2004 Persistent Worlds Whitepaper

Presented by the IGDA Online Games SIG

The IGDA Online Games SIG Steering Committee:

Daniel James	Editor
Gordon Walton	Editor
Brian Robbins	Chairperson
Elonka Dunin	Editor in Chief
Greg Mills	
John Welch	
Jeferson Valadares	
Jon Estanislao	
Steven DeBenedictis	

FORWARD

As you can see from the list below, many people contributed to this document. The IGDA Online Special Interest Group depends on volunteers to create and edit this paper. While it is as accurate as we could make it in the time available, it certainly has some errors and omissions. Visit the IGDA online games SIG website at http://www.igda.org/online/ for updates and more information if you want to contribute to the next release of the paper or just want to correct any fact or opinion within the paper. We hope the paper gives you some valuable information on online gaming. We appreciate you taking the time to read our combined work.

CREDITS

Editors	**Daniel James** CEO, Three Rings Design, Inc. daniel@threerings.net **Gordon Walton** Vice President, Sony Online Entertainment gwalton@soe.sony.com
Market Survey	**Nova Barlow** Themis Group **Daniel James** **Elonka Dunin** Simutronics Corporation
The Business of Persistent Worlds	**Nova Barlow** **Edward Castronova** Assistant Professor, Cal State Hayward **Gordon Walton** **Elonka Dunin** **Daniel James**
Persistent World Design: An Introduction	**David Kennerly** Design Consultant **Gordon Walton**
Technology	**George Dolbier** IBM **Daniel James**
Production	**Justin Quimby** Producer, Turbine Games
Customer Service and Retention	**Gordon Walton** **Daniel James**

IGDA Online Games SIG Presents
2004 Persistent Worlds White Paper

I. INTRODUCTION

The IGDA Online Games SIG Persistent Worlds White Paper is intended to provide a summary of contemporary issues in the development and operation of massively multi-player online persistent worlds.

We have chosen to use the title Persistent State World or PSW to describe these online experiences. Also used are the terms: virtual world, multi-user domain or multi-user dungeon (MUD), massively multi-player (MMP) game, massively multi-player online game (MMOG) and massively multi-player online role-playing game (MMORPG). The sheer variety of acronyms and the inadequacy of all of them are indicative of the youth of this industry. In particular the authors point out that the growth of the persistent worlds market is almost predicated upon its moving away from its origins in fantasy role-playing type games.

Our intention is to provide a broad survey of the market, primarily in the US with some examples from overseas, an overview of design and production, a guide to technology practices and an introduction to the rigors of service and operations.

The focus of this paper is to provide developers, operators and publishers of persistent worlds with information to help them decide whether or not to proceed with development; to identify pitfalls along the way; to deliver; and then to support these games successfully. The field of PSW development and operation is enormously complex and we cannot hope to be exhaustive or complete in our overview, but we hope that developers new to the field, as well as old hands, will find some useful information within.

This paper has been written by a small group of volunteers. As such the views reflected within are neither those of the IGDA nor those of the employers of the authors. This information has been compiled from public sources and "tribal knowledge" of existing game developers, and is being made available as a service to the community. Its accuracy cannot be guaranteed and no warranty is given!

Reproduction of this document in whole or in part may be done without written approval providing that this document is referenced as source and the URL: http://www.igda.org/online/ is included.

II. MARKET SURVEY

The Market for Persistent World games is nascent, and the authors do not believe that the current picture is representative of its future. At present a small number of predominantly fantasy games dominate the US market, which has made limited attempts to branch out beyond the ‘core gamer’ audience to reach the ‘casual gamer’. Korea presents a picture of a more mature and competitive market, with a greater pool of active players and many more games in operation.

This section briefly considers a few of the games launched, provides estimates of present and historical subscription numbers, and lists some games waiting in the wings. It seems increasingly likely that there are too many games in development for the present market. The section concludes with a few analyst forecasts for the market’s future growth, and a very brief overview of the international opportunities.

A. Launched Games Overview

A growing number of PSW games have launched, too many to cover in detail. Here is a brief summary of some key titles, along with estimates of their subscription levels and success. Please note that **subscriber numbers are best guesses by the authors only and are almost certainly not accurate.**

1. Major US Titles

Title	Publisher / Developer	Launch	Estimated Subscribers
EverQuest	Sony	4/1999	430k
Ultima Online	Electronic Arts	12/1997	160k
Dark Age of Camelot	Mythic	10/2001	200k
Star Wars Galaxies	Sony / LucasArts	6/2003	220k
Final Fantasy XI	Square-Enix	3/2004 in North America	500k worldwide
City of Heroes	NCsoft / Cryptic	4/2004	150k
Asheron’s Call	Microsoft / Turbine	10/1999	75k
The Sims Online	Electronic Arts	12/2002	45k

ToonTown	Disney	6/2003	30k?
Horizons	Artifact / Atari	12/2003	40k
Anarchy Online	Funcom	6/2001	30k
Asheron's Call 2	Turbine / Microsoft	2003	10k?
There	There.com	2003	10k?
Second Life	Linden Labs	2003	15k

- **South-East Asian Titles**

Subscription numbers for Korea and China are difficult to compare with US titles as much of play is from PC Game Rooms that charge a per-hour fee and pay operators anything from a per-seat fee to per-hour revenue share. In addition many of the popular casual titles use a micro-payment or 'pay for avatar' model. However, the revenues are vast and the audience sizes illuminate the mass-market penetration.

