CMP
United Business Media

PRODUCT REVIEWS KAYDARA MOTIONBUILDER 5.5 ⁘ RONINWORKS BURSTMOUSE

JUNE/JULY 2004

gamedeveloper

THE LEADING GAME INDUSTRY MAGAZINE

E3 2004: PSP AND DS
WHEN HANDHELDS COLLIDE

OUR LITTLE SECRET
OFFSHORE OUTSOURCING

FEEDBACK LOOPS
REAL-WORLD CONTROL FOR GAMES

POSTMORTEM:
THE CONTROL SYSTEM OF
EA SPORTS FIGHT NIGHT 2004

gamedeveloper

JUNE/JULY 2004
VOLUME 11, NUMBER 6

CONTENTS

FEATURES

14 OUR LITTLE SECRET: OFFSHORE OUTSOURCING

Developers don't want to talk about it, and publishers don't want to admit it, but elements of game production are increasingly being sent abroad. Game journalist Dean Takahashi drills down, separating fact from fiction, and discovers sides to the story that make it less cut and dried than you might think.

By Dean Takahashi

18 FEEDBACK LOOPS: IMPLEMENTING REAL-WORLD CONTROL SYSTEMS

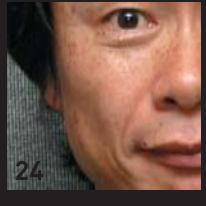
Instead of throwing more algorithms or equations at a movement problem, one lone programmer came up with applying the real-world control mechanics of feedback loops in Zoo Tycoon 2. Read on, and see if it works for your game.

By Terence J. Bordelon

24 DOING MUSHROOMS, MIYAMOTO-STYLE

Many of you reading this sentence were inspired to join the industry because of Shigeru Miyamoto. A much smaller and ever luckier number also gets to work with him. We recently talked with the man himself, along with rising Nintendo stars Eiji Aonuma and Ken'ichi Sugino, and were rewarded with much more than a pixellated princess by the end of the day.

By Jamil Moledina



POSTMORTEM

28 EA Sports Fight Night 2004

Ever notice that boxing games never really felt right? The team at EA Chicago did, and figured out that the disconnect was in the traditional button-pressing control scheme. But how do you work beyond people's expecations? What can you do with control to make a game more fun? EA's breakthrough with the Total Punch Control system might kindle some ideas.

By Kudo Tsunoda

DEPARTMENTS

2 GAME PLAN *By Jamil Moledina*
The More Things Change

4 FEEDBACK LOOP
Our readers respond to Jason Rubin's battlecry and our redesign

6 HEADS UP DISPLAY
The handheld coming out party, and other E3 2004 revelations

12 SKUNK WORKS *By Brian Chambers and Peter Sheerin*
Kaydara Motionbuilder 5.5 and RoninWorks BurstMouse

56 A THOUSAND WORDS
The art of Glyphx Games' Advent Rising

COLUMNS

36 THE INNER PRODUCT *By Jonathon Blow* [PROGRAMMING]
Experiments I'd Like to Work On

40 PIXEL PUSHER *By Steve Theodore* [ART]
Maps and Legends

44 NECESSARY EVIL *By Dan Lee Rogers* [BUSINESS]
The Rapid Retreat of Advanced Royalties

46 AURAL FIXATION *By Alexander Brandon* [SOUND]
Pushing the Envelope

47 GAME SHUI *By Noah Falstein* [DESIGN]
Serious Fun

COVER ART: PHOTOGRAPH OF SHIGERU MIYAMOTO BY JAMIL MOLEDINA, ART DESIGN BY CLIFF SCORSO, EA SPORTS FIGHT NIGHT 2004 IMAGE PRODUCED BY THE EA SPORTS TEAM

GAME PLAN

YOU CAN REACH JAMIL MOLEDINA AT *JMOLEDINA@GDMAG.COM*

THE MORE THINGS CHANGE

WAS IT JUST ME, OR DID A LOT OF THE GAMES AT E3 this year look pretty much the same? There were at least 348 war games, spread all across the 20th century and, as luck would have it, the last couple of years too. Perhaps the similarity of these games, whether they feature trench warfare in World War I or barricades in Fallujah, are a subtle commentary on the fact that war itself doesn't change, people don't change, and war doesn't change anything. Well, that's rather pessimistic; I think I'll go play some videogames. Oh, wait.

But it wasn't just war games—there were 612 racing games, 73 ancient Rome games, 469 gory vigilante games, and 402 gory horror games. To mix things up a bit, genre-bending was out in force, with a pinball RTS, a chess-infused MORTAL KOMBAT game, and a NASCAR RPG. And thankfully, there was outright innovation too, with creative input games such as EYETOY: ANTIGRAV and DONKEY KONGA, the XNA tool suite, and of course the war of the handhelds (see page 6 for a list of our favorite innovations at E3).

However, whether PSP, DS, and XNA are earnestly offered at face value or if they're fancy alphabet soup designed to dazzle and pacify us in light of the quiet period apparently in effect for the next generation of consoles, has yet to be determined. At the same time, this could very well be a combination of both, where the end result may overshadow the impact of the original console hardware (think DVD sales outpacing box office receipts).

There was also a lot of discussion about the vast untapped casual gaming market of soccer moms playing BEJEWELED. While many believe there's substantial headroom open in the existing console market, there are just as many who believe they can convince normal people to spend consistently on games. Microsoft Game Studios general manager Shane Kim and Xbox advanced technology group director Laura Fryer advised me that Microsoft's competency with MSN Zone could give Xbox Live the lead in the casual gaming market when it launches Live Arcade, video chat, and in-depth matchmaking services later this year. Sony Computer Entertainment America's president and COO Kaz Hirai was quite up front about admiring and adapting Apple's iTunes Music Store business model of micropayments to bringing more people and revenue to PlayStation 2 online services. Of course, there's Nintendo's *res ipsa loquitur* approach with gems like WARIOWARE, INC. DS, and the grand experiment of casual online world games. Perhaps it wasn't all the same thing at E3 2004 after all.

HOUSEKEEPING. You may have noticed that this issue covers both June and July. Since this is typically a slow period in our annual cycle, and we have the added task of creating the *2004 Game Developer Career Guide* during the same period, the decision was made to preserve our collective sanity and consolidate two regular issues. The upshot of this is that we'll actually catch up on our e-mails and subscribers will get a one-month extension before they have to renew. If you find yourself looking for something to read in July (or looking for a simple way to deal with people asking you how to get into the industry or how to lateral over to another developer), we expect to have that *Career Guide* out by month's end, and you can always check out the latest news on Gamasutra.com.

FAREWELL JON. This issue marks the final installment of Jonathan Blow's The Inner Product column. His maverick style has provoked more than a few vehement responses, and his avant garde wit has kept a lot of readers engrossed, even if they weren't programmers. What you may not know is that he has been a great friend and resource, helping out with numerous aspects of the magazine and the whole CMP Game Group—and on that level, he's not getting away that easily. In the meantime, his frequent partner in crime Sean Barrett will start a new programming column, bringing his own distinct voice to our pages.

Jamil Moledina
Managing Editor

gamedeveloper
www.gdmag.com

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

EDITORIAL
EDITOR-IN-CHIEF
Dominic Milano *dmilano@gdmag.com*
MANAGING EDITOR
Jamil Moledina *jmoledina@gdmag.com*
DEPARTMENTS EDITOR
Kenneth Wong *kwong@gdmag.com*
PRODUCT REVIEW EDITOR
Peter Sheerin *psheerin@gdmag.com*
ASSISTANT EDITOR
Jill Duffy *jduffy@gdmag.com*
ART DIRECTOR
Cliff Scorso *cscorso@gdmag.com*
CONTRIBUTING EDITORS
Jonathan Blow *jblow@gdmag.com*
Noah Falstein *nfalstein@gdmag.com*
Steve Theodore *stheodore@gdmag.com*
ADVISORY BOARD
Hal Barwood Designer-at-Large
Ellen Guon Beeman Monolith
Andy Gavin Naughty Dog
Joby Otero Luxoflux
Dave Pottinger Ensemble Studios
George Sanger Big Fat Inc.
Harvey Smith Ion Storm
Paul Steed Microsoft

ADVERTISING SALES
GROUP ASSOCIATE PUBLISHER
Michele Sweeney e: *msweeney@cmp.com* t: 415.947.6217
SENIOR ACCOUNT MANAGER, EASTERN REGION & EUROPE
Afton Thatcher e: *athatcher@cmp.com* t: 404.658-1415
ACCOUNT MANAGER, NORTHERN CALIFORNIA & MIDWEST
Susan Kirby e: *skirby@cmp.com* t: 415.947.6226
ACCOUNT MANAGER, WESTERN REGION & ASIA
Craig Perreault e: *cperreault@cmp.com* t: 415.947.6223
ACCOUNT MANAGER, TARGET PAVILION, EDUCATION, & RECRUITMENT
Aaron Murawski e: *amurawski@cmp.com* t: 415.947.6227

ADVERTISING PRODUCTION
ADVERTISING PRODUCTION COORDINATOR Kevin Chanel
REPRINTS Julie Rapp e: *jrapp@cmp.com* t: 510.985.1954

CMP GAME GROUP MARKETING
DIRECTOR OF MARKETING Michele Maguire
SENIOR MARCOM MANAGER Jennifer McLean
MARKETING COORDINATOR Scott Lyon

CMP GAME GROUP
VP, GROUP PUBLISHER, APPLIED TECHNOLOGIES Philip Chapnick
EXECUTIVE PRODUCER, GAMASUTRA Alex Dunne
GDC PROGRAM MANAGER Susan Marshall

CIRCULATION
BPA INTERNATIONAL
Game Developer is BPA approved
CIRCULATION DIRECTOR Kevin Regan
CIRCULATION MANAGER Peter Birmingham
ASST. CIRCULATION MANAGER Lisa Oddo
CIRCULATION COORDINATOR Jessica Ward

SUBSCRIPTION SERVICES
FOR INFORMATION, ORDER QUESTIONS, AND ADDRESS CHANGES
t: 800.250.2429 or 847.763.59581 f: 847.763.9606
e: *gamedeveloper@halldata.com*

INTERNATIONAL LICENSING INFORMATION
Mario Salinas
e: *msalinas@cmp.com* t: 650.513.4234 f: 650.513.4482

EDITORIAL FEEDBACK
editors@gdmag.com

CMP MEDIA MANAGEMENT
PRESIDENT & CEO Gary Marshall
EXECUTIVE VICE PRESIDENT & CFO John Day
EXECUTIVE VICE PRESIDENT & COO Steve Weitzner
EXECUTIVE VICE PRESIDENT, CORPORATE SALES & MARKETING Jeff Patterson
CHIEF INFORMATION OFFICER Mike Mikos
PRESIDENT, TECHNOLOGY SOLUTIONS Robert Faletra
PRESIDENT, CMP HEALTHCARE MEDIA Vicki Masseria
SENIOR VICE PRESIDENT, OPERATIONS Bill Amstutz
SENIOR VICE PRESIDENT, HUMAN RESOURCES Leah Landro
VP & GENERAL COUNSEL Sandra Grayson
VP, GROUP PUBLISHER INFORMATIONWEEK MEDIA NETWORK Michael Friedenberg
VP, GROUP PUBLISHER ELECTRONICS Paul Miller
VP, GROUP PUBLISHER ENTERPRISE ARCHITECTURE GROUP Fritz Nelson
VP, GROUP PUBLISHER SOFTWARE DEVELOPMENT MEDIA Peter Westerman
VP & DIRECTOR OF CMP INTEGRATED MARKETING SOLUTIONS Joseph Braue
CORPORATE DIRECTOR, AUDIENCE DEVELOPMENT Shannon Aronson
CORPORATE DIRECTOR, AUDIENCE DEVELOPMENT Michael Zane
CORPORATE DIRECTOR, PUBLISHING SERVICES Marie Myers

CMP United Business Media
CMP Game Group

FEEDBACK LOOP

E-MAIL YOUR FEEDBACK TO EDITORS@GDMAG.COM, OR WRITE US AT GAME DEVELOPER, 600 HARRISON ST., SAN FRANCISCO, CA 94107

JASON RUBIN STIRS THE MASSES

I HAVE SOME THOUGHTS TO SHARE on the interview with Jason Rubin ("The Anti-Communist Manifesto," May 2004). Are celebrities good for business? Absolutely. People relate to people, not corporations. The phenomenon of celebrities works as a sales tool because we are all fascinated by people whose looks, creativity, lifestyle, talent, intelligence, achievements, or whatever are vastly superior to or radically different from our own. We're drawn to such individuals by an amalgamated elixir of curiosity, envy, excitement, hero-worship, fantasy, competitiveness, and mimicry. An observation of this phenomenon in other industries, not just entertainment, suggests that celebrities are inevitable—but they are only profitable for those companies that effectively exploit them. And while some charismatic overachievers in our industry simply can't be kept out of the relevant public consciousness, there is no doubt that effective star-making machinery could add an exponential boost to their profiles, and subsequently to the profits of those companies whose products are tied to such rising stars.

Chance Thomas, Composer & President of HUGEsound

GAME DEVELOPER'S MAKEOVER

REGARDING THE NEW CONTENT and format, I am extremely impressed. Expanding *Game Developer*'s focus to include the business side of the industry is a welcome, long-overdue development, and the other new features promise to be very worthwhile as well. Anything that gives our best people the notoriety they deserve is a good thing!

François Dominic Laramée, Writer and Game Designer

I JUST RECEIVED THE MAY 2004 ISSUE of *Game Developer* magazine. I generally like the layout of the magazine, but the change to a smaller typeface for the articles is unwelcome. Would you please consider a return to the previous size in future issues, or at least something more in line with other magazines?

This smaller size is particularly bad in the case of colored backgrounds, blue text on white, and white text on black.

Thank you.

Chris Babcock, Icarus Studios, LLC

I LOVE THE NEW FORMAT AND CONTENT of *Game Developer* except for one thing: the size of the type is too small.

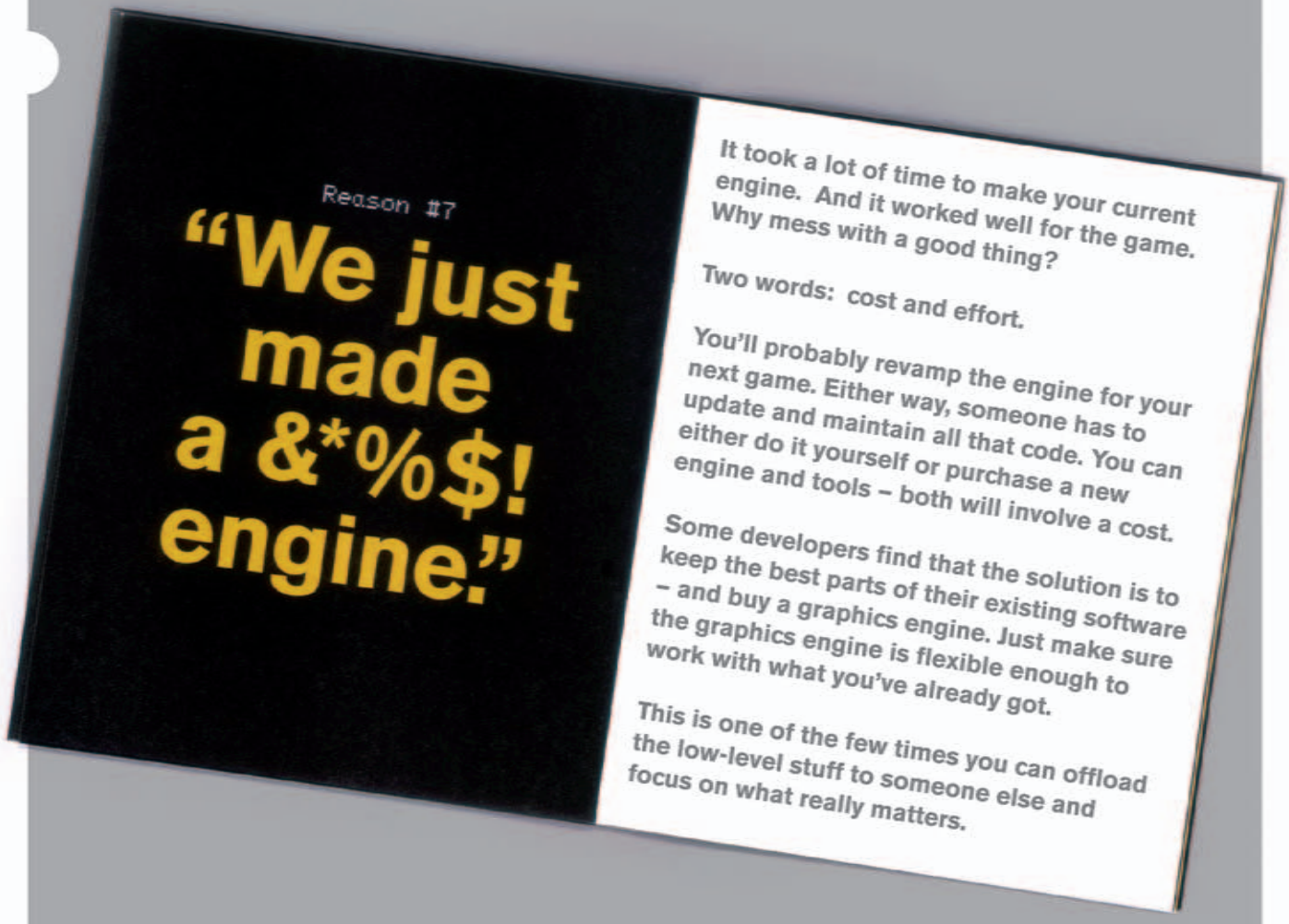
I work in the design biz now (after a previous stint in the game industry), so a large part of my job is spent making type as readable as possible.

Because of *Game Developer*'s too-small type size, reading is harder on the eyes, causing fatigue and making extended reading more of a chore than the pleasure it should be.

Adam Snetman, Designer, Carbone Smolan Agency

Managing editor Jamil Moledina replies: *Thanks for your feedback. You're right, the typeface was too small in the May issue, which made the reversed text difficult to see. This was one of the things we noticed too when we saw the issue in print, and we've bumped the font up to address that. We've also made other subtle enhancements throughout the magazine, reflecting our own internal postmortem on the redesign, and numerous specific suggestions from our readers. Please keep them coming, as we view this redesign as an iterative process.*

Why NOT to buy a commercial graphics engine
(like, say, Gamebryo)



IBM and Sony Announce Cell Workstation

AT E3, IBM AND SONY ANNOUNCED A JOINT EFFORT to create a new content-creation workstation based on IBM's code-named "Cell" microprocessor. According to IBM's Dr. John E. Kelly (SVP of the IBM Systems and Technology Group), this processor is designed to deliver "scalable, supercomputer-like performance to the media, entertainment and videogame industries"—providing a development platform closely aligned with Sony's future entertainment platforms, including the PlayStation 3, which will likely use the Cell chips.

The Cell chip has been under development by partners IBM, Toshiba, and Sony since 1992, with Sony driving the requirements and each partner getting manufacturing rights. Key design elements are a 50nm, Silicon on Insulator (SOI), massively integrated System on Chip (SoC) architecture designed to combine extreme performance with low power usage. IBM will be developing the operating system while Sony develops the content creation tools. The workstation is slated for an ambitious fourth-quarter release (the original target for prototype Cell chip production was 2005), and given the enormous investment in the project, it seems clear that Sony plans on having a content creation and playback architecture that is almost completely severed from the existing Windows and Macintosh-based environments.

—*Peter Sheerin*

MOST INNOVATIVE GAMES:

DONKEY KONG JUNGLE BEAT
(GameCube, Nintendo)
A side-scrolling/action/bongo-drumming game, using a physical interface for locomotion.

KATAMARI DAMASHII
(PS2, Namco)
Play as a tiny ball that picks up objects as it rolls around—unorthodox gameplay that mixes humor and strategy.

ODAMA
(GameCube, Vivarium)
Military strategy meets pinball. Bouce that ball around feudal Japan.

RIBBIT KING
(GameCube, Bandai)
Play as a cartoon character entering the Frolf (Frog Golf) Tournament to gain control of Super Ribbinite power.

XTANGO: SHUFFLING ROSE
(Xbox, Binary Craft)
A dual-player collaborative dance-simulation game where two players play against AI opponents in virtual competitions.

Nintendo DS running Metroid Prime: Hunters

WHEN HANDHELDS COLLIDE

SONY AND NINTENDO DREW THE LION'S SHARE of interest at E3 this year by showing off their new handheld devices for the first time. Lines wrapped around their booths for a glimpse of the next generation devices running 3D game demos. Sony's PSP runs on a custom 333MHz CPU, and features a 4.3-inch 16:9 widescreen TFT LCD (480x272 resolution), IEEE 802.11b wireless connectivity, a proprietary optical drive format called UMD (Universal Media Disc, 1.8GB capacity), and the ability to play back both movies and music. The Nintendo DS runs on two processors, an ARM9 and an ARM7, and features two 3-inch screens, a touch panel, a microphone, both IEEE 802.11b and a proprietary wireless communication protocol, a new compact card drive (the cards currently top out at 1Gb), and a Game Boy Advance port for backwards compatibility (for more on the DS, see our interview with Shigeru Miyamoto on page 24). Both devices are scheduled for release by the end of the year in Japan, with the DS arriving stateside this year too, while the PSP will arrive here by the end of Q1 2005. Neither company announced a price, although speculation on that was a leading discussion point at the show. Over 100 developers announced support for both systems, including EA, Sega, THQ,

CONTINUED ON PG 55

BREAKING OUT OF THE

PHANTOM ZONE

INFINIUM'S PHANTOM GAME CONSOLE IS A PHANTOM no more. After much industry speculation as to its possible vaporous nature, Infinium Labs' Phantom was unveiled at e-focus, a multi-company PR fest held on the eve of E3's show floor's official opening. There on a tabletop display sat a sample unit, turned on and waiting to be fondled. Sadly, the console wasn't connected to the Internet, so all we could see in that preview was the UI. Happily, the folks staffing the exhibit invited any and all to drop by the Infinium booth to spend some time playing with the real deal.

