



CMP

United Business Media

»» PRODUCT REVIEWS DAZ'S HEXAGON 2 «« BETA PRODUCT PREVIEWS

DECEMBER 2006

# game developer

THE LEADING GAME INDUSTRY MAGAZINE



## POSTMORTEM: INTROVERSION'S DEFCON

»» STATE OF THE INDUSTRY

GAME PRIVACY WALKS  
THE REVENUE PLANK

»» CREATE ALL HUMANS

ARTIFICIAL INTELLIGENCE  
FOR OPEN WORLD GAMES

»» 2006'S BEST PRODUCTS

FRONTLINE AWARD  
FINALISTS ANNOUNCED

gamedeveloper  
**FRONTLINE**  
FINALIST '05

*"ReplayDIRECTOR rocks. I doubt we'd have found it otherwise. It turned out to be an occasional array overwrite that would cause random memory corruption..."*

■ *Meilin Wong, Developer, Crystal Dynamics*



## **BUGS. PETRIFIED.** **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 makes certain 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

77 NVIDIA

XBOX



26

## POSTMORTEM

### 26 DETONATING INTROVERSION'S DEFCON

Steam-distributed PC strategy title DEFCON was a completely independent project, continuing Introversion's reputation as "the last of the bedroom programmers." In this postmortem, team members from all aspects of the game's production weigh in about the logistics of releasing and popularizing a game with a sub \$100,000 budget in this era of bloated budgets and gargantuan team sizes.

*By Chris Delay, Vicky Arundel, Thomas Arundel, Gary Chambers, and John Knottenbelt*

## DEPARTMENTS

**2 GAME PLAN** *By Simon Carless*  
Second-Hand Bonanza

**4 HEADS UP DISPLAY**  
Front Line Award finalists, new console consolations, and serious games existential questions

**7 SKUNK WORKS** *By Tom Carroll, John Root, and Ronnie Ashlock*  
DAZ 3D's Hexagon 2, Alastair Macleod's PeelSolve beta, and Nevercenter's Silo 2 beta

**48 A THOUSAND WORDS**  
Capcom's LOST PLANET: EXTREME CONDITION

COVER ART: MORTEN ELGAARD PEDERSEN

## FEATURES

### 13 STATE OF THE INDUSTRY: VIDEO GAME PIRACY

Piracy has been a concern since games were first made available for sale. The PC is a particularly embattled platform as far as digital piracy, though console gamemakers have their fair share of troubles as well. Professionals from id, Macrovision, the ESA and other organizations explain where the industry's biggest concerns lie and what we might do in the future to disrupt pirate activities. Leave your eyepatch at home.

*By Paul Hyman*



13

### 21 CREATING ALL HUMANS

Artificial intelligence is of paramount importance in all games, but flaws therein can be even more noticeable in open-world games, in which the player is given a wide range of control. Using a data-driven AI architecture, Pandemic Studios created a flexible system for DESTROY ALL HUMANS II that alleviated many of the typical problems seen in open-world games.

*By John Krajewski*



21

## COLUMNS

**31 THE INNER PRODUCT** *By Mick West*  
Optimizing Asset Processing

[ PROGRAMMING ]

**34 BUSINESS LEVEL** *By Greg Ballard*  
Mobile Mindset

[ BUSINESS ]

**35 GAME SHUI** *By Noah Falstein*  
Designer Genes

[ DESIGN ]

**37 PIXEL PUSHER** *By Steve Theodore*  
The History Channel

[ ART ]

**40 AURAL FIXATION** *By Jesse Harlin*  
Laboring to Compete

[ SOUND ]





## SECOND-HAND BONANZA

### I'M LOOKING AT AN INVESTOR NOTE FROM COLIN

Sebastian of Lazard Capital Markets, who, as well as commenting on the latest NPD reports, has been discussing the emerging reseller market for the PlayStation 3 and Wii, claiming that eBay might become "the #1 new console retailer this holiday."

Sebastian explains, "With limited supplies expected this holiday for both Sony's PlayStation 3 and Nintendo's Wii consoles, we expect that many gamers will turn to eBay [and other online marketplaces] to locate new hardware."

"Already, our survey of completed auctions on eBay.com indicates that nearly 2,000 pre-ordered PlayStation 3 and Wii consoles have been resold on the marketplace to date. In addition, the consoles are selling on eBay for approximately two and one half times the suggested retail price of console hardware, on average \$1,500 for Sony's PlayStation 3 versus \$599 and \$550 versus \$249 for Nintendo's Wii."

### A BIRD IN THE HAND

Okay, so that's hardware. And a lot of people who buy these consoles at launch are simply planning to resell them, however insidious that is. But this statistic brings up an even larger point: the reselling of video games themselves.

At GDC London in September, I attended a lecture from Mike Vorhaus of research group Frank N. Magid Associates, who discussed the reselling of games in North America. As part of the company's 2006 research, Magid looked at the split between new and used game sales and found that 57 percent of U.S. console users have bought a used game in the last year, an extremely significant portion of the market.

"Clearly," Vorhaus said at the time, "used games in the United States have become a major piece of the overall pie," especially considering retail's propensity to "hassle you to buy the used game," as retail outlets get a larger percentage from selling used. In fact, Magid found that no less than 35 percent of its polled consumers who bought a used game originally intended to buy it new, meaning that over the course of a year, billions of dollars that would have been spent on new games are allegedly being spent on secondhand ones.

### TWO IN THE BUSH

So we did a little informal investigation of our own. We posed the following question to the readership of Gamasutra.com (*Game Developer's* sister web site): "As part of a community of game publishers and

developers, do you buy used video games or go out of your way to buy new ones? If you do buy used, are you concerned about the financial implications of the used game market, or does the free market trump revenue concerns for the game business?"

The results were somewhat surprising. According to the column's editor, the responses were mixed, though developers in particular often "prided themselves on being dedicated to buying new."

"[M]ost saw the effects of the games aftermarket as no different from any other consumer product market, be it music, movies, or cars. Most replies noted, though, that not only are direct downloads a solution and response to the problem, but that if developers and publishers wanted to curb used sales of newly released games, more incentives were needed—be it lower price or collectible insert—to make that new sale essential." (The published responses are available on Gamasutra's Question of the Week archives.)

### FREE BIRD, FREE MARKET

Even though a substantial amount of money—billions—is being taken away from publishers and, if you believe in trickle-down theory, the developers who create the games, people are very happy to buy used.

One respondent, for example, commented, "The fact that I work in the industry doesn't affect how I purchase my games. If I can find a used copy that looks new for a cheaper price, I have no second thoughts about the purchase."

But that's the point, I guess. Free world, free market. And until entertainment sales go completely digital, retailers like GameStop can take advantage of that fact.

One respondent on Gamasutra probably summarized the scenario best, commenting, "I will purchase a new video game over a used video game if one exists at the time of purchase. Until there is a new business model to combat used games sales, I choose to support the current one. ... Ultimately, the free market will trump revenue concerns and force video game companies to be creative in attracting buyers to purchase new titles over used. Digital distribution is one way to combat the issue because there isn't a market for used digital games."

Sweet, simple, and sensible, isn't it? ❖

Simon Carless  
Editor-in-Chief

#### EDITORIAL

##### EDITOR-IN-CHIEF

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

##### MANAGING EDITOR

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

##### FEATURES EDITOR

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

##### ART DIRECTOR

Cliff Scorsio [cscorsio@gdmag.com](mailto:cscorsio@gdmag.com)

##### CONTRIBUTING EDITORS

Jesse Harlin [jharlin@gdmag.com](mailto:jharlin@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* Dave Pottinger *Ensemble Studios*  
Ellen Guon Beeman *Microsoft* George Sanger *Big Fat Inc.*  
Andy Gavin *Naughty Dog* Harvey Smith *Midway*  
Joby Otero *Luxoflux* Paul Steed *Microsoft*

##### ADVERTISING SALES

##### DIRECTOR OF SALES

Steve McGill [smcgill@cmp.com](mailto:smcgill@cmp.com) t: 415.947.6217

##### GLOBAL SALES MANAGER, RECRUITMENT & EDUCATION

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

##### SR. ACCOUNT MANAGER, SOUTHWEST, CONTRACTORS, & MARKETPLACE

Jasmin Davé [jdave@cmp.com](mailto:jdave@cmp.com) t: 415.947.6226

##### ACCOUNT MANAGER, EAST COAST, U.K. & EASTERN CANADA

Cecily Herbst [cherbst@cmp.com](mailto:cherbst@cmp.com) t: 415.947.6215

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

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

##### ADVERTISING PRODUCTION

##### ADVERTISING PRODUCTION COORDINATOR

Kevin Chanel  
REPRINTS Cindy Zauss [czauss@cmp.com](mailto:czauss@cmp.com) t: 516.562.5000  
Julie A. Douglas [jadouglas@cmp.com](mailto:jadouglas@cmp.com) t: 516.562.5092

##### CMP GAME GROUP

##### VP, GROUP PUBLISHER APPLIED TECHNOLOGIES

Philip Chapnick

##### VP, STRATEGIC MARKETING

Michele Maguire

##### GROUP DIRECTOR

Kathy Schoback

##### DIRECTOR OF MARKETING

Tara C. Gibb

##### CONFERENCE DIRECTOR, GDC

Jamil Molelina

##### SENIOR CONFERENCE MANAGER, GDC

Meggan Scavio

##### EXECUTIVE WEB PRODUCER

Peter Leahy

##### EDITOR-IN-CHIEF, GAMASUTRA.COM

Simon Carless

##### ASSISTANT EDITOR, GAMASUTRA.COM

Frank Cifaldi

##### CIRCULATION

##### CIRCULATION COORDINATOR

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

##### CIRCULATION ASSISTANT

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

##### CIRCULATION ASSISTANT

Andrea 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 TECHNOLOGY MANAGEMENT

##### PRESIDENT AND CEO

Steve Weitzner

##### EXECUTIVE VP AND CFO

Adam Marder

##### SENIOR VP, AUDIENCE MARKETING & DEVELOPMENT

Bill Amstutz

##### SENIOR VP, CMP INTEGRATED MARKETING SOLUTIONS

Joseph Brauer

##### SENIOR VP AND GENERAL COUNSEL

Sandra Grayson

##### SENIOR VP, CORPORATE MARKETING

Lisa Johnson

##### SENIOR VP, CORPORATE SALES

Anne Marie Miller

##### SENIOR VP, MANUFACTURING

Marie Myers

##### SENIOR VP, COMMUNICATIONS

Alexandra Raine

##### VP, AUDIENCE DEVELOPMENT

Michael Zane

##### PRESIDENT, CHANNEL GROUP

Robert Faletra

##### PRESIDENT, CMP ENTERTAINMENT MEDIA

Tony Keefe

##### PRESIDENT, BUSINESS TECHNOLOGY GROUP

Jeff Patterson

##### SENIOR VP, GROUP DIRECTOR, ELECTRONICS & SOFTWARE GROUPS

Paul Miller

##### SENIOR VP, GROUP DIRECTOR, COMMUNICATIONS GROUP,

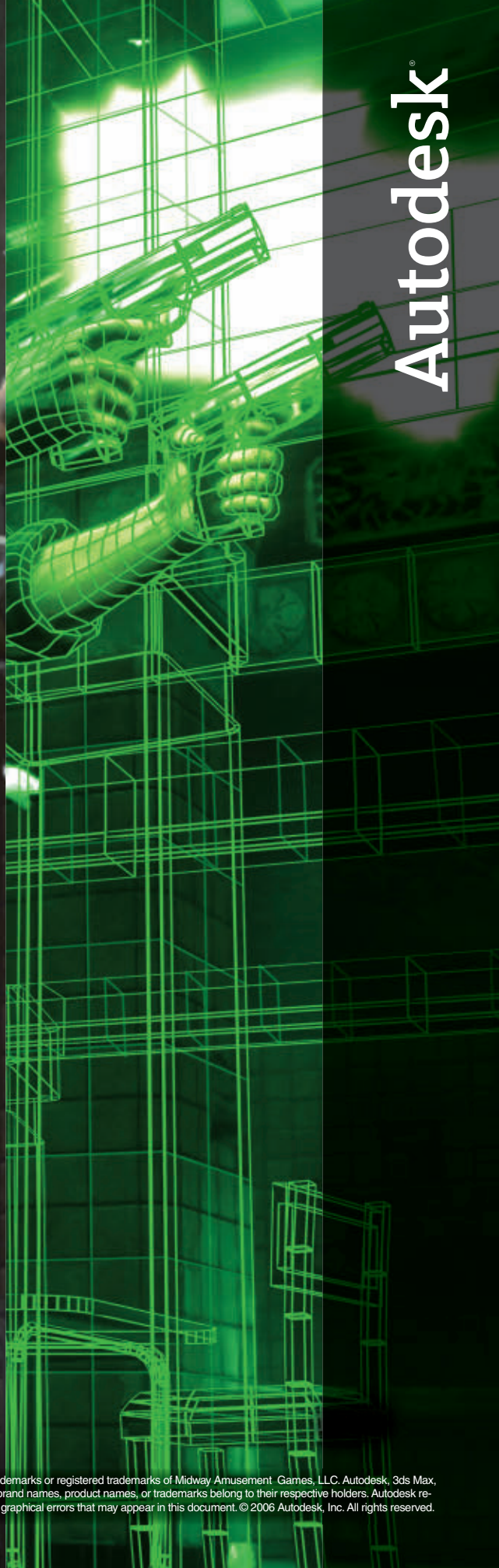
Stephen Saunders



Game Developer is BPA approved






United Business Media



Autodesk®

**Idea:**  
Assemble the ultimate team to create a John Woo sequel that you can play.

**Realized:**  
The line between video game and film is officially blurred. When faced with the challenge of creating a video game sequel worthy of John Woo's movie "Hard Boiled," the team at Midway chose to use Autodesk® 3ds Max®, Autodesk Maya® and Autodesk MotionBuilder™. Using this 3D arsenal, artists are able to create graphically stunning characters, backgrounds and animations with amazing production efficiency. Get the full story on this next-generation masterpiece at [autodesk.com/stranglehold](http://autodesk.com/stranglehold)

AUTODESK® 3DS MAX®  AUTODESK® MAYA®  AUTODESK® MOTIONBUILDER™ 



Stranglehold © 2006 Midway Amusement Games, LLC. All rights reserved. Stranglehold, MIDWAY, and the Midway logos are trademarks or registered trademarks of Midway Amusement Games, LLC. Autodesk, 3ds Max, Maya and MotionBuilder are registered trademarks or trademarks of Autodesk, Inc., in the USA and/or other countries. All other brand names, product names, or trademarks belong to their respective holders. Autodesk reserves the right to alter product offerings and specifications at any time without notice, and is not responsible for typographical or graphical errors that may appear in this document. © 2006 Autodesk, Inc. All rights reserved.



## NEXT-GEN GOLD

**NEXT-GEN GAMES COULD SAVE US** all. Updated financial results have been released by every major game publisher, and despite whether each company's profits were up or down, the consensus was the same for next-gen development.

Before the release of the Xbox 360, publishers were understandably wary of the high costs, often upwards of \$15 million, of next-gen development, and projected that next-gen would begin to pay dividends in 2007. But looking at results for the quarter ended September 30, next-gen dollars are already making their way into publishers' bank accounts.

REVENUE (IN MILLIONS)		
COMPANY	2006	2005
Activision	\$188.2	\$222.5
Electronic Arts	\$784	\$675
Midway	\$27.4	\$29.1
THQ	\$240.2	\$142.7

For the period ending September 30, 2006

Activision revealed revenues of \$188.2 million, which was down from \$222.5 million in the same period last year, but was still better than the company expected, something that CEO Robert Kotick attributes to "improving market conditions," as stated in an investor earnings call. The better quarter was reportedly due in part to CALL OF DUTY 2 on the Xbox 360 crossing the million mark in sales, as well as continued sales of GUITAR HERO, in anticipation of the sequel.

Looking forward, Activision expects its third quarter holiday season revenues to reach \$600 million, with a year-end outlook of \$1.15 billion—ambitious, perhaps. But the company has three launch window titles for the PlayStation 3 and five for the Wii in addition to its Xbox 360 offerings, indicating major next-gen support for all consoles. Interestingly, micro-transactions and digital downloads amounted to \$2 million for Activision, primarily

due to downloads of content in CALL OF DUTY 2.

Electronic Arts saw revenue trending up for its second quarter, with \$784 million, up from \$675 million last year, helped by its lucrative sports titles, such as MADDEN NFL 07. While overall sales for EA games were up, profits weighed in at \$22 million compared to last year's \$51 million, which the company attributes to including employee stock option expenses in its results.

EA specifically called out next-gen as boosting the company this quarter, with \$166 million in revenues on the Xbox 360 alone, making up for a 25 percent decline in current-gen software sales. The company predicts year-end revenues of somewhere between \$2.95 billion and \$3.125 billion, which is up from previous expectations.

EA's digital transactions for the quarter, including the company's casual portal Pogo.com, micro-transactions, and Xbox Live Arcade, took a sharp uptrend to \$28 million, 40 percent higher than the previous year.

Midway announced continued losses for this quarter, with \$27.4 million in revenues, somewhat down from \$29.5 million the year prior. The company is putting all its eggs in the next-gen basket, with high profile titles like STRANGLEHOLD and Midway's next BLITZ title expected to bolster the company.

While revenues were somewhat worse, the company did post improved losses of \$22.2 million for the period, instead of \$29.1 million last year. Official estimates for the holiday period are modest at \$86 million with an expected profit of around \$2 million.

Overall, Midway expects net revenues of \$155 million, but with a loss of \$73 million "primarily due to decreasing margins on current-generation titles," according to CEO David Zucker. Midway's next-gen lineup is a sink or swim proposition for the company, so 2007 will be an interesting time to track its progress.

THQ also released its results, up strong with \$240.2 million for the quarter, compared to \$142.7 million

## THE ROAD TO PURPOSE

SERIOUS GAMES INDUSTRY STILL IDENTIFYING CHALLENGES

### WHAT PRECISELY CAN VIDEO GAMES DO?

That's the question some game-makers have been asking themselves since before the term "serious games" was coined to encompass all kinds of games used for non-entertainment purposes. And they're still asking it today.

Serious games, in general, have been touted for their good-deed doing in major media outlets. After all, what self-respecting news reporter could resist a story about a video game that helps cancer patients? But being only a few years into a quasi-established scene, the serious

games industry is still nebulous, and defining its problems and challenges is half the battle. The players in the space continue to ask very formative questions, such as, "How do we know whether the game accomplishes its goals?" or "Are the game's goals effective?"

"There are myths about games for training," says Mike Zyda, director of the University of Southern California Viterbi School of Engineering's GamePipe Laboratory. Zyda, who spoke at the third annual Serious Games Summit D.C. in late October, believes these myths pose some stubborn stumbling points for developers. "Games clearly train people to play games," says Zyda. "[Developers] have to ask whether they will provide training games,

and not 'games.'"

While console game developers have places, either online or in the world, where they can convene and swap ideas, serious game creators recently have been expressing a strong desire for more centralized information junctions. "There's a huge desire to build archival literature in the serious games space," Zyda says, adding that games are "too expensive to build" to not have it.

The problem is not that rich information doesn't exist—it's just difficult to track down, according to Bob Bates, an independent game designer, writer, and producer, who also currently serves as a board member of the IGDA.

"Hundreds of studies have documented the effect of games,"

MIKE ZYDA (L) AND BOB BATES (R)





in the same quarter of 2005. The company landed a profit of \$12.5 million, well up from last year's loss of \$1.4 million, thanks to heavy sales of next-gen title *SAINTS ROW*, *CARS* licensed games across multiple platforms, and the success of *COMPANY OF HEROES*. THQ

but these are scattered, Bates says. He questions whether those studies actually prove anything or if they are in disagreement or could be made invalid once they are compiled into a literature review. "Studies have found that games are effective in accomplishing different goals," but the researchers of these different reports "are motivated differently" because they are affiliated with different industries, such as gamemaking, research, or education.

