



CMP

United Business Media

▶▶ **PRODUCT REVIEW** MAYA 7 UNLIMITED FOR MAC OS X

NOVEMBER 2005

game**developer**

THE LEADING GAME INDUSTRY MAGAZINE

▶▶ **PURE SCIENCE**

PORTABLE PHYSICS
FOR WIPEOUT PURE

▶▶ **BUT SERIOUSLY**

JAMES PAUL GEE ON
WHAT GAMES TEACH US

▶▶ **POKER FACE**

HOLD'EM-FRIENDLY
AI SHARKING



**SCALING THE CABAL FOR
VALVE'S
HALF-LIFE 2**

changing the face of 3D



Maya® 7, the latest release of the award-winning 3D software, is packed with innovative new features allowing you to realize your creative vision faster and more easily than ever before.

Capitalizing on Alias MotionBuilder® technology, Maya 7 makes character animation easier and more accurate. Other improvements such as advanced render layering and new modeling, texturing and effects tools help you achieve more with Maya.

Join us on Oct 31, 2006 in Washington D.C. at the **Serious Games Summit** for a *Maya 7 Tips and Tricks* sponsored seminar. To register or for more information visit www.seriousgamesummit.com

Image created by Marco Melis (www.3dspace.com)

© Copyright 2006, Autodesk, Inc. All rights reserved.

Autodesk, the Autodesk logo, Maya and MotionBuilder are registered trademarks and the Maya logo is a trademark of Autodesk, Inc. in the United States and/or other countries.

Maya™ 7

Alias | www.alias.com



20 SCALING THE CABAL: VALVE'S DESIGN PROCESS FOR CREATING HALF-LIFE 2

The sometimes-mysterious G-Men at Valve Software outline the process, workflow, and design techniques that brought us HALF-LIFE 2, one of the seminal PC games of the past few years, updating us on the evolution of the unique Cabal work management system developed for the original HALF-LIFE, and touching on everything from story development to voice acting challenges.

By Brian Jacobson and David Speyer

DEPARTMENTS

2 GAME PLAN *By Simon Carless*

Serious Fun

4 HEADS UP DISPLAY

Words of wisdom from Tokyo Game Show, Softimage XSI 5, NYFD gets serious, and more.

7 SKUNK WORKS *By James Alguire*

Alias' Maya 7 Unlimited for Mac

48 A THOUSAND WORDS

BreakAway's FREE DIVE

FEATURES

9 WHAT GAMES HAVE TO TEACH US: AN INTERVIEW WITH JAMES PAUL GEE

James Paul Gee of the University of Wisconsin is one of the most vocal proponents of games for learning. As an academic, his views have been a welcome contrast to the current anti-game sentiment in the conservative public view. In an interview with *Game Developer*, Gee discusses his thoughts on digital interactive entertainment, the importance of games to states' economies, and perhaps most importantly, what game developers can do to improve the industry.

By Brandon Sheffield

14 HOVERING ON A HANDHELD: THE PHYSICS BEHIND WIPeOUT PURE

The physics engine for WIPeOUT PURE was built from the ground up, and as a PSP launch title, it had to be done in a relatively short time. Martin Linklater of Sony's Studio Liverpool shares the experience of creating a speedy physics engine for handhelds without compromising the gameplay, or shattering his programmer's pride.

By Martin Linklater



9



14

COLUMNS

30 THE INNER PRODUCT *By Mick West*

Texas Hold'em AI

[PROGRAMMING]

35 PIXEL PUSHER *By Steve Theodore*

Anatomy for Animators: A Leg to Stand On

[ART]

38 BUSINESS LEVEL *By Careen Yapp*

Working With Hollywood

[BUSINESS]

39 AURAL FIXATION *By Alexander Brandon*

Bringing Down the House

[SOUND]

41 GAME SHUI *By Noah Falstein*

Stealth Education

[DESIGN]



SERIOUS FUN

THIS MONTH'S ISSUE OF *GAME DEVELOPER*

coincides with and debuts at the 2005 Serious Games Summit in Washington D.C. Therefore, you'll find that the some of the contents of this issue have a mild, tangy "serious" flavor.

It helps if you actually concur with the coining of the term "serious games" to describe games for education, health, military, and other sub-sectors of the industry. As academic James Paul Gee claims in his interview (pg. 9), perhaps there's an essential anachronism in the phrase that grates on the more sensitive among us. However, as an overarching term, it fits better than anything else devised thus far. We like it, and we're sticking to it.

HEADCRABS, BOOMSTICKS

The big news for this month: a developer-penned analysis of Valve Software's seminal *HALF-LIFE 2* (pg. 20).

Many months in the making, and the biggest article we've run for some time, Brian Jacobson and David Speyer's piece follows a feature that ran in the December 1999 issue of *Game Developer*, which analyzed the Cabal Process used to create *HALF-LIFE*.

Moving from the first game to its sequel required significant tweaks to the Cabal Process, and Jacobson and Speyer discuss, in detail, the development methods that created one of the most critically acclaimed games of all time.

W-W-WIPEOUT!

While the high-end PCs that run *HALF-LIFE 2* had a little more leeway in terms of processor power for vital in-game systems such as physics, Sony Liverpool's marquee PSP game *Wipeout Pure* was operating under tighter restrictions. In this featured technical article (pg. 14), senior programmer Martin Linklater discusses the work he single-handedly accomplished implementing physics, handling, and collision for Sony's popular title.

Considering that near the end of the project the game was using just 10 percent of the CPU to perform the physical simulation for all eight ships at 30fps—which included the collision tests, physics integration, and ship handling code—Linklater's work on the game was no mean feat.

GEE, POKER, HOLLYWEIRD

The rest of this issue is on the action-packed side, too. Games for learning proponent, author, and academic James Paul Gee expounds on some pretty intriguing topics in his previously mentioned in-depth interview. Code columnist Mick West has some excellent AI lessons learned from his work on a new console poker game (pg. 30), and this month's Business Level is from D3's Carreen Yapp, discussing what happens when Hollywood talent collides with the game industry, and how to make the best of it (pg. 38).

INTERNET WELCOMES CAREFUL DRIVERS

Finally, I want to share some of the improvements we've been making to our editorial operations in the wild, woolly world of the internet.

As for *Game Developer* (www.gdmag.com), we continue to roll out a digital edition, which allows digital subscribers to read all the content online every month, but still in the elegant format of magazine pages. And access to back issues is rolled into the bargain. We've also recently introduced single-issue digital sales, so you can read any one particular issue without tracking down a physical copy.

As for our sister web site (www.gamasutra.com), which wholly or partly shares a number of editors with the magazine, Gamasutra.com has been expanding recently, and now features five exclusive features and five exclusive columns per week, and as many as 15 news updates per day. Thanks to the continual development of new markets, the site now has a daily email newsletter plus fresh newsletters concentrating on mobile games, casual/indie games, serious games, and career-related issues. Look for their expansion into their own websites over the next few weeks and months. Needless to say, all this means that you have plenty of extra information to supplement your monthly copy of the magazine. Use it wisely!

Simon Carless, editor

EDITORIAL

EDITOR

Simon Carless scarless@gdmag.com

MANAGING EDITOR

Jill Duffy jduffy@gdmag.com

ASSISTANT EDITOR

Brandon Sheffield bsheffield@gdmag.com

ART DIRECTOR

Cliff Scorsio cscorsio@gdmag.com

CONTRIBUTING EDITORS

Alexander Brandon abrandon@gdmag.com
Noah Falstein nfalstein@gdmag.com
Steve Theodore stheodore@gdmag.com
Mick West mwest@gdmag.com

ADVISORY BOARD

Hal Barwood	Designer-at-Large	Dave Pottinger	Ensemble Studios
Ellen Guon Beeman	Monolith	George Sanger	Big Fat Inc.
Andy Gavin	Naughty Dog	Harvey Smith	Midway
Joby Utero	Luxoflux	Paul Steed	Microsoft

ADVERTISING SALES

NATIONAL SALES MANAGER

Afton Thatcher athatcher@cmp.com t: 415.947.6217

SENIOR ACCOUNT MANAGER, EAST COAST, EUROPE & EASTERN CANADA

Ayrien Machiran amachiran@cmp.com t: 415.947.6224

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

Susan Kirby skirby@cmp.com t: 415.947.6226

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

Nick Geist ngeist@cmp.com t: 415.947.6223

ACCOUNT MANAGER, GLOBAL RECRUITMENT/EDUCATION & TEXAS

Aaron Murawski amurawski@cmp.com t: 415.947.6227

ADVERTISING PRODUCTION

ADVERTISING PRODUCTION COORDINATOR

Kevin Chanel

Reprints Julie Rapp jarapp@cmp.com t: 510.834.4752

GAME GROUP MARKETING

DIRECTOR OF BUSINESS STRATEGY

Michele Maguire

DIRECTOR OF MARKETING

Tara C. Gibb

CMP GAME GROUP

VP, GROUP PUBLISHER APPLIED TECHNOLOGIES

Philip Chapnick

CONFERENCE DIRECTOR, GAME GROUP

Jamil Molehdina

CONFERENCE MANAGER, GAME GROUP

Evelyn Donis

EXECUTIVE WEB PRODUCER

Peter Leahy

EDITOR, GAMASUTRA.COM

Simon Carless

FEATURES EDITOR, GAMASUTRA.COM

Quang Hong

CIRCULATION

CIRCULATION DIRECTOR

Kevin Regan kregan@cmp.com

CIRCULATION ASSISTANT Manager

Jessica Ward jward@cmp.com

CIRCULATION COORDINATOR

Miguel Mendiola mmendiola@cmp.com

CIRCULATION ASSISTANT

Michael Campbell mcampbell@cmp.com

CIRCULATION ASSISTANT

Adrea Abidor aabidor@cmp.com

SUBSCRIPTION SERVICES

FOR INFORMATION, ORDER QUESTIONS, AND ADDRESS CHANGES

t: 800.250.2429 f: 847.763.9606 e: gamedeveloper@halldata.com

INTERNATIONAL LICENSING INFORMATION

Mario Salinas

t: 650.513.4234 f: 650.513.4482 e: msalinas@cmp.com

CMP MEDIA MANAGEMENT

PRESIDENT AND CEO,

Gary Marshall

EXECUTIVE VP & CFO,

John Day

EXECUTIVE VP & COO,

Steve Weitzner

EXECUTIVE VP, CORPORATE SALES & MARKETING,

Jeff Patterson

EXECUTIVE VP, HUMAN RESOURCES,

Leah Landro

CHIEF INFORMATION OFFICER,

Mike Mikos

SENIOR VP, OPERATIONS,

Bill Amstutz

SENIOR VP, CMP INTEGRATED MARKETING SOLUTIONS,

Joseph Braue

SENIOR VP AND GENERAL COUNSEL,

Sandra Grayson

SENIOR VP, COMMUNICATIONS,

Alexandra Raine

SENIOR VP, CORPORATE MARKETING,

Kate Spellman

VP GROUP DIRECTOR OF INTERNET BUSINESS,

Mike Azzara

CORPORATE DIRECTOR, AUDIENCE DEVELOPMENT,

Shannon Aronson

CORPORATE DIRECTOR, CREATIVE SERVICES,

Kellie Ferris

CORPORATE DIRECTOR, PUBLISHING SERVICES,

Marie Myers

CORPORATE DIRECTOR, AUDIENCE DEVELOPMENT,

Michael Zane

PRESIDENT, CHANNEL GROUP,

Robert Falettra

PRESIDENT CMP ENTERTAINMENT MEDIA,

Tony Keeffe

PRESIDENT CMP HEALTHCARE MEDIA,

Vicki Masseria

VP GROUP PUBLISHER ELECTRONICS,

Paul Miller

VP GROUP PUBLISHER ENTERPRISE,

Fritz Nelson

VP GROUP PUBLISHER SOFTWARE DEVELOPMENT MEDIA,

Peter Westerman



Game Developer
is BPA approved



United Business Media

WHAT'S HAPPENING IN YOUR WORLD?



FIND OUT!

BreakAway uses the latest video game technology to create compelling, highly immersive virtual worlds.

Whatever your training needs are, whatever your world looks like – we have the tools, the skills and the experience to make it real.

In a bunker, the hospital ER or on the streets during a non-violent protest, it's *ALL* there!

Come see how we can make a difference
in your world.

BREAKAWAY
ltd.

www.breakawayltd.com

BreakAway, Ltd. Copyright©2005 - All Rights Reserved



TGS 2005 GOES NEXT-GEN



PHOTO BY SIMON CARLESS

THE TOKYO GAME SHOW 2005, HELD SEPTEMBER 16–18, WAS ATTENDED IN RECORD numbers this year, with a total of 176,065 attendees—16,000 more than the previous year. There were 131 exhibitors, also the largest number for Japan's seminal electronic games show, though a number of smaller companies no longer had their own booths but were instead drawn into larger publisher or console-maker spaces. The advent of the new console generation was one of the biggest draws, with major announcements from all camps.

Microsoft had the most tangible next-gen offerings, with multiple playable games for the Xbox 360 from a variety of third parties. The Japanese market has traditionally been Microsoft's weakest for the Xbox, and the company is keen to up its status in the region, wooing many developers to the cause. In addition to longstanding announcements from Mistwalker and Q? Entertainment, Microsoft has also individually targeted select smaller niche-based developers and publishers, such as Hudson Soft and Hamster Corporation, in order to get a true cross-section of the Japanese market. Robbie Bach, chief Xbox officer for Microsoft, stated in his TGS keynote that the company intends to "succeed as famously here as we did not succeed in the previous generation."

Sony's next-gen unveilings were almost exclusively in movie form, with impressive visuals from a variety of companies. Konami's METAL GEAR SOLID 4 was particularly striking and was also the only game showing real-time visuals (by designer Hideo Kojima) outside of Sony's movie pavilion, proving that the PlayStation 3 may well be as powerful as the company claims. With the PlayStation 3 further down the pipeline, Sony focused on PlayStation 2 and PSP software, showing a much different face than last year's focus on action games, with RPGs (ROGUE GALAXY) and adventure titles (SHADOW OF THE COLOSSUS) taking much of the PlayStation 2 limelight. Quirky minigame-style offerings were given a large portion of the PSP booth, presumably to combat the DS's very strong showing in Japan's casual markets.

Nintendo's major announcement was its Revolution controller, which the company hopes will help capture even more traditional non-gamers. Nintendo does seem to be orbiting in its own sphere of the gaming world, choosing not to tackle the next generation from a graphical standpoint, but rather a design- and accessibility-based platform. This strategy may well see Nintendo through to opening up the market to a whole new audience. In his TGS keynote, Nintendo president Satou Iwata explained the logic behind the new direction. "For the future of video game business, we have to expand the market. We need to get back to the basics," adding the bold statement, "If we can't expand the market, all we can do is wait for the market to die."

Time will tell if the Sony and Microsoft strategy of appealing to the market through the power of visuals and big budgets, as Hollywood has, or the Nintendo strategy of accessibility and casual-market targeting will win out. But with all three major console makers now relative veterans of the industry, if each company succeeds with its expansive goals, there may be room for all three.

—Brandon Sheffield

TALK OF THE FLOOR

GAME DEVELOPER SCoured TOKYO GAME SHOW'S FLOOR for unique developer and publisher perspectives from those in the trenches of Japanese game development, highlighting those voices heretofore unheard in the English-language press. The following are a few choice quotes from our findings.

—Interviews and translations by Brandon Sheffield

"It's true that Japanese developers don't usually interface directly with each other—it's quite closed, on the surface. But it doesn't mean that we don't communicate. Developers have their own blogs now, and on top of that there are more people moving around in the industry than there used to be, and practices and methodology can be shared that way, too."

—Makoto Iwai, manager of international business, Video Game Dept., Bandai

"We felt like the PSP's future was more secure, that's why we're porting [classic RPG] LANDSTALKER to it. Of course with our new LANDSTALKER games, we'll make them for PlayStation 2 or 3.... We lost a lot of our development staff after the 16-bit days, but now we've got most of them back, and can move forward with the series again."

—Shimpei Harada, vice president, Climax, Inc.

"Right now [non-Japanese] Asian fans really like Japanese products and culture. They want the package in Japanese, manual in Japanese, they want everything to be in Japanese, or Japanese style. Japan is cool and popular in China, and right now it seems like they don't want anything else."

—Takeshi Kimura, senior chief, Overseas Marketing Dept. for SNK Playmore

"Over time games have gotten harder and harder, with more difficult rule sets, though of course they also look nicer. So at the same time, a lot of people are nostalgic for the older, simpler style of play. There are lots of these types of users [who are scared away from complex titles], and someone needs to appeal to them."

—Takashi Ishii, Planning and Creation Dept., Hamster Corporation

"As an independent developer in Japan, relationships with publishers are really tough. It's especially hard to hold on to your original IP. In Japan, if a company publishes your game, they will definitely claim your IP. That's why we can't make TENCHU anymore. We're sharing IP with the publisher for [our latest game] SHINOBI DO, but we have to share more of the money. Otherwise we can't keep control of our own properties."

—Takuma Endo, president, Acquire

"We've had to change our workflow, it's no longer linear. Because we've had so little time to work in this, and with so much to do for [RUMBLE ROSES XX] as our first next-gen title, we've really had to change things ... and only some of the people on our CG teams knew how to do new techniques like normal mapping, so we've had to allow them time to experiment."

—Akari Uchida, producer, 5th production division, Konami

GAME TRAINS FDNY

LIEUTENANT TONY MUSSORFITI of the Fire Department of New York is something of a game master. As a hazardous materials technician and training instructor, he creates scenarios for HAZMAT: HOTZONE, a game that trains New York's finest how to deal with emergency situations, such as terrorist attacks involving weapons of mass destruction or hazardous materials.

The game's development began four years ago at the Entertainment Technology Center at Carnegie Mellon by Shanna M. Tellerman, who is producing the game using Unreal Engine, and faculty advisor Jesse Schell, who's also chair of the IGDA.

