



CMP

United Business Media

GDC:06 PREVIEW GUIDE INCLUDED

JANUARY 2006

# gamedeveloper

THE LEADING GAME INDUSTRY MAGAZINE

❖ 2005 FRONT LINE AWARDS

THE VERY BEST TOOLS  
FOR GAME DEVELOPMENT

❖ BETTER GRASS BY DESIGN

MAKING VERTEX BUFFER  
GEOMETRY TEXTURES

❖ FUN FANDANGO

CHARTING INGENUITY  
WITH TIM SCHAFER

POSTMORTEM:  
**THE BRAINS BEHIND WIDELOAD'S  
STUBBS THE  
ZOMBIE**



"To call the graphics **amazing** is an extraordinary understatement." GameSpy

"Oblivion is, at this time, the **best-looking game** I have ever seen in my life." Xbox.com

"One look at Oblivion will **shatter your conceptions** about what is possible in a video game." GameInformer

( **their secret weapon?**  
**Gamebryo** )

To create their most stunning game so far, Bethesda Softworks used a mix of genius, hard work — and the most flexible 3D graphics engine on the market.

**The Elder Scrolls® IV: Oblivion™**, a title for Xbox 360 and the PC, is already being hailed as one of the most beautiful games ever. How can Gamebryo help your vision emerge? Find out at [emergentgametech.com](http://emergentgametech.com).





## POSTMORTEM

### 24 CHOMPING AT THE BIT: WIDELoad GAMES' STUDIO EXPERIMENT BITES BACK WITH STUBBS THE ZOMBIE

Wideload is using a unique model for game development which has generated significant buzz in the industry. Similar to what Hollywood studios do, the company only hires select core staff and outsources the brunt of the work to contractors. Here, Wideload founder Alexander Seropian lays out the ups and downs of his cost-effective model, sharing how it helped his team ship a successful game.

*By Alexander Seropian*

## DEPARTMENTS

**2 GAME PLAN** *By Simon Carless*  
Zombie Zoo

**4 HEADS UP DISPLAY**  
BioWare/Pandemic, Xbox 360 launch, and Valve's Steam

**7 SKUNK WORKS** *By Carey Chico*  
Softimage XSI 5.0

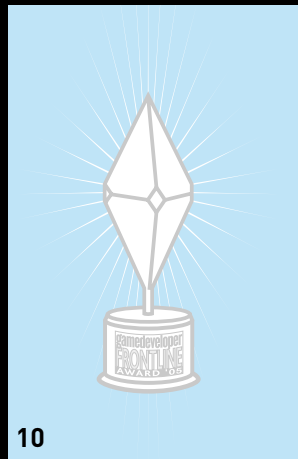
**48 A THOUSAND WORDS**  
Atlus' MAGNA CARTA

COVER ART: JUAN RAMIREZ AND MARK BERNAL

## FEATURES

### 10 GAME DEVELOPER'S 2005 FRONT LINE AWARDS

*Game Developer's* eighth annual Front Line Awards aim to single out those companies and products that have made significant strides in the battle to get developers home by five in the afternoon. Along with a panel of respected industry judges, we have awarded the leading tools, hardware, and book for their excellence, and inducted one distinguished tool into the Hall of Fame.



10

### 19 MAKING THE GRASS

As 3D game worlds become more realistic, small-scale geometric details will become more and more important. While it's vital to showcase our expanded hardware power in impressive ways, these tiny details, be they tufts of hair or blades of grass, can be computationally expensive. This technical article discusses the use of the GPU and vertex buffers to create memory efficient details in vibrant game worlds.

*By Halger Gruen*



19

## COLUMNS

**33 THE INNER PRODUCT** *By Mick West*  
Mature Optimization

[ PROGRAMMING ]

**37 BUSINESS LEVEL** *By Philip Oliver*  
Pigeonholed

[ BUSINESS ]

**38 PIXEL PUSHER** *By Steve Theodore*  
Anatomy for Animators Part III

[ ART ]

**41 GAME SHUI** *By Noah Falstein*  
Schafer on Creativity

[ DESIGN ]

**42 AURAL FIXATION** *By Alexander Brandon*  
The Voice Over Challenge

[ SOUND ]



## ZOMBIE ZOO

### STUBBS THE ZOMBIE, THE MCCARTHY ERA

throwback from Wideload Games' first title, meets the *Game Developer* Front Line Awards in the gory cover image of this issue—or at least it would be gory if the statuette could bleed. Fending off monsters with one hand and giving out plaudits for game tools with the other has left us feeling slightly zombie-fied, too.

### TRENCHANT WARFARE

In a year when game tools and other products to aid game professionals are even more important, especially as game budgets increase during a market transition, we're delighted to present the 2005 Front Line Awards (pg. 10), honoring the best products related to video game development in multiple categories, such as middleware, game engines, hardware, and a number of others.

Thanks to our panel of experienced judges, we whittled down first the nominated products and then winners in each category. Plus, one tool has earned its way into the *Game Developer* Front Line Hall of Fame, joining the ranks of DirectX (2002), Photoshop (2003), and Renderware (2004) for its continued dedication to improving game development.

### INTESTINAL FORTITUDE

When former Bungie Software co-founder Alexander Seropian set up shop on his own with Wideload Games, he decided to do things a little differently. Company-owned IP? Check. Small, manageable internal team and lots of external contractors to complete the game? Check.

In early 2004, when Seropian first announced this route, it was thought to be a sign of things to come, and, to many, an important test of alternate production methods. Now that the quirky 1950s brain-eatin' Xbox action title **STUBBS THE ZOMBIE: REBEL WITHOUT A PULSE** is complete and on retail shelves, Seropian has had a chance to sit down and write a definitive account of its creation. How did these new business attitudes affect the game, both positively and negatively? Wideload's founder spills the beans in the **STUBBS** postmortem (pg. 24).

### MOW NO MORE

We're also pleased to feature some fresh technical features and columns this month, with Holger Gruen's full-length article on vertex buffer-based geometry textures (pg. 19), which discusses ways of adding geometric details to the game world without breaking the processing

bank. Beautifully detailed tufts of grass have never been so accessible.

Elsewhere, programming columnist Mick West adds a highly relevant feature about code optimization; Pixel Pusher's Steve Theodore continues his anatomy lesson; longtime design columnist Noah Falstein interviews Double Fine's Tim Schafer about game originality; Heads Up Display explores the key BioWare/Pandemic merger; and the A Thousand Words art section showcases Korean RPG MAGNA CARTA.

### SLUMPING? MAYBE JUST A LITTLE

As I write this, the full ramifications of the U.S. video game holiday season sales are yet to be realized, but it's pretty clear that game sales were, at the very least, somewhat disappointing in the key post-Thanksgiving weeks. A fully fledged crash? Hardly, but an uncomfortable pause, perhaps.

Of course, critics are already pointing to a sagging game industry that produces tired sequel after tired sequel, and, as usual, they're largely wrong. Sure, there are plenty of follow-ups this holiday season, as in any holiday season, and since companies are working on early versions of next-gen titles, it may be difficult for them to produce all-new games for current generation hardware. There's only so much production game companies can endure.

But iterating on the same game template and gameplay engine often makes titles better in the long-term, and there have still been plenty of interesting, original titles on the market this season. Naturally, I'm not saying that **GAME FRANCHISE VOL. 8** is always the best idea, but I don't think that sequels are what disrupted the market this year. Rather, it was the promise of next-generation gaming that got in the way.

Am I blaming Microsoft's undersupply of the Xbox 360 for all the industry's woes? Not entirely, although it certainly has not helped. But knowing that a new, allegedly transformative graphical and online experience is just around the corner [even if you can't find the bloody hardware] will make anyone back off from buying current-gen games for a while. And that's exactly what happened. Suddenly, the gold at the end of the rainbow is more important than the riches piled around us right now. ❖

S!

Simon Carless, editor

#### EDITORIAL

##### EDITOR

Simon Carless [scarless@gdmag.com](mailto:scarless@gdmag.com)

##### MANAGING EDITOR

Jill Duffy [jduffy@gdmag.com](mailto:jduffy@gdmag.com)

##### ASSISTANT EDITOR

Brandon Sheffield [bsheffield@gdmag.com](mailto:bsheffield@gdmag.com)

##### ART DIRECTOR

Cliff Scorso [cscorso@gdmag.com](mailto:cscorso@gdmag.com)

##### CONTRIBUTING EDITORS

Alexander Brandon [abrandon@gdmag.com](mailto:abrandon@gdmag.com)

Noah Falstein [nfalstein@gdmag.com](mailto:nfalstein@gdmag.com)

Steve Theodore [stheodore@gdmag.com](mailto:stheodore@gdmag.com)

Mick West [mwest@gdmag.com](mailto:mwest@gdmag.com)

##### ADVISORY BOARD

Hal Barwood [Designer-at-Large](mailto:Designer-at-Large) Dave Pottinger Ensemble Studios

Ellen Guon Beeman Monolith George Sanger Big Fat Inc.

Andy Gavin Naughty Dog Harvey Smith Midway

Joby Utero Luxoflux Paul Steed Microsoft

##### ADVERTISING SALES

##### DIRECTOR OF SALES

Afton Thatcher [athatcher@cmp.com](mailto:athatcher@cmp.com) t: 415.947.6217

##### SENIOR ACCOUNT MANAGER, EAST COAST, EUROPE & EASTERN CANADA

Ayrien Machiran [amachiran@cmp.com](mailto:amachiran@cmp.com) t: 415.947.6224

##### ACCOUNT MANAGER, SO. CALIF., SOUTH WEST, CONTRACTORS, & MARKETPLACE

Susan Kirby [skirby@cmp.com](mailto:skirby@cmp.com) t: 415.947.6226

##### ACCOUNT MANAGER, NO. CALIF., NORTHWEST, ASIA & WESTERN CANADA

Nick Geist [ngeist@cmp.com](mailto:ngeist@cmp.com) t: 415.947.6223

##### ACCOUNT MANAGER, GLOBAL RECRUITMENT/EDUCATION & TEXAS

Aaron Murawski [amurawski@cmp.com](mailto:amurawski@cmp.com) t: 415.947.6227

##### ADVERTISING PRODUCTION

##### ADVERTISING PRODUCTION COORDINATOR Kevin Chanel

REPRINTS Julie Rapp [jarapp@cmp.com](mailto:jarapp@cmp.com) t: 510.834.4752

##### GAME GROUP MARKETING

##### DIRECTOR OF BUSINESS STRATEGY Michele Maguire

DIRECTOR OF MARKETING Tara C. Gibb

##### CMP GAME GROUP

SENIOR VP, GROUP DIRECTOR, APPLIED TECHNOLOGIES Philip Chapnick

CONFERENCE DIRECTOR Jamil Moleidina

CONFERENCE MANAGER Evelyn Donis

EXECUTIVE WEB PRODUCER Peter Leahy

EDITOR, GAMASUTRA.COM Simon Carless

FEATURES EDITOR, GAMASUTRA.COM Quang Hong

##### CIRCULATION

CIRCULATION DIRECTOR Kevin Regan [kregan@cmp.com](mailto:kregan@cmp.com)

CIRCULATION ASSISTANT Manager Jessica Ward [jward@cmp.com](mailto:jward@cmp.com)

CIRCULATION COORDINATOR Miguel Mendiola [mmendiola@cmp.com](mailto:mmendiola@cmp.com)

CIRCULATION ASSISTANT Michael Campbell [mcampbell@cmp.com](mailto:mcampbell@cmp.com)

CIRCULATION ASSISTANT Adrea Abidor [aabidor@cmp.com](mailto:aabidor@cmp.com)

##### 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)

##### INTERNATIONAL LICENSING INFORMATION

Mario Salinas

t: 650.513.4234 f: 650.513.4482 e: [msalinas@cmp.com](mailto:msalinas@cmp.com)

##### CMP MEDIA MANAGEMENT

PRESIDENT & CEO Steve Weitzner

EXECUTIVE VP & CFO John Day

EXECUTIVE VP, CORPORATE SALES & MARKETING Jeff Patterson

SENIOR VP, AUDIENCE MARKETING AND DEVELOPMENT Bill Amstutz

SENIOR VP, INTERNET BUSINESS Mike Azzara

SENIOR VP, CMP INTEGRATED MARKETING SOLUTIONS Joseph Braue

SENIOR VP & GENERAL COUNSEL Alexandra Raine

SENIOR VP, CORPORATE SALES Anne Marie Miller

SENIOR VP, MANUFACTURING Marie Myers

SENIOR VP, COMMUNICATIONS Alexandra Raine

SENIOR VP, CORPORATE MARKETING Kate Spellman

VP, AUDIENCE DEVELOPMENT Michael Zane

PRESIDENT, CHANNEL GROUP Robert Faletta

PRESIDENT, CMP ENTERTAINMENT MEDIA Tony Keefe

PRESIDENT, CMP HEALTHCARE MEDIA Vicki Masseria

SENIOR VP, GROUP DIRECTOR ELECTRONICS & SOFTWARE GROUPS

Paul Miller

SENIOR VP, ENTERPRISE GROUP Fritz Nelson

SENIOR VP, GROUP DIRECTOR COMMUNICATIONS GROUP

Stephen Saunders



Game Developer is BPA approved



United Business Media

IN GAME DEVELOPMENT THERE ARE

**NO SECOND CHANCES.**

MISS A DEADLINE OR SHIP A BUGGY PRODUCT AND IT'S

**GAME OVER.**



**DELIVER QUALITY. DELIVER ON TIME. USE SEAPINE.**

Powerful, cross-platform TestTrack Pro and Surround SCM help you manage game development from start to finish—spend less time chasing problems and more time creating quality games.

## TestTrack Pro

Advanced Issue Management

- Defect linking
- Robust reporting
- Fast, secure remote access
- Beta site feedback
- Easy to use and administer
- **NEW!** Full Unicode support and cross-platform client



## Surround SCM

Flexible Change Management

- Advanced code branching
- Changelists and atomic transactions
- Email notifications
- Fast, secure remote access
- Image browsing and thumbnail support
- **NEW!** Web browser access



## TRUSTED BY LEADING GAME DEVELOPERS

ION STORM, ATARI, CLIMAX, SPORTS INTERACTIVE, KUJU ENTERTAINMENT, and other top game development companies trust Seapine products to help them deliver bug-free games. Is your company the next Seapine success story?

Visit [www.seapine.com/gamedev\\_gd](http://www.seapine.com/gamedev_gd) to learn more about TestTrack Pro and Surround SCM and download evaluation copies.

 **Seapine Software**™

## PANDEMIC, BIOWARE: SOLD!

### ELEVATION PARTNERS, A PRIVATE EQUITY FIRM

which includes notable personalities such as ex-EA CEO John Riccitiello and rock star Bono among its board members, announced that the company has purchased a majority share in developers BioWare and Pandemic.

The developers will continue to operate independently of one another, but will have a



Pandemic's DESTROY ALL HUMANS.

centralized executive team and share resources and technology under the banner of the newly-formed holding company BioWare/Pandemic Studios. The founders of both developers will retain shares in their respective companies,

assuring that their power isn't wrested away. Riccitiello will assume the role of CEO for the holding company, which will be headquartered in Elevation's offices in Menlo Park, Calif. Both developers will remain in their original locations—with BioWare, which was recently named one of Canada's top 100 employers, staying in Alberta, and Pandemic keeping its Los Angeles home.

This effective construction of a super-developer is a markedly new tactic for Elevation, and for the industry at large. Traditionally, publishers buy developers in order to fill out their product portfolios and streamline workflow. In April 2005, in fact, Elevation was in the running to purchase U.K. publisher Eidos, but was beaten to the punch by the publisher Sci. Acquiring developers is not typically seen as a money-making venture for potential investors, given that they do not always receive royalties for their games and have high overhead between projects.

"I think things will change only for the better as a result of this deal at both BioWare and Pandemic," says Ray Muzyka co-head of BioWare. "On a day-to-day basis, Greg Zeschuk and I will continue to act as joint CEOs of BioWare, but we will also take on roles at the larger BioWare/Pandemic headquarters, serving as corporate VPs within the larger organization."

"BioWare as a whole will continue to grow in a careful, controlled manner, and we now have the financial backing [with infusion of significant working capital into the company from Elevation



BioWare's KNIGHTS OF THE OLD REPUBLIC.

Partners] to pursue growth in a whole bunch of areas we had thought of as longer term opportunities in the past," Muzyka adds.

"Relative to the broader business, the new company will offer something different to publishers," says Pandemic CEO Andrew Goldman. "We will provide them with well-developed product against commercial IPs where they do not shoulder the full risk of development and commercialization. We believe this will help them to manage their risk portfolio, grow their businesses and bring additional innovation to the industry," he said.

"This deal was our ideal scenario for continuing to build Pandemic."

—Brandon Sheffield

## SPECTOR GAINS STEAM

### ACCORDING TO INFORMATION NOW

available on the official Junction Point Studios web site, DEUS EX creator Warren Spector's new firm is "currently working with Valve on a new game using the Source Engine, to be delivered via Steam," Valve's online game distribution site.

Spector, whose past work includes ULTIMA UNDERWORLD, SYSTEM SHOCK, and THIEF, had been without an announced project since early 2005 when his previous studio closed, Austin-based Ion Storm. He founded Junction Point shortly thereafter.

Following the success of HALF-LIFE 2's online distribution via Steam and the resulting large user-base, Valve

is continuing to sign deals that allow it to sell PC titles directly through Steam—both games that use its Source Engine, as HALF-LIFE 2 did, and games running their own technology.

This latest high-profile deal with Spector is a continuation of Valve's wish to expand Steam into an effective method of digital distribution for PC games from multiple developers.

In early October, Valve announced its long-term plans for Steam alongside the latest changes to its client, which have completely redesigned the client's GUI, and added a discrete "Shop" area to allow separate purchases which

don't fall under Valve's bundle deals for its own games. The first of these is the innovative action title RAG DOLL KUNG FU, from Lionhead artist Mark Healey, and other Valve-developed titles are now also available for individual purchase.