Alicia Sanchez, a project scientist who studies synthetic learning, says much of the research done in the sector is too specific. "There is research being done in the serious games area, but most of it focuses on the effectiveness of particular games. What I'd like to see is more research that can be generalized to all serious games, contributing to a science of learning involving serious games," Sanchez says.

is also the most prolific third-party publisher for the Nintendo DS.

THQ's future forecast sets a range of \$925 million to \$975 million for 2007, banking on—what else?—the sale of next-gen software.

—Brandon Sheffield

"Debates still exist on whether or not serious games need to be fun to be able to teach," she adds. "Without understanding what makes serious games a viable and effective learning tool as a whole, we can't begin to understand what characteristics of games enhance learning."

Although Bates is concerned about research, he identifies another pressing need that's related to the issue of not having a central junction for gamemakers. "Probably the biggest thing holding back serious games is we don't have a marketplace" where the different players can connect, he says.

Brian Williams, a research staff member at the Institute for Defense Analyses, agrees. "There is a market ... but there isn't a viable marketplace," he says, an area where "buyers can meet sellers or problems meet solutions."

—Jill Duffy

THANK HEAVENS FOR OUR FAVORITE SOFTWARE APPS, GADGETS, and shelf-worn books. Let's all tip our hats and give thanks to the products that stand behind today's games.

*Game Developer* is pleased to announce the 2006 Front Line Award finalists. The Front Line Awards are given annually to the top products that help game developers do their jobs better and more efficiently. Winners—and one new inductee to the Hall of Fame—will be announced in the January 2007 issue of *Game Developer*. We salute the top five finalists in each category!

#### ART

Modo 202, Luxology  
Maya 8, Autodesk  
Corel Painter IX.5, Corel  
Softimage Face Robot,  
Softimage  
MotionBuilder 7.5, Autodesk

#### AUDIO

Wwise 2006.2.1, Audiokinetic  
Miles Sound System v7.q, RAD  
Game Tools  
CRI ADX, CRI Middleware  
Vivox Precision Studio and the  
Vivox Network managed  
service, Vivox  
ISACT, Creative Labs

#### MIDDLEWARE

AI.implant, Engenuity  
Technologies Inc.  
SpeedTree RT 4.0, IDV  
Havok FX 4.0, Havok  
Euphoria, NaturalMotion  
Kynapse SDK 4, Kynogon

#### PROGRAMMING

XNA Game Studio Express,  
Microsoft  
Perforce SCM 2006.1, Perforce  
Software  
NVPerfKit2, Nvidia  
IBM Rational Purify Plus v7.0,  
IBM  
DevTrack 6.0, TechExcel

#### HARDWARE

FireGL V7300, ATI  
GEForce 7950GX2, Nvidia  
PhysX, Ageia and BFG  
Technologies  
Alienware Mj12 8550i  
workstation, Alienware/Dell  
Xbox 360 XMA Audio  
Decompression Hardware,  
Microsoft

#### BOOKS

*Better Game Characters By  
Design*, Katherine Isbister,  
Morgan Kaufmann  
*3D Game Textures:  
Create Professional Game Art  
Using Photoshop*,  
Luke Ahearn, Focal Press  
*ShaderX4*, Wolfgang Engel  
(ed.), Charles River Media  
*Game Writing: Narrative Skills  
for Videogames*,  
Chris Bateman,  
Charles River Media  
*Half-Real: Video Games  
between Real Rules and  
Fictional Worlds*, by  
Jesper Juul, The MIT Press

#### ENGINES

Torque Game Builder 1.1.1,  
Garage Games  
Valve Source Engine, Valve  
Unreal Engine 3, Epic  
HeroEngine, Simutronics  
Corporation  
Gamebryo 2.2, Emergent  
Technologies

## CALENDAR

### Asia Game Developer Summit

Orchard Hotel Singapore  
Singapore  
December 9 and 10  
Price: SGD 130–200  
[www.agdsummit.com](http://www.agdsummit.com)

### International CES

Sands Expo and Convention Center  
and Las Vegas Convention Center  
Las Vegas  
January 8–11, 2007  
Price: \$179–\$999  
[www.cesweb.org](http://www.cesweb.org)



# Are Your Characters Developing a 'Tude?

- Author Behaviors Interactively
- Plug in custom procedural nodes
- Leverage the full power of Havok Physics and Animation



Havok Behavior Gives You  
Complete Character Control







EXCEPTIONAL



GREAT



FAIR



POOR



UNFORTUNATE

# DAZ 3D'S HEXAGON 2

By Tom Carroll

## DAZ 3D INC.



### STATS

DAZ 3D Inc.  
12637 S. 265 W. #300  
Draper, UT 84020  
1.800.267.5170  
[www.daz3d.com](http://www.daz3d.com)

### PRICE

\$149

### SYSTEM

#### REQUIREMENTS

Windows 2000/XP and Pentium or compatible 1.5GHz CPU; or Mac OS X 10.2 or later and Power Macintosh G4 1.0 GHz CPU. 512MB system RAM. 128MB video RAM graphic board. Recent series OpenGL graphic board with 128MB Video RAM (Nvidia FX 5000 series/Quadro, or ATI 9000 series or better). 500MB free hard drive space.

### PROS

- Very affordable.
- Intuitive to a fault and fun to use.
- Online documentation and tutorials help quickstart learning.

### CONS

- Viewport needs more convenient labels, especially for viewport views.
- While icons are creatively designed, menu system takes some getting used to.
- Lacks some functionality of bigger packages (but then there's that low, low price).

### MY WIFE IS A POTTER, SO OVER THE

years I've come to see how a potter works. She takes a big lump of clay, smacks it onto the potter's wheel, sets it spinning, and within a couple of minutes she's made a cup, a bowl, a vase—you name it. Sometimes she modifies the basic shape when the clay is leather hard by denting it or by joining two shapes together to make a new and unique piece. The process is very intuitive.

But if I asked her to extrude a face or sweep an edge, she'd look at me as if I had horns growing out of my head. Those things are not very intuitive ... unless you are a Hexagon user.

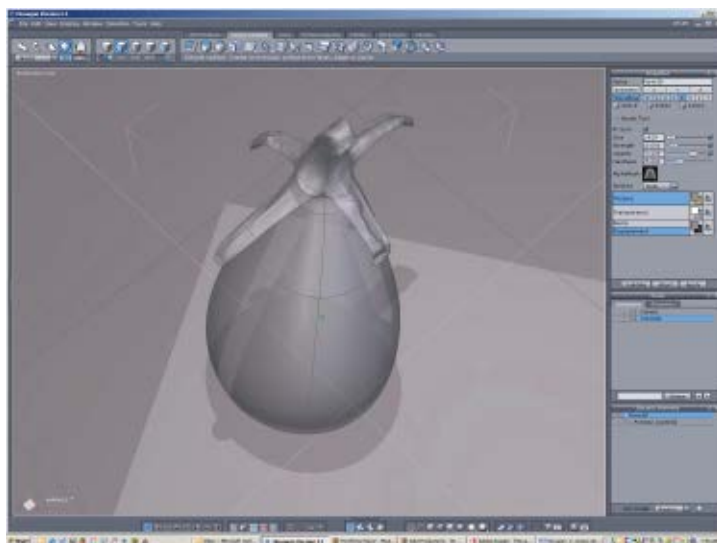
Hexagon, created by Eovis (a division of DAZ), is by far the most intuitive 3D modeling package I have encountered, and now that they've introduced version 2, they've made it even more so.

### THE QUICKER PICKER UPPER

Until Hexagon, most packages required a fairly long learning curve, aided (somewhat) by documentation or some form of "for dummies" book. I can be harsh about it since I worked my way through LightWave 3D back in the early 1990s, through the hunt-and-peck and trial-and-error methods. While it still takes a few hours, anyone with some software savvy will be able to get the hang of it fairly quickly.

The core strength embraced by the Hexagon development team is that modeling should be as intuitive as slapping around a piece of clay. Almost everything can be attacked by taking a primitive shape, such as a cube, sphere, or cylinder, smoothing it a few times to increase its polygon count, and then sweeping, extracting, or extruding various surfaces.

In the previous versions of Hexagon, the problem used to be that once the simple model was completed, you needed another program to texture, light, and render it. But all that has changed with version 2.



Extruding surfaces in Hexagon 2 is as easy as sculpting clay.

### WHAT'S IN THE BOX TOTALLY ROCKS

What's that rental car rivalry slogan?

The number two company "works harder to give you what you want?" Hexagon 2 is kind of like that.

For starters, Hexagon 2 includes a set of brush modeling tools that makes detail modeling more like editing an image (think Zbrush lite). This really helps when painting details onto displacement and bump maps.

The package's texturing capabilities are also greatly enhanced. For example, the new UV-unfold tool lets you UV map coordinates on a complex model surface. The UV-mapping module also combines a 2D view of the UV space, which is very handy. Hexagon 2 also handles transparency much better now and allows users to apply predefined textures from Genetica 2.5.

Last but certainly not least, Hexagon 2's new OpenGL engine is an ambient occlusion powerhouse. "What's ambient occlusion?" you ask. It's the amount of light or shadow a given point is likely to receive. (For a primer, see "Ambient Occlusion and You," in the August 2006 issue of *Game Developer*.)

Ambient occlusion helps artists achieve a subtle but very powerful lighting effect. With Hexagon 2's ambient occlusion features, you can now get very rich shaded-looking model previews, even ones that include real-time shadowing.

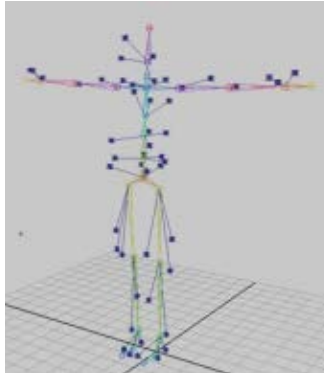
### EXCELLENT WITH AUBERGINES

After watching some of the built-in help videos, I was up and running within a few short minutes. My experience was that Hexagon 2 is just as intuitive as I was led to believe.

Because I'm working on a project that involves a character called The Eggplant King, I decided to model a nice organic asset (see the image). Starting with a cube, I smoothed it down to a tessellated sphere. Using Select Faces, I quickly squashed the sphere and extruded a neck going upward. I then swept the upper edges into thin leaves, and then extruded the star-pointed center face into the upper stem. I made one basic eggplant in about three minutes. Not bad.

I then found the displacement brush section to give the leaves a little bit of a variegated look and it was just like using a basic Zbrush tool or the

PeelSolve, a new plug-in for Maya (in beta), solves for motion-capture data.



Displacement function in Maya—simple, direct, easy. The hardest part was finding it in the menus, but menu design is always the bane of any 3D package. [Anyone tasked with trying to make such a maze into anything other than a laborious labyrinthine ordeal should be given special dispensation from the Queen, Bill Gates, or someone of equal rank.]

Finally, I wanted to shade my eggplant, check out the ambient occlusion, and give my creation shadows. All this was as simple as the Displacement tool.

Texturing my eggplant was just as simple, too, as Hexagon 2 offers a variety of UV projection tools and UV Unwrap. Overall, my eggplant experience convinced me that I would enjoy using Hexagon to create more complex assets given time.

#### EASY ON THE POCKETBOOK

Hexagon 2 is not Maya or Max, but it doesn't pretend to be. And frankly, I'd be completely shocked and awed if it tried to be. But it is extremely good for a \$149 package. Hexagon compares amazingly well when stacked against the big players' price tags: between \$2,000 and \$3,500.

For anyone who wants to get into 3D design or asset creation in their home or small office studio, Hexagon 2 fills the bill in nearly every way. Now, because I believe in putting my money where my mouth is, I'm off to figure out a way to get my wife to try it.

**TOM CARROLL** is a video game artist and freelance writer who strives to understand only enough of his corner of the universe to be able to sleep at night. Email him at [tcarroll@gdmag.com](mailto:tcarroll@gdmag.com).

## PREVIEW: ALASTAIR MACLEOD'S PEELSOLVE

By John Root

### OPTICAL MOTION CAPTURE IS A PRETTY

small niche of computer graphics. If you want to use it, you don't have a lot of solver options—good ones anyway. But there's a new option that's inching its way to commercial release, and it's the best option I've seen to date. It's called PeelSolve, a plug-in for Maya, and as of this writing, it's still in beta.

Written by motion-capture veteran Alastair Macleod of *Matrix*, *Catwoman*, and *The Lord of the Rings* fame, with the enlisted help of math Guru Lucio Moser, PeelSolve is a global optimization (GO) solver that works directly inside Autodesk's Maya. GO solvers are a fairly new technique for solving motion capture, and though it's not exclusive to PeelSolve, it has been made far more accessible due to its integration with Maya.

### THE SOLVING PROBLEM

Solving is the process by which motion-capture data is turned into usable animation, typically skeletal animation. The problem with solving is that in trying to measure a joint's translation and rotation from a non-rigid marker's translation (marker on the model), the joint's exact values cannot be known. Furthermore, even if we could know it, it's not necessarily the desired value for our game character whose skeleton is likely rigid and nothing like the actor's skeleton.

GO is a type of constraint that works on hierarchically connected objects simultaneously. In other words, it will pose your skeleton into your motion

capture markers. The optimization bit is the important part. A GO solver attempts to maintain the desired distance between a marker and a joint by error distribution. The entire skeleton will push and pull until all joints are within their specified error tolerance.

With PeelSolve, you pose your skeleton into your markers, assign the markers to the joints, and let the program animate the skeleton such that its joints are always the same distance from the markers as they were when you assigned them.

During this process you can: adjust or animate per marker weights; adjust or animate per joint stiffness; allow or disallow per translation; and set degrees of freedom per joints.

And because it's all happening inside Maya, everything shows up as animatable attributes right in your channel box.

### SOLVE FOR X, X=DATA

A bit of setup is required: Skeletons need to have their preferred angles defined and their degrees of freedom set. There's also a few limitations in the software that require some workarounds (unless you work exactly the same way the authors of PeelSolve do), but it's not too bad, and it's certainly far less than what you'd be expected to do in other packages.

Trial software is available in beta—but the tool is in a very useable state. Now is a great time to get involved and help guide PeelSolve's direction.

I highly recommend the software for anyone who has had to solve motion-capture data and is wondering how those film people get theirs to look so good.

**JOHN ROOT** is lead animator at *id Software*. Email him at [jroot@gdmag.com](mailto:jroot@gdmag.com).



### BETA PREVIEW

#### Alastair Macleod's PeelSolve

motion capture solver plug-in for Maya  
[www.mocap.ca](http://www.mocap.ca)

### PRICE

Free trial in beta

### SYSTEM REQUIREMENTS

Windows  
Autodesk Maya 7 or higher

## BETA PREVIEW

**Nevercenter's Silo 2**  
enhanced modeling program  
[www.nevercenter.com](http://www.nevercenter.com)

### PRICE

\$109 for version 1.4,  
with free beta testing of  
version 2

### SYSTEM REQUIREMENTS

- Windows 2000 or better (for PC) or Mac OS X 10.3 or better
- 400MHz CPU (1GHz recommended)
- 128MB RAM (512MB recommended)
- Graphics card supporting 1,024x768 16-bit color and OpenGL (24- or 32-bit color recommended)
- 10Mb hard drive space
- Two-button mouse (three-button mouse or two-button mouse with scroll wheel recommended)

## PREVIEW: NEVERCENTER'S SILO 2

By Ronnie Ashlock

### A SHORT TIME AGO, NEVERCENTER

released a version 2 beta of its sole product, Silo, to registered users. Anyone with a license for the first version was eligible to kick the tires and, more importantly, suggest improvements to the new version before it would be released as a final product.

The first version of Silo was a pure modeling package; if you needed to UV or rig and animate a model, you had to look elsewhere. The focus was on modeling, and Silo, with its simple yet deep control scheme, offered enough innovation and "why didn't the other guys think of this?" commonsensical tools to make it an invaluable addition.

### VERSION UP

In this version 2 beta release, Silo has expanded beyond pure modeling.

Nevercenter is still stress testing and weighing feedback on some new features though, so certain tools, such as normal mapping and displacement map generation, are currently disabled (though the company is clear about its intentions to include them soon).

Knowing that the program isn't done and still has unpredictable moments of instability, is it worth the occasional frustrating crash to get onboard the beta train? Absolutely. Silo 2 beta is a little workhorse at accomplishing a whole host of tasks more quickly and intuitively than other mainstay applications. Simply put, Silo 2, even in beta form, is a joy to use.

Version 1 achieved a loyal user base primarily for its simple interface, elegant tools, and low price. Silo fits in nicely with any major pipeline and has proved to be a handy tool in the artist's toolbox.

Even though it's a beta release, the program feels very polished: it's easy

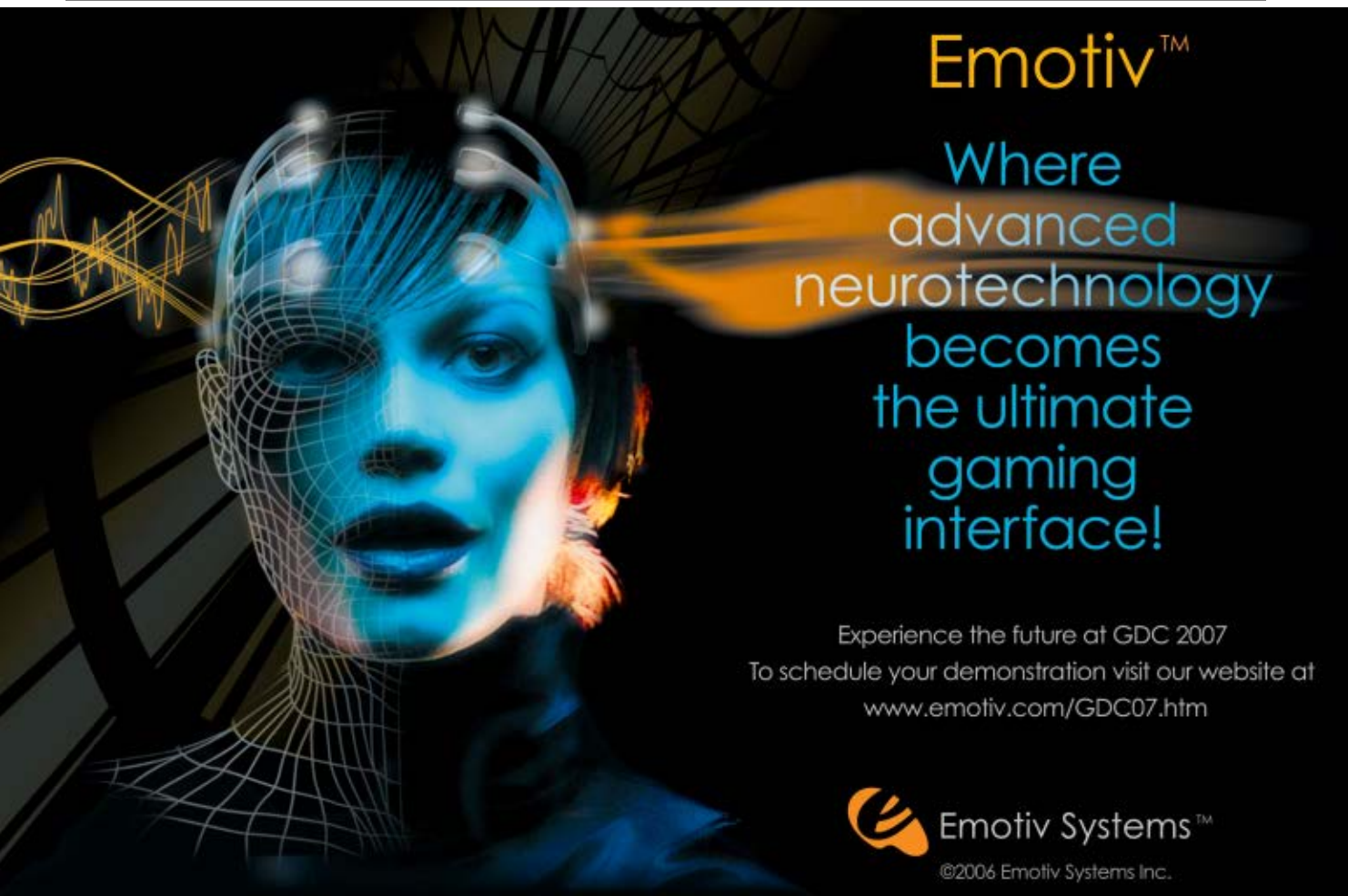
to forget the program is not yet finished, a point the Nevercenter developers repeat, with good reason, on the forums.