And play with the Phantom we did. A few details: the unit is an attractive white casing with black trim and cool blue power-on lighting. The keyboard is designed to sit on a gamer's lap and is tilted with a small shelf beneath it on which rests a mouse. And the keyboard can swivel to accommodate righties and lefties. Slick.

It had been rumored that the Phantom would be able to play any game for any platform—an ambition that was nowhere in sight at the unveiling. Instead, Infinium's goal is to support PC-based games of all sorts through their subscription-based business model in which families would pay $29.95 per month over a two-year contract to rent the console and access via download any game offered by the service. Infinium also plans to offer an a la carte service, but details on how it would work are still being hammered out.

The games offered in the demo were a handful of casual games and Unreal Tournamemt 2004. According to an Infinium spokesperson, the complete list will be announced some time in August. With Kevin Bachus (formerly of Microsoft and Capital Entertainment Group) as president and COO and Kathy Schoback (chairperson emeritas of IGDA and former director of product operations at Eidos) onboard as VP of content strategy, Infinium is poised to make a run at lining up as many titles for the Phantom Gaming Service as possible. *—Dominic Milano*

Sony's PSP, image simulated

t

"The PSP is exciting because this marks the first time a handheld system is aimed at an older demographic (20+). We're also looking forward to the DS, and will be supporting it as well."

—Chip Lange, VP of Marketing for EA Sports and EA Games Nations

REVVED UP 3D TOOLS

BOTH ALIAS MAYA and Softimage XSI have been revised. Maya 6.0 and XSI 4.0 continue to gain workflow enhancements—tweaks to their existing feature sets that enhance usability, flexibility, and customizability. Among the many new features in Maya 6: the Trax NLA editor has a new UI, the ability to mute and solo animation tracks, and enhanced support for animating to sound clips; integration with Adobe Photoshop; improved (Alias claims "seamless") integration with mental ray; the ability to apply deformers to particles; the ability to apply Paint Effects on polygons; support for PNG and DDS file formats; and a plethora of new modeling tools. Also, TDs and programmers take note: Alias is holding a Maya API Developers Conference in Los Angeles on June 20, 2004. www.alias.com.

New to XSI 4.0: integration with Avid's Digital Nonlinear Accelerators (DNA); 2D raster and vector paint; Alienbrain Studio integration; new Rigid Body Dynamics; mental ray 3.3; new mocap library; enhanced audio animation tools; new UI customizability; enhanced UV unwrapping; enhanced poly reduction tools; a Character Development Kit; and lots more, as well as new versions and price points—XSI Foundation ($1,995), Essentials ($3,995), and Advanced ($8,995). A free version, EXP, which only outputs to the HALF-LIFE 2 engine's file format, is also available for download. www.softimage.com.

—Dominic Milano

»EA and Xbox Live Find Common Ground

ELECTRONIC ARTS ANNOUNCED AT E3 ITS SUPPORT FOR XBOX LIVE. UP TO that point, EA's hesitancy regarding Microsoft's online initiative had been perceived as a major impediment to the growth of the service. According to published reports, EA had refused to sign on in the past, citing Microsoft's control of subscription revenue and its requirement that games run on Microsoft's servers. According to Chip Lange, VP of Marketing for EA Sports and EA Games Nations, "EA will have the ability to host its own services that extend the Xbox Live experience within our games. All the benefits Xbox Live provides such as a single Gamertag, Friends lists, and more will be part of the online experience on Xbox for EA games." Gamers will continue to get a single bill from Microsoft for now. The announcement came at Microsoft's press conference at the start of E3, with appearances by sports celebrities such as the champ himself, Muhammad Ali, to underscore its impact. *—Jamil Moledina*

LAWMAKERS IN A TIZZY OVER PARENTS' RESPONSIBILITY

WHAT'S MORE DAMAGING TO A 14-year-old: bashing skulls and hijacking cop cars virtually; or smoking cigarettes?

One California legislator, Assemblymember Leland Yee (D-San Francisco), says interactive, albeit simulated, violence in games such as GRAND THEFT AUTO: VICE CITY and POSTAL 2, harms children to a greater extent than tobacco—and the law must protect these children.

In Manitoba, a similar battle between the gaming industry and lawmakers has emerged, questioning whether the industry or the government is responsible for rating videogames.

And in Washington, D.C., a proposed bill would allow parents to sue purveyors of material harmful to children, including videogames, for up to $10,000, according to the bill's author, Rep. Duncan Hunter (R-Calif.).

ESRB'S CURRENT SYSTEM

To rate a new game, the Entertainment Software Rating Board employs and trains a group of at least three raters who are unaffiliated with the computer and videogame industry. The publisher submits a completed questionnaire revealing the game's contents along with video footage of "the most extreme content and an accurate representation of the context and product as a whole," according to the ESRB. The raters independently rate the game and, if a consensus is reached, the ESRB cross-references the gameplay and final packaging just before the commercial release.

The system is not all too different from the movie rating system established by the Motion Picture Association of America in 1968, which is also a voluntary system that is upheld through voluntary policies on behalf of the National Association of Theatre Owners.

WHAT'S THE HUBBUB?

Why doesn't the voluntary ESRB rating system function with the same credibility as the MPAA's?

CONTINUED ON PG 55

>> first person

IT'S UNDERSTANDABLE THAT AS DEVELOPMENT costs increase, there will be less risk-taking. But we need to continue to stoke the fires of the independent developer community so that new ideas are continually brought into the mainstream. Otherwise we risk the possibility long-term of players getting bored with what we create.

—Mark DeLoura, manager of Developer Relations, Sony Computer Entertainment America

To the members of the California Assembly Arts & Public Safety Committees:

I'm a computer and videogame developer. No, not some irresponsible post-adolescent just out of college with baggy pants and a baseball cap turned backwards on my head, but a fully-qualified adult who is concerned about freedom of expression for interactive entertainment.

In my view, all the arts and media should be treated alike, with full respect for First Amendment rights. Nor do I subscribe to the idea that human character is so simple and malleable that mere influence from any media source or art form is capable of overwhelming good sense.

Does that mean we should peddle mature material to children? Of course not. And just as Hollywood has developed a rating system to alert consumers, so have we, the ESRB rating system. (Be it known, although many children play our games, they are not impulse buys at $40, and the vast majority are purchased by adults.)

AB 1792 and AB 1793 are bills designed to attack not only video and computer games, but the First Amendment that protects their legitimate expression.

One may ask, when the subject matter is often lurid or fantastical, and the play often involves simulated violence, what artistic worth do games represent to our customers? There's no simple answer to this deep question--except to say, exactly the same values all the other arts, sports activities, and theme parks supply: entertainment, fun, accomplishment, a sense of personal growth in undergoing a challenging experience. If my claims of positive value don't satisfy, then consider: at worst, can simulated violence in a game be worse for anyone's psyche than the sham violence depicted in a movie or the real violence endured on a football field? This I doubt, and so should you.

I urge you to vote against both these obnoxious bills.

Personal letter to California legislators from Hal Barwood, Finite Arts

Wireless 3D Underway

THE MOBILE STATION MODEM 7000 CHIPSET, the wireless 3D enabler expected from the union between Qualcomm and ATI, is currently still in incubation, set to become available to OEMs in 2005. But the MSM 6000 and 6100 chipsets, the interim technologies, are already in the market, eliminating some of the hurdles in porting popular 3D PC and console titles to handheld devices. At E3, Sprint PCS began previewing handsets equipped with the MSM 6100 chipsets. The phones—and the games—will be available to Sprint customers by Q3 of 2004.

Mike Yeun, director of BREW developer relations at Qualcomm, offers his views to help developers navigate the climate change in the wireless terrain: "The big players, such as Bandai and Namco, will definitely start to offer high-end 3D titles to their wireless and mobile clients, because that shows they're on the cutting edge. But anyone who focuses on 3D to the detriment of 2D does so at his or her own peril. This is not a sign that suddenly all the wireless and mobile games will become 3D; 2D titles will still be around for a long time—they're needed to serve the casual gamers in the wireless market." *—Kenneth Wong*

IGDA Focuses Its Efforts

LAST SUMMER, THEN-CHAIRPERSON KATHY Schoback told IGDA members that the IGDA was trying to accomplish more than its resources could support. Based on feedback from IGDA members, the board focused the association on two areas: community and advocacy.

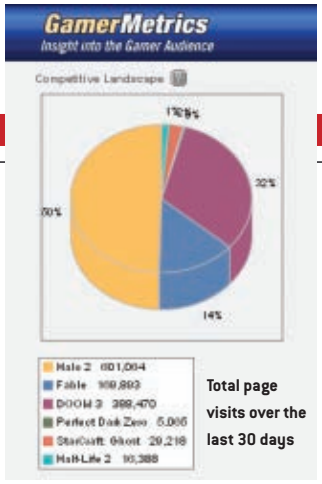
To build community, the IGDA assigned more resources and attention to the 70+ chapters around the world and to its various special interest groups. On the advocacy front, the board defined five criteria which now serve as a litmus test for subjects that the association will advocate. Advocacy topics must:

1) Exceed the scope of any one developer or company's ability to resolve;
2) Require a dialogue with others;
3) Affect the worldwide game development community;
4) Benefit the development community as a whole, rather than any single individual or company directly;
5) Be neutral toward any particular skill, product, or method of game development.

In terms of community, the association held its first-ever members-only party at the Figueroa Hotel in Los Angeles during E3, which drew more than 300 people. On the advocacy front, the IGDA took action this year in Washington and California, in response to proposed legislation that would censor games. The Washington bill was dealt a preliminary injunction. In California, two bills were introduced, as referenced in "Lawmakers in a Tizzy" on page 8 and Hal Barwood's letter above. One of them was blocked and then withdrawn, and another was blocked and is currently being revised.

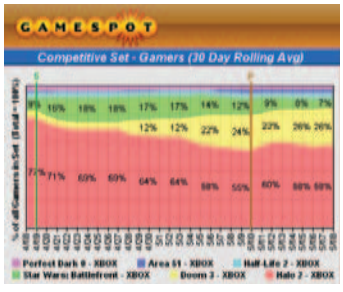
The IGDA also advocates issues within the game development community. It released a "Quality of Life" white paper in March that deals with issues like employee burnout and turnover at game companies and suggests various "best practices" to improve the work environment at studios. The white paper, as well as a full run-down of the association's current efforts, can be found at www.igda.org. *—Alex Dunne*

Show Me the Numbers



Total page visits over the last 30 days

WITH GAME BUDGETS JUMPING from about $1 million in 2001 to $10 to $20 million this year, publishers have to study their numbers. Which of their titles are doing well and which aren't? Which titles are competing with theirs for the same consumers' attention and disposable income? How effective are their marketing efforts? How enthusiastic is the game community about their upcoming titles? How many copies should they produce? Considering the millions of dollars and the careers at stake, no sane executive would just take a wild guess. One of the ways to get reliable data may be through GamerMetrics, a new customer-intelligence gathering tool from IGN/GameSpy.



IGN.com, operated by IGN/GameSpy, touts 15 million unique site visitors a month, representing a sizable sample of the overall gamer population. GamerMetrics gathers statistical data in various ways: by observing how IGN users behave toward certain titles; by analyzing the wish lists, watch lists, and purchase histories disclosed by IGN users; and by calculating the popularity of certain titles based on the buzz, purchase intent, and awareness aggregated through IGN.com and GameStats.com.

A similar application has been available in beta since last year from GameSpot, a property of CNET. GameSpot Trax 2.0 is the second incarnation of CNET's return-on-investment calculation tool. It gathers its data from visitors of GameSpot.com, Gamerankings.com, GameFAQs.com, and Download.com's game section, all operated by CNET.

GamerMetrics allows you to build a custom sample—based on age range, preferred platforms, games purchased, and so on—to analyze a specific sector of the game consumer population. GameSpot Trax 2.0 features graphical analysis of expected retail sales during the first month, based on interest, awareness, and tracking expressed by site visitors. It also lets you visualize the varying degrees of interest expressed by consumers for a certain title, broken down by geographic regions (as deep as specific zip codes).

The strength of these applications is their sophisticated comparative reporting capabilities. They let you see, for instance, not just how popular a title is but how popular it is in relation to the most popular games in the same genre or other genres, in relation to those on the same platform or other platforms, and so on. With options to conduct searches or display results about a title based on release date, media coverage, message board activity, and more, these data gathering and analysis tools can arm an executive with what he or she needs to launch a highly specific PR campaign to attract the right demographics.

As an example, we queried user interest in HALO 2 on both services, with some results excerpted to the left and above. —*Kenneth Wong*

WHAT HAVE YOU PLAYED LATELY, DOC?

GOING UNDER THE KNIFE, YOU WOULDN'T THINK of asking your surgeon if he or she has ever played any videogames, but maybe you should. According to researchers at the New York Beth Israel Medical Center and The National Institute on Media and the Family, surgeons who currently play three or more hours of videogames a week have a 37 percent reduction in errors when performing laparoscopic surgery. Furthermore, they also accomplish the surgical task 27 percent faster than their counterparts who don't play videogames.

A laparoscopic procedure involves using a slender tubular endoscope, inserted through a small incision in the abdominal wall, to examine stomach and pelvic cavities. James Rosser Jr. M.D., who oversees minimally invasive surgeries at Beth Israel, couldn't help noticing how similar the external laparoscopic controllers are to Nintendo controllers. To study the correlation between videogame skills and laparoscopic skills, Rosser and a team of researchers had 33 subjects play SUPER MONKEY BALL (Sega), STAR WARS RACER REVENGE (LucasArts), and SILENT SCOPE (Konami) before undertaking surgical tasks. The findings show that subjects who have previously played videogames score higher and err less than those who haven't.

"The new generations of professionals are more adept at navigating virtual interfaces than the previous ones," Rosser observes. "For many of them, a videogame is their first encounter with a computer. If we use videogames as

CONTINUED ON PG 55

Diversifying the GPU Portfolio

GRAPHICS CARD VENDORS Nvidia and ATI are all about moving polys and textures around on screen. Talking with their developer and ISV relations folks usually means talking about HLSL, DX 8 vs. DX 9 vs. OGL, normal mapping—all the usual graphics accelerator stuff you'd expect. But just as Moore's Law-driven better, faster, cheaper CPUs have made sound cards a thing of the past, graphics card vendors need to find new ways to ensure the life span of their GPU-based businesses.

ATI's dev. rel. manager, Michael Smith, envisions a future in which the GPU is used to accelerate AI and physics. And with Microsoft's next-gen OS, Longhorn (due out before the next millennium, give or take a century), expected sys requirements being 46GHz and 2GB RAM, that vision is well timed.

—*Dominic Milano*

CALENDAR

NYU CAMP: INTENSIVE VIDEO GAME CREATION
Center for Advanced Digital Applications
New York, N.Y.
July 6–August 6, 2004
Cost: $5,145
www.scps.nyu.edu

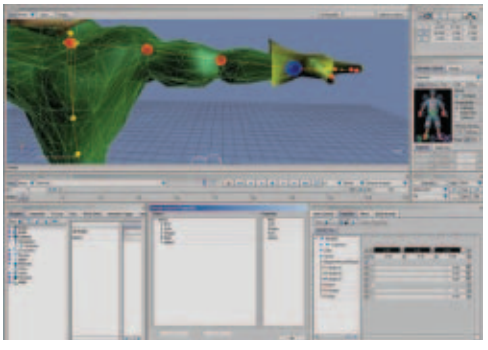
SIGGRAPH 2004
Los Angeles Convention Center
Los Angeles, Calif.
August 8–12, 2004
Cost: $75–$1,100
www.siggraph.org

EDINBURGH INTERNATIONAL GAMES FESTIVAL
Edinburgh International Conference Center
Edinburgh, UK
August 8–22, 2004
Cost: £193.88–£229.13
www.eigf.co.uk

CLASSIC GAMING EXPO
San Jose McEnery Convention Center
San Jose, Calif.
August 21–22, 2004
Cost: $25–$35
www.cgexpo.com

MOTIONBUILDER 5.5

BRIAN CHAMBERS



The custom properties window is very straightforward and easily adjustable, dramatically enhancing workflow.

MOTIONBUILDER IS GETTING CLOSER TO being the tool every animator should have. If you've ever dealt with having to get mocap onto your character just right and you work in the reality of tight schedules, then Motionbuilder is most likely not new to you. Known as Filmbox previously, Motionbuilder is steadily gaining more useful and functional features with each upgrade. Motionbuilder comes in two versions—Standard and Professional. You can find the comparison of features on Kaydara's web site, but if you're getting down and dirty with raw mocap data, then you really have no choice but to go with the Pro version. Many of the changes are simple enhancements or added functionalities, but a few new gems have also emerged.

IMPROVED WORK FLOW. Scripting is now available in the Pro version, though it only supports the Python language. It's not a wonder button, as you can't record your movements and auto-generate scripts as you would in Maya; nonetheless, it's there and offers solid functionality, even though there is no IDE and you are left to create scripts with your favorite text or code editor. I approached this with an open mind, and with a little reading on the language basics, I was up and running in no time. This gives animators the ability to customize their workflow, to take time-consuming, menial tasks and automate them with one simple command. I'm not sure if Python was the absolute best choice of languages, but I'm sure Kaydara chose it for a reason.

Asset management has stepped up a notch in this version, with Kaydara's .FBX interchange format now being supported by NXN Alienbrain. They actually integrated it to the point that anyone can view an .FBX file through the Alienbrain preview window, rotate the scene, view it through various cameras, and so on. This feature provides great bang-for-the-buck when it comes to organizing what's been done for the non-animator. Any member of the team can now go into Alienbrain, browse through the new files you've checked in, and examine them out from any perspective.

ANIMATION FLOURISHES. With 5.5 you gain the ability to add your own properties to any object in your scene, then modify and animate those properties just as you would any other. In the object properties window, there's a custom properties button, giving you access to the different assignable properties, which include bool, color, integer, real, and vector. These property types give you the ability to add check boxes, value fields and sliders, through which you can trigger events, and customize the interface, for example. This is a big step—by streamlining and customizing your own personal workflow in Motionbuilder, you can bring the properties you'll constantly be playing with to the front—something that ultimately saves you valuable time.

Smoothing filters can sometimes, depending on how you use them, soften your curves to the point that most of the subtleties inherent in the original mocap data are lost. The new Butterworth filter smoothes your curves with a more intelligent approach, leaving the curve's minimum and maximum values alone and lightly smoothing out everything in between. I pushed it through some tests using mocap data that had various degrees of noise, and found it provided descent results in every scenario. This is definitely a low-pass filter, as Kaydara calls it, taking out only the small noise and leaving any big jumps alone, as these should be left to other means of cleaning.

Additional features in 5.5 include glove-axis rotation for fingers, which allows you to bend each finger along a single axis when captured with a mocap glove; facial shape animation, through triggers, for creating expressions; and the ability to translate an actor's pivot points in the advanced properties window.

RENDERING ENHANCEMENTS. Kaydara's given us some new things with regard to rendering this time around. First off, you can now choose "off-screen rendering," and continue to work while a render completes and not worry about affecting the image. I have jacked up more than my share of renders due to leaving the mouse on-screen, pop-up windows, and other silly things—not anymore with this feature. They've also implemented audio into their rendering with .AVI and .MOV files. Now, when you're building cutscenes, for example, you can import multiple shots with the audio and render them out as one piece for approval. Gone are the days of bringing

MOTIONBUILDER 5.5

PRETTY SLICK

STATS

Kaydara Inc.
4428 Boulevard St-Laurent, Suite 300
Montreal, Quebec, Canada H2W 1Z5
(514) 842-8446
Fax: (514) 842-4239
www.kaydara.com

PRICE

Standard version $995.00—upgrades from previous versions are $250; Professional version $3495.00—includes 12 months of support and upgrades

SYSTEM REQUIREMENTS

Software: Windows 2000 SP 2, Windows XP or Mac OS X 10.2 and newer. Hardware: 256MB RAM or more, 300MB disk space or more, graphics card with OpenGL support and 16MB RAM or more

PROS

1. The new scripting feature lets you enhance your workflow.
2. Easily view .FBX files in Alienbrain without having to open them or have Motionbuilder installed.
3. It remains the most solid real-time play back of any animation package I've used.

CONS

1. Scripting uses the Python language, which isn't as commonly known as others.
2. Kaydara's .FBX interchange format is considerably larger than most other formats.
3. The Standard version seems inadequate for any type of practical production.

OUR RATING SYSTEM :

DA BOMB | PRETTY SLICK | SLICK | SO-SO | LAME

everything into an second package and re-rendering. The audio also has adjustable properties such as sample rate, stereo or mono, and bit-size. Kaydara's rendering has always been pretty nice, and these new features step it up a notch.

One other new feature that significantly improves the real-time rendering quality is the .FX-shader support. This requires a Nvidia GeForce 3 or better graphics card. With a .FX-shader loaded, you can control the shaders parameters in the shader settings window (color, specularity, and so on), without having to import the model into Max or Maya to see the effects. Two minor features I like are the configurable grid and a camera turntable icon. The latter feature appears in the camera view that you're manipulating, and gives a circular view of the camera's rotations.

OTHER STUFF. You can now cut, copy, and paste individual relations constraints, or entire constraint groups, in the relations pane. This is a very basic feature to add, but a welcome one. The relations pane in Motionbuilder can get very large and complex depending on what you are doing. The ability to quickly duplicate large constraint groups will save time. The addition of import and export filters for .DXF, .OBJ, and .3DS model formats is overdue, but obviously helpful.

BOTTOM LINE. Kaydara has gone from focusing on making mocap data as clean as possible to making the animator as comfortable as possible. Is Motionbuilder flawless? Not at all (and I can't think of any software that is), but Kaydara was steadily making progress towards a tool that animators can comfortably work within. I've been using Motionbuilder (and its predecessor, Filmbox) almost everyday since 1998, so it's valid to think that I may be biased. However, I've personally seen the software come a long way and can honestly say 5.5 is a definite improvement over previous versions. I'm not suggesting that you set Max or Maya aside, but if Kaydara continues on this path, you may find that Motionbuilder eventually suits your needs better.

If you're familiar with any major 3D package currently on the market, you can master Motionbuilder with only a slight learning curve. If your project needs a large amount of animation, has a very tight schedule, and will use mocap, then Motionbuilder is an absolute must.