"The greatest challenge in designing HAZMAT: HOTZONE was to create a tool that would truly be useful in a classroom setting while still maintaining the immersive environments of a video game," says Tellerman. "In order to engage the first responders in a training session, certain elements had to be highly realistic."

HAZMAT: HOTZONE was designed to supplement, not replace, field exercises, which are expensive to conduct [large-scale field exercises are typically run only once or twice



per year] and lectures. The game allows an instructor to set up an emergency situation, initiate the game, and then pause it or trigger new actions during play so that the activity can change at any moment.

"It was also necessary to constantly design for the fact that we are not hazmat experts, and therefore we needed to create a tool that would allow the experienced instructors a mechanism for transferring their expertise to a new generation," says Tellerman. The players also wear radios to communicate with one another, as

they would on the job.

Though technology, in some learning environments, can hinder students, Tellerman says firefighters had few impediments in adding the game to their curriculum. "The fire departments are currently training a new generation of fire fighters. This new generation has grown up in an immersive world of video games and computers. At the same time, we have kept highly aware that in order to gain full acceptance into the training curriculum, the software would have to appeal to

the experienced generations [of firefighters] who are less comfortable on computers. This is why we designed an instructor-centered game experience in which the experienced instructor controls the training session from start to finish, and therefore does not feel as though the computer is taking charge of the lesson," she says.

Unfortunately, HAZMAT: HOTZONE hasn't been well supported thus far. "We had been hoping that the Department of Homeland Security would see this as an opportunity to get behind the development of innovative training techniques that could be distributed for nationwide use, but we have not had the support we had been hoping for," says Tellerman.

Still, Tellerman and her team are working toward distributing the game for free to first-responders.

"When the students finish a training session in which they have discussed in depth the methods of responding to chemical attacks in the subway and then finish by asking if they can play again, you really know you have hit onto something huge for the future of education." —Jill Duffy

AVID DEBUTS XSI 5

AVID HAS ANNOUNCED THE WORLDWIDE AVAILABILITY OF Softimage XSI version 5.0, the latest edition of the company's signature 3D animation software.

Originally unveiled at the Siggraph tradeshow in August, XSI version 5 software includes a wide range of new features, such as non-destructive character tools and a comprehensive set of migration tools for Maya users, plus new interface layouts and navigation

modes that let artists transfer their existing skills and muscle memory to XSI.

Encompassing a broad range of new features, XSI 5 includes the GATOR attribute transfer system for re-purposing properties and animation between models; native 64-bit support for XSI; mental ray 3.4 software to create and render extremely complex scenes, and a new gigapolygon core that leverages multi-processor and multi-core platforms with a new memory management system.

In addition, XSI has integrated the Ageia physX physics simulation engine, and delivers high-performance physical simulation, adding new high-precision actual-shape collision handling. It also now ships with the Integrated Tools Development Environment, which is a single unified development environment to create, manage, and deploy all tools, plug-ins and workgroups.

—Simon Carless



CALENDAR

Montreal International Game Summit

Mount Royal Centre

Montreal

November 2–3, 2005

Cost: \$25–\$380

www.montrealgamesummit.com

America's VideoGame Expo: Philadelphia

Fort Washington Expo Center

Fort Washington, PA

November 12–13, 2005

Cost: free

www.vgXpo.com

Building a great User Interface doesn't have to suck.



Let Anark Gameface improve your entire UI production pipeline.



Artist Benefits

Freedom to create great front-end and in-game UIs without heavy programmer involvement.

Preview your work real-time.

Work interactively with your existing 2D & 3D assets.

Utilize a mature and artist-friendly authoring tool.

Programmer Benefits

Integrate once, and then walk away.

Usable on most game engines with current and next-gen game consoles.

Dedicated support structure.

Work with a full featured SDK.

Producer Benefits

Liberate your team to produce a higher quality title.

Dramatically improve your production time.

Eliminate the need to develop new UI systems for each title released.

Brag to your CEO about how much money you are saving.

An approved tools and technology provider for the Xbox 360

Anark Gameface

Next-Gen 3D UI Development

Contact Anark today to begin your evaluation:
www.anark.com/gameface / gameface@anark.com / 866.705.1010 x 102

MAYA 7 UNLIMITED FOR MAC

JAMES ALGUIRE

MAYA 7



STATS

Alias
210 King St. East
Toronto, Ontario, Canada
M5A 1J7
800-447-2542
www.alias.com

PRICE

\$6,999
Upgrade from v6.5:
\$1,249
Upgrade from v6.0:
\$1,899

SYSTEM REQUIREMENTS

Power Mac G4 and G5.
512MB RAM. CD-ROM
drive. Hardware-
accelerated OpenGL
graphics card. Three-
button mouse. 450MB
hard disk space.

PROS

1. Robust and extremely customizable.
2. Powerful 3D tools.
3. Faster render performance.

CONS

1. Interface can be cramped on smaller monitors.
2. Technical support beyond installation requires a subscription fee.
3. Some web site information not yet updated for Maya 7.

WHEN CONSIDERING MAYA, ALIAS'

flagship 3D modeling, animation, and effects environment, my first thoughts are of the amazing creatures, vehicles, and locations created by film and video companies like Industrial Light and Magic, Sony ImageWorks, and WETA Digital [to name a few]. But Maya also plays a significant role in video game development, as veterans like LucasArts, Capcom, and Electronic Arts can attest.

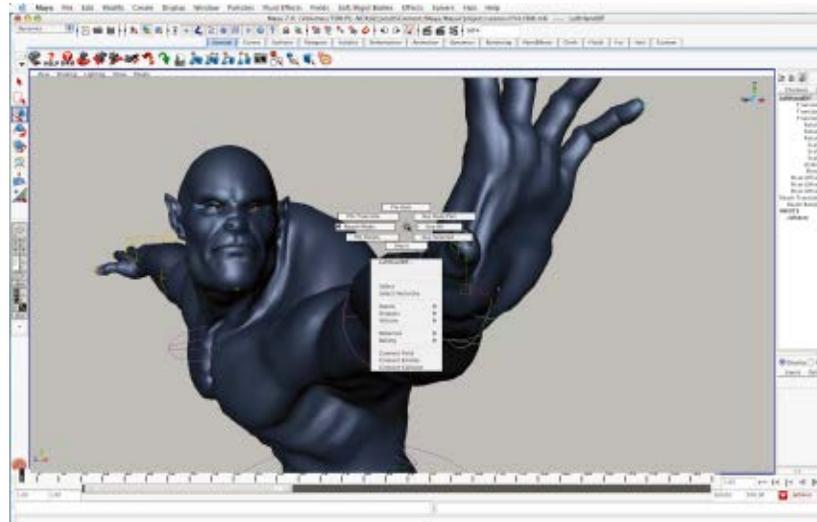
The latest version, Maya 7.0, released in August, adds a remarkable number of enhancements to an already solid suite of 3D tools. Many existing features were revamped and several new tools were added to improve the ease of use, efficiency, and overall workflow of Maya's integrated tools, as well as the workflow between Maya and third-party programs like Adobe Photoshop and Illustrator. Maya 7.0 is available for Windows, Linux, and Mac OS X platforms. This review focuses on the Mac OS X version only.

Note: Last month, Autodesk announced it would acquire Alias. However, Autodesk has announced that it "plans to continue to develop Alias products," and that "we do not anticipate changes with respect to planned product releases."

DEEP AND WIDE

Maya is a deceptively deep 3D environment, integrating several industrial strength tools for modeling, rendering, animation, and visual effects into a single suite. Alias harnesses Maya's layers of complexity within a single primary interface window. While far from being austere, the Maya interface is not as cluttered as others I've encountered. Tool shelves and panels are arranged in a fairly logical layout. If the user interface is too busy for your taste, various UI elements can be easily hidden with a click of the mouse. Maya even includes its own built-in web browser to view web-based content without leaving the program.

The typical hindrance of any truly deep program is clutter. While the interface starts out fairly clean, opening a number of editor windows or taking advantage of the tear-off menus can litter your screen



COURTESY OF ALIAS SYSTEMS CORP.

In Maya 7, Alias has added a new full body inverse kinematics system.

in no time. A large, wide-screen monitor is a definite plus to take full advantage of Maya's interface, and a three-button mouse is pretty much a necessity for working effectively. Also consider using a Wacom tablet, as many of Maya's Marking Menus and other functions can be accessed quicker using gestures, after some practice. Once you get the hang of it, gestures can significantly improve your efficiency.

Two handy new tools in Maya's UI worth mentioning are the View Compass, an onscreen tool that changes the view from perspective to front, back, side, bottom, or top with a mouse click, and the Universal Manipulator, which combines the features of the Move, Scale, and Rotate tools to adjust objects quickly using the mouse or by entering precise numeric values directly in a scene.

THE UPDATE LIST

Maya has too many new features to list in this review, but here are a few that game developers will definitely appreciate.

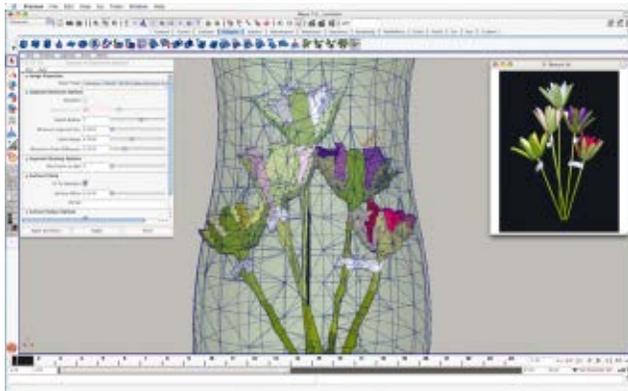
Character riggers and animators will find the newly-added full body inverse kinematics (FBIK) functions allow more natural posing and animating of 3D

characters than the previous IK system. For example, imagine lifting a character from a kneeling position to standing by pulling on its hand. The full body IK also makes it easier to work with quadrupeds. The technology for the system comes from Alias MotionBuilder, a real-time 3D character animation package that Alias acquired when it purchased Kaydara in 2004. In fact, because Maya and MotionBuilder share the same IK architecture, it's possible to transfer characters between the two programs with minimal loss of data, creating a clean and efficient workflow between modelers, riggers, and animators. Maya also sports a new Spring IK solver that helps maintain proportional rotations across joints. That simplifies the job of posing and animating multi-jointed limbs, like those of insects.

COMING TO THE SURFACE

Also of substantial benefit to game developers is the Surface Sampler, a tool for creating texture maps from the surface details of one object (the source) and baking them onto the surface geometry of a second object (the target). This tool can be handy to create better looking characters for game engines that require models with lower polygon counts or to help reduce the calculations needed for

COURTESY OF ALIAS SYSTEMS CORP.



Textures can be converted to polygons smoothly in Maya 7.

scenes created in Maya. For example, you can create film-quality characters for a project that will also have a game developed from its IP. The Surface Sampler can be used to create normal maps from the high-resolution models, which are then baked to lower-resolution models supported by your game engine, giving them a high-resolution look. Baking is the process of applying pre-

rendered materials, textures, or lighting to objects in Maya.

Among Maya 7.0's new UV mapping tools is a feature called Unfold UVs which unwraps the UV mesh of a polygonal object to help prevent UVs from overlapping and minimize texture map distortion. This feature works best with models that have complex organic shapes.

Alias has also added new features to the MEL scripting language and beefed up the Maya API to further assist developers in customizing

Maya to suit their needs and create their own tools and pipelines for transferring Maya data to and from a developer's chosen game engine.

SHAVE AND A HAIRCUT

Installing and using Maya requires a serial number and an activation key. The activation key, which is normally issued through Alias' online product activation

process and sent via email, is tied to the Mac address of your computer's Ethernet card to prevent piracy. If you replace the Ethernet card (or, on the Mac, if you replace the motherboard during a repair) or if you move Maya to a different computer, you'll need to get a new activation key or invest in a dongle. New to Maya 7 is support for USB dongles for Mac and Linux machines (previously Windows only). USB dongles make moving your Maya license between computers easier.

I ran into one snag trying to activate Maya. Since I installed it into a custom location and not the default location, the activation process would not accept my user password for my Mac's Admin account, and I couldn't activate the program. Once I moved it back into the default install location, it worked fine.

Alias provides good quality support for Maya. However, it is tiered into three levels: bronze, silver, and platinum. Bronze is Alias' free support level (you still must sign up for a bronze membership account) and it's a bit lean beyond the user forums and the online tutorials. To actually get technical support and access to the better tutorial materials, including downloadable tutorial DVDs you need to purchase either a silver (\$20/month or \$150/year) or platinum level membership (\$1,300/year). The silver membership is not a bad deal, although I would still like to see a few more tutorials offered at the bronze level. Also, take some time to read through the "What's New in Maya" section of the documentation. Some the improvements in Maya change its behavior significantly from previous versions, and that can affect projects and scenes created in older version of Maya.

Overall, Maya is a phenomenal tool for 3D that's approachable for users just getting started, but has plenty of muscle under the hood for the seasoned professional. ❖

JAMES ALGUIRE is a Mac professional and Apple Certified Trainer with more than 20 years experience in the computer industry. You can email him at jalguire@gdmag.com.

[Searching] for a **NURBS content creation tool** for next generation game titles?



nPower Software's **Power NURBS** is your solution.

TEST DRIVE **POWER NURBS** TODAY ON OUR NEW RE-DESIGNED WEBSITE.
WWW.NPOWERSOFTWARE.COM



WHAT GAMES HAVE TO TEACH US

AN INTERVIEW WITH JAMES PAUL GEE

» **JAMES PAUL GEE IS A WELL-KNOWN ACADEMIC AND STRONG** proponent of games for learning. With a PhD in linguistics from Stanford University, Gee currently operates out of the University of Wisconsin, heading the Games and Professional Practice Simulations program, which deals with digital interactive forms of learning.

Gee has come into recent prominence as one of the foremost thinkers on what games—even consumer off-the-shelf games—can and will teach people, especially children. Since releasing his book *What Video Games Have to Teach Us About Learning and Literacy*, Gee has gained a great deal of press for his views on the game industry from the academic side.

Game Developer spoke with Gee to find out just where he comes from, the ways in which games can teach, and what developers can do to facilitate the maturation of the industry as a whole.

Valve's HALF-LIFE



Brandon Sheffield: *Do you have experience in the game industry?*

James Paul Gee: No, my training is in linguistics, so for the second part of my career—for the last 20 years or so—I've worked in literacy and language and issues to do with schooling and education.

I got into games about four years ago as I was playing them with my six-year-old child. He was playing PAJAMA SAM, and I was helping him. I had no idea what an adult video game would be like, so I bought one kind of randomly. I was just blown away by how hard and complex the game was, and that people paid for it. As a form of fun, it's a very complex, thought-provoking pastime. Especially if you're new to it, it's very very difficult. As I got better and better, it dawned on me that good games, because they are long and hard and difficult, are very good at getting people to learn how to play them. The problem that the game industry has—how we get someone to learn something that's hard and complex—is the same problem that schools have. But the game industry is arguably better at solving it than schools.

BS: *Why do you think that is?*

JG: Games are essentially [about] problem solving in many cases. Whether you're playing HALF-LIFE or MORROWIND or anything else, you are continually solving problems, trying to psyche out a rule system, figure out what the rules will allow, what alternatives there are, and do it as elegantly and effectively as you can. And since games are made in levels that keep getting harder, they continually ask you to make your problem-solving [ability] better. In an odd way, games really are making fun out of tough learning.

BS: *Do you think that can be done with serious games?*

JG: I do. The neat thing about modern games is that they put you in a world where the problems are there to be solved and where the world suggests some of the possible solutions and gives you a strong identity to play, gives you the smart tools.

If you think about RISE OF NATIONS or any of those types of games, all the stuff you get to work with is really smart. You build smart



James Paul Gee



Humongous's PAJAMA SAM 3

Badda bing, Badda boom

For every action in games there is an equal and opposite reaction.

- Newton's Third Law of PhysX



[momentum]



The AGEIA PhysX™ platform provides the most powerful and advanced tool set for creating dynamic physical environments. With unmatched support for collision detection, fluids and advanced multi-threading for massively parallel architectures, your games can feature the next level in immersive game play.

AGEIA PhysX has you covered for today's software effects and tomorrow's game play with powerful hardware-based physics. Learn why more than 60 game development studios use AGEIA PhysX technology.

Visit us at www.ageia.com to learn more.





Pandemic's FULL SPECTRUM WARRIOR

soldiers, you can upgrade them and they do stuff for you. Same with FULL SPECTRUM WARRIOR. Games are worlds that are really good places to see what problems there are to solve, and I don't see why we couldn't have worlds that are more tied to serious content.

BS: How can you gauge the success of those teachings?

JG: Well that's a controversial issue because some people think that learning is successful if you can memorize a bunch of facts; but to me, learning is successful when you can solve problems in a given area. I'm not impressed if you can recite 12 or 14 facts about physics. But if you can solve problems in physics, then I'm pretty impressed. And even more, if you know what physicists do and why they do it, and how they approach those problems, and how physics varies from other sciences, then I'm really impressed. It's the problem solving that's at the core of these academic disciplines, not the facts.

BS: In terms of problem solving, do you think a game like POKEMON helps to shape kids as they're playing it?

JG: For little kids I'm particularly impressed by [games] like YUGIOH and POKEMON, not only because they are very rich problem-solving spaces, but given my interest in literacy, if you look at the language that's in a POKEMON game, or the language that's on the back of a YUGIOH card, or the language on a YUGIOH web site, it's very complex. It's not everyday language. It's just as complicated as the language kids see in school.

One of the major reasons that kids fail in school is the language of their textbooks gets very complicated, very dry, very boring, and they turn off. If you look at the language in YUGIOH, it makes the textbook look easy. It's very early training in complex specialist language, and it's very good for kids' vocabulary and their language development for school. In many cases, we know that kids of seven years old are seeing more complex language in POKEMON and YUGIOH than they're seeing in school.

BS: Do you think a game could ever take the place of a professor?

JG: No. People have suggested that, but what I'm interested in are games that are used as part of a whole learning package, much as the army does. The army doesn't just stick you in a trainer. The trainer is part of a whole package.