### WHAT'S NEW IN TWO?

Probably the most exciting new feature is the ability to sculpt or paint displacement on a subdivision surface model, much like one would do in Pixologic's Zbrush 2. What differentiates Silo 2 beta is the ability to alter surface topology using the original polygon tools as well.

Users can quickly insert, slide, bevel, or otherwise remove edge loops, push and pull vertices, or extrude faces on a model and, with a tap of the T key, begin sculpting high-frequency displacement detail or refine changes with the traditional polygon toolset. A smart new Brush Editor consists of push, smooth, move, flatten, smudge, pinch, and sharp push along with a


CONTINUED ON PG 10



# Emotiv™

Where  
advanced  
neurotechnology  
becomes  
the ultimate  
gaming  
interface!

Experience the future at GDC 2007  
To schedule your demonstration visit our website at  
[www.emotiv.com/GDC07.htm](http://www.emotiv.com/GDC07.htm)

 Emotiv Systems™  
©2006 Emotiv Systems Inc.



CONTINUED FROM PG 9

falloff curve tool and the ability to load alpha images to use as brush shapes. Feedback on higher subdivision levels has been noticeably sped up from the first beta release.

Although the high-frequency sculpts look great, it's questionable whether they will work as game assets. This is a common problem with new sculpting solutions.

Silo 2 provides the new Surface Snapping option to greatly speed up the building of new smart topology using a high-frequency reference model. Surface Snapping allows an artist to trace (for lack of a better word) new geometry on top of an existing model or group of models. What's great about this feature is that all the polygon tools work in conjunction with the snapping constraints, including soft selection and the slide tool. Very subtle and accurate adjustments are possible using the soft selection option, as an artist shifts groups of vertices, edges,

or faces of a lower-resolution mesh while it adheres quite nicely to the higher-resolution sculpt underneath.

New UV tools were added as well. The modeling window can be used to quickly select groups of faces or edges and then quickly convert these selections to flattened UV chunks by generating UV proxies. These proxies can then be unwrapped with a single click and then re-arranged on a 2D texture grid or moved around in a 3D view. The workflow is simple and fast and easy to learn.

Continuing with Nevercenter's minimalist aesthetic, the Silo 2 beta user interface is not buried in mountains of windows, rollout bars, and attributes fields (although users can quickly clutter the screen with different editors should they choose).

Users can modify their interface with their own custom button setup to maximize workflow. The Scene Editor is valuable for managing different models in the scene, but also shows groups and

UV proxies and provides toggles to lock any combination of models as desired. Shading modes are also accessible here and artists can have any variation of shading on as many models as they would like.

With the already available new sculpting and UV tools and the coming ability to generate normal and displacement maps, and an enhanced (but thankfully simple) user interface, Silo users are anxiously awaiting the final release. Game artists not currently on the Silo bandwagon should definitely give it a look soon, as the future looks bright for this can-do 3D program. ✖

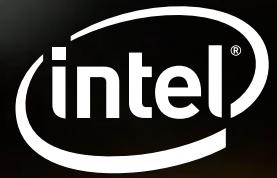
**RONNIE ASHLOCK** *has more than five years experience making video games. He currently works for Sony Online Entertainment's Seattle Studio, working on an unannounced next-generation game. Email him at [rashlock@gdmag.com](mailto:rashlock@gdmag.com).*

**“SketchUp lets me construct and test 3D spaces faster and easier than any other tool I know of.”**

– Jason VandenBerghe, Lead Game Designer  
Z-Axis, a studio of Activision

Get your games out faster with SketchUp. Visit [www.sketchup.com](http://www.sketchup.com) to download a free trial of SketchUp Pro 5.

**SketchUp**  
from Google®



# Translate Multi-Core Power Into Application Performance.

Get your applications ready for scalable, parallel processing. Intel® Software Development Products help C++ and Fortran developers create, debug and optimize threaded applications.

## Intel® Threading Building Blocks<sup>1</sup>

Introduce scalable threading through C++ algorithms.

## Intel® Thread Profiler

Pinpoint bottlenecks and maximize threading performance.

## Intel® Thread Checker

Identify latent data races and deadlocks with a patented error detection engine.

## Intel C++ Compilers

Utilize highly optimized threading capabilities such as OpenMP\* and auto-parallelism.

## VTune™ Analyzers

Identify performance bottlenecks in multi-core sharing of the bus and cache.

**“ We are optimizing RenderMan’s core to be very scalable for future multi-core architectures. Intel’s Threading Tools have accelerated our development cycle dramatically. ”**

Dana Batali  
Director of RenderMan Development  
Pixar

**Put our products to the test!** Download evaluation copies of the Intel® Software Development Products today at [www.intel.com/software/products](http://www.intel.com/software/products)

Find a reseller at [www.intel.com/software/products/resellers](http://www.intel.com/software/products/resellers)



© 2006 Intel Corporation. Intel, the Intel logo, Intel Core, Itanium, Intel Xeon and VTune are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

<sup>1</sup>Intel® Threading Building Blocks available in C++ only.

\*Other names and brands may be claimed as the property of others.

# TRANSFORM YOUR GAME FROM THE INSIDE OUT

**BUILD, TEST AND MANAGE YOUR GAMES IN A WHOLE NEW WAY.** Introducing Emergent Elements,<sup>™</sup> a powerful set of essential game development tools. Modular yet fully in sync, Elements integrates stand-alone metrics, automation and 3D graphics technologies into a flexible framework—all backed by the best service in the industry. Now you'll spend less time on what you *have* to do and more on what you *want* to do.

For PLAYSTATION<sup>®</sup>3, Xbox 360<sup>™</sup> and PC development.

Transform your game. It all starts now.

**Learn more at [www.emergent.net](http://www.emergent.net) or at GDC Booth 1428.**



**EMERGENT  
ELEMENTS<sup>™</sup>**



Emergent Elements<sup>™</sup> is a trademark of Emergent Game Technologies.  
©2006 Emergent Game Technologies. All rights reserved.  
All trademarks herein are properties of their respective owners.





## STATE OF THE INDUSTRY: VIDEO GAME PIRACY

---

PAUL HYMAN was editor-in-chief of CMP Media's GamePower and currently writes a weekly column on the game industry for The Hollywood Reporter. He's covered gaming for more than a dozen years. Email him at [phyman@gdmag.com](mailto:phyman@gdmag.com).

**GAME PIRACY IS AN EVER-GROWING MENACE, OR SO SAY THE EXPERTS WHOSE JOB IT IS TO BE** aware of such things. The rest of us may wonder what's really at stake or what the costs of piracy are. It's no simple matter to determine how much the industry loses from sales it might or might not have made—and there's the rub. Does anyone really know to what extent game publishers and developers are being plundered compared with, say, last year or the year before? It doesn't seem so.

In 2004, worldwide piracy was estimated to have cost the U.S. entertainment software industry more than \$3 billion, according to the Electronic Software Association (ESA), but that didn't include losses attributable to internet piracy.

Last year in an interview Ric Hirsch, the ESA's senior vice president for intellectual property enforcement since 2000, admitted, "we just don't have a good sense as to how much additional



# VIDEO GAME PIRACY



Todd Hollenshead, CEO of id Software.

money online piracy steals; I've seen estimates as low as \$1 billion annually and as high as \$3 billion. The problem is that measuring activity on the internet—legitimate or otherwise—is much more difficult than measuring the loss due to hard goods piracy. When a hacker cracks a game's protection code and puts the game on the web, it's impossible to determine how many downloads will result."

## ONLINE ACQUITTANCE

This year, those still-unquantifiable illegal downloads seem to have become the predominant liability, growing at a rate that has overtaken illegal hard goods sales, such as counterfeit copies of games one

might find for sale on street corners or at flea markets.

"Across the world, the digital environment is evolving and becoming, to some extent, more conducive to the unauthorized use of game files," Hirsch said in a recent interview. "We know this because increased broadband connections to the internet are increasing opportunities for the downloading and uploading of digital files and games. Meanwhile, we are starting to see diminished amounts of pirated hard goods. It hasn't disappeared by any stretch; in certain countries, it's still the prevalent form of piracy. But, in a lot of places, online piracy is supplanting hard goods piracy."

To estimate the amount of damage caused by online piracy, the ESA uses online monitoring services, which create snapshots for the organization to analyze. Hirsch says he has seen an increase in activity, particularly on peer-to-peer networks, "but it's difficult to quantify that since it's uncertain how much of the internet we're able to see. There's a lot that goes on within certain closed networks that aren't entirely open to people on the internet. We have very little idea of what goes on within those networks."

## PC AGONY AT ID SOFTWARE

Although hard and quantifiable data about game piracy is in short supply, anecdotal information isn't. At Mesquite, Tex.-based id Software, CEO Todd Hollenshead calls the number of PC games being pirated "devastating" and says game pirates are destroying the PC entertainment sector.

"We're talking about a \$10 billion industry that is losing \$2 billion to \$3 billion to piracy, according to the latest ESA statistics I saw," says Hollenshead. "I think you could call a 20 to 30 percent loss 'devastating.' In fact, I believe that the only games where piracy isn't an extreme problem are the pure subscription games, like WORLD OF WARCRAFT."

While the impact on independent game developers is difficult to determine, Hollenshead has no doubt that "if every one of the games that people actually played were purchased, some companies that had very popular titles—like Looking Glass Studios [best known for ULTIMA UNDERWORLD, SYSTEM SHOCK, and THIEF: THE DARK PROJECT]—would still be in business today. I'm not saying that piracy caused Looking Glass to go out of

business, but if all their games had been paid for, they would have had the financial wherewithal to stay in business."

Because his company is PC-centric, Hollenshead believes that id Software is affected more by piracy than developers that are console-centric.

"PC games are just easier to pirate. That's because you don't have to modify the hardware to crack the software," he explains. "Once the hacker gets the game and breaks through the copy protection, all he has to do is put it on the internet for someone to download. The only requirement for that is having the time and the bandwidth."

Hollenshead recounts that piracy has always been a problem for id, but "it has gotten so bad that it's hard to justify putting a triple-A title out on the PC. We wind up saying, 'Okay, we really love working on the PC, but if we're absolutely going to be robbed by the pirate community, doesn't it make sense to focus on consoles instead and have the PC become a secondary SKU?'"

In 1996, when id Software released QUAKE, few gamers were on the internet and so there was a greater barrier to piracy. Nevertheless, Hollenshead estimates that half the versions being played were counterfeit, which means that the \$18 million in gross revenue that the game generated should have actually been about \$36 million.

"While [id co-founder] John Carmack is obviously very pro-PC—which is one reason why our games have always come out on PC first—nevertheless, I anticipate that our next game will either come out on consoles first or at least on consoles concurrently with the PC," Hollenshead says. "The current piracy situation doesn't leave us much choice."

## CONSOLE CRAMP

Console games may be more difficult to copy than PC games, but that has enticed pirates to be even more determined to crack the uncrackable.

"It's amazing how quickly we're starting to see pirated versions of games for the Xbox 360, for example," says Loren Hillberg, "sometimes day-and-date with the release of the game." Hillberg is executive vice president and general manager of the Commerce Business Group of Santa Clara, CA-based Macrovision, which manufactures software copy protection products.

But if developers are using copy protection schemes, like Macrovision's, why has piracy, in Hillberg's words, "not declined in any meaningful way over the past year or so"?

The answer, he says, is twofold: First, some developers choose to go without any sort of protection. "Second—and more importantly—the piracy issue has become sort of an arms race between those who build protection and those who choose to entertain themselves by breaking protection," he says.

"The bad guys are out to gain a reputation in their community for being able to do the most sophisticated hacks. The result is that there's an ongoing escalation. We're trying to come up with creative protection schemes faster than they figure out how to crack them."

Whether the pirates are in it for the fun, the glory, or the cash, Hirsch has one label for them all: warez groups.

"Most of them operate through the internet," Hirsch says. "Members of the same group may be in different countries and may never have met each other. But they work together to acquire and crack games as soon after their release as possible. Some of the cracks end up circulating on peer-to-peer networks; others are downloaded, burned onto discs, and used as masters



## Made with Unreal® Engine 3

Official Xbox Magazine **10/10**      1Up.com **10/10**

GameSpy **5/5**      Game Informer **9.5/10**

GameSpot **9.6/10**      Xplay **5/5**

# What could your studio do with this technology?



For engine licensing inquiries please  
email: [licensing@epicgames.com](mailto:licensing@epicgames.com)

For Epic job information visit: [www.epicgames.com/epic\\_jobs.html](http://www.epicgames.com/epic_jobs.html)

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







# VIDEO GAME PIRACY

for large-scale replication in Southeast Asia and Russia. It doesn't really matter what their motivation is. The harm they're causing is the same."

## REALITY HURTS

The problem of piracy may not be quantifiable, but attorney Mark Litvack knows it's on the rise. This year he has more clients than last year, when he had more clients than the year before.

Formerly vice president and director of legal affairs for worldwide anti-piracy for the Motion Picture Association of America, Litvack is currently a partner with the Los Angeles-based law firm Manatt Phelps & Phillips, which specializes in intellectual property matters and piracy in the video games industry. Publishers come to Litvack when they suspect they are being pirated.

"It's a challenge for [publishers] to know how badly they're being hurt," he says. "Mostly, they use search engines and such to determine how many of their files are on the web and from there they estimate how often they are being downloaded."

Can a publisher be certain that the downloaders would have paid for the games if they weren't available for free?

"We call it the Rolex scenario," Litvack says. "We all know it's easy to buy pirated Rolexes. But when you do that, is Rolex really being hurt? I mean, chances are you couldn't afford to buy a real Rolex, right? But the answer is yes, every time someone buys a \$10 Rolex, the brand name is cheapened. There's a certain panache to owning a Rolex and, if everybody owns one, anyone who wanted one because it used to be a status symbol won't buy one now. Similarly, would everyone who steals a video game have gone out and paid for that game? No, but it might dissuade them from buying any games in the future if they know that eventually all games will be available for free in pirated form."

Since video games tend to have a short shelf life, Litvack says his clients tend to be less concerned about their games being hacked post-release than they are pre-release, either from the designers' computers or from the duplicators who manufacture the discs.

"When a publisher tries to get as broad a worldwide release as possible, often they'll go to a duplicator overseas to replicate the game," adds Litvack. "When you do that, once you send your game overseas, you may not always have the control you'd have in this country."

Litvack has three pieces of advice for publishers. The first is to lock down all products and protect them before they are released.

"I always tell them to put a mirror up to their own controls—in their design studios, in their distribution network—and make sure they are adequate," he says. "That's the most effective thing they can do."

The second bit of advice is for publishers to focus their anti-piracy dollars in areas where they will help sales the most. If a publisher becomes aware that its games are being sold illegally, that makes it easier to shut down the person doing the selling.

"If the bad guys want your money, then they need to come up above the radar to get it," he says, "and that makes the person traceable and you can go ahead and sue them, which is a fairly effective remedy in the United States. If they're international, then you've got a more difficult issue to deal with."

Litvack admits he's not a big believer in what he calls "the whack-a-mole" approach, in which one mails out cease-and-desist letters that can cause the pirate to disappear in one place and pop up in another.



Loren Hillberg, executive vice president and general manager at Macrovision.

"When you threaten to sue ... and then you don't ... you become the boy who cried wolf," he says. "Either you do it or you don't."

The third piece of advice to publishers is to accept the fact that they aren't going to wipe out piracy everywhere by understanding the realities of the situation.

"If publishers know there are pirated copies of their game in Afghanistan, well, they aren't going to sell legitimate copies there anyway," says Litvack. "They need to set realistic targets and goals and then work to make those happen. Why do I say that? Because, in the long term, I don't think they have any other option."

## TECHNOLOGY TRIAGE

Macrovision is in business to give publishers and developers other options, of course. Where once copy protection was built merely to lock out hackers, today it is integrated into the game and can serve to frustrate buyers of counterfeits long after they've started playing.

"Hackers believe they've cracked the game and they post it on the Internet," explains Rob Ellison, senior director of product marketing at Macrovision. "But, well into the game, it stops functioning properly, which is embarrassing to the hacker whose reputation is at stake. Perhaps it's a football game in which, once the security is tripped, the ball vanishes. Or on level seven of a shooter, the key to level eight becomes unavailable. Of course the hacker can go back in and continue working on the protection, but that's not really interesting to them. It's not cool enough to be part of their entertainment."

id Software's Hollenshead prefers what he calls electronic countermeasures. "As soon as you see a warez copy of your game go up on the internet, you put up a spoof file that's exactly the same size. The intent is to get people to waste their time downloading the phony version of your game and then get frustrated that it doesn't work. They'll soon realize that if they really want to play your game, they're going to have to go out and buy it."

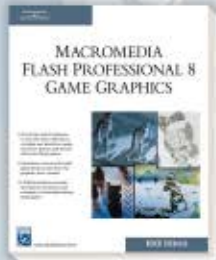
Or, he says, a publisher could post a file that's the same size as the full game but, in actuality, is only the start of the game. After several hours of gameplay, a sign pops up that says,



Mark Litvack, partner of law firm Manatt Phelps & Phillips.

# IMPERFECT GAMES

are not an option.



**Macromedia Flash Professional 8 Game Graphics**  
 1-58450-483-8 ■ \$39.95



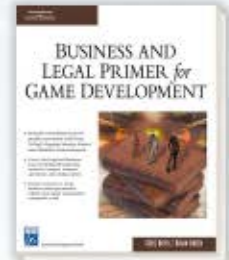
**Game Developer's Open Source Handbook**  
 1-58450-497-8 ■ \$44.95



**Programming an RTS Game with Direct3D**  
 1-58450-498-6 ■ \$49.95



**C++ for Game Programmers, Second Edition**  
 1-58450-452-8 ■ \$49.95



**Business and Legal Primer for Game Development**  
 1-58450-492-7 ■ \$49.95



**Ultimate 3D Game Engine Design & Architecture**  
 1-58450-473-0 ■ \$59.95



**Beginning Math Concepts for Game Developers**  
 1-59863-290-6 ■ \$29.99



**Creating Music and Sound for Games**  
 1-59863-301-5 ■ \$34.99



**Game Programming All in One, Third Edition**  
 1-59863-289-2 ■ \$49.99



**Game Character Animation All in One**  
 1-59863-064-4 ■ \$49.99



**3D Game Programming All in One, Second Edition**  
 1-59863-266-3 ■ \$54.99



**Beginning Game Programming, Second Edition**  
 1-59863-288-4 ■ \$29.99



**Beginning C++ through Game Programming, Second Edition**  
 1-59863-360-0 ■ \$29.99



**Unreal Tournament Game Programming for Teens**  
 1-59863-346-5 ■ \$29.99



**Sex in Video Games**  
 1-58450-459-5 ■ \$39.95



# VIDEO GAME PIRACY

"Thank you for playing the trial version. Please go to the store and purchase the full version."