Brian Chambers is and artist/animator with eight years experience in television, film, and games. He's currently a lead animator at Rockstar San Diego.

RONINWORKS BURSTMOUSE

THOUGH NEARLY ALL GAMES TODAY ARE full-on 3D experiences, most of you working with DCC apps are still using 2D mice. RoninWorks thinks it has created a device cool enough that you'll want to upgrade to 3D for your main input tool. Unlike 3Dconnexion's 3D controllers, which you operate with your non-dominant hand, move only a few millimeters, and which must be used in addition to a mouse, RoninWorks believes these functions belong together, in your mouse.

The BurstMouse is a normal Logitech optical wheel mouse retrofitted with an optical transceiver, matched to a transceiver with a tripod that sits on your desk. As long as you leave it on the desk the BurstMouse operates exactly as any other mouse, but as soon as you lift it more than an inch or so above the desk, the BurstManager plug-in takes over and turns the mouse into a controller with almost six degrees of freedom (rotation about the axis between the transceivers is not detected), with a working space of a cubic foot or so. At this point, holding down the left button lets you move the currently selected object freely, while holding down the wheel lets you move the camera.

Using the BurstMouse with 3DS Max 6, the device is quite effective, but not without a few flaws—most of which could be corrected in future updates. Tying the 3D movement to picking up the mouse makes the transition seamless, without making you find and click an icon or switch devices. And the ability to move the controller in free space makes some operations more natural, because it's almost as if you're grabbing the virtual object in real-life. And this free-space movement makes the mocap mode of the software particularly useful, since you can simply maneuver objects or cameras around in real-time, without having to create paths or keyframes, or use an expensive full-on mocap system when you only need simple, single-point movement.

When used to control objects, the BurstMouse is smooth, silky, and extremely intuitive. But I was confused at first when I switched to "camera" mode, since the movement was the opposite of what I expected. This behavior is documented, though—you're actually moving the scene in relation to the camera. Allowing the user to choose whether to move the world or the camera would be a welcome improvement, since some will feel more comfortable with one approach, and some with the other.

As cool as the BurstMouse is, not everyone will want to hold a mouse in the air for long periods of time—and not for all 3D operations, either. It's great for coarse movements and for mocap, but for fine control, many will fall back to the normal 2D mouse-based controls in 3DS Max and Maya.

Peter Sheerin is the product review editor at Game Developer. *E-mail him at psheerin@gdmag.com.*

The BurstMouse might look a bit odd, but the sensor on the front turns what would otherwise be a normal mouse into a powerful 3D one that works with 3DS Max and Maya.

RONINWORKS BURSTMOUSE

SLICK +

STATS

RoninWorks
3259 Progress Dr.
Orlando, Fl 32826
(305) 433-7115
www.roninworks.com

PRICE

$350

SYSTEM REQUIREMENTS

Windows 2000 or XP; Max 6; Maya 5; one PS/2 and one USB ports; or two USB ports.

PROS

1. One device works as a 2D mouse and a 3D controller.
2. Translates 3D movement in space directly into application.
3. Can position cameras, objects, and provide real-time mocap input.

CONS

1. With no option to flip axis directions, some movements are the opposite of what some users will expect.
2. Doesn't adhere to the USB HID Multi-Axis Controller 6DOF API, and only supports 3DS Max and Maya.
3. Not as small and ideal for packing in your laptop case as 3Dconnexion's SpaceTraveler.

OUR LITTLE SECRET:

OFFSHORE OUTSOURCING

AS THE LARGEST INDEPENDENT PUBLISHER OF VIDEOGAMES, Electronic Arts can afford the $10 million it takes to develop a top quality title. Its chief creative officer, Bing Gordon, prefers to keep development a team-oriented exercise. "It's very hard to take the subcomponents of a game and farm them out," Gordon said in an interview. "Building games involves so many small decisions. Doing that face to face works the best."

Yet EA has found that outsourcing some of the art for its LORD OF THE RINGS games to artists in China is a worthwhile endeavor, and in doing so it's jumping on the same globalization trend that has swept through the software industry. This trend makes business sense because it can lower costs and it allows publishers to tap into a wider pool of talent across the globe. But some industry practitioners are keeping quiet about it, partly to gain an edge over rivals, and partly because outsourcing can cause strife among professionals who fear it will cost them high-paying jobs in North American studios.

"If you look at the software market, there is a lot of backlash about outsourcing now," says Vamsee Tirukkala, CEO of Zinnov, a 10-person consultancy in Bangalore, India, which offers advice and research about outsourcing. "If you look at software, this evolved over the past five years. Now it's happening in the game industry. Over time, it will only accelerate. It may not be as huge as with software, but it will happen."

While the creativity required to make games has held the trend toward offshore outsourcing in check in the past, growing pressure on publishers is forcing both publishers and developers to consider outsourcing, says Jason Della Rocca, program director of the International Game Developers Association.

"The production model is following film, with a tight nucleus of a team and a lot of specialty work farmed out," Della Rocca said.

EA, Microsoft, THQ, UbiSoft, Turbine, and others have all tapped developers in Eastern Europe, China, India, Taiwan, and other countries to make all or parts of games. Japanese publishers rely on China's Colorland studio for a great deal of their cinematics. Blame it on the viral spread of gaming across the globe, which breeds fans who in turn try their hand at the craft themselves. They start out doing ports, localizations, or secondary art and work their way up to cinematics and entire games, says Tirukkala, who works from an office in Santa Clara, Calif.

No one has a monopoly on talent, and the talent is springing up in places where there has been no strong tradition of videogame production. Gabor Feher of Digital Reality in Budapest, Hungary, is a case in point. His 40-person company has worked on a variety of games, and it has lately been specializing in creating music for games, tapping into the Budapest Symphonic Orchestra for help.

"We find that music is the easiest thing of all to outsource," Feher says. "And if you do it right, people will even say that music will enhance the graphics of your game."

So far, most game companies aren't commissioning entire games to offshore teams. Game development makes sense as a centralized team effort because it requires so much iteration as developers, programmers, and artists all tweak their work and then play test it to see if it works, says Bruce Shelley, a co-founder of Ensemble Studios. Daniel James, president of Three Rings, a game development studio with six employees in San Francisco, says, "You'd be a fool to ship a game design offshore and wait for it to come back as a finished product. If you can specify the work, it might work. But there is a communications bandwidth issue. It can be very frustrating."

Yet, as Feher attests, there are a lot of pieces that can be farmed out. Redwood City, Calif.-based EA has 3,100 in-house developers. But the company is still dealing out art tasks to Vykarius Visions, a Culver City, Calif. start-up that has been in business for a year. Vykarius's co-founder and CEO, Xin Chung, in turn travels to China and commissions the art from any one of six different artist studios that employ 350 artists, all spread out from Hong Kong to Beijing.

The art in a game, whether it be 3D models of secondary characters or background art, is the easiest element of game creation to farm out. Companies such as New Pencil in Sausalito, Calif., legitimized this form of outsourcing by making the furniture and other pieces of art in EA's THE SIMS computer game. Chung, a veteran of companies like Interplay and Square, saw the trend in software and the potential of China, and he concluded overseas outsourcing was inevitable.

"Silicon Valley has become a land of headquarters staff and lead designers," Chung says. "Hollywood is a land of art directors and key artists. I predict the same thing is going to happen with the North American game industry."

One of the reasons it makes sense to do the less important tasks overseas is cost. The salary for a programmer at a top studio can amount to $110,000 a year (see "*Game Developer*'s Third Annual Salary Survey," February 2004), compared to a senior programmer in China who might make $15,000. The pay difference for artists is similar, says Jason Robar, a developer in Issaquah, Wash., who brokers deals between game publishers and offshore developers. That kind of savings could shave $250,000 from the cost of a $1 million project, Robar estimated. In-house

DEAN TAKAHASHI *Dean is a staff writer for the* San Jose Mercury News *covering the game industry, and is the author of* Opening the Xbox: Inside Microsoft's Plan to Unleash an Entertainment Revolution.

William Reusch of InterServ International wants to bypass the need for frequent layoffs.

teams also have a lot of idle time in between projects. Publishers often have to resort to layoffs if they don't choreograph the transition between major projects flawlessly.

But Chung realizes that his company has to provide high-quality service on top of low costs. The company is off to a fast start—having already created art for Surreal Software's THE SUFFERING title (see "Postmortem: The Design of Surreal's THE SUFFERING," May 2004)—because it anticipated what kind of communication would be necessary with publishers. Chung and his lead animator Ed Trillo say they know the pitfalls of bad outsourcing deals.

"We wanted to do this by taking what we learned about the pains of outsourcing," he said.

Chung has plenty of competition. Another rival is InterServ International, which has its operations in Taiwan and Shanghai as well as a liaison office in Irvine, Calif. William Reusch, international business development manager, says his company is working to do art for publishers in Europe, Japan, and the U.S.

"We noticed many developers were, due to the project-based nature of the business, having to lay off portions of their work force at the end of a project or upon project cancellation," Reusch said, suggesting that it's often easier to outsource for that period of time.

However, outsourcing nightmares do occur. Joss Ellis, chief operating officer of U.K.-based Argonaut, said communication problems dogged three of four recent outsourcing jobs. Gordon also says it doesn't make sense to outsource games that require cultural knowledge, like baseball. But these problems shouldn't make U.S. talent feel smug.

Vladimir Starzhevsky, executive producer of MIDDLE EARTH ONLINE at Turbine Entertainment in Westwood, Mass., says that he chose to work with a team of 25 artists in St. Petersburg, Russia, because they were by far the most talented team that he could find anywhere.

"These are the only guys who can do this kind of art," he says. "They have 10 years of experience working with us."

That team supplements but doesn't replace Turbine's core art team in Massachusetts. That team concentrates on creating the core art assets, while the Russians finish populating the world with similar pieces of art. That helps Turbine complete the massive amount of art needed in a massively multiplayer online game.

Daniel Bernstein's Sandlot Games uses Polish artists to remain competitive.

Chung's team of a dozen game veterans in Culver City serves as the front office liaison to the publishers. They meet with publishers to define projects and then pass the details along to the Chinese artists. Everyone logs into the same planning tool, dubbed Vyknet, which is an integrated platform for collaboration, workflow, documentation, art production, and project management all rolled into a single web-based software program. But Chung is also fluent in Chinese and spends about a third of his time in China. He is constantly seeking out more artists to handle the growing scale of projects.

Mark Vange, president of Ketsujin Studios in Toronto, Canada, echoes Chung on the importance of good communication with the publisher. He uses a team of 30 game veterans in St. Petersburg, Russia, to either take on entire projects or art projects. But Vange says his presence in North America as the liaison is critical to keeping the publishers happy. They can look at a build in the morning and give him feedback by the end of the day. Thanks to the time difference, Vange can talk to his Russian developers as they are arriving to work, who then send revised work. This work style, called "following the sun," enables Vange to gain a day.

"Usually, I talk by encrypted instant messenger, but if there is rancor on a change, then I talk to them by phone," says Vange, whose studio started in 1991 and has worked on projects such as Microsoft's FIGHTER ACE. "If we decide something, it is documented in an e-mail and inserted into the bug-tracking system."

To win work, outsourcing companies make bids. Vange says that publishers pay upfront for the work and release more money with each milestone achieved. Overall, he estimates his teams get their work done for half the price of a U.S. art team. Besides cost, he notes that one of the unforeseen added benefits is that it's easier to design a game for international markets if you use developers that span more than one region.

"If you consider things like how female figures should look in games, you get a very different answer from one overseas team or another," Vange says.

On rush jobs, outsourcing companies come into their own. Chung says his company had three weeks to fill out the art on the World War II level for Surreal's THE SUFFERING game. His team received a library of textures that showed the art style, as well as a wire-frame model of outlines of the level. A blueprint from Surreal showed Chung's team where the beds should be in the barracks. Chung's team created a cabinet for the barracks room, but the Surreal folks wanted more drawers in it, so the artists revised it.

"I talked to them on a daily basis, and because of the time zone differences we had a very nice meshing of work schedules," Chung says. "At first, our textures weren't right. But it took about two iterations to finish the job."

Some companies are still resisting, but the dominoes are falling. Microsoft is famously contrarian in that it hasn't farmed out much of its software development to India, and its policy in past years was to require all of its developers to relocate to the Seattle area. But Microsoft has tapped Ketsujin as well as Dhruva Interactive in Bangalore, India, in part, the company says, to gain competitive advantages.

It doesn't take a huge company to benefit from outsourcing. Sandlot Games in Bothell, Wash., only has six employees working on casual games that can be downloaded from web sites. But Daniel Bernstein, CEO, has tapped a small team of four artists in Poland to create the art for games like SUPER GRANNY.

Bernstein says the Polish artists produce high-quality art for relatively simple games that take only about four months to complete. Bernstein visits the Poles periodically, but also has a source control system that works across the Internet.

"Our programmers set up the environment, and their artists plug the art in and populate the world," Bernstein says. "I've come to trust them so much that I don't need to do any concept art with them."

Bernstein doesn't look at this business style as costing any jobs in the U.S. He says his company wouldn't be competitive if it didn't use the Polish artists.

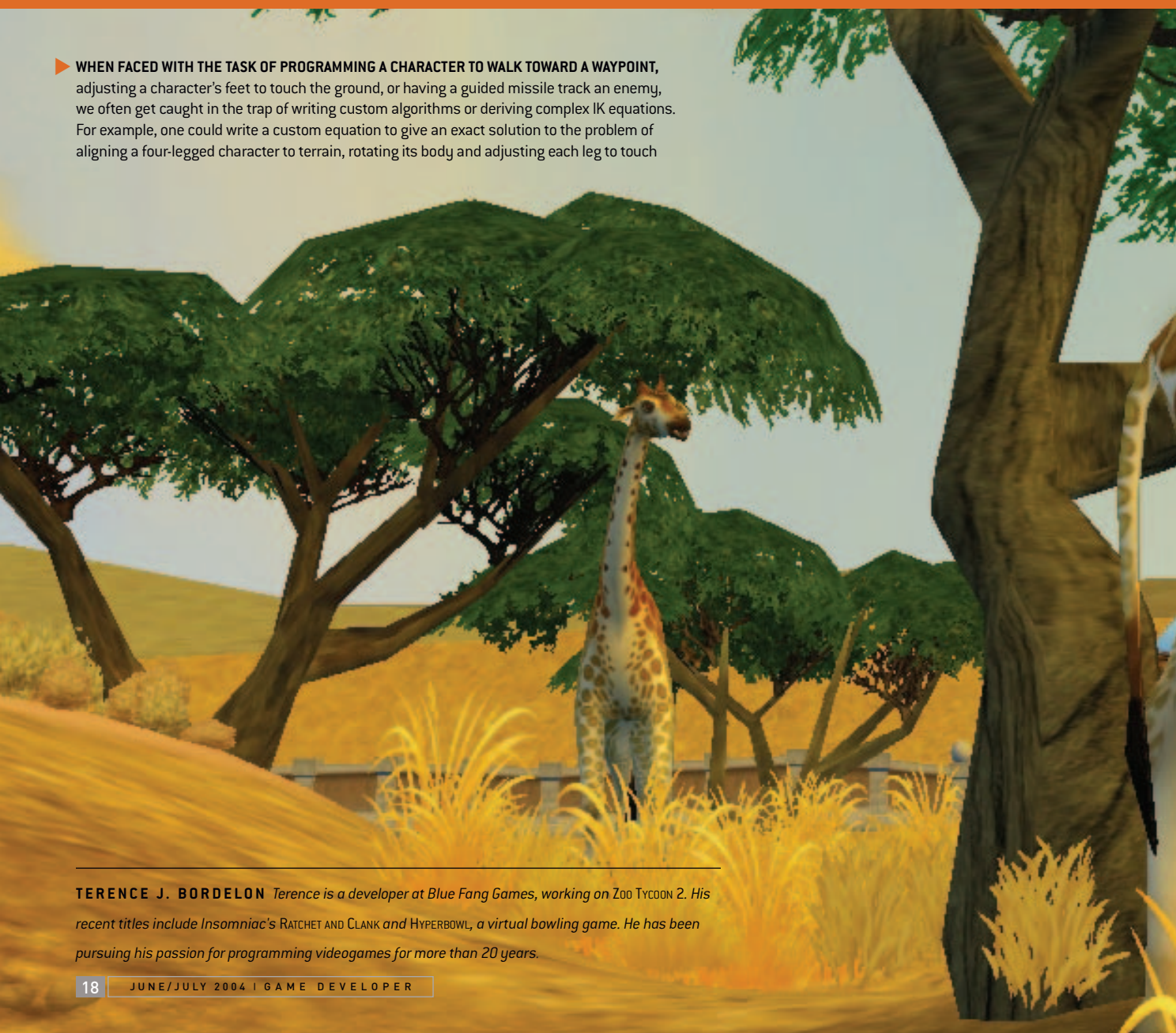
"We would not be able to continue production and stay in business if we were not outsourcing," he says. "Because I can outsource the art, I can employ other people here."

» terence j. bordelon

FEEDBACK

IMPLEMENTING REAL-WORLD CONTROL

WHEN FACED WITH THE TASK OF PROGRAMMING A CHARACTER TO WALK TOWARD A WAYPOINT, adjusting a character's feet to touch the ground, or having a guided missile track an enemy, we often get caught in the trap of writing custom algorithms or deriving complex IK equations. For example, one could write a custom equation to give an exact solution to the problem of aligning a four-legged character to terrain, rotating its body and adjusting each leg to touch



TERENCE J. BORDELON *Terence is a developer at Blue Fang Games, working on* Zoo Tycoon 2. *His recent titles include Insomniac's* Ratchet and Clank *and* Hyperbowl, *a virtual bowling game. He has been pursuing his passion for programming videogames for more than 20 years.*

LOOPS

SYSTEMS IN YOUR GAME

SCREENSHOTS FROM BLUE FANG'S ZOO TYCOON 2

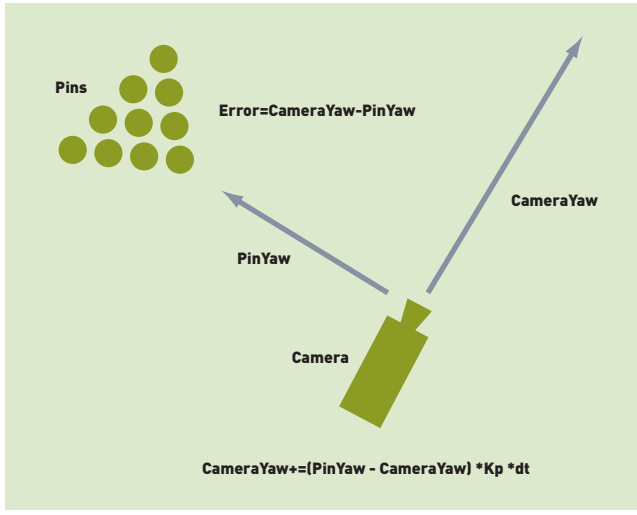


Figure 1. Bowling camera feedback loop.

the ground. Such an exact solution would have a tendency to pop or behave in a snappy fashion as the character moved, simply because of the low-resolution nature of poly-based terrain. We would then need to smooth out or average the character's movements to get rid of this high frequency noise. But let's see what we really did by averaging out our precise equations. We went through the trouble of computing an exact solution, and then introduced inaccuracies because we really didn't want an exact solution. This exercise not only wastes precious cycles in your game loop, but burns development time as well.

Feedback loops can provide a better way to solve problems like these. They are powerful tools commonly used in control systems for autopilots, guided missiles, and probably the thermostat in your office. These equations are very useful in situations where you wish to control an aspect of a system such as the rotation of a character or the position of a vehicle in relation to a goal or command over the span of many frames. Rather than spending time developing custom algorithms from scratch in an already tight development schedule, traditional feedback loops can often be utilized instead. This article will introduce you to basic feedback loop equations, how I've applied them in titles I've worked on, and will lead you in the direction of using them in your own titles. Got a goal? Got an error? A feedback loop will get you there.

DISCOVERING FEEDBACK LOOPS

It was 1998, and I was just wrapping up development on HYPERBOWL, an arcade-style virtual bowling game. There were quite a few pops and jarring transitions on the camera, so I decided to start smoothing them with some simple algorithms I thought up using good old programmer common sense.

One particular case I remember quite well—the camera transition at the end of the bowling frame. The player would bowl down the lane with the camera following the ball through twists and turns. Finally, the ball would impact the pins and the camera would immediately pop to face the action. This looked quite jarring, and I needed a nice smooth transition to polish it up a bit.

After thinking about it for a minute, I came up with a simple algorithm that "felt" right. [Note: There are no proofs presented for this one, as the following algorithm came from instinctual programmer reflex to the problem at hand (see Figure 1).]

I would retrieve two variables: CameraYaw, which represented the camera's yaw in world space, and PinYaw, the bearing from the camera to the pins. I knew I wanted CameraYaw to slowly face PinYaw, so I took the difference between them to get an error, multiplied this error by a constant, and added the result to the camera's yaw. To top it off, I multiplied in my time step so the speed didn't vary with the frame rate. The result? The camera would smoothly transition into facing the pins with a very simple equation:

$$\dot{\text{CameraYaw}} = \text{CameraYaw} + (\text{CameraYaw} - \text{PinYaw}) * K * dt$$

Little did I realize at the time, but I had just re-invented a proportional or P feedback loop.

It wasn't until I began consulting for a small company writing control systems for drone aircraft that I fully learned and started implementing these very common equations known as feedback loops. I was given the task of writing the autopilot firmware for a small drone with the primary goal of steering toward preprogrammed waypoints. At first, this seemed like a daunting task—until I applied a simple feedback loop to the problem. I remember standing there in awe the first time my "Airbot" turned and flew dead on to a preprogrammed waypoint at the other end of the field. In the end, the guts of the code to perform this feat turned out to be a single line. I was immediately hooked and began to implement feedback loops to solve most of my controls-related problems.

WHERE ARE FEEDBACK LOOPS USED?

