

---

# Tools & Middleware Newsletter Vol.9

## GDC 2003 Report

### Table of Contents

#### Game Developers Conference San Jose, CA March 4-8, 2003

- **Acacia Breakfast Seminar: Game Development Market Trends**
  - **Acacia Keynote Address**
  - **Acacia Panel Sessions**
- **GDC 2003 Conference Sessions**
  - **10 Reasons You Don't Want to Make a Massively Multiplayer Game**
  - **SOCOM: Bringing a Console Game Online**
  - **Game Editing Tools**
  - **Effective Evaluation of Middleware**
  - **Maintaining, and Improving 3D Art Production Pipelines**
  - **Facial Animation for Game Characters**
  - **Physics-Based Synthesis of Sound Effects**
  - **Exploring the Fringes: Interactive Entertainment for the 21st Century**

---

## Game Developers Conference San Jose, CA March 4-8, 2003

*By Christine Arrington, Samantha Staples, & Michael Arrington*

The 2003 Game Developers Conference was held in San Jose, California March 4 - 8 with networked gaming taking the spotlight. Session ranging from gaming on wireless devices to online console game development highlighted that the development community is tightly focused on enabling gamers to come together on a variety of platforms in a variety of ways.

Sony Computer Entertainment America had a large presence at the event, showcasing SOCOM: Navy Seals in one of their online pods as well as demonstrating some cutting edge interface technology which allows the user to interact with PlayStation®2 without ever touching a controller.

Sony Computer Entertainment America's March 6 session on the development challenges behind SOCOM is detailed below. Earlier that morning Teiji Yutaka, Vice President, Software Development Department, R&D Division of Sony Computer Entertainment, Inc. delivered the Keynote address at Acacia Research Group's Game Title Development Trends Breakfast.

### Acacia Breakfast Seminar: Game Title Development Market Trends

At Acacia Research Group's fourth annual GDC executive forum we heard from nearly a dozen industry experts on the future of making game titles for consoles and other platforms. The event, held March 6th, kicked off with a keynote address entitled Expectations for the Tools and Middleware Industry from a Console Maker from Teiji Yutaka, Vice President, Software Platform Development Department, R&D Division of Sony Computer Entertainment Inc.

## Acacia Keynote Address

Tools and middleware have taken on an important role in console game title development and Sony Computer Entertainment Inc. has been driving the move toward their use in developing game titles for PlayStation®2. Mr. Yutaka opened with a bit of history about Sony Computer Entertainment Inc.'s Tools & Middleware Licensee program and detailed why Sony Computer Entertainment Inc. is backing the use of these products so strongly.



*Copyright © 2003. Sony Computer Entertainment Inc. All rights reserved.*

In 1996 the company launched Net Yaroze for PlayStation® to allow gamers the opportunity to develop their own game titles for the platform. The company encouraged this experimentation by PlayStation® fans and the platform generated intense interest in the hobbyist community driving more interest in game title development for the console. It also encouraged creativity in new game ideas.

After its experience with the small but enthusiastic audience for Net Yaroze, the company took the next step in providing development assistance to a broader range of third party developers. PlayStation®2 Tools & Middleware program was launched in 1999 with specific goals in mind. The first was to help developers increase productivity through the use of tools and middleware. The company saw that these products could help developers manage costs and increase the production values of their game titles to rival the film industry. Second, the program focused on helping developers introduce cutting edge technologies seen in industries such as military and government, industrial visualization and filmed entertainment into the world of console gaming.

Mr. Yutaka's assessment of the value of tools and middleware in the production of game titles for PlayStation®2 is that the class of development products has crossed a threshold making them critical to the game software development process. As the power and complexity of the hardware increases developers are much better served by integrating these aids into their development process.

Mr. Yutaka described two completely different game software development styles and how the market has changed from the Vertical style where developers spend time creating a

development process and tools that are unique to the game software in development. Programmer skill levels are very high and the quality and costs of development are in turn very high. As the power of the platforms and the complexity of developing for them has increased a Horizontal development style has taken hold. This style demands that developers create a framework that can be reused and can, while maintaining high quality standards, keep the costs of development low.