We're interested in games as part of a whole learning system.

After all, kids don't just play games. All the research we've done shows that kids are entirely social with games. They talk about them to each other, they trade fact sheets, they get on the internet, they look for cheats or look for strategy help from other people. These games are hard enough that most people go at them with help from others, often playing them together. We just want the same thing for education, that people put the game inside a whole social package of learning.

BS: With all the government controversy over violent games, do you think games can teach negative things as well as positive concepts?

JG: First of all, you know as well as I do that the government makes a violent video game. AMERICA'S ARMY is a violent game. It's a beautiful game and a wonderful game. The government clearly didn't object to that. It also funded part of FULL SPECTRUM WARRIOR. Any learning, whether it's books, a movie, or a game, can lead to bad or good results depending on the environment in which it's [played], not the game itself.

Are parents talking to kids about games? Are they relating it to other technologies? Are they getting the kids to think about the games strategically? If a kid plays AGE OF MYTHOLOGY, which is a great game for kids, do [parents] encourage their kid to get on the internet to learn and write about mythology, or is the game a babysitter? If the kid is in a home or culture of violence or neglect, then of course any technology, including books, is likely to lead to bad results.

BS: Do you think there should be ratings for serious games as well as consumer titles?

JG: Yeah, I think there should be ratings, but it would be nice if games, commercial and otherwise, came with some descriptive material that said what kind of people they're directed to, much the way [learning products] used to be age-graded, but I have no objections to ratings.

BS: How do you think public views of games could be changed for the better?

JG: I think the deep problem is that the powerful part of the public is the baby boomers. They're in control of things, they're the right age, and they just don't understand games at all.

First of all, the media does such a bad job of talking about games that everybody thinks GRAND THEFT AUTO is the only one in the world. The other thing they don't realize is that it's a difficult game and it takes a hell of a lot of thought to play.

There's an educational issue. The game associations and the industry itself need to do a better job of putting a good public face on games and making clear the variety of games, how complex they are and how appropriate they are for the older players.

Some of this will take care of itself as the younger generation gets older. We found in our research, when we were going to funding agencies, we were talking to baby boomers, and they take the word "game" to be trivial. If you're talking to people under 30, they do not take the word "game" to be trivial. Anybody who's played HALF-LIFE knows that a game is not trivial. This is really just a cultural difference.

The real failure has been the industry not putting enough money and time into getting clear that it is about a lot of things,



Double Fine's PSYCHONAUTS



Rockstar North's GRAND THEFT AUTO: SAN ANDRAS

more than one game. That could go a long way if they would take that responsibility.

BS: Do you think the fact that serious games have the word "game" in them is a semantic hindrance?

JG: I don't like that word at all. All learning is playful. That's a word that Ben Sawyer [co-director of the Serious Games Initiative] and the organizations have used.

What I think—and this is an issue that just doesn't get discussed—is that the game industry in my mind is immature in that it doesn't develop games for multiple niches. It doesn't do enough to develop new players who aren't in the traditional demographic categories. There are many people over 50 who would love to play games, many women, even older people in their 70s who can't travel but who would love to be in a world like *FARCRY*, but they wouldn't want to shoot something every 10 minutes, because they don't have the coordination.

If you look at the way the film industry has developed different types of films including a whole independent industry for different niches, there are just totally undeveloped niches in the game space. Nowhere near as many people buy games as go to movies, and that could change if the industry actively pursued new niches. The danger is that no one wants to give up the 17th sequel to *JAMES BOND* because they're sure it's going to sell.

BS: What do you think of Nintendo's new Revolution controller?

JG: I think it's fascinating, as is the Nintendo DS—the dual screen thing is revolutionary. There's a real tension in games today about whether innovation can flourish, because innovation, if you think about it, is always a risk, and furthermore if somebody is making an innovative game or an innovative controller, there will always be the possibility of making some mistakes. But you should take a chance.

Take a game like *KILLER 7* or *PSYCHONAUTS*. These are very innovative games, and like all innovation, there are high points and low points. People tend to harp on the low points in the reviews, and the industry doesn't get behind them and ends up making a standardized product, which I think in the end will hurt the industry badly.

I would applaud anything that is innovative, especially in the controller, because the more your experience of the game is like having your body in a world, the better it is. That's what people are thrilled with in games.

I think Nintendo—and I've always felt that the GameCube got a raw deal—I think they've produced superb games. It's certainly far and away produced the best games for younger players. I certainly hope that company keeps innovating.

If you're going to take games to a real mass market, which everybody seems to want to do, there two ways to do that. One is to dumb them down, make them easy, make them trivial, and that to me would kill all the interest that games have, because what's really interesting about them is that they're hard and complex, but yet part of popular culture. The other route is for designers to pay attention to how to keep the games complex, but not while frustrating players, especially those who are not the core aficionados.



Grasshopper Manufacture's *KILLER 7*

BS: What do you think about games for corporate training?

JG: They're going to be big. The reason they're going to be big is the same reason that churches and right wing groups want to make games: because games are exceedingly good at telling people how the world looks from your perspective. This does not mean that the person who plays the game will go out and accept your perspective, but it does mean that they will know what the world looks like.

That's why *AMERICA'S ARMY* is so successful. People know what the American army thinks like, what its values are, and what it looks like from the inside. It doesn't mean they want to be a soldier, but [the government] really branded the army through that game as a high-tech, collaborative space, and you can see why corporations lust after that, even to give to their customers.

Johnson & Johnson for example is interested in making a game for mothers that would help them prepare [to take care of] their babies; there could be some Johnson & Johnson products in it, but primarily they want those mothers to be in a world and feel that they're getting ready for their babies. But also you can get your managers and your employees to know what your company mission is and what your value system is. That's going to be very big.

BS: In terms of actual work that's being produced, what do you think is the most interesting in games for education right now?

JG: This is an interesting issue. There are loads of people trying to create serious games: startup companies, nonprofits, universities—all over. There's a company that we work with which is making games for algebra. Already been tested, doing very very well, and looks like it will hit the market.

There's Muzzy Lane, which has made a history game [*MAKING HISTORY: THE CALM AND THE STORM*] that's very good. A number of other companies that have made good games—like the stuff for terrorist response, homeland security, and emergency response games—some of that stuff has been quite good, but I don't think we have the killer app yet. I think what the so-called serious games industry is waiting for is the killer app, something that everybody points to and says that proves the concept. We haven't got that yet. *BreakAway*, as a commercial



Muzzy Lane's *MAKING HISTORY: THE CALM AND THE STORM*



Big Huge Games' *RISE OF NATIONS*

a new breed emerges



march 2006



Emergent Game Technologies and NDL have joined forces. Together, we're using the same revolutionary thinking behind Gamebryo — our groundbreaking 3D graphics engine — to unleash new technologies that will change the way games are built, tested and managed. Brace yourself.
www.emergentgametech.com





HOVERING ON A HANDHELD

THE PHYSICS BEHIND WIPEOUT PURE

» THE WIPEOUT FRANCHISE HAS BEEN AROUND FOR OVER A

decade. The first version helped launch the original PlayStation and created a distinctive niche for itself in gaming history. Since then, there have been numerous sequels for a number of different platforms, but never a handheld version. When the developers at Sony's Studio Liverpool found out about Sony's plan to release a groundbreaking new handheld onto the market, they decided to breathe new life into WIPEOUT, giving the PSP a killer launch title. WIPEOUT PURE was born.

The technology behind WIPEOUT PURE's physics engine, from general principles to detailed optimizations, was a key factor in the game's success. The coding took approximately 15 months, from first key press to U.S. approval. Ninety-five percent of the code was written from scratch, and for the first nine months, without PSP development kits. The code had to be particularly lean and mean because all the physics work was done by only one programmer (me); I was also responsible for the network ad hoc multiplayer and weapons code. Delivering an enjoyable gameplay experience on time was more important than writing academically perfect code.

This article discusses the general algorithms used in WIPEOUT PURE, delving into some of the optimizations and compromises I had to make.

COLLISION OVERVIEW

The collision system in WIPEOUT PURE is surprisingly simple, comprising a broad phase sweep-and-prune system for quick rejections on collision primitives, and a set of narrow phase functions to test for collisions between the primitives.

WIPEOUT PURE had three types of collision primitives:

- Static polygon meshes contained within axis aligned bounding boxes (AABB) with a couple of extra separating planes added, forming a discrete-orientation polytope (k-DOP) (see References, page 18, and Figure 1)
- Oriented bounding boxes (OBB) describing the ships (see Figure 2, page 16)
- Ray.

Because the track in WIPEOUT PURE is static, the collision data structures are computed at level load time and frozen in memory. The only dynamic collision data needed is for the craft. All other collision checks (for example, weapons and AI) were done using ray intersections tested against the static meshes and the OBBs.

The broad phase rejection algorithm used in WIPEOUT PURE was inspired by I-Collide (see References). I-Collide describes what is called a sweep-and-prune method for finding intersection candidates between 3D objects. Each craft has its world coordinate frame minimum and maximum values tested against the static collision mesh data. Collision candidates are tested using a custom OBB/mesh function. Craft/craft testing is done using a separate dynamic sweep-and-prune list where collision candidates are passed to an optimized



MARTIN LINKLATER is a senior programmer at Sony Computer Entertainment's Studio Liverpool in the U.K. He can be reached at minklater@gdmag.com.

OBB/OBB separating plane function, taken from OBBTree by Gottschalk, Lin, and Manocha (see References).

Raycast tests are sent through the same sweep-and-prune rejection tests, with intersection candidates passed on to ray/mesh and ray/OBB intersection functions. Long rays are split up into smaller segments and passed through the tests from source to destination. Ray tests return the position, normal, and collision IDs of the closest intersection.

The actual collision data for WIPEOUT PURE is a mixture of auto-generated track data and hand-built reset data. An in-house custom Maya plug-in was used to generate the track geometry. This tool also generates lower-resolution and optimized collision meshes. (Additional collision data was created by artists and level designers using Maya.) Track meshes are categorized as either floor, wall, or reset. Floor meshes interacted with the ship anti-gravity system. Wall meshes are simple colliders, and reset meshes are used to trigger the ship reset sequence, which is useful when the player turbo-jumps at 700mph. The full collision data for a track is between 5,000 and 10,000 triangles (see Figure 3, page 16).

PHYSICS OVERVIEW

The only proper physically simulated objects in Wipeout Pure are the ships. By "proper," I mean that they are objects that have their translations and rotations modeled reasonably accurately. Weapons and particles, on the other hand, are modeled as simple point masses. Although I would class the system as a rigid body simulator, it has been heavily simplified and optimized for use on a handheld.

WIPEOUT PURE uses a simple Euler integrator, which runs at approximately 100Hz for the ships, and once per graphical frame for the particles and weapons. (I won't go into the math behind the integrator here because it's already been explained by people far more clever than I am. For a great introduction to rigid body integrators, see Bariff and Witkin in the References.)

The basic logic for the integrator in WIPEOUT PURE can be described in four steps.

- Process and resolve collisions: separate interpenetrating objects and apply impulses to ships
- Process craft handling and player input: generate the forces and torques which will be applied to the ships
- Integrate over delta t: move and rotate
- Repeat.

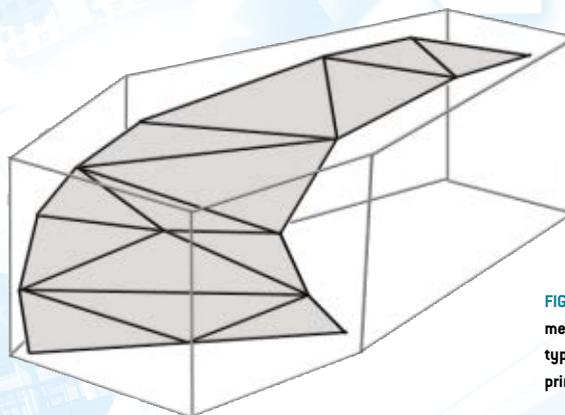


FIGURE 1 Static polygon meshes are one of the types of collision primitives.

HANDLING OVERVIEW

The handling system is the method that allows you to turn a bunch of boring rigid bodies into super-fun anti-gravity WIPEOUT ships. The basic process is to take the raw player pad input, filter it to take care of analog and digital differences, then send that filtered player input through the handling code. After the handling code has finished, the physics integrator is told to apply a force and a torque to the ship. It's as simple as that.

Internally, all player input is analog. Digital input is simply filtered from binary to analog using a simple smoothing function. Each aspect of the handling is dealt with separately in its own function (such as throttle, friction, or aerodynamics). The results from each handling function are summed together in an accumulator to give the final force and torque values for that ship. The anti-gravity function is by far the most complex of the lot, deserving a little more explanation.

A ship in WIPEOUT is basically a cuboid sitting atop four damped springs. The springs are modeled as downward pointing collision rays. The results from these collision tests are used to generate forces and torques at each corner of the ship.

AI control input is dealt with in the same fashion as player input. The AI code generates fake control input, which is then passed to the handling functions. There are a couple of little shortcuts in there to help the AI along, but most of the time the handling functions use exactly the same handling logic as the player.

Within the handling system there are very few hard-coded numbers. Most of the numbers used in the handling functions are exposed to the designers. This enables them to alter and tweak the performance of the different ships without the need for a code recompile.

WIPEOUT PURE uses simple XML-based data files to hold the ship handling data. Using an industry standard file format lets the designers freely use whichever editor package they want to alter the values. Once the XML file has been updated, they simply press a key combination on the PSP and the



HOVERING ON A HANDHELD



new handling stats are uploaded to the running executable.

One of my rules of thumb is that it's always quicker to use someone else's tool for a job rather than write my own, especially now that there's

such an abundance of open source and free software available on the internet. XML is both extensible and simple to understand. And if you feel the need for a custom editor, there's nothing stopping you from bolting a nice GUI on top of the core XML data.

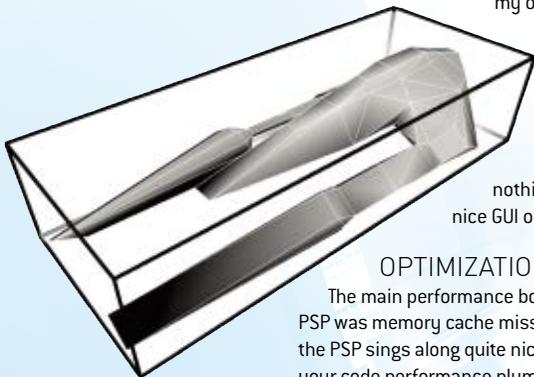


FIGURE 2 Oriented bounding boxes for ship collisions are another of the types of collision primitive.

OPTIMIZATIONS AND SHORTCUTS

The main performance bottleneck that I encountered on the PSP was memory cache misses. If you keep data in the cache, the PSP sings along quite nicely. But if you break the cache, your code performance plummets. With this in mind, I spent a lot of time making sure that the data in the collision system was stored in an optimal way. Keeping frequently-accessed data together and doing away with large malloc block headers proved very beneficial. The high-level sweep-and-prune data was dropped down from floating point to fixed point 16-bit. This reduced the efficiency of the sweep-and-prune function since floating point ranges needed to be fattened into a fixed point (by "fattened," I mean that minimums were floored and maximums were at their ceiling). As a result, more collision primitives were

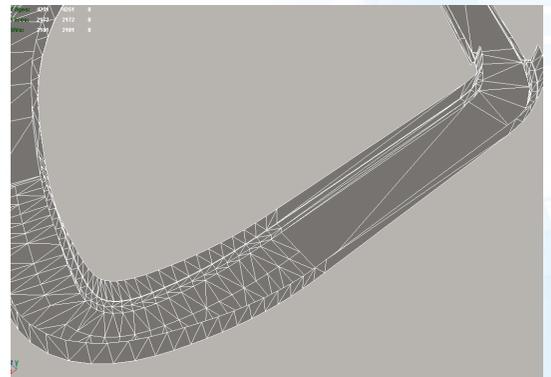


FIGURE 3 The full collision data for a Wipeout Pure track is shown.

passed down to the primitive collision functions. The decrease in rejection efficiency and increase in type conversion were more than offset by the performance improvement attained by minimizing the memory footprint and layout.

The place where I made the largest relative gains was the integrator. The PSP has a very handy SIMD co-processor called the VFPU. Hand-coding the integration code to use the VFPU saw around an 800 percent improvement in performance, which was mainly due to processing all of the integration math in one batch and holding temporary variables and accumulators in registers until processing was completed. A little hand tuning of instruction order meant that dependency stalls were virtually eliminated. Unfortunately, the effect made little difference on the overall performance because the integrator was quite fast to begin with.

Once I optimized the memory footprint and moved a lot of the math over to the co-processor, I realized that I was still way over my CPU budget. I was running at about 50 percent of CPU. As much as I had resisted doing it to this point, I needed to start hacking out chunks of the logic. The tricky part was doing this without ruining the feel of the physics.

The most expensive part of the handling code was the four anti-gravity feeler calculations. I first moved from four rays, one at each corner, to two rays, one in front and one in back. Finally, I moved to a system in which the ships used two rays when moving slowly, but switched to one ray at the front when speed picked up. A ghost rear ray was calculated using the position and normal information from the front ray test. Since Wipeout ships move so fast relative to their length, you can't tell when this switch happens—there's virtually zero effect on the handling.

In a traditional update cycle, you process forces and collisions every time you run the integrator. The control code and collision system are both running at the same speed as the integrator, in my case, about 100Hz, which was too much for the CPU to handle. Collision checking was taking up valuable cycles, and every time I ran the handling code, I had to process the dreaded anti-gravity. I had to make another serious compromise in order to resolve the CPU's load.

I knew the collision code could handle a much slower update, and I had written the handling code to be framerate independent from the outset, so I made the decision to uncouple the

get Physical

IF YOU'RE LOOKING TO EXPAND YOUR knowledge of physics programming for games beyond what's contained in Wipeout Pure, there are several advanced areas worth studying.

Constraints. Look at general joint constraints and maybe modeling collisions as constraints, as opposed to impulse based reactions.