The most recent games to announce distribution via Steam are UNREAL mod extension RED ORCHESTRA: OSTFRONT 41-45 and cult title DARWINIA. Other titles that are available or will soon be available on Steam include NARBACULAR DROP,



DEUS EX

THEY HUNGER: LOST SOULS, PIRATES OF THE BURNING SEA, and SIN EPISODES, showing an intriguing blend of indie PC titles, expansions to popular mods, and episodic efforts from larger developers.

—Simon Carless

# BGT ACQUIRED FOR SERIOUS GAME AI

## A LUCRATIVE FIRE HAS BEEN KINDLING

between the simulation industry and the serious games market. Engenuity Technologies, a global player on the visualization and simulation software side, recently stoked its flame by adding a little kindling: BioGraphic Technologies (BGT).

BGT develops AI software for games, visual simulation uses, and other electronic media. The Montreal-based company counts among its game customers Sony Computer Entertainment America, Midway Games, and Kuju Entertainment.

When Engenuity announced the acquisition the company stated that subsuming BGT will allow it to "offer a broader range of applications to its existing simulation customers, and to penetrate new vertical markets, such as urban simulation, homeland security, and serious games."

"This is the first step in our acquisition



strategy to expand our presence in the broader simulation market," said Patrice Commune, CEO of Engenuity at the time of the announcement. "We are excited about the standalone strength of the AI implant product in its existing games, entertainment, and VizSim markets."

BGT's latest product release was AI implant

version 3.6, software that's used in a number of prominent video games; recent deals include an agreement with Midway to use the technology as its preferred AI engine for products such as JOHN WOO'S STRANGLEHOLD, and a deal with Sony to use it for MAJOR LEAGUE BASEBALL 2006.

The details of the acquisition are as follows: Engenuity acquired all of BGT's outstanding shares (in exchange for around \$835,000 of its shares), about \$162,000 in cash, and a promissory note of \$250,000 to be paid in 12 months from closing. Engenuity also assumed nearly \$1.5 million of BGT's debt, about a third of which was repaid at closing.

—Jill Duffy

# XBOX 360 LAUNCHES WORLDWIDE

## CHIP SHORTAGE CITED AS REASON FOR SOME ORDERS GOING UNFILLED

### THE XBOX 360, MICROSOFT'S

newest console, achieved instant sellout status during its November 22 U.S. launch—but while demand was certainly great, that wasn't the only reason you couldn't find an Xbox 360 the day after the U.S. and U.K. launches. In the *Ottawa Citizen* newspaper, Microsoft CEO Steve Ballmer cited difficult chip production as the primary reason for the crushing Xbox 360 shortage, commenting, "In these new consumer electronics devices based on new chips, there's always the question of what yield will you get out of the manufacturing process of the new chip. We're getting a little less, but not much less than the yields we expected, and we know that the yields we expected will probably outrun supply."

Though limited initial numbers are to be expected, Microsoft's publicly stated plans to ship between 2.5

million and 3 million consoles worldwide within the first three months of the system's life implied that reasonable restocks would follow. Unfortunately, as of press time, some pre-orders at U.S. stores such as GameStop had still not been filled, due to continuing lack of hardware, and the European launch of the console on December 2 suffered from similar problems. The Japanese debut on December 11 was reportedly better, due to less severe demand for the console.

R. Richard Fontaine, chairman and CEO, GameStop commented in November that "the total hardware released to date in the U.S., and to GameStop, are far less than we had anticipated," leading worldwide game retailers, including The GAME Group in Europe, to warn of possible profit-related issues due to reduced supply.

Aside from the supply issues, the Western launches were largely

considered successful, with good reactions to the console's wide-ranging 18 launch titles and an especially positive reaction to the Xbox Live and Xbox Live Arcade service, which offers new opportunity to independent developers.

But the major supply problems have affected game professionals' opinions of the launch, with Coray Seifert of Large Animal Games sharply critical of Microsoft in his response to a Gamasutra.com Question Of The Week, commenting that the company had "suffered a huge setback when it failed to get enough machines to the market for the Christmas

holiday. While coordinating a console launch across multiple continents is truly a massive undertaking, a company as massive as Microsoft has no excuse for being so grossly unprepared."

—Brandon Sheffield, *Simon Carless*

## CALENDAR

### The Mobile Games Forum

London Marriott Hotel Grosvenor Square  
London

January 25 and 26, 2006

Cost: £1,299

[www.osneymedia.co.uk](http://www.osneymedia.co.uk)

### How to Break Into the Game Industry

Pickle Research Campus

Austin, Texas

January 21, 2006

Cost: \$45–\$60

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

Anark is proud to announce that

# ANARK GAMEFACE

is a GameDeveloper Frontline Award 2005 Finalist.



To begin your Anark Gameface evaluation, contact [gameface@anark.com](mailto:gameface@anark.com).



Professional Grade User Interface Tools and Technology

[anark.com](http://anark.com)



# SOFTIMAGE XSI 5.0

BY CAREY CHICO

**A SMALLER NUMBER OF TOOLS WHICH,** when combined together, produce a larger number of results—that's the Softimage XSI way of thinking. And in version 5.0, Avid has given game developers (as well as film creators) a few new toys to play with. Luckily for the game-making group, many of the new tools in XSI 5 are designed for building next-generation games. It's clear that Softimage is taking next-generation game development seriously and is in serious competition with both Autodesk (Max) and Alias (Maya)—or, given their latest news, should I say “Maya Max?”

On the whole, Avid has tweaked and refined XSI, giving it a fresh coat of paint without changing the color. But what makes the new release stand out is a handful of new tools within. From GATOR (Generalized Attribute Transfer Operator) to the refined Ultimapper tool (previously known as GPU SurfaceFX), XSI gives us ways to acquire and manage new texture data types and handle the higher-resolution models that the influx of new games is bringing to the table. Moreover, these new tools are primed to bring game developers into the next generation without losing too much hair.

## U.I. FAMILIARITY HURDLE

The software now sports an additional interaction mode that clearly targets Maya users by implementing the QWERTY key set that they've used for years. Whether this move is an olive branch to make the environment easier and more comfortable for new users, or a stepping-stone to convert them is something of a moot point. Since XSI has always had a very flexible control UI, offering the QWERTY key set option can only be seen as yet another layer on top of that very agile system.

My personal feelings on this new interaction mode are mixed, however. It seems the time it took to create and implement this feature would have been better spent investing in even more tools for this release or expanding some of the Texture editor controls into the main 3D viewport.

## GATOR GOODNESS

One of the most astounding and industry-leading features to come out of this release is the GATOR toolset. Gator has been getting a lot of industry attention, and once you hear what it can do, you'll understand why.

With Gator, users can transfer materials, texture UVs, vertex colors,

property weight maps, envelope weights, and shape animation from one object to another. These models need not share any component, attribute, or even vertex or triangle count to permit transfer. The results are remarkable, and the benefits of using Gator in next-gen game pipelines are clear. I tried it on numerous objects, transferring high-resolution UV, texture, and vertex color data onto lower-resolution LOD models, and it worked magically well.

Softimage also included a great surprise for modelers hidden in the plug-in manager. Under File>Plug-in Manager>Workgroups, you can click Connect and find a button called Try SDK Example Workgroup. Once installed, you'll find a newly realized version of the User Normal Editing script. This tool, once found as a script in the NetView, now is found in the Alt-RMB menu and has an actual PPG window. With it, developers can directly manipulate all or any normals on an object as desired. It's a great tool that provides a needed solution for the game developer crowd.

## TWEAK AWAY

Let's revisit the Softimage philosophy and restate it more fully as “a smaller number of tools which, when combined together, produce a larger number of results.” The bit about “smaller number of tools” basically means that similar operations are grouped together as single tools. When invoked by typical modifier keys such as Alt, Shift, and Ctrl, each of these tools contains a larger number of specific features.

Following this line of thought, the Tweak Component tool bulges with new feature goodness. Replacing the classic Move Point tool previously mapped to the 'm' key, the Tweak Component tool is replete with quick manipulation controls for directly moving vertices, edges, and polygons. Users can just grab and translate components any way they like in any coordinate system.

Additionally, the tool includes weld and magnet features that allow users to

## SOFTIMAGE XSI 5.0



### STATS

Softimage Co.  
3510 St-Laurent Blvd.  
Montreal, Quebec, H2X 2V2, Canada  
514.845.1636  
[www.softimage.com](http://www.softimage.com)

### PRICE

\$495 (foundation version), \$1,995 (essentials), \$4,995 (advanced)

### REQUIREMENTS

Softimage XSI 5.0 supports the following operating systems: XSI for Windows XP Professional; XSI for Linux; SPM for Windows, Linux, and Irix.

The listed requirements are for Windows XP Professional.  
• Workstation with AMD K7 or higher processor or workstation with Intel

Pentium III or higher processor

- OpenGL accelerated graphics card, at least 64MB RAM (recommended)
- 710MB available hard disk space
- TCP/IP service protocol installed on both client and server computers
- 512MB RAM (minimum)
- 200 to 300MB swap space
- 1,280x1,024 screen resolution (recommended)
- Three-button mouse
- Web browser.

### ADDITIONAL RECOMMENDATIONS

Internet connection  
DVD drive  
Software DVD  
WinZip tool  
A backup device, such as a DAT, 1/4" tape drive, removable hard disk, or DVD/CD-R/CD-RW drive

See [www.softimage.com](http://www.softimage.com) for full requirements and recommendations.

### PROS

1. New tools like Gator and Ultimapper will rock the gaming world.
2. Still an excellent melding of UI simplicity with underlying tool complexity.
3. Superb customer service and support.

### CONS

1. New QWERTY UI options not much use for veteran users.
2. New physics engine not intuitive to learn.
3. Would like to see an expansion of some of the Texture editor controls into the main 3D viewport.



The GATOR tool, new to XSI 5, lets you transfer materials, texture UVs, vertex colors, property weight maps, envelope weights, and shape animation from one object to another.

“slide” components across surfaces. You can even loop select with the Alt key and slide that over the surface! I tried this on a medium-resolution head model and watched with glee as I adjusted entire edge loops across the model’s face. Very nicely done.

**ULTIMAP THIS**

Texturing has not been left out of the v5 upgrades—some new features improve upon existing ones to make generating special data-rich textures even easier.

What does the Ultimapper do, you ask? The Ultimapper generates a variety of texture map data types. In addition to normal maps in both tangent and model space, it also outputs ambient occlusion, depth, and albedo map types. (It also outputs material tag image maps.) It operates on a cage-based system and takes full advantage of Mental Ray while doing so. Also, it generates tangent data and creates preview shaders so that you can view the results in real time using any variety of real-time shader formats like CG, DirectX, and OpenGL. And incidentally, this release also delivers .FX support.

I found a great use for the Ultimapper in generating texture data for LOD objects. I tried taking a higher-resolution model with six textures and translating that data onto a very low-resolution model using only one texture. Ultimapper chewed up the data and spit out virtually identical texture data all spaced out on my UV set, deriving a

new upgrades and tools, there are some smaller features that should get some notice, too. The Texture Editor can now handle multiple UV coordinate editing simultaneously. For example, if you select as many as four objects, a drop-down menu allows you to access and view their multiple UVs, a very useful resource when you’re attempting to group multiple object UVs onto a single normal map texture page.

There’s also a brand new dynamics engine implementation using PhysX from Ageia. That is the new “physics-on-a-chip” engine that provides hardware acceleration to dynamics in the same way Nvidia provides hardware acceleration to visuals. As more card vendors get on board, Softimage users will have support already built in.

**ANIMATION AND DATA MANAGEMENT**

On the animation front, there’s improved support for viewing keys on the time line. You can copy, paste, and move these keys without reverting to the Animation Editor. There’s also a brand new Parameter connection editor, which speeds up the creation of linked parameters.

The Shape Editor also makes editing and animating shape keys a snap. With this new interface, you can save separate shape keys, which automatically get put into a list with sliders. You can then both

single texture out of six.

Since Mental Ray automatically writes out the alpha channel data—you get that too—I tried this on a tree model and transferred not only the 3D alpha, but the texture alpha into a single texture.

Behind the major

visualize and animate the weighting of each shape, watching the results in the 3D viewport. Shape Editor greatly enhances the organization and process of shape animation.

Two big changes were made to XSI 5 on the data management side. First, the OBJ importer was improved to import complex Zbrush geometry and displacement maps for rendering.

Second, the dotXSI file format has grown up since version 3.6, and users finally get the source code for the format, a long-time wish list item which now gives the users support for multiple UV and Vertex Color sets with the freedom to add what they need to the format.

**SWEET CHANGES**

XSI 5 gives users even more reasons to stay with Softimage and gives other 3D package users, in light of recent industry developments, an alternate patch of green grass to rest their heads on. It’s hard to find a cleaner integrated feature set with a more fluid interface than Softimage’s.

While I clearly enjoy working with XSI, I do question why a new interaction model was incorporated into this release. Other packages aren’t doing this, and as a user, I’m fully capable of transitioning between key maps from one software package to another when needed.

Ultimately, these issues are small in comparison to the new tools that target the game industry such as Gator and Ultimapper. These two tools alone make our working lives immensely easier and more efficient.

XSI’s 4.0 release was considered a fully robust tool, but 5.0 is all icing on the cake—the whipped cream kind. ❄

**CAREY CHICO**, a 10-year veteran of the game industry, is the executive art director at Pandemic Studios. Email him at [cchico@gdmag.com](mailto:cchico@gdmag.com).

gamedeveloper  
**FRONTLINE**  
FINALIST '05

*"With ReplayDIRECTOR, we only had to find the bug once. Our developer just pressed play and saw the crash in the debugger... We had a fix the next day."*

■ **John Chwanec**, Lead Producer, Eidos Interactive



# **BUGS. TOASTED.**

## **RECORD. REPLAY. FIXED.**

ReplayDIRECTOR™ gives you **Deep Recording**. This is much **more than just video capture**. Replay records every line of code that you execute and guarantees that it will Replay with the same path of execution through your code. Every time. Instantly Replay any bug you can find. Seriously.

## **DEEP RECORDING. NO SOURCE MODS.**

*download today at [www.replaysolutions.com](http://www.replaysolutions.com)*

*email us at [info@replaysolutions.com](mailto:info@replaysolutions.com)*

**REPLAY SOLUTIONS**

1065 E. Hillsdale Blvd. Suite 307 Foster City, CA, 94404 - Tel: 650-472-2208 Fax: 650-472-2210

**CRYSTAL  
DYNAMICS**

eidos

XBOX 360

VIDIA

XBOX

# THE GAME DEVELOPER 2005 FRONT LINE AWARDS

» **2005 WAS ANOTHER EXTREMELY BUSY YEAR FOR THE VIDEO** game industry and the game tool industry alike, with development starting in earnest for next-generation consoles. Many tool developers, accordingly, have been stepping up and announcing new products for Xbox 360, PlayStation 3, and even PSP—as well as hardware and software that produces higher complexity data, which is what's needed for games to look good on increasing high-tech hardware platforms. However, sometimes being bigger isn't enough.

The climate of publisher consolidation requires that tools providers act with equal dynamism. Vendors have learned, this year more than ever, that to simply churn out upgrades that fulfill already-expressed needs simply isn't enough.

One goal of all progressive vendors should be to hear these words from a thankful customer: "I didn't even know that this was what I needed." To create truly eye-opening tools for game developers, tool companies must not only meet needs, but discover and articulate them as well.

In recognition of the heightened competitive atmosphere, the 2005 Front Line Awards have grown equally decisive. In years past, for example, it was possible for two, sometimes

even three products to win in any one category. In years past we allowed an indefinite number of nominees to populate the finalists list. No more.

This year, a controlled number of finalists were selected in the seven categories, and only one winner will receive a Front Line Award in each category.

Nominations for the Front Line Awards were open to all new products and new versions of products related to game development released before September 1, 2005. Finalists were selected by the editors, whose decisions were greatly influenced by previous reviews of products, comments from customers of the tools, and the opinions of the FLA judges. Winners were selected by a panel of judges composed of professional game developers specializing in the fields relevant to each product category.

The winners were chosen based on the following criteria: relevancy to current and next-generation game creation, ease of use, speed of output and/or responsiveness, value, and quality. Books were judged by the editors of *Game Developer*.

Congratulations to the finalists and winners.

—Jill Duffy

5

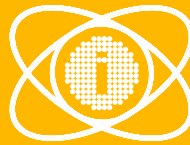


The Front Line Awards would not be possible without our panel of judges, whom we sincerely thank. We also thank Andrew Zaferakis for his contribution.

#### JUDGES

Alexander Brandon  
Tom Carroll  
Michael Dean  
Ron Fosner  
Brad Fotsch  
Spencer Lindsay  
Noel Llopis  
Justin Lloyd  
David March  
Dan Paladin

*Game Developer* also thanks users and licensees of many of the products who submitted anonymous comments, which were integrated into the judging process.



INDEPENDENT  
GAMES FESTIVAL

REWARDING INNOVATION IN INDEPENDENT GAMES

THE 8TH ANNUAL

# INDEPENDENT GAMES FESTIVAL

MARCH 22-24, 2006

SAN JOSE, CA

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

PLAY THE BEST  
INDIE GAMES OF THE YEAR!