"You've accomplished two things," says Hollenshead. "You've manipulated the warez community to actually become an advertising mechanism for you. Then, after someone has played the three- or four-hour file, which is actually a demo of the game, you've convinced them that it's a great game and worth buying. My guess is that they won't want to spend the time and effort downloading another so-called pirated version, only to play it for four hours and find that, once again, it's just another demo."

Will his company be using that strategy on future id Software games? "Let's just say that I'm going to lobby for it."

## BIT BY THE SCRUPLES BUG

Meanwhile, the ESA's strategy is a combination of using law enforcement to crack down on pirates and educating gamers that downloading warez files is illegal.

"It's important to punish people and send a message that there are risks to engaging in this sort of behavior," says Hirsch. "You definitely want people to know that this isn't something they can do for fun and get away with. Unfortunately, in many countries, they don't have up-to-date IP laws and the enforcement environment isn't great.

So it's difficult to get to those people.

"But here in the States, the ESA has had a very active program for almost six-and-a-half years where we offer free training on game piracy to law enforcement officials. That has fostered a certain level of awareness that is now refocusing on dealing with internet piracy, which has become more prevalent than hard goods piracy," Hirsch says.

At the same time, the ESA is working to educate young gamers about the harm that piracy causes by offering educational material to elementary schools.

"Kids need to learn what intellectual property is," notes Hirsch, "and that a

lot of the things they love to interact with—books, games, movies, and music—are protected. We certainly recognize the temptation, but they need to recognize that downloading a copy of MADDEN FOOTBALL to save 50 bucks is no different than going into a store and sticking a copy of the game under your jacket and walking out. We want them to understand the rights and wrongs as opposed to debating which is the alternative where they're less likely to get caught."

## PASSIVE PLAYERS

Ironically, the parties most damaged by pirate activity—game developers and publishers—are the ones who tend not to speak up against the practice.

"Since most developers don't have internal anti-piracy departments or even someone who's assigned to that job, they basically leave the problem to the publishers," says Hollenshead. "They believe there's nothing they can do about it other than to just try and be as secure as possible before the game is released."



In a talk at E3 2006, Hollenshead stated that every id Software title had been leaked prior to release—even WOLFENSTEIN 3D, released in 1992 before the internet was pervasive.

And the publishers, says Hirsch, believe that enforcement is the ESA's job. "We run the industry enforcement program, and they look to us to address those issues." But, in a call to action, Hollenshead believes that there is something developers and publishers can do to, if not stop piracy, slow its spread.

"The first thing we can do is recognize that anti-piracy products are barely temporary at best," says Hollenshead. "They can maybe give you a week or two on the store shelves before your software is cracked. And once it's cracked, it doesn't matter how many secure copies you have, because all it takes is one that isn't secure.

"So I'd like to see the companies that make those solutions put their heads together with the rest of the industry as a whole to invest in one that works. I'm also very hopeful that, when Microsoft releases Windows Vista, the company will take a leadership role and make some inroads against piracy."

Lastly, he supports the idea of more subscription-based games that require online authentication to begin gameplay. "If every time you boot up a game you've got to be connected to the internet, that seems to be the only way that we're going to be able to stop the warez community," he says.

## SHOVELING AGAINST THE TIDE

If piracy can be stopped, it won't be in the near-term, opines the ESA's Hirsch who believes piracy will get worse before it gets better.

"Technology always races ahead of the law," he says. "It's going to require a lot of concentrated effort on the part of the entire industry, including publishers and developers, to raise public consciousness about the harm that piracy causes. A lot of people shrug their shoulders and portray piracy as a victimless crime, but you can bleed to death from a thousand paper cuts, and there are a lot of small- to medium-sized game companies that have been hurt by piracy."

While anti-piracy techniques will continue to improve, so will the skills of the hackers, says attorney Litvack, who jokes that "the only way to stop piracy is not to distribute your product at all. Just make one copy, never show it to anybody, keep it for yourself.

"The industry is compelled to continue investing in anti-piracy techniques ... and I think it will continue to do that or piracy will win the battle. But the bottom line is this: As the flow of pirated goods grows, we will have to shovel against the tide more quickly and build bigger dams," says Litvack. "Will that push the water back upstream? No, but it'll stop the water from overflowing the dam." ❖



Rob Ellison, senior director of product marketing at Macrovision.



DAVID H. EBERLY

SERIES IN INTERACTIVE 3D TECHNOLOGY

# 3D GAME ENGINE DESIGN

A PRACTICAL APPROACH TO  
REAL-TIME COMPUTER GRAPHICS  
SECOND EDITION

BY DAVID H. EBERLY

November 2006. ISBN 978-0-12-229063-3

**LEARN ABOUT HOW WILD MAGIC WILL  
HELP YOU GET STARTED ON YOUR OWN  
ENGINE TODAY!**

SIMPLY ORDER ONLINE AT  
[WWW.MKP.COM/GRAPHICS](http://WWW.MKP.COM/GRAPHICS) OR  
BY PHONE AT **800.545.2522**

**Price: \$69.95**

*This new edition includes:*

- More than twice the coverage of the essential techniques of scene graph management
- New methods for managing memory usage in the new generation of game consoles and portable game players
- Expanded discussions of collision detection, collision avoidance, and physics
- Wild Magic, a commercial-quality game engine in source code that illustrates how to build a real-time rendering system from the lowest-level details, all the way to a working game
- Step-by-step, how to make a shader-based graphics engine and how to tame the new technology



**EBERLY**  
**HAS DONE IT AGAIN!**

**GRAB A COPY** OF THIS NEW, *REVISED*  
CLASSIC BEFORE  
YOUR NEXT PROJECT.





MARCH 5-9  
**2007**  
SAN FRANCISCO

# TAKE CONTROL

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

Register by January 31, 2007 and save up to 35%  
on conference passes!

1 to 15,000 Players

Extreme Networking

Developer Control

Content Access

GAME DEVELOPERS CONFERENCE, 600 HARRISON STREET 6TH FLOOR, SAN FRANCISCO, CA 94107. GDC IS A TRADEMARK FOR DEVELOPERS EVERYWHERE.

Licensed for play in San Francisco, March 5-9, 2007. Conference passes sold separately. Minimum system requirements: love of game industry, desire to advance the art of game creation. Go to [www.gdconf.com/gdc4u](http://www.gdconf.com/gdc4u) for proof of GDC excitement. Legalese is your friend.

WARNING - CRITICAL SESSION AND SPEAKER INFORMATION AVAILABLE ON [GDCONF.COM](http://GDCONF.COM)

MOSCONE  
CENTER

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

**game**developer

OFFICIAL GDC 2007 RATING:

INSPIRATIONAL

**I**

Old Friends  
Networking  
In-depth Classes  
Top Publishers  
Innovative Game Developers  
Independent Games  
Non-Entertainment Games

**WWW.GDCONF.COM**



# CREATING ALL HUMANS

## A DATA-DRIVEN AI FRAMEWORK FOR OPEN GAME WORLDS

**POPULATING AN ENTIRE GAME WORLD WITH CHARACTERS THAT** give an impression of life is a challenging task, and it's certainly no simpler in an open world, where gameplay is less restricted and players are free to roam and experience the world however they choose. The game engine has to be flexible enough to react and create interesting scenarios wherever the player goes. In particular, the demands on the AI are different from a linear game, requiring an approach that, while using established game AI techniques, emphasizes a different aspect of the architecture.

This article discusses the data-driven AI architecture constructed for Pandemic Studios' open world title DESTROY ALL HUMANS 2. It describes the framework that holds the data-defined behaviors that characters perform, and how those behaviors are created, pieced together, and customized.

The premise of an open world game with sandbox gameplay is to give players the freedom to do what they want, the freedom to

create their own game within the world the developers provide.

Their play is not linear, which is fantastic for a sense of immersion, but reduces the ability of the game developer to control, limit, and pre-script scenarios that the players encounter.

The AI code needs to be built on a foundation that is flexible enough to respond to any eventuality. It needs to handle a domain of gameplay that is broader in scope than a linear title and react to situations that might not have been anticipated. In effect, the AI needs to have a strong emphasis on breadth of behavior over depth. That is, the architecture must promote the ability to create large numbers of behaviors and make applying them to characters as easy as possible.

One solution to this challenge is to make the behaviors data-driven. They should be created without requiring changes to code, pieced together and reused as shared components, and substituted out for specialized versions. Ideally, the developer

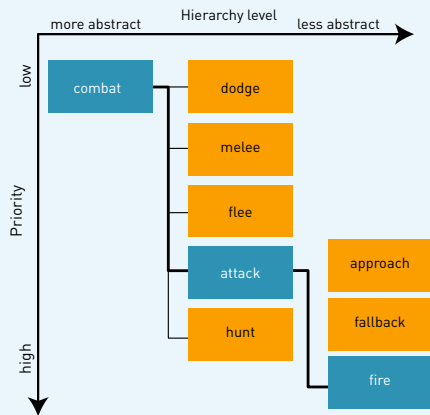
---

**JOHN KRAJEWSKI**  
is senior AI programmer for Pandemic Studios Australia. He previously worked at Surreal Software for several years before changing his world view and moving to Pandemic in Brisbane, Australia, where he most recently finished DESTROY ALL HUMANS 2. He is currently designing the AI for a new next-gen project. Email him at [jkrjewski@gdmag.com](mailto:jkrjewski@gdmag.com).





# CREATING ALL HUMANS



**FIGURE 1** A combat behavior starts prioritized children, which in turn starts more prioritized children. Pending behaviors are orange and active ones are blue.

should be able to tweak not only the settings of a behavior, such as how long a timer lasts or how aggressive an enemy is, but also the very structure of the behavior itself. For example, what steps are needed to complete a given task or define how those steps are performed? By allowing our behaviors to fit into multiple situations, be reusable, and be quick to create and customize, we can more effectively create all the actions that the characters will need to give the game life.

## BEHAVIOR FOUNDATION: PERFORMING TASKS WITH SUB-TASKS

The basis for the behavior system in DESTROY ALL HUMANS 2 is a

hierarchical finite state machine (HFSM), in which the current state of an actor is defined on multiple levels of abstraction. At each level in the hierarchy, the states will potentially use sub-level states to break their tasks into smaller problems (for example, `attackenemies` is at a high level of abstraction and uses the less-abstract `fireweapon` below it to perform part of its function). This HFSM structure is a common method used in game AI to frame a character's behaviors. It has several immediate benefits over a flat FSM. (For current information about HFSMs, see References, page 25.)

In our implementation, each state in the HFSM is called a behavior and makes up the basic architectural unit of the system. Everything that characters can do in the game is constructed by piecing together behaviors in different ways that are allowed by the HFSM. A behavior can start more behaviors beneath it that will run as children, each performing a smaller (and more concrete) part of the task of the parent. Breaking each task into smaller pieces allows us to reap a lot of mileage out of the behavior unit—reusing it in other behaviors, overriding it in special cases, dynamically changing the structure, and so forth—and spend more time making the system intuitive and easy to modify.

**Starting children.** There are many ways to break a task into smaller pieces, and the correct choice ultimately depends on the type of task. Does the task require maintaining certain requirements, performing consecutive steps, randomly performing an action from a list, or something else? In our implementation, we allow several methods of breaking down a behavior into smaller pieces by allowing different ways to start children behaviors.

**Prioritized children.** The first and most common way to start children is as a list of prioritized behaviors. Behaviors that are started as prioritized will all be constructed at once (memory is allocated for them and they are added as children of the parent) and set into a special pending mode. (See Figure 1.)

When a behavior is in pending mode, it is not updated; instead, it waits until the behavior itself decides it can activate, based on its own settings. When activated, the behavior will in turn start

any children it has. Only active behaviors can have children, so a pending behavior will wait before starting its children.

As a rule, only one active behavior can run beneath a given parent, which creates a problem: what to do when multiple behaviors are able to run. We need to set a priority to determine which sub-task is more important. When starting children as prioritized, we define their priority implicitly based on the order the behaviors are added to the parent. The earlier a behavior is listed, the higher its priority (see *Isla* in References).

This solution avoids the problem that would have resulted had we determined priority strictly by number, such as priority-creep, in which priorities become larger and larger, trying to trump the rest. Here we localize the priority definition, so it's only relative to the small subset of behaviors that are started as siblings.

In the example above, we see a hierarchy of behaviors and the children available under each, with `fire` active as a child of `attack`, which in turn is active as a child of `combat`. If the currently pending behavior (`dodge`) were to determine that it needs to start (when the NPC detects it is being fired upon), it will interrupt its active sibling (`attack`) and revert it to pending, which in turn will delete all children of `attack`. Once no other active sibling behaviors are running, `dodge` may begin.

This method of applying priority implicitly works well in most cases, but sometimes the importance of the tasks cannot be described with a simple linear ordering. To handle cases in which a behavior is doing something important and should not be interrupted by non-critical tasks (even if they're higher priority), we can implement a feature called "can interrupt." Essentially, an active behavior may receive a boost in priority, preventing interruption during specific parts of its execution.

With this boost, priorities can be specified in ways more complex than simple linear ordering. For example, while `melee` is listed at a higher priority than `dodge` in Figure 1, it should still be allowed to finish its animation even if `dodge` decides to start—by giving it a boost in priority while running, we prevent `dodge` from cutting it off mid-animation.

**Sequential and random children.** Other ways to start child behaviors are known as sequential and random. Behaviors that are started sequentially are run in the order that they are listed. If the first can run, it will do so until it completes on its own, followed by the second, and so on. When the last behavior in the sequence finishes, the parent finishes as well. For a group of child behaviors started randomly, only one will be chosen to run, and the parent will complete once its child finishes.

**Non-blocking children.** Behaviors can also be started as non-blocking, in which case they may activate even if there are already other active behaviors running beneath the parent. They exist outside the prioritized list. These behaviors are useful for performing tasks that work simultaneously with others, such as firing while moving, or playing a voice over on a specific condition, or activating and deactivating effects. Generally, anything performed by a non-blocking behavior must not interfere with any other sibling behaviors that might be running, since a non-blocking behavior will only be interrupted when its parent is deactivated (and never by a sibling behavior).

By using various combinations of these methods up and down the tree, we can form decisions and task handling over multiple levels that would be difficult to define in a single behavior.

## BUILDING NEW PUZZLE PIECES

From this framework, we have defined the basic unit that will be used to construct HFSMs: the behavior. Now, by making behaviors sufficiently data-driven, we can expose not only their values and settings for modification, but the structure of the actions they perform as well. (See Figure 2.)

Each behavior is self-contained. Everything about it is determined within the behavior itself: when it can activate, when it can no longer run, what interrupts it, and what it does on activation, deactivation, and update. Most significantly, though, it defines what children it starts. Each of these features of a behavior are set within a .behavior file, one per behavior, which is parsed and read in with the rest of the game data.

Once a behavior is created in a pending state, its duty from then on is to decide when it can activate, since it contains its own activation requirements. These requirements are defined by attaching a list of precondition objects to the behavior. A precondition is a set of rules that may be evaluated to be true or false after checking conditions from the world. Game events, the health level of the character, and commands issued from a squad leader are all conditions that can contribute to activating a behavior, and a precondition can be configured to query any of them.

The settings that define a precondition are stored in a separate file and can be reused by other behaviors as well. This allows us to construct a library of preconditions that are easily selected for a new behavior simply by listing them in the behaviors configuration file.

Once a behavior is activated, it can perform its actual purpose, which in most straightforward cases is data-defined as well. Any behavior can start an asset on the character (play an animation, sound, or effect) or start more children without requiring any code side changes, which gives us tremendous flexibility in being able to quickly flesh out the structure of new behaviors. For more complicated actions, we use code-supported behaviors.

A code-supported behavior has all the configuration settings available to a base-level behavior, but with some extras available that plug into a corresponding module created in code. `fireweapon`, `pathfollow`, and `melee` are examples of behaviors that have a code side associated with them, performing actions that would be too specific to generalize and make available for all behaviors. In `fireweapon` we have, for example, settings for “delay between shots” and “shots to fire in a burst,” that would be useless on most other behaviors. These code-supported behaviors fit into the tree like any behavior and are typically found as the leaves of an HFSM, used at the bottom level as children of more abstract behaviors. Generally, the role of the higher-level behaviors is to separate out objects and tasks that need performing on those objects, and then start children to make them happen.

However, because the objects we’re dealing with aren’t defined until the game is actually running (such as the current target of a character during a battle or the last person to damage the player), we need a way to reference and make decisions about those objects within our behaviors. We need a parameter system.

## CONNECTING OBJECTS TO BEHAVIORS

Giving behaviors a list of settings in their configuration file adds a lot of generalization to the system, but it does not allow objects to be dynamically manipulated at run time, responding to arbitrary situations. For example, the target of `fireweapon` cannot be known when that behavior is being written, as it will vary.

Each behavior must be able to understand and make decisions about arbitrary game objects in the world, decided at run time.

The solution we used in `DESTROY ALL HUMANS 2` was to allow parameters to be passed from a parent to its children when the children were created, letting those behaviors query and manipulate the object passed in as a parameter or pass it along to their own children.

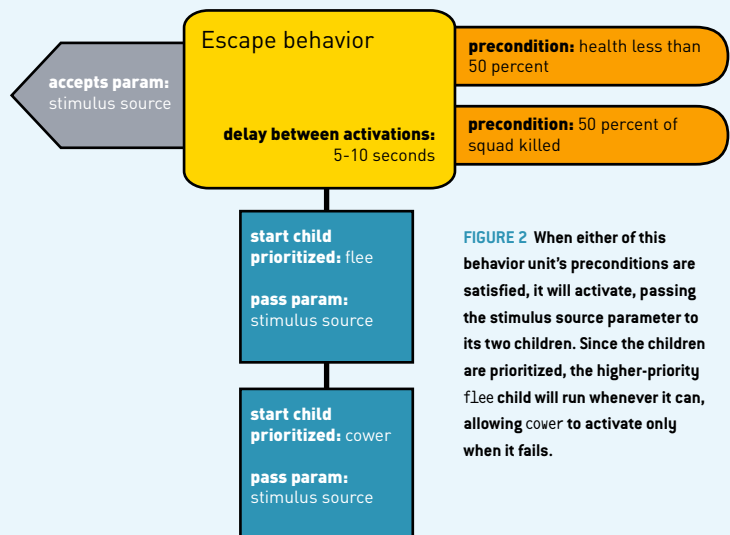
Any behavior that needs to accept a parameter defines a slot for it, which must be filled by whatever parent behavior activates it as a child. From there, the variable in that slot can be used in a number of ways: it can be passed on to its own children, sent an event or message, or, for code-supported behaviors, made available to the code side. `Pathfollow` will seek to the object passed as its first parameter, for example, while `melee` will swing in the direction of its first parameter.

With this addition, the states in our HFSM can essentially send objects along their transitions (in our case, between parent and child), plugging them into other behaviors that expect them and use them as a target of their functionality. It’s a way to blend some script-like functionality into the more rigid structure of an HFSM, giving extra flexibility without sacrificing organization.

Between concrete and abstract are partially-implemented behaviors. The ability to bypass parameters opens up a lot more ways to use behaviors, and many uses of parameters became quite common in our implementation. To help us with some of the standard ways in which parameters could be used, a few partially-implemented behaviors were created that would handle some standard tasks on the code side, even though they themselves were not full code-supported behaviors. That is to say they didn’t belong as leaves of the HFSM; they were just helper behaviors that were still abstract until they were given the data that configured their actions.

The most commonly used partially-implemented behavior was `rangetest`, which would accept a target parameter that it would track and store throughout its existence, using it to make a number of decisions.