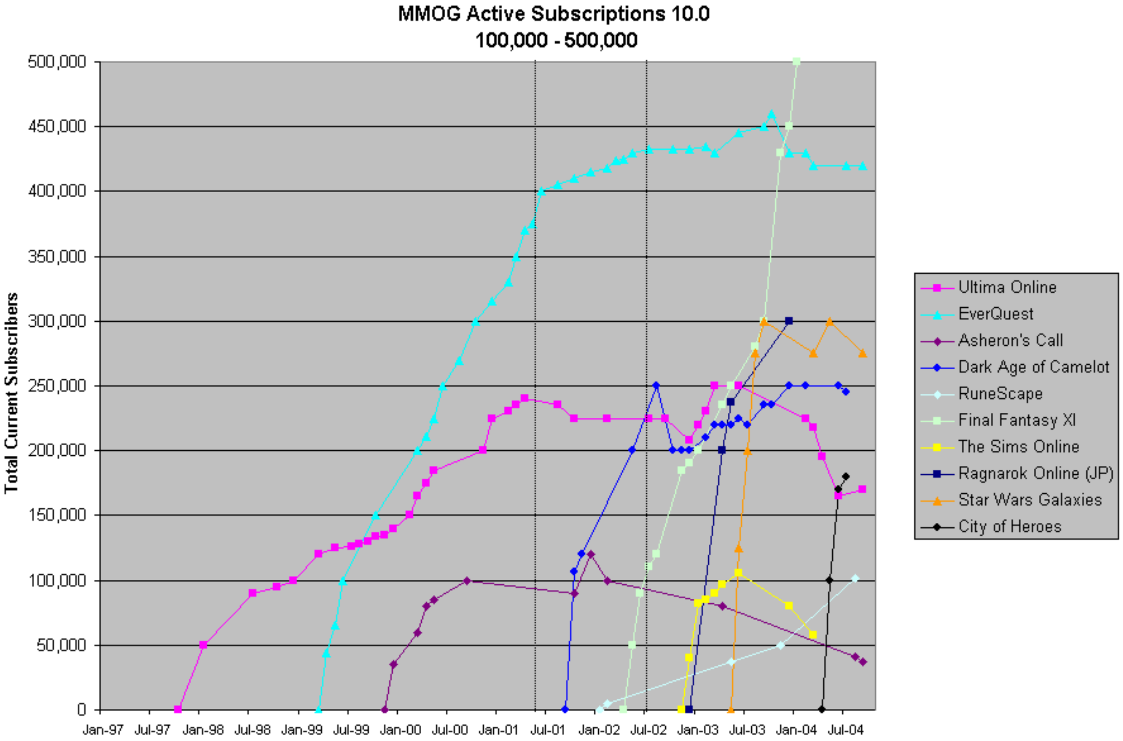
Game	Developer	Launched	Players	Notes
Lineage	NCsoft	1998	Reported 2m+	Korean market leader
The Kingdom of the Winds	Nexon	1996	500k?+	First to market, incredible longevity
Legend of Mir II	Actozsoft	2002?	1m?+	Very successful in China
MU	Webzen	2000	2m?+	Largest game in Asia
Crazy Arcade	Nexon	2002	1m+	Casual market, micropayments
Ragnarok Online	Gravity	2000	500k?	First successful Korean game in US

B. Subscriber Numbers

Data on subscriber numbers for active PSW games varies by company; many of the larger games have openly stated subscriber numbers in milestone press releases, whereas others have kept them under wraps.

Bruce Sterling Woodcock has done excellent work and the community a great service by gathering public and anonymously-reported data into his *Analysis of MMOG Subscription Growth* found here:

http://pw1.netcom.com/~sirbruce/Subscriptions.html



Source: Bruce Sterling Woodcock, An Analysis of MMOG Subscription Growth – V.10.0, 9/2004

C. Recent Launches and Games in Development

The Persistent Worlds space is becoming increasingly competitive. Considering the complexity and expense of development, and the difficulties that many launched games have had in the market, there are a significant number of new entrants waiting in the wings. It is the opinion of the authors that too many of these games are competing for the same audience of ‘core gamer’ subscribers. Here we have listed a few key titles.

Major Titles in Development or Recently Launched

Title	Genre	Developer / Publisher	Stage
The Saga of Ryzom	Fantasy	Nevrax	Launched Q3 2004
Lineage 2	Fantasy	NCsoft	Launched Q3 2004

Everquest 2	Fantasy	Sony Online Entertainment	Launched Q4 2004
World of Warcraft	Fantasy	Blizzard Entertainment	Launched Q4 2004
Guild Wars	Fantasy	NCsoft / ArenaNet	Beta Testing
Middle-earth Online	Fantasy	Turbine Entertainment Software	Development
The Matrix Online	Sci-Fi	UbiSoft	Development
Ultima X: Odyssey	Fantasy	Electronic Arts	Development, possibly cancelled
Pirates of the Burning Sea	Historical	Flying Lab Software	Development
Wish	Fantasy	Mutable Realms, Inc.	Early Beta
Dungeons and Dragons Online	Fantasy	Turbine Entertainment Software	Development

D. Cancelled and Closed Games

It is worthwhile to note that many PSW Games are cancelled in development, and a few have been closed after launch. The fate of *Warhammer* and *Dragon Empires* can, in the opinion of the authors, be attributed to the increased competition in the fantasy genre. Other titles, notably *Earth and Beyond* and *Motor City Online*, failed to reach subscription numbers deemed necessary to sustain operations, and were shut down by their parent company.

Warhammer Online	Fantasy	Climax	Cancelled, may be restarted
Dragon Empires	Fantasy	Codemasters	Cancelled
True Fantasy Live Online	Fantasy	Microsoft Game Studios	Cancelled
Earth and Beyond	Science Fiction	Electronic Arts	Closed
Majestic	Mystery	Electronic Arts	Closed
Motor City Online	Racing	Electronic Arts	Closed

E. Market Forecasts

Where will the industry be, 10 years from now? Even in the best of situations, forecasting can be as much of an art as a science. And in the tech sector especially, there is a scarcity of good forecasting, not least because this is a nascent industry for which the growth curve is still highly uncertain. Technological advances have been rapid, and early forecasting from several years ago was often wildly inaccurate when compared to the actual state of the industry today.

As a further hurdle to the independent game developer, many forecasting reports cost thousands of dollars, making it difficult for a small company to access some of the information. And even for the larger companies, it's often the case that the reports aren't that useful to them, since they're the companies that provided the information to the forecasters in the first place!

1. Forecasting Companies and Info Sources

There is no formal list of "the best forecasting sites and companies" for the online gaming industry. But in any informal discussion, there are certain companies that are accepted as being "pretty good at providing information and forecasts" for the online gaming industry. The top names that tend to come up are:

- Jupiter Research Corporation
- NPD
- DFC Intelligence

There is also a great deal of other useful information that can be obtained via the web. A recent scan for "online gaming market forecasts" produced both the above names, and many other entities such as the following:

- Datamonitor
- Electronic Gaming Business: www.electronicgamingbusiness.com
- Entertainment Software Association: http://www.theesa.com/pressroom.html
- Global Industry Analysts
- IDC
- Informa Media
- Intermarket Group
- Juniper Research Limited
- Mintel International Group Ltd.
- Pyramid Research
- SMI Publishing
- Themis Group
- www.marketresearch.com (search on "online gaming")