PLATINUM SPONSOR:

**[adult swim]**<sup>TM</sup>

GOLD SPONSOR:

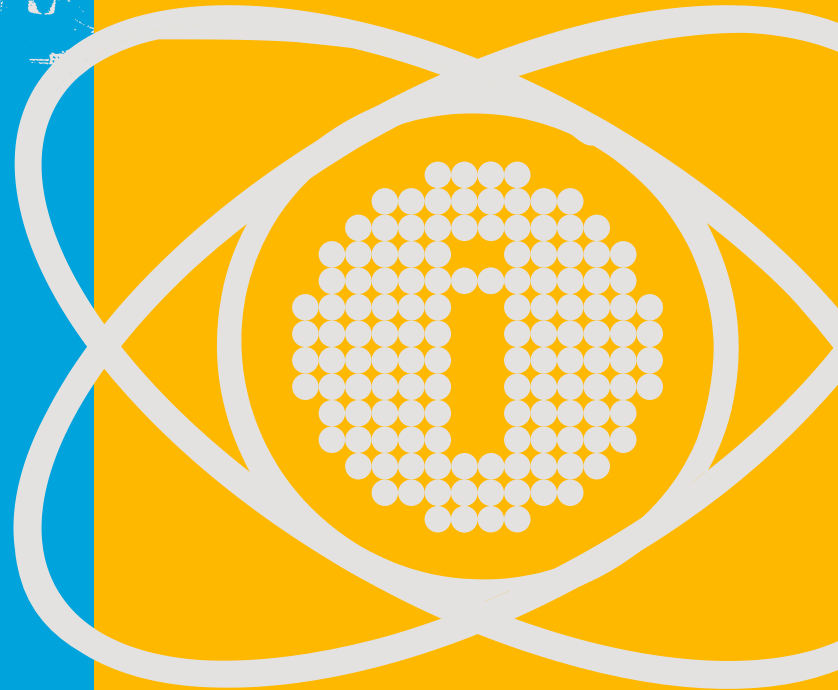
**Microsoft**<sup>®</sup>

MODDING PAVILION SPONSOR:



MEDIA SPONSORS:

 [DOWNLOAD.COM](http://DOWNLOAD.COM)



**FINALISTS ANNOUNCED!**  
FIND OUT MORE AT [WWW.IGF.COM](http://WWW.IGF.COM)

- Visit the IGF Pavilion at GDC
- Attend the IGF Awards ceremony Wednesday, March 22
- All GDC attendees are invited!

**GameDevelopers**  
Conference

**MARCH 20-24, 2006**  
**SAN JOSE, CALIFORNIA**

Register by February 15, 2006 and save up to 35%  
Use priority code IGFMAXX when registering



# Hall of Fame

## 3DS MAX

AUTODESK

[WWW.AUTODESK.COM/3DSMAX](http://WWW.AUTODESK.COM/3DSMAX)



**3DS MAX HAS UNDERGONE** more under-and-above-the-hood changes than any other 3D package I know of. Based on its rather simple beginning, it's amazing to see just how innovative Autodesk has continued to be over the years.

Artists in the rapidly evolving game industry, struggling to keep up with greater and greater demands from consumers, have always demanded that their 3D packages be up-to-date on the latest technologies, and no package has answered their calls faster

than or as frequently as 3ds Max. Though some of the tools still contained in Max have quickly become outmoded, there's no denying that the new ones constantly being added to the package have increased artists' options for efficiently creating next-generation content.

Though I've criticized Max in the past for not doing more than "lifting the hood" (I've often felt that it needs to be put up on a rack and completely overhauled), there is one huge benefit of keeping the package largely the same: there's basically

no learning curve between versions.

Autodesk has added support for technologies that have proven they're here to stay, for at least the foreseeable future, such as normal maps. Support was added quickly and minimally, with successive versions expanding upon and fleshing out the technology's implementation. And the company has continued to show its commitment to this type of support.

In the latest version, Max 8, we were introduced to an extraordinary time saver for the UVW unwrapping process—Pelt Mapping—which allows artists to quickly define the seams of their models, bringing them into the mapping utility, and stretching out the UVs intuitively and effectively.

The results are

phenomenal. Unwrapping the torso or face of a character is now a trivial matter and can be accomplished almost instantly. It is the biggest time-saver integrated into a 3D package that I can remember. One wonders how we ever got so far without it.

3ds Max is so ingrained into so many development houses that it would be easy for Autodesk to sit back and capitalize on the software's success with a largely captive customer base. However, the Max team has proven that they've truly earned their customers' loyalty by continually expanding upon an already solid package. They've proved they're willing and able to lead the way for a long time to come.

—Michael Dean

### Books Finalists

*GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation*, Matt Pharr and Randima Fernando (eds.), Addison-Wesley  
*A Theory of Fun for Game Design*, by Raph Koster, Paraglyph Press  
*The Game Localization Handbook: Localization Production Pitfalls*, by Heather Maxwell Chandler, Charles River Media  
*Introduction to Game Development*, Steven Rabin (ed.), Charles River Media



# Books

## AUDIO FOR GAMES: PLANNING, PROCESS, AND PRODUCTION

BY ALEXANDER BRANDON, NEW RIDERS PRESS (2005)



**WHAT I LOVED ABOUT *AUDIO FOR GAMES: Planning, Process, and Production*** is that it delivered on its title impeccably. In a sense, the book dares to be narrow, defining a very specific audience—people who know at least a little about audio production and want to know more specifically about game audio—and outlining in

meticulous detail (Brandon even explains the market rate for hiring an orchestra in both the U.S. and Europe) what to expect, how to survive, and how to advance one's career while working in game audio. A handful of interviews with industry leaders, including Warren Spector and Marty O'Donnell, add balance to Brandon's perspective.

*Audio for Games* explores the audio personnel's place in game development, too, stressing the importance of communication in the workplace and sharing street-smart tips, like requesting specific documents from the game design team rather than waiting for the information to be given to you—or more likely, waiting and waiting

until you realize you've been forgotten about. Brandon captures the Law of the Land for audio game developers, acknowledging their place in the pecking order of gamemaking, with a good sense of humor, albeit black at times. It's not pretty, but it's a reality that Brandon contends with keenly.

—Jill Duffy

Books were judged by Simon Carless, Brandon Sheffield, and Jill Duffy, the editors of *Game Developer*.



## Middleware

### SPEEDTREERT v1.7

INTERACTIVE DATA VISUALIZATION (IDV)

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



**SPEEDTREE, WHICH** despite its name makes all manner of foliage, is an essential component to most video game development efforts, especially if you're working on an MMORPG and an open world title.

SpeedTree is totally painless to use, provides tons of visual feedback, and produces results that are relatively easy for

programmers and technical artists to insert into various game engines.

IDV's marvelous tree-making tool also supports lighting and wind effects, which can be quite useful for real-time games. And because the company sells source code as part of a standard license, developers can finally do more with their trees than simply blow them around.

It's tool providers like IDV, who give asset creators more control over more intricate details, that fully understand what we're facing in the next generation of game development.

—Tom Carroll and David March



## Art Tools

### ZBRUSH 2

PIXOLOGIC

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

**ZBRUSH 2 IS QUICKLY BECOMING** the de facto standard of developers everywhere looking to easily add high-quality normal maps to their real-time models. Artists have several tools at their disposal to complete a task using the path that makes the most sense to them.

For normal mapping chores—sure to become a tidal wave of employment opportunity in the near

future—Zbrush is a must buy, must learn tool. Despite the steep learning curve and less-than-intuitive menu system, Zbrush broke into the game art market in a heartbeat for good reason and will likely mature quite gracefully in future releases.

—Michael Dean and Tom Carroll



#### Art Tools Finalists

FX Composer, NVIDIA ClayTools system for 3DS Max v.1.1 and Maya v1.0, SensAble Technologies, Inc. Modo 102, Luxology Maya 7, Alias



GameDevelopers  
Conference

MARCH 20-24  
SAN JOSE, CALIFORNIA

WHAT'S NEXT  
GDC:06

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

GAME DEVELOPERS CHOICE AWARDS

INDEPENDENT GAMES FESTIVAL

GDC MOBILE

SERIOUS GAMES SUMMIT

GAME CONNECTION

The next generation will be defined by the games you create. At GDC:06, learn lessons from next-gen, handheld, and current game development, and gain access to the people, technologies, and tools that will define what's next for our industry. LEARN MORE AT [WWW.GDCONF.COM](http://WWW.GDCONF.COM).

REGISTER BY FEBRUARY 15, 2006 AND SAVE UP TO 35%!  
USE PRIORITY CODE PAMAXX WHEN REGISTERING.



## Audio Tools

### ISACT v1.60

CREATIVE LABS

[HTTP://DEVELOPER.CREATIVE.COM](http://developer.creative.com)

**THE ISACT SOUND SYSTEM IS THE** first commercially available adaptive music system I've seen. It provides a beat matching system that can be

used to implement an adaptive soundtrack into nearly any game type.

Given this groundbreaking move into the marketplace, as

well as its ease of use and strong support structure, it deserves the FLA in the game audio tool category—congratulations.

—Alexander Brandon

#### Audio Tools Finalists

Miles Sound System, Rad Game Tools  
Lipsync SDK 3.0, Annosoft  
Harmony Hard drive, DeWolfe Music  
CRI ADX, CRI Middleware Co.



## Hardware

### MX40 MOTION CAPTURE SYSTEM

VICON

[WWW.VICON.COM](http://www.vicon.com)

**VICON'S MX 40 CAMERAS ARE THE** world's first to record with sub-millimeter-accuracy. How can you freaking beat that? Many motion capture systems are still black and white to lighten their processing. The MX cameras, however, record the whole of the image in 10-bit grayscale.

Vicon's MX cameras can also dramatically increase accuracy by fitting a circle more accurately around the marker image.

What does this mean? You get cleaner and better data, which will increase your quality and speed. In addition, with Vicon's new IQ software, you can decrease your clean up and processing time. Creating a full body capture with hand and face markers at the same time would be pretty impossible with



the old black and whites. The MX 40 are the way to go if you want to create the most stellar looking work to date.

—David March

#### Hardware Finalists

Razer Copperhead, Razer  
Quadro FX 4500, Nvidia  
SpacePilot, 3Dconnexion  
DX1 Input System, Ergodex



# Programming Tools

## PRODG FOR PSP

SN SYSTEMS LTD

[WWW.SNSYS.COM/PSP/PRODG.HTM](http://WWW.SNSYS.COM/PSP/PRODG.HTM)

### SN SYSTEMS' PRODG FOR PSP

offers game developers a package of unrivaled tools that are used to create interactive titles for Sony's newest handheld console. The SNC compiler, specifically developed to deliver optimized code for consoles, generates some of the best code compared to the regular GNU C++ compilers that they have shipped in the past. The new IDE, modeled after Visual Studio .NET, is a good addition to the venerable

line of development tools that SN Systems provides.

When the PSP launched in the U.S., 20 out of the 24 coinciding launch titles were developed using SN Systems' ProDG, including DYNASTY WARRIORS, LUMINES, FIFA SOCCER 2005, RIDGE RACER, and WIPEOUT PURE.

—Justin Lloyd

### Audio Tools Finalists

Intel VTune Performance Analyzer, Intel Corporation  
ReplayDIRECTOR v2.0, Replay Solutions LLC  
Perforce SCM 2005, Perforce Software  
SlickEdit 10, SlickEdit



# Engines

## UNREAL ENGINE 3

EPIC GAMES INC.

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



### UNREAL ENGINE 3 PROVIDES

a powerful foundation to build PC, PlayStation 3, and Xbox360 games. For developers starting a new project, the state-of-the-art graphics renderer, physics, asset pipeline, networking, AI framework, and UnrealEd let an entire team hit the ground running and enter production quickly.

Although Unreal Engine 3 is not a finished engine, with some of its subsystems incomplete or not optimized, it does come with full source code, and Epic's technical support is highly responsive.

Licensees of Unreal Engine 3 include Atari, NCsoft, Namco, Buena Vista Games, Microsoft Game Studios,

TimeGate Studios, Midway Games, Silicon Knights, and The U.S. Army (AMERICA'S ARMY).

—Andrew Zaferakis,  
High Moon Studios

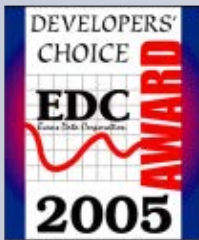
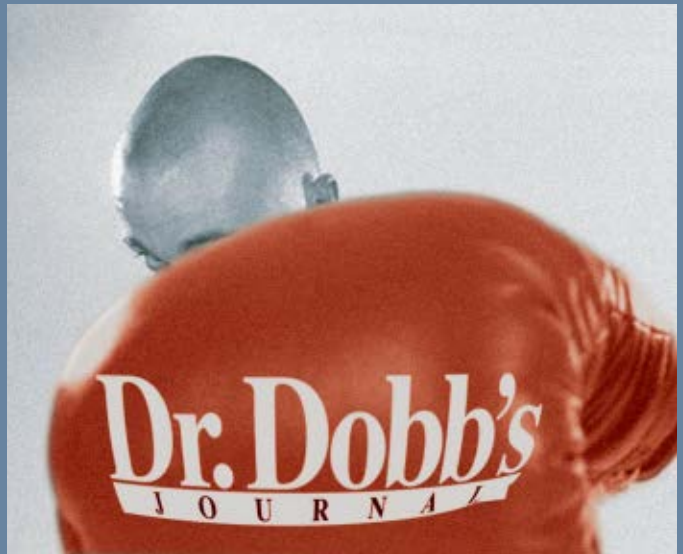
### Engine Finalists

Virtools Dev 3.0, Virtools Source, Valve

BigWorld MMO Technology Suite V1.6, BigWorld Pty Ltd.  
Gamebryo 2, Emergent Technologies

# Technical Knockout

The magazine of choice  
for champion developers



Once again, Software Developers and Development Managers have selected *Dr. Dobb's Journal* as the best industry publication, according to the 2005 Evans Data Developer Marketing Patterns Study.

**WHY?** Here's what *Dr. Dobb's* readers have to say\*:

*“DDJ not only provides a theoretical basis for using the new technology but usually provides a real-world implementation that I can study to determine applicability as well.”*

*“DDJ is invaluable to me as a software architect/developer.”*

*“DDJ is my connection to the world of people who actually think about how computers and software work.”*

*“DDJ frequently suggests to me new tools and techniques for solving the problems that I deal with in my job.”*

So, if you're serious about software development and design, why not make sure *Dr. Dobb's Journal* is in your corner every month?

Subscribe today at:

Domestic: <https://www.neodata.com/ddj/qualformWDDJ.html>

International: <https://www.neodata.com/ddj/2awa.html>



CMP

United Business Media

PO Box 52226  
Boulder, CO 80322-2226  
ddj@neodata.com

\*Source: Signet *Dr. Dobb's Journal* Ad Study, February 2005

**Dr. Dobb's**  
JOURNAL  
SOFTWARE  
TOOLS FOR THE  
PROFESSIONAL  
PROGRAMMER

# MAKING THE GRASS

## VERTEX BUFFER-BASED GEOMETRY TEXTURES

**EVERY 3D GAME WORLD LOOKS MORE REALISTIC WHEN IT** contains small-scale geometric details. These details can be tufts of grass, small stones, flowers, or even fur. Ideally, developers want a rendering technique for these details that consumes processing and rendering power only near the viewer. The technique should be simple to implement as well as inexpensive computationally and in terms of video memory. Ideally, the technique should have a low vertex memory footprint.

Assume we have object  $A$ , which represents some geometric detail. We generate a set  $S$  of randomly scattered points on a generic triangle describing each point by three barycentric coordinates. For every point  $P$  in  $S$ , we add the vertices of a modified copy of  $A$  to a vertex buffer,  $VB$ . In addition to the original vertex data, a set of barycentric coordinates defining  $P$  are added to every vertex.

If we draw from  $VB$  and pass the 3D positions of the corners of a triangle  $T$  of a game world to a vertex shader, then this shader can compute a point  $TP$  on  $T$  from its corners and the per-vertex barycentric coordinates. If the shader further adds the original vertex position also available in the vertex to this position, we can effectively instance  $A$  at  $TP$  as its origin.

The outcome is that the vertex buffer can be mapped to  $T$  similar to the way a texture can be mapped to it. The vertex buffer that we now call vertex buffer-based geometry texture (VBGT) can therefore be used to add geometric detail—for example tufts of grass, flowers, stones, or even fur—to arbitrary triangles.

This concept is a form of instancing, but instancing techniques can be used to make drawing VBGTs more efficient, as I show in this article, if we want to texture several triangles with a VBGT.

Seamless level-of-detail transitions can be achieved for some geometric detail. This article describes techniques that handle base triangles with varying areas.

### MOTIVATION

But first, let's briefly review the methods that have been used to date to create geometric details, taking note of their pros and cons.

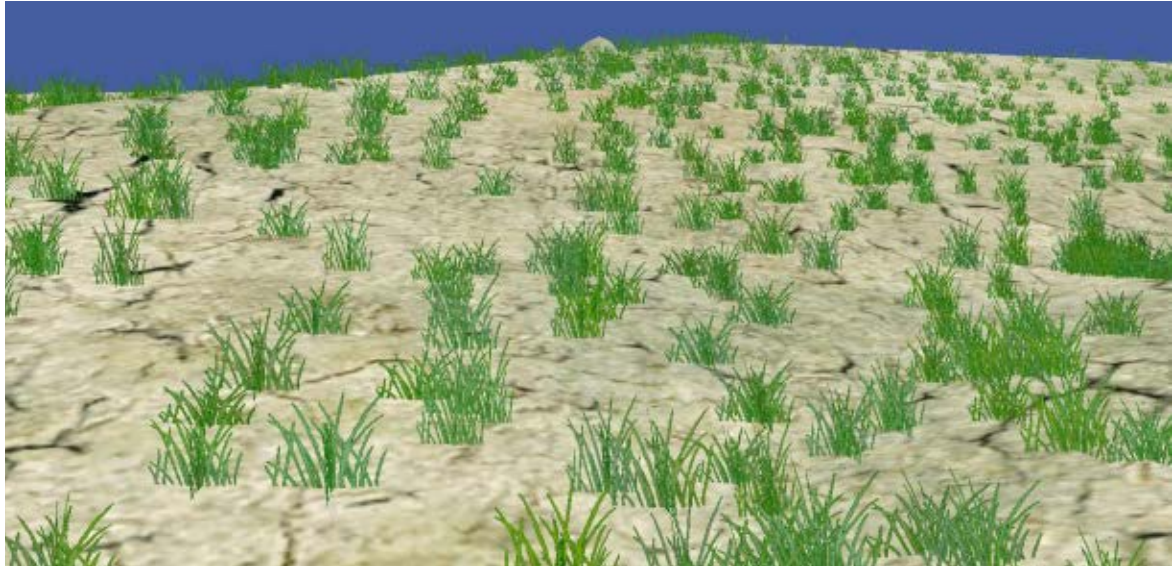
An artist can, of course, model small geometric details along with the other scene geometry. At run-time, no further CPU load is generated to display the geometry. The downside of this method is that a very high amount of vertex memory is consumed by the geometric detail. Also, a very high rendering load will be put on the game if no LOD mechanism is used. You can often prevent a prohibitively high batch count if all this geometry is properly combined into a reasonably small number of vertex buffers.