Because parameters are passed as soon as a behavior is created, even before it is activated, we can use them to determine whether we should activate it. In the case of `rangetest`, extra



**FIGURE 2** When either of this behavior unit’s preconditions are satisfied, it will activate, passing the stimulus source parameter to its two children. Since the children are prioritized, the higher-priority `flee` child will run whenever it can, allowing `cower` to activate only when it fails.



# CREATING ALL HUMANS

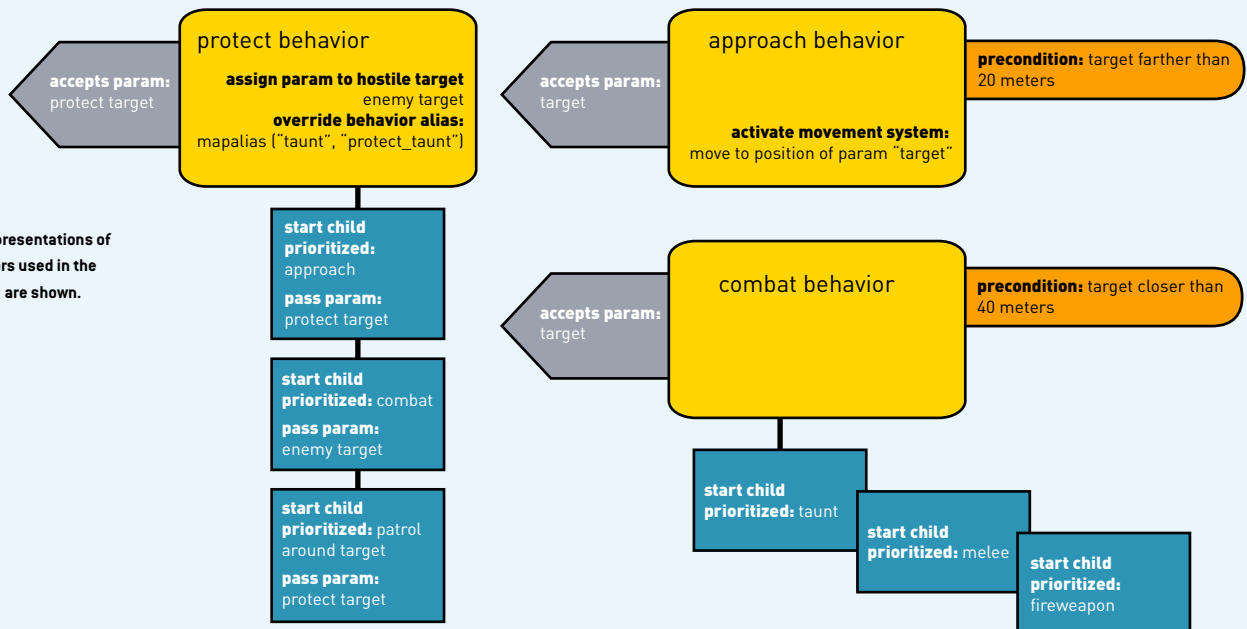


FIGURE 3 Representations of three behaviors used in the protect HFSM are shown.

settings were available to test the range of the passed target (the first parameter) and deactivate it if it left a second range.

This functionality proved generally useful. Because the behavior can serve as the parent for both data-defined behaviors and code-supported behaviors, like `pathfollow` or `melee`, it is able to break tasks into responses based on properties of objects in the world and provide failcases implicitly when targets are too far away. By nesting `rangetest` behaviors and applying them to different objects at different times, we can define complex decisions about dynamic objects with just a few simple pieces.

## PROTECT BEHAVIOR

The behavior `protect` illustrates this point: when it is activated, it's cued to protect as its first parameter. It then starts a list of prioritized behaviors, the first of which is to stay at all times within a reasonable distance of the character being protected,

represented by `approach`. `Approach` is implemented as a `rangetest` behavior that activates when the target is too far away, calling into the movement system to run closer. (See Figure 3.)

Second, a character in `protect` should engage any nearby enemies, which in our implementation are detected by a query to a target selector module running on the character (modules like these run separately from the behavior system). `Combat` is started in order to

maintain this requirement (another `rangetest` behavior), and if an enemy is too close, it will activate and start children to engage the target.

If neither `approach` nor `combat` are able to start (because their preconditions are not satisfied), `wander` will activate by default, since it is the lowest priority behavior and has no preconditions. The character will then patrol randomly around the actor being protected.

By creating behaviors that accept parameters at runtime, we are able to define a structure entirely in data to perform two very different tasks, while acting on multiple entities involved in those tasks—the person who should be protected and the enemies that pose a threat.

## SHARING AND REUSING BEHAVIORS

A distinct advantage of defining a character's actions hierarchically is the ability to reuse, replace, and remove behaviors from the hierarchy easily and intuitively.

Every behavior in this system (except for the simplest) has its settings defined in its `.behavior` settings file. However, when a behavior spawns its children, it does not start them using the filename directly; it uses an alias. Any character that starts behaviors will define a list of aliases, mapping each to a `.behavior` file. By abstracting behavior referencing by one layer, we are able to customize and reuse the behavior components by changing how that mapping is defined.

One of the down sides of using a flat FSM instead of an HFSM to drive your characters is the inability to quickly modify an existing behavior into a new one by changing only one aspect of it. To solve this in a FSM, you would need to recreate the entire state machine again, save the one difference.

We avoided this problem with our implementation by adding the ability to swap out behaviors at any layer in the hierarchy. By changing the mapping of a behaviors alias for a given character, you can swap out the actual behavior that's created when it references that alias, regardless of where the

### Common combat behaviors map

```

MapAlias("Combat", "pedestrian_combat.behavior")
MapAlias("Melee", "pedestrian_melee.behavior")
MapAlias("PathFollow", "pedestrian_pathfollow.behavior")
MapAlias("Patrol", "pedestrian_patrol.behavior")
MapAlias("Protect", "pedestrian_protect.behavior")
MapAlias("Approach", "pedestrian_approach.behavior")
...
  
```

### Ninja behaviors map

```

INCLUDE("Common Combat Behaviors Map")
OverrideAlias("Melee", "ninja_melee_claw")
OverrideAlias("FireWeapon", "ninja_throw_shuriken")
OverrideAlias("Flee", NONE)
  
```

FIGURE 4 A common behavior map, and a specialized one for ninjas, who have different `melee` and firing methods, and never `flee`, are listed.



behavior sits in the hierarchy. This feature was commonly applied to give characters customizations on the behaviors that were widely shared, such as giving ninjas a special variant of melee or fireweapon. (See Figure 4.)

Aliases are typically defined in the file associated directly with the actor, but we gain some extra flexibility by allowing aliases to be redefined anywhere in the behavior tree as well. For example, consider the protect behavior described earlier. There are actually two variants of the approach behavior that are used as children within the HFSM, one to follow the character they are meant to protect and another for approaching enemies to engage in combat.

Conceptually, these behaviors do the same thing (move the character to a target), but the way they do it is different. For example, the combat variant can strafe around and move in a more aggressive way, while the protect variant simply runs to the target's location of the character it is sent to protect.

We can allow this customization in separate branches with the override alias setting that any behavior can contain. When this setting is present in a behavior, it triggers any child behaviors that it or its descendants activate to instead use the new mapping. For example, we can override approach in the combat branch to use a more aggressive version, while the opposite branch separately overrides it to use a less aggressive version. Now, whenever any of the descendants activate that alias, it will filter up through each parent in the HFSM, checking in turn for a new alias mapping until it finds the ones we assigned.

Another simple method that was very successful in customizing behaviors was adding the ability to not just customize, but remove entire behaviors from beneath a parent. This was accomplished by mapping a behavior alias to a special "none" keyword, which when encountered would simply not start the behavior. This was very useful in producing variants of enemies that didn't throw grenades or didn't dodge, for example.

## AI FOR THE MASSES

In a game that features sandbox-style play, the AI needs to provide enough different and interesting characters to interact with in the world, and the size of the world doesn't have to get very big before it becomes unfeasible to hard code them all. Sometimes, even exposing behavior settings isn't enough—the structure of the tasks and subtasks must be exposed as well, in a way that's powerful but also simple to use.

In *DESTROY ALL HUMANS 2*, the choices we made regarding AI architecture were intended to promote those traits. It's an adaptable, puzzle piece-like system in which functions are exposed in a generic way. It attempts to skirt the line between behaviors that are entirely hard coded and ones that are entirely script-defined, left to the technical designers to manage.

Instead, a system like Pandemic Australia's packages that complexity and exposes it as individual pieces to be fit together at various levels of abstraction. As a result, we generate extra flexibility in the way those pieces can be reused and expanded, multiplying their usefulness and bringing us a step closer to populating a virtual world with virtual life. ❖



**DESTROY ALL HUMANS II** places a large number of non-player characters in an open world.

## RESOURCES

Alt, G. "The Suffering: A Game AI Case Study," in *The Proceedings of the Nineteenth National Conference on Artificial Intelligence*, July 2004.

Atkin, M. et al. "Hierarchical agent control: a framework for defining agent behavior," in *The Proceedings of the Fifth International Conference on Autonomous Agents*, May 2001. pp. 425-432.


Freed, M. "Managing Multiple Tasks in Complex, Dynamic Environments," in *The Proceedings of the Fifteenth National Conferences on Artificial Intelligence*, July 1998.

Fu, D. and Houlette-Stottler, R. "The Ultimate Guide to FSMs in Games," in *AI Game Programming Wisdom 2*. Hingham, Mass.: Charles River Media, 2003.

Isla, D. "Handling Complexity in the HALO 2 AI," in *The Proceedings of the 2005 Game Developers Conference*, San Francisco, March 2005.

Yiskis, E. "Finite-State Machine Scripting Language for Designers," in *AI Game Programming Wisdom 2*. Hingham, Mass.: Charles River Media, 2003.

# Anark Gameface



**The UI tool of choice  
for AAA developers**


## Version 3.7 10.17.06

Optimized Workflow

Enhanced  
Performance

Collada Animation  
Support

- ▶ Rapidly author next-generation 3D UIs
- ▶ Flexible artist-driven workflow
- ▶ Mature 3D authoring platform
- ▶ Reduce programming requirements
- ▶ Optimized for PS3, Xbox 360 & PC



**Evaluate now:  
anarkgameface.com**

POSTMORTEM  
~~POSTMORTEM~~

# DETONATING INTROVERSION'S DEFCON

**DEFCON WAS CONCEIVED AND PROTOTYPED DURING THE LONG DEVELOPMENT** period of our previous game, DARWINIA. Looking for a change of scenery, I decided to try to put together an entire multiplayer game in just 24 hours. DEFCON's main source of inspiration was the 1983 cult film *WarGames*. It was a film that I had always loved as a child, and references to it can be found in the hacking elements of our first game UPLINK, and more generally with DEFCON.

Watching it again about a year ago, I realized that there was a really cool game idea just waiting to be tried out, which to my knowledge, nobody had previously attempted. A lot of games that simulate global warfare tend to be heavily strategic, and thus very turn-based and stat heavy. I wanted something that looked like the movie, with vector-lined Soviet subs closing in on the coastline. I stole code from all of our other games in development and built a very rough prototype as quickly as I could. In hindsight, it's obviously impossible to produce a real game in a day—but I finished the prototype within a week, with working multiplayer gameplay over LAN.

The intention was always to capture the feeling of senseless Armageddon and the claustrophobia of being buried deep underground, trying to win a war that simply cannot be won. DEFCON was always meant to be an experiment in the creation of a mood. We wanted to see if we could evoke that vast and terrible detached feeling of genocidal mania.

Once the prototype was finished, I went back to completing DARWINIA, but eventually we hired Gary Chambers to help complete DEFCON. Gary was one of the original Introversion die-hard fans, and he used to run fan sites for our first two games. We were looking for somebody to help finish DEFCON and Gary had just graduated from university, so it seemed like the perfect opportunity. In terms of programming, DEFCON was a two-man effort, lasting a little less than a year.

The other key players are Alistair Lindsay and Michael Maidment, the audio team behind DEFCON and our previous game DARWINIA. Alistair is a master at designing and composing just the right sort of moody music and sound effects. It's all geared toward making the player feel a

---

**CHRIS DELAY, VICKY ARUNDEL, THOMAS ARUNDEL, GARY CHAMBERS, and JOHN KNOTTENBELT** are respectively lead developer, press officer, commercial director, developer, and technical director at Introversion Software, an independent developer based in the U.K. Email the DEFCON team at [Introversion@gdmag.com](mailto:Introversion@gdmag.com).

DEFCON  
EVERY  
BODY  
EVER







new york hit, 11.9m dead

launch detected

launch detected

launches detected

launch detected (enemy unit)

launch detected

santiago hit, 5.5m dead

- △  
units
- ⊙  
alliances
- ◎  
comms
- 📍  
scores
- 📡  
radar
- 👤  
population
- ☢️  
orders
- 📄  
info



## GAME DATA



### DEVELOPER

Introversion Software

**NUMBER OF FULL-TIME DEVELOPERS**  
2

**RELEASE DATE**  
September 29, 2006

**BUDGET**  
£50,000 (about \$95,000)

**LENGTH OF DEVELOPMENT**  
12 months

**PLATFORMS**  
Windows PC, Mac and Linux versions in development

**DEVELOPMENT HARDWARE**  
P4 3Ghz, 2GB RAM, GeForce 7700GT

**DEVELOPMENT SOFTWARE USED**  
Visual Studio 2005, Subversion source control

**TECHNOLOGIES USED**  
OpenGL

**PROJECT SIZE**  
100,000 lines of C++

certain way—detached from the horror. We worked very closely with Alistair to figure out what sort of sound we wanted, and he then went away and composed and recorded the music, with Michael's help. The finished audio soundtrack is made up of 60-second segments which are seamlessly blended together in real time during the game. We run a number of filters and effects on top of that, slowing the sound down and bringing up the bass levels as the player's population dies, and distorting the music when large numbers of nukes detonate at once. The results are incredible.

—Chris Delay, lead developer

## WHAT WENT RIGHT?

**1 INCREASED VISIBILITY.** My first job at Introversion was to call local newspapers and magazines to see if they'd be interested in reviewing our first game, UPLINK. It was a nightmare job, as no one had even heard of us, let alone our game. Five years later, DEFCON has been a joy to promote—the media has been incredibly enthusiastic about reviewing the game.

With UPLINK, we actually spent quite a lot of money on marketing but saw very little return on our investment, as we couldn't afford to back up our advertising with longer running campaigns. With the launch of DARWINIA, and to an even greater extent with DEFCON, we decided to ditch the marketing to concentrate our efforts on PR and find ways to improve our profile for free. Our positioning as "the last of the bedroom programmers" has really captured the imagination of journalists and consumers alike, which has led to numerous interviews and features, independent of the game reviews.

—Vicky Arundel, press officer

**2 THE PRICE WAS RIGHT.** Getting the price point right for a new launch is always tricky. If it's too high, you run the risk of seeing a low take-up until it's discounted; too low, and you'll lose out on revenue, and (particularly in retail) your product might be perceived as being of poor quality or a budget title. Getting an army of early adopters is especially important for multiplayer games, as nothing kills the prospects of a multiplayer game like an empty server.

While our previous game DARWINIA had multiplayer capabilities, we still learned the hard way about overpricing at launch. DARWINIA was put online and into U.K. retail initially at £30, which in hindsight was too much. By the time we discounted the price for DARWINIA, we had already missed a large number of customers who had moved on to other games. It required a veritable re-release on Steam for \$20 and U.S. retail for \$30 to re-attract those customers—at almost half our initial asking price.

With DEFCON, I'm confident that we got the \$15 price point just right. The price may need to be modified slightly for a boxed retail release, but with digital distribution we were able to set the price point almost entirely as we pleased. DEFCON is a scenario-based game and as such doesn't have a whole lot of content. The game experience can vary vastly based on whom you're playing with, so it was important for us to make a demo version of DEFCON for people to try before purchasing. By combining the demo and the full version into the same build and separating them by a server-checked key (a method common in the casual games space) we were able to focus on distributing keys and letting the game executable spread virally, as people asked their friends to join them for a friendly game of nuclear war.

—Thomas Arundel, commercial director

**3 WORLD VIEW.** Early on in DEFCON's life, when we still referred to it by its code name, Wargames, we were going for a fairly authentic 1980s experience from a nuclear power point of view. The game only had two playable teams, which were NATO—made up of what is now the North America and Europe territories—and Soviet, which was just Russia. About two months in, Chris had the idea of expanding the teams so that instead of the two fixed groups, there would be six, made up of all the major territories of the world.

We split NATO and added South America, Africa, and South Asia to the list of playable territories and ended up with what you currently see in the game. From this one change came all the alliance options in the game, as well as a lot of the other options that make DEFCON so customizable. It was a turning point in the game's development, and without that change, DEFCON would have turned out very differently.

**4 EXISTING NETWORK STRUCTURE.** Although DARWINIA was a single player game, it was built on a multiplayer core. Instead of sending packets across a network, DARWINIA simply drops them straight into its own inbox for immediate processing.

The original prototype of DEFCON was made in just a few days, and was put together so quickly because it borrowed heavily from DARWINIA's network architecture. DEFCON still uses the same system now, albeit heavily modified, but because of this initial step, we were able to make progress on DEFCON much quicker than we would have been able to if we had needed to write the entire networking system from scratch. Because of this, we were able to play multiplayer games relatively early on, which helped us refine the multiplayer experience.

**5 BASED ON REAL-WORLD DATA.** Since all the actual information that DEFCON uses to represent the world is readily available on the internet, it was a very simple task to get data about international borders, continental coastlines, and the location of major cities and create a parser to read them directly into DEFCON. Not only was this much faster than generating all this information ourselves, it was also a lot more accurate. As a side effect, because the information is stored in such a simple way, people have already started creating tools to edit these files and create their own fully working maps for DEFCON.

—Gary Chambers, developer

## WHAT WENT WRONG?

**1 BUMPY LAUNCH.** The launch for DEFCON wasn't the smoothest. The first big lesson we learned is that having a countdown timer on your web site requires some serious planning. Of course, everyone checked back to see it count down, and our web site died





when it spiked to over 20 times the normal usage toward the end. Our store also took a beating an hour before the keys were due to be issued at 6 p.m. local time, as a number of people thought that 6 p.m. GMT meant 6 p.m. BST. Since the keys hadn't been issued, customers kept checking their key page every 20 seconds to see if it had been updated (even though it was still an hour off), so the store also crashed, due to exceeding normal usage by 100 times. That took us by surprise.

We were also unlucky in that our ISP's Canadian data center—which hosted our web sites—lost all internet connectivity, as both its primary and secondary fiber-optic uplinks were severed. We were plagued by the slow DNS update problem over the weekend as we tried to deal with parts of our server farm that had hardware faults, and it didn't help that our ISP dismantled our largest download server on Sunday morning. What a disaster! It's hard when you're small—we can't afford top tier ISPs, which can make dealing with load spikes very tricky.

The most important server, used for game-matching, did hold up well though, so I think our priorities for server fixes over the launch weekend worked out well. We've learned from all this and have a better idea of what to expect for the next game, so future launches should be much smoother.

—Thomas Arundel

**2 OPTIMIZED IN THE WRONG AREAS.** Although the game-matching server held up very well, with no connectivity problems, we did encounter a few problems. One of the main problems was that during stress testing we incorrectly modeled the player's expected usage pattern. We thought that most people would want to join an existing game of DEFCON rather than start a fresh one, so we had optimized the server to cope with more listings requests than

game registration requests. In practice, however, most new players have been more interested in creating their own game than joining an existing one. This resulted in longer than expected game lists, which caused a higher than expected load on our server, which resulted in game listings that were more out of date. Since launch, we have worked hard to optimize the server for the real usage pattern, and the server listings are now current to within one minute.

