visually pare down your sound choices.

visual browser for your sample libraries (see Figure 4). With LaunchPad, you can select multiple items from a chart of boxes based on your libraries, categories, frequent search terms, and more. Truthfully, I still find it much quicker and more precise to type a few words to search for sounds. Those with large libraries will have a longer time navigating to the proper boxes to select them than they would typing in the proper search word. I can see this feature being improved in the future such that it becomes a quick and easy way to refine searches, but for now it seems a bit unwieldy with a large library. There is also a live filter browser (see Figure 5), which provides another graphical means to pare down potential selections. Hitting the eye icon will bring up three customizable tables

where you can select your various libraries, categories, subcategories, and more, to help you track down what you are looking for.

Still not sure Soundminer HD is right for you? Like BaseHead, the company offers a 30-day trial. An iLock (or HASP key) is required, but it's a great way to put the product through its paces and see if its features match your needs. For the record, I evaluated the full Plus version on the Mac and a 30-day trial on the PC. They are identical in features, look, feel, and version numbers, although as indicated, the behavior with certain unsupported applications differed.

Soundminer HD is not intended to replace V4; Soundminer Inc. is very adamant about that. V4 will remain its flagship (albeit Mac-only) product with some exclusive, potentially important

features like VST and ReWire support and a customizable thesaurus. But if you're looking for a more affordable tool with most of the features you really need on Mac, PC, or both, look no further than Soundminer HD. It takes the power and core functionality of the venerated Soundminer V4 and packs it into a cross-platform tool at a more reasonable price point. New studios, growing audio departments, and cross-platform developers take heed: Soundminer HD is a great way to enable access to your audio assets from wherever you need them. 🎧

**REV. DR. BRAD MEYER** is audio director at Free Range Games, where he makes lots of sounds and writes more code than he should be allowed to. Contact him at [brad@bradleymeyer.com](mailto:brad@bradleymeyer.com)



## MAIN CONFERENCE TRACKS

TUESDAY, OCT 11 - THURSDAY, OCT 13

- Business & Marketing
- Customer Experience
- Design
- Production
- Programming
- Monetization (sponsored)

## SUMMITS

MONDAY, OCT 10 - TUESDAY, OCT 11

- Game Narrative
- GDC Virtual Items
- Smartphone & Tablet Games

## PLUS

- Game Career Seminar
- Game Writing Tutorial
  - » Expo Floor
  - » 2nd Annual Choice Online Awards

Join us at the 9th Annual **Game Developers Conference® Online 2011**, October 10-13, for four days of world-class content across 120+ sessions, covering a wide-ranging scope of online and connected game development topics led by leading industry experts.

## FEATURED KEYNOTE



**Playing Well With Others - How PopCap Creates Compelling Social Game Experiences**  
John Vehey (PopCap)

## SUMMIT HIGHLIGHTS

### Just Go: A Roundtable Q&A with Valve's Writers

Erik Wolpaw, Marc Laidlaw, Chet Faliszek, Ted Kosmatka, and Jay Pinkerton (Valve)  
 Game Narrative

### COMBAT ARMS Postmortem: The Art of Selling Guns

Jungsoo Lee (Nexon America)  
 GDC Virtual Items

### Successful Publishing on Smartphone Platforms: iOS vs. Android

Giordano Bruno Contestabile (PopCap Games)  
 Smartphone & Tablet Games Summit

## SESSION HIGHLIGHTS

### DEFCON: A Basic Guide to Crisis Management

Valerie Massey (CCP Games)  
 Customer Experience

### Designers are Human Too - Causes of Poor Design Decisions

Tom Cadwell (Riot Games)  
 Design

### GARDENS OF TIME: Reinventing Hidden Object Games for Facebook

Eric Todd and John Hsu (Playdom)  
 Programming

### MARVEL SUPER HERO SQUAD Online Postmortem - An MMO For the Whole Family in Under Two Years

Jay Minn and Jason Robar (Amazing Society)  
 Production

### Engineering CITYVILLE

Robert Zubek (Zynga)  
 Programming

# GDC 11 Online

**ONLINE REGISTRATION ENDS OCTOBER 7TH REGISTER NOW AND SAVE UP TO 25%!**

## Game Developers Conference® Online 2011

October 10-13, 2011 | Austin Convention Center | Austin, Texas

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

For the latest news, contests and promotions visit us on: [twitter](#) and [facebook](#).





# AUTOMATED OCCLUDERS FOR GPU CULLING

## HIERARCHICAL Z-BUFFER OCCLUSION CULLING AND HOW TO AUTOMATE OCCLUDER CREATION

THERE ARE LOTS OF OBVIOUS REASONS WHY PERFORMANCE CAN TAKE A HIT IN YOUR GAME, BUT IT'S WHAT YOU CAN'T see that could be hurting you the most. One of these areas is unseen geometry. The art of culling unseen geometry from games has evolved over the years—developers often use a combination of frustum culling, BSP, portals, precomputed visibility, and occlusion queries to prevent the game from rendering objects that are outside the player's view. One of the more interesting solutions to emerge recently is hierarchical z-buffer occlusion culling (hi-z culling). This algorithm provides us with a fast and nearly fixed-time solution to determining visibility for static and dynamic objects in the game world. It should not be confused with the hardware hierarchical z-buffer that is used to reject pixel quads.

In this article I'll introduce you to the hierarchical z-buffer occlusion culling algorithm, and explain how you can automate the creation of proxy geometry for the occluders to minimize impact on the productivity of your artists and level designers.

### BACKGROUND

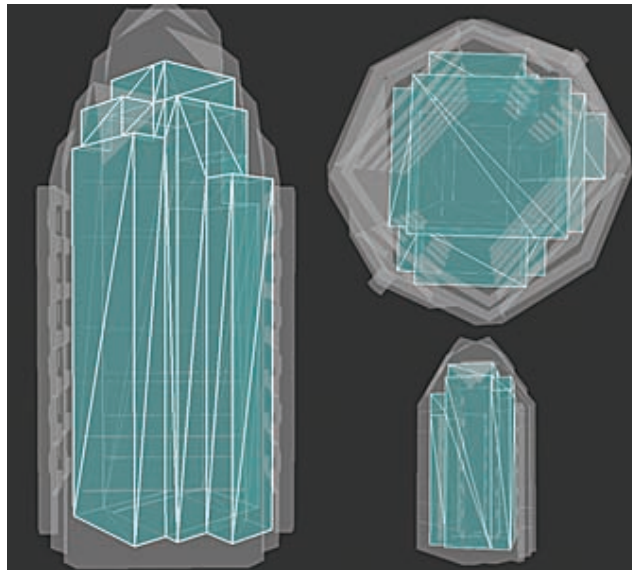
The first example of this hierarchical z-buffer occlusion culling algorithm can be traced to a paper called "Advances in Real-Time Rendering" from SIGGRAPH 2008 (Section 3.3.3). Since then, several high-profile games have extended this method to cull geometry in their renderers. TOM CLANCY'S SPLINTER CELL: CONVICTION and KILLZONE 3 both use this approach, and DICE has integrated it into the company's Frostbite 2 engine.

Before I explain how to automate generating occluders, let me first get into how the Hi-Z culling algorithm works, to illustrate why automating occluder generation is important.

### HOW IT WORKS

The goal for any culling algorithm is to determine the smallest visible or potentially visible set as fast as possible. GPUs offer their own occlusion culling system in the form of occlusion queries, which report the number of pixels that pass the z-buffer test, but that means having to render the exact mesh you're trying to avoid rendering. So in the end you don't save that much GPU time unless the object contains very complex materials.

A far better solution is to take a simpler representation of an object, like its bounding box, and test to see if just the bounding box is visible. In order to know if the object's bounding box is visible, we need to know about the large occluders that could block it.



Determining the Building Mesh (White) versus the Occlusion Mesh (Teal).

This is the core concept behind hierarchical z-buffer occlusion culling. First we render simple representations of the large occluding bodies (buildings, walls, terrain, and so forth) to a depth buffer. We render simple proxies because we need the rendering to be performed very quickly, and the original meshes are likely to be a little too complicated or may need too many draw calls to be useful.

After rendering the proxies to a depth buffer we create mipmaps (a hierarchy) of depth buffers. Using the hierarchy we can batch up the entire list of potentially visible object bounding boxes and test them all at once on the GPU at different levels in the mipmap, representing different granularities of depth buffer depending on the size of the object. Large objects use a very coarse

granularity depth buffer and are more likely to be visible. Small objects use a very fine granularity depth buffer and are less likely to be visible.

### 1 // RENDER OCCLUDERS

The first thing we need to do is render simple pieces of geometry representing the occluders in the level. You should start out by frustum culling the occluders on the CPU to reduce the draw call overhead. If you have a bake or cooking step in your level editor, that would be a great time to merge groups of the static occluder meshes into a single mesh to further reduce the draw call overhead.



Figure 1: Hierarchical Z-Buffer Downsamped Mipmaps.

The remaining potentially visible occluders will be rendered to a small [512x256] render target with a full mip chain. You don't have to use a render target with the same dimensions in your implementation, but using a power of two textures simplifies the code. You could also choose a smaller buffer, but you risk a loss in accuracy for smaller objects on the screen, potentially culling objects that would be visible in a full-size depth buffer.

DirectX 11 cannot create a hardware depth buffer with a mip chain. So instead we need to render the scene and write the depth values to the first mip level in our render target.

## 2 // DOWNSAMPLE THE DEPTH BUFFER

The next step is to fill out the hierarchy of z buffers by downsampling the first mip level that we just rendered to into the other levels of the mip chain. This will result in conservative depth buffers with coarser and coarser approximations of the occluders in the environment that we can test bounding boxes against (see Figure 1).

In order to perform the downsampling operation we can render a full-screen quad with a pixel shader that takes as input the previous mip level and conservatively downsamples it into the current mip level. Conservatively downsampling the depth values differs from standard mipmap downsampling because we have to preserve the highest depth value in a sample group of four pixels that are merging into a single pixel.

We do this because as the depth information is compressed and lost at each mip level we have to assume the worst case—that more things are visible—otherwise we may cull something that is actually visible. Therefore we take the furthest depth from that group so that we don't accidentally cull something that might be visible in one of the compressed pixels.

When performing the downsampling operation, one advantage of DirectX 11 over 9 is that you can constrain the shader resource view so that you can both sample from and render to different levels in the same mip chain. In DirectX 9 you had to copy a separate render target into the mipmap because there was no way to isolate a single mip level to sample from. The following pixel shader excerpt demonstrates how you can efficiently downsample four pixel groups conservatively.

```
Texture2D<float> PrevMip;
SamplerState PrevMipSS;

...

float4 PS( PS_INPUT IN ) : SV_TARGET {
    float4 vTexels =
        PrevMip.Gather(PrevMipSS, IN.texcoord);
    return max(max(vTexels.x, vTexels.y),
               max(vTexels.z, vTexels.w));
}
```

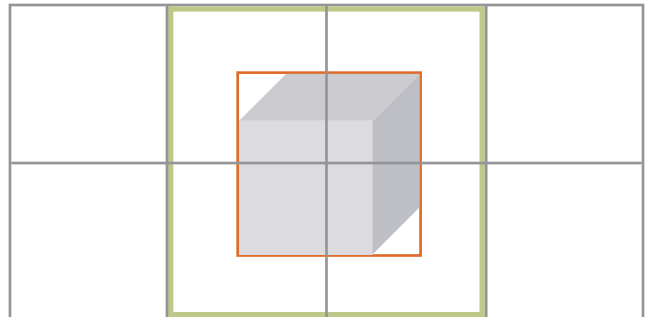


Figure 2: Screen Space Bounding Box (Orange), Sampled Pixels (Green).