Another option is to place instances of predefined detail objects in the scene. This method performs much better when considering memory consumption. Again, CPU load is not an issue. If you can guarantee hardware with support for DirectX 9c Summer SDK-type instancing, this method can be very fast and will not add too many batches. Still, we need to spend precious

---

**HOLGER GRUEN** began working with 3D real-time graphics in 1993 and four years later joined a U.K.-based game middleware developer as a research engineer. He has also headed the development team of a now defunct German game company and worked for a simulation company developing computer image generator software. Email him at [hgruen@gdmag.com](mailto:hgruen@gdmag.com)

# MAKING THE GRASS



Virtual grass created using VBGT.

vertex buffer memory for the buffers containing per instance data. And if hardware instancing cannot be guaranteed, rendering performance could easily become batch limited.

There are methods (see Pelzer in References, page 22) that rewrite dynamic vertex buffers, filling them with the vertex data for nearby details. Alternatively, you can fill a vertex buffer with instance modifying data and use instancing to place objects.

When you implement a method like this, you have to make sure that the vertex buffers you write to do not get too big.

If they do, you might not have enough memory bandwidth for other effects using dynamic vertex buffers. Just consider the other effects that you want to display at a high frame rate.

You also have to ensure that your algorithm places geometry in a repeatable and consistent way, since you don't want the detail to look different when you return to a place in the scene. Finally, writing to the vertex buffer and especially

computing positions for placement on the surface of the game scene uses a lot of CPU time that you may need for other game features.

You can do better in terms of CPU load, batch count, vertex memory consumption, and simplicity of implementation using what I call a VBGT method.

In this article, I will first introduce the basic idea behind VBGT. After describing different usage scenarios, I'll further discuss how LOD transitioning can be implemented with VBGTs. Finally, I'll address how one might improve the VBGT technique.

## BASIC IDEA

The train of thought that leads to VBGT starts with the observation that we would like to define a texture that, instead of applying color to a surface, applies geometric detail to the surface. The notion behind displacement mapping (see Forsyth in References) comes to mind but does not allow encoding of arbitrary geometry. Ideally, we want to define a texture that describes complex objects relative to some point on a surface.

Let's initially restrict the points to be points on a triangle with some corners:  $v_0$ ,  $v_1$ ,  $v_2$ . Now every point on the plane of this generic triangle can be computed using the equation

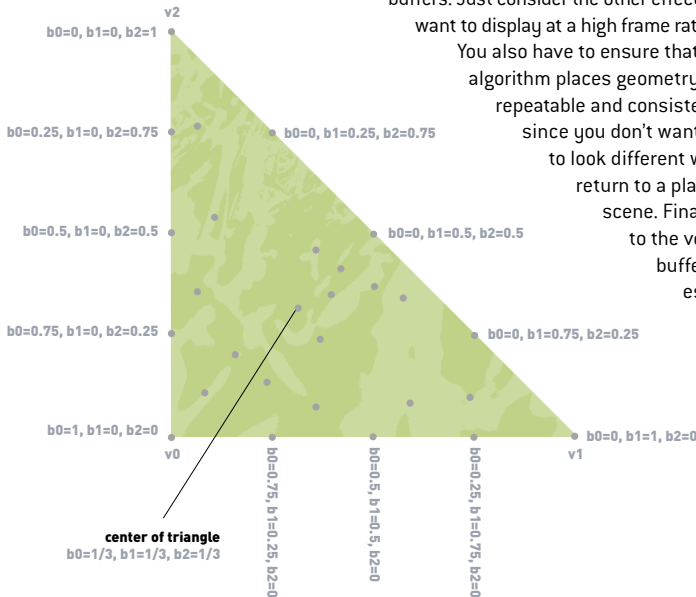
$$v = b_0 \times v_0 + b_1 \times v_1 + b_2 \times v_2$$

and appropriate values for  $b_0$ ,  $b_1$ , and  $b_2$ . If we further restrict ourselves to obey the constraint  $b_2 = 1 - b_0 - b_1$  these points will be inside the triangle. Please note that values defined as just described are usually called barycentric coordinates. Also note that barycentric coordinates can be expressed with just two of the three values. See Farin in References for more about the definition of barycentric coordinates.

Using barycentric coordinates, we can now code geometric detail relative to some point  $P$  on a generic triangle. The only thing we have to do is add the barycentric coordinates of  $P$  to the set of vertex attributes of every vertex of the geometric detail.

To get a better understanding of what "barycentric coordinates" mean, see Figure 1.

To turn what we have found so far into something that works similarly to a texture, we will assume that we have an object  $O$  that describes some geometric detail. Now we will



**FIGURE 1** The figure shows a triangle along with a set of points on it. Barycentric coordinates are shown for the vertices along the outer edges of the triangle and for the center. Points inside the triangle have been placed by randomly selecting barycentric positions.



create a vertex buffer that contains a number of copies of 0, each placed at a random position described by barycentric coordinates. Look again at Figure 1 and imagine, for example, a tuft of grass at every random point inside the triangle. The goal is to later map this vertex buffer, which we'll call VBGT from here on out, to ground triangles near the camera in order to enrich the visible and close-up game world.

I've provided some sample code (Listing 1, available on [www.gdmag.com](http://www.gdmag.com)), which effectively does just that. The code assumes an incoming pointer (`in_pDetailObj`) to a C++ object that can hold data describing the geometric detail. It computes the size of the vertex buffer used to hold `in_instanceCount` copies of the detail object in line 1.

In the next several lines, a vertex buffer and an index buffer are allocated to hold these new copies. For every copy of the detail object, the code now chooses a random barycentric position (see lines 10–13). It adds a copy of all vertices of the original detail object to the new vertex buffer, adding per vertex barycentric coordinates. Lines 24 and 25 leave room to randomly change the appearance of the copy of the tuft of grass to provide a non-uniform look. The rest of the code decides how to fill the index buffer used in conjunction with the vertex buffer that has just been filled.

Again, we could get away with just adding two of the three barycentric coordinates.

We now have created a VBGT that contains a number of copies of a detail model. We have further added barycentric coordinates to every vertex of these copies. So how is this vertex buffer different from statically coded local geometry?

The answer is that the origin of each detail object inside the vertex buffer can be moved to a position inside any triangle by using the barycentric coordinates found in every vertex by a vertex shader. To achieve that, we have to take the corners of the triangle we want to map the vertex buffer to and pass it to this vertex shader. The vertex shader then computes the final position of every vertex. See Listing 2, the second part of the sample code available online, which is a fragment of vertex shader code that shows how to do this. Listing 2 uses the barycentric coordinates embedded in the vertex attributes to move the actual vertex of the detail model to a position relative to its new origin on and in the triangle.

We wanted to place a random pattern of detailed objects, such as grass, on triangles near the viewer, which we can do by collecting all triangles near the camera by using the in-game

Code samples for this article can be found online at [www.gdmag.com](http://www.gdmag.com).

**THOMSON**  
COURSE TECHNOLOGY  
 Professional ■ Technical ■ Reference

## The game of a lifetime is inside you. **GET READY TO UNLEASH IT.**

|  |  |  |
|--|--|--|
| <p><b>Beginning Game Art in 3ds Max 8</b><br/>             ISBN: 1-59200-908-5 ■ \$29.99</p> | <p><b>Basic Drawing for Games</b><br/>             ISBN: 1-59200-951-4 ■ \$29.99</p> | <p><b>Advanced Visual Effects with Direct3D</b><br/>             ISBN: 1-59200-961-1 ■ \$59.99</p> |
|--|--|--|

|  |  |  |
|--|--|--|
| <p><b>Advanced 3D Game Programming All in One</b><br/>             ISBN: 1-59200-733-3 ■ \$59.99</p> | <p><b>In the Mind of a Game</b><br/>             ISBN: 1-59200-937-9 ■ \$39.99</p> | <p><b>Serious Games:</b><br/>             Games That Educate, Train, and Inform<br/>             ISBN: 1-59200-622-1 ■ \$34.99</p> |
|--|--|--|

Call **1.800.354.9706** to order ■ Order online at [www.courseptr.com/gamedev](http://www.courseptr.com/gamedev)

## REFERENCES

DirectX 9c Summer SDK, available online from [www.microsoft.com](http://www.microsoft.com)

Farin, Gerald E. *Curves and Surfaces for Computer-Aided Geometric Design: A Practical Guide*. London: Academic Press Inc., 1996.

Forsyth, Tom. "Practical Displacement Mapping," presented at GDC 2003.

Pelzer, Kurt. "Rendering Countless Blades of Waving Grass," in *GPU Gems*. Boston: Addison Wesley, 2004.

Tatarchuck, Natalya. "Richer Worlds for Next Gen Games: Data Amplification Techniques Survey." GDC Europe 2005, available online at [www.atl.com](http://www.atl.com).

collision detection routines. For all triangles found, we draw one copy of our VBGT of detail objects.

If the number of triangles is small enough, this operation will not add too many batches to the rendering budget. If it does, there are two ways to alleviate this problem.

First, you could pass the corner vertices of several triangles to the vertex shader and prepare a VBGT vertex buffer that contains several copies of the one created by Listing 1. Additionally, you would have to augment the vertex data of every vertex by an index that is used to access the right set of corner vertices. Using this technique, a set of ground triangles can be textured by a VBGT with one draw call.

The second option would be to use hardware instancing support and then use the corners of the collected triangles as per-instance data. You could then write to a system memory or dynamic vertex buffer filling in the triangle corners and draw all instances of the VBGT for all ground triangles with one draw call.

However, you would not use the processor to determine positions to place every single instance of the detail object anymore. All this data has been pre-computed and stored in the VBGT. The pattern of detail objects on the ground will always look the same since the order of triangle corners passed to the vertex shader will always be the same.

Detail objects, as mentioned previously, could consist of tufts of grass, little stones, lichen, bushes, trees, and so forth. In order to make the scene interesting, you could apply several VBGTs to ground triangles in order to improve the resulting visuals.

To prove that this theory really works, take a look at the demo and source code that accompanies this article [available online at [www.gdmag.com](http://www.gdmag.com)]. The demo creates a random height field based terrain and adds a grass VBGT and a rock VBGT to ground polygons near the camera.

## EXTENDING THE BASIC IDEA

There are several things that can lead to problems with the simple idea as it's presently laid out.

*Ground triangles with varying aspect ratios and/or areas.* VBGTs that look nice on triangles of a certain size and aspect ratio may look too dense or too loose on triangles with a much different size or aspect ratio. One way to get around this problem is to create a set of VBGTs in which each member of the set is optimized to look good on triangles of a certain range of areas or aspect ratios.

If you have scattered your geometry instances in an irregular pattern instead of a randomly displaced regular point pattern, then you can get away with just drawing a portion of a VBGT's vertex buffer to accommodate smaller or thinner triangles.

You could further augment the vertex format of the VBGT's vertices by adding an attribute that carries the maximum triangle area that a vertex is allowed to show up in. If all vertices of a copy of the small geometry object agree on this attribute, you can move off objects with the wrong size attribute to infinity inside the vertex shader.

The area of the triangle can either be passed to the vertex program as a parameter or computed in the vertex program from the vertices of the triangle.

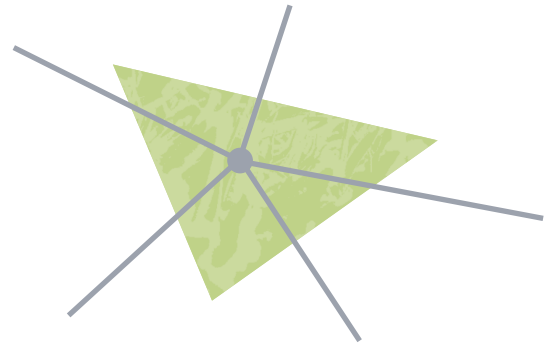


FIGURE 2 A partitioning with five sectors is shown.

*LOD.* The easiest way to handle LODs is to keep VBGTs that represent different distinct LODs of the appropriate high detail VBGT. You can generate lower LODs by creating the VBGT from lower LOD versions of the objects that are used to build the highest LOD VBGT. If this is not feasible, for example, if your model is a very simple tuft of grass, you can generate a lower LOD by scattering fewer objects.

Similar in spirit to the first solution (for how to deal with varying triangle areas), we can turn the area of the projected triangle into the value that's used to select LOD levels or to perform LOD transitioning. In addition to the three corner vertices of a triangle, we also pass the projected area of the triangle to the vertex shader handling the VBGT. We further add an attribute to every vertex of the VBGT. This attribute carries an acceptable minimum projected triangle area. If the projected area stored inside the vertex is found to be smaller than the area of the triangle passed to the vertex program, the vertex is moved to infinity. This way, the attribute can also be used to perform smooth transitioning.

By using a projected area as an LOD scaling factor, we will also make sure that polygons which are at grazing view angles will receive only a small number of object instances; this tip is particularly handy when working on fur or grass.

So far, all the ways I've described to switch LODs will result in visible popping if not used extremely carefully. In order to make the process smoother, we can again use the area of the projected triangle as a guide. For completely opaque objects, such as stones, you can achieve smoother LOD transitions by fading out the alpha value of vertices of some parts of the VBGT. Or, you could flatten the stones inside the vertex program and also fade the color or texture toward that of the ground below the stone.

For grass tufts, depending on projected area, you can simply shorten them or fade them out starting at the tips.

In general, the kind of smooth transition to be used depends on the type of object that comprises the VBGT.

*Transparency issues.* Since the drawing order of the geometry inside a VBGT is fixed, correct alpha blended transparency is harder to achieve. The easiest way around it is to just alpha test. Many game developers alpha test in order to avoid sorting and ordering problems that might destroy their state change optimized drawing techniques.





If you can afford to use a multisample anti-aliasing drawing mode, set up alpha to convert to sub-pixel coverage masks. This is properly supported within OpenGL on all major graphics cards, and on at least one graphics card under DirectX.

If you cannot use multisample transparency and don't want to resort to pure alpha testing, there are still things you can do. If you're not transformation limited and can afford a second pass, in a first pass just draw completely opaque pixels, without blending but testing for an alpha that equals one. For the second pass, enable blending and test for alpha smaller than one. This minimizes the artifacts for not properly sorting all transparent parts of the VBGT.

The approach just described can get pretty expensive. One way around the second pass might be to partition space around the center of a triangle by choosing a certain number of sectors (see Figure 2).

We can now create one version of a VBGT vertex buffer for each sector that has its content sorted, to minimize blending error for a camera position inside that sector. Selecting the right vertex buffer to be used as a VBGT will then allow using approximately sorted geometry without a second pass.

## FURTHER EXTENSIONS

I've only been adapting the origin of the detail objects of a VBGT inside the vertex shader. Consider what you could do if, in addition to just the vertices of a triangle, you could also pass normals and bi-normals to the vertex shader. Inside the vertex shader, you could set up a local frame to transform the vertices of the building block objects, thereby aligning objects along interpolated normals and bi-normals. And in addition to normals, you might add other attributes, such as color or size values that would be used to further modify the VBGT.

Of course, we can implement animation of VBGT inside the vertex shader, for example, by varying the positions of the topmost vertices of the tufts of grass based on some wind pattern. Say you had a sphere; a vertex shader could, for every tuft of grass that intersects the sphere, bend its topmost vertices away from the center of the sphere and push grass tops away from moving objects.

The upcoming DirectX 10 will introduce geometry shaders, which will allow data amplification (see Tatarchuck in References). This will allow us to procedurally generate geometric detail. Still, VBGT will most probably run faster than geometry shaders on early DirectX 10 hardware, and it will also run on every DirectX version that supports vertex shaders. ❖



## "Premier software for in-game voice audio"

- Reliable automatic lipsync
- Fast, nonlinear animation, gesturing, and game control
- Automatic closed captioning
- Flexible speech data that can fit any animation architecture

Annosoft's **Text Based Lipsync SDK** is one of a kind.

It provides the richest feature set and highest quality of any lipsync technology. It is used to lipsync, gesture, and close caption very long or very short audio files.

It supports English, French, Spanish, German, Italian, and Russian and opens up whole new spectrum of automation techniques.

The **Textless Lipsync SDK** is the fastest way to generate quality lipsync.

It produces realistic mouth animations without the need to write a text transcription and works with all languages.

In a pinch, there is no easier way to generate mouth animation.

**Whether you have 100 or 10000 audio files, Annosoft has C++ SDKs and tools that can get the job done. Find out why many of the top game studios use Annosoft's lipsync technologies. Visit us at [www.annosoft.com](http://www.annosoft.com) to learn more.**

*"The software has been a dream to use - certainly on of the best 3rd party packages I've worked with. I'll be recommending you."* - Mark Adami, Linohead Studios

*"Annosoft lipsynced a total of 10,000 lines in four separate languages; it's worth its weight in animators. Support has always been prompt, effective and rarely needed."* -John Scott, Raven Software





# CHOMPING AT THE BIT

WIDELOAD GAMES' STUDIO EXPERIMENT

BITES BACK WITH STUBBS THE ZOMBIE

## THREE YEARS AGO, AFTER LEAVING THE

helm at Bungie I found myself without a job and on the sidelines of the game industry. It was weird. After spending six months tooling around at home, I realized I had to get out of the house and do something—be productive, set an example for my kids, take over the world—that kind of thing. Games are what I know and love, but I faced a real dilemma in trying to figure out how to get back to making games on my own terms.