Over the launch weekend we also had some problems authenticating Steam users. Several customers complained that the system would recognize them only as demo players, even though they had purchased the full game. Fortunately, with some excellent support from Valve, we were able to successfully resolve this problem.

—John Knottenbelt, technical director

**3 ERROR TRACKING.** DEFCON uses a deterministic physics system in order to make sure that all players in the game see the same thing, instead of sending the entire game state every second. While this is efficient, any errors in the system can become very hard to track down. It works under the principle that every client will calculate the same result for every calculation. This means that any calls to random functions must be made in the same order, and anything that affects the

state of the game must be done in network safe code.

The result is that the system uses very little bandwidth and is very tolerant of latency, but when something causes the clients to fall out of sync, the cause can often be very hard to track down, and a lot of time was spent fixing these issues.

**4 RELEASE DELAY.** The original target release date for DEFCON was early April, but a number of schedule slips meant that beta testing on the game didn't even start until mid-April. Because there was so much new technology to be tested, most specifically the deterministic physics system, the release date slipped by nearly six months to the end of September. DEFCON is the first multiplayer game we've made, so it was all new ground for us, and lots of things we hadn't properly anticipated needed to be fixed before the game was viable for release.

**5 LACK OF ESTABLISHED DESIGN.** The overall goal for DEFCON was quite fuzzy for some time. We didn't know exactly where we were going with the game at the start and spent quite a lot of time playing with ideas and adding features to the game, which ultimately cluttered the display, making it very hard to follow what was going on. Many of these elements ended up being scaled back or removed completely. While it helped us get a better idea of what worked and what didn't, and to get a better idea of what we wanted DEFCON to be, a little bit more planning early on would have saved us quite a bit of time, enabling us to either move the release date forward or make the game better overall. ❄

—Gary Chambers



The Introversion-eers from left to right: Tom Arundel, John Knottenbelt, Chris Delay, and Mark Morris.



INDEPENDENT  
GAMES FESTIVAL

REWARDING INNOVATION  
IN INDEPENDENT GAMES

THE 9TH ANNUAL

# INDEPENDENT GAMES FESTIVAL

MARCH 7-9, 2007 • SAN FRANCISCO, CA

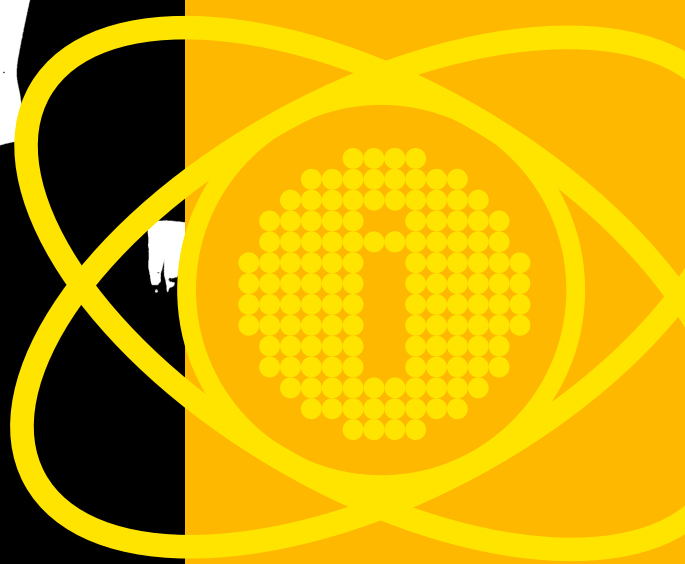
**FINALISTS ANNOUNCED**

MAIN COMPETITION **DECEMBER 9**

MOD COMPETITION **DECEMBER 18**

STUDENT SHOWCASE **JANUARY 4**

**WWW.IGF.COM**



PLAY THE BEST  
INDIE GAMES  
OF THE YEAR

THE IGF PAVILION  
AT GDC 2007  
MARCH 5-9, 2007  
SAN FRANCISCO, CA  
[WWW.GDCONF.COM](http://WWW.GDCONF.COM)





MICK WEST

## » THE INNER PRODUCT

# OPTIMIZING ASSET PROCESSING

### THE FUNDAMENTAL BUILDING BLOCK OF

any game asset pipeline is the asset processing tool. An asset processing tool is a program or piece of code that takes data in one format and performs some operations on it, such as converting it into a target specific format, or performing some calculation, such as lighting or compression. This article discusses the performance issues with these tools and gives some ideas for optimization with a focus on minimizing I/O.

### THE UGLY STEPSISTER

Asset conversion tools are too often neglected during development. Since they are usually well specified and discrete pieces of code, they are often tasked to junior programmers. Generally, any programmer can easily create a tool that works to a simple specification, and at the start of a project the performance of the tool is not so important because the size of the data involved is generally small and the focus is simply on getting things up and running.

However, toward the end of the project, the production department often realizes that a large amount of time is being wasted waiting for these tools to complete their tasks. The accumulation of near-final game data and the more rapid iterations in the debugging and tweaking phase of the project make the speed of these tools of paramount importance.

Further, time may be wasted trying to optimize the tools at this late stage, and there's a significant risk that bugs will be

introduced into the asset pipeline (and the game) when making significant changes to processes and code during the testing phase.

Hence, it's highly advisable to devote sufficient time to optimizing your asset pipeline early in development. It's also advisable to use the people who are highly experienced in doing the types of optimizations needed. This early application of optimization is another example of what I call mature optimization [see "Mature Optimization," *Game Developer*, January 2006].

There's a limited number of man hours available in the development of a game. If you wait until the need for optimization becomes apparent, you will have already wasted hundred of hours.

### THE NATURE OF THE DATA

Asset processing tools come in three flavors: converters, calculators, and packers. Converters take data that are arranged in a particular set of data structures and rearrange them into another set of data structures, which are often machine- or engine-specific. A good example here is a texture converter, which might take textures in .PNG format and convert it to a form that can be directly loaded into the graphic memory of the target hardware.

Asset calculators take an asset or group of assets and perform some set of calculations on them such as calculating lighting and shadows or creating normal maps. Since these operations involve a lot of calculations and several passes over the data, they typically take a lot longer than the asset conversion tools. Sometimes they take large assets, such as high-resolution meshes, and produce smaller assets, such as displacement maps.

The third processing tool type, asset packers, take the individual assets and package them into data sets for use in

particular instances in the game, generally without changing them much. Using an asset packer might involve simply gathering all the files used by one level of the game and arranging them into a .WAD file. Or it might involve grouping files in such a way that streaming can be effectively performed when moving from one area of the game to another. Since the amount of data can be very large, the packing process might take a lot of time and be very resource intensive, requiring lots of memory and disk space, especially for final builds.

### TWEAKING OPTIMIZATION

You may be surprised how often the simplest method of optimization is overlooked. Are you letting the content creators use the debug version of a tool? It's a common mistake for junior programmers, but even the most experienced among us sometimes overlook this simple step.

So before you do anything, try turning the optimization settings on and off to make sure there's a noticeable speed difference. Then, in release mode, try tweaking some settings, such as "optimize for speed" and "optimize for size." Depending on the nature of the data (and the hardware your tools are running on), you might actually get faster code if you use "optimize for size." The optimal optimization setting can vary from tool to tool.

Be careful when tweaking the optimization settings to test the speed of your code. In a multitasking operating system like Windows XP, a lot is going on, so your timings might vary dramatically from one run to the next. Taking the average is not always a useful measure either, as it can be greatly skewed by random events. A more accurate way is to compare the lowest times of multiple runs of two

MICK WEST was a co-founder of Neversoft Entertainment. He's been making games professionally since 1988, and currently works as a technical consultant. Email him at [mwest@gdmag.com](mailto:mwest@gdmag.com).

different settings, as that will be closest to the “pure” run.

### PARALLELIZE YOUR CODE

Most PCs now have some kind of multicore and/or hyper-threading. If your tools are written in the traditional mindset of a single processing thread, you’re wasting a significant amount of the silicon you paid for, as well as the time of the artists and level designers as they wait for their assets to be converted.

Since the nature of asset data is generally to be large chunks of homogeneous data, such as lists of vertices and polygons, it’s generally very amenable to data level parallelization with worker threads, where the same code is run on multiple chunks of similar data concurrently, taking advantage of the cache. For details on this approach see “Particle Tuning” (*Game Developer*, April 2006).

### TUNE YOUR MACHINES

Antivirus software should be configured so that it does not scan the directories that your assets reside in, nor the actual tools. Poorly written antivirus and other security tools can significantly degrade the speed of a machine that performs a lot of file operations. Try running a build both with and without the antivirus software and see if there is any difference in speed.

Then consider removing the antivirus software entirely.

If you have any form of distributed “farm” of machines in the asset pipeline, beware of any screensaver other than “turn off monitor.” Some screensavers use a significant chunk of processing power. You need to be especially careful of this problem when repurposing a machine; the previous user may have installed her favorite screensaver, which doesn’t kick in for several hours, and then slows the machine to a crawl.

### WRITE BAD CODE

In-house tools don’t always need to be up to the same code standards as the code you use in your commercially released games. Sometime you can get performance benefits by making certain dangerous assumptions about the data you’re processing and the hardware it will be running on.

Instead of constantly allocating buffers as needed, try allocating a “reasonable” chunk of memory as a general purpose buffer. If you have debugging code, make sure you can switch it off. Logging or other instrumenting functions can end up taking more time than the code they are logging. If earlier stages in the pipeline are robust enough, then (very carefully) consider removing error and bounds checking from later stages if you can see they are a significant factor.

If you have a bunch of separate programs, consider bunching them together into one uber-tool to cut the load times. All these are bad practices, but for their limited lifetime, the risks may outweigh the rewards.

### MINIMIZE I/O

Older programmers tend to write conversion tools using the standard C I/O functions: fopen, fread, fwrite, fclose, etc. The standard method is to open an input file and an output file, then read in chunks of data from the input file (with fread or fgets), and write them to the output file (with fwrite or fputs).

This approach has the advantage of being simple, easy to understand, and easy to implement. It also uses very little memory, so quite often tools are written like this. The problem is it’s insanely slow. It’s a holdover from the (really) bad old days of computing, when processing large amounts of data meant reading from one spool of tape and writing to another.

Younger programmers learn to use C++ I/O “streams,” which are intended to make it easy for data structures to be read and written into a binary format. But when used to read and write files, they still suffer from the same problems that our older C programmer has. It’s still stuck in the same serial model of “read a bit, write a bit” that’s not only excessively slow, but also mostly unnecessary on modern hardware.

## LISTING 1 old-fashioned file I/O

```
FILE *f_in = fopen("IMAGE.JPG","rb");
FILE *f_out = fopen("IMAGE.BIN","wb");
fseek(f_in,0,SEEK_END);
long size = ftell(f_in);
rewind(f_in);
for (int b = 0;b<size;b++) {
    char c = fgetc(f_in);
    if (c == 0) c = 0xff;
    fputc(c,f_out);
}
fclose(f_in);
fclose(f_out);
```

## LISTING 2 reading the whole file into memory

```
FILE *f_in = fopen("IMAGE.JPG","rb");
fseek(f_in,0,SEEK_END);
long size = ftell(f_in);
rewind(f_in);
char* p_buffer = (char*) malloc (size);
fread (p_buffer,size,1,f_in);
fclose(f_in);
unsigned char *p=(unsigned char*)p_buffer;
for (int x=0;x<size;x++,p++)
    if (*p == 0) *p = 0xff;
FILE *f_out = fopen("IMAGE.BIN","wb");
fwrite(p_buffer,size,1,f_out);
fclose(f_out);
free(p_buffer);
```

Unless you're doing things like encoding .MPEG data, you will generally be dealing with files that are smaller than a few tens of megabytes. Most developers will now have a machine with at least 1GB of memory. If you'll be processing the whole file a piece at a time, then there's no reason you should not load the entire file into memory.

Similarly, there's no reason you should have to write your output file a few bytes at a time. Build the file in memory, and write it out all at once.

You might counter that that's what the file cache is for. It's true: The OS will buffer reads and writes in memory, and very few of those reads or writes will actually cause physical disk access. But the overhead associated with using the OS to buffer your data versus simply storing it in a raw block of memory is very significant.

Listing 1 shows a simple file conversion program that takes a file and writes out a version of it with all the zero bytes replaced with 0xFF. It's simple for illustration purposes, but many file format converters do not do significantly more CPU work than this simple example.

Listing 2 shows the same program converted to read in the whole file into a buffer, process it, and write it out again. The code is slightly more complex, yet this version executes approximately ten times as fast as the version in Listing 1.

## MEMORY MAPPED FILES

The use of serial I/O is a throwback to the days of limited memory and tape drives. But a combination of factors means it's still useful to think of your file conversion essentially as a serial process.

First, since file operations can proceed

asynchronously, you can be processing data while it's being read in and begin writing it out as soon as some is ready. Second, memory is slow, and processors are fast. This can lead us to think of normal random access memory as a just a very fast hard disk, with your processor's cache memory as your actual working memory.

While you could write some complex multi-threaded code to take advantage of the asynchronous nature of file I/O, you can get the full advantages of both this and optimal cache usage using Windows' memory mapped file functions to read in your files.

The process of memory mapping a file is really very simple. All you are doing is telling the OS that you want a file to appear as if it is already in memory. You can then process the file exactly as if you just loaded it yourself, and the OS will take care of making sure that the file data actually shows up as needed.

This gives you the advantage of asynchronous I/O because you can immediately start processing once the first page of the file is loaded, and the OS will take care of reading the rest of the file as needed. It also makes best use of the memory cache, especially if you process the file in a serial manner. The act of memory mapping a file also ensures that the moving of data is kept to the minimum. No buffers need to be allocated.

Listing 3 shows the same program converted to use memory mapped I/O. Depending on the state of virtual memory and the file cache, this is several times faster than the "whole file" approach in Listing 2. It looks annoyingly complex, but you only have to write it once. The amount of speed-up will depend on the nature of the data, the hardware, and the size and architecture of your build pipeline. ❌

## LISTING 3 using memory mapped files

```
HANDLE hInFile = ::CreateFile(L"IMAGE.JPG",
    GENERIC_READ, FILE_SHARE_READ, NULL,
    OPEN_EXISTING, FILE_ATTRIBUTE_READONLY, NULL);
DWORD dwFileSize = ::GetFileSize(hInFile, NULL);
HANDLE hMappedInFile = ::CreateFileMapping(hInFile,
    NULL, PAGE_READONLY, 0, 0, NULL);
LPBYTE lpMapInAddress = (LPBYTE) ::MapViewOfFile(
    hMappedInFile, FILE_MAP_READ, 0, 0, 0);
HANDLE hOutFile = ::CreateFile(L"IMAGE.BIN",
    GENERIC_WRITE | GENERIC_READ, 0, NULL,
    CREATE_ALWAYS, FILE_ATTRIBUTE_NORMAL, NULL);
HANDLE hMappedOutFile = ::CreateFileMapping(hOutFile,
    NULL, PAGE_READWRITE, 0, dwFileSize, NULL);
LPBYTE lpMapOutAddress = (LPBYTE) ::MapViewOfFile(
    hMappedOutFile, FILE_MAP_WRITE, 0, 0, 0);
char *p_in = (char*)lpMapInAddress;
char *p_out = (char*)lpMapOutAddress;
for (int x=0; x<dwFileSize; x++, p_in++) {
    char c = *p_in;
    if (c == 0) c = 0xff;
    *p_out++ = c;
}
::CloseHandle(hMappedInFile);
::CloseHandle(hMappedOutFile);
::CloseHandle(hInFile);
::CloseHandle(hOutFile);
```

## RESOURCES

Llopis, Noel. "Optimizing the Content Pipeline," *Game Developer*, April 2004.

Carter, Ben. "The Game Asset Pipeline: Managing Asset Processing," *Gamasutra.com*, Feb. 21, 2005.

[www.gamasutra.com/features/20050221/carter\\_01.shtml](http://www.gamasutra.com/features/20050221/carter_01.shtml)





GREG BALLARD

## » BUSINESS LEVEL

# MOBILE MINDSET

## What developers should know before turning to phones

**WITH THE CONSOLE BUSINESS IN A STATE** of generational transition, and with new platforms lengthening development cycles, some developers have considered moving into the mobile space. The mobile business is one of the most rapidly changing segments of the video game business, and developers might find creating games for the mobile platform a refreshing change, especially if they're tired of the long and arduous months (or years) they put into each console project.

### RETURNING TO THE ROOTS OF VIDEO GAMES

Mobile developers have always touted the phone as a platform that gives gamemakers an opportunity to return to the core of video game development.

Games for phones are simpler, smaller, and take a shorter amount of time to develop. The inherent limitations of mobile devices make the work challenging, but also offer opportunities and rewards that developers may not otherwise have.

For example, the average development cycle for a console title is about 24 months, and developers work on a team of 30 to 40 members or more. Each team member is usually extremely focused in his or her area of responsibility and expertise. For those two years, a developer may be solely devoted to one aspect of one game.

But in the same two-year period, a mobile developer might work on a closely-knit team of only three or four people, have ownership of a broader skill set, and fully complete four or five projects.

Working at this pace allows developers to build their skills by learning from their other experiences. And the most notable benefit is that developers have new projects on the horizon almost continuously instead of being placed on the same project for years on end.

### HOW TO CULTIVATE A MOBILE MINDSET

Beyond considering the inherent differences in handset capabilities and vastly shorter development cycles, there are other principles that a developer should take into consideration when transitioning from the console world to the mobile one.

**Flexibility.** The mobile game industry is still young, much like the internet was a decade ago and television 50 years ago. No one who's working in mobile games has been there for more than seven or eight years, and the start-up environment of a mobile games company has a very different pace and structure than a traditional console company. All these factors make the mobile sector a workplace that demands flexibility from all its constituents.

**Global audiences.** In some cases, mobile games are developed for a particular geographic region—but more often, games are designed with a global audience in mind. In order for a mobile game to be a hit, it needs to cross cultural and language borders and appeal to a variety of consumers around the world.

Having a broad and diverse audience always in mind is vastly different from how the console business operates, wherein developers may spend 24 months on a title that is sometimes designated for a specific group of consumers. If a mobile game maker were to work this way, it would prevent up to 75 percent of the potential audience from ever seeing the title. A global mindset is critical for mobile.



**Embrace licenses.** Due to the current business environment, licensed properties are a fact of life for mobile gamemakers. Developers must be prepared to create games that meet the needs and expectations of the license holder.

**Simple design and casual audiences.** So-called casual games—games that have simple yet engaging gameplay and appeal to a wide consumer audience—are the most successful on the mobile platform.

There is a very simple formula to their success: simple game mechanics in an environment that gets progressively more challenging and allows for 5- to 10-minute games. Casual games are the basis of success for mobile developers because they almost always appeal to non-traditional players as well as hardcore console gamers.

### BIG BUSINESS, SMALL PRODUCTS

The growth of the mobile gaming industry and advances in handset technology are reason enough for some to want to make mobile games. But for console game developers, the most compelling reason to develop games for phones is the refreshing change of pace—in workstyle, content, speed of development, and team size.

The mobile game industry is an area where game industry veterans can find new career challenges. It's a different world, with multiple handsets to port to and a smaller screen and processors, but it comes with a unique array of benefits as well. ❖



GREG BALLARD is president and CEO of Glu Mobile. He was previously president of Capcom Software in the U.S. and has extensive experience in the music and e-business fields. Email him at [gballard@gdmag.com](mailto:gballard@gdmag.com).



# DESIGNER GENES

## A field guide to game designers

### A GAME DESIGNER IS AN ODD TYPE OF

beast—but we are definitely a specific type. Although on a given game team there may be design contributions from many professions and personality types (producers, programmers, artists, and others), I'm speaking here of the hardcore designers who have chosen to specialize in design at all levels.