## 3 // TESTING BOUNDS

In order to determine visibility we need to find the level in our depth buffer mip chain where the object takes up at most a four-pixel region. This is so we only have to test four points for visibility instead of testing every point that makes up the bounding box in screen space. This implies that large objects on the screen will sample from low-resolution mips, and small objects on the screen will sample from high-resolution mips. Even though the lower mip levels are coarser and thus less accurate, large objects are likely more visible on the screen anyway.

The testing algorithm is broken up into five discrete steps to perform in the compute shader:

1. Perform a frustum cull on the bounding box or sphere. If it is outside the frustum, it's not visible.
2. Determine the maximum screen space width or height of the bounding box (represented by the red box in Figure 2), taking the maximum value calculate the mip level to sample from so that the object takes up no more than a two pixel square region [the green box in Figure 2]. This value can be calculated using  $\text{ceil}(\log_2(\text{max\_screen\_size}))$ .
3. Take the four values making up the screen space bounding box of the object's bounding sphere or box and sample at those four locations in the mip level computed in step 2.
4. If the maximum sampled depth value is closer to the camera than the closest point on the bounding sphere or box then we can conclude the object is entirely behind the occluding surface and therefore not visible.
5. Store a float in a writable buffer representing visible (1) or not visible (0).

## 4 // READING THE RESULTS

After dispatching the compute shader to test all the bounds for visibility, you'll have time to do some processing on the CPU while you wait for the results. The amount of time obviously varies depending on the GPU and the number of bounds, but for a reasonably new GPU and 1,000 bounds the processing time should fall well below 1ms. Keep in mind, though, that the GPU could be performing previous work when your compute shader is dispatched, which would increase the time it takes to complete.

Once the compute shader finishes, it's just a matter of locking the writable buffer from steps 3–5, and copying the float values representing visible or hidden objects into a CPU-accessible buffer so that a final list of visible objects can be prepared for rendering.

## 5 // EXTENSIONS

The algorithm can be extended to also test whether shadows are visible as well. By creating a hi-z buffer from the large directional shadow casting

light in the scene (generally the sun) and using the hi-z buffer of the observer you can determine whether a shadow volume is visible.

By extruding the bounding box of an object in the light space hi-z buffer until all four sampling points collide with a surface, you can generate a new volume representing the shadow bounding box. This “extrusion” all happens in the shader; you’re not actually generating geometry for this volume. This new bounding box can then be tested against the observer’s hi-z buffer for visibility to know if an object off-camera is casting a shadow visible to the player. If the shadow volume is not visible, you don’t have to render the object into your shadow map.

## ART PIPELINE INTEGRATION

With any new change in the rendering pipeline a studio must consider what, if any, ramifications it will have on artist and level designer productivity. Because hi-z culling, in practice, renders proxy geometry to represent the occluders in a level, there is an inherent change in artist or level designer workflow because they must create this proxy geometry, unless they were already doing so for another purpose.

You may already have one source of these proxy meshes, in the form of physics meshes. In cases where your physics mesh is conservative (does not extend beyond the surface of the visible mesh) it can make for good occlusion proxy geometry. This is especially handy if your level has lots of destructive walls or buildings that also make for good occluders. One downside of this approach is that there are now many small occluders that must be updated.

Also, not every game has a physics mesh for objects that can be used as occluders. In these cases you might want to be able to automatically generate the occlusion geometry from the visible mesh, perhaps during export time in Max or Maya.

## GENERATING OCCLUSION VOLUMES

Before we generate any occlusion volumes let’s consider the important characteristics of good occlusion geometries:

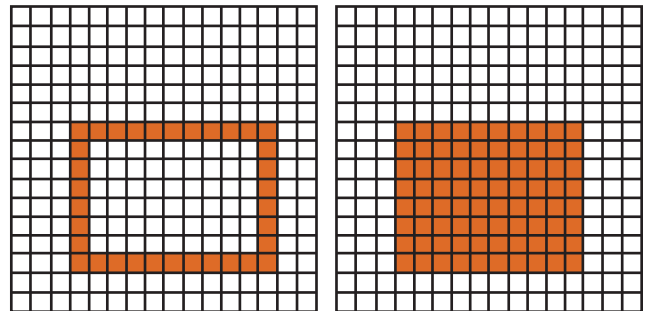
- **CONSERVATIVE** – Doesn’t extend beyond the surface of the mesh.
- **SIMPLICITY** – The occlusion mesh is made of very few triangles or is fast to render.
- **VOLUME CONSERVATION** – Closely matches the original mesh’s volume.
- **MOVABLE** – Some games have large moving occluders or destructible walls.

Normal methods of simplifying a mesh such as naive triangle simplification can cause both a significant reduction in volume as well as triangles penetrating the surface of the mesh. Neither of these are desirable outcomes.

What if instead we took the mesh and first converted it into a voxel representation. With a voxel representation we could perform our simplification operations on the volume structure of the object instead of on the topology of the object. The technique presented here does exactly that. However, it does have some caveats, and should be seen as a starting point to be extended and improved.

Let me start by summarizing the process:

1. Find all the voxels completely inside a mesh.
2. Find the voxel at the densest point in the volume.
3. Expand a box from this point until all sides collide with the mesh surface or another box.
4. Repeat 2–3 until you’ve consumed X% of the total volume.
5. Use a Constructive Solid Geometry (CSG) algorithm to merge the boxes you create.



Surface

Solid

Figure 3: Surface vs. Solid Voxelization.

## 1 // VOXELIZATION

First you have to find all the voxels completely inside the mesh. That way we can have complete confidence that anything we generate contained inside these voxels will be conservative. It also gives us a very easy way of quantifying the total volume and the volume remaining in the object.

The algorithm I used to perform my voxelization is laid out in a paper by Michael Schwarz titled “Fast Parallel Surface and Solid Voxelization on GPUs,” sections 3.1 and 4.1.

The paper talks about two voxelization types that we care about, surface and solid voxelization (see Figure 3). Surface voxelization gives us all the voxels that intersect with any triangles in the mesh providing us a shell of the mesh. Solid voxelization gives us the voxels making up the inner volume of a mesh. By determining both of these sets we can calculate the set of voxels completely inside a mesh by removing any voxel in the surface set that is also in the solid set.

## SURFACE VOXELIZATION

The algorithm for finding all the voxels that intersect with triangles on the surface of the mesh is fairly simple. You are performing a collision check between a triangle and a box. After taking each voxel and determining its size and position in model space you can perform a check against every triangle for collision.

## SOLID VOXELIZATION

In order to determine whether a voxel is part of the solid voxel volume we need to know if the voxel is inside the mesh. To do this we have to shoot a ray down the center of a column of voxels along one of the major axes. When the ray intersects with triangles along the column we can use the xor operator to set the correct inside/outside status of a voxel.

Imagine a column of voxels all starting at 0, and the ray “->” starting on the far left, with triangles “|” dividing the region into inside and outside areas. We would start with this:

-> 000 | 00 | 000

As the ray crosses the boundary, all the voxels beyond the first triangle are xor-ed with 1 and are now:

000 | -> 11 | 111

Continuing on, after the ray intersects with the next boundary we again xor the values with 1 beyond the new boundary, giving us:

000 | 11 | 000 ->

★ REGISTER WITH CODE ★

**GDCC11GDMA**

AND

**SAVE 10%**

ON ALL ACCESS AND MAIN  
CONFERENCE PASSES!

# GDCC China

**Game Developers Conference™ China**

November 12-14, 2011 | Shanghai Exhibition Center | Shanghai, China

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







0 represents outside and 1 represents inside; which is to say inside regions will be xor-ed an odd number of times, and outside regions will be xor-ed an even number of times.

One unfortunate limitation of this approach is that the mesh has to be watertight, so if there are any holes, this solid voxelization algorithm is likely to fail. This is one area that will require improvement, as much of what artists tend to create is not watertight.

## 2 // FIND THE HIGHEST-DENSITY VOXEL

In this step you will need to find the voxel that is the furthest away from any empty voxel, excluding any voxel already enclosed in an occlusion volume in step 3. Because the number of surface voxels is likely smaller than both the number of empty voxels and the number of total voxels you should create a list of surface voxels to test the distance against.

## 3 // BOX EXPANSION

Once you've found the densest voxel you should create a  $1^3$ -voxel size box at that location. You'll then proceed to iteratively expand each side of the box in voxel space until you can't expand any side of the box without entering an empty voxel, or another box that has already been placed.

As you verify each expansion of the box you'll mark the enclosed voxels so that the next time you choose the densest voxel you can exclude any already enclosed in a box.

The only approach I've tried is using a uniform expansion strategy. However a better approach might be to include a small amount of prediction allowing the growth of the box to maximize the number of expansions by not colliding early along one side if there is a lot of room to expand along another axis first.

While this approach will work well for axis-aligned objects, like buildings and other man-made objects, it won't work very well for more organic structures. So perhaps instead of expanding boxes, shrinking OBBs around the voxel structure of an object creating an OBB tree would be a better approach overall. However, this approach is completely untested and is only mentioned as an area of research for possible improvement.

## 4 // REPEAT 2-3

Continue to repeat the process of finding the high-density voxel and expanding a box until you reach a desired stopping condition. Examples of a stopping condition might include:

- An absolute percentage in volume is consumed.
- The volumes being constructed fall below a point of diminishing returns or are too small either in terms of total percentage or size in voxel space.
- A maximum number of boxes have been created.

## 5 // MERGE BOXES

We could stop here and create a single mesh out of these separate boxes, but rendering many overlapping individual boxes could potentially cause lots of overdraw. Instead, it would be better to merge the boxes together using mesh Boolean operations, commonly referred to as Constructive Solid Geometry (CSG).

The topic of CSG is too large to cover in this article, but a great explanation and implementation can be found in the book *Game Development Tools* in the article "Real-Time Constructive Solid Geometry," written by Sander van Rossen and Matthew Baranowski.

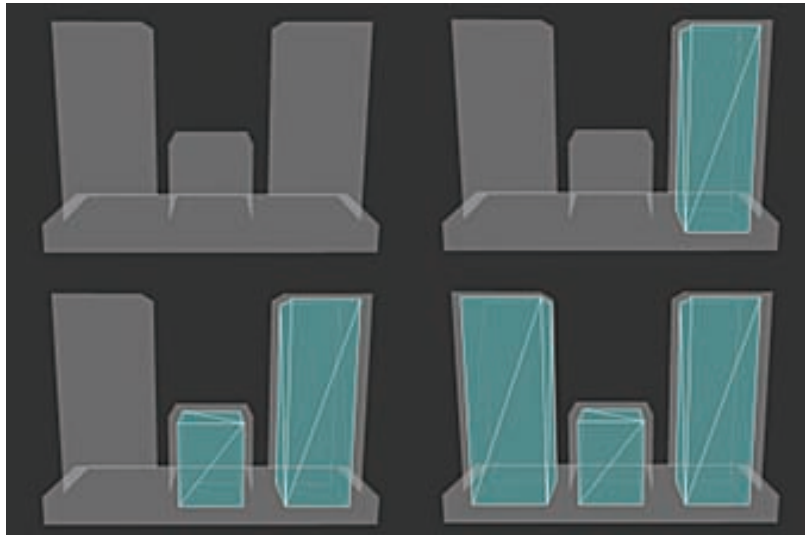


Figure 4: Box Expansion Time-lapse.

## THE RESULTS

When you're finished, the process can generate results like those in Figure 4. Starting with the voxelized mesh, boxes are expanded until enough of the volume has been consumed resulting in low-poly occlusion mesh that covers most of the original volume, regardless of the polygon count of the original mesh.

## CAVEATS

While this technique is a nice first step, it still has a long way to go. It's important to be able to handle non-water-tight meshes, and not having to worry about that makes things easier on the artist.