To be an independent developer in the current climate of publisher consolidation and rising costs seemed impossible, but somehow Wideload was created. I challenged myself to create a company with a set of commandments essential to my personal and professional happiness.

## THE COMMANDMENTS

*First Commandment:* We shall establish our game's creative direction.

*Second commandment:* We shall own our intellectual property.

*Third commandment:* We shall not let a third party determine our success, such as the publisher who's doing (or not doing) the marketing, or the funding source (likely a publisher) making demands that are not in-line with our goals.

*Fourth Commandment:* We shall have a small manageable team. We don't want 50 employees making one game over three years in house (we want low overhead), and we don't want to suffer the churn of ramping up and down for projects.

---

ALEXANDER SEROPIAN began his career as the founder and CEO of Bungie Software. In 2000 he negotiated the merger of Bungie into Microsoft, moved west, and launched HALO for the Xbox. After leaving Bungie in 2002, he founded Wideload Games and is currently the company's president. Send comments about this article to [asero pian@gdmag.com](mailto:asero pian@gdmag.com).





## GAME DATA



**DEVELOPER**  
Wideload Games

**PUBLISHER**  
Aspyr Media

**NUMBER OF FULL-TIME DEVELOPERS**  
12

**NUMBER OF CONTRACTORS**  
56

**LENGTH OF DEVELOPMENT**  
17 months

**LINES OF CODE**  
600,000

**RELEASE DATE**  
October 18, 2005

**PLATFORMS**  
Xbox, PC, and Mac

**DEVELOPMENT HARDWARE**  
Xbox development kits, various PCs, and a dual processor Mac G5

**DEVELOPMENT SOFTWARE USED**  
Photoshop, 3ds Max, CharacterStudio, Bink, QuickTime Pro, ProTools, Audacity, SmartFTP, SourceOffSite, Visual Studio, JIRA, phpBB



STUBBS' co-op mode.

## DEVELOPMENT THEORY

In a time when ten million bones hardly gets you a game, and development teams are crossing double-century headcount, I realized the key to these commandments was size—as in small size. Figuring out how to make quality games with a small team would solve the challenge of making original games, while remaining independent and having a shot at surviving that way.

Here's how the theory works. If the team is small, the overhead is low. Time equals money, so low overhead gives you lots more time to experiment and prototype (good for originality).

Additionally, every project starts small and ends big. But if you think of each project as a cycle of life, your company goes extinct pretty quickly when you have 75 people wrapping a project and then you only need 10 or so to start the next one. Staying small was the key.

## EVERYTHING WORKS IN THEORY

There's no getting around the fact that shipping a major console title requires a lot of talented people. We took a page from Hollywood's playbook and decided to hire the "above the line" talent as the core Wideload team, but use outsourced independent contract talent to staff our production department. This would allow Wideload to have a consistent and manageable burn rate, yet work with a wide array of people who could provide the exact resources we would need. We also decided early on that we would license engine technology rather than create our own, as we did not want to spend the time investment and internal headcount cost to compete with the likes of Bungie, Id, and Epic.

Robots from Punchbowl: the City of the Future, STUBBS' game world.



Stubbs has the ability to possess his human enemies with his detached hand, often with explosive results.

## STUBBS THE ZOMBIE

Our first project is STUBBS THE ZOMBIE, in which Stubbs, a wisecracking zombie, takes on an ultra-modern city of the future using nothing but his own carcass and the weapons of his possessed enemies. The gameplay consists of eating brains to create zombie allies, piloting various vehicles, and possessing enemies via a detached hand. Though the subject matter is mature, the mood and atmosphere is light.

We decided at the very beginning that Wideload had to establish itself as a brand. Our games should have a common thread that identifies them as something uniquely Wideload, and that thread is humor. What I'm most proud of in STUBBS is that everyone who has played it, reviewed it, loved it and, well ... maybe didn't quite love it, agreed that it's funny.

STUBBS just shipped, and we built it using our outsourced production model. Putting our theory into practice was, politely put, a learning experience.

## WHAT WENT RIGHT

**1 BRAINSTORM TO LIVING ROOM IN ONE EASY STEP.** We only have 12 people on staff, and we all work in one big room. Our original business plan said nothing about the creative dynamics of large versus small teams, but this is probably the single best result of our model.



YOU'RE INVITED!

# The 6th Annual Game Developers Choice Awards



Join us at the **6th Annual Game Developers Choice Awards** to recognize the industry's greatest achievements.

The Choice Awards are the premier accolades for peer recognition in the digital games industry, celebrating creativity, artistry and technological genius.

**Nominations Open January 2nd**

VISIT [WWW.IGDA.ORG/AWARDS](http://WWW.IGDA.ORG/AWARDS) TO SUBMIT YOUR BALLOT



**EUGENE JARVIS**, *PRESIDENT, RAW THRILLS*  
**2005 Lifetime Achievement Award**

*"Receiving the Game Developers Choice Award for Lifetime Achievement was the ultimate 15 minutes of fame for a gamer. It doesn't get any better than that!"*



**Wednesday, March 22, 2006**

San Jose Civic Auditorium

San Jose, CA

**All GDC attendees welcome.**



PRESENTED BY  **igda** international game developers association



The many faces of death - Wideload commissioned a number of artists for Stubbs' concept art, and used the best of each.

Granted, the acoustics of our wood-floored, brick-walled loft space stink because I'm too cheap to buy rugs, but we have a comfortable, casual, light, and quick workflow that allows anyone to blurt out ideas and have them propped or chopped on the spot. A small team doesn't need a lot of hierarchy, management, team meetings, strike teams, and a lot of organizational overhead, so we can focus our energies on being creative. It's hard to believe we didn't consciously plan this, but our emergent culture turned out to be a great side effect of our model.

For example, substantive, creative conversations often began when someone cracked a joke and the rest of us riffed on it. Unlike at a big company, where authority is always out of earshot, at Wideload we put those gems straight into the game! The dance battle in STUBBS emerged this way.

## 2 FREEDOM FROM THE PUBLISHER'S SHACKLES.

Because we're small and our overhead is low, we were able to spend nine months working on various game ideas, mechanics, and story elements before we signed a publishing deal. This allowed us to take our time and experiment. It also gave us the power to say no to a few deals that weren't consistent with our commandments. We actually came within a few feet of signing a publishing deal early on with a major publisher, but it would have required us to give the publisher final say on creative decisions. We thought the deal could have been a disaster in a worst-case scenario, so ultimately we passed on it. Being able to reject a publishing deal could be considered having leverage, and for independent developers that's pretty rare.

We also had the ability to miss milestones. We didn't want to miss milestones, but because we could effectively manage our cash flow without having to live hand-to-mouth with each milestone payment, we avoided the situation of our publisher using money to take control of our project.

## 3 COST STRUCTURE.

A big problem facing game developers is the budgeting process. In order to secure project funding, you need to estimate cost and schedule on day one. Developers don't have a constant development platform (hardware and software are always changing) so making this projection requires a little voodoo science. Publishers tend to expect developers to stick to their budget and schedule, and often times that doesn't work.

STUBBS shipped four months behind our initial plan, but

two circumstances made our lateness workable. First, our overhead was low because our staff is small, and second, our contractors deliver specific assets to us for specific fixed prices, so if scheduling failed, the asset price did not increase like it would have if it were being developed by a full-time employee. As a result (to a certain degree) our contractors assumed schedule risk on our behalf.

Table 1 shows how much money the Wideload method saves per schedule slip, compared to the costs incurred by a fully staffed team.

There were also points in the development of the game when we needed a little time to design something before opening the asset floodgates. We were able to prototype animation ideas for feel and gameplay without having a bunch of animators waiting around for us to get the plan together. Once our animation system and mocap list was ready, we pulled the trigger and got it done quickly.

## 4 STAFFING.

We only had 12 seats to fill. The team was split evenly between artists, designers, and engineers. Basically, we had enough engineers to take the HALO engine and bend it to our will, enough artists to prototype and manage contractor submissions, and enough designers to write and script levels.

We assembled our core team very quickly. The prospect of trying to recruit and relocate 45 game developers to Chicago is daunting for a startup. If it costs roughly \$10,000 to recruit and relocate for each position, then we saved \$330,000 by keeping the team small. It's also hard to convince a publisher that as soon as they write that first check, 20 more people will come to work the next day.

There was also this great side effect to our model that when contractors didn't work out, we could simply fire them, which may sound cold and heartless, but the fact is that hiring presents a risk. When you bring someone on staff you create a semi-permanent bond. To break it is intellectually and financially expensive. When we had contractors that weren't cutting it, they were fired and we moved on. No one had to relocate. It was just business. No hard feelings. That fact made us quite a bit more maneuverable in terms of staff.

In addition, since our whole model is set up to find and manage contracted talent, we were able to add extra contractors when we needed to speed up production. We found ourselves pretty late in the game



Table 1

| Model       | Headcount | Monthly Burn | Cost of four-month slip |
|-------------|-----------|--------------|-------------------------|
| Wideload    | 12        | 100,000      | \$400,000               |
| Traditional | 45        | 375,000      | \$1,500,000             |

Calculations assume a fully-burdened headcount cost of \$100,000 per employee.



without enough scenery objects and no main menu. If we had to rely on already scheduled internal team members to handle these tasks, we'd have been screwed. We were able to find a lot of help externally to complete these two parts of the project.

The market for independent artists, designers, and programmers in the game industry is definitely growing. It's still small though, especially compared to film or television, where everyone is independent. Finding great talent is really important to us though, so we took advantage of all the tools we could to locate talent. We created a database of every company we could find that was doing contract work. We populated it with my personal contacts and those we found through resources like Gamasutra.com and Conceptart.org. We then began to make calls, evaluate reels, and meet with potential hires. Ultimately, and this should be no surprise, we had the best luck with the people that we had history with or who had experience with the tools we were using.

**5 THE INTERNET.** We had contractors all over the world contributing to STUBBS. SourceOffSite and instant messenger were invaluable tools for us. An important part of our process is making iteration time as short as possible. The more versions of something we did, the better it turned out. Giving our contractors the ability to create game-ready assets remotely and put them into our source control database allowed us to review

submissions immediately, in-engine, and in the form that end-users would see.

Once we had that process set up, the feedback loop with the contractors tightened a lot. We also used online forums to develop concept art. I never thought in a million years it would make sense to do concept art with contractors, but it worked great. For our main character, we were able to get different artists to contribute their ideas simultaneously, which we could review and white-board online in real time.

## WHAT WENT WRONG

**1 COMPLICATED TOOL CHAIN.** Engine licensing is important to our model. We don't have the staff or desire to spend years creating an engine from scratch. On STUBBS, we used the HALO engine, which was great for us because the engine kicks ass and we all knew how to use it. But, with that said, the HALO engine had never been licensed before; there was no documentation and no internet forum or third party support for it. Any training our contractors got on asset creation had to come from us. The HALO engine has its own unique asset path and idiosyncratic behaviors, so the learning curve slowed us down and wasted time.

In some cases, we decided it wasn't worth training a contractor to produce game-ready assets; we'd just bear the burden of cleaning and importing the assets ourselves. This was tremendously inefficient. In other cases, we had to devote art director time to basic training. I think in the future we'll dedicate someone to tool training so as not to create a production bottleneck internally.

**2 CONTRACTOR SELECTION.** We could have done a way better job of vetting potential suppliers. We got lucky and found some incredible people to work with, but our selection process had three problems. First, not every asset class that went into production had a shippable asset reference to go with it. We made it a goal to develop the first version of every object type (character, vehicle, environment, scenery, weapon) internally and send that as the level-setting reference,





but we got a little too enthusiastic in some cases to wait for that. This made it difficult to set the bar for everyone.

Second, not every contractor was required to submit a test asset. This was another goal we set,

but again, we jumped the gun in a few instances, which was a mistake. Omitting this step allowed incorrect expectations to emerge and caused underbidding. In the future we'll set expectations of quality and scope for potential contractors before they submit a bid and start working.

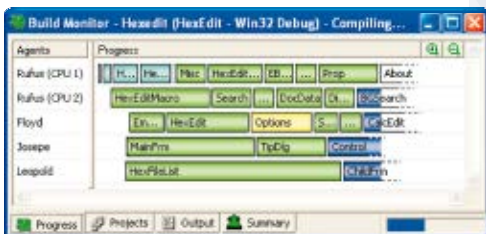
Third, we underestimated how important good management and art direction is for contractors. We worked with one art house in particular that was stretched too thin and sold us on the A team, but gave us the B team. They experienced a bad cash flow squeeze during production, which strained our relationship. Additionally, their art director was not experienced enough, which made it really difficult for us to manage quality across their team. Had we discovered all this in the selection process, we would not have had to waste time replacing the contractor during the middle of production.

Stubbs' chomping grounds.

## Slow C/C++ Builds?

**IncrediBuild** accelerates C/C++ builds by distributing compilation tasks across the network, cutting down build time by 90% and more!

Download  
FREE, Fully Functional  
30-Day Trial!



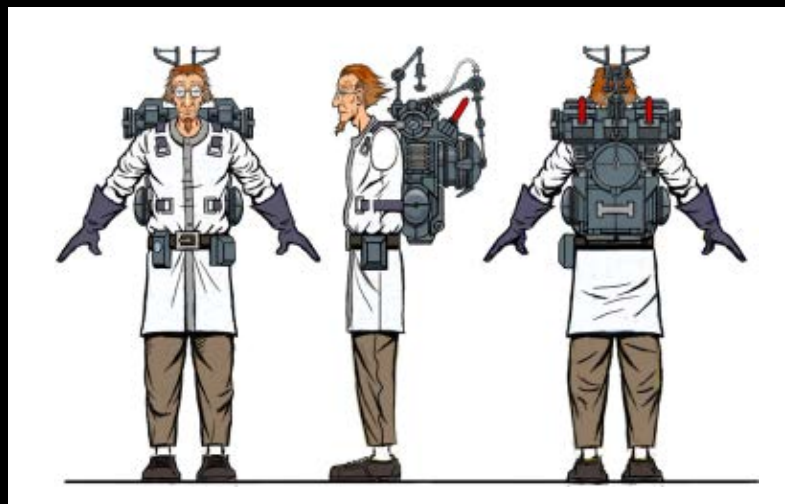
- Simple setup, requires no changes in code or settings
- Fully integrated with Visual Studio 6.0, .NET and 2005 IDE
- Supports all build platforms

XOREAX

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







Stubbs takes a cue from Ice-T.

**3 CONTRACTOR MANAGEMENT.** We underestimated just how much time was required to manage contractor submissions. We knew it would be time consuming, but even with that expectation, the combination of art directing and art production was more work than we had time to do. We were short on producers and our artists were scheduled to produce content on their own. We didn't have enough bandwidth available for reviewing submissions in a timely manner. We realized too late that our production phase requires an intense focus on the work coming in from the contractors. Focusing our internal efforts on the contractor feedback loop should have been a higher priority for our art direction team.

**4 NOT ENOUGH PRODUCERS, FALLING THROUGH THE CRACKS.** Our project director doubled as our producer. This was bad. We let a few aspects of production fall through the cracks and as a result ended up dealing with our scenery object build and the game shell during post-production. It's hard to believe we didn't have the foresight for this, but our model requires serious production management. There are tons of assets to track and multiple parties contributing to the process. To think we shipped this game without a full-time producer is nuts.

**5 CRUNCH AVOIDANCE SYSTEM—FAILED.** I had this crazy idea that since the bulk of the work was being done by contractors, we would be managing them to deadline and we at Wideload wouldn't have to work crazy long hours. This logic was used to form the basis of our "crunch avoidance system," and it was an abject failure.

We crunched for a solid three months, which isn't too bad relative to past experiences, but is way worse than zero. Because we let some major components slip into post-production and were four months behind schedule, and we let ourselves get behind on

contractor approvals, we ended up with more than post-production tasks during our post-production phase. Quality of life is a big issue for game developers. It certainly is for me, having young kids at home. I still hope to create a better work/life balance using this model.

On future projects, we will endeavor to keep production phase deliverables comfortably within our production phase. For us, the key to this is good production management and timely feedback to our contractors. If we can focus on post-production during the last three months, we can avoid working double duty through the end of the project.

#### EXPERIMENTING ZOMBIES

Throughout the development of STUBBS, I received lots of calls from industry friends who asked me, "How goes the experiment?" STUBBS was released for Halloween and has gotten some solid reviews. Wideload survived the process and will strike again.

Pragmatically speaking, the experiment has been a resounding success. Adhering to our commandments resulted in an original game that has succeeded in large part because we had the freedom to take risks. Had we done things the old-fashioned way, where the publisher has all the leverage, our lovable undead anti-hero might have met a tragic end at the hands of a focus group.

That said, I wear new battle scars and have tattooed new lessons to the back of my hand so as to never forget. For anyone hoping to follow our blazing trail, or at least learn something from our foibles and fables, I hope this article helps. ❖



# What was this "bad boy" doing at a wireless show?

## Making his wireless debut.

**Come see what's hot in mobile music, wireless fashion  
and entertainment this year at CTIA WIRELESS 2006.**



**CTIA WIRELESS 2006**

A Department of CTIA-The Wireless Association®

**The Most Important Technology Event of the Year!**

Opening Keynote, **CTIA WIRELESS 2005**  
**Sean "Diddy" Combs** *Founder & Chairman*  
**Bad Boy Entertainment Group**

**April 5-7, 2006**

Las Vegas Convention Center Las Vegas, NV USA

LOOK FOR SCHEDULES CO-LOCATED CONFERENCES EDUCATION HOUSING @ [WWW.CTIA.ORG](http://WWW.CTIA.ORG)



MICK WEST

## » THE INNER PRODUCT

# MATURE OPTIMIZATION

### LAST MONTH, I DESCRIBED AN

optimization that needs to be done early in a project, if done at all. Here, I expand on the theme of early optimizations and present a few examples of what I call “mature optimization.”