When developers moved from PlayStation® to PlayStation®2 the dynamics of the hardware and the industry pushed the development style of most studios to the horizontal style. Mr. Yutaka noted that the Japanese development community has maintained much of the vertical style, while the American development community has moved to the horizontal style.

To support this change in how developers needed to work and further the efficiencies of production, Sony Computer Entertainment Inc. launched PlayStation®2 Tools & Middleware Licensee program. The program is designed to help developers bring new ideas to gaming rather than spending resources on developing their framework.

Mr. Yutaka pointed out that cutting edge technology advances in game hardware have come every ten years or so. In 1972 the first game console, the Odyssey, was launched. Just over 10 years later in 1983 advanced 2D graphics were the next paradigm shift with the introduction of the Nintendo Entertainment System and slightly more than 10 years later in 1994 3D graphics acceleration once again changed the gaming experience.

Gamers are now starved for a totally new experience from computer entertainment system, according to Mr. Yutaka. Allowing developers to focus on that creative endeavor rather than creating tools and middleware in-house is one of the main reasons Sony Computer Entertainment Inc. launched its Tools & Middleware program. It now has 150 licensees participating in the program.

Mr. Yutaka predicted that the demand for tools and middleware will grow and that the demand of a new user experience will come about because we are on the verge of the next new technology revolution.

## **Acacia Panel Sessions**

During the panel Advances in Gaming Hardware: Consoles, PCs, & Beyond, Intel's Robert McNair kicked off with his company's near-term vision for handheld technology. He said the company sees a convergence between PDAs and cell phones, increasing connectivity rates, video playback capability, advanced graphics, and increased battery life as the key upcoming advances.

One key to expanding the potential market for games is through content that spans multiple platforms - consoles, handhelds, PCs, set-tops, etc. - according to McNair. Everyone plays games of some sort, said i3Dimension's Anil Sabharwal, and more and more will be playing those games in electronic media as time goes by - and as the current generation grows up.

Getting across all of those platforms will require at least some level of standardization, according to Sabharwal. This is amplified in the case of handhelds, set-tops, and similar devices, said Neil Trevett of the Khronos Group - which is striving to bring OpenGL to the embedded market.

In the panel Pipeline Productivity: The Integrated Tool Chain in Game Title Development, participants from tools and middleware companies discussed the growing interactions between components in the game software development pipeline. In general the consensus seemed to be that things are getting better, with many clear paths beginning to emerge from content and data to game engine.

Even so, Criterion's Adam Billyard said there is still a lot of work to do with regard to gluing together tools and middleware solutions. According to Geoff Foulds of Alias|Wavefront, the customer is king and you have to give them what they want in tool integration and pipeline features. Both of those companies' products, for instance, are tightly integrated with those of co-panelist Gregor vom Scheidt of NXN Software.

As development teams grow ever larger, teams will be looking for ever more toolchain integration. According to Brian Gildon of Metrowerks and Jeff Yates of discreet, toolmakers need to focus on their piece of the pie, making the best tools they can while ensuring compatibility. Gildon said that no one company could ever hope to do it all.

## **GDC 2003 Conference Sessions**

This year's GDC conference included a number of lively, informative presentations and panel sessions. A few of the most interesting are covered in the following sections.

### **10 Reasons You Don't Want to Make a Massively Multiplayer Game**



*Copyright © 2003. Sony Computer Entertainment Inc. All rights reserved.*

Online gaming, specifically Massively Multiplayer Online Gaming, was a huge topic of discussion at GDC 2003. The topic has finally come out of the smaller back rooms of the show and made its way to the large facilities at the convention center and were typically greeted by standing room only audiences.

It seems, however, that many presenters were doing all they could to scare developers away from making MMOGs. Gordon Walton the executive producer of The Sims Online gave the crowd a sobering assessment of the challenges

in developing such an ambitious product. After the 10 Reasons You Don't Want to Make a Massively Multiplayer Game session on March 6, no developer in the room should have any

illusions about the difficulty of developing and deploying a successful MMOG.