Speed can also still be an issue. Even after moving to the GPU to generate the voxel structure of objects it can still take several seconds to complete. Even though this process occurs at export time, it may still feel burdensome.

As always, the artists should be able to override anything automatically generated with a custom mesh of their own. This is important for structures like planes, where a voxel volume can't be generated for any internal structure, because a plane has none.

## JUST SCRATCHING THE SURFACE

Hopefully this article has been both informative and thought provoking. While there are still many areas that need improvement, there is some benefit even at this early stage.

Additionally, as GPUs become more advanced they may become capable of issuing draw calls of their own. This is already marginally possible with instanced meshes. If it becomes possible for an arbitrary draw call, the synchronization step could be removed from the hi-z culling algorithm, allowing the GPU to both determine what should be drawn and issue the command to draw it. 🙌

*The author wishes to acknowledge Mike Acton at Insomniac Games for encouragement, and starting AltDevBlogADay.com. The author would also like to thank Stephen Hill, Michael Noland, and Shaun Kime for their help and input.*

**NICK DARNELL** is a senior software engineer currently working at Activate3D, advancing the state-of-the-art motion gaming experience for Kinect. Previously he worked on tools and engine development for Gamebryo at Emergent Game Technologies. He is also an active member in the game development AltDevBlogADay community. Nick can be found online on Twitter @NickDarnell and his blog [www.nickdarnell.com](http://www.nickdarnell.com).

////////// We tend to talk about Bill Budge for his games. His influential RASTER BLASTER and PINBALL CONSTRUCTION SET blazed trails for user-generated content in later years. He was an early Electronic Arts designer, and is the gentleman wearing a single glove in the classic EA group photo. But games aren't really his passion. Bill Budge likes to code. He likes making the software that makes the games.

Post-EA, Budge left the traditional game industry for most of the late '80s and early '90s, toiling in solitude without much outside influence. But the world of code and tools was changing around him, so he returned to civilization with EA, then 3DO, then Sony, and now Google, continuing to build tools that enable other developers to do their jobs better. He's now in Google's Native Code division, working to make raw code viable in a browser.

Budge likes talking code almost as much as he likes coding. And so we did.

**Brandon Sheffield: It seems like you've always loved software tools. Where does that come from?**

**Bill Budge:** I don't know how well I can explain it. I've just always liked the idea of taking what you have, the pieces, and putting them together to build something that didn't exist before. That goes all the way back to being a little kid, playing with blocks and construction sets. Those were always the kinds of things I liked. We scrounged around for stuff and made go-karts. That was always really fun and exciting. Programming just feels to me like that really.

**BS: What made you want to construct in the virtual world versus the physical world, long-term?**

**BB:** You could build things in the virtual world that can't really exist. You could make a pinball construction set out of real parts with pinball pieces, but it would

cost thousands of dollars. But, yeah, in a computer, you can make something that costs \$20.

**BS: You've worked in several programming languages. What did you like the most in its time?**

**BB:** The early, early days when everything was in assembler, I think the Pentium architecture was really beautiful. The first Pentium. It was probably the most fun assembler language to do because it had two pipelines, and you could schedule instructions so that they were both busy. There's some amazing beautiful code. QUAKE has a little segment of code that's unbelievably beautiful. It uses every register and every cycle. The floating point unit, too. And it's all kind of choreographed so that everything is busy, and it's pumping out the maximum number of pixels. Beautiful, beautiful code by Michael Abrash.

**BS: Like early multithreading.**

**BB:** Yeah. It's at the very lowest level. You're keeping all the hardware busy. I really like C# for building big apps fast with big teams. I think it's awesome. A lot of people don't like Java or C# because they're kind of verbose, I think they sort of strike a balance between—I don't know how technical you want to be.

**BS: Go ahead!**

**BB:** I like C# and Java because they're very readable. It's a great language where people can communicate. They're not the most powerful languages, but really powerful languages have this problem that as you get more abstract, as the expression gets more powerful, it's harder for other people to understand. So, I think it's kind of a balance. Maybe to have a little bit of verbosity, you have to say things multiple times, more boilerplate, but large teams can collaborate. C++, if it's kept to a nice subset, is very powerful. Tools are unfortunately kind of crippled, but it's a reasonable language if you restrict yourself.

The problem with C++ and all the old C languages is a couple of

things. Preprocessor just makes it impossible for the tools to really understand the code reliably. There are very few environments that actually can deal with C++, where you can find every reference a symbol would say. In large programs, it's really important to be able to navigate and browse code powerfully, to be able to say, "I want to find every place that this symbol is used." When Chrome is loaded into Visual Studio, which is a fine tool, it does a terrible job. It just doesn't find all the things, so it really slows you down.

**BS: Kenta Cho, he's an indie guy, he programs everything in D. That's curious.**

**BB:** Yeah. I'm intrigued. The problem with D is it's just not adopted as C++, which, you know, has a lot of problems. A lot of people like D. I certainly would be rooting for it to be catch on. Another language that's even more different, and this is going to sound like marketing speak, but I'm intrigued by Go, because in a way they're ejecting a lot of problematic old things that are in object-oriented programming, like inheritance for instance. And I like the treatment of interfaces. It's kind of intriguing.

If they can get their performance story to where they are equivalent to C or C++, because I think right now in a lot of benchmarks are at maybe half the speed ... They'll claim that benchmarks are tweaked, you know, for the language, but ... So, when they get that performance story, which I think they will, even though it's a garbage-collected language, that will be a very compelling language.

I have this fantasy of writing a browser in Go because it's a really great language for concurrency. And it's clean like C#. There's no preprocessor, no header files, so it compiles really fast. I mean, those guys are insane about making a compiler fast, and that's very important also, reducing friction. When it takes a couple seconds to build, that's a huge difference from when Chrome takes like three minutes, and I go grab a snack. And that's on a special machine with

sixteen cores. That's really not bad for a C++ program on the scale of Chrome, but still, it's a big difference.

**BS: Why is Chrome built in C++ then?**

**BB:** Performance. Webkit was built in C++. You just have so much control over memory and how things are laid out. Webkit, I wouldn't criticize it. It's the fastest rendering engine for HTML on the web, and Chrome is built on that.

Yeah, control. It's a pain to debug. You might step through ... A point of view reference takes you to another file because it's a template. Whereas you have references just built into Java like C#. Nothing is as pleasant as building in C#. C# and Visual Studio, it just works great.

**BS: Do you think it would be possible to create like a game-oriented interface of some kind that could actually teach people programming, more than just pulling things together and assembling things?**

**BB:** Full-power programming? That's kind of a holy grail really. I don't think anyone's really come very close. There are specialized languages. Logo is the really interesting one for drawing. You can buy software to essentially program by building state machines and transitions, or defining actions and events, timelines with repeats, like Excel. There are little ways to do that, but really full-on programming where you're creating abstraction layers with data structures just seems very far away.

I mean, programmers think in a certain way. The computer is going to have to do a lot of translating at the very least. You know, for specialized domains ... There are programs where you can take business entities that programmers have built and defined like connections or processes. But it does seem like it's far away.

It would be wonderful, but I think programmers are just a group that thinks very differently. Machines at the low level are still the same. They have registers and memory, and they're fairly complicated. And

# BEAUTIFUL

software is complicated. There are a lot of possible ways to structure stuff, and the ability to simulate what's going on in a computer like that is something that a small number of people percentage-wise have.

**BS:** *It really is like learning a language, though, as well. I feel like games don't do a great job of teaching people language on a deeper level, but on a higher, Dora the Explorer-type level, they can teach you a bit of Spanish and how to string words together or something like that.*

**BB:** Really, if you wanted to teach in a certain domain, I mean, you can certainly create very interesting physics labs, for instance. I don't think that the possibilities for those kinds of programs have even been scratched. There can be some amazing stuff. I mean, somebody built a 6502 emulator. They actually went and got a 6502 chip under an electron microscope and sort of extracted the masks from just the images, and then they run a circuit simulation. You can run this in a browser. It's an amazing little tool. If you're interested in microprocessor design, this is a microprocessor with 3,000 transistors that would be very understandable by a person. [See [www.visual6502.org](http://www.visual6502.org) for more.]

**BS:** *It seems like everyone is kind of trying to go the other direction, where they're making it so you don't have to program as much. Unity and Unreal Engine, they both feel like they're moving in the direction to where you can get a lot more done with a lot less knowledge.*

**BB:** It's definitely easier with tools like that, I guess, if you can afford it. I'm actually a huge fan of them. I mean, this was exactly what I was doing at Sony. We built kind of a more one-level meta up, which is we built a bunch of C# components and samples that showed how to build tools. And the teams could take this stuff and very quickly make an animation blend tree editor or level editor. We had prototype samples that they could very quickly



Electronic Arts designers in 1983 (Bill Budge in upper right)

just start modifying, grow, and make their own. So, it was kind of a construction set. The technology is always getting better. Languages are better. Computers are faster. And, you know, there are tools like Unity.

It's a little chaotic. There are a lot of open-source editing tools. It takes a while, I guess. There's nothing really compelling like Maya where you can build your own modeler. I think that's something that's missing. There's the modelers, and they can have plug-ins, but if you wanted to make a very different kind of game engine that did procedural geometry and you wanted to define procedures, it's more difficult, and you're sort of limited to what's out there already. Or you have to write plug-ins, which is demanding.

**BS:** *Have you felt specifically like you learned anything from your time so far at Google?*


**BB:** Everybody's really smart there. I wanted to be challenged.

Everybody's really smart, at least in my build group, I think, just from conversations. It can become a little overwhelming. Everybody is trying so damn hard to be the smartest guy in the room, and you're just trying to keep your head above water. And, you know, you don't need to always be trying to impress everybody. But I've been impressed. Everybody is really competent. I was on Windows, so I wasn't really up-to-date on Linux and Mac OSX. I was programming a pretty high-level language and was not doing any low-level systems. Our group does much more system-level stuff. Native Client has got a lot of system level kind of coding in it. Chrome and C++, I haven't done much C++ because I was able to avoid it. And there are a lot of considerations to working with hundreds of people, all the collaboration.

I've learned a lot of nuts and bolts. I've done very little just

coding away like I used to, which I do miss. There's a certain discipline that you do need on really big projects. They have an excellent code-reviewing process. There are a lot of great people. I've been learning from people who are the top browser writers in the world. That part I love.

**BS:** *It seems the only person you can really try to impress is yourself. And that's impossible.*

**BB:** At the end of the day, I think all that matters is what have you done. It doesn't matter how smart you are, or how brilliant do you sound, or whether you sound like an academic paper when you talk. What really impresses me is people who have built things, who made things that really worked, who did something that nobody else thought would work, or followed their vision and made it real. That, to me, is very admirable—the only thing that counts. 



# REALM OF THE MAD DEVELOPERS

SOMETIMES A NICHE, EXPERIMENTAL GAME IS A LESS RISKY BET THAN ANY TRADITIONAL CONSOLE GAME

**WHAT WOULD THE TYPICAL PUBLISHING EXECUTIVE** do if someone came to them and said, "We've taken open-source 8-bit art and created a free-to-play (F2P) NetHack-inspired MMO with permadeath. You can attain the maximum character level in just 30 minutes of play. The game currently has no means of generating revenue and can accommodate only 60 concurrent players per server. Will you work with us on it?"

That's essentially the question posed to Spry Fox one year ago by Alex and Rob, co-creators of Wild Shadow Studios, when they presented us with an early build of REALM OF THE MAD GOD (RotMG). And I can guess what others might have said to them, because when we described the project to our contacts, the reaction was inevitably one of skepticism. Permadeath? In 2011? How the heck are you going to retain users? And surely you mean 600 concurrent players per server, not 60!