**Conventional wisdom ... calls for ignoring efficiency in the small; but I believe this is simply an overreaction.**

—Donald Knuth

**MICK WEST** was a co-founder of Neversoft Entertainment. He’s been in the game industry for 17 years and currently works as a technical consultant. Email him at [mwest@gdmag.com](mailto:mwest@gdmag.com).

### CONVENTIONAL WISDOM

Every year, a fresh crop of young programmers enters the games industry with the wisdom of books and lectures swirling around their heads. One pearl of wisdom is drilled deep into their tender programming muscles, the pearl sometimes known as Hoare’s Dictum:

*“Premature optimization is the root of all evil.”—C.A.R. Hoare.*

Unfortunately, in the mind of the freshly minted programmer entering the game industry, this quote seems to turn into:

*“Early optimization is evil.”—Junior game programmer*

The legendary Donald Knuth most famously weighed in on this subject when he wrote these often quoted (and often misleadingly punctuated) sentences:

*“We should forget about small efficiencies, about 97 percent of the time. Premature optimization is the root of all evil.”—Knuth in Literate Programming, 1992.*

Nobody wants to argue with Knuth. It’s the equivalent of arguing against the second law of thermodynamics. However, the problem here is that too often people misunderstand what Knuth was getting at. Ask a game programmer when optimization should be done, and he or she will tell you, “After the code is written, we can profile it, see where the bottlenecks are, and eliminate them.”

And often, that’s exactly what happens. Toward the end of the project, we profile the code and find that 20 percent of our CPU time is taken up by a function like `HandleMessageEvent()`, so we track down

why that’s taking so much time, then fix it.

Then we tackle our next function on the profiler’s list, maybe one that takes 10 percent of the CPU time. We fix that, and then tackle a few more. Our game now runs 30 percent faster, and our fattest function takes just 1 percent of the CPU time.

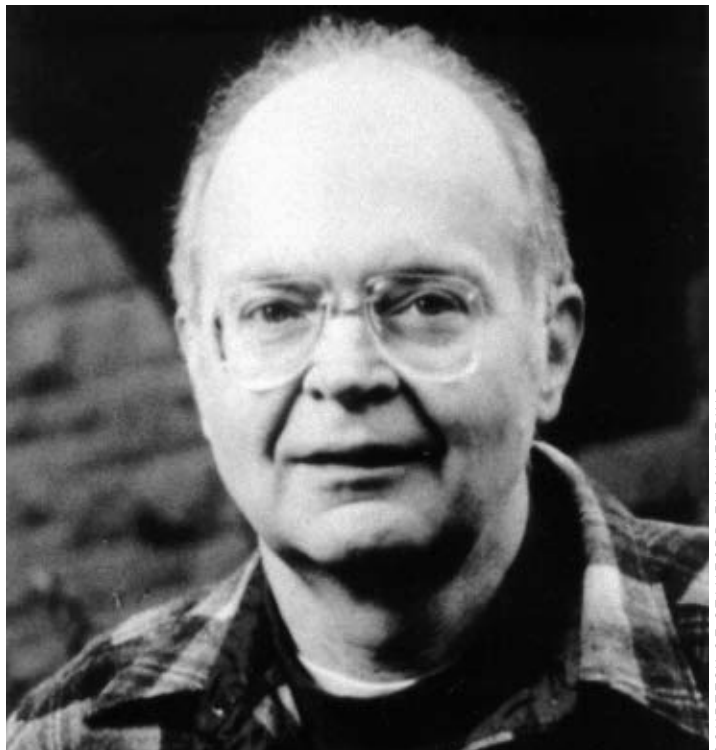
However, if after that the code is still too slow, it’s not because of a few weighty functions, but rather because there’s a general problem with all the code. The code is just generally inefficient. By ignoring optimization until the end of the project, you have ended up with “sick code” that contains multiple

inefficiencies too deeply ingrained within the fabric of the code to be removed in a timely manner.

### WHAT KNUTH REALLY SAID

When Knuth wrote, “*We should forget about small efficiencies, about 97 percent of the time,*” he was actually defending something he had done on the previous page of his book: optimize a loop to make it 12 percent faster, using “goto.”

Knuth then states my main point here, something contrary to what most programmers initially seem to take from Hoare’s Dictum:



Donald Knuth, Professor Emeritus of Computer Science, Stanford University

COURTESY OF STANFORD ENGINEERING

## We should forget about small efficiencies, about 97 percent of the time

*"The conventional wisdom shared by many of today's software engineers calls for ignoring efficiency in the small; but I believe this is simply an overreaction to the abuses they see being practiced by penny-wise-and-pound-foolish programmers, who can't debug or maintain their 'optimized' programs."*  
—Knuth, 1992

There you have it. Conventional wisdom was wrong in 1992. Sadly, conventional wisdom continues to be wrong 14 years later. Programmers—even game programmers—are still starting out with the misconception that optimization is evil, and so they ignore optimization until the end of the project.

### MATURE OPTIMIZATION

What is evil here is a very specific type of optimization—premature optimization, which is optimization done early in development without a real understanding of whether it's necessary, and which may have an overall negative effect on the project. Given that we have an evil type of optimization called "premature optimization," it follows that there is a contrasting one that's not evil, which we naturally call "mature optimization."

A mature optimization is any optimization done early in development that you know in advance will provide a significant performance boost without unmanageable side effects. Mature optimizations are often well-known techniques that were successfully used

before. They include small local modifications, coding standards, and architecture-level design decisions.

### GAME OPTIMIZATION

Game programming is becoming more complicated every year. I think it's now more complicated than either Hoare or Knuth really could appreciate. A game engine comprises a large number of systems, each handling a part of the game; each system is itself often vastly more complex than an entire game of 20 years ago.

Since each system is so complex, and each system only contributes a little to CPU usage, then individual optimizations are not often going to have much effect on overall performance. To ensure our game runs as fast as possible, we need to optimize our code as it's developed, using every mature optimization available to us.

Developing a game is more and more about developing the content for that game, rather than developing the code.

**DON'T MISS  
THIS YEAR'S  
MOBILE GAMING  
MUST ATTEND EVENT!**

#### MOBILE TRACKS:

Mobile Vision  
3D Development  
Audio & Sound Design  
Deals & Distribution  
Game Design  
Planning Ahead  
Production

# Game Developers Conference Mobile

March 20-21, 2006 • Fairmont Hotel • San Jose, CA

GDC Mobile 2006 is the definitive mobile gaming event for leading industry professionals working within this exploding entertainment sector. Mobile game developers from around the globe will gather to debate the future of their medium with executives from leading network operators, MVNOs, content aggregators, advanced technology and infrastructure providers, and major console game publishers. No other conference offers in-depth technical and business-oriented seminars that define *What's Next* for mobile games.

A complete list of GDC Mobile 2006 event and speaker information is available at [www.gdconf.com/conference/gdcmobile.htm](http://www.gdconf.com/conference/gdcmobile.htm).

PLATINUM SPONSOR:



GOLD SPONSOR:



RECEPTION SPONSORED BY:



TO ATTEND GDC MOBILE, YOU WILL NEED EITHER A VIP, MOBILE OR MOBILE PLUS PASS.  
REGISTER BY FEBRUARY 15, 2006 AT [WWW.GDCONF.COM](http://WWW.GDCONF.COM) AND SAVE UP TO 35%  
USE PRIORITY CODE **PAMAXX** WHEN REGISTERING

When developing that content, you cannot really get a good idea of how the game plays unless the code is running at the speed you hope it will run when it ships. This is especially true of action and sports games, where the "feel" of the game is a vital part of the gameplay and essential in establishing a difficulty curve. Thus, it's beneficial to do as much optimization as possible in the early stages of a project.

Most games are real-time applications. A very large chunk of the code is running 30fps (or hopefully 60fps). In developing business applications, a response time of a large fraction of a second for a single transaction is quite acceptable. Game programmers need to advance the state of the entire game in just 0.0166 seconds. It's very rare for a single transaction or operation (say total collision detection) to be budgeted for more than one millisecond (0.001 seconds).

Processing of triangles and other graphics primitives has mostly been off-loaded to the GPU. The CPU is mostly

concerned with iterating over game objects and doing things with them. Optimization of games even a few years ago was often heavily weighted toward optimizing the rendering loop. The balance here has been shifting over the years to the bulk of the CPU time being used by game logic, AI, and physics. The result is that there is less low hanging fruit to optimize.

The relatively high power of modern CPUs has also contributed to a shift toward physics calculations taking up more and more of the CPU's power. Again, it's important that optimizations are completed early so that the designers can design their game scenarios within well-defined limits as to the complexity of the physics simulations they can use, specifically the number of objects in a scene. If you leave the optimization until too late, then you're either going to have a scene that runs too slow (and may require some last minute butchering), or a scene that is much lower in complexity than the engine can support, and which

looks weak compared to the competition.

Now that we have an overview of the issues in place, I want to map out a few representative examples of mature optimizations.

## AVOID ALLOCATIONS

Memory allocations are expensive. If you have some system doing a large number of small allocations from the main heap on a continuous basis, then consider making some kind of custom allocator. A good example is a particle system. If you know a particle system has a maximum of 1,024 particles, then you should use a high-speed pool rather than a heap.

This kind of memory system optimization is used widely in game development. However, it falls squarely into the category of a mature optimization, since it can be tricky and dangerous to replace one method of memory allocation with another toward the end of a project.

CONTINUED ON PG 36


ATTEND THE SERIOUS GAMES SUMMIT AT GDC:06

-  EDUCATION
-  GOVERNMENT
-  HEALTH
-  MILITARY
-  CORPORATE
-  FIRST RESPONDERS
-  SCIENCE



# SERIOUS GAMES SUMMIT

SAN JOSE, CA  
MARCH 20-21, 2006

# GDC

AT THE SAN JOSE MARRIOTT

**REGISTER BY FEBRUARY 15, 2006**  
**AT [WWW.GDCONF.COM](http://WWW.GDCONF.COM)**  
**AND SAVE UP TO 35%**  
**USE PRIORITY CODE PAMASX WHEN REGISTERING**

 The **Serious Games Summit** spotlights the use of interactive game technology within non-entertainment sectors in the rapidly growing serious games market. The summit provides a forum for game developers and industry professionals to examine the future course of serious games and their applications in areas such as education, public policy, science, national defense, homeland security and corporate training.

LEARN MORE ABOUT THE SERIOUS GAMES SUMMIT AT [WWW.GDCONF.COM/CONFERENCE/SERIOUSGAMESSUMMIT.HTM](http://WWW.GDCONF.COM/CONFERENCE/SERIOUSGAMESSUMMIT.HTM)

WWW.GDMAG.COM **35**

CONTINUED FROM PG 35

## AVOID RTTI

Real-time type inspection (RTTI) is what allows you to use `<dynamic_cast>` on a base class pointer to see what type of derived class it actually is. The problem with RTTI is it's very slow. It depends on the specific compiler implementation, but on many compilers a `<dynamic_cast>` is implemented using the `strcmp` function on the name of the class. If your profiling shows that `strcmp` is registering above 0.1 percent of CPU time, then suspect RTTI.

Instead of using `<dynamic_cast>`, you should first identify those classes that require some kind of run-time type inspection, consider if you really need it, and then incorporate a type member variable into the base class, with an `enum` providing the type numbers. Do your own RTTI by checking the type variable to verify the type (if needed), and then use `<static_cast>` to do the actual cast.

Avoiding RTTI qualifies as a mature optimization because it can be almost impossible to apply toward the end of a project. If a programmer starts implementing a complex system with many usages of RTTI, then it can be a significant undertaking to remove it. Adding your own optimized version of RTTI is best done early.

## AVOID ALIASING PROBLEMS

Aliasing is an insidious problem with compiled code. Aliasing itself is a straightforward concept—the same piece of memory can be pointed at by two different pointers.

Performance problems occur when the compiler assumes that the contents of memory can be changed by something other than the current code, and hence will continually refetch something from memory when you the programmer know that it's not going to change.

How you handle aliasing problems depends on your compiler. You could simply turn off aliasing entirely. You might be able to use the `"_restrict"` keyword selectively to tell the compiler that a pointer cannot be aliased. Both of these solutions have the characteristics of a mature optimization in that they are best done early in the project if they are able to have an effect safely. It's also a mature optimization in that it often takes a mature programmer to spot it as a problem. Detecting aliasing as an issue can

require that you look at the disassembly to see exactly what instructions are being generated.

## AVOID PER-FRAME CODE

Not everything needs to be done at 60fps. Making sure certain aspects of the game logic run at a frame rate slower than the screen refresh rate saves huge amounts of CPU time. Most physics will look quite reasonable running at 30fps. Some types of logic, such as pathfinding, can be spread over a large number of frames.

Time-slicing logic is another good example of a mature optimization, and many games use it. It can be a massive time saver, cutting as much as 20 to 50 percent off your CPU time. And of course, it's very difficult to add late in the project. The change in timing from running logic synchronized with the rendering frame advance to logic running independently can introduce all kinds of obscure timing bugs.

## AVOID ITERATIONS

Doing any single thing on a modern CPU takes no time at all. But we fall over when we need to do that single thing multiple times. Whenever you are iterating over some list of things in your game, consider how you might be iterating over fewer things, or not iterating at all.

For example, let's say we want to automatically target enemies that are within range. The simple way to do this is to iterate over all objects in the game, see which ones can be targeted, and of those, see which are within range.

The problem here is that the list of all objects might only contain a few that are actually targetable. You could optimize by maintaining a separate list of all the objects that can be targeted. When targetable objects are created and destroyed, they are added to and removed from this list. Then, when you want to target something, you just iterate over the shorter list.

## PROFILE INLINE FUNCTIONS

When profiling your code at a function level, most profilers won't be able to give you any kind of reading for inline functions because the optimizing compiler will interleave the inline functions instructions with the instructions in the calling function, making it impossible to tell when the inline function is being executed.

While this behavior is expected, it can

often hide problems with weighty inline functions that are either too slow or called too often. The culprit will often show up as some large higher-level function that calls several inline functions.

To get a clearer picture of exactly what's going on, you can turn off the expansion of inline functions just for the compilation unit that contains the high-level function, revealing the inline functions to the profiler. Recognize that the picture is not entirely accurate of what's going on, but it will give you an indication of the balance of CPU usage between the high-level function and the inline functions, and hence indicate on what you need to focus your optimization efforts.

## PROCESS OFFLINE

Load times for games need to be short. Yet too often we get to the end of a project and find the load times are over a minute. Some advance optimization here can save you a lot of trouble later.

The most obvious optimization is to load your assets all as one big file (ZIP, PAK, or WAD), rather than as individual files. This is a very mature optimization. All developers should use this technique, especially when developing for consoles—and yet it's surprising just how often the problem comes up. Developers often start by loading individual files, as at the start of development there aren't that many to load, making the load times bearable.

Toward the end of a project, though, the number of assets in the game rapidly increases, and load times soar. Patching a quick load system at this point can be very difficult, especially with limited memory available on a console to act as a buffer. You will be forced to do some kind of optimization at the end of a project.

Better to do it early. And do it maturely. ❌

## RESOURCES

Grinke, Sebastian. "Minimizing Agent Processing in CONFLICT: DESERT STORM," in *AI Game Programming Wisdom 2*. Hingham, Mass.: Charles River Media, 2003.

Hyde, Randall. "Embedded-system programmers must learn the fundamentals," *EDN*, May 26, 2005. [www.edn.com/article/CA601846.html](http://www.edn.com/article/CA601846.html)

Kirmse, Andrew. "Optimization for C++ Games," in *Game Programming Gems 3*. Hingham, Mass.: Charles River Media, 2002.



PHILIP OLIVER

## » BUSINESS LEVEL

# PIGEONHOLED

## When reputations define business

**DEVELOPERS ON THE WHOLE DON'T SELL** directly to the public. They sell their ideas to publishers.

The thing about selling games is that more often than not, you're selling a dream, since the game hasn't been made yet. Perhaps that dream amounts to a concept, some video, maybe even a demo, but until it gets funding and a full development schedule, it's not much more than an idea.

Publishers need to not only like the game idea, but also have confidence that the developer can deliver a high quality game on time. Image is vitally important.

Industry folks often say, "Making video games is not so much an investment as a gamble." If developers want to win work, they must make sure the publisher sees their business as an investment rather than a gamble.

### FIRST IMPRESSIONS COUNT

Smart publishers tend to seek out independent developers who make high quality games, act professionally, and always deliver what's expected or better. Independent developers can be focused, efficient, and can often do a better job than inefficient internal studios.

Publishers obviously look into the developer's track record: the quality of past games and the developer's ability, technology, and plan to deliver the new game. Now, this review process takes time, especially if the publisher plans to evaluate multiple developers, so it's only natural that they take shortcuts, like

sticking to people they know, or doing a quick background check on the web, or asking a few friends in the industry about the developer's reputation.

These shortcuts are natural and sensible, but from a developer's point of view, they mean it's crucial to become known. Obviously, developers must make sure that what is known about them is positive and will help win game contracts.

### BOILED TO BULLETS

Like it or not, publishers will form opinions about each developer, which will be boiled down to a few bullet points either in their minds or possibly in a database. These bullet points form very generalized types of developers, such as:

- *Developer A.* Medium size. Great for console racing games. A bit pricey. Pretty professional.
- *Developer B.* Great at kids' licensed games. Fair price. Will deliver high quality on time and budget.
- *Developer C.* Small. Not much capacity, but great conversion house and cheap.
- *Developer D.* Good for GBA licensed games. Fair price. Will deliver high quality on time and budget.