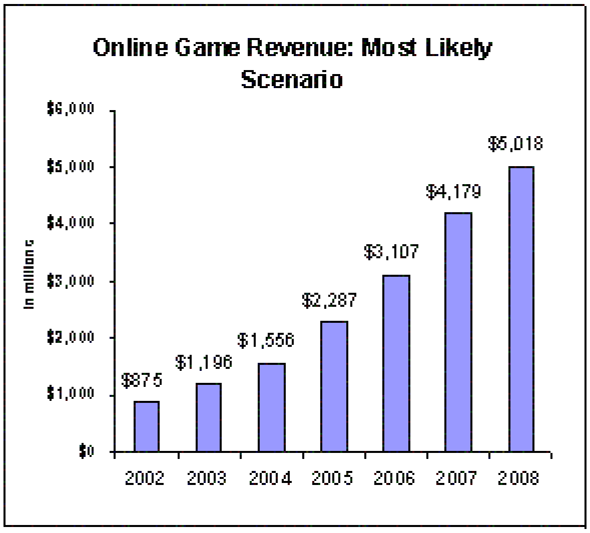
Providing details from the many forecasts available is outside of the scope of this White Paper, but a few excerpts are offered from some of the reports below:

2. DFC Intelligence

In June 2003 DFC Intelligence published a report, ‘The Online Games Market 2003’. On the subject of predicted market size, they said in their summary article:

http://www.dfcint.com/game_article/june03article.html

DFC Intelligence feels fairly confident that online game usage will grow significantly. Where we feel less confident is forecasting how companies will generate revenue from that growing usage. The report contains 10 scenarios for market growth, each based on different revenue generating schemes employed by the industry. Under what we deem the "Most Likely" scenario, online game revenue grows to $5 billion by 2008 (see chart). Under this scenario, we think the bulk of revenue will come from consumer subscriptions and other "pay-to-play" services. While we present scenarios for the growth of advertising revenue, we think they are rather unlikely to occur.



Source: *The Online Game Market 2003*, DFC Intelligence

3. Themis Group

The Themis Group is a consultancy that works exclusively in the area of persistent world gaming, with an emphasis on outsourced customer support, community management and marketing functions. In 2004 they released a substantial report on the future of online games, making projections to 2014 based on a Delphi survey. The report can be found here:

http://www.themis-group.com/uploads/Themis%20Report%202004%20%20Preview.pdf

The report predicts that in 2014, there will be four major revenue streams for MMG operators:

- Membership fees for access to the virtual world will remain the leading revenue generator.
- A second revenue stream will be direct fees for services or premium experiences provided to MMG members, such as tours, "guaranteed adventures", opportunities to play special characters, events, etc.
- The third revenue stream will be direct sales or leasing of virtual property in the MMG. Only about 10% of total virtual property purchases will be

direct from the MMG operator; the remainder will be third party or consumer to consumer.

- These C2C transactions will still generate money for operators, through transaction fees, generally set at 5% (the higher transaction fee will be justified by the security and accountability involved in having the operator govern the exchange), constituting the fourth revenue stream.

Also, the Themis Group Report 2004 projects worldwide online game revenues will reach US$1B+ in 2004.

> *"“It’s clear that the online gaming portion of the industry continues to experience surging growth, especially in the area of subscription-based games. In calendar 2004 alone, we expect to see over $1.1 billion in total subscription revenues for persistent worlds, such as EverQuest, Star Wars Galaxies and Final Fantasy Online.” -* Alexander Macris, Themis Group
>
> *"Growth beyond 2004 will likely be driven by a combination of an increasing base of subscribers, entry into emerging markets such as China, and new business models such as virtual property sales and online rentals. Virtual property sales are already going to generate over $20m in revenue in 2004, or around $400m in trade volume and could grow to as much as one-quarter of subscription revenues by 2008."* - Nova Barlow, Themis Group

4. IDC

In their 2003 forecast, IDC estimated revenues exceeding $1.5B for online games in 2007. However, they painted a dismal picture of prospects for the PSW space, noting that it is increasingly overcrowded with similar products.

IDC highlighted that casual-oriented PSW games such as ToonTown and There.com have unknown potential to expand the market.

Source: U.S. Online PC Gaming Forecast and Analysis, 2003-2007
http://www.idc.com/getdoc.jsp?containerId=30098

F. International Business Opportunities

Editors note: We have a cursory survey of the International Opportunities presented in this White Paper. We hope to be able to include a more thorough discussion in the 2005 update, and would like to encourage you to help us achieve this goal. If you would like to volunteer to assist with future papers, please contact the IGDA Online Games SIG: http://www.igda.org/online.

1. Korea and China

A few years back, Korea eclipsed North America as the largest online market worldwide, and now China has surpassed Korea, at least in number of players that are paying. It is uncertain if China’s revenue has exceeded that of Korea, given the much lower prices charged to consumers there, though this seems certain to eventually happen given their potential and rate of customer growth.

Korea’s market was valued at 587bn won (~$500MM) in 2003. No accurate figures are available for China, but it has been memorably described as ‘the zero billion-dollar market’ in a sniping reference to the difficulty of extracting royalties.

Many US PSW developers are seeking to place their games in Asia, but so far there have only been two examples of a game successfully migrating across the Pacific: *Ultima Online* achieved 100k subscribers in Japan and *Final Fantasy XI Online* had a very successful launch in North America after achieving significant success in Japan. Note that both of these cases are games that moved from Japan to North America; of the many examples we could not find any North American games doing nearly as well in Korea or China. The best performer seems to be the Korean title *Ragnarok Online*, which has done fairly well in the U.S. and Japanese markets .

Some useful information on the Korean Online Game market can be found here:

http://www.ica.or.kr/lib/ITKOREA_Eng(0403)/051Special.pdf

2. Europe

Europe traditionally has had a relatively weak online game market, with most games seeing only a very small fraction of players relative to North American numbers, regardless if the game was sourced in Europe or North America. The one example of significant success is *Dark Age of Camelot*, which has had over 60,000 European subscribers. Europe potentially could be a much larger market for these games given the availability of connectivity and disposable income, but we have not yet seen the breakthrough product needed to do so.

3. Rest of the World

The rest of the world has very minimal opportunities in paid online gaming. The largest nascent market is India, which has not yet seen the rise of game rooms as were seen in Korea, but this is expected to change in the near future. Australia, Thailand, Russia and other countries have small but active gaming populations.