A mature company behaving in the stereotypically mature manner [i.e., risk averse] would have passed on RotMG. Its design was unconventional and terribly hardcore. It was written in Flash and unsuitable for distribution on consoles. It was relatively expensive to operate. Its developers did not have an established pedigree in gaming. The list went on and on. Better to get behind yet another first-person shooter with slick 3D graphics and call it a day.

We (Spry Fox) had a different perspective. Here's how we evaluated a risky project, managed that risk, and created a financial and critical success.

Alex and Rob were new to the game industry, but they had advanced degrees in computer science and substantial experience working on massively scalable systems at Google. They were smart, earnest, and motivated, and obviously willing to buck convention. So we partnered with Wild Shadow, with the goal of refining RotMG's design and implementing a coherent monetization plan. And, crucially, we treated the project not as a huge bet or investment that could not be allowed to fail, but as one of several experimental games in our portfolio. And as with all our other titles, we accepted—and embraced—the possibility of

failure, because we do not believe it is possible to truly innovate in any other context.

One technique we used to identify and fix major design issues in RotMG was to skip the private beta and iterate rapidly with a public audience throughout the majority of the development phase. Despite the public nature of our work, we regularly made dramatic changes to the game. Some of the changes were well-received by players; others caused riots on the RotMG forums. In each case, we did our best to explain our rationale to the game's slowly growing community, but we never stopped making big, public changes and observing the results. Most companies plug away at their games in secret, using (at best) highly controlled playtests to learn how to improve them. For an MMO, especially an MMO aspiring to any sort of originality, that's an incredibly slow and taxing process. We believe that our methods are faster and more effective.


We essentially ripped the beta label off of RotMG when we launched it on Chrome Web Store (CWS) on June 20, 2011. Google featured RotMG on the CWS home page as well as two subpages. Shortly thereafter, the game became the subject of an ongoing series of articles on the Rock, Paper, Shotgun web site, and was reviewed favorably by many other sites and individuals. The subsequent increase in traffic and publicity has been gratifying; we hope to leverage that and launch RotMG to great fanfare on many other online game portals in the months to come.

During the final phase of RotMG's public beta, the average user spent approximately \$1.68 per month. (There's really no such thing as an average user; the vast majority of players spend nothing, and a very small minority spend enough to support everyone else.) Post launch, monthly ARPU (Average Revenue Per User) peaked at \$3.40, partially because of an increase in retention, and partially because of high-value conveniences that new players tend to purchase soon after deciding they enjoy the game, like more inventory space (vaults) and the ability to use multiple characters concurrently (slots). We expect our ARPU to eventually settle somewhere north of \$2.00 but below \$3.40 until we:

- \* Enhance our methods of collecting revenue. With direct integration of a mobile phone payment solution, gift cards, and additional payment platforms that are locally relevant (i.e., outside the United States) we expect our ARPU to climb substantially.
- \* Identify additional premium features and/or items that we can sell in RotMG without jeopardizing the spirit of the game.
- \* Provide an optional subscription offering to our players, many of whom have told us that it is easier for them to sign up for a recurring billing plan than to pay piecemeal for enhancements in a game.

We believe that a monthly ARPU of \$5-plus is achievable for a game like RotMG. That's a heck of a lot better than selling games for 99 cents on iTunes! Our positive experience developing and commercializing RotMG is yet one more reason why we have abandoned the old world of disposable downloadable content and embraced the new (and much more satisfying) world of F2P games. The vast majority of players enjoy our content without ever paying a dime, yet we still earn more revenue than we would on XBLA, PSN, etc. What's not to like?

And most importantly, this business model enables us to keep iterating and innovating. The web is a huge and wonderful place where kooky ideas like RotMG can not only survive, but flourish. Technologies like Flash and HTML5, plus business models like F2P, make it entirely possible to bring original niche content to millions of people.

There will always be a big market for the next derivative console game. And there will always be big publishers too risk averse to make anything other than the next derivative console game. Savvy independent developers can and should aspire to do better than that. 

---

DAVID EDERY is the CEO of Spry Fox and has worked on games such as REALM OF THE MAD GOD, STEAMBIRDS, and TRIPLE TOWN. Prior to founding Spry Fox, David was the Worldwide Games Portfolio Manager for Xbox LIVE Arcade.



learn / network / inspire

**GAME DEVELOPERS CONFERENCE®**

SAN FRANCISCO, CA  
MARCH 5-9, 2012  
EXPO DATES: MARCH 7-9

**2012**

[www.GDCONF.COM](http://www.GDCONF.COM)

Call for submissions now open through early September.





# CONAN THE ILLUSTRATOR

## THE LIFE AND WORK OF FRANK FRAZETTA

**LAST MONTH, WE TALKED ABOUT THE MANY WAYS IN WHICH THE BUSINESS IS** changing, about how new genres of games and new markets are broadening the culture of game art and bringing us new visions and stylings. The new frontiers of games aren't beholden to the stale old genre obsessions and creaky style conventions of our broadsword-boobs-and-blasters past. This month, in the interest of fairness and balance, we're going to dive right back into the wellsprings of chainmail bikini culture. With the new Conan movie hitting theaters and bringing familiar fantasy tropes back to life with a vengeance, this is a good time for a little reflection on the past master of swords-and-sorcery artwork, the legendary fantasist Frank Frazetta.

For a whole generation of fantasy art fans, particularly those who came of age in the late 60s or 70s, the words "fantasy" and "Frazetta" are almost interchangeable. From plate-armored brassieres to rideable polar bears, many of the hallmarks of modern fantasy art are traceable directly to Frazetta's vision. When he died last year, comic artists and game artists all over the world paid homage to his impact on their work (see: <http://kotaku.com/5539115/video-games-finest-pay-tribute-to-artist-frank-frazetta/gallery>). Thousands of WARCRAFT fans even donned black tabards in mourning of an artist who, indirectly, helped give shape to their fantasies. While his work has never commanded a lot of respect in the official art world, its echoes are everywhere: he's been frequently described as the most influential artist of the latter half of the 20th century.

Of course, influence can be [to use a very appropriate metaphor] a double-edged sword. Precisely because so many of Frazetta's images became icons, they've also become fodder for countless imitations. The game business has certainly offered up its share of Frazetta homages, as you can see in [Figure 1](#). If you're a cynic, there's no more enduring measure of artistic success than creating a new cliché—and by that standard, Frazetta is one of the greatest artists ever. That kind of success, though, makes it hard to properly appreciate his work. It's difficult to judge it on its own merits, without reference to the vast army of imitations, rip-offs, and wannabes that came after.

### THE SORCERER'S APPRENTICE

Frazetta was born in Brooklyn in 1928. He grew up in an Italian neighborhood, and as a young teen studied with a classically trained Italian painter. Despite his early training, he always claimed that the great artists of the Italian Renaissance didn't interest him. His taste in the classics were the early Romantic paintings with strong colors, moody settings, and powerful geometric compositions—very much like his own later work. Frazetta was regarded as a young prodigy, and there was even talk of sending him to Europe for more advanced studies until the Second World War intervened. His teacher died before the war ended, and the art school folded. By age 15, he'd found employment doing clean-up work at a small comics publisher—and within a year he was working full time as an artist.

In those days, working illustrators had to bang out a lot of images to pay the bills. In addition to pulps, Westerns, and adventure stories, Frazetta also worked on animal funnies. His most lucrative gig in the comics business was, somewhat amazingly, the long-running hillbilly strip *Li'l Abner* (see [Figure 2](#)). It's odd to think of the creator of the *Death Dealer* toiling away on talking farm animals and soap-opera strips, but like most of us mere mortals, Frazetta spent a lot of time cranking out content to make a living, rather than invoking the muses. Troll through the forums on deviantART or similar sites today and cross-index them with game industry credits, you'll find a few equally surprising differences between "day job" art and the art that comes from within.

With static poses and mostly balanced compositions, Frazetta's 50s work gives few hints about the distinctive ambience of his classic swords-and-sorcery paintings. Still, behind those mid-century mannerisms something much more vital was lurking. George Lucas later claimed that Frazetta's covers for *Buck Rogers* and *Buster Crabbe* were the first inspiration for *Star Wars*, so something more than run-of-the-mill commercial draftmanship was evolving. Even in his work-for-hire comics, the men were a little beefier and more intimidating than the typical clean-cut junior varsity types in fashion then; the women were frankly physical in a



Figure 1: A small selection of games whose art show the influence of Frazetta.



Figure 2: Before Hyboria, Dogpatch. Frazetta spent most of the fifties working on the popular comic strip Li'l Abner.

way that was rare in those days of the Comics Code. One of Frazetta's few "solo" projects in the 50s was a short-lived Tarzan-like comic called *Thun'da* (see Figure 3), which clearly shows some of the hyperbolic anatomy and daring compositions that would eventually become de rigueur for barbarians from the deserts of Mars to the center of the Earth. Unfortunately, the work is spoiled for a modern reader by some horrendous racial stereotyping, so it's hard to lament the series' early demise.

### JOURNEYMAN TO THE CENTER OF THE EARTH

The decade-plus gap between *Thun'da* (1952) and Frazetta's seminal illustrations of Edgar Rice Burroughs and Robert E. Howard books in the mid-60s seems amazing to modern fans, but that's a trick of perspective. In the 50s and early 60s, fantasy subject matter was pretty far out of mainstream tastes. Swords and sorcery were regarded at best as kid stuff, fit for newspaper comics like *Prince Valiant* or B-movie serials. The internet—and the explosion of fan-driven otaku culture that lets people from every end of the earth argue endlessly about the technology of Gundams or how to correctly pronounce "Galadriel"—was far in the future. Revealing your interest in wizards, dragons, or axe-wielding barbarians back then was roughly equivalent to declaring yourself a furry today. There was, to put it bluntly, a lot more money in bodacious hillbillies. Although it's clear that heroic fantasy (particularly heroic fantasy with lots of exposed skin) was at the heart of Frazetta's personal aesthetic, it wasn't much of a way to make a living. In the end, he fell into it as a career almost by accident.

After quitting *Li'l Abner* in 1964, Frazetta found that comic styles had evolved; the penciling style that served him well as a journeyman had started to seem old-fashioned in the age of famed comic artists Jack Kirby and Steve Ditko. He had a hard time finding work in comics, so he switched media, doing painted advertising work and also contributing to the raunchy parody strip *Little Annie Fanny* in *Playboy* magazine (which was, whatever its other attractions, the only regularly running comic in the country that used painted frames and ran in full color). Ironically, he was fired from that job because Hugh Hefner thought the work was too sexually charged for *Playboy*.

However, his introduction to professional work in full-color illustration gave him access to new clients and new assignments. It turned out that Frazetta's big break didn't come from mighty swordsmen or even cheesecake: it was a caricature of Ringo Starr for *Mad* magazine. That caught the eye of Hollywood, and kicked off a string of movie-poster work (beginning, somewhat improbably, with Woody Allen's *What's Up Tiger Lily?*, but eventually including the iconic images for *Gauntlet* and *Mad Max*). Every poster earned him about as much as he'd previously earned in a year.

### MASTER OF MAYHEM

With more money and time, Frazetta pursued more painting work, which in those days meant paperback covers. The Frazetta era really dates from 1966, with his cover for a new edition of *Conan The Adventurer* (see Figure 4). In a single shot, it combines several Frazetta trademarks: overdeveloped musculature, lots of exposed, bronzed skin (particularly female), and meticulous attention to detail of costume and weaponry. It also shows very clearly his debt to Romantic paintings: bold composition with strong, saturated colors and dramatic rather than naturalistic lighting. Carefully rendered details, such as the filigree work on the weapons and jewelry, give the impression of fanatical precision, but in fact, most of the painting is loose, almost abstract.