To describe its workings in slightly more detailed (but still general) terms, a feedback loop is given an error signal, or an indication of how far away it is from its desired state. Additionally, gains are given to the loop to control how it responds to error. The output of the feedback loop then adjusts the system under control, moving it

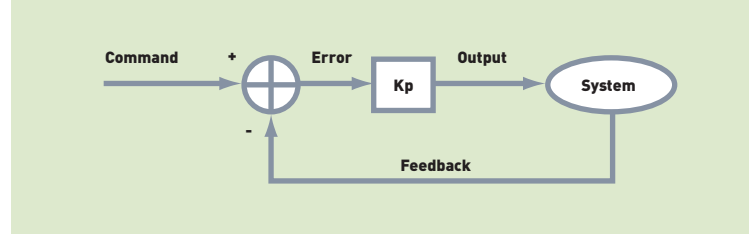


Figure 2. Block diagram of a proportional feedback loop.

ever closer to the goal and reducing the error. If you have a problem that doesn't need to be solved immediately (in this frame right now), and you can generate an error signal and directly control a system that affects the error (such as the rotation of a bowling camera), then you can use feedback loops to solve your problem.

I have used feedback loops extensively in camera control, keeping a character's legs adjusted to match terrain, keeping a character's head trained on a target, guiding an auto-targeting weapon to an adversary, and many other diverse systems.

You can save yourself development time (not to mention a few headaches) by using feedback loops where appropriate, as in the HYPERBOWL problem, where I really didn't need an exact solution but could tolerate slow adjustment over many frames.

Feedback loops are simple enough to be implemented in hardware with a couple of operational amplifier chips, or in software with a single line of code. Their simplicity would explain their penetration in many fields dealing with control systems.

TRADITIONAL SOLUTIONS

Nowadays, programmers faced with the task of modeling the behavior of a system generally resort to writing equations to accurately and rigidly solve for the solution, much as we were taught to do in grade school. Thinking back to my own experiences in algebra class, I remember being told to "draw a diagram of the problem to be solved, and write an equation that solves for 'x.'" This is the traditional way videogame programmers generally solve controls-related problems.

Keeping a character's head trained on a target could easily be solved with grade school level algebra, giving us an exact "look at" angle. We know the absolute coordinates of the character and the target, and can retrieve a vector between the two. A quadrant check and an arctangent later and we have our exact answer. And since we don't want our head to snap to the new position, we would introduce some averaging to move slowly over time.

Keeping a character's legs firmly planted on the ground can also be solved in this way as well. We can easily draw a diagram of the angles involved, the distance between the foot and the ground, and an IK (Inverse Kinematics) equation that will return the new rotation angles of the leg to plant the foot exactly on the ground where it belongs. As in our previous example, we would then smooth out the response once again to eliminate snaps as the foot moves over polygon seams.

We could even solve the HYPERBOWL problem by addressing it with grade school algebra instead of feedback loops. The result would be the same: the camera would slowly rotate to look at the pins.

While we could use traditional methods to address each of these cases, feedback loops are a better choice. In all these cases, a feedback loop solution takes less processing time, is simpler to implement, and rarely explodes from bad math. Conversely, many numerical solutions tend to have degenerate cases where the answer goes to infinity. Furthermore, feedback loops let you use the same equation to solve a multitude of problems. This means that you don't have to get out the graph paper and solve a specific problem for a specific case, saving you some time and gray hairs.

THE MATH OF PROPORTIONAL FEEDBACK LOOPS

The block diagram of a simple proportional or P feedback loop is shown in Figure 2. I've found that most of the control system problems encountered in game programming can be tackled with this simple loop, without resorting to more complexity.

The proportional loop simply outputs a signal proportional to the error. In Figure 2, the "system" is the object in your game under control. It could be the rotation of a character's head, the height of the camera above the ground, or (as in the case of my earlier HYPERBOWL example) the rotation of the camera. The "command" is basically the state you wish the system to be in. It could be a desired rotation, position, or other attribute you want to command the "system" to be at.

Following the diagram from the left to right, we see the current state of the system is subtracted from the command. This leaves us with an error signal that is multiplied by the proportional gain *Kp*, and the result used to change the state of the system. The equation of this block diagram can be stated as:

Equation 1: $\dot{O} = (\text{Command} - F) * Kp$

O is the current state of our system, and *F* is the feedback signal giving us the current system state. The dot over the *O* says that this equation gives you the derivative or "rate of change" of the system. So, given a command, the current system state, and a gain, we are able to compute the derivative (or rate of change) of the system. Take a look at Figure 3 for an idea of how the P loop generally responds.

SIMPLE INTEGRATION, EULER STYLE

But this isn't the whole picture. Since the previous equation only gave us the rate of change of the system, we need to integrate these results over time. I usually do this with Euler integration, but some people prefer fancier methods such as trapezoidal, midpoint, or even Runge-Kutta. I'll use Euler when I can get away with it, and so far that's usually 100 percent of the time. Euler integration is performed on our previous equation to integrate the result into our state variable:

Equation 2: $O_f = O_{f-1} + \dot{O} * dt$

O_f is the system state in our current frame, and O_{f-1} is the previous frame's state. So, multiply the result from Equation 1 by the time step *dt*, then add it to the previous frame's state and voilà—we have computed the new state of our system! If we do this every frame, our system variable *O* will move toward the commanded state, as expected.

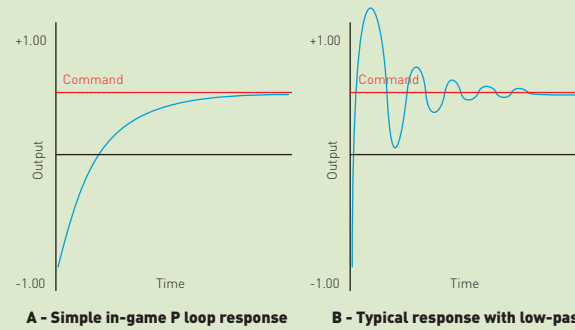


Figure 3 (A and B). Proportional feedback loop outputs.

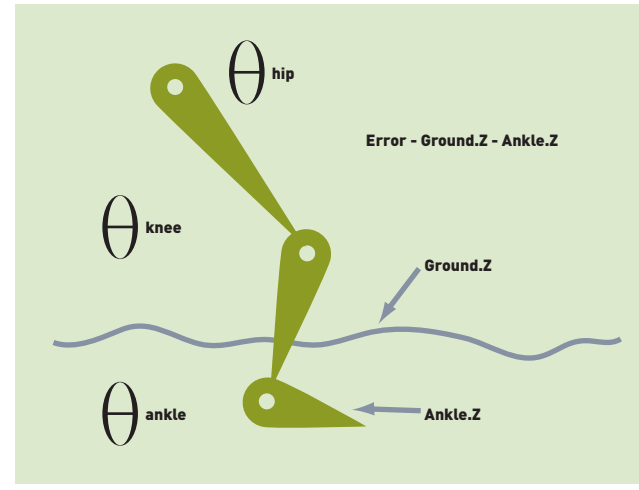


Figure 4. Fitting character legs to uneven terrain.

PLAYING WITH *KP*, THE PROPORTIONAL GAIN

Setting the gain (*Kp*) in Equation 1 is somewhat of an art. We know the sign is negative since we want the signal to be corrective instead of adding to the error. As a general rule, the higher the gain, the faster the response, while lower gains usually give a slower response. It usually helps to provide yourself with some kind of interactive adjustment so you can fine-tune your gains in real time.

Kp works proportionally on the error. This means the bigger the error, the bigger the adjustment. As the error approaches 0, the adjustment slows down, effectively "decelerating" the system to a stop right where you want it.

THE FEEDBACK SIGNAL ISN'T ALWAYS THE OUTPUT

In our HYPERBOWL example, the feedback signal just so happened to come from the same variable we were controlling with our output signal. Our feedback loop treated the system as a variable, controlled the value of that variable, and "fed back" that same value.

But let's imagine that you are given the task of writing AI code to control boats running a physical simulation. In this example, the goal is to have the boats steer to a programmer-controlled compass heading. To do this, let's assume that we want to write a feedback loop that controls the rudder position. But note that in this example, feeding back the rudder position will do you no good! What you really want to do is to feed back the current compass heading.

Just remember that the "feedback" signal is basically just an indication of how we're doing. In the boat example, it is the current compass heading. The "command" is where we'd like this feedback signal to be. For our boats, this is the commanded compass heading, such as "steer to 45 degrees." These two signals are subtracted, scaled by the gain, and used to control the rudder, which is completely unrelated to degrees or compasses.

USING PROPORTIONAL FEEDBACK LOOPS FOR "HEAD LOOK"

Let's say that you wish to have your character's head look at an enemy in your game. We have a variable named R_{head} that controls the yaw rotation of the character's head in local space. This is the system state we wish to control. Next, we need to generate a "command" to control R_{head}. In this case, we wish to command the head to face R_{enemy}, or the rotation angle that would face the head to the enemy in local space. With these terms, we can write a proportional feedback loop by plugging these variables into Equation 1 creating:

Equation 3:

$$\dot{R}_{head} = (R_{enemy} - R_{head}) * Kp$$

And using Euler integration from Equation 2 we get:

Equation 4: $$R_{head_f} = R_{head_{f-1}} + (R_{enemy} - R_{head}) * Kp * dt$$

This simple line implements a P feedback loop that will keep our hero's head pointed at our enemy, slowly decelerating as the head reaches its target.

POOR MAN'S IK— ADJUSTING LEGS TO UNEVEN TERRAIN

Recently, I was faced with the challenge of writing a control system to keep animals planted to the ground in my company's (Blue Fang Games) upcoming title ZOO TYCOON 2. I did this task with a simple P feedback loop on animal height and another on animal tilt. This did most of the job, but the legs would still penetrate the ground on slopes and uneven terrain.

I defined this problem as shown in Figure 4 so I could begin to write a feedback loop to control the legs. I decided the feedback signal would be the base of the foot's *Z* value and my command into this system would be the *Z* of the ground directly above the foot node. (I wanted the foot to be on the ground.) I also decided that this feedback loop would control θ_{adjust}, which would add a corrective rotation to the hip, knee, and ankle in order to raise or lower the leg.

In Figure 4, there are three joints that must be moved to lift the leg, so I had to write some simple equations that could take θ_{adjust} and apply it to all three joints in such a way that it would move the leg up or down. The math for this turned out to be quite simple:

$$\theta_{hip} = \theta_{animation} + \theta_{adjust}$$
$$\theta_{knee} = \theta_{animation} - \theta_{adjust}$$
$$\theta_{ankle} = \theta_{animation} + \theta_{adjust}$$

At this point, I have all the information I need to write my P feedback loop by plugging values into Equation 1:

$$\dot{\theta}_{adjust} = (\text{Ground.}Z - \text{Foot.}Z) * Kp$$

And applying Euler integration once more:

$$\theta_{adjust_f} = \theta_{adjust_{f-1}} + (\text{Ground.}Z - \text{Foot.}Z) * Kp * dt$$

Running this equation every frame and on every leg did the trick. In fact, the solution surprised me in how robust it actually was. I've successfully implemented this logic with very satisfying results. Animals walk on uneven terrain, up and down hills and ravines, all with their legs adjusting to the ground. The best part is that I didn't have to open a calculus book or spend months deriving custom equations to do it!

OSCILLATIONS AND OVERSHOOT

There is a nasty little property of feedback loops in that they generally like to oscillate with high gains. In other fields such as motion control, this is usually a problem that creeps up with the design of any feedback loop.

Figure 3B shows us what the response of a real-world system running a P feedback loop might look like, such as a real security camera tracking a target. Notice the loop causes overshoot, but eventually settles out at the commanded state. This overshoot is generally caused by the camera having mass, which basically slows the response of the system and acts like a low-pass filter. Once we have an object with mass moving, it won't stop instantaneously and will tend to go beyond the commanded position.

Now check out Figure 3A. This is the response of a typical videogame feedback loop. Notice that oscillations are completely absent from the output of this loop. This is due to the fact that objects in the game can stop on a dime. They generally have zero mass, and when commanded to will perform a 5,000-G deceleration and stop exactly where commanded.

In other fields, more advanced feedback loops are generally used to dampen these oscillations. Fortunately for us, the simple P feedback loop will be all we need a majority of the time.

For those of you implementing feedback in physics simulations, you will have to worry about oscillations just as real control systems engineer would. For P feedback loops, the general rule is to keep your gains as low as possible to get the job done. There are more advanced loops designed to address this very problem, and we'll touch on those a bit later.

ACCURACY CAN BE AN ISSUE

If I were writing an aircraft flight simulator targeting hard-core aviation buffs, feedback loops would be a poor choice to use as a flight model. Conversely, if I were writing a mini-game and needed to simulate flight on an escape pod, they would work just fine. As long as you think hard about the need for absolute accuracy and you answer "no," feedback loops are a great alternative to traditional mathematical modeling. They are designed to give you ballpark answers—just the type of answers we generally need in videogames.

ADVANCED FEEDBACK LOOPS: PI, PD, PID

We've only gone over the simplest of feedback loops, the simple P loop. This is usually sufficient in videogames since the only input signal into the system is usually the one we give it. But in real life, (and sometimes in physical simulations) there are external influences on the system other than the one our loop applies. These external influences sometimes necessitate bringing in gains that act on the integral (I) and derivative (D) of the error.

Take a look at Figure 5 and you'll see the block diagram for a full PID loop. We'll explain the upper I and lower D branches shortly, but what you should remember is that these new branches can be mixed and matched to create several new types of feedback loops. If you don't need a branch, you can leave it out. For example, a PD feedback loop simply omits the upper integral and *Ki* blocks.

Let's go back to our boat example. You'll find that a P feedback loop is adequate only if there are no additional forces affecting the boat. Throw wind into the equation, and the boat will settle into a compass heading that isn't quite what you commanded.

This deviation is what's known as "steady state" error and can be taken care of by a PI loop, as shown in Figure 4. The steady error builds up in the new integrator block, and eventually gets subtracted out of the equation.

The equation for the PI feedback loop looks like:

$$\text{Error} = (\text{Command} - F)$$

$$\dot{O} = (\text{Error} * Kp) + (\int \text{Error} * Ki)$$

PD loops are similar to PI loops, except we look at the derivative as opposed to the integral. PD loops allow higher proportional gains to be used by tuning the derivative gain to dampen any oscillations.

$$\dot{O} = (\text{Error} * Kp) + (\dot{\text{Error}} * Kd)$$

Finally, PID loops bring the terms all together to yield an all powerful feedback loop that can have high gains and even handle steady state error. The only trouble is that they can be rather hard to tune.

$$\dot{O} = (\text{Error} * Kp) + (\int \text{Error} * Ki) + (\dot{\text{Error}} * Kd)$$

And once again, you'll have to run these new equations through an integrator much like we did in Equations 1 and 2 to use them in your games.

SIMPLIFYING COMPLEX BEHAVIOR

As games get more complex, the utilization of real world control systems to control a wide variety of systems should become far more commonplace, not to mention that controlling complex behavior from a single, reusable algorithm is usually a welcome alternative, especially when a milestone is fast approaching. I hope the examples in this article can provide some guidance to those not already familiar with their uses and benefits.

FOR MORE INFORMATION

Control System Design Guide, George Ellis, Academic Press. Very straightforward and includes software to help design feedback loops

Expert Tune. This company exists to sell software, but has articles that are useful on PID loops. **www.expertune.com/articles.html**

FAQ on PID controller. **www.tcnj.edu/~rgraham/PID-tuning.html**

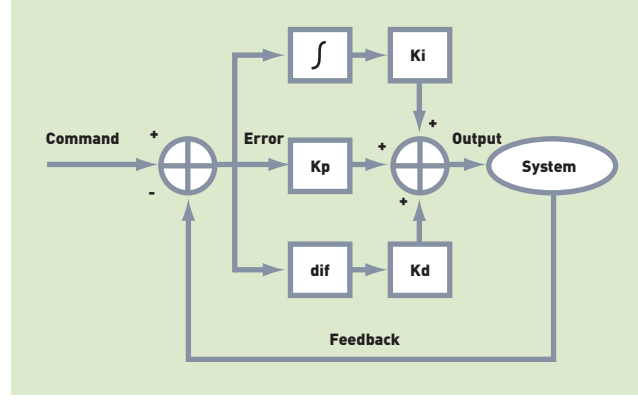


Figure 5. Block diagram of a PID feedback loop.

DOING MUSHROOMS, MIYAMOTO-STYLE

WHEN PEOPLE START READING SUBTEXTS INTO THE EVENTS OF your works, such as Frodo inserting his finger into Lord Sauron's ring of power, Lucy spreading apart the professor's wife's furs to discover Narnia, or Mario diving into a magical warp pipe fishing about for her majesty, you know your time has come. The trouble is maintaining that position.

Yet stamina doesn't seem to be an issue for Shigeru Miyamoto. Whether it's a healthy supply of 'shrooms, or little blue pills, he's definitely got access to some good stuff. Since his creation of Mario Bros. and Donkey Kong, through to The Legend of Zelda, cart racing, and Pikmin, punctuated by his introduction and refinement of handheld gaming, analog control, and controller force feedback, Miyamoto has dominated and deeply influenced the videogame industry for decades. Furthermore, he has demonstrated the ability to transfer his surreal development prowess to others such as Eiji Aonuma, whose artistic and commercial success as director of The Legend of Zelda: The Wind Waker earned him the authority to oversee the entire Zelda franchise. Recently, Miyamoto's infectious innovative spirit flourished yet again at E3 with the unveiling of the Nintendo DS, which he co-created with Game Boy Advance SP industrial designer Ken'ichi Sugino.

Given this mentoring trend at Nintendo, we thought it might be possible to have a few batons passed your way. What follows is a Triforce of interviews, with all three of these leading Nintendo developers sharing their insights.

Jamil Moledina: *How do you create that elusive fun factor, and how do you know you're getting it right?*

Shigeru Miyamoto: The most important thing is that the game director not lose sight of the point of origin, the reason they're creating the game they're creating. Every game starts off with some core element that you want to create and you want people to experience, but gradually a lot of times when people are creating games, things don't develop the way they expect them to, so to solve that problem, people gradually add new elements to make that game better. In doing that, you can end up going down this path where you've added all these different elements, and the game changes from what it was originally intended to be. Now, of course, if in doing that the game gets better, that's not a problem, but a lot of times it's very important for directors to refer back to that starting point and make sure that they're staying true to that. And obviously, there are exceptions to the rule where you do have to add on new elements, then the problem with that is that the game development never really ends, because you keep adding new things until you decide it finally becomes interesting.

Eiji Aonuma: For us, we're always thinking of new ideas, even during the development of say the previous title, and we'll look at what we're doing, what we want to do with that title, that we weren't able to accomplish for whatever reasons, be they technical or time constraints, and then try to use those core elements and find ways to expand them, find ways to do them better in the next game. And essentially, use these ideas from past games or ideas we had while creating a game and kind of let that evolve into the theme of the next game.

Another thing for us that's very important is that we don't just try to think up ideas, but we actually allow our experiences to spawn ideas, or instigate ideas for us. Even if I'm out with my family and I find something interesting, or experience something that I think is very fun, I might look at that and say, "That's kind of fun. How can I take that and bring it back to Zelda games?" And implement it in a way that people can interact with it and experience the same feeling of fun that I experienced when I first saw it in the real world. With Zelda, we've created this world for the player to go play in, and in this world we put in things that we think the player might want to do—that they might want to play or might want to interact with. And in doing that, what we've done is that we've given the players the opportunity to use their own imagination to come up with their own way of playing. I think that's what makes Zelda fun.

JM: *As the game industry has grown, games have been subject to political scapegoating, particularly with relation to violence in society. How do you feel about the way the industry is treated?*

SM: Any new media or industry that grows rapidly is going to be criticized. That's just because the older, more established media have been around, and a lot of adults can be very conservative. They may not have an open mind to new things that weren't around when they were growing up, and are replacing the things they grew up with. You know, looking at the games that I've made, fortunately they haven't met with a lot of the same criticism that a lot of the other games have. That's really important to me. I want to create games that don't fall into those strong stereotypes about videogames and instead I want to create games that others will instantly see primarily as a fun entertainment form to be enjoyed. With things like the DS and its touch panel and the new style of control that that's going to offer, and the Donkey Konga drums we've introduced with the Gamecube, I think those are really going to change little by little the image that videogames have. You know,

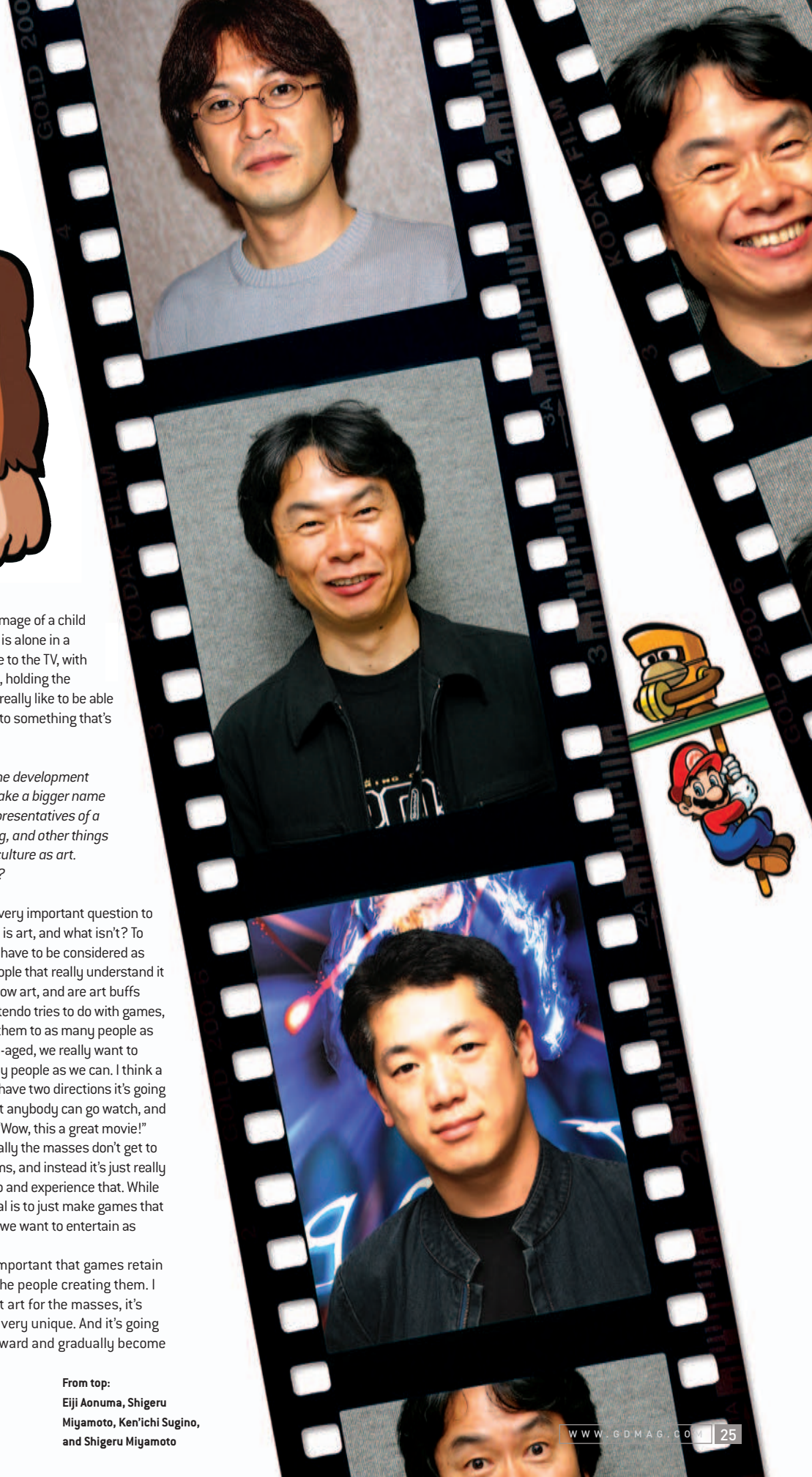
JAMIL MOLEDINA *Jamil is the managing editor of* Game Developer *magazine. He can be reached at jmoledina@gdmag.com.*

over the years I've seen this standard image of a child playing a videogame in which the child is alone in a darkened room, with his face very close to the TV, with the light of the TV reflecting off his face, holding the controller, and just staring at the TV. I'd really like to be able to change that image of videogames into something that's a little more positive.