Better collision resolver. My current knowledge of multiple-body

collisions is somewhat lacking; for example, Wipeout Pure doesn't deal well with stacked boxes.

Time. Modeling very quick objects can be tricky. In Wipeout Pure, I simply modeled them as a point mass traveling along a ray. In the future I'd like to use cast volumes to improve accuracy and open up some more possibilities.

Better testing and debugging

features. It's notoriously hard to track down bugs in collision and physics code. I've had situations where bugs happen only at one particular polygon when the player ship is at one particular orientation. It's very hard to reproduce and pin down exactly which piece of math is breaking in a situation like this. Better debugging tools and unit testing should hopefully help this in the future.

IN GAME DEVELOPMENT THERE ARE

NO SECOND CHANCES.

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

GAME OVER.



DELIVER QUALITY. DELIVER ON TIME. USE SEAPINE.

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

TestTrack Pro

Advanced Issue Management

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



Surround SCM

Flexible Change Management

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



TRUSTED BY LEADING GAME DEVELOPERS

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

Visit www.seapine.com/gamedev_gd to learn more about TestTrack Pro and Surround SCM and download evaluation copies.

 **Seapine Software™**

RESOURCES

I-Collide:
www.cs.unc.edu/~geom/I_COLLIDE/index.html

Gottschalk, Stefan; M.C. Lin, and Dinesh Mahocha. "OBBTree: A Hierarchical Structure for Rapid Interference Detection." (SIGGRAPH Conference Proceedings, 1996).
www.cs.unc.edu/~geom/OBB/OBBT.html

van den Bergen, Gino and Gino Johannes Bergen. *Collision Detection in Interactive 3D Environments* (San Francisco, Elsevier, 2003).

Witkin, Andrew and David Baraff. "Physically Based Modelling: Principles and Practice." Online SIGGRAPH 1997 Course Notes.
www.cs.cmu.edu/~baraff/sigcourse



collision and handling from the integrator. As long as the framerate stayed above 15 or 20fps, we would be in the clear. Thankfully, the framerate in WIPEOUT PURE mostly stays around 30fps. In the shipping version, the collisions and handling are processed once every graphical update (frame), whereas the integrator still runs at about 100Hz.

At this point, I was using roughly 10 percent of the CPU to perform the physical simulation for all eight ships at 30fps, which included the collision tests, physics integration, and ship handling code—a triumphant moment in the physics engine's development.

However, the order in which my optimizations were done was opposite to what's normally considered best practice. In best practice, you would first optimize the algorithm and then the memory, and finally you would drop to assembler. My excuse for not adhering to this sequence is that I was very reticent to change the handling system until I absolutely needed to. We had been running with good physics for more than six months and the last thing I wanted to do was compromise the handling for CPU. In the end, I managed to find compromises that met the CPU's needs and also retained the feel of the handling.

Of course, there are further optimizations that could have been included had I been given development time, but I had to make the decision to move on to other areas of code that I was responsible for. We had a solid deadline that absolutely could not move, and optimizing the physics was entering the arena of diminishing returns.

COMPROMISES IN THE NAME OF PHYSICS

Writing fun physics code for a game necessitates making compromises. Console physics programmers never have as much CPU or memory as they would like and must always compromise between what they would like to do and what they are able to do on the platform—especially when it's a handheld.

A great uber-physics simulator is no good if it restricts the game design, kills the framerate, or makes the player feel

frustrated. Remember, the game must be fun first, and so the physics should be fun too (at least for the person programming them). During the development of WIPEOUT PURE, I had to tone down or remove physically correct behaviors when the game designers told me they felt wrong. I also had to justify movements and actions that weren't quite as good as they could be because of some fundamental design decisions I made early on (like using boxes for ship collisions).

The Internet is awash with articles that try to persuade gamers that physics is the next big thing, and to some extent I agree. The improved performance and storage capacities of new hardware open up a lot of interesting design possibilities that were not available in the past. But physics programmers must not lose sight of the ultimate goal of game programming—to make a fun game that sells. All else is secondary.

In the course of writing game physics engines for more than a decade, I've learned a few rules of thumb which will hopefully be of some use to other developers. First, find a balance between robustness and performance that fits the game design. It's no good spending time writing super physics if the game doesn't need super physics to be fun.

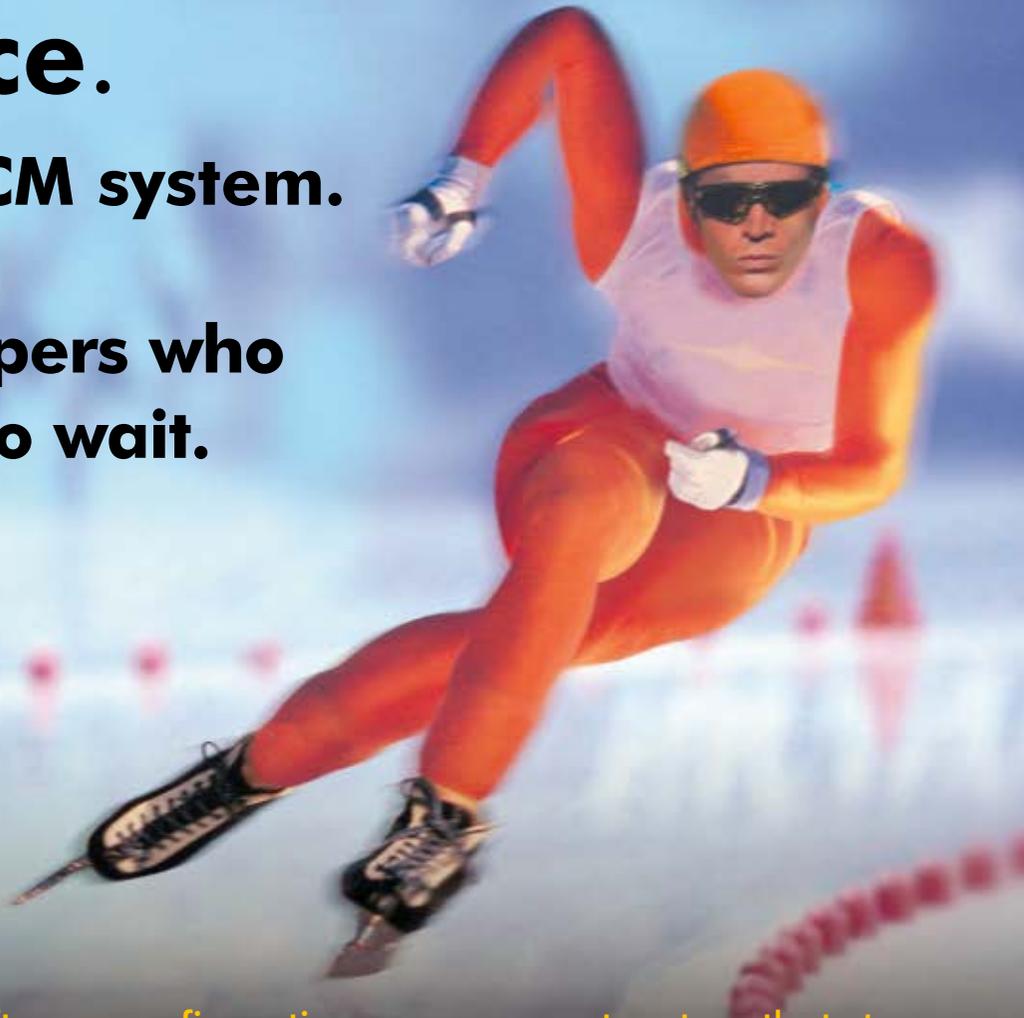
Second, make realistic design decisions early on. Making hard decisions early on helps the development process. It gives the designers firm limits to work within and helps stop feature-creep. Just remember not to aim too low. I usually try to be as ambitious as needed to make myself feel just a little out of my comfort zone.

Finally, get the physics prototyped as soon as possible. It's amazing how much team buy-in you can get when the core control system for a game is up and running early on. Having something that feels fun from the outset gives the team a great focus. Not to mention giving you more time later on to find those tricky math bugs. ❌

Perforce.

The *fast* SCM system.

For developers who don't like to wait.



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

[Fast]

[Scalable]

[Distributed]

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

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

PERFORCE
SOFTWARE

Download a free copy of Perforce, no questions asked, from www.perforce.com. Free technical support is available throughout your evaluation.

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

SCALING THE CABAL

VALVE'S DESIGN PROCESS FOR CREATING HALF-LIFE 2

✦ **WHILE BUILDING HALF-LIFE, WHICH SHIPPED IN** November 1998, Valve created a method of decentralized design called the Cabal Process (described in the December 1999 issue of *Game Developer* and available online at www.gamasutra.com), which used a small cabal of a few people from various disciplines to tackle the design. Needless to say, when design began on HALF-LIFE 2, we had great interest in applying the same structure and principles to its development, too. However, the greater scope of the sequel posed some problems for the Cabal Process, so we had to tweak it until it worked for us again. This article discusses the revised Cabal Process used to make HALF-LIFE 2.

PROJECT SCALING

HALF-LIFE 2 was a project with ambitious goals. We nearly tripled the team size, and embarked on a

huge technology push on all fronts. Acting, physics, AI, sound, rendering, and networking systems were all built from scratch. During the technology push, an expanded version of the original HALF-LIFE cabal met for months, attempting to create a complete design document similar to the first one. Design work during the early phase of development progressed very slowly because we found it difficult to predict what kinds of designs our technology would enable once it was finished. To make matters worse, the resulting design relied on many game elements that were purely theoretical.

By the time the Source technology had matured, we found ourselves in a position similar, in some ways, to where we were at the start of the Cabal Process for HALF-LIFE, but very different in others. In terms of design, we were better off. We had a full story timeline, detailed story snippets, all the major character profiles, a set of locations and drawings, and a fairly clear idea of what technology we would have for the final game. In terms of production, though, we only had a bunch of raw material in the bank: some weapons, some cool monsters (and some not-so-cool monsters), and pieces of interesting levels. However, as with HALF-LIFE, at this stage of development, the technology was not being taken advantage of. You couldn't play the game all

BRIAN JACOBSON, a software developer at Valve, is currently hard at work on *TEAM FORTRESS 2*.

DAVID SPEYER is a software engineer, and before coming to Valve in 1999, spent six years as a programmer and project manager developing retail communications software. Send comments about this article to editors@gdmag.com.



GAME DATA



PUBLISHER:

Sierra Studios (Vivendi Universal Games)/Electronic Arts

NUMBER OF FULL-TIME DEVELOPERS:

40

NUMBER OF CONTRACTORS:

15 (not including testers)

LENGTH OF DEVELOPMENT:

Initial Development Phase: 48 months;
Final Development Phase: 23 months

RELEASE DATE:

November 16, 2004

PLATFORMS:

PC Windows, Xbox

DEVELOPMENT HARDWARE:

AMD and Intel CPUs: 1.2GHz–3.4GHz CPUs; DirectX 7–9; ATI and Nvidia GPUs, 256MB–2GB RAM

DEVELOPMENT SOFTWARE USED:

Softimage XSI, Visual Studio 6, PhotoShop, Perforce, ZBrush, Cubase, Sound Forge, Havok

SCALING THE CABAL

the way through, and none of the levels were tied together in a coherent fashion.

Once we knew what our engine could do and had enough raw material in the bank to use as constraints to drive the design, the Cabal Process began to work as efficiently for us as it had during the development of HALF-LIFE.

The problem now was, given the much larger scale of the game and larger number of people working on the project, the Cabal Process itself became a bottleneck. It couldn't produce content fast enough. As a result, we created three nearly independent design cabals, each responsible for designing and producing roughly one-third of the game, plus dedicated cabals for art, sound, and acting.

BACK IN THE SADDLE AGAIN

Each cabal consisted of four or five people, half level designers and half programmers. While developing HALF-LIFE, we decided that this was the ideal size. Larger cabals resulted in diluted design meetings and smaller ones risked a dearth of ideas. We included both programmers and level designers because most design iteration occurred through changes to AI, game code, or levels. Each cabal also included one engine programmer who would develop new technology required by the designs. For productivity reasons, we wanted each team member to have a "demanding customer" on the same cabal, someone who consumed that person's work. Level designers were customers of programmers in that they used the gameplay elements and AI created by the programmers. Programmers were customers of level designers in that they needed levels as a venue to refine



The helicopter battle at the end of the Canals sequence was a keyframe moment used to constrain the design of the Canals.



Although the concept was born years earlier, the APC was not introduced into the game until the last four months of the project.

their code. The members of each cabal shared an office to reduce communications overhead and, as we discovered, improve prioritization. People were far less likely to get sidetracked by non-critical tasks if their teammates sat nearby to serve as instant triage.

The HALF-LIFE cabal included artists and a writer, whereas HALF-LIFE 2's multi-cabal structure prompted us to treat artists and writers as shared resources. We created an art team, an acting team, and a sound team (actually just a single sound designer). The art team collaborated with the design cabals on the look of the environments, monsters, and characters in the early stages of development and made the levels look great once the gameplay in those levels was stable. The sound team worked with the design cabals to produce stand-in sounds during gameplay prototyping and to create a final sound treatment of the levels after the design stabilized. The acting team

collaborated with the design cabals to seed levels with mission goals and story rewards, and they produced any animations the levels required. The acting team also served as an independent fourth design cabal for the story-heavy sections of the game, such as Kleiner's Lab, Black Mesa East, and Breen's chamber.

Despite the large structural changes to the Cabal Process, there still were many aspects of the original process (as described in our previous article) that remained intact. The way each cabal generated designs remained largely unchanged. We preserved our edict, "He who designs it, builds it," in the belief that the best designs are influenced by the realities of production. People who are very cognizant of all the tradeoffs inherent to a given implementation are going to make better design choices. We continued to discourage a sense of sole ownership because we believe that having more hands on a given section of the game ultimately produces higher quality. Our playtesting techniques remained the same, and we continued to use them as a way to settle design arguments. As with HALF-LIFE, the cabals were completely responsible for meeting the quality standards in the levels they owned.

The result was that we had six teams, all of whose work—models, materials, sound, animation, lighting, story, and game design—had to come together in the levels themselves.



Valve used over three hours of recorded dialogue to bring the HALF-LIFE 2 characters to life, compiled from multiple recording sessions with each voice actor.

[Other competitive solutions require multiple supporting packages to handle our assets. With **Alienbrain**, everything we need is in one package.]

Christopher Bretz Art Director, Secret Level



Brainpower

Avid Alienbrain Studio 7.1 has the intelligence to help your team work faster and smarter.

Its intuitive interface tells you more than the what, where and who of an asset. It tells you what an asset references, what it's referenced by, and whether it's buildable, it's optimized, or it's redundant. It knows what platform an asset is for and what should happen to it next. It even knows an asset's location in a texture map, its resolution in level three, and how much damage it inflicts when eaten. Alienbrain knows. You know. Your pipeline knows.

ASSET MANAGEMENT FOR CREATIVE TEAMS

Free eval copies available at www.alienbrain.com



www.alienbrain.com

Avid
computer graphics

SCALING THE CABAL

Clearly, managing this process was going to be tricky but essential for us to succeed.

There were some obvious problems, of course. How would we manage and reduce the cost of the many interdependencies between our six teams? How would we allow every team to apply important constraints to the design? How would we create a consistent design and level of quality in the face of three independent design teams? These problems were eventually solved on a case-by-case basis.

KEYFRAMING PROSE

HALF-LIFE 2 contains more than three hours of acting, and recording the dialogue for these scenes wasn't always easy. In some cases, it required flying to Los Angeles, exploiting a limited window in an actor's busy schedule, and using a fixed number of studio sessions, after which we would be on our own.

In an ideal world, we would have gone through a more traditional screenwriting process, but that would only have been possible if we knew in advance where our game design process was going to take us. We couldn't leave all the acting until the end because then there wouldn't be enough time to improve it; so story and gameplay had to develop concurrently.

At first, the two seemed inextricably linked, which presented an interesting challenge: How would we give the gameplay cabals, whose process (and result) was fluid and unpredictable, the freedom to experiment while presenting a stable enough framework on which we could hang a story? We eventually settled into a process whereby story provided keyframes that served to constrain the game design. For example, in designing the Route Kanal and Water Hazard chapters, we knew the player would start on the run from City 17 forces outside Kleiner's Lab and finish at Black Mesa East, far from City 17. The story elements that fell between those two story keyframes were purposely left vague until later in the process when the gameplay had solidified. As long as the gameplay cabal satisfied the constraints of the story keyframes, the cabal was free to take the gameplay in whatever direction seemed most promising without fear of leaving the story in an untenable position.

Once a chapter's gameplay was finalized, the responsible gameplay cabal and the acting cabal met to draw up a list of



An early concept for the streets of City 17.

Orange maps of the Prison area, used for gameplay prototyping.



The final Prison map, after input from the orange map tests.

places within the chapter where story elements could be added. Some were required by the gameplay, such as the delivery of short-term mission goals or the explanation of a game mechanic. Others were important for the story or for player motivation, such as the reinforcement of a larger overarching goal (like reminding the player that they had to get to Eli's during Route Kanal). Finally, some were story-based rewards that served to enrich the player experience. Even with this process, the story still had to be supple enough to respond to unexpected gameplay demands, such as when Ravenholm moved from before Black Mesa East to after, once the potential of the gravity gun to enhance Ravenholm was realized.

INSIDER ART

The art burden of HALF-LIFE 2 was an order of magnitude greater than that of HALF-LIFE. HALF-LIFE 2 used more than 3,500 models, nearly 10,000 materials, and individual maps as big as 20MB (compared to HALF-LIFE's 300 models, 4,000 materials, and 3MB map files)—a tremendous investment in visual quality. In order to produce this many art assets with a relatively small team of artists, we had to optimize the art production pipeline and insulate it from gameplay changes as much as possible.