Walton cautioned that there are too many MMOGs in development right now, most of those targeting the PC gamer.

Walton's list read as a serious warning to potential developers:

**#10. Too many are being built.**

As with any genres that have proven to be successful the potential payoff and the reports of success have attracted many developers to the idea. Walton said that with over 100 titles in development the market just could not support all the game titles. Most users will not be willing to pay the subscription price and the expansion pack prices on more than a few game titles. That limits the market unless a developer creates a game title that breaks out of the copycat mode and provide the user with a new experience.

**#9. The craft requires mastery of too many disciplines.**

Game title developers are quite aware of the needs for graphic artists and lead programmers, but add in the need for mastery of billing systems, community relations, and management of a subscription based product and suddenly the team is spread too thin across too many disciplines which leads to reason number 8.

**#8. It requires a huge team with diverse skill sets.**

Developers getting into the MMOG space are finding out that they need to have skill sets on their team that have never been required before. Networking, billing, community building, just to name a few become critical skills in developing this type of game title. Walton drove his point home as he asked the audience to raise their hands as he listed off the skill sets he had on his development team. As he approached the billing and network administration skill sets the number of hands in the air fell dramatically.

**#7. Getting the credit card from the customer is hard.**

This is a challenge for the gaming community. The young male gamer which we usually associate with MMOGs is not likely to have a credit card. Asking a parent for access to a credit card introduces and older demographic which continues to be suspicious of online transactions. Those who do have access to credit cards don't always understand the value proposition of MMOGs until they try it.

**#6. Online games are counterintuitive to packaged-goods company management.**

The launch model for an online game title changes the way game title developers and publishers do business. Rather than a single retail launch the game title is essentially a subscription based product. Developers have to manage the user base and provide new booster or expansion packs to keep the title fresh and compelling for users. Game title developers are used to launching a game title and moving on.

**#5. Everything developers know from making single-player games is wrong for MMOGs.**

Things that are a nuisance in single player game titles become dangerous in multiplayer

game titles. A cheat code that affects only the player who chooses to use it in a single player game title can affect the game title for hundreds even thousands of players. Paying subscribers will not tolerate that interference. Documentation that is often ignored in single player game titles is very important to MMOGs.

#### **#4. The Internet sucks as a commercial delivery platform.**

This piece of the puzzle is very difficult for game title developers to deal with. Internet service is a long chain of equipment, service providers, and users. If something goes wrong the game title developer is often faulted for the problem. Regardless of who is responsible for an outage or security breach the responsibility will, in the users eyes, tend to fall on the content provider.

While Walton was addressing the PC aspect of the market primarily this will likely be more dramatically felt in the console market. While the brand recognition may help users trust the game title on the billing side, they will expect higher quality and reliability from a game title with a console brand attached.

#### **#3. Customer service is hard.**

Running a customer service department for a subscription based business is a lot harder than most developers think. While most businesses think of customer service as a cost center whose expense is to be minimized, in MMOGs customer service is the business. It is costly and labor intensive. It also has to be there 24/7/365. Gaming also doesn't follow the traditional patterns of call centers and customer service where most managers have experience. This is a completely new area with a very dedicated and knowledgeable user base, meaning the customer service staff has to be highly trained and knowledgeable as well.

#### **#2. There are lots of legal issues.**

The legal questions in online gaming are huge. They range from terms of service and end user license agreements to IP protection, labor law violations (if volunteer game managers appear to be working they can claim to be employees and sometimes they win) and frivolous lawsuits. The laws governing online gaming are constantly in flux. Walton suggests that MMOG developers find a good lawyer and budget appropriately to account for necessary legal representation. Walton also pointed out that law enforcement help on security issues was almost non-existent. In the scheme of things in today's environment hacking in online game titles is not taken very seriously or given high priority among law enforcement officials and agencies.