The *Conan* cover was an instant hit. The *Conan* series in paperback went on to sell more than 10 million copies over the next decade, completely unimaginable for the time and genre. While Frazetta's art was of course not wholly responsible for that success, the names of Conan and Frazetta are now inextricably linked. But even fans who never saw the signature on those cover paintings were caught up in the sudden explosion of fantasy art. From a standing start in the mid-60s, images of half-naked barbarians, writhing monsters, and undressed warrior women blossomed all over. By the mid-70s you could find Frazetta's visual vocabulary on everything from magazines like *Heavy Metal*, to t-shirts, to album covers, and the sides of custom vans.

### CLICHÉ OR CLASSIC?

Fifty years later, it's hard to imagine there was ever a time when leather-clad sword-slingers and snarling beast-men were radical and subversive. In a world where we don't even joke about a former Conan being governor of California because it's funnier to joke that he was a killer robot, it's hard to appreciate how powerful these images were when they first appeared. In the cultural upheavals of the 60s, though, Frazetta's graphic break from the goody-two-shoes traditions of Hal Foster touched a jangling nerve. Maybe it was the subject matter; at a time when a lot of people felt modern society was coming apart at the seams, Frazetta's obsession with primal power



Figure 3: The 1952 comic *Thun'da* shows hints of Frazetta's dynamic compositions and exaggerated anatomy.



and violence seemed urgent and real. More likely, it was the outsized sensuality, made suddenly available at paperback prices to teenage boys too scared to ask for the adult magazines kept behind the shop counter.

That cheesecake factor is one of the big question marks over Frazetta's work. It's not impossible to find viewers who note how many of Frazetta's women are formidable warriors and claim the images are empowering rather than demeaning; however, it's a lot easier to find viewers who hate (or love) the paintings because of the way they exhibit women. It's a little easier for folks in our business to take a balanced view, because that iron triangle of power fantasies, sexuality, and violent emotion is still where a lot of us make our living. Some of us do it enthusiastically, others with resignation, but most of us understand the difference between the art we'd make for ourselves and the art we make to pay the bills. Frazetta had the happy knack of loving the things his core audience loved, and it was the feedback loop between his obsessions and those of millions of (mostly male, mostly adolescent) fans that gave his particular visual solutions their staying power.

Whether you find the subject matter timelessly primal or laughably clichéd, it's unfair to deny the virtuosity of the execution. Every choice Frazetta makes cranks the emotion in his scenes. Like the heavy metal bands that used his pictures for their LPs, he specialized in intensity rather than subtlety (see Figure 5). But like the best metal bands he knew how to achieve that intensity with economy of effort. His compositions are surprisingly idiosyncratic; rather than following simple lines or arcs, they are masses of twisting, boiling curves—a powerful trick for adding drama to a static image that goes all the way back to Michelangelo. In the same way, the details are carefully rationed. The central figures have delicate detailing, but most of the canvas is loosely, almost impressionistically brushed. This focuses the viewer's eye in the same way as a strong depth of field. The palettes are rich and strategically

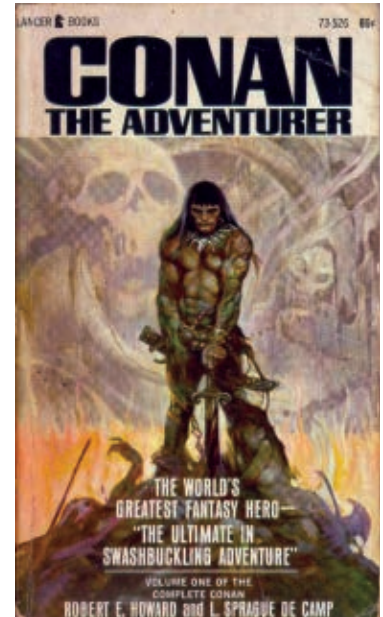
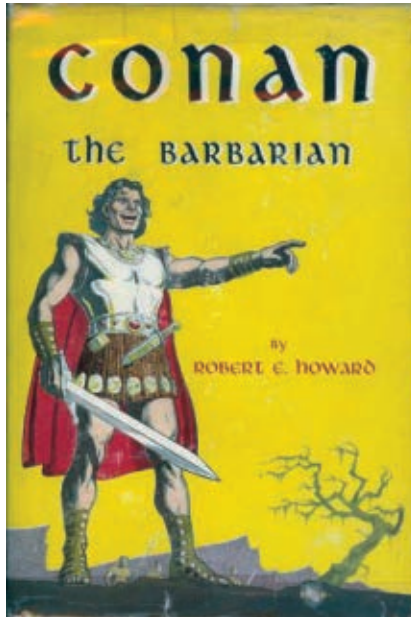


Figure 4: A great illustration of Frazetta's legacy: Two Conan paperback covers for two different eras in fantasy art. First image: Ed Emshwiller (1954). Second: Frazetta's groundbreaking cover for Conan the Adventurer (1966).

oversaturated, but the values are surprisingly harsh, with strong blacks and fewer midtones. If you compare, say, a scantily clad Frazetta Vampirella with an equally bodacious one from Julie Bell, you'll see that the Bell version has much more lush modeling and more realistic lighting, while Frazetta's is far more graphic, reflecting his years as a comics artist. All of Frazetta's key tactics enhance the melodrama of his scenes and underline their emotional tensions, where many more recent artists who use the Frazetta formula put their efforts into a glossy surface rendering that plays up the soft-porn side of the equation.

### TO LIVE WITH CROM

Frazetta passed away last year at the age of 82. Sadly, his last years were marred by a messy family dispute over his works; with individual paintings selling for more than a million dollars apiece, the infighting between heirs was vicious even before he died. These days it can even be difficult to buy prints due to legal wrangling over the rights to different works, and his museum has been closed while the legal and family issues are fought out.

Frazetta's passing also reignited a long simmering border war with the fine art world. The question of whether Frazetta was a "real artist" or merely an "illustrator" has sparked innumerable internet flame wars and snarky editorials. If you look only at the commercialism and the content, it's easy to dismiss him as a mere peddler of standard-issue adolescent daydreams. If you look at the technical accomplishments, though, it's hard not to be impressed. What you get out of the discussion depends a lot on what you bring to it.

For us these arguments are very familiar and, truth be told, a little beside the point. His influence is so widespread that you should learn about it whether you want to embrace it or hope to root it out. We're all children of Frank's revolution. [FD](#)



Figure 5: Pop culture transcendence.

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.





# A Life Worth Living

## SCOTT BRODIE LEAVES MICROSOFT TO START HEART SHAPED GAMES



*For four years, Scott Brodie was a producer at Microsoft in the XBLA group. While helping facilitate the games of indie developers was somewhat satisfying, he realized he needed to start pursuing his own indie dreams. So he left Microsoft and started his own company, Heart Shaped Games. The new company, which is focusing on casual and social games, but with a bit of an indie bent. As Brodie says, "There is no reason to spend another minute not doing exactly what I love."*

**BRANDON SHEFFIELD:** *What made you decide to go off on your own? Why now?*

**SCOTT BRODIE:** It was a combination of wanting to be more hands-on with design and development, and recognizing there were new models emerging that could make it viable for me to build smaller, more personal games. Before starting Heart Shaped Games I was a producer with Microsoft's XBLA team. I really enjoyed that role, helping indies realize their vision, but my background is more on the technical side of things and I was looking for an opportunity to be more involved in day-to-day development. The emergence of free-to-play, mobile, and all the new social platforms just made the decision to take a chance on myself easier.

**BS:** *The social platform is a challenging one—what do you feel is the future and potential of the platform?*

**SB:** Particularly with Facebook, when I look past the current crop of games, I see a platform that connects games with massive numbers of players through free-to-play, and that makes it more likely that people will be playing my games with their friends. From my time at XBLA, I learned that these are two big factors of success for online games, particularly the "playing with your friends" part. So seeing those things, and seeing how low the barrier to entry for

developers still is, I see a lot of potential in building games for social platforms.


I can't say I've thought about the future of social much, but I do think that game designers will have a lot of influence over where things go. We can try applying the design patterns we learned from building single-player and narrative-driven games, or embrace the unique aspects available to us now to make new game types. As of late I've tried hard to start from a place of, "okay, the platform is based around relationships between people, how can I make a game that's all about that?" I'm guilty of staying in my comfort zone at times on HERO GENERATIONS, but as I gain more experience with building online games, I get more excited about the new design landscape that has opened up to me.

**BS:** *HERO GENERATIONS, is as much about rearing a family as it is about questing. How do you give the concept of virtual family building meaning for the player?*

**SB:** HERO GENERATIONS is very much a design that generates meaning as a whole. It's difficult for me to analyze each system in isolation, because on their own they are relatively shallow. Each turn in the game is an entire year of your hero's life, so each action is more symbolic of how and where you choose to spend your time, versus being a deep simulation of raising a

family and so on. But over the course of multiple generations, you start to see how your history of choices affects the opportunities available to your virtual kids and grandkids, and I think as a result all future choices are engendered with more meaning. It all really stems from this core tension created from having the choice to pursue personally beneficial things like quests at the expense of investing in your future family.

**BS:** *You've talked about "real value" in games versus grinding and leveling. What to you represents real value?*

**SB:** I'd like to make games that are meaningful and memorable because they challenged a player in a way that led to some sort of personal growth. Ideally that personal growth would extend beyond the skills required for the game, and become useful as a model for doing well in other aspects of life. I think that's what I mean by "real value." By contrast, if you look at the casual game space I've entered, the popular design aesthetic seems to revolve around making sure players are always feeling good, feeling special, and making forward progress. That usually comes at the cost of the type of longer-term depth that is only revealed when a player is confronted with something they don't quite understand yet. 

## who went where

Massimo Guarini, director of recent third-person thriller SHADOWS OF THE DAMNED, has left developer Grasshopper Manufacture to set up his own company called Ovosonico Productions, with the slogan "the sound of bold ideas."

Former Veraz Networks CFO Al Woods has joined mobile ad network Tapjoy, where he will contribute to the company's rapid growth plans following a recent \$30 million funding round.

MASS EFFECT 2 and 3's lead gameplay designer, Christina Norman, recently left BioWare for a lead designer position at LEAGUE OF LEGENDS house Riot Games.

Leading U.S. retailer GameStop has appointed Shane Kim, formerly Microsoft Game Studios vice president and overseer of franchises such as HALO and GEARS OF WAR, to its board of directors.

Blizzard Entertainment and Cryptic Studios veteran Bill Roper has recently joined the Disney Interactive Media Group to oversee the creative direction for the publisher's Marvel-based games.

## new studios

A new development studio called ThreeGates has opened on the Swedish island of Gotland, which is set to focus on online multiplayer and co-op titles and will deliver its first game AETHERUS later this year.

Two former 2K Games designers have founded a new independent development studio called Uppercut Games, with the aim to create AAA-quality mobile and downloadable games.

Irrational Games co-founder Jon Chey has announced the formation of Blue Manchu, a new studio devoted to making the "more genre, nichey games that I really want to make."

Super Rewards founder Jason Bailey along with POT FARM developers Josh Nilson and Galan Akin have secured \$1.5 million in angel funding to found a new Vancouver social games studio called East Side Games.



# GOOD GRIEF!

## DESIGNING FOR THE DARKER SIDE OF MULTIPLAYER

It's natural for the designer to think of himself as being at odds with the player; he is, after all, the guide on the player's journey through the game experience. He needs to ensure the game is interesting and challenging throughout. However, the rise of multiplayer gaming has resulted in a different strain of fun—griefing—that offers a new, more adversarial dimension to the designer/player relationship.

Griefing is the act of players exerting power over other players inside the game space, usually (but not always) in a manner that is independent of the rules and goals of the game. Beyond this, the definition gets a little tricky. While grievers frequently cheat, a player can (and often does) grief without doing so. While grievers often engage in direct player confrontation, they can also use backhanded channels. In most games, griefing is seen as a negative, but a precious few are actually built on it as the backbone.