III. THE BUSINESS OF PERSISTENT WORLDS

Building a business out of Persistent Worlds is challenging, but can be highly profitable when successful. A number of companies have made large investments into the space with varying degrees of success. Curiously, there has been little or no correlation between the level of success and the size of the investment. For example, Electronic Arts in particular has been very successful with its first and lowest-budget title, *Ultima Online*, only to subsequently lose money on later, more expensive undertakings such as *Majestic* and *The Sims Online*.

Subscription revenues make for an attractive business case and are necessary to provide a return on the high costs of development and ongoing operations commonly associated with PSW games. There are, unfortunately, limited funding options available to developers. This section concludes with some discussion of the revenue models commonly used in PSW games and the issues surrounding virtual goods sales.

A. The Business Case

Given the risks involved, it may seem surprising that so many people want to participate. The reason is because when the PSW business model works, the result is a highly profitable service business, which can last for 5+ years. Consider the revenue:

250,000 customers x $12.99 / month = $3,247,500 gross revenue / month

This is a number that makes almost anyone in the computer gaming business sit up and take notice. Particularly when multiplied out by 12 months to get yearly gross revenue of almost $39,000,000. Further, this does not even include the retail sales and expansion pack revenue yet!

Typically, to get 250,000 subscribers, a developer would need to sell about twice that in retail units. 500,000 units x $25 wholesale price garners another $12,000,000 in gross revenue. As an example, both *Ultima Online* and *EverQuest* went over the $100 million revenue mark years ago and are still going strong.

Gross revenue is desirable, but what really matters to most companies – and certainly to smaller ones that are the audience for this paper – is the net profit after all expenses. There are two classes of expenses related to PSW games: variable expenses related to the number of subscribers and/or concurrent players, and fixed expenses that stay the same as subscriber counts change. Fixed expenses include the initial and ongoing development team, community management, the equipment the game runs on, and networking staff. Variable expenses may include licensing fees, bandwidth, customer service, and possibly ongoing marketing.

There are often “step functions” in either type expenses, for example a floor full of people to stay in business. A good example of a fixed expense that changes based on success would be when a developer chooses to expand or contract the live development team based on how well the game is performing in the market. For a variable expense, the amount of bandwidth needed is variable and directly related to the number of concurrent players.

The good news for game producers is that if a game has over 100,000 subscribers it is not hard to realize 25+% net profit on subscriptions and if it is over 200k, 40% is possible. Conversely, a game of 50,000 subscribers may only have a 10% net profit. Scale is clearly a significant issue in profitability, mainly because of the massive fixed expenses required to build, launch and run this kind of business. The reason so many companies are creating PSW games is that there is significant revenue and profit for successful titles. The risk is that failing to capture significant numbers of customers can become very costly.

Small can be Beautiful, or at least Cheap

All of the above applies to the large-scale massively multiplayer games, not to the very small games run by teams of 2-10 people for small audiences (less than 20,000 subscribers). These games can be profitable at lower subscriber levels because their costs are very different. For example, almost all of these games are distributed online rather than via retail, they have a more limited set of content, and they provide a more personalized service to their smaller audiences.

B. Costs and Development Funding

Budgets for Persistent World games range from $2MM up to $30MM and higher, typically 2-5x that of a non-persistent (or “stand-alone”) PC game. The primary reasons for the extra costs are complexity and the need for ongoing content as well as special online requirements particular to PSW games, which include:

> Having (or connecting to) a registration, authentication and billing system
>
> A client/server networking system
>
> Servers to run the game on, which are separate from the client software
>
> A database for these servers
>
> Security coding to prevent hacking and cheating

Teams that previously developed stand-alone games typically do not have all of the skills to accomplish these tasks, and adding people to do them is part of the challenge and expense.

Complexity by itself will increase development costs. Even though the overall development size of a massively multiplayer game is only approximately 3x the size of a stand-alone game, it is often 10x more complex. While much of this complexity is due to the increased number of systems being built, some of it is due to the online environment itself. Specifically it is nearly impossible to test all the possibilities in a free form online environment, leading to large numbers of unanticipated errors. Contrast this with the highly linear progression of the typical single-player game or the more highly constrained rule and goal systems of stand-alone multiplayer modes/titles.

Finally there is the cost of content. A typical stand-alone game today delivers 12-20 hours of non-repetitive gameplay. A stand-alone RPG title might deliver 40-80 hours. An online game needs to keep players entertained for months to be successful. This translates to 500+ hours of content needed to retain customers.

Developers have funded persistent worlds as follows, in order of likelihood. There is no doubt that the funding options for PSW games are limited and this fact represents a huge barrier to a new developer.

1. Publisher Funding

By far the most common method for funding PSW games is through publisher funding. Indeed, most MMORPGs have been not only funded by publishers but also developed by their in-house studios.

A small number of titles have been developed externally with Publisher funding, notably *City of Heroes*, developed by Cryptic Studios and published by NCsoft, and the *Asheron's Call* games developed by Turbine and published by Microsoft. Turbine has since bought back both AC games using venture capital, but is now developing high-profile titles, *D&D Online* and Tolkien's *Middle-earth Online* in partnership with publishers.

An entire volume could be written on the vagaries of developer-publisher relationships, and with PSW games another layer of complexity is added. Perhaps the most significant issue – and one which the developer can have some influence in determining – is who will be responsible for support and operations? Because of the viral nature of PSW games, customer retention (which is driven by the quality of the online experience) is arguably the single most important success factor – even more important than budget and tactics applied to marketing and new customer acquisition. Many developers have regretted deals in the past that gave server operations and support responsibility to their publisher. They felt that this division of labor led to patchy service and slow updates/fixes to technical issues. Therefore, we encourage developers pursuing publisher-funded PSW games to own responsibility for at least the technical aspects of service operations. Retaining control of these areas also justifies a much higher slice of the revenue pie -- It gives the developer a much stronger strategic position for raising future financing, and better positioning for a rewarding liquidity event.

2. Venture Capital

A few developers have had success in raising substantial venture capital investments, notably: Linden Labs ($8MM Oct 2004), Turbine ($18MM Dec 2003, at least $10MM in earlier rounds), Mythic ($32MM 2003, at least $3MM in earlier rounds), There.com (over $30MM in various rounds) and Artifact Entertainment (at least $10MM in various rounds, including $5MM from NCsoft). Raising VC is extremely difficult, and most of the above developers had established their skills and credibility by working with publishers before raising capital; for example, Mythic produced over 5 online games for various publishers before raising ~$3MM to produce *Dark Age of Camelot*.