#### **#1. The number one reason You Don't Want to Make Massively Multiplayer Online Games....They cost too much money to build and launch.**

Walton estimated that to build and launch an online game title correctly it would cost \$5 - \$6 million on the build side and an equivalent amount on the deployment side.

However, after all of the cautionary tales the biggest reason to develop MMOGs is the potential upside. While many of the game titles in development may never even see the light of day, some will make incredible amounts of money compared to single-player game titles. The potential for these titles to live on long after the initial launch and generate continuing revenue through subscription fees and expansion packs is what is driving a lot of developers

to the genre.

Walton had some harsh criticism for developers. He said that the MMOG market is stagnating with most developers recreating what has been done before. He asked that developers not pollute the waters with their game titles. He asked that they look hard at their product and make sure they are worthwhile before they are launched. He also warned developers not to frustrate their users with technology, saying if a user is frustrated with your design you will permanently lose that customer.

## **SOCOM: Bringing a Console Game Online**

Seth Luisi, Producer, Sony Computer Entertainment America; Glen Van Datta head of the SCE-RT development team at Sony Computer Entertainment America and Dr. Bob Guttman of Zipper Interactive provided attendees with a detailed look at the challenges they faced developing an online game software for PlayStation®2.

Luisi provided a bit of background on the success of the game software to date. SOCOM has sold through 1 million units in the US since its launch in August 2002 and is scheduled to launch in May 2003 in Europe. As of the March 6 presentation the game software had 300,000 active users. Luisi noted that they time out accounts if the user has not played in the last 30 days. He said that 50,000 to 60,000 people play the game software every day and 170,000 hours are logged by players on the weekends. Over 1 million hours had been logged in the 7 days leading up to the presentation. The game play's peak hours are 3-7 pm Pacific Standard Time and there are generally 11,000 to 12,000 simultaneous users.

Having set the stage for the discussion, Luisi very quickly showed attendees that the usage patterns for a console online game software looked very different than the usage patterns for a PC online game title. Given the loyalty and enthusiasm of console gamers and the backing from console manufacturers on the business side, the numbers for usage look a lot larger than those for even the most successful PC game titles. While this is great for the business, it presents a lot of technical challenges for developers and many of those challenges are new to development teams.

Luisi, Van Datta and Guttman proceeded to lay out those challenges.

In developing an online console game companies must essentially develop two game titles, once stand-alone and one online, all in one game title. Users must be able to purchase the game title for both uses. In creating the online game titles developers must create something that is competitive with PC game titles including having a 16 player minimum to start with and providing voice chat and community support (clans, ladder ranking, etc.).

One of the key aspects of the genre is that developers must make the online part of the game title easy for console users. PC players may have the patience to configure their game titles for online play, but console gamers do not. The user interface must be very well designed with quick and easy features, familiar conventions, persistent appearance, and psychological

grouping. Luisi listed some familiar conventions including the America Online start up screen or Yahoo's directory format. People are used to these online paradigms and developers would do well to keep that in mind when designing their UI. He also pointed out that it is very important that when a user spends time in these online environments the look and feel is persistent throughout the experience, online game title developers need to keep that in mind as well as they are developing their navigation environments. Psychological grouping has created a community feel by allowing users to feel that they are part of a group even though the game play may be happening on one centralized server located in California. In SOCOM users are grouped into USEAST, USWEST, USCENTRAL, etc.

Even though the game title must be stand alone as well Luisi advised developers to focus on the online game play aspect. And, within that find one thing that the game title can do very well and focus on that. Some PC game titles have been in development for four years trying to get every aspect of their game perfect. Console title developers do not have four years to develop their titles so they need to find something they can do very well and get the game title to market.

Luisi was very adamant that developers understand the need for testing. He said that developers need to perform a real public beta test that will expose any problems with the game title. PC game titles can be fixed much more easily than console game titles and developers don't want to be faced with trying to patch a problem after the game title has gone to market. One very important aspect to the public beta test is getting experience with firewalls and other security protocols such as NAT in cable modems. Without a public test the issues of users and their security procedures can become the developer's problem and a huge cost drain in customer support.