We are fascinated with games, of course, but also tend to share many other qualities. We:

- are generally very smart, but often in unconventional or quirky directions
- balance introvert and extrovert personality traits
- possess logical-technical and intuitive-artistic talents (a rare blend, also notable in architects)
- are fascinated with a wide variety of topics, often being well-versed in both classical and popular culture
- have in-depth knowledge of several (or many) specialty areas (for example paleontology, baseball, the Norman Conquest, and *Buffy the Vampire Slayer*)
- wonder constantly how things work, and why they are the way they are.

It's not only common interests that knit game designers, but perhaps more importantly a common way of looking at the world. For instance, many designers, when presented with a task like driving to and from the office every day, will analyze all the different routes, probably timing them to see which are the most efficient and what variables affect travel time.

### PERSONALITY PLUS

Several years ago at an annual gathering of a few dozen long-time game designers, the question of the Myers-Briggs personality test came up.

Most of the designers had taken it and knew their scores—not surprising, since doing so fits the trait of being intensely curious and wanting to quantify things. Nearly everyone was classified as an NT type, and many like myself were INTJ, characterized for our introversion, intuition, thinking, and judging.

Nearly all the designers were closely balanced between introverted and extroverted, too. Most of us fell into a couple of groups that together make up only about two percent of the general population.

Having such parallel Myers-Briggs test results was one fairly objective confirmation of just how similar we can be. It was also interesting that many of the outlying data points came from people who tended to work more as producers than designers.

### NATURE OR NURTURE?

I suspect there's a strong genetic component to this. It's odd how much kinship I've felt with fellow game designers, even when we can barely understand each other's language—I think foreign linguistic ability may even be inversely proportional to design talent.

But start sketching out the relationship of one level to the next, or gesture to show the trajectory of a series of jumps the character must take, and suddenly we're communicating.

### HOW TO START?

I'm often asked about how to become a professional game designer. One major Catch-22 about becoming a designer is that hardly any company will hire one unless they have experience making games, and of course that begs the question of how to get experience in the first place.

#### Myers-Briggs Personality Key

E	Extroversion	I	Introversion
S	Sensing	N	Intuition
T	Thinking	F	Feeling
J	Judging	P	Perceiving

In the Myers-Briggs analysis, four dichotomies are used to describe personalities; anecdotally, game designers tend to be INTJ.

Ultimately, there's a strong component of the adage, "If you have to ask, you can't afford it" at work. Most of the successful designers I know couldn't help but design—did it compulsively—making board games or even just constructing the rules to play with a ball when they were kids, and sketched out or wrote up or cobbled together their ideas on how to improve existing games.

Even without the professional responsibilities, a hardcore designer will tend to generate a portfolio of design material. And the main way people become designers is by doing something else on a game first, which takes advantage of the trait of having several specific passions. Some designers start in programming, others in art, and others as researchers or assistant producers on a title. Q/A or playtesting is sometimes seen as the 21st century equivalent of starting in the mailroom and working your way up.

If you're an aspiring designer wondering how to get your big break in the industry, and particularly if you feel you have a killer idea for a game, I recommend stopping in on Tom Sloper's web site: [www.sloperama.com/advice.html](http://www.sloperama.com/advice.html). He's assembled a great set of lessons on the joys and pitfalls of being a designer, and although the truth can be painful to face, I think he's right on target. But one thing to remember—if you feel you *must* design games, just be persistent and don't give up hope. You are not alone! ❖

NOAH FALSTEIN has been a professional game developer since 1980. 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).

Game Developers  
Conference Mobile

# GDC MOBILE

March 5-6, 2007  
Moscone Convention Center North  
San Francisco

The definitive mobile gaming event for leading industry professionals looking to expand their knowledge of, and contacts within, the exploding mobile entertainment sector.

Learn more at

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



To attend GDC Mobile, you will need either a VIP, Mobile, or Mobile-Plus pass.  
Register by January 31st, 2007 and

**SAVE UP TO 35%!**



# SERIOUS GAMES SUMMIT GDC

March 5-6, 2007 • [SGS.GDCONF.COM](http://SGS.GDCONF.COM)  
Moscone Convention Center North • San Francisco

The Serious Games Summit provides a forum for game developers and industry professionals to examine the future course of serious games development in areas such as:

- Education
- Government
- Health
- Science
- First Responders
- Corporate Training
- Military
- Social Change

To attend SGS GDC, you will need either a VIP, Giga, or Serious Games Summit pass.  
**REGISTER BY JANUARY 31ST, 2007 AND  
SAVE UP TO 35%!**







STEVE THEODORE

## PIXEL PUSHER

# THE HISTORY CHANNEL

## How I learned to stop worrying and love construction history

### JOURNALISM'S GREATEST MOMENTS

happen when a writer challenges our prejudices and stands up for the oppressed, the outcast, and the unfairly persecuted. In that noble tradition, this month's Pixel Pusher wants to say a few words in defense of one of the most unfairly vilified features in the world of 3D graphics.

Call it the HyperGraph, call it the Modifier Stack, call it things we're not allowed to print. Whatever you call it, construction history wants to be your friend. Sure, the relationship has some issues, but construction history tools are ready and willing to move on if you are.

A lot of veteran poly pushers don't like working with history because,

until the last couple of years, it used to be painfully slow. However, this is not nearly the problem it used to be. A modern dual core box runs about 30 times faster than the machines that 3ds Max and Maya debuted on. Add in bigger caches, more memory, and dedicated graphics hardware and suddenly operations that used to take place in "get some coffee" time are nearly in real time. If you're hitting Collapse Stack or Delete History by reflex because you're afraid of slowing yourself down, you may be prematurely throwing away some power that could make your life a lot easier.

Of course, some people don't like working with history because it doesn't have the immediacy or predictability of old-fashioned poly slinging. Mucking around with a bunch of nested parameters and invisible relationships isn't as satisfying as laying down verts by hand. Unfortunately, hand-details are becoming luxuries.

In the brave new world of next-gen content, where the models have grown but the schedules haven't, it's time to take a second look at any tool that can help us pump out assets. This is not a good moment to be sticking by our prejudices. Let's give history a chance to redeem itself.

### LEARNING FROM HISTORY

Making good use of history features requires a bit of up-front investment in analytical thinking—that's another reason a lot of artists become suspicious. It can be a stretch if you're a seat-of-the-pants intuitive type who doesn't like to plan ahead.

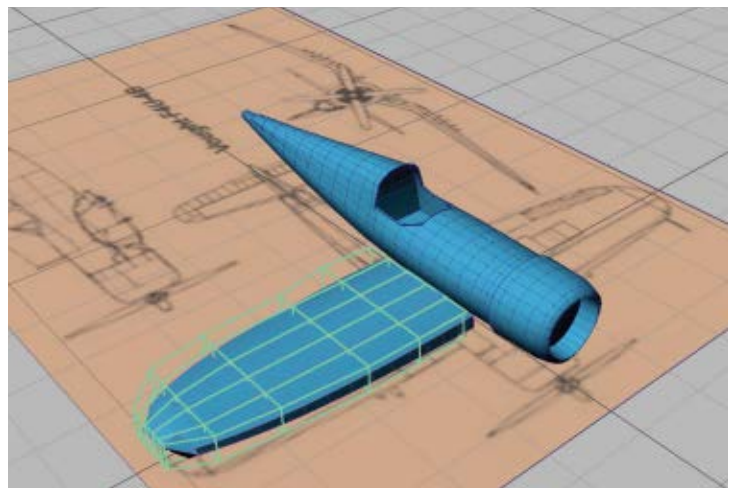
Modeling parametrically rewards the ability to isolate the important characteristics of a shape and break them down into a set of formal relationships and deformations. It's not surprising that mechanical subjects are often the best candidates for history-centric modeling, since engineered contours are much easier to dissect than messy organic forms.

Okay, I admit "analyzing formal relationships" doesn't sound like fun, but it's not just art school BS—it's what any good modeler does by instinct. "Formal relationships" in ordinary language is just the way the shape "works," little more than a concise description of the subject.

A concrete example can make this plain. The WWII vintage Corsair fighter in Figure 1 sports an elegantly curved gull wing, which is tricky to model convincingly since it curves in three dimensions. Although you could approximate the shape pretty fast using subdiv or Nurbs curves, managing the flow lines in all three dimensions is a



**FIGURE 1** Modeling the gull wings of this WWII vintage airplane by hand would be tough.



**FIGURE 2** A simple extrusion, tapered along the front-back axis with an FFD, provides the elliptical wing shape.

**STEVE THEODORE** has been pushing pixels for more than a dozen years. His credits include *MECH COMMANDER*, *HALF-LIFE*, *TEAM FORTRESS*, and *COUNTER-STRIKE*. He's been a modeler, animator, and technical artist, as well as a frequent speaker at industry conferences. He's currently a founding partner at Giant Bite. Email him at [stheodore@gdmag.com](mailto:stheodore@gdmag.com).

huge hassle—misplacing a single control point could screw up the continuity of the entire surface.

### HISTORY 101

However, the shape is actually quite simple if you think about it as a series of steps. Using deformers to build the shape, you can fine-tune the elliptical curve of the wing, its tapering thickness, and its gull-wing bend, all without having to eyeball every vert into place.

The basic shape of the wing is just an airfoil profile (see Figure 2) that's extruded to match the length of the wing. You might be tempted to start with the elliptical shape from the overhead view, since that's the most visible aspect of the "shape." But if you think about how the wing was built in real life (using of a series of airfoil cross-sections), you'll see that the elliptical form is just a warp applied to the extrusion along the wing's forward axis. In this example, it's done with an FFD lattice (see Figure 2).

The next formal element of the finished wing is the way it tapers from nearly a foot thick where it meets the fuselage to only a couple of inches at the wing tip. You could easily model this taper by adjusting the control points of the FFD lattice by hand, which creates the elliptical wing shape. The problem is that it will be difficult to edit. If you found that the finished wing had the wrong taper or

the wrong thickness, you'd have to manually move each control point, possibly in two or three axes at once. It's safer to use a second FFD (in 3ds Max, a Taper deformer would also do the trick) to apply the taper, as shown in Figure 3.

The last and most obvious element of the wing's shape is the bend, which is easily applied with a bend deformer (see Figure 4). The basic shape of the wing is complete. All that's left to do is tweak the first two FFDs to make sure the bend hasn't changed the overall length of the wing too much. Debugging the interactions between all those deformers is very important, which is why letting the history live as long as possible is always a good idea. As you can see from this example, all the talk about "formal relationships" really just translates into a little attention to how a shape is created. If we built the same shape by hand, it would be tedious to make sure that every loft section or edge loop was planar and tangent to the sweep of the wing, but here that's all free. See Figure 5.

### REVISIONISM

The fighter plane example is very simple, but it illustrates neatly the first commandment of deformer-based modeling: Each deformer is responsible for only one aspect of the model.

In this case, separating the contour, the taper, and the bend into three operations

gives us three sets of fast and easy tweaks rather than one complex and painstaking one. As soon as you start layering multiple jobs onto a single deformer, you're effectively back to hand-building the mesh. You will have simply traded in a high-resolution poly mesh for a lower-resolution set of deformer control points. On the other hand if you have done a good job of thinking through how the shape works you'll rarely need more than two or three deformers. If you find you need more than a handful of operations to describe a shape, you need to break down the model into simpler pieces.

At this point you may be thinking, "Yeah, yeah, I use deformers for this sort of stuff all the time, but I still don't see why I have to leave the stinking history on."

It's true that you could build the airplane wing even if you deleted the history after each step. But being too eager to kill the history is very risky. Even working from an accurate plan view—a luxury we don't get all too often—it's easy to misjudge proportions or spatial relationships. Keeping the history around lets you refine the signature components of the form separately.

Keeping the history also lets you separate the form of the object from its poly count. In the example, we could have added or removed cross-sections from the original extrusion without

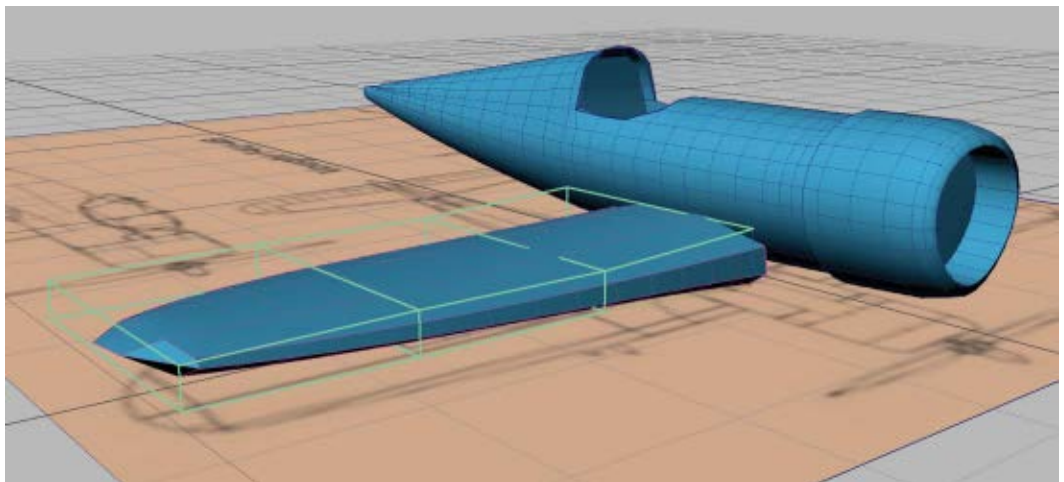
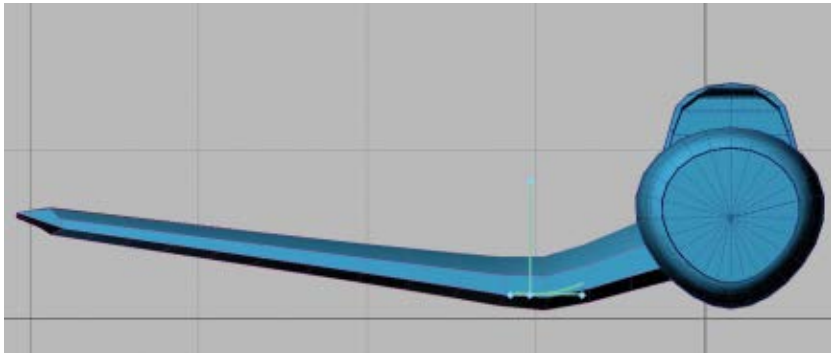


FIGURE 3 A second FFD creates the taper in the wing thickness.



**FIGURE 4** A bend deformer creates the distinctive gull wing curve. Keeping all the deformers alive allows for tweaks if the final result is out of proportion.

compromising the shape as long as we kept the history alive.

This doesn't just mean tweaking face counts, though. History can also be a great aid to detailing. If you've nailed a form with a series of deformers, you can turn the deformers off temporarily and add details to the original surface in a more convenient and accessible position. Adding bevels, UV maps, or surface details, like scales or rivets, is much easier when the object is rolled flat. [See also "The Scale's the Limit," *Game Developer*, September 2006.]

## HISTORICAL PRESERVATION

For all these reasons, you should hang on to your history as long as you can. Nowadays, your viewport draw performance is more likely to slow you down than history is.

More often than not the real breaking point will come when it's time to start editing individual vertices or faces. Neither Max nor Maya, alas, fits vertex or face operations into the flow of construction history very neatly.

Seemingly simple operations, like deleting a vertex or turning an edge, can have surprising effects farther down the history stream. This happens because vertex level modifiers and operations only know vertices by an index number; for example, a UV map operation effectively says, "Apply a projection to face X." Any edits done upstream that

cause the vertices to be renumbered will produce strange results, like mysteriously mobile vertex colors or wacky UVs. Many artists' distrust of construction history stems from nasty experiences with vertex order problems.

If you understand the limitation, it's not too difficult to work around it, but there's no package that seems to handle it neatly either. The safe bet is to try as hard as possible to restrict your vertex editing to the last phase of the model, after all the big forms have been nailed.

Easier said than done. Art classes have been telling us to keep big gestures and small tweaks separate at least since the Renaissance, and most of us still don't listen. At least you can't say you weren't warned.

Luckily, vertex order issues aren't as important as they used to be. When a hero character had 3,000 or 4,000 verts, each one needed to be placed with elegance and precision, and history operations that made that difficult were justifiably shunned. But now, individual

vertices are a lot less important. Fast ways of building denser, more tessellated shapes are coming to the fore—exactly the sort of thing which deformation modeling does well.

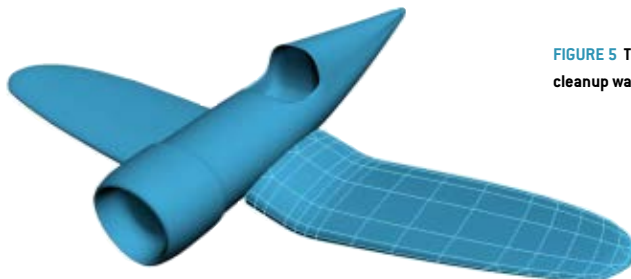
3ds Max users should also learn to love the Volume Select modifier, which allows you to pass vertex selections down the stack based on their location rather than their index numbers. It's particularly handy for UV and vertex color

operations, but also for limiting the effect of other modifiers without handcuffing you to a particular poly count.

## ANNALS OF ART

Wrestling with history really exposes the schizophrenia at the heart of computer art, that funky hybrid of intuition and calculation. It may be that in the next few years newer approaches to modeling will free us from having to think like engineers and let us concentrate on the fun stuff.

For the time being, though, we're stuck with the old tools we've inherited from the auto parts designers and aircraft manufacturers. Even if twiddling numbers in a dialog box doesn't make the heart sing, bending the software to our will has a certain perverse satisfaction all its own. And I don't know about you, but even after all these years I still get a kick out of slapping bends and warps on my models and just making them go all wobbly now and then. ❌



**FIGURE 5** The finished result, with fast hand cleanup was completed in about 15 minutes.





JESSE HARLIN

## » AURAL FIXATION

# LABORING TO COMPETE

## How a new stance on music outsourcing could affect game audio

**GAMES HAVE BEEN CHASING AFTER FILM** since the 1990s in terms of production values, budgets, and audience. For game composers, this has meant shaking off the shackles of chip-set MIDI and stepping onto the recording stage. However, the arrival of live recording to game music introduced a question that film composers have been wrestling with for years: Should they use unions?

The answer has almost exclusively been “no,” due to budgetary restraints and a union interactive media contract that doesn’t keep pace with the evolving technology of our industry. In practice, this has resulted in the choice between hiring non-union musicians, paying union musicians under the table, or turning to alternative musical enclaves for larger orchestral sessions. By far the most successful of these alternatives has been found in Seattle.

Most professional musicians in the United States belong to the American Federation of Musicians (AFM), and it’s the AFM’s contracts that have dictated the terms for film and television music usage for decades. Seattle musicians, however, are part of the separate International Guild of Symphony, Opera, & Ballet Musicians and work from a different standard of contracts. Their no-fuss “buy-out” payment policies (one-time payment with no royalties or reuse fees) have made Seattle the new go-to location for a vast portion of scoring sessions. In addition to Seattle, recent improvements in telephony and data transfer technologies have made outsourcing orchestral recording to European non-union orchestras a viable option for budget-conscious composers.

### ENOUGH IS ENOUGH

For years, AFM has watched unhappily as more and more professional recording dates slip from the union into the non-union column. Despite their displeasure, there was little movement from AFM in terms of a resolution. All that changed this past July.