VCs are generally more interested in technology-oriented companies than media companies because they are historically more likely to have the potential to generate annual revenue of $50MM+ and lead to a public stock offering or an acquisition with 5-10X+ return on investment. VCs are generally not interested in building non-scalable lifestyle companies, even those with sustainable $1MM+ monthly revenue. Despite the historical reluctance to invest in games though, the venture capital industry is increasingly interested in the space, so for those with the necessary team, contacts and plan, it may be an option. Firms that have made recent investments in gaming include: Trinity Ventures, Mayfield, Kleiner Perkins, Sequoia Capital, NEA, and Draper Fisher Jurvetson.

3. Angel investors, or Friends, Family and Founders

A more common approach for smaller teams is to invest personal money and solicit investment from personal friends and acquaintances. This is particularly viable for smaller games with budgets of ~$1MM or less; *Puzzle Pirates* from Three Rings, *Second Life* from Linden Labs, and *A Tale in the Desert* from eGenesis were all funded this way. It is, however, incredibly risky. Other games funded in this manner (*Fighting Legends, Maximum Charisma*) have failed, leaving their investors out to dry. That said, if costs are strictly controlled and can scale within the bounds of cash flow, this is an attractive option when available.

4. Ramen and moonlight (or the starving artist option)

It is possible to develop games in the evenings alongside a full-time job, or for students to develop excellent games. Unfortunately the complexity of PSW games and their support requirements make this option extremely difficult. Developers with this inclination are encouraged to pursue very simple projects that are highly achievable with a quick result, for example a text MUD or simple multi-player game before moving on to any sort of major PSW undertaking. Also, developers with day jobs should carefully review their employment agreements and seek legal counsel to avoid creating assets that are, at the end of the day, the property of their employer.

C. Distribution and Marketing

Distribution and marketing has traditionally been the province of publishers rather than developers, but with online games there are more elaborate possibilities than simple retail release. More importantly, there exists an opportunity for developers to form lasting direct relationships with their players.

1. Digital Client Distribution vs. Retail

For many years now the prophets of digital distribution have proclaimed the end of the retail release and the dawn of a new era of direct-to-consumer distribution over the Internet. Alas, this era has yet to truly arrive for PSW games. The reasons are as follows:

- Publishers are still reliant on the income from boxed retail products.

- Broadband is prevalent amongst hardcore gamers but among the general public, it is still not widespread enough. And even for existing users, it is still not fast enough to support wide release of large (>500mb) game clients.
- Retail releases have credibility in the eyes of press and consumers that downloadable games do not.
- Retail releases are accompanied by a much larger degree of advertising and promotions. As long as online-only releases are considered second-class in terms of marketing dollars they will not be able to compete with a retail box release.
- Customers who have invested in a retail product are substantially, perhaps an order of magnitude, more likely to subscribe than those who have downloaded a free trial. This is perhaps the most compelling (though annoying) reason for retail's continued dominance. Despite often-heard complaints about ‘paying for the box and again for a subscription’, it would appear that consumers prefer to do this than to try a PSW game for free via download and then subscribe if they like it. This is, of course, related to the marketing budget issue.
- Retail provides a valuable marketing channel; people browse in stores for games.
- There is a lack of viable marketing and distribution channels for downloadable clients. For most games the casual gaming sites are not suitable, although sites such as Gamespot DLX, IGN Direct2Drive, Gigex.com and Yahoo! Games on Demand are beginning to fill this need.

This is not to say that download is not a viable marketing and distribution channel; it is. Developers would be wise, however, to consider reaching the greatest audience possible, and retail is still the most effective way to ensure that.

2. Game Portals and the lack of them

The rise of the downloadable games business in the casual gaming segment has been driven by the game portals; sites such as Yahoo! Games, Shockwave.com, pogo.com, Real Arcade and Microsoft’s Gaming Zone. The ‘core’ PSW market is much more fragmented, with a multitude of destination sites with much more narrow focus, such as the various ‘fan sites’ grouped under the Warcry Network, IGN’s Vault, Stratics, and so forth.

Whilst publishers can reach their audiences through advertising on these sites, most visitors are not visiting to discover new games, but rather to gather information about their existing obsession.

3. Cover-disks, OEM, Cheap CDs and Bundling

Few publishers have pushed hard on alternative distribution of PSW clients, although this would seem to make sense from a business perspective. Those that have tried it report unimpressive results. *Lineage* led the field with its North American release on cover-disks and with an impulse-purchase $2.99 CD on retail store checkouts, but results were disappointing. So far the jury is out, but expect more publishers to attempt wide distribution through these channels in the next year or two.

4. The efficacy of a large, long Beta

In the 'second wave' of MMORPG releases in the U.S., a number of publishers offered extensive open public betas. It is often said that this did them no favors, *Asheron's Call 2* and *Earth and Beyond* being two examples of games with long beta cycles that were then poorly picked up at launch and beyond. Anecdotally there is evidence that players either played the game to their satisfaction during beta, or found reasons to become frustrated and never try the full game. It is believed that both of these examples had well over 100,000 beta players, but fell far short of these levels of paying subscribers when they went live.

There is also a growing group of PSW players who 'beta surf', simply moving from one new beta to another. The continual stream of new releases ensures that they can easily find a new game and never have to purchase a subscription. The authors caution developers to consider carefully to what extent they wish to support a large open beta, and note that recent publishers have been more cautious, only admitting large numbers of players towards the very end of the beta cycle.

D. Revenue Models

1. Revenue through Distribution: Retail Sales

The majority of large PSW games have shipped via the traditional computer games retail channel. As discussed above, despite the prevalence of downloads for beta testing and promotions, retail remains the preferred method of launching the game. Revenues from retail sales of the client software at full-price might generate $25 per unit for the publisher, recouping a substantial part of development expenses.

a) Collector's Editions

One way companies increase revenues is through releasing Collector's Editions. Collector's Editions are packages that tend to include pins, collector's maps, and other knick-knacks as well as the latest expansion release for a game. By providing an up-to-date client, these packages serve as an "all in one" enticement for new players, and an easy way to re-install for old/returning players. *Star Wars Galaxies* perfected the concept of the Collector's Editions, making one abundantly available while restricting the concurrent release of the regular retail client; this tactic drove the hardcore early adopter to pay $20 extra to get into the game.

2. Primary Revenue Model: Subscriptions

At present, and for the foreseeable future, the primary revenue source for persistent online games, at least in the U.S., is the monthly subscription.