Griefing is most commonly associated with massively multiplayer online games, but nearly any genre with multiplayer has the potential for grief. Designers of shooters have to deal with spawn campers and team killers. Facebook game designers have to deal with players who repeatedly kill the same opponent over and over again. Any game with a chat room has potential for trash talk so toxic that more mild-mannered players may be dissuaded from playing at all.

Even tabletop games run the risk of grief, such as the *Pandemic* player who insists on not helping the cause, or the D&D player who tries to burn down the town. Most of these smaller games have some recourse: the owner of the game or server can stop inviting a jerk to play. Designers of MMOs and other games with "public" servers, on the other hand, have to come up with alternative solutions, or deputize themselves as wardens defending the peace.

### THE EXPRESSION OF POWER

» Griefing is about power. Killing a player 20 times in a row by spawn camping him is addictive fun not because you win the deathmatch, but because he can't stop you. This same strain of fun can be found, albeit with a very different tone, by those who dance naked on mailboxes in *Orgrimmar*—while generally harmless to some, it's irksome to others, and that is the motivation. Or, for that matter, that unique friend on your Facebook roll who insists on telling the world the endings of all the *M. Night Shyamalan* movies.

When I was working at Ubisoft, a game called *URU* was being developed at a sister studio. This was meant to be an MMO version of the classic puzzler *MYST*. I had several earnest discussions with the designers about what form of griefing might take place in a game with no combat. A top concern was puzzle-griefing—players standing by puzzles, shouting out the answers as players came near. And while it is amusing to imagine players wasting time shouting "blue triangle, green circle, red horseshoe!" before you start moving puzzle pieces around, one can imagine the devastating effect it would have on those who loved *MYST* for its core gameplay.

One doesn't have to attack or kill another player to grief. Sometimes, not being able to be killed is the griefing tactic. Consider *Fantasy the Bard*. In the early days of *EVERQUEST*, *Fantasy* started a career on the PvP-heavy, "no rules" server of *Sullon Zek*. He carefully kept his character below level 6, where due to the game mechanics he couldn't be attacked. This worked wonderfully in his favor when he led gigantic enemy creatures (i.e., "trained" them) to other

players completely unable to retaliate in any way. Due to complaints from the most hardcore of the hardcore *EVERQUEST* players, the "no rules" server had to make an exception to deal with *Fantasy*, with customer service intervention.

### A CULTURAL THING

» What constitutes griefing depends highly on the culture found within a game. The designers must identify the culture they want to cultivate within the game, because promoting or defending it is going to be as much a part of their job as laying down levels or designing the combat math. The cultural cues the designer puts into the game can have a huge effect; for example, designing a testosterone-drenched game with scads of violence and/or women as sex objects (say, a *BULLETSTORM* or a *DUKE NUKEM FOREVER*) is going to attract a very different audience, and have very different griefing thresholds, than online components for, say, the *SETTLERS OF CATAN* on Xbox Live game or a more casual MMO like *MAPLESTORY* or *FREE REALMS*. In the latter, the bar for what equates griefing will be much lower, but the former will likely have a lot more players eager to test the boundaries.

Games designed for a younger market have to consider the solutions to griefing as a core component of their game, especially due to the uniquely ominous turn that sort of activity can take for a young audience. *WIZARD101* and many other games go so far as to not allow most players to chat with each other without special safeguards; most players playing the freeware version can communicate only using preset words and phrases. It's still possible to annoy or frustrate another player, but these avenues

are limited primarily to in-game mechanics. For the most part, parents can feel at ease when their kids are playing *WIZARD101*—an important consideration for that market.

Some games, however, have a much more expanded version of what is reasonable behavior versus what is griefing. Many MMOs in particular have attempted to embrace a libertarian ideal for the genre, encouraging players to do whatever the game allows, and then allowing players to use the threat of force to correct problems on their own. While this ideal often captures the imagination of the player base, the reality of griefing often catches up with them. A couple months before *ULTIMA ONLINE* came out, there was an article on the game's web site giving the helpful hint that one player could lead the guards out of town, while his friends go on a player-killing rampage throughout the city. After the game launched, the development team spent a lot of time trying to quell these sorts of strategies.

### ENTER EVE

» This is not to say that design of a more permissive game is not possible. *EVE ONLINE* initially launched with a permissive attitude, and has not wavered much from that design stance since. It has been rewarded amply, both in the press as well as in the marketplace.

One example is the player *Cally*, an entrepreneur who started the *EVE Intergalactic Bank*. He took players' money for safekeeping, offering it out to other players as loans, complete with interest rates and payment plans. At some point, he got bored, stole all of the money (by some estimates, worth more than a hundred thousand real dollars), spent it all on a souped-



EVERQUEST: LORDS OF EVERQUEST.

up capital ship, and then proceeded to spend his time mocking those who formerly trusted him across the web.

In most games, this would be perceived as an enormous example of catastrophic griefing, and countless customer service hours would be spent trying to correct it. But CCP, the makers of EVE, decided that in their vision of the game, such activities are fair game, so long as the money was earned through non-exploitative means [i.e., through legitimate game mechanics]. Their attitude: buyer beware.

The history of EVE is a rich tapestry of such scams and acts of personal betrayal, and they succeed in keeping the game on the front page of *Wired*. Such events keep the idea of the game fresh and exciting. EVE ONLINE is a game where anything can happen, but it is also a wild frontier. The game is, in many ways, defined by where it draws the line on griefing.

## ENDING THE GRIEF

» Griefing can be hard to define and even harder to stop, largely because different players (and sometimes designers) can have wildly varied ideas within the same play space of what actually constitutes griefing. Roleplayers in ULTIMA ONLINE considered the guild of players who roleplayed elves to be grievers—because everyone should know that there are no elves in the canonical ULTIMA! Note that, in this case, the elves were only griefing accidentally.

When considering griefer activities inside your game, some simple rules of thumb are:

- \* Be clear and consistent. Be sure that players understand what is expected of them, that your game's mechanics support your decision, and be sure your designers, community personnel, and customer service all have the same idea of what is permissible or not permissible. This is usually very hard the first day a potential griefing tactic is found.
- \* If you don't want it, block it. Designers should catch themselves when they think "oh, players will never do that," especially if what you're talking about is a way for one player to negatively impact another player's experience. Players can be extremely clever when it comes to finding ways to annoy and frustrate other players.
- \* You get the behavior you incentivize. If you give players achievements or other rewards for griefastic behavior, you will teach players that this sort of activity is permissible and encouraged! Be sure you're not giving rewards for spawn-camping or killing the same player 20 times in a row, unless that's really the culture you want to encourage.
- \* Anonymity breeds grief. The less attached players are to their character and reputation, the more likely they will engage in grief tactics. This is one area where subscription-based MMOs have an advantage over free-to-play games and

public server FPSes, but even then the designer needs to be wary that the player who grieves is typically far less attached to his character than his victim.

Griefing cannot be completely stopped in any multiplayer game, but it can be managed to an occasional distraction. Failure to take griefing seriously can result in your game getting a negative reputation, and can result in a community where "good" customers flee, leaving a more unruly customer base to manage.

A wise producer once described a griefer as "a customer who costs me more money than he gives me." This simple description is an incredibly effective way for designers to think about grievers and their potential impact on the community as a whole. It is also a useful reminder to designers that designing a multiplayer game is not just about laying out maps and designing weapons, but also about shaping the culture and permissiveness of their game. 🎮

---

**DAMION SCHUBERT** is the lead systems designer of *STAR WARS: THE OLD REPUBLIC* at BioWare Austin. He has spent nearly a decade working on the design of games, with experience on *MERIDIAN 59* and *SHADOWBANE* as well as other virtual worlds. Damion also is responsible for Zen of Design, a blog devoted to game design issues. Email him at [dschubert@gdmag.com](mailto:dschubert@gdmag.com).

## GAME DEVELOPERS CHOICE ONLINE AWARDS HONOR KESMAI FOUNDERS, SOE'S EVERQUEST

Organizers of GDC Online have revealed that the 2011 Online Game Legend Award at the second annual Game Developers Choice Online Awards will go to John Taylor and Kelton Flinn, the founders of Kesmai Corporation and the creators of several seminal online games including ISLAND OF KESMAI and AIR WARRIOR.

In addition, the second persistent online game to be inducted into the Choice Online Awards Hall of Fame will be Sony Online Entertainment's beloved fantasy game EVERQUEST, a still-operating title that was one of the most important early 3D MMOs. Both recipients will be honored during the Choice Online Awards ceremony, taking place October 12, 2011, at GDC Online in Austin.

The special awards are a celebration of the iconic developers and games that have had a significant impact on the shaping of the now massive online games category. Honorees were selected through open nominations from the online game community and the distinguished GDC Online Advisory Board.

The board includes game industry veterans, leaders, and luminaries such as Zynga Austin's John Blakely, Blizzard Entertainment's J. Allen Brack, BioWare Mythic's Eugene Evans,

Playdom's Raph Koster, Nexon's Min Kim, and Riot Games' Brandon Beck.

The GDC Online organizers chose John Taylor and Kelton Flinn for the Online Game Legend Award in recognition of their achievements as game creators who have made a permanent impact on the craft of developing online games; and for having provided a launch pad for many other accomplished developers' careers for nearly 20 years.

Flinn started writing multiplayer games while a college student, competing against and eventually collaborating with Taylor. This work resulted in the CompuServe-hosted MUD ISLAND OF KESMAI, which they wrote following graduate school in the early 1980s. In the ASCII-based world of ISLAND OF KESMAI, real-time text communication between players—while solving dungeon quests and participating in combat—was a major influence on many of the online games to come.

In addition to that title, Kesmai Corporation, founded in 1982 by Taylor and Flinn, sought to commercialize the game ideas they had cultivated, and produced titles including MEGA WARS III, HARPOON ONLINE, and multiplayer BATTLETECH. They also produced 1986's online multiplayer AIR WARRIOR, one of the

first persistent titles to employ 3D (albeit wireframe) graphics, and a title largely credited with the birth of the online flight sim genre.

Key players in today's online game development, such as industry veterans Gordon Walton (STAR WARS: THE OLD REPUBLIC), Bill Dalton (ULTIMA ONLINE), and many others have gone on to successful careers in online games after starting out at Kesmai Corporation with Taylor and Flinn.

The game voted into the Choice Online Awards Hall of Fame, EVERQUEST, is now in its twelfth year of continuous operation. The fantasy MMORPG title launched in March 1999, and was one of the first to introduce the concept of guilds and raiding within an online world.

Since then, millions of gamers have ventured into its fantasy MMO universe, which has hosted 17 expansion packs as well as multiple spinoff console titles, sequels, novelizations, and even a board game.

The game has seen such enduring popularity among fans that it still runs independently alongside its sequel, EVERQUEST II. Members of the original SOE EVERQUEST launch team will be on stage alongside the current operators of the influential MMO title to accept the Hall of Fame award.

## GDC CHINA REGISTRATION NOW OPEN, FIRST TALKS REVEALED

GDC China organizers are proud to announce that registration is now live for the 2011 show. The first batch of announced sessions includes speakers from the show's Social and Indie Games Summits on topics spanning Western-focused development and essential indie dev tools.

Taking place November 4–6 at the Shanghai Convention Center, the event will once again serve as the premier game industry event in China, bringing together influential developers from around the world to share ideas, network, and inspire each other to further the game industry in the region.

This year, GDC China will feature tracks on Online Game Development & Business, Global Game Development, and Social Games, with notable summits on Independent Games and Mobile Games. In addition, the show will boast a two-floor exhibition hall, and will once again host the Independent Games Festival China for the third year running.

All the sessions at GDC China will be simultaneously translated between English and Chinese during the event, and the

following talks are the first among many to be revealed for this year's show.