JM: *There's a growing sense in the game development community that developers need to make a bigger name for themselves, as the creators and representatives of a burgeoning art form—like film, painting, and other things that are commonly accepted in world culture as art. Do you perceive games as an art form?*

EA: As someone who studied art, it's a very important question to ask: Where do you draw the line? What is art, and what isn't? To me, I don't necessarily feel that games have to be considered as art. If you really think about art, the people that really understand it are those that have studied art, and know art, and are art buffs really. Whereas, if you look at what Nintendo tries to do with games, we want to create games and present them to as many people as possible, young, old, middle-aged, teen-aged, we really want to entertain people, and entertain as many people as we can. I think a good example is the film industry. You have two directions it's going in. You have the mass market films that anybody can go watch, and enjoy, and be entertained by, and say "Wow, this a great movie!" And you have arthouse films, where really the masses don't get to experience or enjoy the art of those films, and instead it's just really people who are film buffs who get to go and experience that. While there's nothing wrong with that, our goal is to just make games that are fun and entertain people, and thus we want to entertain as many people as we can.

On the other hand, I think it's very important that games retain individuality and the individuality of the people creating them. I mean, if you look at art, even if it's not art for the masses, it's very distinct, and each piece of art is very unique. And it's going to have its own flavor. If games go forward and gradually become



From top: Eiji Aonuma, Shigeru Miyamoto, Ken'ichi Sugino, and Shigeru Miyamoto

DOING MUSHROOMS

more and more alike so that there is no more individuality in games, that's not going to be good for the industry. So to me, it's very important to have that uniqueness and those distinct characteristics and try to continue to evoke that in the games that we make.

JM: *In developing multiplayer games, Microsoft and Sony have both acquired an expertise in online gameplay. Instead, Nintendo has traditionally focused on in-person multiplayer experiences. The Wi-Fi-enabled DS seems to signal a transition for Nintendo, to integrate a broader range of players who are online. What accounts for this change?*

SM: What we're looking at with the DS really stems from what we're doing with connectivity. We introduced connectivity to the Gamecube, so that if you also had a Game Boy Advance and a link cable, the gameplay we offered was very fun. But unfortunately, there were a lot of people who didn't have all these elements. So with the DS, we thought, what if we took all that and the fun elements we innovated with that, and put them all into one system from the get-go? So all you do is buy this one hardware system, and right out of the box you have all these connectivity options. Whether you've got one DS and one game pack, and your friends have their DSes, you'll be able to download the game from the game pack in one DS wirelessly to your friends and you can all play together wirelessly. So it's these kinds of "straight out of the box" and "nothing more to buy" ideas that we're excited about with the DS. It's because we're focusing on that that we can look at other options. It's like a jumping off point, where you can expand from there into the online realm. But for us, the main focus is that we want to provide people with this experience straight out of the box.

JM: *Given the fact that DS introduces so many innovations at once, not just the dual screens, combining 2D and 3D elements, but also two wireless systems, both touch and voice as new game interfaces, what advice would you give third party developers for taking full advantage of the DS?*

Ken'ichi Sugino: Rather than trying to give them advice of what to do, I'm really looking forward to seeing what they come up with on their own.

SM: There are a lot of people who have been in the industry a long time, that have been making games, and are always thinking up new ideas for games. Part of the problem is that a lot of people who come up with these ideas for games haven't had a hardware system that's been able to bring these ideas to fruition. One really good example of this is a game called Pac-Pix that Namco has created on the DS. Essentially, what you do is, on the touch screen, there are ghosts running around on the screen, and you with the touch stylus draw Pac-Man [draws a simple Pac-Man figure on his notepad] like that, and the Pac-Man you just drew starts animating. Then you draw a line like that [draws a line underneath the Pac-Man figure] and Pac-Man goes down, you draw a line that way [draws a line to the side of the Pac-Man figure] and try to keep Pac-Man on screen until he catches all the ghosts. This was an idea that Namco had had for a long time, and the problem was that they never had a hardware system that would allow them to realize this. So when we took the DS to them and told them that this is the hardware we're making and these are the features it has, immediately they said, "We have a game we want to make." So, we're looking forward to seeing what other ideas people have had that they weren't able to realize before. And this is just one example.

Another thing to look at is this whole idea of multiplatform games. It's gotten to the point now where people are fighting so hard for exclusivity on a particular game for their platform and it's just gotten into this big weird battle. Mr. Yamauchi, the former president of Nintendo, was very averse to the idea of multiplatform games. From the standpoint of a developer, it's very easy for them to create one game and port it to other systems, but from our perspective, we want to provide people with new forms of gameplay and really fresh ideas. And so rather than have the same game on every system, we like the idea of having people bring their games to a Nintendo platform and then add in new elements that will make it very unique to the Nintendo platform. The idea with the DS, with having these new features like wireless, the touch panel, the built-in mic, is that people who created a game for one hardware system might bring it to the DS and say, "Well, look at all the things I can do on the DS, I can add all these new features." We think there are a lot of ideas floating around like that, in addition to those that people want to make strictly for the DS, because of all the things we can do with it. And so I think that's going to bring a lot of creativity and freshness to the games that we'll see.

JM: *What are your impressions of the PSP?*

SM: I actually haven't seen it yet. I'm sure it's probably got a big screen on there, and I'm sure they've tried to pack as many technical specs in there as they can, but I wonder how long their battery life is going to be. I haven't heard anything about that yet.

For us, we didn't make the DS because we wanted to make our console games portable, or because we wanted to make our Game Boy Advance games more gorgeous. We really wanted to create the DS so that people could create completely new styles of games that no one has ever experienced before. And so in that

sense, looking at what they're doing with the PSP, I don't really see it being a competitor with the DS, because it's really similar to all the systems we've seen before; it's just more specs.

JM: *How did the DS come together?*

KS: Rather than being one idea that started in one spot and was brought somewhere else, we kind of always had a two-way street between me the hardware developer, and Mr. Miyamoto, the software developer, with an exchange of ideas. Rather than him saying, "Hey I want this," and us saying, "Okay, we'll do that," we brought our ideas to the table at the same time. It had only been six months since we introduced the touch panel and in just those six short months, we've seen all kinds of ideas that have come up since then. So it was really an ongoing communication of going back and forth between each other, exchanging ideas, and coming up with it.

SM: So the DS is a result of a two-year process, and it's only been six months since we've had a working version of the touch panel, and that's where all the ideas have come from. We have different research tracks going on at Nintendo, one that was researching the touch panel, one that was researching the wireless, and one that was researching the high-tech graphics chips, and so it was really a question of how do we combine these elements together in a way that we can create this hardware. So it was really an amalgamation of these research tracks that resulted in the DS.

JM: *What games are you playing now?*

SM: Right now I'm mostly playing the DS games I'm working on, and I'm having a lot of fun with those. I don't have a lot of time to play other products beyond what falls under my sphere of influence. I've been playing The Legend of Zelda: Four Swords Adventures, and Mario vs. Donkey Kong for the Game Boy Advance, which I think is really fun.

KS: I'm not just saying this because I'm in an interview, and I'm supposed to behave for the PR guy, but I'm actually playing a lot of Super Mario Bros. from the NES classics series. That was from the era of gaming that I grew up in, so I really like that.

EA: Before attending the Game Developers Conference, I saw the list of nominations for the Developers Choice Awards, and in there I saw Prince of Persia listed. I had heard from various people that this game was supposed to be very fun, and so I managed to get my hands on a U.S. copy of the game, and played it. I thought the game was indeed very well done and very fun.

Note: For the raw, unadbriged transcripts of Jamil's interviews with Shigeru Miyamoto, Eiji Aonuma, and Ken'ichi Sugino, visit www.gamasutra.com.
Special thanks to Bill Trinen, Yasuhiro Minagawa, and Chris Olmstead for their translation and facilitation services.

POSTMORTEM

THE TOTAL PUNCH CONTROL OF

EA SPORTS FIGHT NIGHT 2004

FROM THE VERY BEGINNING OF DEVELOPMENT ON EA SPORTS FIGHT NIGHT 2004, our goal was to change the way boxing games were played. For years, the boxing game genre had become stale and boring. Boxing gameplay was nothing like real boxing. Even from a pure gaming perspective, the gameplay offered did not make for a fun game. It was completely mindless gameplay. Boxing fans were disappointed in the games as were sports gamers, fighting game fans, and, well just about everyone. In order to get people interested in the boxing game genre again, we needed to change the way the games were played. This was our mission, and something which we were able to achieve in EA SPORTS FIGHT NIGHT 2004 with an innovative control system we call Total Punch Control.

(continued on page 30)

WRITTEN BY:

KUDO TSUNODA *With 10 years in the industry, Kudo leads the newest Electronic Arts studio—EA Chicago. He served as executive producer on EA SPORTS FIGHT NIGHT 2004 and also worked on BATTLESTATIONS, the ARMY MEN AIR ATTACK series, and UPRISING X. Prior to working in videogames, Kudo was a world-renowned lion tamer and ran the Cobra Kai Karate Dojo chain in Southern California. He is also an avid collector of butterflies.*


JONES JR

Kudo training with the team in preparation for developing EA Sports Fight Night 2004.

THINGS THAT WENT RIGHT

1 **FINDING THE MUSE.** Developers making a boxing game without ever having stepped into the ring just did not make sense. So we sent the entire design team, myself included, off to a local gym to learn how to box. Each of us participated in a rigorous training regimen that culminated in actual amateur fights against other real life boxers. You quickly learn what tactics and skills a boxer needs to be successful in the ring when the penalty for not knowing these things is getting smacked in the face. Hanging out at the gym as much as we did, we got to spend a lot of time talking to pro boxers and trainers. Our main goal was always to make the game fun first. But we wanted the game to be accurate to the sport wherever possible as well. You can't beat getting key input on your ideas from people like Roy Jones Jr., or our trainer at the local gym.

While learning how to box, we also spent long weeks playing every boxing and fighting game available and watching an endless amount of fights on tape. Often, our entire day was spent going to boxing class, training and fighting in the gym, and then going back to the office and fighting each other in different games until late at night.

All of this left us with an enormous amount of ideas to choose from. This is the key to design—not just coming up with the one good idea, but having a huge list of great ideas to choose from. Do whatever you have to do to generate this type of list, as it will be useful to the entire development team throughout the course of the project.

2 **IDENTIFYING CORE DESIGN PHILOSOPHIES.** Two things became very clear to us. The first was how important being able to control your fists is to boxers in real life. How you move your fists, where you throw your punches, and how you use feints and fakes are critical to landing punches on your opponent. One thing that was really surprising to me is how important fist control is to having a successful defense. Boxers are constantly moving their fists into position to block incoming punches from their opponents. Throwing punches only makes up a small percent of the fist movement that occurs during a fight.

The second thing we realized right away was how little control over fists previous boxing games have given gamers. Despite all the technological innovations in gaming since the Atari 2600, the way to throw punches in boxing games was always the same: you press a button, and a punch animation plays. Boxing games had given gamers no ability to do the one thing most important in boxing: control their fists. It was easy to see at this point why boxing games had become stagnant. It resulted in gameplay in which both boxers would run into the middle of the ring and mash on the punch buttons as fast as they could. There was nothing to learn. You could play the game for weeks and still lose to people who had just started to play.

After putting these two things together, what we needed to do started to come into focus. We wanted to give gamers control over their fists while providing a punching mechanism that would give a better feeling of actually throwing your punches. Our design focus quickly shifted to the user interface and how the gamer would control his or her boxer via the controller.

3 **DESIGNING THE CONTROLS.** Since the controls were the single most defining element of the game, they needed to be designed as early as possible. So we set some main design rules of engagement that should be followed when determining the controls:

1) Increased control for gamers over boxer fists
2) Change the simple button-press punching found in all other games.

These two simple rules quickly focused our efforts on the analog sticks. So we sat down and started mapping potential controller

It's about using the right tools.

"Stunning physics bring our worlds to life. To bring physics into our world, we use Havok dynamics."
Gabe Newell, Valve Managing Director

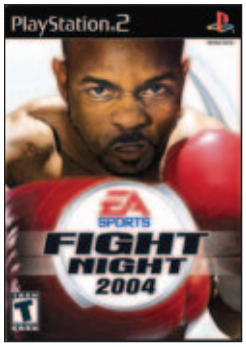
"Making a game with the complex physics interactions and physics gameplay of Half-Life 2 would have been impossible if we hadn't started with robust fundamental physics technology from Havok."
Jay Stelly, Senior Software Engineer, Valve

Check out Havok on the web at:
www.havok.com
Want Havok 2 in your next game? Send email to:
eval.gdm@havok.com

HALF-LIFE 2

powered by havok®

GAME DATA



PUBLISHER:
Electronic Arts

NUMBER OF FULL-TIME DEVELOPERS:
45

RELEASE DATE:
April 2004

PLATFORMS:
Xbox, PlayStation 2

TOTAL HOURS SPENT IN GYM:
922

DEVELOPMENT TEAM FIGHT RECORD:
24–19

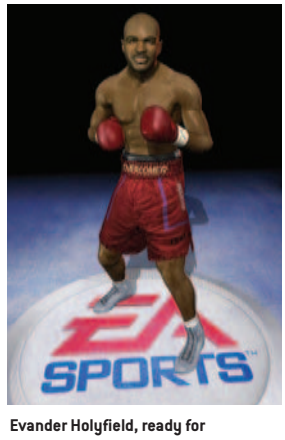
SIZE OF TEAM'S FIGHT LIBRARY:
479 fights

configurations on paper. While having paper controls defined is not the best way to define controls, it is quick and easy. Picking up a piece of paper with control ideas scribbled on them and picking up a controller to see how potential user actions might feel is an effective way to start debugging your thinking.

Many of our ideas failed our two-rule test at this stage. We had ideas that would throw punches based on a simple analog directional movement. But all of these really gave no extra control over your fists. Moving the controller in a single direction to throw a punch was just an analog way of doing button-press punching. Move the controller in this direction and a punch animation would play. No improvement to the old ways of doing things. Other concepts were just not easy to do while holding a controller. We just were having trouble creatively figuring it out. So we went back to the gym. We all spent an hour or so working on the heavy bag throwing punches and thinking about controllers.

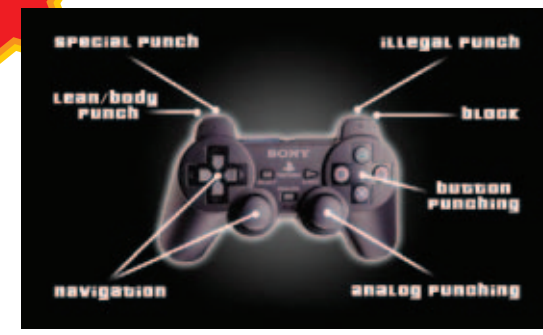
With punching being somewhat second nature at that point, it freed our minds up to think about controllers while throwing punches. With each of the basic punches, jabs, hooks, and uppercuts, you could see basic motions that your fist made with each punch. It finally dawned on us that you could just give gamers total control of their fists and let them make the same motions with the analog sticks that boxers make in real life with their fists. A left jab goes straight out and to the left. Right hook goes out and to the right and arcs forward. Uppercuts start out low and then make a big arc upwards. Instead of having gamers memorize un-intuitive patterns, you could have the motion the boxer's fist makes on screen in the game mimic the motion the user makes with the analog stick. So simple, yet so hard to figure out!

Once we had the overall concept of what we wanted to do, it was easy to drop it into the game and start getting a feel for it. We tweaked control responsiveness and other control elements hundreds of times until we felt it was tight. As soon as we were happy with the control system, we started to focus test it. We had people from other teams in our company playing it. We were doing something like three or four focus groups a week. Seeing other people play with the controls revealed some flaws in our thinking, and every time we got feedback from the focus groups, we did a new build addressing problems we saw. We always had a new build ready for the next focus group. The rapid iterative process, getting feedback, making changes, and getting more feedback was great for getting things perfected in a timely manner. Sometimes, focus groups are a pain in the neck for the development team, but if used correctly they can be instrumental to the development process.



Evander Holyfield, ready for action.

4 IF IT WORKS, EXPAND ON THE ORIGINAL IDEA. Another key



The controller layout for the PlayStation 2 version.

deficiency in earlier boxing games was the complete lack of defense. They had movements such as blocking and leaning in the game, but by the time you tried to use them you got hit in the face three times, which defeated the purpose. With the way Total Punch Control was implemented in our game, it really was more like total fist control. You can move them where you want and when you want extremely quickly using the analog sticks. With our boxing experience and knowing how much fists are used to pick off incoming punches, it became a natural fit to have the analog fist control be the basis of our blocking system.

The idea and implementation at this point was simple. As somebody playing the game, all you have to do is use the analog stick to move your fists into the path of incoming punches. Blocking immediately went from slow and cumbersome to the reflexive, lightning-fast movements you actually make in the ring. You can move your fists to cover any area of your body that you want. With the analog controls, you can do it fast enough to see a punch coming in and move your fists into a blocking position in time to stop the punch.

Total Punch Control was evolving into something much grander in scale than our original plans. With all the pre-production work we had put in, the ground work had been laid so well that the ideas just started to develop themselves. What we could do to improve the game became more and more obvious as features like blocking went in. The other main defensive tool boxers have is using upper body movement to slip punches by leaning out of the way or rotating their upper-body. The range of upper body movements boxers have is the exact same range of motion you have with an analog stick. It did not take the genius behind the winch mechanic in the ARMY MEN AIR ATTACK series to realize we could directly map the in-game upper body movement of your boxer to an analog stick. Our analog capabilities just evolved so quickly—from Total Punch Control to total fist control to total boxer control. Not only did we have a great, innovative punching system—we had made defense fast and easy to use. The controls were tight, tested, and focus tested early and often in the process. They became the core of our gameplay and it was amazing to see the game dynamics that came from our original design work on the controls.

5 AN ENDLESS SUPPLY OF GREAT GAMEPLAY. The gameplay in EA SPORTS FIGHT NIGHT 2004 broke down to two distinct experiences: head-to-head gameplay and our one-player career mode. The two-player gameplay mostly just created itself as we started playing the heck out of the game. Everybody on the design team was fighting head to head 40 or 50 times a day, figuring out which tactics were successful and which weren't, using our new control scheme. In multiplayer, it became clear that we really had an endless supply of tactics you could use to land punches. With defense being so effective, button mashing gameplay or just throwing non-stop punches became completely ineffective. Counter-punching was the key to being successful. You needed to create offense by using your defense, just like in real boxing.

Like any great multiplayer game, the tactics and skills we

$60k for some California property or...

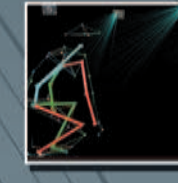
...a VICON motion capture system in your office, at your command.

DON'T HAVE ENOUGH SPACE?

Bet you do.
The New VICON Vcam will allow you to capture in a conference room.

DON'T HAVE SOMEONE TO RUN IT?

You don't need to.
VICON's award winning iQ software lets you capture and process your data automatically and export to the format of your choice.



THINK IT IS TOO EXPENSIVE?

Not any more.
Turn-key systems for $60k.
Limited time offer. US and Canada only.

VICON USA T. 949 472 9140
VICON UK T. +44 (0) 1865 261800
60k@vicon.com

OMG

www.vicon.com

VICON MOTION SYSTEMS

used evolved over an endless series of one-upsmanship. If I were playing somebody on the team and beating them, they would figure out a strategy to beat my tactic. Once they started to beat me, I would have to evolve my style of play to beat them. Just like in actual boxing, styles really made the fight. I could constantly beat team member A for a stretch of time but I would lose to team member B that team member A beat regularly. It all depended on the tactics and styles of boxing you used and how they matched up with each other. We carefully documented all the different tactics we used along the way in playing the game in multiplayer. And these became a big part of our AI design. One thing that hit home in this project is that the best gameplay system allows the person playing to develop the tactics necessary to win. I have played EA Sports Fight Night 2004 for nearly two years now and I still learn cool new things I can do in the game. You just can't beat that type of gameplay.

THINGS THAT WENT WRONG

1 **CHANGING THE DEVELOPMENT TEAM MINDSET.** Just due to the logistics and cost of doing so, not every team member was able to participate in the extensive research we did that led to defining our key features for EA Sports Fight Night 2004. Without all the knowledge the people on the design team had, some team members were definitely skeptical about Total Punch Control and what it would mean to the gameplay of the title.