Subscription pricing has increased steadily from the early days of the web. Simutronics' play.net site was one of the early subscription-based models, charging $9.95/month in 1997. Ultima Online, launched in 1998, followed suit, also charging $9.95. Today, prices tend to range between $10 & $15 per month, with $14.95 seeming to be the most popular price point. It seems that the gaming audience, once they are already signed up to a game, does not appear to be particularly price sensitive. Indeed, it has proven possible to raise prices on customers without suffering substantial attrition; in 2001, Funcom and Mythic launched their titles at $12.95 per month, and in 2002, Sony Online Entertainment raised the monthly price of *EverQuest* from $9.89 to $12.95, with virtually no increase in customer churn as a result. *Ultima Online* and Funcom's *Anarchy Online* raised rates as well.

Upon a price increase, some customers, to be sure, will quit and say that the price change was what prompted their departure, but the numbers do not bear this out. There are always new customers joining a game each month, and old customers choosing to cancel for a wide variety of reasons. Typically, there is a 3-8% "churn" each month in most PSWs; that is 3-8% of users will cancel each month, regardless of what else is going on in the game. During price increases, this percentage seems to hold constant -- the only difference is in the exit interviews, where customers state that the price change was the reason for their departure (as opposed to listing some other reason). But there still tends to be no higher percentage than normal of cancellations.

It should be noted though that raising prices in this manner will only go so far, and may exacerbate the problem of concentrating the audience for PSW games in the 'core' gaming market, by excluding more price-sensitive casual gamers, or alienating potential new customers.

a) Game Aggregation

An elusive but attractive prospect for publishers of multiple games is to offer a combined subscription in order to keep players loyal to their primary offerings, as well as entice the player to try out additional offerings from the same company.

Sony Online Entertainment (SOE) has introduced a "Station Pass", allowing customers to access all games published by SOE for $21.99. While this option covers the *EverQuest* franchise and *PlanetSide*, SOE's other popular offering *Star Wars Galaxies* is not included. Information on its success is not public, but Electronic Arts terminated a similar offering with *Ultima Online* and *Motor City Online*. Skotos similarly offers an array of graphical and text games for a single subscription.

b) Multiple accounts

There is no doubt that a large number of PSW players maintain accounts across multiple games as well as, in many cases, multiple accounts on the same game. Little empirical data exists and companies are not prone to disclosing the information, but anecdotal tales are of players with three or more largely unused accounts on games such as *Ultima Online*, paid up purely to maintain ownership of characters, items and other in-game property. However, it is the author's opinion that such players are extreme examples of relatively wealthy and/or super-hard-core consumers and do not warrant much consideration until a game achieves scale.

3. Revenue Generation from Services

In the search for additional revenue from their affluent core gamer customers, PSW operators have increasingly begun to explore premium service offerings.

a) Selling Experiences and Special Events

Simutronics games (www.play.net) pioneered the concept of making money from virtual experiences/events. Tickets averaging $20-70 can be purchased ahead of time for various events or quests. Players can also purchase wedding packages costing $200 or more for elaborate in-game weddings with staff-created special effects. In just one of the Simutronics games, *GemStone,* reportedly over 600 weddings have taken place, with a recent calendar scan revealing between two to six weddings per month. Sony's *Ultima Online* attempted something similar with the "Magic Moments" program, but it was later cancelled, reportedly in part due to the complexity of operations.

b) Pre-made Characters or Power-leveling

Within an Advanced Character Service offered by Ultima Online, players may purchase a ‘pre-built' character with abilities and skills usually earned over a long period of play. Doubtless inspired by the robust eBay market for established characters, it is still unknown how much revenue has been generated by this effort and whether it warrants the negative publicity associated. See: http://support.uo.com/advancedcharacter.html

c) Character Transfers

With the rise of special servers, as well as the ongoing problem of server overcrowding, companies have started to offer character transfer services. Character transfers range from $15 to $50 for 1 character up to a range of $49.99-$125 for a full set of characters, and some services make it possible to transfer these characters between accounts, as well as servers, for additional fees. While only a few games have adopted this idea, large games such as *Ultima Online* are currently discussing the idea with their player base for feedback before implementing. In this author's opinion, while developers of new games might consider providing character transfer as a feature, they should not look to it as a major source of revenue until their game hits large scale.

d) Account Transfers

One activity that customers take part in is "selling" a game account from one player to another, much as offline gamers might sell collectable cards or comic books. The legality of this type of transfer is ambiguous (see "Virtual Goods Sales" below). Recognizing the popularity of this practice though, *Ultima Online* developed a “safe account transfer” service by warning players who do not use the service, “if you choose not to use the official Account Access Transfer service created by Origin, and a dispute over the use of the account ever comes up, we may not be able to determine who rightfully has access to the account and, as a result, the account could be terminated.” See: http://www.uo.com/acct_xfer.html

e) Character Renaming

In conjunction with the account transfer service, *Ultima Online* adopted the “character renaming” service, which, when combined with the above service, allows a non-original account holder to mask the origins of an account. This service has already shown up in newer games after a period of time, such as *Earth and Beyond* and *The Sims Online*. *EverQuest* also has had a character renaming service for several years.

E. Virtual Goods Sales

The most popular Persistent World Games have seen the rise of robust markets exchanging in-game currency, goods and characters for real money. This phenomenon has been concentrated on eBay and a few specialty sites.

There is considerable controversy surrounding these markets, and reactions from publishers and customers have varied.

- A vocal part of the core gaming community resents the very idea of these markets, where in-game achievements (often directly related to time invested by the player) can be traded for real money.
- The purchasers of such virtual goods clearly value the service highly enough to part with their money. Considering the incredible time investment required by most persistent world games, players with more available money than time are, in many ways, well advised to make use of these services if they wish to compete with other players who have more free time to play and build up their characters. The demand is often exacerbated by design, in particular the necessity that a group of players’ characters are of similar ‘level’ in order to usefully collaborate, and the inability in most games to switch character type while retaining invested time.
- Large-scale PSW games have only scratched the surface of the business opportunities in this area, for fear of alienating their core clientele. *Ultima Online*’s aforementioned pre-made character-sales option was limited and yet still caused an outcry.
- *Magic Online*, the online implementation of *Magic: the Gathering*, has

proven that in the right context players will spend substantially on such services.