Testing is also key in preparing for the unknowns in post-launch server deployment as the user base grows. Developers must be aware that launch is only the beginning. Supporting the game title and user growth after launch can be more challenging than launching the game title itself.

Luisi had some good advice for developers. He said they should look to PC online game titles for guidance, but there is no reason to adopt things that don't apply to the console world. Specifically on the 'do copy' side is to enable in-game player communication. Without player communication there is no community and community is key to the success of online game titles.

Luisi also called for quality control in the online game title market. With everything to pay attention to in the daunting task of adding online play to a game title, do not forget the production values. Keep the console level production values in online game titles as well.

Then, when all that is done, expect the worst. And, reality will be much worse than what you expect. Players will cheat, be rude, and exploit any holes in the game title. Earlier in the session Luisi suggested that developers give players a way to police the game title themselves. In many online game titles players have proven they will do that; even kicking members of their own team off the board if they do any of these things, so make sure users

can do it.

Gutmann also gave some detail behind the technology used in the game software. Zipper used Sony Computer Entertainment America's SCE-RT technology with an added voice chat system. SCE-RT handles the client-server operation, message and scheduled data transmission, and other network processes. The game software's voice chat system was developed by Secret Level Inc. and was based on a public domain G.723 CODEC.

He also detailed some of the things that went well and poorly during development. In general the developer seemed pleased with the performance and integration of SCE-RT with the rest of the game's code base but the product continued to change while SOCOM was in development, causing some difficulties. The key to multiplayer success, according to Gutmann, is getting out of the single-player mindset and focusing on the issues that matter to multiplayer - what things generate data transmissions, what actions should be available to players, what important animations should be limited, etc.

## **Game Editing Tools**

This informative roundtable discussion was attended by a standing-room-only crowd of more than 40 game title developers. The first order of discussion was to define problem areas and there was no shortage of input from those in attendance. Then, solutions were discussed which allowed the developers to take this knowledge back to their own studios, where it could be implemented to help streamline their production pipelines.

One of the first issues brought to light was the difference between engineers designing tools and engineers using the specifications of others to create tools, which can cause problems down the line when artists try to use those tools. It's important to remember that editing tools themselves have a target audience, such as artists, and to work with that audience to make sure the end result is truly functional.

Documentation was discussed as another problem area. Many times an in-house tool's functionality is conveyed through training, but documentation can provide a more stable means for training current and future team members.

One suggestion was that editing tools be engineered like any other piece of software, with established quality control procedures, but the majority decided that this wouldn't work in the very different world of game title development. And at many game studios, it was pointed out, junior engineers are assigned the task of building tools because it's not considered an interesting or exciting task. But the importance of stable editing tools requires that in many cases the senior engineers be required to build them.

In order to avoid having a tool designer diverted to other tasks during crunch time, it's a good idea for larger game title development studios to have an employee dedicated to the task of building tools. If feasible, it's also a good idea to assign someone the task of managing the pipeline, so problems don't result in nonproductive finger-pointing between artists and

engineers.

Many times, tools are designed in a rush scenario and the attitude is to get them done so they fulfill the immediate needs of the artists. However, it's a much smarter idea to consider in advance how long the tool will be needed - such as, for the current title only or for future development - and to build a tool that's stable enough to stand the test of time. In other words, it's more important to do it right than do it quick.

Production tools are not an area where research should be performed. They provide a specific purpose and should be designed to accomplish that purpose as efficiently as possible. Experimenting on editing tools during a production cycle can leave artists with nothing to do as they await the tools they need. The most effective method is to build part of the tool, check it, build a little more, check again, and so on until the tool is complete and ready to press into service. During the checking process, the tool programmer should not only run his own quality assurance tests, but should observe the tool being tested by the end users.