The art production for a chapter began with the creation of concept sketches, which were developed early in the cabal's design process once the general setting was established. In many cases, the concepts were developed even earlier based on the broad story design, in which case they served to inspire the game design. Once the concepts and gameplay were deemed compatible, the concepts were developed into styleguides—maps devoid of gameplay that would serve as a template for building final production maps. The styleguides both influenced and were influenced by gameplay prototypes that were developed simultaneously. For example, the buggy's handling characteristics influenced the scale of the coastal landscapes in which it was used and vice-versa.

AGENT ORANGE

Initial gameplay prototyping for each chapter took place on orange maps. Orange maps use an orange grid texture for walls and a gray one for floors and ceilings, and using them solved a number of issues we ran into early on.

principles of half-life

IN DEVELOPING HALF-LIFE 2, A FEW principles were accepted upfront as constraints because of what the team learned while making HALF-LIFE.

- Don't impose a personality on the player (never let Gordon talk).
- Don't implement cues that separate

the player from Gordon Freeman (never leave first-person perspective; maintain a continuous timeline as much as possible).

- Provide a strong visual grammar for gameplay elements without breaking realism.

- All training (specifically for HALF-LIFE 2) should be accomplished within the context of the game.
- Provide distinct gameplay mechanics and themes in each chapter.

THE ONLY TRUE GAME-PLAY
PHYSICS TECHNOLOGY.

STOP WATCHING.
START PLAYING!



- ✦ Unsurpassed cross-platform game physics with an integrated animation system and tool chain
- ✦ Includes HydraCore™ technology for multi-core and multi-threaded platforms
- ✦ Backed by the most experienced support in the middleware industry
- ✦ Enables next-generation character animation that blends physics and animation in real-time

havok
www.havok.com

SCALING THE CABAL



Alyx's original companion was a ferret-like alien. Somehow Dog seemed like a better friend.



This illustration of Gordon Freeman was created by a member of the HALF-LIFE community.

First, it prevented level designers from investing in the art for an area before the core game mechanics had been validated through playtesting. Effecting this practice dramatically reduced the iteration cost and avoided any early commitment to the look of an area. It also solved the problem of prematurely critiquing art when team members were supposed to be critiquing a gameplay prototype. Finally, it allowed the art team more freedom to experiment with visual themes, since they could do so independent of the gameplay prototypes.

Successful gameplay prototypes and styleguides were used as the basis for building the final levels. Once those playtested successfully, they were handed off to the art team for an art pass. During the art pass, all level designer-created stand-in geometry was replaced by final models. Final materials were applied to the level, lighting was adjusted or recreated from scratch, and auxiliary elements, such as fire, fog, and skyboxes, were added.

Through this process, the playable level was made to more closely match the vision of the original concept art without breaking the gameplay. In practice, though, gameplay did sometimes break in unexpected ways, such as when playtesters refused to walk on a large suspension bridge once a realistically thin-framed model replaced the chunkier, level designer-created predecessor. Because of this inherent dependency between visual design and the communication of game mechanics, the design cabals always held playtests after the art pass to verify that gameplay still worked.

SYMBOLIC LINKS

To allow multiple teams to work simultaneously on a single level without causing stalls, we tried as much as possible to structure our tools around independent files that were connected by a system of symbolic links. Symbolic links are human readable references, resolved at runtime, that both code and content use to refer to another code or content resource.

For example, we replaced direct references to raw sound files in our maps with names of entries in a sound script file instead.

Each entry in the script file specified such variables as pitch, volume, and random file selection for the sound. This allowed our sound designer to replace or modify sounds without affecting level designers. Before we had symbolic links, level designers had to hand off maps to the sound designer and not work on them until the sounds were finished. Also, by using level-specific sound names for level-specific sounds, the sound designer could change sounds without disturbing other maps.

Our acting sequences used symbolic links to indicate where actors would walk or look in a level. Facial animation, animation blending, and sequencing of a scene's events could then be authored while another person worked on the world geometry. This technique was also used to script citizen dialogue, allowing our writer to quickly iterate it.

Though these are just a few examples, we pushed symbolic links into as many areas of the pipeline as possible. The general strategy was to increase the number of iterations by specialists by reducing iteration cost, since we believe that more iteration results in a higher quality product. Lower iteration cost also reduced the cost of experimentation, which is really just another kind of iteration. This technique also allowed us to make changes far closer to shipping than previously possible because the interdependencies were removed.

GLOBAL CONSISTENCY

All our chapter designs began with the same core set of design principles, many of which were derived from HALF-LIFE, but some were new. The team wanted to extend the direction of HALF-LIFE without losing sight of what we felt were the things that made it successful. The overarching goal was to create an immersive first-person experience, so we accepted some



mods get serious

WHEN YOU THINK OF HALF-LIFE 2 MODS, you probably think of COUNTER-STRIKE: SOURCE, DAY OF DEFEAT: SOURCE, GARRY'S MOD, DYSTOPIA, and others. Guns, explosions, enemies.

But have you ever heard of PULSE!! or GNNVIZ? These are games that are classified as serious mods—serious games modified from source code of mass-market games, in this case, HALF-LIFE 2 mods.

PULSE!! is a prototype virtual learning environment for medical personnel from Texas A&M University—Corpus Christi, and GNNVIZ is a forest environment visualization tool from Oregon State University.

As more and more researchers, teachers and others begin to explore the serious games space, they run into one major issue: limited money to fund development projects. Serious mods solve this problem nicely, offering an excellent entry path to

development with little up-front cost.

HALF-LIFE 2 is very affordable, as are the tools, if not outright free. And the web abounds with resources to help the development process. Inexpensive and free development tools combined with an extremely powerful and moddable engine enable researchers to explore their serious game ideas cheaply, but with high quality results.

Both games that I worked on, PULSE!! and GNNVIZ, were funded in part due to the fact that I was able to show the funders an inexpensive and innovative approach to solving a particular problem.

PULSE!! puts the player in the role of nurse or other medical support personnel in a Combat Support Hospital or civilian ER. Instead of guns or crowbars, there are instead stethoscopes, IVs, and medical supplies. There's no enemy to beat, but a patient to be healed. In first-person view,

players use their medical knowledge to win the game by paying attention to the patient and doctor, acting on what they see, hear, and know. If the player fails to pay attention or act, the patient dies. Game over.

PULSE!! makes extensive reuse of the civilian character models and AI from HALF-LIFE 2 to simulate patients, doctors and nurses. The VGUI is used to create live EKG displays and other in-game information sources.

The second game, GNNVIZ, is another serious mod, funded by the U.S. Joint Fire Science Program at Oregon State University. GNNVIZ creates immersive visualization environments of forests based on large scale GIS and other forest metadata. Players can not only visually explore a simulated real forest, but access accompanying metadata on forest composition, fire hazards, roads, streams, and land ownership.

GNNVIZ stretches the engine's abilities a bit (figuratively and literally), scaling players down to one-quarter of their normal size to simulate larger areas. Custom display code based on the detail object system enables the engine to draw massive numbers of trees over a simulated area several kilometers across.

Of course, if your serious game project advances to the next level, beyond the free mod stage, as mine did, then it's time to talk with Valve directly about licensing.

If you have a serious game project in the works, consider the mod approach as your prototype development model. Low cost, high return, excellent results. Seriously!

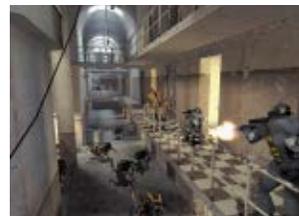
For more information on both these games, see <http://oregonstate.edu/~holt/seriousmods>.

—Tim Holt, Oregon State University



so players would get a steady flow of new weapons as they progressed through the game. The weapons cabal also worked with each design team to make sure the weapons had an interesting introduction, with enough incentive shortly thereafter for players to use learn how to use the weapon.

Many of our project management decisions were also made with global consistency in mind. The gameplay cabals had weekly reviews with cross-cabal resources (management, art, animation) to help propagate design decisions. These reviews had the goal of helping each cabal operate with similar scope, schedule, deliverables, and methods. We used comparative metrics where available (how many maps per level-designer-week are you trying to ship?) to analyze each cabal's output. Code was constantly published—in order for one cabal to use it, it had to be made available to all—and shared as another means of propagating design choices. We did our best to synchronize the deliverables across groups, which increased the effectiveness of team-wide playtests and other cross-cabal feedback mechanisms. It forced the teams to solve similar problems at the same time, and it fostered positive competition. No cabal wanted to be behind or have lower-quality levels when it came time for the playtest.



principles as constraints up front (see Principles of HALF-LIFE, page 24).

Despite the fact that each design cabal followed the same high-level principles, design inconsistencies were a natural consequence of the multi-cabal structure. The designs of the individual cabals reflected the strengths and weaknesses of the various members—therefore each group developed different game mechanics and made different decisions about, for example, the level of difficulty, density of experience, and the relative proportions of combat to puzzles. Our toolset was so large that cabal members tended to prefer designs that used tools they were most familiar with. One team had a rendering specialist, while another had an AI specialist. Some level designers were great at developing combat, while others excelled at optimizing performance. Some were great at authoring terrain, others were best at working with entities, and still others had better artistic sensibilities than the rest. So how did we produce a cohesive game despite all these disparities?

First, we tried to achieve an economical design. Each cabal was encouraged to ask the question, “How well does this element leverage our other gameplay elements?” as a framework for evaluating design choices. This led naturally to a more cohesive experience, since the same elements tended to be used throughout the game.

We used team-wide playtests to expose game mechanics created by one cabal to the other cabals so that they could identify and share the successful game mechanics, spreading them throughout the game. For example, the Ravenholm cabal enabled the gravity gun to interact in specialized ways with particular objects (such as the sawblades). This inspired the Citadel cabal to make the super gravity gun. The energy balls resulting from that work were later used by the Follow Freeman cabal to open the Nexus gates. Later still, they were incorporated into the alternate-fire for the Combine assault rifle.

These team-wide playtests also helped highlight the inconsistencies in other areas, such as quality of visuals, combat, and puzzles and so forth. When one cabal saw that another was producing better work, the two groups were quick to come together and discuss the techniques they were using.

Because certain design elements, such as weapons and monsters, crossed cabal boundaries, it was sometimes hard to change those elements without breaking another cabal's levels. We solved this problem for weapons by forming a weapons cabal, which comprised representatives from the three gameplay cabals and included both hardcore FPS and less expert players so that the needs of both player types were considered. The weapons cabal's goal was to produce a varied and balanced palette of weapons, wherein each had a unique function but no obvious best weapon emerged (at least not until we wanted it to). The weapons cabal tuned weapon placement within the game timeline to eliminate clumping and droughts,

A SECOND GO AROUND

Even before production began, we planned to do a quality pass over the entire game once we hit alpha to evaluate all our choices within the global context of the game. It quickly became apparent that we would also need to use this second pass to solve consistency problems that had not been solved during the first pass over all the levels. This second pass, which ended up taking only about four months, resulted in a huge improvement in the quality of the game.

At the start of alpha, the game's quality was fairly variable, and it had wildly varied pacing. Transitions between chapters were often nonsensical, as it was hard for one design cabal to predict what another was doing at the beginning of the adjoining section. There also were fairly large inconsistencies in the level of difficulty from chapter to chapter. Some of these problems were fairly straightforward to fix. Chapter transitions, for example, were trivial to smooth out once each cabal could see what was on both sides of the transition. Of all the inconsistencies, the most difficult one to solve was ensuring consistently high quality across the entire game.

To evaluate the game as a whole, at the beginning of alpha, the entire team took a break from building the game to play through the entire experience, sending feedback for general discussion. As a means of distilling the disparate feedback into a consistent actionable message, a new group called the Cabal Cabal was formed, a team that included one member of all six teams, as well as a few others, and which met daily throughout the weeklong, teamwide playtest to critique, chapter by chapter, the entire game.

The Cabal Cabal's goal was to provide feedback to the other teams so each could maximize overall quality. The final decision of how to respond to the feedback was left up to each



Concept art of the poison zombie.

SCALING THE CABAL

responsible design cabal, with each cabal allocating its resources where it felt the best results could be achieved.

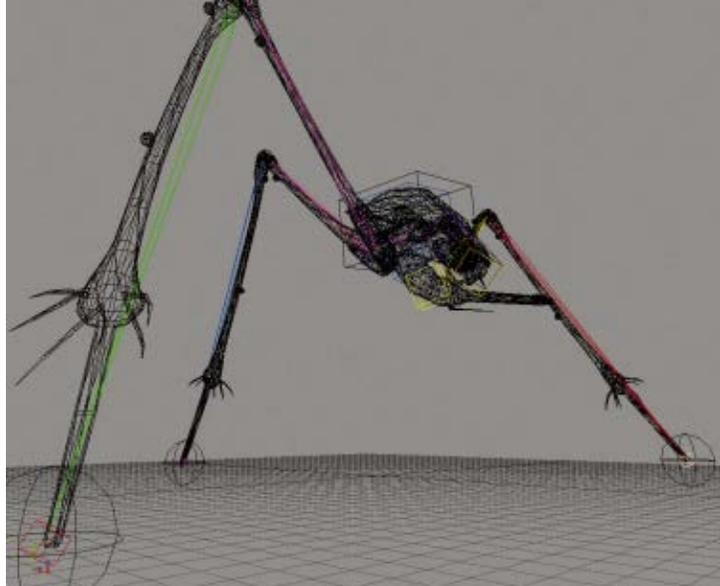
The Cabal Cabal focused its discussions on the high and low points of each chapter. The high points were identified for polish and amplification, as these presented the easiest opportunities to maximize quality. Opportunities for cross-pollination of highly popular game mechanics or experiences were noted, which helped us not only leverage our best elements, but also improve our design economy and consistency.

Low points were typically sections of the game that were frustrating, confusing, empty, or simply very rough. Sections of the game that were relentless to the point of being fatiguing were broken up with puzzles or downtime while sections that felt empty were filled with additional content. Some low points were too costly to fix, which led to a final round of cuts. These amputations were really painful because anything cut this late in the project had been invested in heavily. This taught us that the only thing more painful than an early cut is a late one, so it's best to be decisive in the beginning. But we reminded ourselves that we cared far more about that content than our customers would, since they would only see the final product. It was also comforting to remember that cutting content meant the rest of the game would receive more attention and thus achieve a higher quality.

MULTIPLE ITERATIONS, MAXIMUM GAINS

Many of us were surprised at the large improvement in quality between the game at alpha and the game after we finished our second pass, given the relatively short amount of time it took. We now consider multiple iterations to be a key to HALF-LIFE 2's success and a mandate for future projects, the major benefit being that it allowed us to make far better decisions.

During the development of both HALF-LIFE and HALF-LIFE 2, we found that decisions made later in the project were always better than decisions made earlier. Some were better simply because they were better informed by the experience we had in making the game up to that point. For example, work on the Citadel began only six weeks before alpha, and unlike the rest of our chapters, we didn't already have a plan for what major gameplay element was going to be used. The prototype of all gameplay elements in the Citadel levels took a single day, and our first pass on that chapter was finished in three weeks. The reason the super-gravity gun was created was that we knew at



The Combine stun stick was a solution for the player not having a gun at the start of the game—a device was needed to prompt the player to keep moving.

that point in development that the gravity gun was a highly successful element in our game. Development was extremely efficient because we knew the engine well enough to choose game mechanics we could implement very quickly.

Other decisions couldn't possibly be made until later in the project because they required more of the product to exist around them before they could be made. For example, the qualifier "good enough" (and its dreaded opposite, "not good enough") proved especially elusive during the early production phases of Ravenholm and Nova Prospekt (the first two chapters produced), but became clear and well understood once the game was assembled as a whole. Balancing the level of difficulty as well as maintaining an appropriate pace were two other problems that couldn't even be addressed until we saw the game as a whole.

Obviously, making certain decisions too late in development can wreak havoc with a shipping schedule. We used time as the primary constraint on how issues could be resolved to avoid this problem. The closer we were to shipping, the less acceptable it became to make changes with broad dependencies. For example, in the prototype phase, new technology or AI could be added, spaces could be defined, and levels could be reordered. After the art pass, changes to the world geometry and the general lighting scheme were constrained. After alpha, the game mechanics, art assets, level progression, characters, and most dialogue were fixed and could only be altered for cases in which the repercussions were isolated and well understood. Our investments in symbolic links really paid off during this phase because it allowed us to make a large number of fairly significant changes with low cost. ❖

fruits of labor

CREATING HALF-LIFE 2 WAS A TREMENDOUS

learning experience for everyone on the team. Behind commercial success, perhaps one of the more creditable signs that our process succeeded is that everyone on the team is genuinely proud of the product we created, and excited to repeat the process. Hopefully some of the many lessons we learned creating HALF-LIFE 2 are generally useful and could be applied to other projects.

Here are some of those lessons that we feel are most important:

- Decentralize your design.
- Make rough, but global decisions early (weapons, story, basic monster behaviors). With investment comes constraints; minimize investment until you hit critical mass of quality, then iterate until good becomes great.
- Don't design using theoretical mechanics. Validate designs first using prototypes. It doesn't have to look good at all (use "orange" maps) and perhaps can be prototyped in your previous generation technology.
- If you have a one-year schedule, try to reach alpha in eight months to give yourself a few months to iterate your design anew. In our experience, every week of work after alpha is worth well over four weeks of work prior to alpha.
- Create demanding customers for everyone on your team—it's a great technique for improving efficiency and prioritization.
- In the traditional tradeoff of scope, quality, and time, reduce scope to get better results through iteration.
- Attempt to reduce pipeline stalls by carefully thinking about where those stalls occur in your production pipeline.
- Use symbolic links to eliminate pipeline stalls and allow as many low-cost late changes to your work as possible.
- Processes are cheap and disposable—try to measure how they are succeeding or failing to achieve game and company goals. Don't be afraid to change a process if it stops working.

GameDevelopers
Conference

MARCH 20-24
SAN JOSE, CALIFORNIA

WHAT'S NEXT
GDC:06

www.gdconf.com

GAME DEVELOPERS CHOICE AWARDS

INDEPENDENT GAMES FESTIVAL

GDC MOBILE

SERIOUS GAMES SUMMIT

GAME CONNECTION

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

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



THE INNER PRODUCT

TEXAS HOLD'EM AI

I RECENTLY PROGRAMMED THE AI FOR WORLD SERIES OF POKER, developed by Left Field Productions and published by Activision. I started out thinking it would be an easy task, but it proved a lot more complex than I initially thought.