- The text-MUD *Achaea* and its siblings have demonstrated that the sale of in-game content by the operator is a viable business model on a small scale, and titles such as *Quiz-Quiz* and *Crazy Arcade* in Korea, and Habbo Hotel in Europe, have shown that it can work well for a mass-market audience. There.com has a similar model in the U.S., and has reportedly achieved revenues of over $100 per customer per month. All these operators report substantially higher average revenue per customer from such sales than the usual subscription prices.
- Most U.S. PSW operators have explicitly outlawed the resale of game accounts and property via auction services, yet with the notable exception of Mythic Entertainment, do not tend to enforce these Terms of Service provisions unless called upon to resolve customer service requests caused by such sales. Only *Ultima Online* has actively condoned the secondary market in game content.
- It is also worth noting that along with the out-of-game transactions, comes a fair amount of fraud. One game developer estimates that 5-8% of all out-of-game transactions each month involve some sort of fraudulent activity, ranging from a player selling items that do not perform as advertised, to the tried-and-true "rubber check" syndrome -- paying someone with a check that does not have sufficient funds behind it. Game developers can expect to receive complaints from customers to the effect of, "So and so bought my sword but his check wasn't any good -- do something!" It is the editors' recommendation that game developers have clear policies in place ahead of time to deal with this situation, even if it's just, "We are not responsible for trades that occur outside of our control."

Here we present the views of some other individuals who have studied virtual goods sales.

a) Edward Castronova, Economist

From the earliest days of online game interaction, commodities and accounts from games have been sold between players on dollar-based markets outside the game. These markets emerge under the following conditions:

a. There is something scarce in the world, in the economic sense: the players would like to have more of it than there is.

b. Access to the scarce thing(s) can't be secured in-game using US dollars.

c. The scarce thing(s) can be traded in-game between players.

Note how widely these conditions apply to online games. As for item (a), for the game to have any challenge/reward system at all, some things have to be scarce. Some things have to require work or skill or (in-game) money to get. For persistent worlds, content itself has to be made scarce: elder content is supposed to be reserved only for those players who spend the time immersing themselves in the world, as a reward for going through the advancement process. The rags-to-riches storyline is a core element of all character-advancement games; without scarcity, it makes no sense and provides no emotional satisfaction for the player at all. As for item (b), character-advancement games need to avoid selling content within game for $US, for exactly the same gameplay reasons as with item (a): a game where you buy your way to level 50 isn't a character advancement game and can't provide players with a sense of accomplishment and satisfaction. Item (c) refers to player markets, an element that is universally considered a core element of the fun factor in any persistent world.

All three of these features are going to appear as core elements in the design of a typical persistent world. Once written in, however, they hard-code in an incentive for every player to seek advancement using dollar-based markets outside the game. As the scope of online interactive entertainment has widened, opportunities to trade things outside the game have also widened. eBay is the most successful market in human history, with billions of dollars traded every year. 'eBaying' has evolved into an everyday feature of games. The eBay market has become a meta-game; if you can't advance in-world using your resources of time and player skill, you can just advance out-world using your real-world incomes.

The eBay meta-game does strange things to the design of a world that is supposed to give users an immersive role-playing personal-advancement experience. Truly idealistic users will faithfully struggle through challenges to advance. But then one day they encounter other players who have simply bought their way to the top. It is difficult to sustain immersion, role-playing, and the illusion of personal fulfillment from that moment forward. Suddenly, the real game appears to be about spending money wisely on eBay rather than slaying the right foe.

Does it matter? In some designs eBay makes no difference at all. Indeed, it could be a welcome factor. Embrace eBay, and you don't have to code in-world markets at all. It could be made consistent with various kinds of lore; eBay is 'the Spice trade' or 'smugglers from the outer galaxy.' This would be a stretch, of course. But even if we assume that eBaying wrecks immersion for 90 percent of the players - especially now that it has become so widespread - it still might not be so terribly bad for a design. The supply-demand pressures are certainly there; if they are not satisfied through eBay, will they be satisfied at all? If they are not satisfied, subscriptions may fall. Thus it may be the case that eBaying has to be accepted; it ruins lore but enhances user satisfaction so much that it increases the subscription base.

But there are many reasons not to accept eBaying as it is today. When eBaying becomes an accepted part of a design, the project changes from an immersive, fantasy role-playing game into an avatar chat room with certain crafted entertainments, and special content for those who want to spend money on it. The latter is a perfectly acceptable project design, but the point is, it's probably not what the buyer expects. And because of that, the acceptance of eBay probably does more to damage subscriptions than to help them. True, many individual users will report happiness that there is a fluid external market where various game challenges can be overcome. But collectively, the user base may well judge this game to be less than what they hoped it would be: less immersive, less escapist, and less introspective. The social logic is like that involving pollution: every person who tosses a wrapper on the ground is happy to be able to get rid of it; but if we just accept that practice, then everyone suffers from a mountain of garbage on the street.

In economics-ese, eBaying in a fantasy role-playing game is individually incentive-compatible but collectively sub-optimal. It is a market failure: people left to their own intentions and devices will pursue courses of action that make sense to them as individuals but, once everyone does it, result in everyone being worse off.

There are other reasons why eBaying is a problem. The outside market has a powerful effect on what happens in the in-world market. If the external market becomes dominated by a few big marketers, then the world suddenly has acquired an independent Finance Ministry and Central Bank, capable of dictating how the in-world economy will respond to design decisions. As such developers may find themselves negotiating with them about patch notes.

The eBay market also represents an expression of user interests. It only exists because there are things that players want to do but can't do within the world. In a sense, it represents a design flaw. If the design produces conditions a, b, and c above, the design needs to deal with the incentives that those conditions create. eBay is a place where users go to satisfy something; if they can instead satisfy it within the world, the world has become a more fun place. Or, to view this more crassly: with hundreds of millions of dollars being traded for a world's items on eBay, why would developers allow someone else to profit from that revenue stream? Handle that market in-world and the revenues will become the developers, in the form of enhanced subscriptions.

Still another reason: eBay monetizes a world's items and tells Earth's governments, lawyers, businessmen, and economists that what's going on in there is not a game. Someday, perhaps soon, the officials of the outside world may reach the further conclusion that it is really just a tax haven, and respond accordingly. They may also begin to hold developers liable for the dollar value of lost/destroyed/bugged items and characters. This precedent has already been set in Korean law. Needless to say, this would seriously affect the financial status of the game.

On some level, again, none of this is necessarily a problem. If the intent is to design an avatar-mediated communication space that has entertaining things to do, then the only problems revolve around how to capture revenues from the incentives behind the eBay market, and do it without arousing the interest of the tax authorities or the courts. As eBaying becomes more common and more people become familiar with avatar-mediated communication, the revenues from virtual items sales will grow. Trade in these digital goods may well become one of a largecomponent of economic production.