Milestones for tool production should be set at the same time as those for the game title itself - and the tools need to be ready in time to support the game's milestones. Designing all the tools up front can lead to problems as changes are made to the game title itself - this can lead to tools that are no longer effective, resulting in inefficiencies among the team.

Studios frequently do not want to officially set aside time for tool design and development, but one way to make this a less bitter pill is to amortize the costs of tool development over time. Subscription-based game titles require constantly shifting editing tools, and this process whereby tool development is budgeted on an ongoing basis can be emulated by amortization.

Some of the game title developers at this roundtable preferred the idea of one editing tool to perform all the necessary functions, while others thought a variety of tools were more efficient. It was agreed that in many cases the "one tool to rule them all" is actually a variety of tools with a really good interface. And the interface itself can make or break an editing tool - one of the best suggestions at the roundtable was to design the tool's interface to be similar to that of the art tool used in your shop, so there will be a level of instant familiarity for everyone on staff.

Because many game titles follow the spiral model of development - spiraling in towards the target rather than proceeding in a linear fashion, as it were - it's easy to let tool development follow the same path. However, this can result in down time for artists and missed milestones. The development of game title editing tools requires foresight and planning in order to enhance the game title development process, rather than impede it.

## **Effective Evaluation of Middleware**

Middleware is the most significant time and cost-saving advancement to impact the game title development industry in the last few years. But the considerations for choosing middleware

are varied and each studio must decide whether middleware is the right solution - in addition to deciding if the time is right to implement such a solution.

For years, game title development has continued without much in the way of standard business practices. That is beginning to change as budgets reach epic proportions and publishers attempt to instill certain standards at game studios. Publishers look more closely at the bottom line and expect their studios to meet certain milestones. Middleware can be the deciding factor between making or breaking budgets and milestones.

Middleware faces emotional factors that are unique to the game title industry. Because it replaces code that traditionally has been written by the game title programmers themselves, there's a sense that using middleware surrenders control of certain aspects of the game title, causing programmers to have to explain why they didn't write the code themselves. It's as if certain bragging rights are relinquished when programmers don't write all the code.

One of middleware's biggest competitors - besides entrenched attitudes that it's an easy way out - is the code already written that can be re-used from one game title to the next. Most studios already have their own game engines and it's a challenge to convince them to license an external product. A good time, then, to approach the issue of middleware adoption is when a developer is about to switch platforms, a move that would render previous code less viable for a new title.

One of the worst things that can happen in the game title development process occurs when time constraints impact integral parts of the game title. This usually occurs towards the end of the development cycle, when something's got to give - it might be storyline, graphics, cut scenes, or the like. Making the decision up front to use middleware can help alleviate these types of last-minute cuts that rob the game title of some of its personality.

When middleware vendors talk about demonstrable ROI, it most often refers to the fact that choosing to license middleware can result in the displacement of one or more employees. This introduces a real human factor into the purchase decision, and it's a factor that has to be carefully considered as to how it will affect team morale. Game title development studios that are more art-driven, as opposed to programmer-driven, have cultures that will more readily accept middleware as part of the process. Middleware benefits artists directly in that it helps avoid last-minute cuts in graphics quality, as noted above, and it allows them to avoid sitting around while the programmers build a 3D game engine.

There are a couple of minor drawbacks to middleware upon which everyone can agree. First, there will always be certain aspects of the platform that your staff never fully understands because they haven't had to get down and dirty in the code. Second, middleware often becomes the scapegoat for anything that goes wrong in the game title. After all, it's easy to blame the code that no one's around to defend.

Every production pipeline is unique and even the best middleware will require a certain amount of customization. An important purchase consideration is whether the vendor has the tech support capabilities to make this as painless as possible. To some game title

developers, selecting a middleware provider with gaming experience ensures their problems will be well understood, but this seems to be an issue mostly with studios thinking about licensing middleware for the first time. An important concern for everyone, however, is the stability of the middleware provider, as it is not unheard of for some smaller companies to fail and product support to vanish.