Button-press punching had been the norm in every single boxing game before EA Sports Fight Night 2004 and there were definitely people on the team who could not understand why we wanted to change that. As a design team, we probably did not do as good a job as we could have in not just documenting and disseminating our conclusions, but involving and updating people on the conversations we were having and the things we were finding out along the way. This is very time consuming to do and, with the long hours we were pulling, was not a top priority at the time.

But figuring out some easy mechanisms for getting all this information out to people without dragging them in to long meetings or forcing them to read extensive brainstorming documents would have greatly increased team buy-in on the feature at the beginning. We ended up spending more time explaining things to people after the fact than we would have keeping people updated along the way. But this was clear to all of us in our internal postmortem and we have developed some great ways for getting a ton of information to people and allowing them to contribute to the ideas without taking a lot of people's time or forcing them to contribute if they do not want to. Fortunately, once the advantages of Total Punch Control became clear, getting team buy-in was a lot easier. But we could have done this more effectively early on.

Analog punching chart.

2 **FEATURE IMPLEMENTATION DEPENDENCIES.** As I discussed, we were really focused on getting our controls implemented as early in the development process as possible. But we did not do a very thorough job of figuring out all the dependencies for this and making sure the production plan was geared up for delivering all the assets we needed to get Total Punch Control in the game in its final form. This is a very difficult challenge as you don't really know what you need until the feature is implemented, but you need those same things to get the feature in the game.

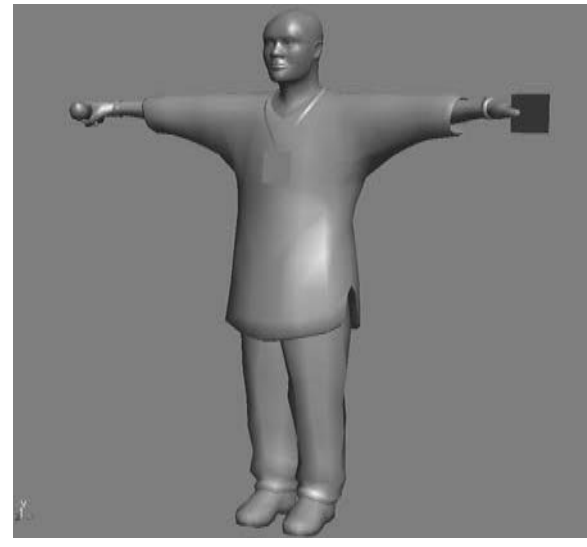
For instance, our punch and defensive controls are heavily animation-dependent. But you can't know exactly what animations you need until the controls are final and in the game. With mocap shoots being so expensive, you can't go back and do this several times for each control iteration. Plus, it's hard to know when to schedule the shoot, because you don't have a good estimate of how long the iterate-and-update process is going to take for the controls. So we had placeholder animation in the game that helped us get the controls most of the way there. But our mocap shoot was scheduled for weeks after we had a control system we were happy with. We ended up having to go back and re-tune the controls after getting the final animations and lost some of our design work on tactics and game ideas that were effective with the placeholder animations but not with the final ones.

Textured screen capture.

3 **TEAM MEMBERS MUST PLAY ON DEFAULT SETTINGS.** One of the things that we have in EA Sports Fight Night 2004 is a lot of different options for customizing the game. There are different difficulty settings, camera settings, control settings, rule options, and so forth. You try and find which combination is the best for consumers and make that the default while providing gamers the ability to customize it to their personal wishes.

Development team members are no different than consumers. They like to customize their play experience to their wishes. Unfortunately, this can lead to some confusion when people are implementing and testing different features with a different set of settings. With so many different options at your disposal, it is important to have people work on a single group of settings, preferably the default ones, as they are used the most, and build the game from that common frame of reference. We definitely had some wasted work due to people implementing things that worked well on their machine but not as well on others.

In the end, you want the game to play equally well on all settings configurations. But you need to have everybody start at the same point and make that fun so the way the game is being tuned is consistent. You can then take the default gameplay and make sure it works on all settings. From a development perspective, it's so much easier to have one common frame of reference you are all starting from and then make sure it works on other setups than it is to have people all starting at different points and hope the gameplay experience ends up being a cohesive experience for the gamer.



Shaded screen capture.

4 GOING ALL THE WAY. One of the main things we wanted to do with Total Punch Control from the very beginning was to provide gamers with a gameplay sensation on the analog sticks that was far superior to that of the button-press punches. While this was completely accomplished, we still left button-press punch configurations in the game for people to use if they wanted. I have always been a big proponent of giving the people playing the game the final choice of how to play. In retrospect, it would have been a better decision to completely eliminate the button-press punches.

Button-press punching had been the norm since the very first boxing videogame. It is ingrained in people's heads. They already know how to play that way. In talking with consumers who have purchased the game, some people switched right to button-press punching without even ever using Total Punch Control. No matter how innovative a feature is, human nature is to stick with what you already know. EA Sports Fight Night 2004 is a genre-defining product. People were going to buy the game whether we had included a button-punching configuration or not. In the end, this is one case where we might have been better off to eliminate a user choice and get everyone using our game's biggest feature.

5 FIGHTING THE "TOO CLOSE TO THE GAME" SYNDROME. One of the hardest things about tuning gameplay is trying to keep yourself in the mindset of somebody who has just purchased the product. By the time a development team gets to the part of the project where they actually tune the gameplay experience, you have played the game so often and know how it is supposed to be played. That makes it hard to make decisions on what is best for somebody who does not have the experience and knowledge you do.

It is critical to come up with some methodologies for doing this. Obviously, focus testing the game with novice users is a great way to get a different perspective. But development teams make decisions every day and you cannot go around focus testing each and every one of them. Well you can, but that certainly slows down production.

One easy thing to do is to start playing on controller configurations that you are not used to. Often, the alternate control schemes of a game are different enough that it throws you off your game and forces you to re-learn where the controls are for what you want to do. But you can pretty quickly get re-acclimated, so we started to figure out new ways to bring about this same goal. After playing so much that we were used to all the control schemes, we went back to the default controls but switched up our hands on the controller so our right hand was performing left hand tasks and vice versa. This messed us up and made us feel like we had not played the game before. When we mastered the default controls, we moved through all the alternate control schemes as well. Then we turned the controller upside down and tried playing that way. Again, this playing method totally threw us right back into novice user mode. Once we were masters at playing that way, we played with our hands crossed up and the controller upside down on every control scheme. We also tried playing this way and alternating control schemes each fight.

Concept art for the ring.

In any case, this is a hard thing to get the development team to do as it is frustrating and cumbersome and nobody wants to do it. But the exact point is that if it is frustrating and cumbersome for you, it could very well be a giant pain in the neck for somebody who purchased the game. Coming up with all these ways to turn ourselves into novice users allowed us to make quick and sound decisions which were then verified by focus group results instead of having to wait for the focus groups to tell us what we needed to fix.

FINAL ROUND

Our team took a stale genre and energized it with an immersive, yet faithful, control system and had a lot of fun along the way. Looking back on the experience, I feel that this kind of process, of focusing analysis and real-world experience on the control system, could very well enhance the gameplay of other genres and styles of play. So pick up a copy and play with it a bit, and see if our control design philosophy sparks something for your next project!

JONATHAN BLOW

EXPERIMENTS I'D LIKE TO WORK ON

THIS IS MY LAST COLUMN FOR THE INNER Product—for a while at least. I'm going to spend some time focusing on game creation, away from distractions like writing magazine articles. Upcoming columns will be written for you by the capable and curmudgeonly Sean Barrett.

Over the past several years, many interesting projects have accrued on my "I Really Want To Do This" list. As a parting shot, I'll talk about the more exciting items on this list: why they are interesting technically and how they are important for the future of games.

None of these items have to do with graphics. I think graphics is well-covered now; a lot of research has been put into it and we've come a long way. Certainly, there are some big problems in graphics that ought to be solved, like real-time global illumination. Even so, I think it's clear the current level of graphics technology is far advanced relative to all the other technical aspects of games. As a result, our games have nice graphics but still lack in lots of other ways, and there's no easy fix we can employ. We need to do a lot of hard work to develop the other areas of game technology; this list is a set of places where I would start.

MAKING PEOPLE MOVE

THE GAME I'M WORKING ON IS AN ACTION role-playing game. You play a guy who walks around the world talking to other people, who are all doing their own things. Since this game is in 3D, I need to show animated 3D people going about their daily activities. They need to walk around, talk to one another, climb into bed, jump out the window of a burning building, eat a shoe, and so on. Traditionally, we make this happen by pre-authoring a bunch of animations for all these characters, then playing back the appropriate animations at runtime, perhaps blending different example animations to achieve the desired effect.

These animations take up a lot of memory, even when compressed. A bigger problem is that they're difficult to make. Competent 3D artists are hard to come by, but even if you get some, the task is daunting and expensive. And even if you can afford it, there are deeper, more troubling problems: your characters can only perform actions you have explicitly animated and you tend to have great difficulty interfacing animated characters with a physically simulated world.

Ideally, we would like all character motions to be generated dynamically at runtime: if AI-controlled characters want to sit down, they figure out how to move their muscles to get over to the chairs and place their butts firmly upon them. This is a very difficult thing to do; as a general problem it is far out of our reach. (It's difficult enough just getting those characters to remain standing.)

I believe it's important for us to work our way toward the goal of dynamically simulated motion, so I've chosen an intermediate approach that ought to be more achievable. Rather than authoring full

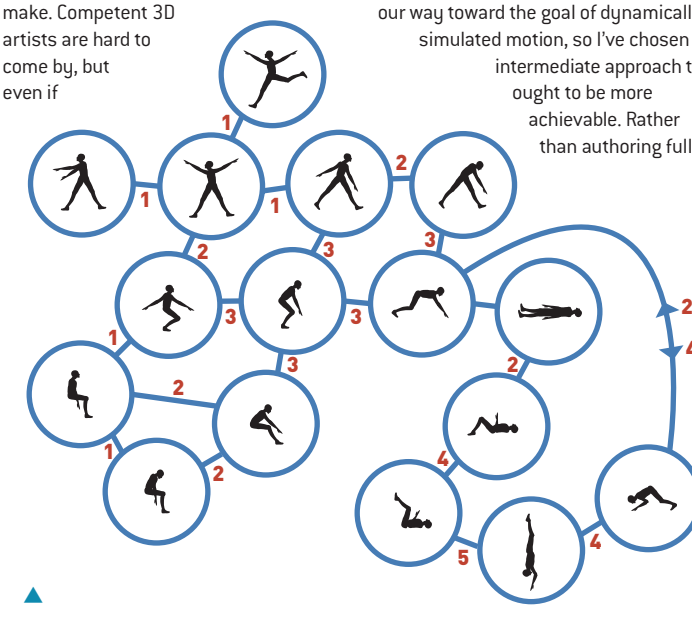


FIGURE 1a. **The pose graph as constructed at preprocess time. The red numbers indicate the uncomfortable labor involved in traversing any given arc.**

JONATHAN BLOW

Jonathan Blow is going to go walk the earth, like Caine in Kung Fu. *E-mail sent to jblow@gdmag.com may eventually reach him.*

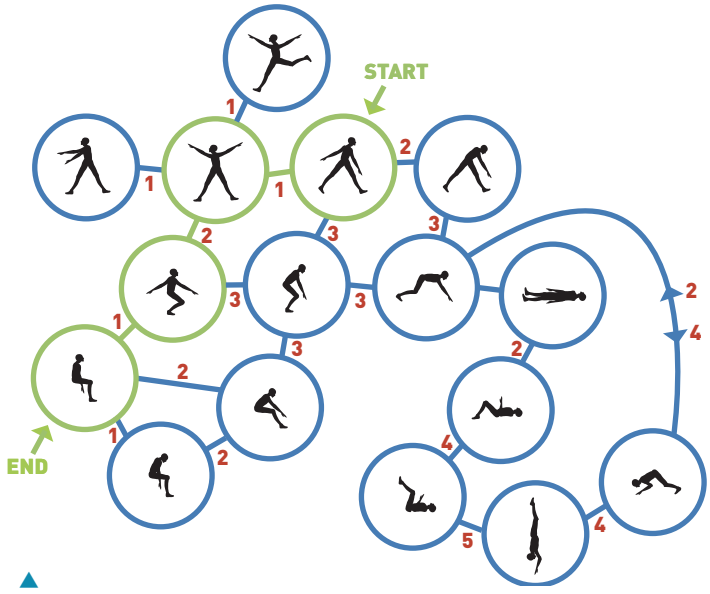


FIGURE 1b. At runtime, given appropriate START and END nodes, we trace a path through the graph, so the sum of the red numbers is as low as we can reasonably make it.

animations, I want to just author a set of single-frame full-body poses. At preprocess time, the poses are arranged in a graph so similar poses are neighbors (Figure 1a).

At runtime, if the AI character is standing next to a chair and wants to sit in it, he or she searches the database for the pose that's nearest to his or her current body state, which we'll call START, and he or she also looks up the pose that is associated with sitting, which we'll call END. The AI then performs a graph search to find a short and comfortable path from START to END. This path represents a plan of how to move his or her body to achieve the sitting action (Figure 1b). From frame to frame, the AI just needs to interpolate along this path until it reaches END.

All sorts of methods could be used for this interpolation. At the simplest, you could imagine just slerping the rotations at the joints of the AI's skeleton. This would look unnatural, but it'll get the job done and it's a good first step when you're working toward building a full system. As a next step, I would have the body apply some forward kinematics to get it from pose to pose—apply muscle forces to the joints to try to get them to do the right thing. As I said before, we're nowhere near solving this muscle force problem in the general case, but now that we have a well-defined pose path, the problem is somewhat constrained. We can use the pose path to impart some magical balancing forces on the character to keep him from getting all wacky. To visualize how this might work, imagine that the path through the graph represents a spline in pose space. When we consider the set of all points within some distance of this spline, we get a sort of bendy rod (you might choose to visualize this as a sphere swept along the spline). The forward kinematics is free to do what it wants so long as it doesn't go much outside this region. If it does, we clamp it or pull it back toward the spline with a penalty force.

So the role of the forward kinematics is mainly to provide flavor, whereas the pose targets serve as a guide-rail to generally control locomotion. As we improve our character balancing technology and begin to attain mastery, we can shift the balance of power between these two systems, so the forward kinematics becomes more in control of overall locomotion and we use the guide-rail less.

The way I've explained this so far, the nodes of the graph are just pose-space targets, but you probably want velocity envelopes attached to the joints as well. Imagine you want your character to throw a ball. If you don't specify the velocity of the hand at the time the ball is released, you can get some motions that are much unlike what you want. I am worried that adding velocity envelopes might cause an explosion in the amount of pre-authored data required and that the authoring process itself would become more difficult. If this is so, perhaps the pose and velocity targets should be automatically acquired from motion capture data. Or perhaps the targets should not contain velocity data; instead, velocities should be imposed by a separate set of constraints used by the forward kinematics (experimentation is required here). Also, even if we're only dealing with poses, we want the ability to decouple different parts of the body somewhat to reduce the sample data—otherwise, for example, we'd need to duplicate each pose of the arms for a lot of different stationary leg poses, which gets silly very quickly. (At the same time, arms and legs are not really independent, so some kind of coupling should still occur. I suspect a lot of IK fixups would be done to the legs for center-of-gravity purposes.)

This system solves many problems. For example, suppose a character is sitting down and you throw a ball that hits his shoulder and knocks him back a bit. All the AI has to do is notice that its current pose is far from where it had hoped to be (measured as distance to that spline), and then perform a "re-plan": find a new START node and re-traverse the pose graph. From then on, it's business as usual. Also, we've eliminated the Covered-Action Assumption that was inherent in pre-authored animation (that your guys can only do what you thought of in advance). If one were to write an AI that could think up arbitrary body poses as solutions to problems, it would then be able to achieve those poses, resulting in actions that, overall, were not pre-authored. (The name Covered-Action Assumption is something I just made up, but I think it denotes an important concept: the set of all actions a character can perform is covered by the set of pre-authored animations.)

The big question with this method is the quality issue: how human will the resulting animations look? I think the results would be highly customizable, by adding new data points into the graph and by tweaking the weights between the nodes to create preferred avenues of motion. In the limit, you could just dump whole animations into the graph frame by frame, for cases where very high quality is desired.

You may notice that this method is a fusion between forward kinematics and inverse kinematics. Choosing the END node, and the path to it, is an inverse kinematics activity, whereas navigating between targets is best done via forward kinematics. So I have dubbed this technique "Alternating Kinematics." Of course the final technique will not be as simple as what I've

said here, as there will be many implementation details to overcome. I hope to report about the results in a future write up!

BUFFING UP COMPUTER VISION

RECENTLY, SONY MADE THE ADVENTUROUS MOVE of releasing the EyeToy as a commercial product—it met with success. This helped show that cameras have an interesting future as game controllers. Unfortunately, our technology for parsing images is very poor. The EyeToy games primarily use a method equivalent to background-subtraction, with perhaps a little bit of augmentation here and there. But this level of technology severely restricts the kinds of games that can be done, since the computer doesn't really understand what the various parts of your body are, or what they are doing. In fact the computer will frequently become confused by changes in scene illumination, motion in the background, and plain old camera noise.

To us, a picture contains a lot of organized shapes and colors that seem obvious. But to a computer, it's just a mess of colors without semantics. It's hard to appreciate the enormity of this issue unless you've spent time developing vision algorithms. If you haven't, I encourage you to give it a try some weekend, so you can come face to face with the problems yourself.

As we overcome these technical limitations, the scope of games that can be done with a camera becomes much wider. Recently, I read Donald Hoffman's book *Visual Intelligence: How We Create What We See* (see For More Information). I found it very exciting. In addition to providing some valuable insights about the human perception of the world, it methodically constructs a framework for a low-level vision system.

Early in the book, Hoffman summarizes the difficulties by defining "the fundamental problem of vision: The image at the eye has countless possible interpretations." This is followed by some guiding principles, like "The Rule of Generic Views: Construct only those visual worlds for which the image is a stable (i.e., generic) view." (A stable reconstruction of the world is one in which significant features don't change for small motions of the viewpoint.) He then presents a series of more concrete rules to help reduce the ambiguities of the vision problem and to explain how the human visual system parses images. There are 35 of these rules in all and they range from the basic ones, such as "Rule 3: Always interpret lines co-linear in an image as co-linear in 3D," to the more involved ones, such as "Rule 32: Construct the smoothest velocity field." These rules are usually justified by actual experiments on humans, many of which you can try yourself by looking at images printed in the book.

This book can be a pretty good high-level recipe for anyone who wants to sit down and write a next-generation computer vision system. The rules are all clear enough that an experienced programmer can clearly see how to start implementing them. Of course we're still very far from solving the vision problem and such a system would be nowhere near perfect (probably we need to solve the full AI problem to make vision really work!), but this would be a good beginning. Our current image-parsing technology is so poor it really shouldn't be too hard to achieve better results. That translates directly into new types of gameplay for camera-based games, which is pretty exciting.

AI FRAMEWORKS: A GENERALIZED HOFSTADTER-STYLE SOLVER

WHEN BUILDING SUCH A VISION SYSTEM, it becomes clear that a unidirectional flow of information—with monotonic construct-building that moves from low-level pixels to high-level shapes—is an insufficient paradigm. Unfortunately, it's the default way we approach these problems; in fact, it's just about the only kind of architecture we've developed strong methods for, which means we're especially ill-prepared. Vision, like every other AI problem, wants its systems to have feedback—a channel of bidirectional influence that unites the low and high levels. A high-level layer postulates the existence of a certain shape in a specific area of the image, which might trigger something in a middle layer to remember it found some features that might be line segments in that region, which might trigger the low level to examine the pixels more closely, which would feed it back upward into the middle layer providing evidence to help confirm or deny existence of those features, which influences further postulates made by the high level. We don't really know how to engineer software that works this way.

The best work I've seen in this area has been done by Douglas Hofstadter. His *Fluid Concepts* book (see For More Information) demonstrates a software architecture that can be applied to this kind of problem. A bunch of tiny agent routines called "codelets" are designed so that their emergent statistical behavior will solve the problem. The codelets run in an essentially parallel fashion, communicating by reading and modifying the notions that compose the current attempted solution. This kind of approach can be seen as an extension of the classical "blackboard architecture," but it operates in a way that's more robust and fine-grained.

As presented in Hofstadter's book, a codelet architecture must be carefully designed for the problem at hand. I'd like to attempt to generalize this kind of system so that it can be applied to a wider problem space with less customization. Computer vision would be a good place to employ this kind of system, but there are lots of other places, like the perception/action cycle for game AIs. Such a generalization is a difficult challenge, so the task is a bit quixotic, but sometimes I like things that way.

Hofstadter argues that AI isn't what the good old-fashioned AI researchers say it is, and I think he's right. I believe in his idea that fluid, continuous pattern-matching lives at the core of intelligence and I look forward to experimenting with this kind of system.

GOODBYE!

THAT'S ALL FOR NOW. Keep on keeping on and I'll catch you later.

FOR MORE INFORMATION

Donald D. Hoffman, ***Visual Intelligence: How We Create What We See***, W. W. Norton & Company, 2000.

Douglas Hofstadter, ***Fluid Concepts & Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought***, Basic Books, 1996.

Sponsored by ACM**SIGGRAPH**

"SIGGRAPH is always great. The Electronic Theater and the Art Gallery are inspiring. The Exhibition is packed with powerful technologies that give me new ideas and new questions. And SIGGRAPH is the community where I reconnect with people I know from all over the world."





Kevin Mack
Visual Effects Supervisor
Sony Pictures Imageworks

Interaction
Knowledge
Reconnection

REALIZE

Creation
Return on Investment
Your Self

conference
8-12 AUGUST 2004
exhibition
10-12 AUGUST 2004

Los Angeles Convention Center
www.siggraph.org/s2004

SIGGRAPH2004

the World's Premier Conference on Computer Graphics and Interactive Techniques

Body-Brush ©2002 Horace Ip, City University of Hong Kong | The Boxer ©2002 Tae Sik Shin, Tuba Animation Studio | Cohesion ©2002 Carlo H. Séquin
Influencing Bone Remodeling in Osteoporosis ©2002 Jane Hurd, Hurd Studios | Molecular Visualization of DNA ©2002 The Walter and Eliza Hall Institute of Medical Research

STEVE THEODORE

PIXEL PUSHER

MAPS AND LEGENDS

Wherein our intrepid columnist manages to smuggle Roman History into a discussion of UV mapping.