This article should give the budding poker AI programmer a foundation for a simple implementation of *No-Limit Texas Hold'em* AI, covering the basics of hand strength evaluation and betting. By following the formula set out here, you'll build a solid foundation of knowledge on which to implement a reasonably strong poker AI. Just a note: This article assumes you're familiar with the basic terminology of poker and the rules of *Texas Hold'em*.

The goal of any game-playing AI is twofold. The primary purpose is to give the player a fun and enjoyable experience. The second purpose, subordinate to the first, is to play a strong enough game to provide sufficient challenge to the majority of players in your intended audience.

POKER DATA TYPES

To create a *Texas Hold'em* AI, you'll need to implement the following data types. I'm going to describe them at the bit/byte implementation level, leaving the high-level abstraction up to you.

A *suit* is an integer in the range 0 to 3, where 0=clubs, 1=diamonds, 2=hearts, 3=spades.

A *rank* is an integer in the range 0 to 12, where 0=2 (deuce), 1=3, 11=king, 12=ace.

A *card* is an integer in the range 0 to 51, hence

$$\begin{aligned} \text{card} &= \text{suit} * 13 + \text{rank} \\ \text{suit} &= \text{card} / 13 \\ \text{rank} &= \text{card} \% 13. \end{aligned}$$

♣	-	-	-	A	K	Q	J	T	9	8	7	6	5	4	3	2
♦	-	-	-	A	K	Q	J	T	9	8	7	6	5	4	3	2
♥	-	-	-	A	K	Q	J	T	9	8	7	6	5	4	3	2
♠	-	-	-	A	K	Q	J	T	9	8	7	6	5	4	3	2

FIGURE 1 64-bit data structure representing a hand of poker, using four 16-bit words.

-	-	Type	Rank1	Rank2	Rank3	Rank4	Rank5
---	---	------	-------	-------	-------	-------	-------

FIGURE 2 32-bit data structure representing a hand value with six nibbles.

A *hand* is a 52-bit data type; each bit represents a single card. This can be stored as four 16-bit words for ease of use, for which each 16-bit word represents the potential cards in one suit (using 13 of the 16 bits). See Figure 1.

A *hand type* is an integer representing the type of poker hand you have (see Handtypes).

ENCODING HAND VALUES

A hand value is a 32-bit integer representing the relative strength of a hand of cards. By comparing two hand values, you can see which hand is stronger in a game of poker.

The hand value can conveniently be represented as a series of six 4-bit nibbles, where the most significant nibble represents the hand type; the next five nibbles represent the different ranks of the cards in the order of significance to the hand value (see Figure 2).

EXAMPLE 1. AH QD 4S KH 8C is a no pair hand type (sometimes called a high card, or in this case ace high). So the hand type nibble is set to 0. The remaining nibbles in the hand value are filled out with the ranks of the five cards in descending order [A, K, Q, 8, 4], which translates into rank indices: 12, 11, 10, 6, 2 (or C, B, A, 6, 2 in hexadecimal), and when combined with the hand type [0] in the high nibble, gives us a 32-bit integer: 0x000CBA62.

The individual suits of the cards are basically ignored in the final hand value. The only time suit is significant is when

it contributes to a flush. Also, note the top two nibbles of the hand value are always zero.

EXAMPLE 2. 4D JD 3D 4C AD is a pair of fours, with ace, jack, three kickers. The hand type is a pair (type 1) then the ranks follow, starting with the rank of the pair, then the ranks of the kickers: 4, A, J, 3, which gives us 0x0012C910.

EXAMPLE 3. 7C, 6C, 5C, 4C, 3D is a straight (type 4). More specifically, it's a seven high straight. The only rank of import here is the seven (rank 5). The hand value is encoded as 0x00450000. We save ourselves a bunch of instructions in ignoring the four low cards after determining that the hand is a straight.

Look at the resultant hand values of these three examples. You can clearly see how the better hands always have a higher hand value. We determine the winning hand with a simple comparison.

CALCULATING HAND VALUES

What we now need is a function that takes a hand and returns a hand value. This involves determining the hand type, then inserting the nibbles for the hand ranks, as done in Examples 1-3.

A hand is four words (clubs, diamonds, hearts, spades) of 13 bits each, which can be arranged in 8,192 combinations. We can accelerate the evaluation of a hand by pre-calculating 8K tables of things like the number of bits set in a (13-bit) word (if you have five or more of the same suit, then you've got a flush), or the highest card of any straight in the hand. You can also pre-calculate a table of the highest five cards

HANDTYPES

- 0=no pair
- 1=pair
- 2=two pair
- 3=trips
- 4=straight
- 5=flush
- 6=full house
- 7=quads
- 8=straight flush

MICK WEST was a co-founder of Neversoft Entertainment. He's been in the game industry for 17 years and currently works as a technical consultant. Email him at mwest@gdmag.com.

from a particular bit combination, which you can then use to set the kicker cards.

If you calculate **ranks** (hearts | diamonds | clubs | spades), then the value **ranks** is a bit-field with a bit set for every card rank that you have at least one of. The number of bits set here is the number of unique ranks you have. We calculate the number of bits in each of hearts, diamonds, clubs, and spades, and subtract the total number of bits in the unique ranks, giving the number of duplicated ranks to be used as the basis of determining what type of hand you have.

EXAMPLE 4. If you have 2D AS AH 2C 2H, you can quickly determine that you have five cards, that there are just two unique ranks, and hence you must have either a full house or four of a kind. A few more simple tests will determine exactly what hand you have. The entire evaluation function will consist of tests like this, gradually whittling down the possible hand types.

Since the function consists mostly of bit-wise operations, table lookups, and simple comparisons, finding the hand type takes no time at all. It's also very amenable to fine-tuning optimization, and the exact implementation will depend on

the target architecture. You may be able to take advantage of some processor specific instructions to squeeze out the last few cycles or performance.

CALCULATING HAND STRENGTH

Hand strength is the probability that you will win the hand, given your hole cards, the community cards, and the opponents who remain in the hand. Hand strength is a floating point number between 0.0 (certain loss) and 1.0 (certain win). For example, a hand strength of 0.33 means you have a 33 percent chance of winning.

The easiest and most flexible way of calculating the hand strength is to simulate the progress of the game many many times and count the number of those times you win. Say you simulate the game 1,000 times, and in the simulation, you win 423 games; you have a hand strength of 423/1,000 or 0.423. See Listing 1.

To be more accurate, we have to run our simulation with other players dropping out if they are dealt hole cards below a certain threshold. In practice, whether a player

LISTING 1

- 1 Create a pack of cards
- 2 Set score to 0
- 3 Remove the known cards (your hole cards and any community cards)
- 4 Repeat 1,000 times (or more, depending on CPU resources and desired accuracy)
 - 5 Shuffle the remaining pack
 - 6 Deal your opponent's hole cards, and the remaining community cards
 - 7 Evaluate all hands, and see who has the best hands
 - 8 If you have the best hand then
 - 9 Add 1/(number of people with the same hand value) to your score
 - 10 End if
- 11 end repeat
- 12 Hand Strength = score/number of loops (1000 in this case).

Procedure for simulating a game.

TABLE 1

If rate of return is ...	fold	call	raise
<0.8 then ...	95%	0%	5% (bluff)
<1.0 then ...	80%	5%	15% (bluff)
<1.3 then ...	0%	60%	40%
≥ 1.3 then ...	0%	30%	70%

If fold and amount to call is zero, then call.

If the rate of return for the computer's hand is of a certain value, then the computer's action, whether it will fold, call, or raise, has a varying percent change of occurring.

stays in is a probabilistic function of the strength of their hole cards, table position, stack size, previous behavior, and the blind size. For now, we can just modify the simulation so that after dealing the opponents' hole cards, we remove any non-blind players with hole cards worse than, say, a pair of sixes. While not particularly elegant, it will still give you a useful number.

POT ODDS

The pot odds number is the ratio of your bet or call to the size of the pot after you bet (the amount you would win). For example, if the bet is \$20, and there is \$40 in the pot, then the pot odds are $20/(20+40)=0.333$.

RATE OF RETURN

Rate of return is the on-average proportion of how much you will multiply your bet by if you stay in the hand.

$$\text{rate of return} = \text{hand strength} / \text{pot odds}$$

The base strategy we implement is to mostly stay in hands with a rate of return greater than 1.

FOLD, CALL, OR RAISE

For each round of betting, the computer needs to decide if it should fold, call, or raise (the FCR decision). Ignoring the question of how much to raise for the moment, and given a rate of return (RR), it's possible to

provide a very simple mapping between RR and FCR (see Table 1).

Don't pay too much attention to the precise percentages listed in Table 1. The numbers will depend on the way you calculate your hand strength, and you'll want to vary them depending on which betting round you're in. You'll also want to vary these numbers to create players with different personalities.

Using this very simple mapping between the RR and the FCR decision can give you a surprisingly reasonable and entertaining player. They will tend to play strong hands, they will occasionally bluff, they won't scare easy if their hand is good, they will abandon weak hands when raised, and they will stick around on a reasonable chance of a flush or straight draw.

The fact that none of the percentages is 100 is also important. You can never deduce the hand strength of your AI opponent based on their actions (unless they fold, in which case the information does you no good). If the opponent raises, then it could have any kind of hand strength, probably strong, though it might be the rare time (1 in 20) when it is bluffing with a very weak hand.

RELATED TOPICS

- Pre-flop hand strength tables
- Opponent modeling
- Implied odds
- Personality modeling
- Positional play
- Probabilistic search space
- Game theory and Nash Equilibrium

STACK PROTECTION

These simple rules work well when your stack of chips is large and the blinds are small. However, as your stack shrinks and the blinds increase, the amount of money you need to commit to stay in a hand can become a very substantial proportion of your stack. Also, other players occasionally might go all-in, so we need some logic to prevent the AI from making bad calls when short stacked.

Say you have AD 2D and the flop is QC KC 2C. You have a pair of twos, but a possible flush is out there. There's \$500 in the pot and the bet is \$100 to stay in against two players—but it's your last \$100. The pot odds are $100/600=0.1666$, your hand strength is 0.297, so your rate of return is about 1.8.

If you could play this situation over and over again, you would make on average an 80 percent profit each time. However, it's your last \$100, and you have about a

70 percent chance of losing everything. Don't make that bet!

To handle this situation, we can use a simple heuristic, along the lines of:

If my proposed bet will substantially commit my stack, then don't do it unless I have a strong chance of winning;

which might be implemented in part by:

*if (stack-bet) < (blind * 4) and (HS < 0.5), then fold,*

meaning if the call would leave you with less than four times the big blind, then don't call unless you have a greater than 50 percent chance of winning.

Poker is a complex game, with a surprisingly large number of different types of situations like this that you have to handle somehow. I recommend you have as few special cases as possible, as it reduces the risk of an exploit being introduced into the game via some obscure special case. However, you should anticipate a number of heuristics (rules of thumb) being hard coded into the AI logic.

TESTING POKER AI

Playing a quick single table game of *Texas Hold'em* takes around 30 minutes on average with human players. Ideally, you would perform your test by having humans play against the AI and trying to find problems with it. Unfortunately, due to the random hands being dealt, it's very easy for one player to simply get lucky and win the game with sub-par logic, or

even flawed logic. I've found it takes at least 10 games to begin to get a clear picture of the qualities of an AI player, and more like a hundred games to be really sure. This often creates an unreasonable burden on the testing department and introduces a very long delay in getting feedback on AI changes.

The solution: automated testing. The AI should be set up so that different variants of AI can play against each other in set of high-speed games. You should also code a few simplistic poker AIs into the mix, such as an AI that always goes all in, or another that simply always raises with a hand better than a pair of fives. Then you set your AI loose against these opponents and make sure it wins the appropriate percentage of games. If you coded your evaluation and simulation appropriately, then you should be able to simulate an entire game in about a second. (You might want to reduce the iterations of the simulation a bit to speed up testing).

The best use of your human testers is to try to get them to find an exploit of the AI, then you can codify this exploit into a temporary AI opponent to include in your test suite. You can then tweak your AI until it defaults the exploit, while still being able to defeat all the other opponents.

MORE WORK

What I've set out here is just a foundation for poker AI. By following the process laid out here you will get a reasonably strong and entertaining opponent. ❄

RESOURCES

Sklansky, David. *The Theory of Poker*. Henderson, Nev: Two Plus Two Publishing, 1999.

Provides various discussions of pot odds, implied odds, and so forth, with many heuristics that might be useful.

The University of Alberta Computer Poker Research Group
www.cs.ualberta.ca/~games/poker

A number of research papers on implementing poker AI.

Peretz, Evin. Hold'em Killer.
www.holdemkiller.blogspot.com
A blog on implementing poker AI.

Poker-Eval
<http://freshmeat.net/projects/poker-eval>
A GPL Licensed poker hand evaluation library.



INDEPENDENT
GAMES FESTIVAL

REWARDING INNOVATION IN INDEPENDENT GAMES

THE 8TH ANNUAL
**INDEPENDENT
GAMES FESTIVAL**

MARCH 22-24, 2006
SAN JOSE, CA
WWW.IGF.COM

The stage is being set for the eighth annual Independent Games Festival—the most respected event celebrating the indie game developer.

All Game Developers Conference attendees are invited to visit the IGF Pavilion, and to attend the IGF Awards on Wednesday, March 22



PLATINUM SPONSOR:

[adult swim]

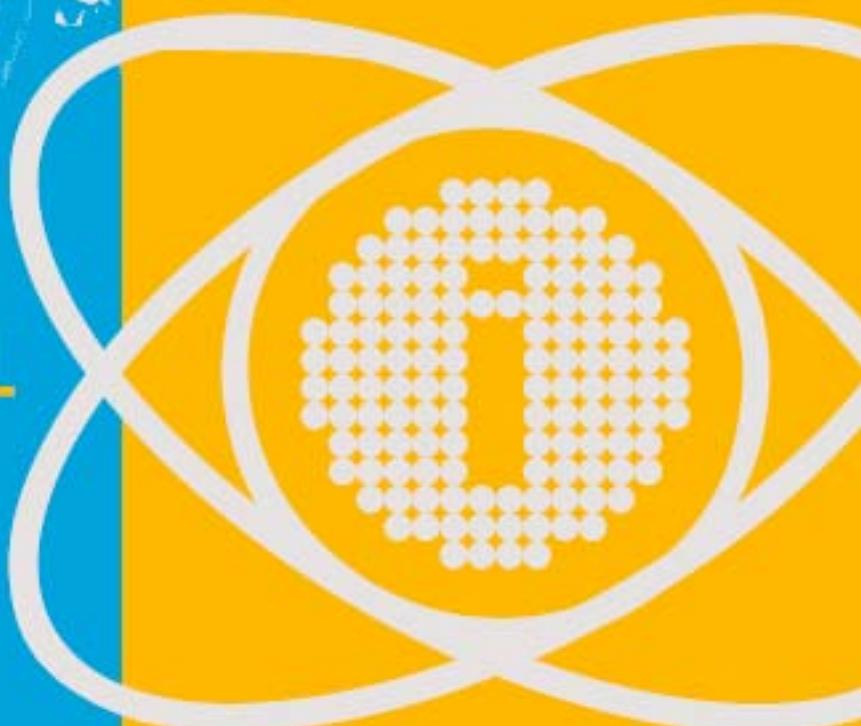
GOLD SPONSOR:

Microsoft



CMP

United Business Media



OPEN CALL FOR ENTRIES

IGF COMPETITION SUBMISSIONS CLOSED

Open to all independent game development teams, the IGF Competition offers over \$35,000 in cash and prizes.

IGF MODDING COMPETITION SUBMISSIONS CLOSED

New Competition! Non-commercial mod teams compete in one of four categories for a chance to take home a \$2500 cash prize.

IGF STUDENT COMPETITION DEADLINE FOR SUBMISSIONS: NOVEMBER 15, 2005

A competition for games developed by full-time students.

FOR INFORMATION ON CATEGORY DESCRIPTIONS,
ENTRY FORMS, AND ELIGIBILITY RULES GO TO

WWW.IGF.COM



PIXEL PUSHER

ANATOMICALLY CORRECT PART I

A leg to stand on

RECENTLY, AS I STARTED BUILDING THE skeleton for a new character, I was struck by the depressing thought that, despite setting up hundreds of rigs, I still had no

reliable rule for placing the joints. It made me uncomfortably aware of the fact that though many traditional artists have a great and detailed knowledge of skeletal and muscular anatomy which underlies the surfaces they draw, few animators (myself included) have anywhere near the same level of information, even though our jobs are far more intimately related to the complex workings of muscle and bone. I decided to research the bare bones, as it were, of anatomy and share my findings with my fellow animators.

SHOW SOME LEG

In the world of the animation tutorial, a leg consists of a hip, knee, ankle, and toe, all lined up neatly from a side window view. In the real world, though, legs are fantastically complex machines. The feet alone account for about a quarter of the 206 total bones in the body. Even the knees are remarkably sophisticated, multi-axis, shock-absorbing mechanisms.

A precise simulation of how the feet and knees work with precise bio-mechanical accuracy is a stretch even for a full-blown Hollywood muscle rig, so it's clearly too much to ask of a typical four-bone game character leg. Even so, a good grasp of underlying anatomy is a real leg up when it comes to building characters that fit in with our innate understanding of how human beings look and move. To build better deformation skeletons, we can start by better understanding the basic anatomy of legs and feet.

But there's one important rule that has to be stated up front: No amount of research work or fancy vocabulary should trump the evidence of your eyes. Any figure-drawing class will teach you to draw what you see, not what you know. Similarly, the only real test of a successful character setup is how well it appeals to the eye. So use this information to inform, not replace, your intuitions.

PECULIAR PELVIS

The pelvis itself is actually a system of eight bones bound together by flexible ligaments. However, for animation purposes, it's easiest to treat it as a single unit, so there's no point in listing the various names of intricate parts.