Some game title developers feel that licensing middleware causes them to give up their competitive edge, and resistance to change is sometimes based on emotional factors. As more and more successful game titles continue to depend on middleware, it'll become apparent to savvy game title developers that licensing the internal building blocks lets them concentrate on game play itself - and that's the real competitive advantage.

## **Maintaining, and Improving 3D Art Production Pipelines**

This roundtable discussion focused on the issues specifically affecting 3D artists responsible for creating the assets for game titles. The group consisted primarily of 3D artists but several programmers in attendance helped round out the discussion.

There were numerous opinions expressed about the strengths and weaknesses of different modeling & animation tools and their ease of use vis a vis a production pipeline. Many of the opinions were purely subjective, as each artist naturally prefers the tool with which he is most familiar. There was, however, a general consensus about the difficulty in exporting art assets between different types of content creation software.

Part of the discussion focused on the differences between small- and large-scale productions, and how team members are best deployed in each type of situation. For smaller-scale projects, it makes sense to build a certain amount of flexibility into team members' duties and schedules in order to make deadlines. On larger projects, it usually is more effective to have specific tasks assigned to the people best equipped to deliver results on time. Integration is more complex on larger projects but it's ineffective to bring everything and everyone to a halt when it comes time to perform such integration. As unromantic as it may seem, it's definitely more effective to approach asset creation in an assembly-line manner.

Perhaps the biggest pitfall in creating a production pipeline is making sure tools will be available and able to work together in such a way as to insure minimal down time for team members. If the engine is ready to go but the artists are manually creating textures or particle effects, it's time to think about plug-ins - better yet, the time to think about it is before the need arises.

The rigorous requirements of modern game title development require new ways of working, and this session helped artists and programmers alike learn to think outside their immediate areas of concern. As a result, each team member can begin to take responsibility for the overall success of the development process.

## **Facial Animation for Game Characters**

Tito Pagan, Senior Artist/Animator for WildTangent, gave a March 8th presentation at GDC on common techniques for real-time facial animation. Though Pagan noted that the session wasn't intended to cover any new ground it was conceived as a way of increasing the interest of game artists in an area of animation that is often neglected.

Facial animation is an excellent tool for communications, according to Pagan. In addition to body language, the movements of the face help to engage the viewer in the story and can benefit almost any story-based game software. The process, especially with regard to conveying emotion and mimicking speech, is not easy but today's modern animation tools help ease the burden somewhat.

One of the first considerations should be the method used. This depends on what type of game title or product you're developing, the level of detail and range of expression and speech needed, the art pipeline and more. It also depends heavily on the technology supported by your engine - i.e. whether your animation system uses vertex morphing, joint translation, puppeteering, or motion capture.

Key components of the process include building the anatomy of facial features, integrating natural and random head movements, creating and mixing expressions and emotions, and creating a phoneme library.

The first component requires an understanding of how your character's face is put together - where muscles, fatty tissues, etc. are located and how they move. This is fairly easy with a human character, because the information is readily available via reference materials, but for animal and other non-human models it could be more complex. One of Pagan's examples used a humanoid model with stalk-like features - but through proper virtual musculature he managed to impart recognizable behaviors to the model, capable of conveying emotions and speech as well as any human head.

Natural and random head movements - gestures like nods and sub-conscious movements such as eye blinks and muscle twitches - are also very important to bringing virtual characters to life. Adding these behaviors can create a much more realistic experience and create less distraction from the message being offered by the character.

Creating and mixing expressions for the model is also a key component. Pagan noted that there are six base expressions that are recognizable nearly universally - though they don't always mean the same things in all cultures. These are (in Western culture) sadness, anger, joy, fear, disgust, and surprise. Your own goals will determine how exaggerated or subtle you should model these expressions.

A phoneme library is also essential for those animators wishing to create speaking characters. Essentially a phoneme is a phonetic part of speech, broken down into its smallest component. Visually, there are a fairly small number of these - Pagan noted that you can probably get by with less than ten. An example would be a face with the lips in a round or oval shape - such as when you say a long 'o' or 'u'. Tools exist to help animators break spoken audio into recognizable phonemes.