As part of the Social Games Summit, Ubisoft Chengdu project manager Xiaojuan He will host a talk dubbed "Project Managing a Social Game for the Western Market," revealing how the China-based Ubisoft branch developed the Facebook title CASTLE & CO. for Western markets.

He's lecture will focus particularly on her role as project manager throughout the game's development, outlining the team's approach to designing a Western-focused game, the problems encountered during production, handling team morale, post-launch support, and more.

Over in the Independent Games Summit, Ye Feng, co-founder and CTO of independent iOS developer Coconut Island Studio (FINGER BALANCE, IDRAGPAPER) will discuss indie dev tools in "Brewing Your Own Game Engine - The Pros & Cons of Using Open Source Software to Rapidly Develop Cross-Platform Indie Games and Tools."


In this lecture, Feng will weigh the pros and cons of using home-brewed tools versus

middleware, and will show attendees how his studio used custom tools to its advantage to streamline and improve the development pipeline. In addition, he will suggest a number of tools that make indie development easier.

Also within the Independent Games Summit, Dongxu He of indie studio Gamegou (WATERMELON!, SOCCER STEALERS) will host "Product Development Strategy of Indie Studios," a talk that will explain why indie developers cannot always rely on creative ideas to survive in today's market.

Drawing from his experience at Gamegou, He will discuss how developers should balance innovation, tenacity, and good business sense to survive in a mobile and indie market that remains in constant flux.

With registration for GDC China now open, interested parties can go to the event's official website to start the registration process and gain access to the numerous talks, tutorials, and events the show will have to offer.

For more information on GDC China as the event takes shape, please visit the official GDC China website ([www.gdcchina.com](http://www.gdcchina.com)). 



# PLAYING THE FIELD

## A SURVEY OF POPULAR FIELD RECORDERS

### FIELD RECORDING IS ONE OF THE ETERNAL

frustrations of game sound design.

Commercially produced sound libraries are, by their very nature, available to anyone with the money to afford them, and subsequently are used all too frequently by everyone in game, television, and film sound. The more dangerous a sound is to record, the more likely designers are to reach for existing libraries. The result is a media soundscape littered with the crackle of identical flames, the spark of indistinguishable electrical arcs, and the boom of generic explosions. Field recording, on the other hand, is a great opportunity to go out into the world and capture unique sounds that no other libraries have—not to mention soak up some much-needed vitamin D. Unfortunately, it can be time consuming, expensive, or both.

In an effort to encourage us all to get out of our offices and record new and unique source material, I asked audio professionals for recommendations of their field recorders of choice.

### DO-IT-YOURSELF

» When you talk to sound recordists about field recorders, the first company that always gets mentioned is Zoom. In 2005, the Japanese company Zoom began to market what has today grown into a line of versatile and inexpensive digital field recorders. Zoom's best field recorders are its H series, and every recorder from the H1 to the H4n includes multiple built-in microphones with a specific focus on stereo recording.

At just under \$300, the H4n is the unit most often mentioned by the field recordists when talking Zoom's products. The H4n includes two built-in condenser microphones in an X/Y pattern and a windscreen for exterior recording. In addition, the unit also has two balanced XLR mic inputs, a tripod mount, phantom power, onboard pre-amps, a headphone jack, and runs off of either batteries or a power supply. Though rugged and small enough to be a handheld unit, the H4n's small size also results in a small digital display. Perhaps the biggest feature of the H4n is its built-in four-track recorder that allows for simultaneous quad recording.

Rather than using tape or disc media, the H4n records to compact SD or SDHC flash

media cards with a maximum recording time of over 15 hours at 28-bit/96kHz and up to over 50 hours at 16-bit/44.1kHz. Smaller SD cards cost under \$20 while a 32GB card will run about \$85.

Another handheld option is the D&M Professional PMD661. Much like the Zoom H4n, the PMD661 runs off of SD or SDHC compact flash cards and comes complete with XLR inputs, onboard pre-amps, built-in playback speakers, a tripod mount, phantom power, and records up to 24-bit/96kHz. Like the H4n, the PMD661 includes built-in stereo microphones, but it also has a larger OLED display and the ability to store user-defined presets. The unit runs about \$600, and it should be noted that Oade Brothers ([www.oade.com](http://www.oade.com)) offers a variety of upgraded pre-amp modifications with the unit for an additional \$150.

After Zoom, the most frequently mentioned company is Sound Devices. Since 1998, Sound Devices has been specializing in field recording units, and the company's flagship line is the 700 series. Unlike the H4n and the PMD661, the Sound Devices units are larger and more expensive than their handheld competitors.

The most comparable unit to those previously mentioned is the Sound Devices 702. Again, the unit records onto compact flash cards, though it can also be connected to external FireWire drives. The 702 can record up to 24-bit/192kHz, which is an improvement over the H4n and PMD661. In addition to onboard pre-amps and phantom power, the 702 includes onboard limiters, high-pass filters, and a programmable LED metering display. The 702, though, does not have built-in microphones. The Sound Devices 702 is generally regarded as the top of the line in field recorders, and so is significantly more expensive than the other options, at a price tag of about \$1,800. For an additional \$700, the Sound Devices 722 includes all the features of the 702 plus an internal 160GB hard drive.

These aren't the only options. Other units like the Fostex FR-2 or Sony's PCM-D1 exist, offer comparable feature sets, and are cheaper than the 702—though not by much. Audio gear mainstays like Roland and TASCAM have their own lines of field recorders, too. Field recording apps even exist for iOS and Android smartphones. The units listed above, though,



represent the most common field recorders in active use by working sound designers.

### DID-IT-THEMSELVES

» Of course, not everyone has the time, the budget, or the interest in field recording. Thankfully, the digital distribution model of the internet has given rise to an entire market of boutique sound effect libraries created by those who enjoy getting out into the urban sprawl with a microphone close at hand. These online storefronts are usually focused on smaller libraries than you would expect from behemoth sound distributors like Sound Ideas. As they're smaller in size, they also tend to be smaller in scope and focus on a particular function like bangs, blips, or blasts of air. If you want to get some new sounds without getting your hands dirty, check out Chuck Russom's work at [chuckrussomfx.com](http://chuckrussomfx.com), the electricity and whooshes of [tonsturm.com](http://tonsturm.com), or the booms, beasts, and metallic bonks of [boomlibrary.com](http://boomlibrary.com). <sup>41</sup>

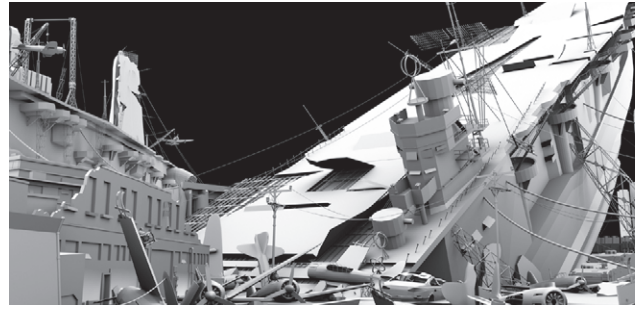
---

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



# ACADEMY of ART UNIVERSITY®

FOUNDED IN SAN FRANCISCO 1929 BY ARTISTS FOR ARTISTS



## TAKE CLASSES ONLINE OR IN SAN FRANCISCO

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

## ENROLL NOW

---

### EARN

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

### ENGAGE

IN CONTINUING ART EDUCATION COURSES

### EXPLORE

PRE-COLLEGE SCHOLARSHIP PROGRAMS

---

**WWW.ACADEMYART.EDU**

**800.544.2787 (U.S. Only) or 415.274.2200**

79 NEW MONTGOMERY ST, SAN FRANCISCO, CA 94105

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

*\*Acting, Architecture (BFA) and Landscape Architecture  
degree programs not currently available online.*

Visit [www.academyart.edu](http://www.academyart.edu) to learn about total costs, median student  
loan debt, potential occupations and other information.

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



<http://wompgame.tumblr.com>

# WOMP!

THE STUDENT TITLE WOMP! OFFERS AN UNUSUAL TAKE ON THE CONCEPT OF COUCH-BASED MULTIPLAYER. RATHER THAN GIVING EACH PLAYER AN AUTONOMOUS AVATAR, THE GAME PUTS EACH USER AT THE HELM OF A SINGLE APPENDAGE OF A BIZARRE UNDERWATER CONTRACTION. PLAYERS MUST COOPERATE TO SWIM, FLOAT, AND PROPEL THEIR WAY THROUGH THE GAME'S SERIES OF UNDERGROUND CAVERNS, LEST THEY FIND THEMSELVES FIGHTING FOR CONTROL OVER THEIR AWKWARD, CLUMSY VESSEL.

DURING DEVELOPMENT, INDUSTRY PROFESSIONALS TOM FRISINA OF THATGAMECOMPANY AND BRENDA GERSHKOVITCH OF SILICON SISTERS HELPED GUIDE THE STUDENT TEAM AT THE CENTRE FOR DIGITAL MEDIA IN VANCOUVER, EVENTUALLY HELPING THEM WIN THE BEST STUDENT GAME AWARD AT THE 2011 CANADIAN VIDEO GAME AWARDS. HERE, WE DIVE INTO THE DEVELOPMENT AND CREATIVE PROCESS BEHIND THIS UNUSUAL CO-OP TITLE.

**TOM CURTIS:** *How did your team come together to work on this project?*

**WOMP! Team:** Our last semester of school allowed us to work on our own projects. However, to do so, we needed to create our own team, get industry support, pitch the idea to the school, and get approval. Team formation was quite organic; all of us wanted to create a fun, playable game at the end of the semester, and we naturally gravitated to each other to form our current team. It was important for us to have a wide range of perspectives while sharing the same goal.

**TC:** *What process did you use to design the game? Sketches? Paper prototyping?*

All of the walls in our project room at the Centre for Digital Media were covered with whiteboards. It's a great way to doodle and draw without any constraints or inhibition. Our early sketches and initial drawings were up on the wall, and we had the chance to refine them as we saw fit. We didn't make a paper prototype, but instead jumped right into a digital version of our game, which was playable within a week.

**TC:** *What was your playtesting process like? How did testers react to your game's concept?*

We had a playable prototype up within a week, and put it in front of people. There is a fear in any creative industry of sharing an unpolished product, but we didn't worry about that, and in return received valuable feedback early on. We were lucky that our school gets tons of visitors!



**TC:** *What were the biggest challenges you faced during development?*

The initial feedback we received from playtesters, faculty, and mentors was very positive, but we wanted to continue to push ourselves as well as our game. Keeping the momentum was challenging at times.

**TC:** *How did your mentors Tom Frisina and Brenda Gershkovitch get involved in the project?*

Tom Frisina, who works at thatgamecompany and is the former VP of EA Partners, is a faculty member at the Centre for Digital Media. He is a very inspiring teacher who supports his students beyond the classroom. He became the obvious choice for us. And Brenda got involved after giving a fantastic presentation at the school. As a local Vancouverite,

she was able to offer feedback throughout the development process.


**TC:** *How did your mentors influence the project?*

Tom and Brenda are veterans in the industry. Aside from general encouragement, both were instrumental when it came to preparing a budget and pitch for our presentation to Microsoft Game Studios. Both Tom and Brenda helped us understand the business side of making games when we were focused more on the art of making them.

**TC:** *After WOMP!, what are your plans for the future?*

After finishing our master's degrees in the spring of 2010, we were the lucky recipients of a small development grant from Microsoft Game Studios under the guidance of Don Mattrick.

That gave us another few months to polish the game, but soon we needed to look for full-time employment.