After watching a century’s worth of protections for professional musicians slowly erode, AFM president Thomas F. Lee issued a press release that officially took a bold stand on Seattle’s alternative scene, so to speak. According to Lee’s statement, as of October 1, 2006 the union now holds all members involved in non-union Seattle recording sessions to be in violation of AFM bylaws, which state:

“No AFM member may perform services (whether as composer, arranger, copyist, proofreader, instrumentalist, leader, contractor, cutter, editor, or in any other capacity): (1) where the product of the services is intended to result in, or be embodied in, recorded music made outside of the United States and Canada and the possessions of either; or (2) for the purpose of producing, editing, or dubbing recorded music except where expressly authorized and covered by a contract with the AFM or when expressly authorized by the AFM.”

### TOUGH COOKIES

Additionally, Lee clarified that—as stated in the AFM bylaws—any member found to violate the code “shall be subject to a fine not exceeding \$50,000 and/or expulsion.” The full press release can be found at: [www.afm.org/public/press/seattle.pdf](http://www.afm.org/public/press/seattle.pdf).

These rules aren’t new and have been part of the union bylaws for some time. What is new is the aggressive drive AFM is making to enforce its existing policies. As for the means of enforcement, the AFM is counting on its smaller local chapters to report members that are in violation and file official charges against them.

While the new hard-line stance is clearly aimed at stemming the tide of film work headed toward Seattle, there’s

nothing in Lee’s statement that exempts game or television work. As such, any members of the union who find themselves involved in game production should be aware of the potential for fines.

Although the AFM doesn’t cover the craft of composition, many composers join in ancillary roles, such as orchestrators, conductors, or instrumentalists to take advantage of union benefits. In addition, AFM members such as orchestrators or copyists often have little—if any—say over whether their work goes to Seattle. It remains to be seen if there will be any leniency when it comes to the filing of official charges. As it stands, there are no distinctions made in the AFM’s policy based upon role or level of involvement, and any union member involved in any capacity with a game project headed for Seattle should be cautious.

### GRUNGE VS. ROMA

For composers looking to avoid the issue all together, it should be no surprise that in addition to non-AFM instrumentalists, Seattle also contains non-AFM conductors, copyists, and music preparation houses. As would be expected, the quality of their work is top-notch and pricing comparable to that of AFM contractors.

Furthermore, orchestras in the Eastern European cities of Prague and Bratislava are now servicing media projects. These non-union options cost roughly 50 percent less than recording in the U.S. However, the level of musicianship does not match that of their domestic counterparts.

It remains to be seen what the AFM’s new policy of directly targeting its members will do to the issue of orchestral outsourcing. What is clear is that the AFM is tired of sitting idly by while globalization spreads to the scoring stage. ❖

JESSE HARLIN has been composing music for games since 1999. You can email him at [jharlin@gdmag.com](mailto:jharlin@gdmag.com).

ARTISTS • DESIGNERS • PROGRAMMERS



LET'S RIDE!

ALL EXPERIENCE LEVELS NEEDED

**SECRET**™

{ [WWW.SECRETFLEVEL.COM/JOBS](http://WWW.SECRETFLEVEL.COM/JOBS) }

**SEGA**®



# CREATE IN NEXT GEN



Imagery from Midway's next-gen console game *John Woo Presents Stranglehold*

- > PROGRAMMERS
- > GAME DESIGNERS
- > ARTISTS
- > PRODUCERS
- > ANIMATORS
- > Q.A. ANALYSTS
- > MANAGEMENT

[jobs.midway.com](http://jobs.midway.com)

RECENT RELEASES:

AREA 51

BLITZ: THE LEAGUE

MORTAL KOMBAT: SHAOLIN MONKS

NBA BALLERS: PHENOM

THE SUFFERING: TIES THAT BIND

ANNOUNCED NEXT GEN TITLES:

JOHN WOO PRESENTS STRANGLEHOLD

TNA WRESTLING

THE WHEELMAN

Visit us at the Austin Game Conference,  
Booth Number 616



AUSTIN \* CHICAGO \* LOS ANGELES \* NEWCASTLE UK \* SAN DIEGO \* SEATTLE

Midway is an Equal Opportunity Employer and does not discriminate on the basis of race, color, religion, sex, age, national origin, disability, veteran status, sexual orientation or any other classification protected by Federal, State or local law. This notice contains a "forward-looking statement" within the meaning of the Private Securities Litigation Reform Act of 1995 concerning future business conditions and the outlook for Midway Games Inc. (the "Company") based on currently available information that involves risks and uncertainties. The Company's actual results could differ materially from those anticipated in the forward-looking statement as a result of these risks and uncertainties, including, without limitation, the Company's ability to maintain the scheduling of product introductions as more fully described under "Item 1. Business - Risk Factors" in the Company's Annual Report on Form 10-K for the year ended December 31, 2006, and in any more recent filings made by the Company with the Securities and Exchange Commission. The forward-looking statement speaks only as of the date on which it is made, and the Company undertakes no obligation to update it to reflect events or circumstances after the date on which it is made or to reflect the occurrence of anticipated or unanticipated events or circumstances.



**SEEKING NEW RECRUITS**

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

**Foundation 9 Entertainment**

- BACKBONE ENTERTAINMENT**  
Emeryville, CA  
Vancouver, BC
- the collective**  
Newport Beach, CA
- PIPEWORKS**  
Eugene, OR
- imaginengine™**  
Boston, MA

Need to learn the skills that today's employers are looking for?

**Give us a call.**

- Love checking out the latest video games?
- Have you ever thought about what it would be like to create your own video games?

**You could be a candidate for career training in Game Design.**

If you want to combine your talent for design with today's modern technology, find out more about our **Bachelor of Fine Arts Degree in Visual Communications with a concentration in Game Design**. Contact us today and start studying for a rewarding career in the growing field of Game Design.

**The Academy Offers:**

- Financial aid is available for those who qualify
- Continuous Education Benefits available to alumni
- Focused hands-on career education
- Career Services assistance is available for all students



# MAKE MORE ENEMIES

VFS Student work by Jelmer Boskma

VFS™



[vfs.com/enemies](http://vfs.com/enemies)

SCHOOL OF: COMPUTER ANIMATION > DIGITAL ARTS & DESIGN > ENTERTAINMENT BUSINESS > FILM > GAME DEVELOPMENT > RECORDING ARTS > SHOW PRODUCTION & TOURING



School of  
**Game Development**

School of  
**Computer Animation**

3300 University Boulevard  
Winter Park, FL 32792

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

## ONE OF THE TOP FIVE GAME-DEGREE PROGRAMS

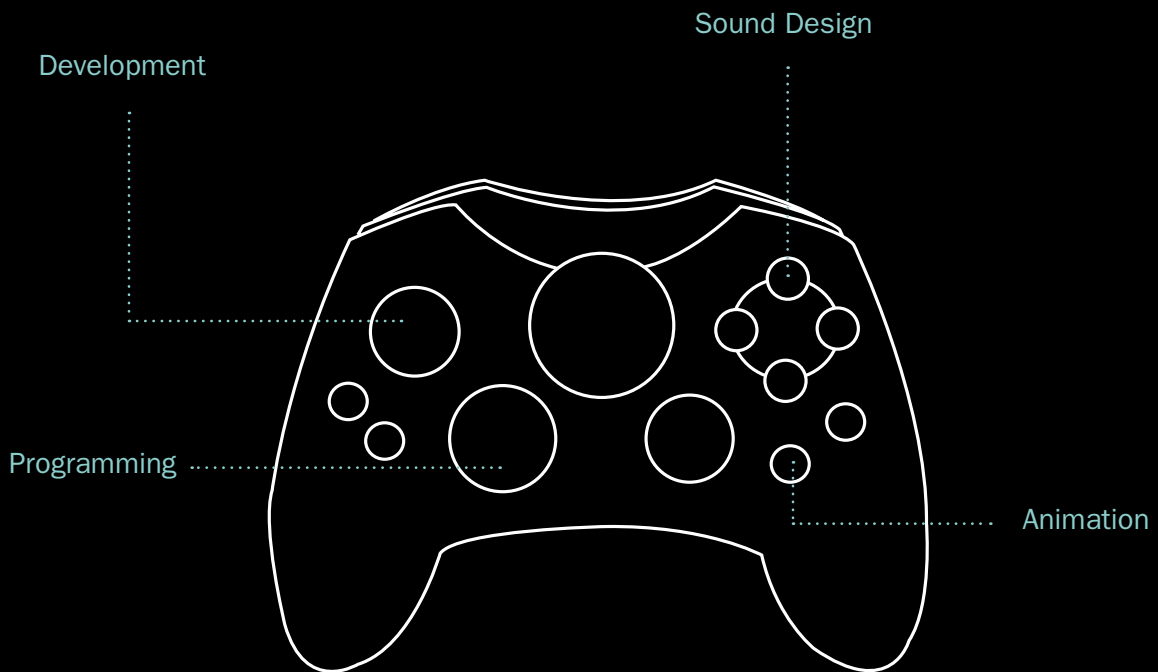
— *Electronic Gaming Monthly*



Student Artwork  
Francisco Cruz

800.226.7625  
[fullsail.com](http://fullsail.com)

GET STARTED



## Say it in game.

Take your ideas to the next level.  
Join Columbia's community of intensely creative minds.

create...  
change

Look for Columbia College's booth (#5607) at  
**Game Developers Conference '07** in San Francisco, March 5-9.

Columbia   
COLLEGE CHICAGO

[GAME.COLUM.EDU](http://GAME.COLUM.EDU)

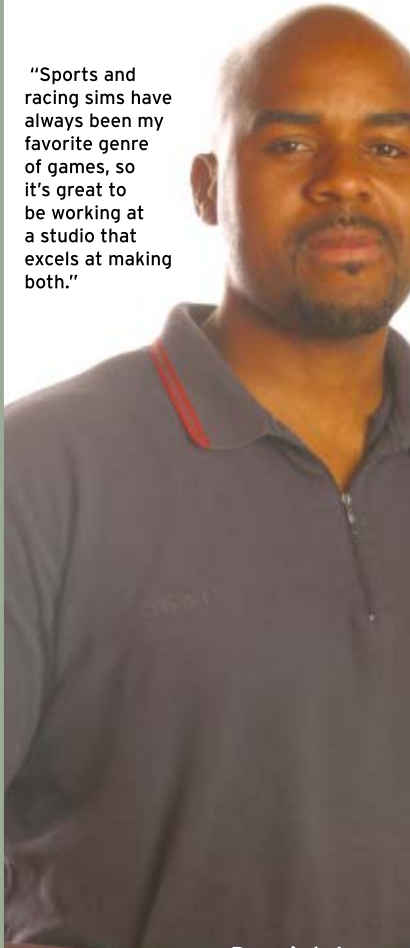
INNOVATION IN THE VISUAL, PERFORMING, MEDIA,  
AND COMMUNICATION ARTS



THE  
**GUILDHALL**  
AT SMU

96% placement

"Sports and racing sims have always been my favorite genre of games, so it's great to be working at a studio that excels at making both."



Derrick Levy  
Guildhall Graduate 2004  
Software Engineer: EA - Tiburon

GRADUATES EMPLOYED BY

- BETHESDA SOFTWORKS
- BLIZZARD ENTERTAINMENT
- EA
- ENSEMBLE STUDIOS
- GEARBOX SOFTWARE
- ID SOFTWARE
- INFINITY WARD
- NCSOFT
- RITUAL ENTERTAINMENT
- OVER 25 MORE...

214.768.9950  
guildhall.smu.edu

ANNOUNCING...  
**GAME CAREER GUIDE.COM**

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

- News and features for students and educators
- Comprehensive listing of schools
- Getting Started section – an invaluable how-to guide
- Message Boards
- Faculty and student research

You can purchase your digital edition of the 2006 **game developer** Game Career Guide online at  
[www.gamecareerguide.com](http://www.gamecareerguide.com)

Get Your **GAME** On.

*A Video Game Design and Development diploma could get you on your way to a great career!*

**International Academy of Design & Technology**  
TORONTO

»» The Bureau of Labor predicts that the growth of software publishing, of which video game design is a part, will expand by 67.9% between 2002 and 2012. This is the fastest-growing industry across all sectors.

*Call now before classes are full!*

**1.888.730.8306**  
[www.design-toronto.com](http://www.design-toronto.com)

CBC Broadcast Centre - 39 John St., Toronto, ON M5V 3G6

# ADVERTISER INDEX

COMPANY NAME	PAGE	COMPANY NAME	PAGE
Anark .....	25	Intel.....	11
Autodesk.....	3	International Academy of Design & Technology... 43, 46	
Center for Digital Imaging Boston Univ. ....	47	Midway Games .....	42
The Collective Inc.....	43	Perforce Software.....	C3
Columbia College Chicago.....	45	RAD Game Tools.....	C4
Elsevier .....	19	Replay Solutions LLC.....	C2
Emergent.....	12	Secret Level .....	41
Emotiv Systems .....	9	The Hartecenter at SMU Guildhall.....	46
Epic Games.....	15	Thomson Course PTR.....	17
Full Sail Real World Education .....	44	University of Advancing Technology .....	47
Google Inc. ....	10	Vancouver Film School .....	44
Havok .....	6		

*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.



**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)**



GET EDUCATED



(pwned)



(pwner)



**Please geek responsibly.**  
You may speak the language,  
but are you geeked?  
Here's a chance to prove it.

- GAME DESIGN
- DIGITAL ANIMATION
- ARTIFICIAL LIFE
- DIGITAL VIDEO
- WEB DESIGN
- COMPUTER FORENSICS
- NETWORK SECURITY
- SOFTWARE ENGINEERING
- TECHNOLOGY MANAGEMENT
- GAME PROGRAMMING

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



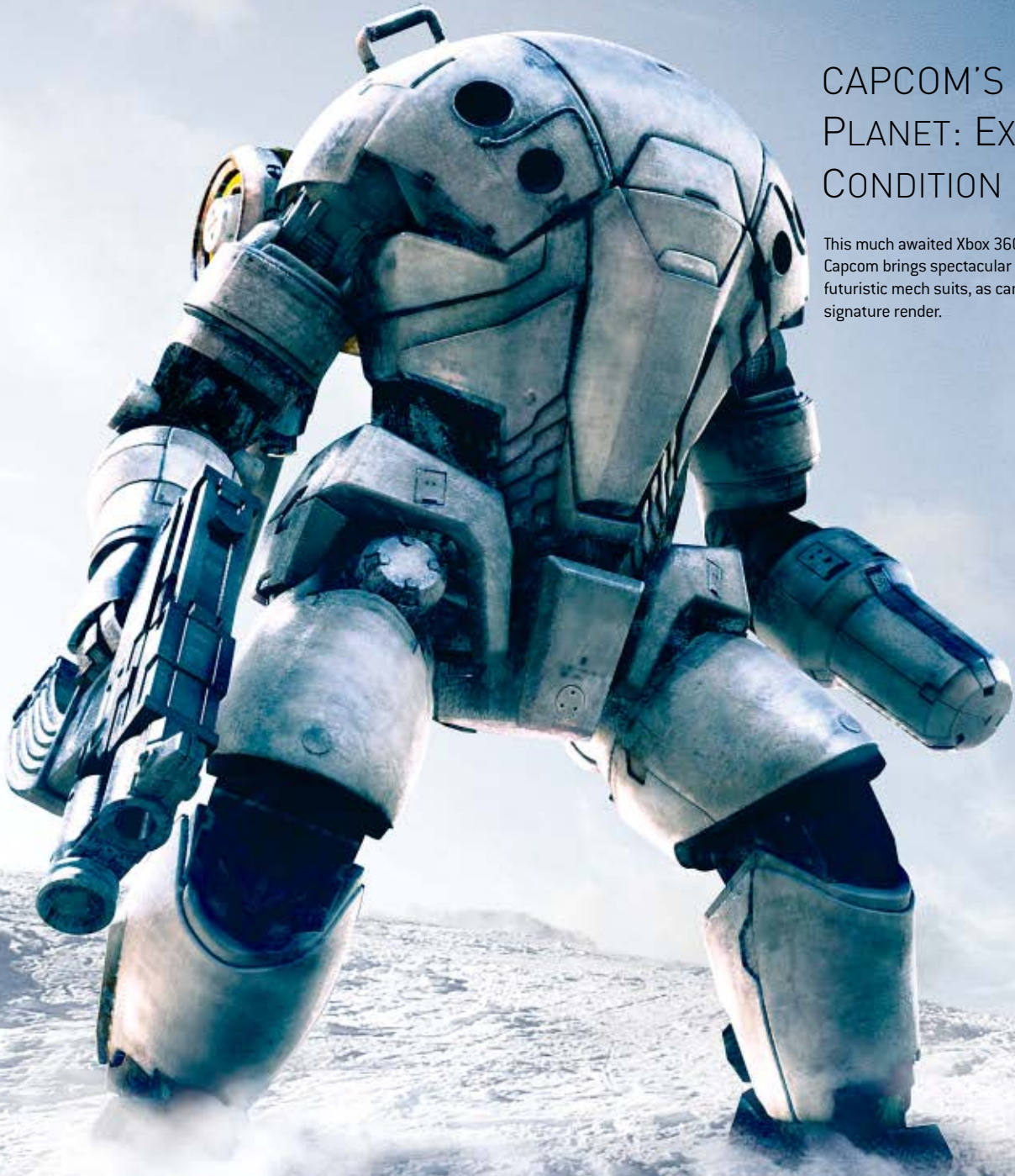
---

## » A THOUSAND WORDS

---

### CAPCOM'S LOST PLANET: EXTREME CONDITION

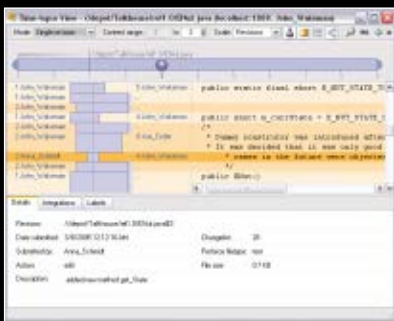
This much awaited Xbox 360 exclusive from Capcom brings spectacular graphics and futuristic mech suits, as can be seen in this signature render.







## Introducing Time-lapse View, a productivity feature of Perforce SCM.



Perforce Time-lapse View

Time-lapse View lets developers see every edit ever made to a file in a dynamic, annotated display. At long last, developers can quickly find answers to questions such as: 'Who wrote this code, and when?' and 'What content got changed, and why?'

Time-lapse View features a graphical timeline that visually recreates the evolution of a file, change by change, in one fluid display. Color gradations mark the aging of file contents, and the display's timeline can be configured to show changes by revision number, date, or changeset number.

Time-lapse View is just one of the many productivity tools that come with the Perforce SCM System.

# Bink takes the grind out of game video!

## The video codec for games

*Bink has shipped in 2,900 games for a reason. It is the standard for video in games - it is faster (up to 4 times faster), it uses less memory (up to 18 MB less), it has a built-in audio codec, it supports every platform, multiple audio tracks, data interleaving, alpha and RLA files (for Z-depth, normal and uv per-pixel data), and much more! Use HD video on Xbox and PS3!*

## It's like having your own codec

*Using Bink is like using a video codec you wrote yourself. You can, for example, play to the screen, to a 3D texture, to a back buffer, to an overlay, to a plain old chunk of memory, or whatever. Bink works the way you want it to.*

## RAD knows games

*We have been selling middleware solutions since 1991. We can solve the problems you face, because we (or our 2,000 previous customers) have had to solve them too!*

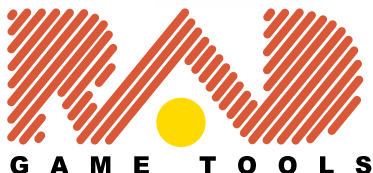
**New!**

**Now for Nintendo Wii, Sony PSP & PLAYSTATION® 3!  
New Japanese Manual!**

For more information:

**425.893.4300**

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



©

**THE BEST IN GAME DEVELOPMENT TECHNOLOGY**



All trademarks mentioned herein are property of their respective owners.