The familiar Mickey Mouse ears of the pelvic structure, known as the iliac crests, are important landmarks for animators because they are useful indicators for the locations of hip joints and the base of the spine. The iliac crests are usually visible on a male figure, underlining the oblique muscles of the lower torso, forming the lower boundary of love handles.

On a female figure, the iliac crests can usually be seen as subtle bumps in the upper slope of the hips, just a little lower than the navel. If you can't puzzle them out (on an unusually muscled or abnormally hefty character), try to image what the character would look like when resting his or her hand on a hip. That's about the right height for the iliac crests and thus the base of the spine. Obviously that spine joint should be centered from the front view.

Placement from the side is very much a tradeoff between biological and technical realities. Biological purists say the spine should bend between the high point of the iliac crests, about four-fifths of the way back through the body, which is technically and biologically the correct location. Most animators, though, will pull that point forward, closer to the centerline of the body to minimize scrunching in the deformations.

HIPPY HIPPIY SHAKE

The hips are ball and socket joints locked away inside the ear-shaped arches of the pelvis. Correctly locating the pivot of the hip can be very tricky, since it's buried deep

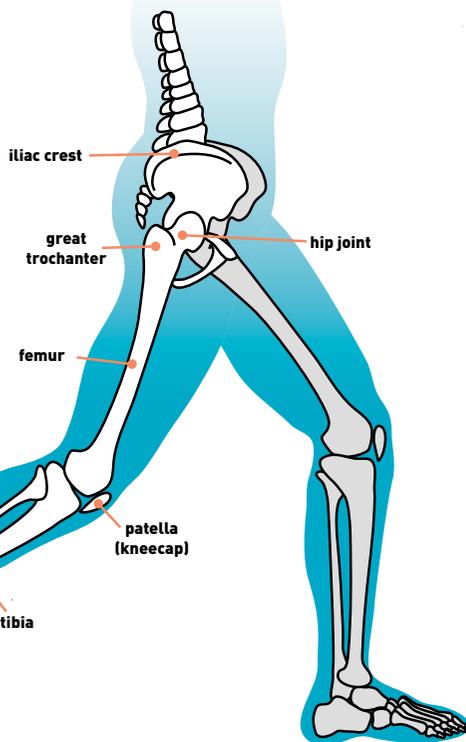


FIGURE 1 Knowing the major landmarks of the lower body skeleton can greatly inform your ability to rig a character.

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

inside the flesh without obvious landmarks.

From the side, you may be able to see a superficial bump of the great trochanter, the angled knob of the femur (thigh bone), which is just a little behind and below the actual hip joint (see Figure 1). If the trochanter isn't expressed in your model, you may be able to spot the place where the bulk of the gluteus maximus (buttocks) and the gluteus medius (the bulge of muscle just below the crest of the hip) meet.

If all else fails, start just below the halfway mark from the navel to the crotch and just move forward of the centerline of the thigh; then experiment to find the best visual balance. You'll probably want to cheat the position forward a little to help minimize the creasing that comes from raising the legs.

From the front, the hip joint should be about halfway between the curve of the iliac crest and the centerline of the body. On most models, this is noticeably inboard from the visual center of the thigh. If this seems odd, remember that the femur has a distinct bend in its upper end, like a lowercase "r," so the line from hip to knee shouldn't be straight up and down.

Being highly mobile three-axis joints, the hips benefit a great deal from procedural fix-up bones (described in "Twist and Shout: Fixing Twisted Deformations," April 2004). A good typical setup might include a single fix-up located behind the hip joint, which moves downward as the hip lifts; rotation diminishing fix-up located just outboard of the hip itself; and a twist fix-up in the area of the quadriceps to minimize twist collapses as the leg raises.

KNEE BENDS

In the real world, the knee is a much more than a simple hinge joint, but luckily for us, most of that subtlety doesn't show up on the outside. However, the pivot of the leg doesn't fall directly behind the mass of the kneecap; it's

about two-thirds of the way down the kneecap's mass.

Simple skeletons without deformation helper bones often push the pivot point forward to help preserve the kneecap's shape when the model bends. If you can spare a couple of transforms, though, a dedicated helper bone run with a driven key will produce better results and more realistic movement in the lower leg.

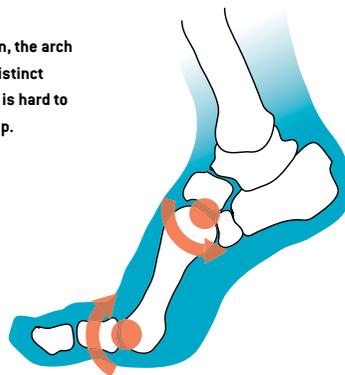
TWISTED ANKLES

The actual mechanics of the ankle are very complicated. Technically, the ankle itself is only a one degree of freedom joint, which only rotates up (supination) or down (pronation). Some of the twist component of the ankles happens in the twisting of the tibia and fibula in the calf, just like the better known twist of radius and ulna in the forearm. The remainder of the twisting motion is provided by a second joint hidden inside the heel itself, known as the sub-talar joint. But on a game character, there's no visual cost to combining the twist and elevation rotations into a single joint, particularly if you add a twist fix-up (like the one described in the April 2004 article) to the calf to help preserve the volume of the ankle as the foot twists.

The rotation point of the ankle is easy to find. In height, it's about midway between the ankle bones (the malleoli). Remember that the malleoli aren't level. The inner malleolus is distinctly higher, so get a good front or rear view as well as a side one when positioning the ankle joint. Don't be too slavish about splitting the line of the ankle bones. You can often achieve better visual results at runtime by dropping the pivot a quarter of an inch or so.

From the front view, the midpoint between the malleoli may seem a bit inward of where you might expect, but it's correct. The ankle joint ought to line up with the second toe, rather than the third. From the side, many animators

FIGURE 2 When the foot is pressed down, the arch of the foot folds downward, creating a distinct "stepped" appearance in the foot which is hard to capture with a typical one-link foot setup.



prefer to push the joint a bit ahead of the centerline between the malleoli as a way of preventing ankle crunch when the foot bends up; you may also be able to achieve the same effect with a fix-up bone that moves a bit forward as the foot lifts.

TIME WOUNDS ALL HEELS

Although the feet are full of bones (27 apiece) there are plenty of situations when bare feet can be rigged well with only the traditional single "toe" joint. The bend of the one joint toe is really an abstract combination of flex on the toes and some deformation in shape of the foot itself.

The joint in a one joint foot should be in the center of the ball of the foot, right about at the root of the big toe (see Figure 2). A single joint foot will have to flex a lot—more than 45 degrees—so be very careful with the vertical position of this joint to avoid either inflating or deflating your character's toes. Remember that this ball joint is a two axis joint. The foot should be able to twist along its length by a few degrees as well. In reality the parallel bones of the phalanges can fold like a fan to improve ground contact and traction, but this effect is hard to get really right without cumbersome setups. A little twist can go a long way toward grounding your character properly.

For highly polished character work, such as in fighting games or detailed cinematics, it's a good idea to add an extra joint to represent the arch of the foot. The arch works somewhat like a truck's leaf spring, diffusing the shocks that are transmitted up from the ball of the foot to the ankle and amplifying the

PIXEL PUSHER

downward thrust of the leg in a jump. This gives the foot a slightly “stepped” appearance when the foot is pronated downward. More importantly, the flexibility of the arch slightly counteracts the leverage of the foot, so it will show up in a walk, run, or jump cycle and will affect the knees’ movement.

A SHOE-IN

All of these rules apply to barefoot characters. Getting really high quality results for shoes, unfortunately, is an art, not a science.

The behavior of the foot itself is pretty complex to begin with, but the relationship between the foot and the shoe is unpredictable. The rule of “whatever looks best” is the only hard and fast one, but there are some general principles you can observe.

If the character is wearing ordinary shoes or sandals, the flexion of the foot will appear farther back along the appendage. Try to get most of this effect with vertex weighting, rather than by moving the ball pivot back to the middle of the arch. Too long a lever on the toes will make for funny walk cycles. Moving the pivot back by an inch or less should be fine, however, because shod feet are a bit longer than bare ones. For characters in heavy boots or thickly soled shoes, it might be a good idea to use two or even three joints to spread out the flex of the foot more evenly in an arc.

TWINKLE TOES

If you’re a masochist, a foot fetishist, or are working on a highly detailed cinematic about pedicurists, you’ll find that articulated toes are pretty straightforward to build but consume a lot of effort when they need to be animated. The main surprise is that the big toes have only two joints, while the rest of the toes have three; this is because the big toe’s primary job is to lever the foot up, whereas the other toes grip the ground to provide traction. The big toe functions much like the main foot joint in a single joint rig. If you don’t care about the grasping action of the other toes, you can use the big toe to control a driven key setup on the rest, which will simplify the business of animating the other toes a great deal.

SHE BLINDED ME WITH SCIENCE

We’ll return to “Anatomy for Animators” in the near future, but in the meantime, don’t forget that these guidelines are intended to be aids to making better skeletons, not rigid rules to be obeyed in every case. The final test is always the artist’s eye, not the scientist’s textbook.

Still, the eye can always be taught to see more clearly. It’d be good for the entire profession if we all set as much importance on anatomical understanding as our colleagues on the pencil-pushing side do. ❌

JAMES PAUL GEE

CONTINUED FROM PG 12

serious games company, is probably the best example that we have right now.

BS: *Have you ever gotten in trouble for your defense of games?*

JG: I expected when I wrote this book that I would get all sorts of hostile response, but we have found, by and large, very positive response. I think it’s because people know that our schools are not giving rise to deep learning.

I don’t know if you’ve read Thomas Friedman’s new book *The World is Flat* [Farrar, Straus and Giroux, 2005]. He’s a well-known neo-liberal columnist for *The New York Times* and basically argues that the American education system is in deep trouble because we’re not training people to be innovative and creative anymore; we’re just training them to do the basics, the skill and drill, whereas countries like India and China, which previously took our low-level jobs, like our manufacturing jobs, have now taken our high-level jobs. He says that in the future, whether a job is high status or low status, if all that job involves is standardized skills, it’ll go to India or China, and so the only high level jobs that will stay here are those in which people can be creative and innovative.

Eventually the game industry will outsource a tremendous amount of its programming. The future belongs to young people who are tech-savvy and are comfortable with technology. Kids who get into computer games and video games often use that as their entry to become tech-savvy. They build mods, they build web sites, they form guilds, and if we don’t start producing, both with boys and girls, a lot more tech-savvy people, our economy’s going to eat it.

My own book has often been read to say that I think we should put games into schools, and I think that’s a great idea, but what the book is really arguing is that game designers have forced themselves to engage in good learning to get people to play something that’s long and hard, and we should look at the principles of what they’re doing, many of which are supported by research and put those principles into schools, whether they’re integrated in games or not. So I’m really saying that people have many things to learn from the game

industry about how learning works and how to engage people and motivate them. That’s true even if you don’t want to build games, even if you want to build some other type of learning.

BS: *What are the implications of states trying to restrict games?*

JG: I don’t think any of that will have any effect at all. I don’t even know why the industry is that frightened of it, aside from some sales being lost. The technology is inevitable. You know, when book literacy came around, when print arose, people tried to restrict who could read. They didn’t want working-class people reading because they might rebel, they might get liberal. Every new technology has been feared by older people who tried to restrict it. It has never worked. Books are everywhere, including pornography, including very violent books. We’ve tried to restrict every technology—it will have no effect. What I think we should really worry about though, is the economy of states, especially those that already have good universities and some military presence. Digital media is going to be the new biotech. It’s going to be as important to the economy of some states in building digital entertainment and digital media into a variety of niches as biotech was. And there’s going to be a synergy between the military, universities, and businesses that want to build games, as there was with biotech, and if the government keeps demonizing it, they are going to jeopardize the economy of these states, and the national economy in an area that is very crucial for the future.

This is an industry that we often view only in an entertainment light, but once the military and schools use it, it’s going to be a huge business of building digital worlds and digital games all over the place, and the people that get with it first are going to do very well. I know the governor of Virginia is trying to push this. He said in a meeting that the only thing that stops him is that he doesn’t want to get tarred over the violence issue. That’s why the game industry has got to communicate to the public that is not just teenage boys. ❌



Join the Pulse of Gaming™

JOBS.MIDWAY.COM

Midway Games Inc. is a leading developer and publisher of interactive entertainment software for all major video game systems, releasing some of the most popular games in video game history.

Our goal is to create the most compelling, high-quality games around. We're always searching for world-class talent with energy, commitment to excellence and unrestricted imaginations to join our exceptional teams.



MINNEAPOLIS

Producer

SEATTLE

Producer
Lead Technical Designer
Lead Effects Designer
Lead Writer
Lead Animator
Lead Gameplay Programmer
Technical Producer
Adult Creative Director
Tools Programmer
Tools Design

LOS ANGELES

Project Tech Director
Project Art Director
QA Supervisor
QA Designer
Lead Programmer
Sr. Graphics Programmer
Sr. Online Programmer
AI Programmer
A. Level Designer
Associate Producer
Producer
Effects Artist
Technical Artist

SAN DIEGO

Producer Development
Associate Producer
QA Supervisor
QA Lead
Software Engineer
New Product Development
Localization Manager

AUSTIN

Sr. Software Engineer
Software Engineer
Producer
AI Programmer
Senior Designer
Level Modifier
Concept Artist
Associate Producer
Network Programmer

DALLAS

Sales Development

CHICAGO

Gameplay Programmer/PC
Post Lead
AI Programmer
Gameplay Programmer
Software Engineer
Sr. Software Engineer
PS2 Platform Lead
PC Platform Programmer
Sr. Programmer
Software Engineer
Programmer
Sr. Game Designer
Game Designer
Lead QA Artist
QA Artist
Animator
Lead Environment Artist
Sr. Environment Artist
Environment Artist
A. Environment Artist
Character Artist
A. Character Artist
Concept Artist

SPRINGFIELD

Technical Programmer
Game Programmer
Character Animator
Designer
Script Writer
Cinematographer

ADelaide

Junior Artist
Senior Artist
Sr. Character Animator
A. Character Animator
Level Designer
Programmer
Studio Creative Director
Studio Art Director



CAREEN YAPP

» BUSINESS LEVEL

WORKING WITH HOLLYWOOD

THE VIDEO GAME INDUSTRY IS CURRENTLY

the darling of the entertainment industry. Just pick up any Hollywood trade magazine and there will be an abundance of stories testifying that interactive entertainment is the place to be. Everyone—directors, writers, and actors—wants to at least have a toe in the business, and where they want to go, their agents will lead them. Attempting to schedule a meeting with every person in the entertainment industry that is interested in working in games could keep a person booked 24 hours a day, seven days a week.

Surely, this must be fantastic. Imagine access to all of that talent—how could it fail? Well, there can be a litany of potential difficulties, yet there are certainly advantages that result from the influx of talent and access to Hollywood-related assets that the interactive entertainment industry is increasingly receiving for games such as PETER JACKSON'S KING KONG.

Put simply, the focus and attention being given to games from the entertainment world is good for the business. Publishers and development teams are being allowed unprecedented access to talent and are allowed to run with the ball and create standalone experiences that extend the original IP, and the result is often better entertainment. More often than not, brainstorming meetings are scheduled at the beginning stages of game development. Writers, artists, directors, and animators sit down with producers, designers, and programmers to share ideas and agree on the general direction of the game, and everyone is given much more freedom.

A MEETING OF MINDS

It is an experience to be in a room filled with creative talent from both industries.

The collaboration will likely get to the point where the game designer and film director will become so excited by the creative opportunities that they will seem to speak in a secret language. It will sound less like a conversation in a foreign country and more Disney's affable chipmunks, Chip and Dale, conspiring to build a contraption to store their nut bounty. Of course, this invaluable exchange of ideas makes for a richer experience for the audience. Often, film and television creative teams provide original characters, environments, and storylines that they have developed and are unable to use in their own projects, or an original offshoot of the core story can be developed, as the creators of *Hi Hi Puffy AmiYumi* are doing for us at D3 Publisher.

The result of this creative interaction provides a litany of benefits. The film and television creative teams are happy to see their beloved characters live on in the game space, and the game development team receives in return a pre-approved way to help build out its game design. The audience is happy to receive a game that isn't just a carbon copy of a film or television show.

Developing additional original content can range from a show writer editing the game script dialogue to incorporate the essence of the show, all the way to award-winning directors of a major motion picture changing the ending of their franchise, as Shiny's *THE MATRIX: PATH OF NEO* is doing. But overall, there seems to be greater trust, and that can only be good for the game business.

KNOWING YOUR POISON

As much as Hollywood enthusiasm and willingness to work in creating an engaging title can help, it can also get in the way. It's important that the talent working on an interactive project have a basic knowledge of the game development process. Knowledge of timing constraints associated with development can help to ensure that

meetings are scheduled and approvals are received when needed. It can be frustrating to reach alpha, only to receive a list of proposed changes from an approval request originally sent two months ago.

It's also helpful if the talent understands not only basic game play, but also the potential limits when it comes to game design. If an award-winning writer does not understand the basic content required for a game story line and dialogue, he or she might end up writing an amazing script that features little that relates to the title's gameplay, or the addition of new and exciting characters with fantastical physics-defying "moves" that require technology that exceeds the limits of your game engine. The issues that arise from working with some of the most creative minds in entertainment can be resolved, but it takes time and unfortunately, time is a developer's most valuable commodity.

A HAPPY MARRIAGE?

An influx of talent from the entertainment industry, both at lower and higher levels, is inevitable, and the addition of some of the most creative and imaginative writers, directors, animators, and artists can do nothing but help our industry move forward in its quest to create memorable interactive experiences. Whether these talented individuals fully understand our industry is not the point; their contributions can help make better games.