For a long time, we were Developer B—only we didn't want to make kids games exclusively. We had been pigeonholed though, and rarely had an opportunity to do anything else.

Blitz was established in 1990. Before that my brother and I, The Oliver Twins, were known for creating the DIZZY games. As a small, private, independent company with no external investment, there were only so many games we could produce. Over time, we became well known for our kids' licensed games, which is unsurprising if you look at the back catalogue: FROGGER 2, CHICKEN RUN, DISNEY'S LITTLE MERMAID, LILO & STITCH, BARBIE HORSE RIDING, and others. It's good business and we're proud of our games and reputation, but Blitz has become a victim of its own success.



With new label Volatile Games, Blitz can tackle more mature subject matter, such as **POSSESSION**.

### YOU CAN'T DEBUNK THE STEREOTYPE

For several years, we tried to change publishers' perceptions of us by simply making other kinds of games. But because we usually weren't a publisher's first choice (because we fit the Developer B bill), we only won games that were on tough budgets or tight time lines, such as MUMMY RETURNS and BAD BOYS II. Making these titles didn't help, as we weren't able to achieve the quality we all wanted or needed to become a publisher's first choice to develop mature games.

We needed the industry to think about us differently, but the danger was that if we succeeded, we could lose the kids' games. Our conclusion was to establish a new division and label that could build mature games while Blitz could continue to make titles for younger players.

Due to our size, more than 150 staff, we had the ability to maintain both sides. After much internal debate on the name, we decided to call the new division Volatile Games. Though building a new division does require a good chunk of change, we view the cost as an investment toward our long-term business. We're trying now to establish and manage the new image correctly, so that we're not forever held to only making kids' games.

### IMAGE CONTROL

As developers, your company will probably be summed up by a series of three or four bullet points in the eyes of publishers. What do you want those points to be? As a developer, your challenge is to decide what image you want, and get publishers to think this of you through good marketing. It's going to take time and money, but failure to do so could result in missed contracts and ultimately the success or failure of your business. We've learned that if you can't beat your reputation, it might be best to evolve. ❖



The Oliver Twins' **DIZZY** games made a name for the duo in the early years.

**PHILIP OLIVER** co-founded Blitz Games (then called Interactive Studios) in 1990 with his twin brother Andrew, following nearly 10 years of designing and writing games. He is currently the CEO of Blitz and is also a founder, member, and current board director of TIGA. Email him at [poliver@gdmag.com](mailto:poliver@gdmag.com).



## PIXEL PUSHER

# ANATOMY FOR ANIMATORS

## The dark mysteries of the shoulder

**WORKING OUR WAY UP THE TRUNK AND** spine since last month's column, we're now ready to round out the upper body with that most accursed of extremities—the shoulders. Experienced riggers approach the shoulder in about the same way parents approach changing diapers: It has to be done, but it's not going to be fun for anyone involved.

The shoulder is probably the most under-used part of the animator's toolbox. Many animators hardly touch the clavicles except for shrugs or violent throwing motions, but in fact shoulders are in motion all the time. It's almost impossible, in fact, to keep your shoulders from moving if you lift your elbow more than 20 degrees out from your side.

Riggers are well aware of the limits of conventional skinning techniques for handling three-axis joints, a topic covered in detail in "A Joint Effort" (February 2004). In the case of shoulders, the technical difficulties are compounded by mind-boggling anatomical workings. The main aim of this column, therefore, is to clear up what's actually going on inside a moving shoulder, so that riggers and animators will have a clearer idea of what they're hoping to approximate with those creaky smooth-skinning algorithms.

### SHOULDER COMPONENTS

The first question an animator is likely to ask about shoulder anatomy is, "Where do I put the clavicle?" However, it doesn't make sense to look at the clavicle in isolation. The clavicle is just one part of a very complex set of bones, muscles, and tendons known as the shoulder girdle,

which wraps around the ribcage, sort of like a set of football shoulder pads. The whole system works together to place the arm in the most advantageous position for leverage, and unfortunately for us it does so in a very non-linear way. So before we discuss how to place the shoulders, we need to understand the parts list for the entire shoulder girdle.

There are three bones in the shoulder girdle: the clavicle, or collarbone, the humerus, and the scapula, or shoulder blade.

**Clavicle.** The clavicle is the only part of the shoulder girdle that is actually anchored to the ribcage. The clavicle is plainly visible as a kind of shelf or platform running from the base of the neck toward the mass of the shoulder. In men, this form curves slightly upward, while in women it arches down, but in both cases it's pretty easy to spot.

When setting the bones in a model, it's easy to locate the clavicle pivot by looking for the jugular notch, the small depression at the base of the neck where the main neck muscles attach to the torso. The clavicle pivots are just below and to the outside of the jugular notch. If the neck muscles aren't clearly

visible on the model, you can estimate the distance at about half a hand-width from centerline of the body along the line of the visible clavicle. The pivot is very far forward, just below the skin, although usually you'll want to move your pivots in a little deeper.

**Humerus.** The upper bone of the arm is known as the humerus, not to be confused with the "funny bone," which is the head of the radius bone in the forearm.

The pivot of the humerus is easy to locate on most models. Looking at your model from the front, follow the slope of the trapezius muscle (the one that forms the broad triangular base of the neck) until it hits the line of the clavicles. Typically, this point will just about line up with the widest portion of the ribcage. Next, drop down about one-quarter of the way from the end of the trapezius to the height of the nipples, or, if you can't find the nipples, about two-fifths of the way to the armpit. From the side view, the joint should be halfway between the base of the clavicle and the back. This may seem rather far back, but the shoulder really ought to be well aft of the ear.

Although it's fairly easy to find the

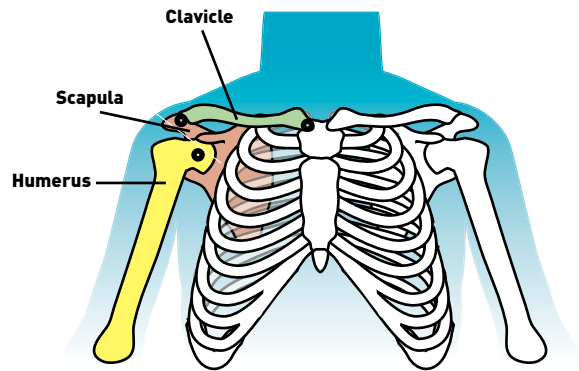


FIGURE 1 The bones of the shoulder girdle pivot points are marked with circles.

### FOR MORE INFORMATION

If you're curious about how real mechanics of the shoulder girdle function, or if you're just a masochist with a lot of time on your hands, there's a very complete and math-heavy description (with a couple of useful pictures) at <http://ligwww.epfl.ch/~maurel/Publish/PhD98/pdfs/Chap2.pdf>

**STEVE THEODORE** started animating on a text-only mainframe renderer and then moved on to work on games such as *HALF-LIFE* and *COUNTER-STRIKE*. He can be reached at [stheodore@gdmag.com](mailto:stheodore@gdmag.com).



# PART III



**FIGURE 2** In the rest position of the shoulder girdle, note how the scapula (brown) rests on the ribcage.

“correct” pivot for the shoulder, you’ll probably discover that you have to finesse the placement somewhat for good deformations. Models commonly push the round mass of the deltoids muscles up too high, forming an unnatural ball around the shoulder. In most cases it’s safer to err on the side of placing the joint a little higher than the anatomy books say to compensate for the extra bulk. Also, if your model is built in a 45/45/45 rest pose, with elbows lifted and brought forward, it’s likely that the model’s clavicles are lifted about 8 to 10 degrees from their natural rest position, so the pivot will have moved upward and inward accordingly.

**Scapula.** The tricky bones in the shoulder girdle are the scapulae, better known as the shoulder blades, two large triangular bones that make the familiar “wings” shape under the skin of the back. The wide base of the triangle is parallel to the spine, and the point opposite contains two bony “fingers” which create the socket of the shoulder proper. Locating the scapula is pretty easy, since the bone can be seen and felt under the skin of the back, even on characters who are not particularly buff. The pivots are equally easy to find, since they’re effectively the same as the pivots of the humerus.

## A SHOULDER TO CRY ON

Pivot placement for shoulders is easy, and that’s about all the good news I can share. The bad news is, alas, that what’s going on under the skin is extremely complex. In fact, it’s difficult to describe or even illustrate properly, much less simulate correctly. The Johns Hopkins University Biomechanics Project web site has a couple of the clearest animated examples of shoulder behavior (see Resources, page 40). There are also some extremely cool fluoroscopic x-ray movies that character technical artist Christopher Evans has on his web site (see Resources).

Here’s a vastly oversimplified description of what goes on as the shoulder moves. You can imagine the clavicle almost like the boom of crane, with the trapezius muscle filling the role of the cable that raises the boom. The lift and front-back swing of the clavicle define the possible positions of the shoulder joint proper. The clavicle “crane” is mounted to the front of—not on top of—the solid, egg-shaped mass of the ribcage; the looming bulk of the ribs limits the freedom of the clavicle to swing back or down, which explains why you have to lift your shoulders a bit to really squeeze your shoulder blades together.

On the other side of those ribs sit the scapulae. In the real world, they provide most of the power and leverage for the shoulder joint. To understand how they move, though, picture the scapulae as though they were magnetized to the mass of the ribs. Thus, as the clavicle rises, the scapula tilts up, almost parallel to the clavicle, with the base sort of sliding up across the ribs (the scapula is a “child” of the clavicle, and is hinged just above the shoulder proper). As the clavicle rotates forward, the scapula slides around and over the top of the ribcage. When the clavicle is fully extended forward, the scapula almost appears on the side, rather than the back, of the torso. When the clavicle is all the way back, the base of the scapula slides inward across the ribs, very close to the spine. If all that weren’t enough, remember that the ribcage itself is a curved surface, so the scapulae rotate in all three axes as they slide.

If that sounds bad, just be glad we didn’t try to cover all the muscles and tendons that drive the scapulae. It will be a while before we have real-time muscles that can cope with all that.

## BUILDING BETTER SHOULDERS

The real question to ask when building a shoulder is not “Where do I put the clavicles?” but rather, “Do I have to do all that?”

The familiar animation skeleton with a single clavicle bone is still acceptable for low-resolution models, tight bone budgets, or characters whose clothing makes anatomical details less important. But understanding the anatomy will at least give you some idea of the effects you should shoot for with vertex weighting and pivot wrangling. On the other hand, higher-resolution models may do better with a simple driven key setup that actually tries to reproduce some of the bafflingly complex behavior of the scapulae.

There is one important tool you need for just about any type of shoulder setup. It’s



**FIGURE 3** As the clavicle elevates, the scapula rotates in and down to stay in contact with the ribs.



**FIGURE 4** When the clavicle rolls forward, the scapula rides around the outside of the ribcage.

essentially impossible to get decent shoulder deformations without some method of removing twist from the humerus, regardless of how you build the shoulder. One of my previous columns, “Twist and Shout” (April 2004), discusses one good method for handling twisting shoulders. Despite which shoulder arrangement you build, remember that the shoulder girdle as a whole doesn’t twist—it only has two degrees of freedom.

### SINGLE VS. MULTI BONE SHOULDERS

A single-bone arrangement will have a hard time simulating the three-way relationship of the solid mass of the ribcage, the lever action of the clavicle, and the sliding motion of the scapula. The number one priority for single bone setups is making sure that the visual volume of the shoulder is preserved as the bone moves around. For this reason, you probably won’t want to use strictly anatomical pivots—the long throw of the natural clavicle tends to create a squishy, rubbery feel in the upper back. The most common response is to place the clavicles much farther outboard and further back than they “ought” to be—basically the same trick as building spines in the center of the torso, as we discussed last month.

If you are working on a high-resolution character for whom good deformations are particularly important (and bone count isn’t critical), you’ll get better quality shoulders by building working scapulae.

The pivots, you’ll remember, aren’t hard to place, but the behaviors are a bit daunting. A good solution has to do two things well: preserve the volume of the shoulder against standard smooth-skinning collapses, and illustrate the “scrunch” of the scapulae when the shoulders are rolled back. It would be nice to really capture the complex three-dimensional sliding motion of the scapulae, but for most game applications that’s overkill.

The clavicle and shoulder (glenohumeral) are simple, since they are basically standard kinematic bones. Remember, each scapula should be a child of a clavicle, with a pivot right on top of the pivot of the humerus. This way the elevation of the clavicle moves the “point” of the scapular triangle. The tricky part is getting the other two corners of each scapula to act as if they were sliding along the surface of the ribcage, and to do it without any attention from the animator.

The easiest way to achieve this—we’re talking relatively easy, not absolutely easy—is with driven keys. This shouldn’t be too difficult, but it is somewhat tweaky. Most driven-key solutions become slightly unpredictable when you’re dealing with more than one input, and in this case, you’ll need to drive the rotation of the scapulae according to both the elevation and the yaw of the clavicles.

If you’re looking for a simple solution, you can try rotating each scapula only around a vertical axis. For many applications, that will be enough. If you

go whole hog and let the scapulae rotate realistically in all three axes, you’ll probably get better results, but you’ll need to pay careful attention to the resources I’ve provided, particularly the ones that contain video clips.

If you’re a real IK enthusiast, you might consider using an IK-pole vector combination to simulate the movement of the scapulae. In this system, you would put an IK target where one corner of the scapula should rest, and plop the pole vector control at the other corner. You might be able to animate such a setup using only geometry constraints to keep the corners of your “bone” glued to a proxy ribcage object, but this will likely result in history dependency problems. Driving the position of the IK and PV targets with driven keys is simpler and less risky.

### COLD SHOULDER

Overall, the key thing to keep in mind if you do go for a multi-bone scapular setup is minimalism. Even if you were to perfectly capture the movement of the scapula bones, you wouldn’t be getting any of the complex bulging and sliding behaviors in the muscles that overlay it, so don’t drive yourself insane. Instead, be guided by your knowledge of what’s really going on inside to find the simplest, most bulletproof approximation.

And of course, let’s not forget the critical refrain for *Anatomy for Animators*: Go with what looks good, not what’s necessarily “correct.” ❄️



**FIGURE 5** When the shoulder is pulled back, the scapula slides across the ribs toward the spine.

## RESOURCES

Johns Hopkins  
University Biomechanics  
Project  
[www.biomech.jhu.edu/  
projects/shoulder](http://www.biomech.jhu.edu/projects/shoulder)

Christopher Evans  
[http://chrisevans3d.com/  
research.htm](http://chrisevans3d.com/research.htm)



NOAH FALSTEIN

» GAME SHUI

# SCHAFER ON CREATIVITY

**THERE HAVE BEEN MANY SPEECHES AND** articles on the creative rut that the industry has fallen into, with big studio releases following a few successful formulas, competing over tiny incremental changes to graphic engines or role-playing mechanics, revisiting familiar fantasy worlds or sports licenses.

But Tim Schafer, head of Double Fine Productions, is known for his creative

approach. In fact, he put it all on the line with the game **PSYCHONAUTS**, released in 2005. First, a disclaimer: I've been a fan of Tim's work for nearly 20 years, when he came to what was then Lucasfilm Games to work on **THE SECRET OF MONKEY ISLAND**. His resume featured a cartoon he drew to illustrate how much

he wanted the job, and his sense of humor and creativity were so evident we brought him in right away.

He went on to create other popular and varied games like **DAY OF THE TENTACLE**, **FULL THROTTLE**, and **GRIM FANDANGO** before leaving to found Double Fine.

**PSYCHONAUTS'** gameplay builds on typical 3D action platformer style, but the story, characters, and settings are radically innovative. The player controls Raz, a kid stuck at a summer camp intended to train children with psychic powers. Much of the game takes place literally inside the minds of various characters, including some who are

paranoid schizophrenics, megalomaniacs, or just plain weirdos. The game is full of Tim's characteristic humor, often sarcastic or sardonic, and is hysterically funny.

I recently talked to Tim about some of the creative risks he took, as well as the unusual techniques he employed to come up with such unique characters and settings.

**Noah Falstein:** Which came first: characters or the art design?

**Tim Schafer:** Mostly, the characters came first, and then the settings were created by imagining what was inside the characters' heads. That's why the premise was fun. What would it be like to be inside the head of some of the more extreme characters? We were constantly asking how do we keep it "krazy with a k," so it doesn't get too serious. And we always wanted to have a conspiracy theory guy.

**NF:** That was one of my favorite parts. How did you come up with the look for the inside of his mind?

**TS:** Paranoid people are said to be self-centered, in that they imagine the world as expanding circles revolving around them, so we set up the neighborhood to look like a web wrapping around him. He was based on a real person who used to hang around in front of the old Double Fine offices, sweeping up the street. He was really friendly but nuts, and every once in a while he would get angry and break his broom, claiming the government was watching him through bits of broken glass on the street, and downloading stuff into his head. Half of the dialog in that particular level is quotes from him.

**NF:** You spoke at a GDC 2004 about using a Friendster-style interface for character development, filling out data for all your fictional characters. How did that work?

**TS:** I tend to write in real time, getting ideas down quickly after months of

procrastination. That makes the dialog sound more natural, but you have to know the character very well. That's where the Friendster thing was great. It forced us to answer silly questions like, "What is the favorite CD of that character?" You'd never tell someone in the game what Raz's favorite album is, but having his backstory in your head helps you make him sound more believable. I had a huge document of backstory that I never even put up on the web site because it is important to imply and suggest the world is bigger than it is. That's cooler than letting the player see everything you've got.

**NF:** Are there other games you find creatively inspiring?

**TS:** I've been playing **KATAMARI DAMACY**, that's an amazing game with some really creative things. It's a lucky break it even happened. Also **ANIMAL CROSSING**, **NINTENDOGS**—but these days, unless you are in a position like Shigeru Miyamoto's, it's hard to get the chance to take a real creative risk.