THE ART OF MAPPING HAS ALWAYS involved a certain amount of compromise. Ever since people started unwrapping the spherical world onto flat pieces of paper, they've run up against the fact that no method can ever be perfect for relating a complex 3D surface on a flat 2D one.

Figure 1 is the world's earliest UV map, the atlas of Roman-Egyptian geographer Claudius Ptolemy (originally completed around 150 AD, this image is a reproduction printed in 1505). Ptolemy began by inventing the world's first projection mapping gizmo, and the way he did it tells a lot about how texturing works. Since the known world (known to the Romans, anyway) was a lot wider east to west then it was tall north to south, and it was located mostly around the Mediterranean sea, he used a conic projection—basically a modern cylindrical projection that was wider at the equator and narrower toward the north pole. For Ptolemy, the conic projection was a smart choice because it preserved size relationships well in the latitudes Ptolemy knew about. That's why the Mediterranean area in the map looks recognizable. As you head toward the mysterious southern latitudes where one-legged Monopods and the man-eating Anthropophagi lived, things become deformed, so to speak: the world's first example of texture smearing.*

Figure 1. Ptolemy's atlas, circa150 AD: the world's first texture map.

The useful lesson here is that mapping the 3D world onto a 2D plane has always involved an element of choice. It is not a process that can be "solved" or "done right" in an absolute sense. Mathematicians and mapmakers have been trying to unwrap 3D objects onto 2D surfaces for a couple thousand years, and they have yet to discover a method that's completely foolproof, even for something as simple as a sphere. If you visit a cartography web site (there's a great one at www.progonos.com/furuti/Others/toc.html for the curious), you'll find intense discussions about different types of projections and the virtues of each, but they're not about proving one method to be the "best." Professional mapmakers understand there's no such thing as a perfect map. Instead, they focus on knowing what different kinds of maps can do well or poorly so they can choose appropriate tools for their jobs. This is as true for us as it is for them, so this month we're going to try to lay out a good foundation for making the right choices for UV mapping jobs.

AUTHALISM

PROBABLY THE MOST OBVIOUS thing we expect from a good UV map is consistent texture density across the model. The lofty, highfalutin term for this is authalism. A perfectly authalic map is one where every triangle's texture area is exactly proportional to its real-world size. A perfectly authalic map has no texture smearing, since every texture pixel is square and the same size as every other pixel.

Despite its intimidating sound, authalism is actually pretty easy to achieve with modern texturing tools. Since the

STEVE THEODORE

Steve started animating on a text-only mainframe renderer and then moved on to work on games such as Half-Life *and* Counter-Strike. *He can be reached at* ***stheodore@gdmag.com****.*

** Ptolemy also under-estimated the circumference of the earth by nearly a third. The error was largely unnoticed until 1,350 years later when Columbus planned his expedition based on a wildly optimistic notion of the distance to China (thereby inventing the modern discipline of scheduling software development). Generations of producers have subsequently assumed that they, too, would stumble across untold riches just before their budgets ran out—but this, unfortunately, seems not to have become a precedent.*

Figure 2. A perfectly authalic—and perfectly unreadable—map, generated by applying an individual planar projection to every face in a model.

facets of a polygonal model are planar, you could create a perfectly authalic map by simply applying a planar projection to each face in a model, aligned along the face's normal. As long as all the projections are the same size in the world, the texture densities of all the triangles will be the same. This is more or less how most automatic mapping systems work, and in general auto-mapping is an easy way to achieve greater authalism (Figure 2).

EFFICIENCY AND CONTINUITY

IF YOU DO USE AUTOMATIC MAPPING often though, you know that eliminating stretches is only part of the story. Perfect authalism requires addressing each face individually, but this means a texture seam on every triangle edge. Moreover, there's a fundamental limit on how closely you can pack triangles: disjoint sections of the map have to be far apart enough on the texture sheet so unrelated areas of texture don't bleed into each other at lower MIP levels. This means highly fragmented maps are also very inefficient, because so much texture space is wasted on buffer zones between the triangles. Finally, of course, disjoint maps are hard (if not impossible) to paint directly in a traditional 2D paint package. Naturally large numbers of seams have the same drawbacks whether they're generated by automatic mapping or by any other technique, so it's fair to say that a high degree of continuity is also a virtue in a good mapping scheme; however, continuity and efficiency aren't necessarily interchangeable: for example, eliminating seams by stitching UV edges together can leave you with all sorts of oddly shaped groups of faces (UV shells), which present packing problems of their own.

ERROR DISTRIBUTION

LOW CONTINUITY, inefficient packing, and discontinuities explain why automatic mapping has failed to replace traditional planar and cylindrical projections, despite completely eliminating unsightly smears. Unfortunately the demands of continuity and authalism frequently contradict each other. The easiest way to increase continuity is to lower the number of individual projections. Instead of mapping each triangle, you could grab groups of faces that are more or less coplanar and assign them a common planar projection. Within that UV shell the edges will all be shared, so seams won't be a problem.

Of course, this method naturally gives up the goal of perfect authalism. Since most of the faces in the projection will be slightly off-axis from the projection, most individual faces will be slightly distorted in the map. While there is no way to recover that lost accuracy, it is fairly easy to hide it with UV relax tools. UV relaxing basically treats the UV edges inside a shell as if they were springs. The tension of the springs is relative to the real-world length of the edges in the model. Applying a UV relax encourages the springs to reach a stable compromise between their real-world lengths and the deformed lengths they got from UV projection—in effect, it averages out the distortions across all the faces in the shell (Figure 3). This doesn't actually increase the authalism of the mapping, but hopefully it tones down the most annoying artifacts. Naturally, this strategy works best when there are a lot of texture vertices within the shell to distribute the errors in smaller chunks.

READABILITY

I**F AUTOMATIC MAPS** are immune to stretching and smearing, but also impossible to paint by hand, then the opposite end of the spectrum is old fashioned Quake-style sheet-texturing (Figure 4), in which an entire model is textured on a single planar projection (well, actually two planar projections: one for the front of the model and one for the back). Since it generates so much stretching around the edges, the drawbacks of this method are pretty well understood; nevertheless, it has some noteworthy virtues that deserve to be pointed out. Single plane texturing offers pretty good continuity, but the best thing about it is readability. Single plane projections are perfectly suited to traditional paint packages, because they slap the model into a completely consistent frame of reference for the texture artist (albeit at the expense of distortions around the horizon of the projection). So even though single-axis projection is a bit awkward, the combination of straightforward readability and minimal seaming is still powerful—and this is why it still turns up so often despite all the advances in texturing that have happened since Quake.

Figure 3. A complex 3D model with a single planar projection shows noticeable smearing. Relaxing the UVs distributes the errors across the whole surface, reducing the most obvious problems.

Figure 4. The QUAKE guy in all his glory.

CONFORMALITY

ROUGHLY SPEAKING, single planar projections have the lowest learning curve for the texture artist, but they also have some serious limitations. In theory, you could try to improve an old-fashioned single sheet map by using the UV relaxation to distribute the errors more evenly. While this would certainly lessen the worst smearing artifacts, it would of course undercut the intuitive connection between the 2D map and the 3D model. This "trade-off" business is getting to be rather a bore, isn't it?

In any case, an old-school QUAKE skinner can tell you that QUAKE maps may be easy to understand, but they're hard to do well, because there's no guarantee a shape you laid down in the texture will be the same shape when it shows up on the model. Cartographers say a map is conformal if it preserves relative angles—a line bearing north-by-northwest should always be north-by-northwest if you translate it anywhere on a conformal map. A good example of conformality would be the classic satellite photo texture applied to a sphere using a cylindrical projection. Cylindrical projections are conformal because the pixel grain of the map runs north-south and east-west, retaining the angular relationships on the surface—a coast that runs north-by-northwest in the map will do the same on the surface of the sphere. The same map applied as a planar projection would clearly not be conformal, since north-south lines in the original texture would look quite different away from the center meridian of the sphere.

In the traditional cartographic world, where the model is always the same old boring sphere, conformality is a relatively simple problem. For us, though, there are additional complications. Consider the UV map of a character's torso in Figure 5. The map is reasonably authalic, and in the front facing polygons of the chest the map directions seem to be predictably the same on the model. However, the orientation of the map gradually changes as the map unrolls around the character's torso. This is analogous to the curved panels in a clothing pattern. As you can see, the belt line of the model has to match the curvature of the UV shell or end up curving downward—a difficult task for the texture artist. Within each triangle, though, the map actually is conformal (note how the pattern in the belt is preserved within the individual triangles). This is both a blessing and a curse. If the triangles are large enough, painting within them is fairly easy, but the changes in local orientation between adjacent triangles will be very noticeable; conversely, smaller triangles will make it easier to paint smoother, visibly flowing lines but will need continuous adjustments on the artist's part to maintain patterns.

MAKING GOOD CHOICES

AS WE SAID AT THE OUTSET, the infuriating thing about developing a UV map is that achieving any of the individual goals involves making sacrifices in some or all of the others. Highly authalic maps are usually inefficient and nearly impossible to texture by hand. Planar maps that are easy to work with generally involve a lot of distortion, and if you try to eliminate the distortions by adding more projections you introduce seams, which both complicate the textures and lower the efficiency of the texture. As Ptolemy would have said, *exempla multiplicanda* (examples can be multiplied).

The point of all this is not to drive you to despair, but simply to provide a framework within which you can pick and choose the compromises you need to get things done. Here are a few *exempla*:

- If you're doing cinema models with lavish texture budgets, and you have a good 3D paint package you can go crazy with automatic mapping and damn the costs. But if you have to get an entire character onto a 512-pixel map sheet, you'll probably need to accept some smearing in the name of efficiency.
- If your primary limitation is texture memory, adding a few extra vertices will let you do a more satisfactory UV relax and disguise the smears you get from packing for efficiency; conversely, if you have a low poly model with a lavish texture budget you'll be able to indulge in a few extra projections, or perhaps even a full auto-map, to avoid smears.
- Working with inexperienced texture artists means you'll probably have to avoid texturing schemes with poor conformality. On the other hand, you can use 3D paint packages or projection baking to hide the conformality problems if you have reasonably authalic maps.

Ideally, having a good framework for judging the effects of different texturing methods will make it easier to decide quickly among the available options.

Now that we've got a good stock of impressive sounding technical terms and a basic idea of the trade-offs to be made in UV mapping, we can get down to talking shop again. We'll pick up UV mapping again in August with some tips and tricks for unusual UV mapping problems. ⁘

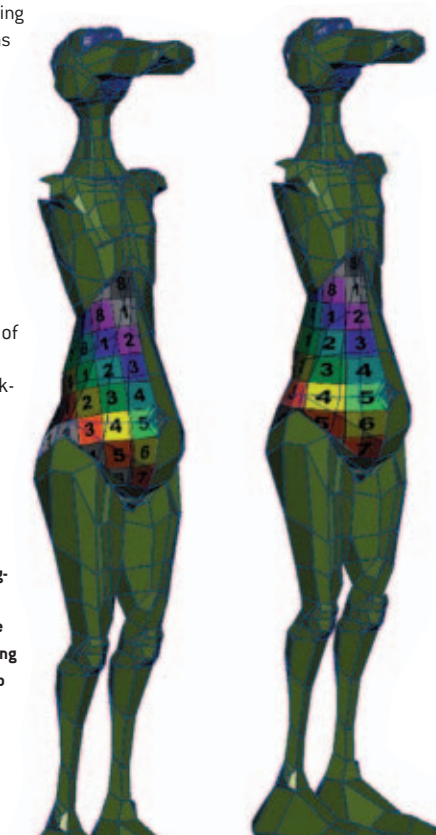


Figure 5. The mapping on the left hand figure is more authalic, but that on the the right is more conformal. Note how the belt line is perpendicular to the body's long axis near the navel but slopes as the map unfolds on the left hand mode.

"Alienware understands the needs of game developers; their systems are on the cutting-edge and consistently deliver the power we need to create great products."

- Reid Schneider, Lead Producer, Battlefield Vietnam

MJ-12™ DCC SYSTEMS

The MJ-12 line of professional workstations applies the uncompromising speed and power that Alienware is famous for to the digital content creation environment. Productivity is maximized by eliminating the critical workflow barriers that can delay your projects. The MJ-12 line features Seagate® Serial ATA hard drives and is certified to deliver a robust and stable solution for DCC applications. Backed by Alienware's fully owned and operated customer service center with toll-free 24/7 phone support, MJ-12 DCC systems offer the ultimate turnkey solution of speed, value, and reliability.

- Certified by leading 3D software developers for use with cutting-edge graphics applications and high-performance rendering software
- One of the most comprehensively validated, robust, and stable platforms on the market
- Specifically designed to maximize 3D animation, architectural visualization, and visual effects applications

Seagate

MJ-12™ PERFORMANCE[1] OPTION

Exclusive Alienware® Chassis with 460-Watt PSU
Intel® Pentium® 4 Processor with HT Technology 2.80GHz
Advanced 800MHz System Bus with 512KB L2 Cache
High-Performance Heatsink-Cooling System
Intel® 875 Chipset-Based Motherboard
1024MB PC-3200 DDR SDRAM at 400MHz
NVIDIA® Quadro™ FX 500 Ultra 128MB 8X AGP
Seagate® 160GB 7200RPM Serial ATA-150 with 8MB Cache
Black 52X/32X/52X CD-RW Drive
Optimized & Configured for High Performance

$1,929.00 or as low as **$58**/mo.[2] After $100.00 rebate.

ACCESSID: GDM0604PJ1

MJ-12™ EXTREME OPTION

Exclusive Alienware® Chassis with 550-Watt PSU
Dual Intel® Xeon® Processor 3.06GHz
Advanced 800MHz System Bus with 512KB L2 Cache
High-Performance Heatsink-Cooling System
Intel® E7505 Chipset-Based Motherboard
2GB PC-2100 DDR SDRAM at 400MHz
NVIDIA® Quadro™ FX 1100 128MB 8X AGP Dual DVI
Seagate® 320GB 7200RPM Serial ATA-150 RAID 0 Array 8MB Cache
Creative Sound Blaster® Audigy 2™ ZS 7.1 Advanced 24-bit Audio
Intel® 1000Mb and 10/100Mb Integrated Ethernet Adapters
Black 52X/32X/52X CD-RW Drive
Black 8X DVD±R/RW Drive
Microsoft® Windows® XP Professional
Optimized & Configured for High Performance
AlienCare: Advanced Technical Support including:
1-Year Personalized Online Support Portal with Chat and Forums
1-Year Toll-Free 24/7 Phone Support by Alienware Technicians
1-Year On-Site Warranty

$3,999.00 or as low as **$120**/mo.[2] After $100.00 rebate.

ACCESSID: GDM0604PJ3

BUSINESS LEASING AVAILABLE[3]

To receive an Instant Rebate on an MJ-12, MJ-12m, or Roswell, enter an AccessID at:

WWW.ALIENWARE.COM/GDM

ALIENWARE®
HIGH-PERFORMANCE SYSTEMS

For the latest pricing and promotions go online or call:
1.800.ALIENWARE | 1.800.254.3692

DAN LEE ROGERS

NECESSARY EVIL

THE RAPID RETREAT OF ADVANCED ROYALTIES

THE VIDEOGAME BUSINESS IS A HIGHLY competitive contest for the hearts and wallets of game consumers. And beneath the surface, beneath the art assets and programming code, are hundreds of business decisions critical to a game's success.

Today, independent developers are among those hit hardest by industry consolidation and the escalating development costs. These developers are the entrepreneurs responsible for such blockbuster hits as BATTLEFIELD 1942, MAX PAYNE, and RATCHET AND CLANK to name but a few. They include over 400 active, small companies around the world, and few realize just how valuable they are:

- In 2003, independents created nearly 60 percent of all games shipped in the U.S (BizDev independent research).
- Last year six of the top 20 PC games were developed by independents (NDP).
- Over 30 percent of last year's industry revenue was derived from games developed by independents (BizDev).

Nevertheless, over the past several years, independents have been earning less money for doing more work. Why? First, because there are more fingers in the pie. Beyond the fees paid to Sony, Microsoft, and Nintendo, publishers are also paying more for intellectual property (namely the NFL, J.R.R. Tolkien estate, Tiger Woods, and so on). Second, because the business environment reflects a supply and demand equation that currently favors publishers.

Today, independents rarely earn more than the narrow difference between what it costs them to develop a game (labor, equipment, and others) and what they are able to charge a publisher for it. For them, the term "royalty advance" is duplicitous.

Essentially, a royalty advance is an advance of money to be earned in the future, and which is then recouped by offsetting future royalties against it. The term is borrowed from literary law, but in our world it doesn't quite fit. Developers rarely receive a royalty advance. What they get is a production loan. And there's a big difference.

Last year Hillary Rodham Clinton received $8 million in "advance" for her book, *Living History*. Obviously, she didn't spend $8 million to create it (unless she is still using a 1978 Wang word processor), so the difference is an advance payment on future earned income. That's a royalty advance.

In the videogame business we use the same words, but we play by different rules. When money is loaned for the development of a game, it is called a royalty advance, but the publisher expects the funds will be used solely for game production. The unspoken truth is that for developers to earn a profit, they must overestimate the job slightly, and then hope they haven't miscalculated. Before they can make any more money, the publisher will earn three to four times its initial investment.

Here is how it works—as the owner of Playware, Inc. (a fictitious company), I am considering a standard PS2 deal:

- Development advance—$4 million ;
- Royalty percentage—12 percent of net income;
- Wholesale cost—$32.95.

Based on these terms (and most industry deals), my game must sell more than one million copies before I receive any additional money (royalties). In 2003, less than three percent of the console games sold over a million units, so my prospects are remote. Put another way, before I receive one nickel more, the publisher will generate over $45 million in gross revenue and earn over $13 million in net profits.

So is this fair? From a publisher's perspective, it is. The $4 million they invested up front is high risk/high reward capital. Their business model is built on a supposition that a few hits will pay for many failures.

At the same time, one questions whether a proven developer's hits should pay for a publisher's new ventures. In other industries, content creators and talent have been much more successful in taking a larger piece of the pie based on their abilities.

Harry Potter's creator J.K. Rowling was recently added to *Forbes* billionaire list. Like many sports and movie stars, she is able to demand an extraordinary amount of money for her creations. The difference between what J.K. Rowling spends to create a book and the royalty advance she receives before ever touching a key on her PC is astonishing.

Few in our industry are in a position to demand more. But as developers become more sophisticated in their business dealings, this rapid retreat of advanced royalties could reverse direction. After all, history has proven that over time content creators are often able to demand more than content distributors.

DAN LEE ROGERS

Dan Lee Rogers is president of BizDev, Inc., a leading business management firm specializing in the videogame market. Reach him at drogers@gdmag.com.

ALEXANDER BRANDON

PUSHING THE ENVELOPE

FOR YEARS GAME AUDIO PROFESSIONALS dreamed about having what is now available in commercial and in house proprietary engines. However, did these same pros dream about the kind of audio that would be benefiting from these new and improved feature sets? Audio technology in games is a basic consideration of a good audio manager, in addition to how audio assets interface with this technology. In this article we will examine ways of thinking ahead of the current technology curve.

For example, let's take a look at the musical genres most prevalent in games today:

ORCHESTRAL: Whether live or Gigastudio, the orchestra is taking its place in RPGs, 3D FPSs, MMORPGs, and real-time as well as turn-based strategy games.

LICENSED: Music from the pop charts is filling games faster than you can say "what are those marketing guys thinking?" Hip-hop is currently the most used licensed music, but folks such as Steve Schur at EA are trying to license the right music for the right game, and fend off the hordes of record companies looking to expand into a more lucrative distribution channel for the sake of profit alone. It's a damned hard job.

ELECTRONICA: Referred to most often (though not often correctly) as techno. There are dozens of kinds of electronic music, just as there are quite a few orchestral styles. Electronic music is easy to write and offers far more variety in terms of sonic texture than acoustically generated music. However, this same ease of use has caused what seems like endless repetition in style with little experimentation.

Now that these styles have worked their way into videogames just as much as movies and television, the uniqueness of game music production has dropped off. It's therefore up to us to develop what is still unique about games and apply that to music and sound: adaptability and interactivity.

Let's move on to discuss audio engine capabilities that are far from widespread use, if used at all. Brian Schmidt, audio director of Xbox, says he'd like to see the following features implemented:

- In-game editing and mixing
- Randomization
- Interactive music
- Variability
- 3D sound
- DSP effects processing
- Streaming
- "Don't have to bother the programmer much" philosophy
- Easy to make the game change how it sounds based on game events/variables (decided by sound designer, not programmer)

And here's a similar list from Pete Cole, audio programmers manager at Tao Group (based on the Intent sound system, **www.sseyo.com/tao_group/ave/iss/api.html**):

- Cross-platform
- Optimized on all supported platforms
- Play audio sample as spot effect, play audio file as stream, mixing multiple streams
- Support for all mainstream PCM-based audio formats (AIFF, WAV)
- Plug-in support for developing/new audio formats
- Wide range of non-PCM-based formats (MIDI, MOD, XMF)
- Support for Ogg/Flac formats to reduce media footprint and help reduce licensing costs
- Audio plug-in framework with lots of pre-coded plugins to make adding audio treatments (3D) very easy
- Lots and lots of sample code

Kurt Harland, composer/sound designer at Crystal Dynamic, points out some things that could be quite helpful if installed into non-audio middleware:

AI AND ANIMATION RANDOMIZATION: Animations and AI for characters and objects (especially enemies) needs some kind of automatically-generated randomization. For example, when four enemies confront you, and they all do exactly the same thing at the same time, it may be visually acceptable, but it creates terrible problems for audio. Four enemies chasing you in lock-step may look okay, but it can sound horrible. The animations and the AI need a bit of random timing changes. Footsteps that are, say, 15 frames apart should have the ability to receive automatic, on-the-fly interpolation, so they can range from 12 to 18 frames instead. That would alleviate the rhythmic attention-getting, and the stacking artifacts that tend to come up.