From a developer's point of view, when working on licensed IP, we've learned the following lessons. Take the time to meet with creators at the beginning of the development process to present your team, procedure, and goals. Meet with as many members of the film and/or television creative team as possible. You never know who will hold the key to pulling everything together. Most people are willing to try and resolve the issues that arise when merging one medium into another because, in the end, no one really wants a bad game, much less a bad game that does not sell well. Viva la union! ❖



THE MATRIX: PATH OF NEO is a good example of games and Hollywood getting together to extend a franchise.

CAREEN YAPP is vice president of worldwide licensing and business development for D3 Publisher of America and was previously senior manager of licensing for THQ and THQ Wireless. Contact her at cyapp@gdmag.com.



ALEXANDER BRANDON

❖ AURAL FIXATION

BRINGING DOWN THE HOUSE

A LOT OF ARTICLES IN TECH-RELATED

publications tend to examine technical problems and new technical innovations. This is wonderful and vital information, but there is another equally vital element to the equation of game development: the human factor. This is true for all elements of development—art, programming, design, and audio.

Hardly anyone writes about human technology for very good reasons. Every human being is different and has different motivations. How can you create solid rules for dealing with people when each person responds differently? A big, related topic in the game industry in general is outsourcing, which has to do with the physical location of humans and how people in disparate places interact. Let's talk about how it's used in the audio world, in-house versus out-of-house. Both methods have benefits, and we'll look at three each.

IN-HOUSE

More communication. Someone in an office down the hall is more accessible than someone you need to reach by phone or email alone. An average of 15 to 20 percent more communication takes place when you compare in-house to out-of-house employees. To back this up, in a personal analysis of how often I spoke to a producer, sound designer, and composer in my latest project (in-house) as compared to a previous project (out-of-house), I found an average of 10 emails were sent or received per day, an average of five of which were responded to that same day when working in-house, compared to three for out-of-house. In

addition, an average of two face-to-face conversations took place per day during the last two months of production in-house compared to one face-to-face conversation per month out-of-house and one phone call every two days.

And the in-house communication is quite effective. The easiest example to illustrate this is when a producer needs to demo the game with the content creator present on a regular basis. Such regular meetings are prohibitively expensive with out-of-house contractors.

Communication needs change depending on the content, as well. For music, less back and forth communication is required as the material itself is more subjective than programming, for instance. Sound effects and voice over are a different matter, though, and having your talent and content creators close by can be extremely beneficial for fast turnaround and change requests.

Multiple projects. An in-house staff is devoted only to the projects of one company. Out-of-house contractors are invariably working for multiple companies. You can intelligently leverage an in-house team to be scheduled for multiple projects, which causes costs to drop dramatically.

Ownership. In-house staff content is nearly always owned outright by the company that hires them. There is no confusion about residuals or bonuses. Need another piece of music? Ask for it and it's done. There's no need for additional contracts and legal fees.

OUT-OF-HOUSE

Choice. You have more choice when you hire an out-of-house contractor to complete your audio work. There are



more styles available from more content creators, each with their own background and experience, when compared to using a dedicated in-house team.

Decreased cost in certain areas. While you'll be spending more in legal fees on out-of-house contractors, you won't be paying for office space, insurance, or benefits. Then there's the overseas argument that seems to be working for a number of companies, which contract work from China and the Ukraine.

Single projects. If you're working on one project at a time, it is quite possibly cheaper to hire out-of-house contractors. A single composer can work for one year and produce an average of 120 minutes of good quality music for about \$1,000 per minute (of finished music) at a \$69,300 average salary (based on *Game Developer's* 2005 Salary Survey). An out-of-house composer can often charge less. A well known Texas-based musician-for-hire charges roughly \$800 per minute, for example.

NUMBER CRUNCHING

There are situations that demand both approaches; the hard part is accurately measuring your costs for both scenarios to decide which will work best and be most cost efficient for each project. If you can accurately do that, especially in-house, you can make an intelligent judgment on your financial plan for each project. ❖

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



NOAH FALSTEIN

» GAME SHUI

STEALTH EDUCATION

THE TERM “STEALTH EDUCATION” HAS been getting a lot of press recently with the rise of training and educational games. To my knowledge, the term was coined by Douglas Crockford while working on an educational title for Lucasfilm Games in 1987.

The idea is pretty straightforward: make a game with no overt teaching in which the player’s enjoyment is enhanced the

education formula is to make the game fun in its own right, regardless of the educational component. In *CARMEN SANDIEGO*, the rationale for the gameplay involved a fun, apparently non-educational focus: finding and catching a master criminal.

The consequence of this structure was that parents bought the game for their kids under the pretense that their kids would learn something. And kids enjoyed the game, reassured that they didn’t have to learn anything they didn’t want to. But for many sleuth players, they gradually found that they enjoyed the game enough to start looking up the facts instead of blindly guessing. Stealth education in action.

THE RULE

Make learning the educational content of a game optional, but integral to maximizing enjoyment of the game.

THE DOMAIN

The domain of this rule seems at first glance to be educational games. But as I noted in my March 2004 column (“Beyond Entertainment”) all games are educational at heart. In fact, every successful game that involves learning lots of facts has made it easy to start the game without really knowing much at all about what the units or characters do. This includes quite a range of titles, from *POKEMON* to *MAGIC: THE GATHERING* as well as all strategy and role-playing games.

Then they make it easy to learn more in the context of the game and make that learning critical to maximizing the player’s enjoyment. So if stealth education involves making a game with fun gameplay that just happens to be more fun when you learn more about the facts behind that play, the only real distinction between an educational game and a purely entertaining one is the nature of the facts you are learning.

THE RULE TRUMPS

This rule trumps the “gameplay comes first” rule. One of the few places where content can be more important than delivering fun gameplay is in the

educational realm. Stealth education can mitigate this conflict by teaching just the “fun facts” for a game.

THE RULE IS TRUMPED BY

The stealth education rule is trumped by the “some facts just aren’t fun” one. I recently consulted on a project for a company that teaches compliance with state and federal regulations and laws to health workers. No matter how you slice it, learning about red tape and regulation can be painful. But that doesn’t mean you shouldn’t at least try to make it better. An interactive game to teach legal procedures doesn’t have to be fun enough to compete with the latest *STAR WARS* game; it only needs to be more fun (or more effective) than other methods of learning the same information. Even “dull” can trump “excruciatingly boring.”

Stealth education is also trumped by sheer difficulty. It’s hard enough to make a fun game with no real-world educational requirements at all. It becomes significantly harder when you have to teach something too. In a pure entertainment game it’s often possible to bend the rules and create new creatures, settings, and technology. When you’re trying to teach something real, you can still selectively bend those rules, but you risk losing credibility when you do so.

Sid Meier’s *CIVILIZATION* is being used in some college classes to teach the actual rise and fall of civilizations—but only in conjunction with other materials that fill in the corners or illustrate where Sid’s *CIV* oversimplifies. Jared Diamond’s book, *Guns, Germs, and Steel: The Fates of Human Societies* (Norton, 1997) is a great complementary companion. Games like *CIVILIZATION* excel at giving the player an intuitive sense of the technological and political forces involved, but aren’t as good at teaching basic facts about what happened when in our history.

But in those cases where it is possible to both teach and entertain effectively, the stealth education rule makes for a great way to have your cake while learning about baking it too. ❖

Game Options Home Dossiers



Games like *WHERE IN THE WORLD IS CARMEN SANDIEGO?* can teach, but have their limits.

more he or she learns about the subject matter. The player learns without even realizing teaching is taking place.

The first big success in this area was Broderbund’s 1985 release *WHERE IN THE WORLD IS CARMEN SANDIEGO?* for which the designers took the innovative step of making the learning optional. Although the core game involved answering geography questions, you could actually play through pure guesswork and have a fairly good time. But the more you learned about world geography (and a world almanac was included with the game), the better you did.

Another key component of the stealth

NOAH FALSTEIN is a 25-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. Email him at nfalstein@gdmag.com.

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



NOW HIRING

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



Jobs at Nintendo of America Redmond, WA

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

Jobs at Retro Studios Austin, TX

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

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



www.nintendo.com



www.retrostudios.com

WANTED

ARTISTS, PROGRAMMERS
DESIGNERS, PRODUCERS



GUN

TONY HAWK'S
AMERICAN WASTELAND

REWARD

A KILLER PLACE TO WORK



Apply at www.neversoft.com



Eugene, OR

BACKBONE



Emeryville, CA
Vancouver, BC



the
collective

Newport Beach, CA



imaginengine™

Boston, MA

JOIN OUR TEAM

FOUNDATION
ENTERTAINMENT

Visit www.f9e.com
for current job listings

VFS™

VANCOUVER FILM SCHOOL
WHERE RESULTS MATTER

GAME DESIGN

Work Hard, Play Harder

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

For detailed program information call **1.800.661.4101** or visit vfs.com.

» GET EDUCATED

The difference
between playing and gaming
is the difference between
life on the playground
or digital life and death.



COMPUTER ARTS-
NEW MEDIA



PHOTOGRAPHY



ARCHITECTURE



INTERIOR
ARCHITECTURE
& DESIGN



ILLUSTRATION



GRAPHIC
DESIGN



FINE ART



MOTION PICTURES
& TELEVISION



FASHION



ADVERTISING



INDUSTRIAL
DESIGN



ANIMATION &
VISUAL EFFECTS

SCHOOL OF
Animation &
Visual Effects



REGISTER NOW
FOR SPRING-
CLASSES START JANUARY 30

1.800.544.ARTS | www.academyart.edu

79 New Montgomery St., San Francisco, CA 94105

Nationally Accredited by ACTEC, NASAD, FIDER (NPA-IAD), NAAAB - Candidate Status (M-AROB)

ACADEMY of ART UNIVERSITY

FOUNDED IN SAN FRANCISCO 1929

BY ARTISTS FOR ARTISTS



SERIOUS GAMES SOURCE



Access the only email newsletter dedicated to serious games industry news, features, and career growth opportunities.



EDUCATION



GOVERNMENT



HEALTH



MILITARY



CORPORATE



FIRST RESPONDERS



SCIENCE

SIGN UP TODAY AT
www.seriousgamesource.com



create YOUR WORLD

CERTIFICATE PROGRAMS AVAILABLE

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

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

CONTACT US TODAY: call 800.808.2342 or visit www.cdiabu.com

BOSTON UNIVERSITY
 The Center for Digital Imaging Arts at Boston University

Collins College >> Your passion. Our profession.

Add a new dimension to YOUR life.

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

STOP BY OUR BOOTH #2204 AT SIGGRAPH.

CALL NOW FOR INFORMATION **1-888-356-7777**
www.CollinsCollege.edu

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

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

by Rick O'Connor

The Future

Brought To You
By Inspiration
—And Coffee



www.uat.edu or 800.658.5744

Available online or on-campus

- | | |
|-------------------|-----------------------|
| GAME DESIGN | VIDEO PRODUCTION |
| DIGITAL ANIMATION | NETWORK SECURITY |
| ARTIFICIAL LIFE | SOFTWARE ENGINEERING |
| MULTIMEDIA | TECHNOLOGY MANAGEMENT |
| WEB DESIGN | GAME PROGRAMMING |

Over 50 Non-Union
Voice (VO) Talents.
Over 10 million
units sold.

Voices
SFX
Music
for games

www.chakrasound.com

ADVERTISER INDEX

Academy of Art University	44	Integrity Ware Inc.	8
Ageia/Novodex	10	Midway Games	37
Alias Systems	C2	Neversoft Entertainment	42
Anark	6	Nintendo of America Inc.	41
BreakAway Ltd.	3	Perforce Software	19
Center for Digital Imaging	45	Rad Game Tools	C4
Chakrasound	46	Seapine Software Inc.	17
Collins College	45	Softimage Co.	C3, 23
Emergent Game Technology	13	The Collective Inc.	43
Full Sail Real World Education	47	University of Advancing Technology	46
Havok	25	Vancouver Film School	43

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

**Bachelor of Science &
Associate of Science
Degree Programs**



school of

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

800.226.7625
fullsail.com

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

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

3300 University Boulevard • Winter Park, FL 32792

©2005 Full Sail, Inc. All rights reserved. The terms "Full Sail," "If you're serious about your dream, we'll take your dream seriously," "Full Sail Real World Education," and the Full Sail logo are either registered service marks or service marks of Full Sail, Inc.



**UNITED STATES POSTAL SERVICE
Statement of Ownership, Management, and Circulation**

1. Publication Title: Game Developer. 2. Publication No.: 1073-922X. 3. Filing Date: September 30, 2005. 4. Issue Frequency: Monthly. 5. Number of Issues Published Annually: 12. 6. Annual Subscription Price: \$49.95. 7. Complete Mailing Address of Known Office of Publication (Not Printer): CMP Media LLC, 600 Harrison St., San Francisco, CA 94107. Contact Person: Kenneth Savoy. Telephone: 516-562-7297. 8. Complete Mailing Address of Headquarters or General Business Office of Publisher (Not Printer): CMP Media LLC, 600 Harrison St., San Francisco, CA 94107. 9. Full Names and Complete Mailing Addresses of Publisher, Editor, and Managing Editor: Publisher: Michele Maguire, CMP Media LLC, 600 Harrison St., San Francisco, CA 94107; Editor: Simon Carless, CMP Media LLC, 600 Harrison St., San Francisco, CA 94107; Managing Editor: Jill Duffy, CMP Media LLC, 600 Harrison St., San Francisco, CA 94107. 10. Owner: CMP Media LLC, 600 Community Drive, Manhasset, NY 11030, an indirect, wholly owned subsidiary of United Business Media PLC, Ludgate House, 245 Blackfriars Rd., London, SE1 QUJ, U.K. 11. Known Bondholders, Mortgagees, and Other Security Holders Owning or Holding 1 Percent or More of Total Amount of Bonds, Mortgages, or Other Securities: None. 12. Tax Status: Has Not Changed During Preceding 12 Months. 13. Publication Title: Game Developer Magazine. 14. Issue Date for Circulation Data Below: October-05.

15. Extent and Nature of Circulation:	Average No. Copies Each Issue During Preceding 12 Months	No. Copies of Single Issue Published Nearest to Filing Date
a. Total No. Copies [Net Press Run]	42,260	43,980
b. Paid and/or Requested Circulation		
(1) Paid/Requested Outside-County Mail Subscriptions Stated on Form 3541. [Include advertiser's proof and exchange copies]	31,563	32,043
(2) Paid Inside-County Subscriptions Stated on Form 3541. [Include advertiser's proof and exchange copies]	0	0
(3) Sales Through Dealers and Carriers, Street Vendors, Counter Sales, and Other Non-USPS Paid Distribution	3,780	3,772
(4) Other Classes Mailed Through the USPS	0	0
c. Total Paid and/or Requested Circulation [Sum of 15b. (1), (2), (3), and (4)]:	35,344	35,815
d. Free Distribution by Mail [Samples, Complimentary, and Other Free]:	0	0
(1) Outside County as Stated on Form 3541:	0	0
(2) In-County as Stated on Form 3541:	0	0
(3) Other Classes Mailed Through the USPS:	0	0
e. Free Distribution Outside the Mail [Carriers or Other Means]	1,944	2,660
f. Total Free Distribution [Sum of 15d and 15e]	1,944	2,660
g. Total Distribution [Sum of 15c and 15f]	37,288	38,475
h. Copies Not Distributed	4,972	5,505
i. Total [Sum of 15g and 15h]	42,260	43,980
Percent Paid and/or Requested Circulation [15c Divided by 15g Times 100]	94.8%	93.1%

16. Publication of Statement of Ownership: This Statement of Ownership will be printed in the November 2005 issue of this publication. 17. Signature and Title of Editor, Publisher, Business Manager, or Owner (signed): Michele Maguire, Date: September 26, 2005.

» A THOUSAND WORDS



BREAKAWAY GAMES' FREE DIVE

FREE DIVE is a serious game developed using BreakAway's Crate Engine, for use in physical therapy. The game allows users to interact with an underwater world, searching for treasure and visiting sea creatures. Art by: BreakAway Games' art team, under art director Steve Langmead.

Seat of power.

SOFTIMAGE® | XSI® v. 5.0

- Work with ten times the detail
- Easy migration from Maya
- Film quality normal maps
- mental ray® v. 3.4

"The *Prince of Persia - The Two Thrones*' cinematics came to life with **SOFTIMAGE | XSI**, allowing us to fine-tune these characters and environments using one single tool, from modeling to texturing all the way through animation and rendering."

Ubisoft Cinematic Team 2005
Prince of Persia® The Two Thrones

© 2005 Ubisoft Entertainment

Come to softimage.com/tive
starting at \$495

Interested in migrating to XSI?
Visit softimage.com/changeisgood
for details.

SOFTIMAGE | XSI



Prince of Persia
YER YONG YERASH

© 2005 Ubisoft Entertainment. All rights reserved. Ubisoft, Prince of Persia, YER YONG YERASH, and the Ubisoft logo are trademarks of Ubisoft Entertainment in the U.S. and/or other countries. Prince of Persia is a trademark of Avid Technology in the U.S. and/or other countries and under license to Ubisoft Entertainment.

© 2005 Avid Technology, Inc. All rights reserved. All other names, trademarks, and logos are the property of their respective owners. Contact your local Avid representative in the U.S. and Canada, Product Support, 2200 Linden Drive, Sausalito, CA 94965, or visit www.avid.com for more information. All other names are the property of their respective owners.

Avid
computer graphics

Bink takes the grind out of game video!

The video codec for games

Bink has shipped in 2,500 games for a reason. It is the standard for video in games - it is faster (up to 4 times faster), it uses less memory (up to 18 MB less), it has a built-in audio codec, it supports every platform, multiple audio tracks, data interleaving, alpha and RLA files (for Z-depth, normal and uv per-pixel data), and much more! Now supports HD video on Xbox!

It's like having your own codec

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

RAD knows games

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

New!

**Now shipping for Xbox 360
and Nintendo DS!
New Japanese Manual!**

For more information:

425.893.4300

www.radgametools.com



THE BEST IN GAME DEVELOPMENT TECHNOLOGY



Pixomatic
Software
Renderer



Miles
Sound
System



Pixomatic
Software
Renderer

*PlayStation® is a registered trademark of Sony Computer Entertainment Inc.