Expression and phoneme libraries can either be built as vertex morph targets or as keyframed bone rotations. Either may be used by a game engine in real-time depending upon the technology included in the game title. For the most part, Pagan seems to use bones to construct his facial rigs. He noted the minimum rig for proper movement is probably around 14, with many more needed for more convincing motion.

## **Physics-Based Synthesis of Sound Effects**

There are times when samples aren't enough for realistic sound effects in game titles. Perry Cook, Associate Professor of Computer Science at Princeton University, has been researching physics-based sound synthesis and control for some time and presented some of his findings at a March 8th GDC session. With physical models playing an increasingly important role in the visual and dynamic aspects of game software creation Cook believes that a large disparity still exists between game graphics and sound.

According to Cook, developers should consider integrating physically driven audio effects into their game titles for some special effects. In addition to being computationally cheap in most cases, these effects have the potential to be highly realistic and many can be driven off of computations already taking place in the game engine.

Though the genesis of audio signals in the world around us is often quite complex, Cook showed some general and cheap solutions for simulating many different types of audio events. Essentially these can be broken down by general type - a vibrating string or bar, airflow through a tube, etc. He showed how to model these using a 'digital waveguide filter' which uses simple recursion, delays, and filters to synthesize the sound according to desired parameters. He also discussed 'modal synthesis' a similar, but non-recursive technique for modeling sounds. The main problem is that there is no single generic system for modeling sound - it requires lots of models.

Other, more complex, methods can yield impressive results but can quickly become computationally expensive. Cook discussed using two and three dimensional models such as waveguide meshes and finite element/difference analysis. These techniques, according to Cook, might incur 4%-5% additional overhead beyond animation but many can ride on the back of calculations that are already occurring in the engine.

By way of example, Cook showed some (pre-rendered) animations done with a colleague who has been working with animating physical simulations. One particularly impressive example was of a bowl hitting the floor, bouncing, rolling, and then coming to a rest. Cook was able to derive data from the actual physical model and the physics of the simulation to create an auditory simulacrum for the animation.

## **Exploring the Fringes: Interactive Entertainment for the 21st Century**

Ernest W. Adams, considered by many to be one of the industry's foremost game title design

experts, gave an unorthodox GDC presentation March 8th on the unusual inspirations that can help game designers think outside of the box. Drawing from a diverse collection of examples ranging from game titles with a message (often controversial) to old school text adventures and MUSHes. His assertion is that the future of video game entertainment may well come from the fringes of interactivity.

It's difficult, Adams said, to quantify exactly what factors combine to help make a game title a commercial success. What is fun and what is entertainment? Different people have different ideas about this and they rarely coincide. One of the most successful game title series of all time and a key early driver of multimedia PCs into consumers' homes, the Myst series, goes against almost all canon precepts of interactive entertainment design - a string of barely connected puzzles accessed via a picture-book walkthrough.

So, how can a developer judge how his concepts will be accepted by the market? One way to start might be changing your notions about computers and game consoles. Adams said that interactive entertainment platforms are essentially just a medium for entertainment - much like oil paints, video, and others. Content doesn't need to reflect the medium and you should abandon your assumptions about how and why people want to be entertained.

Adams showed many examples of the myriad ways people have found to entertain themselves via the medium. Though many of these are admittedly niche they do go to the point of how fun and entertainment are clearly linked to individual point of view. Perhaps one of the oddest examples was VATSIM - an association of professional pilots and air traffic controllers who get together to fly their flight simulator jets (in real-time!) across the virtual world. In the 'game' there's no dog-fighting, no barrel rolls, just the same routine: take off, turn on the auto pilot for six or more hours, and then land. It's what these people do or did (many are retired) in everyday life - and apparently they just can't get enough of it. Granted, VATSIM's user base is only about 50,000 at present.

---

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

\* Company names and products mentioned in the contents are the trademarks of their respective owners.

Copyright © 2003 Sony Computer Entertainment Inc. All rights reserved.