OBJECT AND ENVIRONMENT CREATION: Some aspects of audio need to be inherent to the mesh and the objects that the 3D artists are creating. Obviously, things like occlusion spaces and footstep textures need to be installed by the designers, not the audio people. It took an audio person three times as long as a 3D artist, not just because the 3D artists are more familiar with the work, but also because the audio person has to figure out what the 3D artist has done before he or she can add the audio stuff. Any mesh or object creation middleware needs to have audio tagging and flagging in it, as opposed to trying to create audio middleware that tries desperately to somehow take in the whole 3D mesh aspect of things.

Hopefully these concepts will give the audio community some more food for thought in future titles. With the information presented in last month's Audio Manager's Guidebook about prototyping, realizing these concepts should be a piece of cake.

ALEXANDER BRANDON

*Alexander has been involved with game audio since 1994 and is currently the audio manager at Midway in San Diego. You can e-mail him at **abrandon@gdmag.com**.*

NOAH FALSTEIN

GAME SHUI

SERIOUS FUN

IN THIS COLUMN, I'LL REPORT ON THE Serious Games Summit held in March at the Game Developers Conference and also respond to some more reader e-mail. The summit focused on games with purposes beyond entertainment—chiefly education, training, and research. We saw a mix of lectures, panels, case studies, and presentations of games that ranged from Dean for America, a grassroots political simulation game, to BioHazard (not to be confused with Capcom's Biohazard: Code Veronica), an Unreal-based game for training emergency personnel to deal with terrorism and toxic accidents.

The Summit was quite a success. Aiming for at least 100 attendees, it attracted nearly 300 and packed the room with lots of activity. Participants from the game development community as well as educators, academics, military professionals, health professionals, foundations, and more, avidly swapped business cards, information, and war stories (sometimes about real wars). There was a definite growing buzz and vitality—a welcome contrast to the cynical grousing about all the money being spent on license-based games, a typical GDC topic.

Designed with the Unreal engine, the Carnegie Mellon Entertainment Technology Center's BioHazard is a simulation game designed to help train emergency response teams.

A FEW SPECIFIC HIGHLIGHTS. I particularly enjoyed the comments by Wisconsin University's James Gee, the author of *What Video Games Have to Teach Us About Learning and Literacy*. He analyzed just why games are so effective at learning and training, and suggested they will play a large part in the future of education. I appreciated his point that it is necessary to have a range of game choices to allow players to learn the consequences of their actions—certainly a better approach than our politicians' attempt to sanitize games by limiting the players to "morally correct" actions only.

Jesse Schell of Carnegie Mellon University talked about creating BioHazard to train fire departments to respond to chemical and biological emergencies. He observed that the graphics turned out to be surprisingly important. When the game designers initially got the color of the uniforms wrong, the participants ("players" seems an inappropriate term for emergency personnel) scoffed and said it looked fake; conversely, when the verisimilitude in the visuals increased, they suddenly became immersed and excited. Clearly, attention to detail can make a big difference here, a lesson that should be familiar to flight simulator designers.

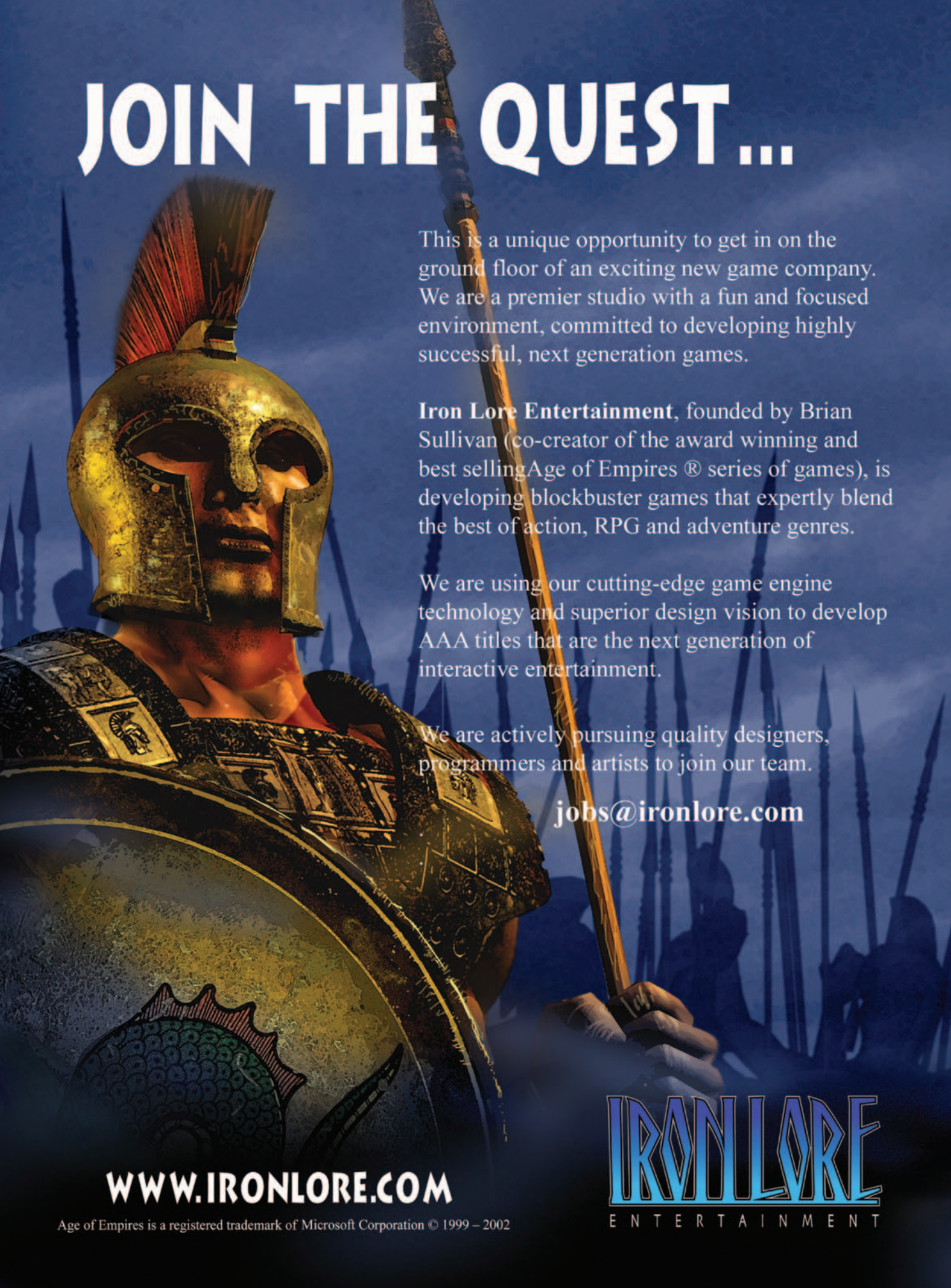
As a longtime advocate of making sure even learning games are fun, I was also persuaded to broaden my perspective to consider the potentials of games that are not necessarily fun. Compared to common, non-interactive forms of learning like classroom lectures, games feature many advantages: for instance, the ability to tailor the learning to the individual, inexpensive mass distribution, and clear and accessible depiction of complex systems. I quoted Ben Sawyer on this point in March (Game Shui, "The Flow Channel"). At the time, I must admit I was skeptical, but the Summit convinced me that focusing on fun to the exclusion of other positive qualities could trigger knee-jerk negative reactions from many of the people who fund these projects. Also, it is important to consider that the alternative for a game-based trainee is, for example, a workbook and not a session of Doom 3. I am still convinced the optimum solution is to achieve the teaching goals and be fun, but we may need to be patient—the generation that has grown up with computer games is only beginning to move into positions with authority over training budgets.

MAILBOX. I got an interesting e-mail about my February 2004 column (Better by Design, "To Globalize or Localize"), where I said that German and Austrian developers I've worked with "generally tend to make games that focus more on details, especially on opportunities for players to indulge in micromanagement." The letter, from Jochen Hamma, an independent producer/designer/teacher, recounted a discussion in Berlin at a game critique session. The remark from a participant, a student named Sebastian Bombera, could be summarized as follows: Germans possess a historical and climatic need to be in control down to the smallest level, which leads to an obsession with micromanagement and a desire for non-linear, personally customizable stories. What I think would be particularly interesting is to not just state the rule but analyze what it is about the German character—or should I say Zeitgeist?—that led them to prefer this kind of game.

I won't go into details about Jochen's analysis that focuses on the variability of Central European temperature extremes and other environmental factors, but I am struck by the similarity to some of the academic analyses of games I heard at the Serious Games Summit. After decades of being treated as inconsequential children's matters by many, games are finally receiving some scholarly attention that takes them—dare I say it—seriously.

NOAH FALSTEIN

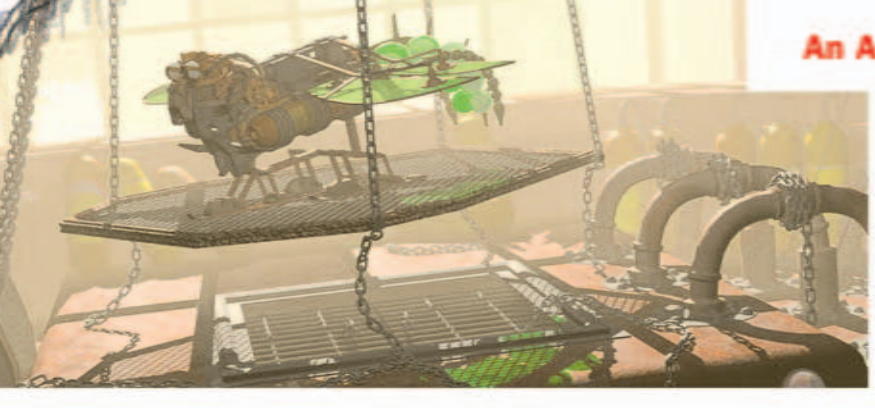
Noah is a 24-year veteran of the game industry. His web site, ***www.theinspiracy.com****, has a description of The 400 Project, the basis for these columns. Also at that site is a list of the game design rules collected so far, and tips on how to use them. You can e-mail Noah at* ***nfalstein@gdmag.com****.*


JOIN THE QUEST...
This is a unique opportunity to get in on the ground floor of an exciting new game company. We are a premier studio with a fun and focused environment, committed to developing highly successful, next generation games.
Iron Lore Entertainment, founded by Brian Sullivan (co-creator of the award winning and best sellingAge of Empires ® series of games), is developing blockbuster games that expertly blend the best of action, RPG and adventure genres.
We are using our cutting-edge game engine technology and superior design vision to develop AAA titles that are the next generation of interactive entertainment.
We are actively pursuing quality designers, programmers and artists to join our team.
jobs@ironlore.com
WWW.IRONLORE.COM
Age of Empires is a registered trademark of Microsoft Corporation © 1999 – 2002
IRONLORE
ENTERTAINMENT

You know where you want to be.

You have the talent, ideas and creativity. You love animation and video games. You know there are careers out there that allow you to bring your concepts and characters to life. How do you transform your abilities into the skills that you need to start a professional career in the animation or video game industries? How do you get from here to there?

creativity works



An Art Institute can help you get there.

Starting at an Art Institute, you will hit the ground running. Our faculty, many of whom are industry professionals, bring their experience into the classroom to help you develop and focus your skills and creativity. An Art Institute education is intensive, practical and designed specifically to help you begin your career. Call us today and put your creativity to work.

Degree programs offered | **Game Art & Design** (BS) | **Media Arts & Animation** (BS) | **Visual & Game Programming** (BS)*

The Art Institute of California–Los Angeles
2900 31st St.
Santa Monica, CA 90405
888-646-4610
www.aicala.artinstitutes.edu

The Art Institute of California–Orange County
3601 West Sunflower Ave.
Santa Ana, CA 92704
888-549-3055
www.aicaoc.artinstitutes.edu

The Art Institute of California–San Diego
7650 Mission Valley Rd.
San Diego, CA 92108
800-591-2422
www.aicasd.artinstitutes.edu

The Art Institute of California–San Francisco
1170 Market St.
San Francisco, CA 94102
888-493-3261
www.aicasf.artinstitutes.edu

Financial aid available to those who qualify. Individual job assistance. BS=Bachelor of Science. *Offered at San Francisco only. Student Artwork: Wireframe head: Shiew Yeu Loh, The Art Institute of California-San Francisco; Rally Racer: Phi Tran, The Art Institute of California-Orange County; Ornithopter: Peter Gend, The Art Institute of California-Los Angeles.

HANDHELDS COLLIDE

CONTINUED FROM PG 6

Activision, Atari, Ubisoft, Majesco, Capcom, Konami, Namco, Koei, and Square Enix. Nokia's N-Gage also had a big reveal, showing off its extreme makeover. The svelter N-Gage QD corrects most of the problems mentioned by developers and players by shedding some weight, adding a game card slot to its exterior, and dropping the price by a third.

Vicarious Visions is developing games for all three handhelds, and according to VP of business development Dwight Cheu, they each feature unique characteristics that serve specific target markets. The new platforms also make the development process more rewarding, since "new input devices are a great way to introduce fresh gameplay mechanics—things you just can't do with other systems," according to senior programmer Matthew Conte. Conte is also upbeat about building games for the PSP's 16:9 screen, saying that "the guaranteed widescreen aspect ratio gives you a more cinematic feel, which makes for a greater sense of perspective and ultimately better games."

While most of the attention was on the PSP and Nintendo DS, Nokia previewed what producer Shane Neville describes as the first mobile MMOG, POCKET KINGDOM, taking full advantage of Nokia's connectivity experience. Kurt Southworth, product manager for Motorola's Freespace division, also demonstrated their work in reducing latency on racing games played over their Game Boy Advance wireless adapter. Finally, the cult following of the Tapwave Zodiac, a 3D multimedia handheld that's already available online today, stands to see its ranks swell as the company announced its first brick-and-mortar sales agreement, with CompUSA.

—Jamil Moledina

WHAT HAVE YOU PLAYED

CONTINUED FROM PG 10

training tools, not just in the medical profession but in other disciplines as well, it can help a great deal in recruiting generations X and Y."

Rosser's study provides another example of the value of serious games initiatives, pointing out that videogames as learning tools offer a number of advantages: they're widely available, portable, and relatively inexpensive. Furthermore, in medical cases, they help bypass the ethical dilemmas involved in the use of live subjects for training.

"It's reported that currently only 15 percent of surgeons routinely perform advanced laparoscopic procedures," says Rosser. He hopes to dramatically increase that number—with the help of videogames, if that's what it takes.

—Kenneth Wong

LAWMAKERS' TIZZY

CONTINUED FROM PG 8

As for the movie business, MPAA President Jack Valenti says, "The entire rostrum of the rating program rests on the assumption of responsibility by parents. If parents don't care, or if they are languid in guiding their children's moviegoing, the rating system becomes useless."

Douglas Lowenstein, president of the Entertainment Software Association, says the problem is with parenting, not policy. "What people seem to ignore is the old maxim, 'You can lead a horse to water, but you can't make him drink,'" he says. The ESRB's system, Lowenstein adds, is one of the best ratings system in the country, citing Senator Joe Lieberman's and the Federal Trade Commission's consent. He says the call for legislation is unnecessary because the problem lies at the home, not at the state level. "You can't pass a bill to enforce good parenting," Lowenstein says.

Parents must learn to trust the ESRB's system, which more thoroughly evaluates content than the proposed governmental system in Manitoba. The baby-boomer generation of parents (and lawmakers) at the helm have always been a part of the ubiquity of movies, whereas the gaming industry presents new and unfamiliar complications, namely, a child's physical interaction with a game. In California, Assemblymember Yee—a certified child psychologist—points most of his concerns to how first-person shooter games and other interactive violence might damage a child's disposition.

But the statistics just aren't solid. The juvenile arrest rate for violent crime in 1999 was 36 percent below its peak in 1994, according to the Office of Juvenile Justice and Delinquency Prevention. In the United States, violence tends to rise in segregated areas of low socio-economics, not in communities where families can afford computers and game consoles.

THE KEY IS COMPLIANCE

"Retailers representing more than 90 percent of the market pledged last November to implement carding policies for Mature-rated games no later than the end of 2004," Lowenstein says. "This progress toward effective voluntary enforcement ... makes the proposed ordinance unnecessary."

And if a nine-year-old is combing the shelves of a videogame retailer alone, with a $50 bill in his pocket, maybe we've got bigger issues to worry about than whether the M-rated games are at his eye level.

—Jill Duffy

ADVERTISER INDEX

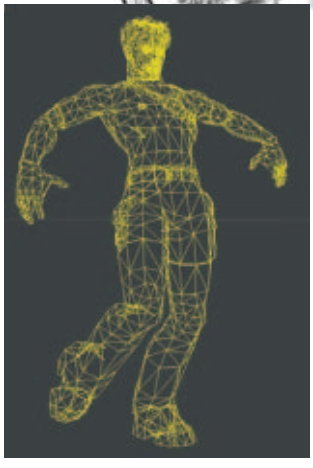
COMPANY NAME	PAGE	COMPANY NAME	PAGE	COMPANY NAME	PAGE
Academy of Art University	51	Ironlore	49	Swingin' Ape	50
Alienware	43	Kaydara	C2	Techexcel	11
Araxis	27	Konami	50	Time Gate Studios	48
The Art Institute Int'l	53	Microsoft Direct X	3	University of Advancing Technology	52
Atari	48	New Pencil	54	Vancouver Film School	52
Collins College	52	Numerical Design	5	Vicon Motion Systems	33
Datascope Recruitment	54	Perforce	17	Xoreax	27
Full Sail Real World Education	66	RAD Game Tools	C4		
Havok	31	Softimage	C3		

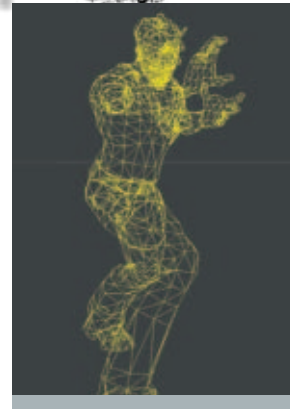
Game Developer (ISSN 1073-922X) is published monthly by CMP Media LLC, 600 Harrison St., San Francisco, CA 94107, (415) 947-6000. Please direct advertising and editorial inquiries to this address. Canadian Registered for GST as CMP Media LLC, GST No. R13288078, Customer No. 2116057, Agreement No. 40011901. **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.

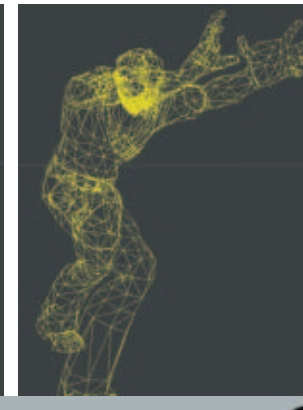
A THOUSAND WORDS

WHILE GLYPHX GAMES' UPCOMING ADVENT RISING (PUBLISHED BY MAJESCO) DRAWS depth from a collaboration with legendary science fiction novelist Orson Scott Card, according to project lead Donald Mustard, every frame of the game is designed to look more like a graphic novel. Mustard chose to slightly exaggerate the anatomy of the protagonist Gideon, so the character would have a greater sense of weight and appear more like a superhero. At the same time, the world around him is bright and vibrant, so Gideon is colored darkly to make him pop off the screen. Furthermore, colors and shapes define the composition of each scene, as opposed to building them as collections of minute details. The sequence below represents six months of work—to design, model, texture, mocap, and tweak—using a combination of Maya, Motionbuilder, 3DS Max, and After Effects. While Mustard himself drew the sketches, set up the camera work, and performed the mocap, a total of 16 developers worked on the art of the sequence, as credited below.

– Jamil Moledina







Artists:
Lanny Barrows
Brandon Beckstead
Cameron Dayton
Matt Judd
Gavan Knowlton
Boyd Lake
Bert Lewis
C.J. Markham
Donald Mustard
Geremy Mustard
Shane Olson
Christian Perry
Christopher Salmon
Todd Sheridan
Nathan Trewartha
Jared Trulock


SOFTIMAGE® | XSI®
Jun Takeuchi on XSI
竹内 潤　XSI を語る
"We continued to work with Softimage on Onimusha 3 because we knew the project would require reliable support as well as leading edge technology. With its flexible character creation workflow, profound tool integration and unique Animation Mixer, SOFTIMAGE | XSI has significantly improved our workflow. And, of course, Softimage's support and responsiveness continue to be assets as we push the boundaries of game design and development."
Jun Takeuchi, Producer
Onimusha 3
「鬼武者3でソフトイマージと共に仕事をし続けたのは、このプロジェクトにおいて先進のテクノロジーとサポートが必要になると考えていたからです。またXSI が提供する柔軟なキャラクター構築のワークフロー、強力なツールの統合、そしてアニメーションミキサーは我々のワークフローを変革するほどのインパクトを与えてくれました。ソフトイマージからのプロダクション・サポートや責任感ある対応が積み重なり、より一層のクオリティアップが図れたと考えています。」
鬼武者3 プロデューサー 竹内 潤
I am it. the future of animation.
softimage.com/games
Avid

Just Drop Us In!

The **Miles Sound System** *provides the most sophisticated sound for the PC, Mac, Xbox, and Linux. It supports 2D, 3D, streaming, MIDI with DLS, the world's fastest MP3 decoder (with patent rights), internet voice chat, software reverb, and much more!*

Pixomatic *is our powerful software renderer for PCs. Essentially, it is a high-end DX7 video card in software form! MMX, 3DNow, and SSE optimized assembly is generated at run-time. Don't let low-end hardware or bad video drivers hurt your game's sales!*

Granny *is a powerful toolkit for building all kinds of interactive 3D applications. It features the most efficient and flexible content exporters, data manipulation, normal mapping and run-time dynamic 3D animation system you'll find anywhere.*

Bink *is the finest video codec for games! It provides better than DVD quality at half the data rate along with a perceptually lossless 10:1 audio codec in a simple and clean API. It is available for PC, Mac, Xbox, GameCube, Linux, and now Sony Playstation 2!*

©

THE BEST IN GAME DEVELOPMENT TECHNOLOGY

www.radgametools.com **425.893.4300**