**NF:** What do you think about publishers' current attitudes about creativity?

**TS:** It's become necessary to hide it these days. When publishers say a game sounds really creative, it's their way of saying, "it has a nice personality."

For our next game, one publisher said, "No one is into creative stuff now," and another said, "Our only red flag is innovation on the gameplay. Have you considered doing this as a GTA style game?"

I'm not trying to do gaming's *Un Chien Andalou* [Bunuel and Dalí's bizarre experimental film], just be innovative—but you can't pitch innovation. You just have to be smart and tenacious about hacking your way through the system just to get your ideas out there. ❖



**PSYCHONAUTS** takes place largely within the characters' minds.

**NOAH FALSTEIN** is a 25-year veteran of the game industry. His web site, [www.theinspiracy.com](http://www.theinspiracy.com), has a description of *The 400 Project*, the basis for these columns. Also at that site is a list of the game design rules collected so far and tips on how to use them. Email him at [nfalstein@gdmag.com](mailto:nfalstein@gdmag.com).



## » AURAL FIXATION

# THE VOICE OVER CHALLENGE

**VOICE OVER IS PERHAPS THE SINGLE** most time consuming and problematic aspect of game audio production today. Music can be retrofitted more easily to changing game designs, as can sound effects. Voice over, however, directly involves people and their time, right down to the time it takes them to speak words and syllables.

When it comes to the dynamic development process of games, the more people you have involved, the more problems you'll encounter. This is why, in voice over sessions, you need more options and strategies to deal with the problems that do come up.

Guest columnist Zak Belica wrote an excellent article ("Sound Principles: Voice Direction," April 2005) on what to do when you're in the booth, as well as how to prepare a script. In this article we'll tackle the overall process.

### SCRIPT PREPARATION AND CASTING

There are two important categories of script that need voice over files: cinematic and gameplay. Cinematic scripts are easier to lock down because it's easier for the writers to maintain an overall linear storyline than it is to hold together player-influenced gameplay. The gameplay script is also different in that game mechanics can be adjusted right up until your ship date.

With this in mind, as a producer or design lead, you should identify your overall story early and lock it down in pre-production, just the way a storyboard is done for a film. How that story is

presented can change (shots, cues, and lighting, for example), but the story itself should not. You need to verify this script early on, reading it aloud in its entirety. The story can be developed relatively independent of the game mechanics.

Next, when you've locked down the story, you should secure a cast, along with good character descriptions and concept art. This will serve you well if you're not doing your casting internally and have to deal with bids from agencies.

Keep in mind though, that your initial bid won't be 100 percent accurate until your gameplay script is locked down—and be prepared for multiple pickup sessions. Depending on your game type and the scope of your project, you'll have a varying number of pickup sessions. By the way, using internal staff for your voice over talent is getting riskier and riskier from a quality standpoint. In short, don't do it if you care even one iota about dramatic presentation.

### TALK AIN'T CHEAP

Finally, there's the subject of contracts, the part of the process that everyone in production despises. Unfortunately, the fact remains that legal responsibility is dictated by limiting risk and preventing loopholes that sneaky people have exploited in the past. So don't necessarily blame lawyers or unions. Blame the morons who took advantage of the system.

You should give yourself a solid month, minimum, to plan and execute a voice over contract of any size, whether using union or non-union actors. This will allow you to determine what you need, such as a casting director, celebrity talent, engineer, or session director, and how you want to negotiate.

Recent changes to union contracts have been making the contract process more difficult, so paying special attention to this task is vital to getting effective voice over in your game. Either hire a contractor or bring someone in full time, but make



Games like FABLE utilize a dedicated voice director to help manage ever-growing workloads.

sure you have a voice over supervisor devoted exclusively to dealing with unions, directors, talent, and studios, and have them interface with your legal team as well as the marketing and licensing departments (if necessary) so there are no obstacles preventing you from getting quality voice over into your title. If there's any one area of specialization (in addition to audio implementation engineers) that you'll need, it's this one.

### NASCENT NARRATIVE

Why is it so important to get voice over just right? Games are now competing with films for dramatic presentation through dialogue. On a major film (take *The Lord of the Rings: The Fellowship of the Ring* as an example) there's at least one person devoted to casting, one to three people devoted to dialogue editing, at least one dialogue coach, and the director, who will devote 20 to 30 percent of his or her time to the dialogue alone. Dialogue represents and shows how characters interact. Dialogue is a key vein for communicating the story.

In a game such as FABLE, which is an excellent example of an original story presented in a nonlinear way, there was one person in charge of voice alongside a team of three or so sound editors led by the venerable Russell Shaw. It's unclear how much time Peter Molyneux spent on dialogue himself, but it's safe to say that it was less than a film director would have been able to.

The bottom line is this: If dialogue is important to your title, be flexible, realistic, ready, and most importantly committed to your quality level. ❖

ALEXANDER BRANDON has been involved with game audio since 1994 and is currently the audio manager at Midway in San Diego, Calif. You can email him at [abrandon@gdmag.com](mailto:abrandon@gdmag.com).

*The Worldwide  
Leader and innovator  
in the creation of  
Interactive Entertainment*



# NOW HIRING

## **Jobs at Nintendo of America Redmond, WA**

3D/2D ARTIST (NST)  
UI ARTIST (NST)  
CONTRACT - CONCEPT ARTIST (NST)  
BILINGUAL PROJECT SPECIALIST  
ASSOCIATE SOFTWARE ENGINEER/  
SOFTWARE ENGINEER  
SR. SOFTWARE SUPPORT ENGINEER  
BUSINESS DEVELOPMENT MANAGER  
BILINGUAL PROJECT SPECIALIST  
MANAGER, MARKETING SUPPORT

## **Jobs at Retro Studios Austin, TX**

LEVEL DESIGNER  
CONCEPT ARTIST  
GAME DESIGNER  
STORYBOARD ARTIST  
GAME PROGRAMMER  
WORLD ARTIST  
CONTRACT - WORLD ARTIST

*The Award-Winning  
Developers of  
Metroid Prime and  
Metroid Prime 2: Echoes*



**For more information on these positions,  
please visit the web sites below:**



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



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

Why not change the world?

**Tenure-Track  
Faculty Vacancy**

**Assistant Professor of New Media/Electronic Arts**

We are seeking a motivated and energetic candidate who will contribute to the visual computing initiatives of the Department of Arts, including animation, digital imaging and interactive hardware and software development to begin in August 2006. This position includes teaching undergraduate and graduate studio courses and coordinating curriculum with other faculty, as well as advising on the configuration and operation of a lab facility. A priority for this position is competency in the area of computer animation and real time graphics.

The ideal candidate will be an experienced practicing artist and educator who uses techniques of animation in both linear and interactive media in their creative practice, and has experience working with professional digital platforms and applications. Additional desirable skills and interests may include robotics and/or interactive installation, bio-art, game development, and related theoretical topics. Candidate must be willing to become an active member of the Arts Department, with a strong commitment to creative work, research and teaching.

The Arts department at Rensselaer is the home of a highly visible program in integrated electronic media which includes the iEAR Studios (integrated Electronic Arts at Rensselaer), state-of-the-art facilities dedicated to interdisciplinary creative research and artistic development in audio, interactivity, video, computer imaging, animation, web, multi-media installation and performance. As an art program situated within the context of a technological university, we offer a unique creative environment in which to develop and realize cutting edge electronic art.

**Qualifications:** Professional activity and visibility as a practicing artist and previous experience in university teaching and organizational administration are desired. This position requires either MFA, PhD, or equivalent professional accomplishment and recognition.

Rank: Tenure Track, Assistant Professor

Salary: Commensurate with experience

To apply, send a resume, a cover letter describing your qualifications, your teaching philosophy, and a sample of your work. Please include the names and contact information (current phone, email, and address) of three persons from whom letters of reference may be obtained. Letters of recommendation may be requested after receipt of your application. Work samples may be in the form of DVDs, videotapes (MiniDV, DVCAM, VHS), websites, and CDs. Books and articles can also be submitted for amplification. Please also include the work of your students and sample syllabi. Applications will be considered beginning January 15, 2006, and will be accepted until the position is filled. Applications should be sent to: **Prof. Kathleen Ruiz, Chair/Animation Search Committee, Arts Department, Rensselaer Polytechnic Institute, West Hall, 107, 110 8th Street, Troy, NY 12180, tel: (518)276-4784, fax: (518)276-4730, email: ruiz@rpi.edu, http://www.arts.rpi.edu**

*We welcome responses from individuals who will bring diverse intellectual, geographical, gender and ethnic perspectives to Rensselaer's work and campus communities.*

*Rensselaer Polytechnic Institute is an Affirmative  
Action/Equal Opportunity Employer.*



Eugene, OR

BACKBONE



Emeryville, CA  
Vancouver, BC



Newport Beach, CA



imaginengine™

Boston, MA

**JOIN OUR TEAM**



Visit [www.f9e.com](http://www.f9e.com)  
for current job listings

**Do you think that  
the game industry  
can do better?**

**If so, we'd love to  
hear from you.**

We're looking for veteran game designers/developers who want to share their vision of the future with the students of tomorrow. The Game Development program at DePaul is accepting applications for full-time tenure-track faculty. Candidates without a PhD or MFA may be eligible for full-time non-tenure track positions. For more info, visit <http://www.cti.depaul.edu/news/jobs.asp>



**DEPAUL UNIVERSITY**

School of Computer Science, Telecommunications and Information Systems

**Chicago, Illinois**

**[www.cti.depaul.edu](http://www.cti.depaul.edu)**

DePaul University is committed to equality in educational and employment opportunities.

**Conjure up an Exciting Career!**

Currently Recruiting:

Lead Designer  
Producer- Mobile/Mini Games  
Programmers- Mobile/Mini Games

Apply: [careers@jagex.com](mailto:careers@jagex.com)

Jagex, UK based company,  
Developer and Publisher of  
the MMORPG 'RuneScape'  
seeks talented & passionate  
people to join us as we  
expand our games portfolio.



**VFS™**

**VANCOUVER FILM SCHOOL  
WHERE RESULTS MATTER**



**GAME DESIGN**

Work Hard, Play Harder

VFS. A proven education for everything you see, hear and experience in the entertainment industry.

For detailed program information call **1.800.661.4101** or visit **[vfs.com](http://vfs.com)**.

**GET EDUCATED**

Add a new dimension  
TO YOUR LIFE.



**COLLINS COLLEGE**

*Your passion. Our profession.*

Collins College offers a Bachelor of Arts degree in Game Design and a Bachelor of Arts degree in Visual Arts with a major in Game Art.

**CALL NOW FOR MORE INFORMATION** **1-800-850-0100**  
[www.CollinsCollege.edu](http://www.CollinsCollege.edu)

Main Campus – Tempe, AZ >> West Campus – Phoenix, AZ

*Collins College is accredited by the Accrediting Commission of Career Schools and Colleges of Technology (ACCSC). Financial aid is available for those who qualify. Career services assistance. Collins College West is a branch of Collins College. Not all programs available at all locations.*



by Rick O'Connor



Artwork by Full Sail Student-  
**William "Forrest" Crump**

## Bachelor's Degree Programs in **Game Development & Computer Animation**

school of

|                           |
|---------------------------|
| ▶ Computer Animation      |
| ▶ Digital Arts & Design   |
| Entertainment Business    |
| Film                      |
| ▶ Game Development        |
| Recording Arts            |
| Show Production & Touring |

**800.226.7625**  
**fullsail.com**

*If you're serious about your dream,  
we'll take your dream seriously.*

3300 University Boulevard Winter Park, FL 32792

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





# create YOUR WORLD

**CERTIFICATE PROGRAMS AVAILABLE**

Game Art & Design | 3D Animation | Visual Effects | Recording Arts

Intensive nine-month programs for the skills and tools you need to turn your ideas into reality. Financial assistance and career services available. **APPLY NOW.**

**CONTACT US TODAY: call 800.808.2342 or visit [www.cdiabu.com](http://www.cdiabu.com)**

[ GEEKED AT BIRTH. ]

You can talk the talk.  
Can you walk the walk?  
Here's a chance to prove it.  
Please geek responsibly.

[www.uat.edu](http://www.uat.edu) > 800.658.5744

Game Programming, Game Design  
Artificial Life Programming  
Computer Forensics and more!

IM Geek PH: 800 658 5744

## ADVERTISER INDEX

|  |    |
|--|----|
| Anark .....                              | 6  |
| Annosoft .....                           | 23 |
| Center for Digital Imaging .....         | 47 |
| The Collective .....                     | 44 |
| Collins College .....                    | 46 |
| Course PTR .....                         | 21 |
| DePaul University .....                  | 45 |
| Emergent Game Technology .....           | C2 |
| Full Sail Real World Education .....     | 46 |
| Jagex Ltd. ....                          | 45 |
| Nintendo of America .....                | 43 |
| Perforce Software .....                  | C3 |
| Rad Game Tools .....                     | C4 |
| Rensselaer Polytechnic .....             | 44 |
| ReplaySolutions .....                    | 9  |
| Seapine Software .....                   | 3  |
| University of Advancing Technology ..... | 47 |
| Vancouver Film School .....              | 45 |
| Xoreax Software .....                    | 30 |

## CORRECTIONS

The image that appeared in "Bringing Down the House" (Aural Fixation, November 2005) was uncredited. The photographer is James Lin.

In "Game Trains FDNY" (Heads Up Display, November 2005), the duration of the project was misstated. HAZMAT: HOTZONE has been in development for three years; Shanna Tellerman has worked on the game for two years. In addition, around 30 students from Carnegie Mellon University contributed to the making of the game.

We regret the errors.

*Game Developer* (ISSN 1073-922X) is published monthly by CMP Media LLC, 600 Harrison St., 6th Fl., San Francisco, CA 94107, (415) 947-6000. Please direct advertising and editorial inquiries to this address. Canadian Registered for GST as CMP Media LLC, GST No. R13288078, Customer No. 2116057, Agreement No. 40011901. **SUBSCRIPTION RATES:** Subscription rate for the U.S. is \$49.95 for twelve issues. Countries outside the U.S. must be prepaid in U.S. funds drawn on a U.S. bank or via credit card. Canada/Mexico: \$69.95; all other countries: \$99.95 (issues shipped via air delivery). Periodical postage paid at San Francisco, CA and additional mailing offices. **POSTMASTER:** Send address changes to Game Developer, P.O. Box 1274, Skokie, IL 60076-8274. **CUSTOMER SERVICE:** For subscription orders and changes of address, call toll-free in the U.S. (800) 250-2429 or fax (847) 647-5972. All other countries call (1) (847) 647-5928 or fax (1) (847) 647-5972. Send payments to *Game Developer*, P.O. Box 1274, Skokie, IL 60076-8274. For back issues write to *Game Developer*, 1601 W. 23rd St. Ste. 200, Lawrence, KS 66046-2703. Call toll-free in the U.S./Canada (800) 444-4881 or fax (785) 841-2624. All other countries call (1) (785) 841-1631 or fax (1) (785) 841-2624. Please remember to indicate *Game Developer* on any correspondence. All content, copyright *Game Developer* magazine/CMP Media, LLC, unless otherwise indicated. Don't steal any of it.

---

## » A THOUSAND WORDS

---



Softmax/Atlus'

## MAGNA CARTA

Magna Carta, a Korean RPG published in the U.S. by Atlus, features character art by Hyung-Tae Kim. Illustrations presented here depict the characters Orha (top) and Rothly (center). In-game art by the Softmax art team.



# Perforce.

## The *fast* SCM system.

### For developers who don't like to wait.



Tired of using a software configuration management system that stops you from checking in your digital assets? Perforce SCM is different: fast and powerful, elegant and clean. Perforce works at your speed.

[Fast]

[Scalable]

[Distributed]

Perforce's lock on performance rests firmly on three pillars of design. A carefully keyed relational database ensures a rapid response time for small operations plus high throughput when the requests get big - millions of files big. An efficient streaming network protocol minimizes the effects of latency and maximizes the benefits of bandwidth. And an intelligent, server-centric data model keeps both the database and network performing at top speed.

**It's your call. Do you want to work, or do you want to wait?**

**PERFORCE**  
SOFTWARE

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

All trademarks used herein are either the trademarks or registered trademarks of their respective owners.

# GRANNY<sup>3D</sup>

# Has all the tools!

## The World's Finest Run-Time Animation Engine

Silky-smooth b-spline playback • Multi-animation blends • Seamless transitions • Per-bone masking and feathering • Real-time IK • Flexible clocking • Motion extraction and prediction • Built-in animation L.O.D.

## Fast, Accurate Normal-map Generation

The results are so good, we even used our low-res demo model for this ad! That's our real-time Granny you're looking at. Her high-resolution counterpart that weighs in at over ten times the triangle count.

Blazingly fast • Tool-integrated • Generates tangent-space, object-space, and displacement maps • Copy high-res textures onto low-res models with any UV mapping

## Fast Mesh Deformation

Highly optimized CPU vertex deformers • Tangent space deformation • Generate GPU-compatible hardware-skinable vertex buffers • Full support for multiple vertex streams

## Powerful Exporters

Export from MAX, Maya and Lightwave • Mesh, animation and texture data • Advanced b-spline fitting and reduction • Full material graphs • Text and binary annotation • Animated custom attributes

## Model & Animation Previewing

Preview animations on any model • View transitions • Inspect all exported data and annotation • View mesh tangent spaces • Check texture-map assignment • Overlay bone structures

## Source Code Included

Full run-time engine source code included • Clean, cross-platform design • Modular and independently reusable components

For more information:

**425.893.4300**

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



THE BEST IN GAME DEVELOPMENT TECHNOLOGY



Bink  
Video  
Technology



Miles  
Sound  
System



Pixomatic  
Software  
Renderer