Currently all five of us work on separate projects. Dave Marhal is lead level designer on X-MEN DESTINY at Silicon Knights. Salvia Dhall and Bryant Drew Jones are building a start-up funded by Canada Media Fund, focused on bringing digital games into playground environments. Karin Schmidlin took a marketing position at Allegra False Creek, and Bryan Clarke works at Qwick Media as a front-end Flash programmer. But after winning Best Student Game at the Canadian Video Game Awards earlier this year, we got the momentum to pursue opportunities to bring our game to a wider audience. 

# 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



© 2011 Full Sail, LLC

### Campus Degrees

#### Master's

- Entertainment Business
- ▶ Game Design

#### Bachelor's

- Computer Animation
- Creative Writing for Entertainment
- Digital Arts & Design
- Entertainment Business
- Film
- ▶ Game Art
- ▶ Game Development
- Music Business
- Recording Arts
- Show Production
- Sports Marketing & Media
- Web Design & Development

#### Associate's

- Graphic Design
- Recording Engineering

### Online Degrees

#### Master's

- Creative Writing
- Education Media Design & Technology
- Entertainment Business
- Internet Marketing
- Media Design
- New Media Journalism

#### Bachelor's

- Computer Animation
- Creative Writing for Entertainment
- Digital Cinematography
- Entertainment Business
- ▶ Game Art
- ▶ Game Design
- Graphic Design
- Internet Marketing
- Mobile Development
- Music Business
- Music Production
- Sports Marketing & Media
- 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

# MAKE MORE ENEMIES

Game Design at VFS lets you 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, Canada, a world hub of game development.

The LA Times named VFS a top school most favored by game industry recruiters.



VFS student work by Aldo Martinez Calzadilla



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

"VFS prepared me very well for the volume and type of work that I do, and to produce the kind of gameplay that I can be proud of."

DAVID BOWRING, GAME DESIGN GRADUATE  
GAMEPLAY DESIGNER, SAINTS ROW 2



## 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)



**DOWNLOAD AND PLAY THE LATEST STUDENT GAMES!**

Download your FREE digital edition of the 2011 *game developer* Game Career Guide online at: [www.gamecareerguide.com](http://www.gamecareerguide.com)



## ADVERTISER INDEX

COMPANY NAME	PAGE	COMPANY NAME	PAGE
Academy of Art University	52	Montreal International Game Summit	21
Academy of Interactive Entertainment	17	Perforce Software	C2
Blizzard Entertainment	6	Rad Game Tools	C4
Epic Games	C3	TwoFour54	3
F+W Media Inc.	17	Vancouver Film School	54
Full Sail Real World	54		

*gd Game Developer* (ISSN 1073-922X) is published monthly by United Business Media LLC, 303 Second Street, Suite 900 South, South Tower, San Francisco, CA 94107, (415) 947-6000. Please direct advertising and editorial inquiries to this address. Canadian Registered for GST as United Business Media LLC, GST No. R13288078, Customer No. 2116057, Agreement No. 40011901. **SUBSCRIPTION RATES:** Subscription rate for the U.S. is \$49.95 for twelve issues. Countries outside the U.S. must be prepaid in U.S. funds drawn on a U.S. bank or via credit card. Canada/Mexico: \$69.95; all other countries: \$99.95 (issues shipped via air delivery). Periodical postage paid at San Francisco, CA and additional mailing offices. **POSTMASTER:** Send address changes to *Game Developer*, P.O. Box 1274, Skokie, IL 60076-8274. **CUSTOMER SERVICE:** For subscription orders and changes of address, call toll-free in the U.S. (800) 250-2429 or fax (847) 647-5972. All other countries call (1) (847) 647-5928 or fax (1) (847) 647-5972. Send payments to *gd Game Developer*, P.O. Box 1274, Skokie, IL 60076-8274. Call toll-free in the U.S./Canada (800) 444-4881 or fax (785) 838-7566. All other countries call (1) (785) 841-1631 or fax (1) (785) 841-2624. Please remember to indicate *gd Game Developer* on any correspondence. All content, copyright *gd Game Developer* magazine/United Business Media LLC, unless otherwise indicated. Don't steal any of it.



# OUTSOURCING LIFE

A SURVEY OF THE JOYS OF BEING AN OUTSOURCER

Outsourcing firms are an often-overlooked part of the industry, but their efforts are vital to the development process of today's biggest games. So instead of taking them for granted, pause for a moment to consider and appreciate the special brand of pain they often experience.

## ASKED TO MAKE BLIND STABS

» Hi there Outsourcing Company,

We wanted to reach out to you regarding doing some work for our latest game! It's going to be a large project with a ton of content and work to do, so we're looking to partner with the best outsourcing companies in the industry (like you) to help us get there! To tell you about the game, it's an epic fantasy—no, wait, it's actually science fiction ... actually ... it's sort of a science-fiction game, but with fantasy elements, with a little bit of steampunk in there, too. Basically imagine sci-fi fantasy steampunk meets FINAL FANTASY, but more gritty, like *Black Hawk Down*. Does that make sense? Anyway, could you do maybe three or four concept paintings for us to help us nail down our visual style?

Thanks,  
Developer

## ADHERENCE TO A NONEXISTENT PLAN

» How much work do you think you'll have for us over the course of the project?

We don't know yet. It could be a little, or it could be a lot. It depends on what we decide to do, who we've been able to hire, ongoing negotiations with our publisher, and the whims of our creative leads. By the way, quality is very important to us. If we do

end up going with you, we don't want your "B" team on our game. You'll provide your best talent for our job, right?

*Of course, we'll do our best to free up our top resources just as soon as it's time to get started. That leads to our next question: what's the time frame we're talking about here?*

Good question. Our current plan has us ready to begin outsourcing assets sometime between March of this year and November of next year. That's assuming everything falls into place, of course.

*Okay, that's a pretty big window. I guess you can let us know when you're ready, and we'll pick up the conversation then?*

Hold your horses there, potential partner! That won't work for us. We need you to commit people now for the work that we may or may not have!

## NICKEL-AND-DIMING

» Hi Outsourcers,

Thanks for your prompt bid. We've reviewed it with the management team here, and there are a few questions that came back about your quotes. Let me know when you can answer them, and I'll pass them along.

1. We noticed that you charged the same amount for each character model in our character list. However, three of the characters

we sent you are dwarves. Dwarves are a lot shorter than the other characters, so they should cost less to make—half of what a human character costs, maybe? Please rebid on those items.

2. Under "Creatures," you have full animation sets listed for the mounted elephants and the dire wolf packs. Can't you just reuse the wolf animations for the elephants? We understand there may be a little adjustment involved, but they're really the same basic shape. Just trying to see if there can be a little cost savings there.
3. Also, about animation—you have personnel costs for animators here. What about instead of paying animators, you just do mocap? It's cheaper, right? And mocap looks better anyway.

Thanks!

## COMPETITION

» How are things going? I just wanted to let you know that we found these other guys who say they will do all the work for free. I mean, we like your work for us so far, but we were wondering if you could match or beat that competitor's offer before we continue.

*"Free" is a tough offer to beat, but I can maybe run some numbers and see if—*

So what I'm hearing is, you don't want our business. Is that it? I should reiterate that we do like your work, but there are literally 20 or 30 million other outsourcing companies that are falling all over themselves to work with us.

*Well, if you do like our work, then we hope you'll feel that paying us is worth it.*

But don't you know that we are a BIG-DEAL DEVELOPER making a BIG-DEAL GAME? The sheer power of being able to tell people you worked on the game is incalculable! I mean, can't you work for us on the basis of our name alone?

## THE SOLUTION TO EVERYTHING

» Hey guys, we've fallen a bit behind on where we'd like to be on this game, and we wanted to contact you to see if you could supplement our capacity and help us ship on time!

*Alright. What kind of work were you looking for?*

Nothing too fancy. We've already got our Unreal Engine officially licensed and everything, so all we need now is just little bit of concept art, modeling, texturing, rigging, animation, environment art, weapons, vehicles, props, effects, cinematics, gameplay programming, level scripting, sound design, music—you know, just a little bit of all of those things so we can get our game done. If you could take on those tasks, that'd be great! Thanks! 🙌

---

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



## MOBILE GAMING GETS INTO ACTION WITH UNREAL ENGINE 3 FOR IOS

As if we didn't already spend enough time on our smartphones and tablets, a new wave of high-end handheld gaming built with Unreal Engine 3 is giving us even more reasons to tap and swipe.

Many new iOS projects are being developed with the Unreal Development Kit (UDK), the free edition of UE3, which requires little to no programming and only costs \$99 per studio license (US dollars), with royalties only kicking in after \$50,000 in net earnings.

Crystalised is one studio that wasted no time picking up UDK for *Desert Zombie: Last Stand*. When asked why, production manager Cam Phillips said, "It was a no brainer, really! UDK is an industry-leading, AAA games engine that's been made accessible to smaller development houses with a royalty-based licensing deal. No other engine on the market can compete with UDK on mobile platforms."

Phillips explained that his team has UE3 experience through previous projects and education, and "with the advent of iOS support for the Unreal Engine, the time to enter the marketplace and have a resounding impact on the mobile gaming space has never been better."

MacLife describes the third-person shooter looks "absolutely gorgeous" and Touch Arcade says it looks "pretty incredible." Phillips says several notable UE3 tools have really helped Crystalised achieve the desired look and feel of *Desert Zombie: Last Stand*.

"Unreal Matinee and Unreal Cascade have been pivotal in creating visceral, dramatic and explosive gameplay moments that really bring the game to life and give players a real 'sense of war,'" he said.

"The new Simplygon tools included in the latest iterations of UDK have helped us to optimize assets on-the-fly to squeeze the very most out of the iOS device's performance, allowing us to deliver more bang for your buck. Amazing tool and a massive time-saver!"



from *The Dark Meadow*

in the editor window have really helped us keep a consistent feel across all platforms," said Phillips.

In addition, Phosphor Games Studio is using UE3 to develop *The Dark Meadow*, a first-person action adventure that will keep players on the edge of their seats as they attempt to escape an abandoned hospital by tracking down an elusive witch and battling her nightmarish minions by way of sword and crossbow combat.

Kotaku summed up its early impressions of *The Dark Meadow*: "It sounds like the developers reached into a bag filled with the best horror-suspense games, mixed in a little first-person shooter, and sprinkled it liberally with other games that have had tremendous success on the iOS platform: *Angry Birds*, *Infinity Blade*, and so on. I've absolutely no problem with that."

We can't wait to get our hands on both games.



Canadian-born Mark Rein is vice president and co-founder of Epic Games based in Cary, NC. Epic's Unreal Engine 3 has won Game Developer magazine's Best Engine Front Line Award five times along with entry into the Hall of Fame. UE3 has won three consecutive

Develop Industry Excellence Awards.

Epic is the creator of the mega-hit "Unreal" series of games and the blockbuster "Gears of War" franchise. Follow @MarkRein on Twitter.

WWW.UNREAL.COM

### UPCOMING EPIC ATTENDED EVENTS

**GDC Online**  
Austin, TX  
October 10-13, 2011

**Montreal Int'l Game Summit**  
Montreal, Quebec  
November 1-2, 2011

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





# THERE'S NOTHING LIKE BEING A RAD GAME DEVELOPER



It feels like you have **SUPER POWERS** when you use Bink, our amazing, super **FAST** video and audio codec.



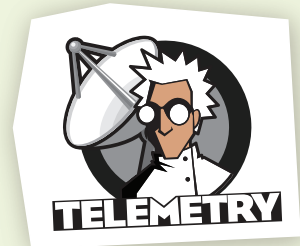
You'll **LOVE** using Granny, our run-time dynamic **3D ANIMATION SYSTEM**.



There's nothing like Miles, our multichannel sound system.



And our newest tool, Telemetry, is a library for profiling, tuning and visualizing application **PERFORMANCE**



So it's obvious, there's nothing like using **RAD** tools.

Well, nothing except having a killer



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