But this document is written for game designers, and most game designers view themselves as designers of, well, games. Not as designers of chat spaces with digital item trade. Chat spaces are not why they got into this business. They got into the business because they wanted to make imaginary worlds, separate from Earth, where the rules of advancement are different from the rules we face every day. eBaying is a potent threat to this vision. It imports the status and resources and practices of the everyday Earth economy into what is supposed to be a different realm. It does not completely destroy a game design, but it does remove its heart: the aspect of alternative reality.

The Holy Grail for the game design community, c. 2004: Build a fantasy role-playing world, with markets and character advancement, where eBaying does not happen - not because eBay is forbidden, but because no player wants to use it.

b) Gordon Walton, Executive Producer

eBay exists and will continue to exist to satisfy consumer needs. Unless developers use artificial (and non-immersive) design techniques such as no-drop items to thwart players exchanging items in-game, people will exchange items. And if they can exchange items that have perceived value, there will be an out-of-game market for them, with or without an eBay to facilitate it. Even if they can't exchange items, they will sell their accounts with the characters and items included to other players. Even if developers sold the items in-game for dollars, all that would do is set the ceiling price for the item in the external market. My advice is to focus their energies on making a really fun and compelling game rather than spending excessive time attempting to stop eBay. After all a so-called "eBay problem" is a high class problem, meaning it only exists because a game is successful.

c) Elonka Dunin, General Manager of Online Community, Simutronics Corp.

One other point-of-view, reflecting why allowing item sales can be a bad thing:

Part of the formula of making money from a game has to do with customer lifecycle. The way this lifecycle works is as follows: Customers are attracted by some means to a game, hooked on it (hopefully because it's fun!) and converted from free trials to paying subscribers. The next stage in the lifecycle is that they play the game for some period of time, before they get tired of it and move on to another activity.

There are indeed some people who get hooked on a game and stay with it for years (*GemStone* has customers who have been with the company since the late 1980s). However, it seems that a more typical play cycle is for most customers to stay with a game from 6 months to 2 years. The longer duration of time that this typical lifecycle can last, the lower the churn rate, and ultimately the better for the company's revenue.

It is my opinion that promoting any mechanism which allows players to get ahold of advanced game items rapidly, whether it be via out-of-game sales or giving away too-powerful items in "Monty Hall" quests, is detrimental to game balance. It may allow some customers to become more engaged with the game more quickly, but it will also cause them to work through the game's challenges more quickly, get bored more quickly, and, eventually, leave more quickly than they probably would have if such boosted items were not easily available. For a persistent-state world to be successful, it must keep an eye on long-term trends and game balance -- not just short-term gratification.

d) Daniel James, CEO, Three Rings

The use of real money to create an advantage in a virtual world is no doubt egregious to players who believe that their gaming experience should provide a 'level playing field'. Imagine a game of Monopoly in which a wealthy player bribed the banker for additional game funds – and consider whether such a player would be a popular gaming partner?

However, moral debate over the impact of the influence of hard currency on a persistent world's economy is of limited utility. I believe that in any popular game with an in-game economy in which such trade is possible, the infiltration of real money is inevitable. If the developer is opposed to such trades, the onus is upon them to engineer the game in such a way that they cannot take place.

City of Heroes has taken substantial strides in this direction by creating a world in which there are effectively no items to be traded, and currency is of limited utility. None the less, 'Influence' can be purchased for *City of Heroes* on ige.com (the leading broker of PSW game currency, item and account sales). To completely harden a game against such trades would require a more robust obfuscation of the trading mechanic, such as that proposed by Randy Farmer in Kidtrade: http://www.fudco.com/habitat/archives/000023.html

It is my personal belief that developers will either take the above routes towards games without a currency-trading mechanic, or that they will embrace virtual goods sales and transactions within their games and seek to concentrate profits within their own walls. Examples of this can be seen with Achaea, Habbo Hotel, *Project Entropia*, There.com, and *Second Life*. In *Entropia's* case the developers support exchange to and from game currency (a risky move!), whereas There.com and *Second Life* leave such exchanges to a third-party peer-to-peer service.

It may be that many successful games will continue to muddle along in the ambiguous middle-ground, so long as the myth of their level playing field can be perpetuated, but I suspect that player concerns and potential profits will, in the end, drive us all in one direction or the other.

IV. PERSISTENT WORLD DESIGN: AN INTRODUCTION

The field of Persistent World design is young, yet a remarkably substantial amount of material exists on the subject. Formed by the crucible of MUD development, PSW design is only now beginning to diversify, and there are still remarkably few completed designs that explore new mechanics and genres beyond the Fantasy ‘level treadmills’ that originated in text online worlds.

In this section we provide only a brief overview of the complexities of PSW design, as an exhaustive overview would require a book or three. We encourage the reader to explore the Further Reading section at the end of the White Paper.

A. Design objectives

PSW games contain a diverse array of features, yet share a basis of common characteristics. The PSW game is generally characterized as having an environment that continues to exist even when no one is playing it (thus the word "persistent”), that supports a large number of simultaneous players (thus the alternative moniker “massively multiplayer” or MMP), and retains records of player data indefinitely (also implied by ”persistent”). Exceptions exist, but for the scope of this brief introduction, this suffices. For these characteristic definitions of a PSW, a survey of game design should agree on the objectives of the design. For brevity and clarity, let us consider the objectives of good PSW game design to include:

- customer renewal
- fair opportunity
- low harassment
- low service call volume
- low resource consumption

The first three may be broadly considered as elements of player satisfaction, and the last two may be considered as elements of what economists call the efficient allocation of scarce resources. These goals are not unique to PSW games, but the collective experience presented in this White Paper suggests that failure to meet one of these objectives encompasses the majority of PSW design failures.

Instead of focusing on the reasoning behind these objectives, this article focuses on a few well-tested means to achieve them. To do so, let us briefly define each.

Customer renewal is offered as a measure of user satisfaction, instead of user self-reports, because in several instances the customer complains loudly yet continues to subscribe; and conversely, may not complain at all yet silently cancel their subscription. Although not all PSW games employ the subscription model, the design principles outlined here also apply to pay-to-play, premium goods, and premium services business models.

Fair opportunity means that each equally competent player has an equal chance to succeed.

Low harassment refers to a low rate of the malicious disruption of enjoyment of the game for any player; whether reported or not, whether by player or operator.

Low intervention is a low rate of aggregated service calls, including e-mail, phone, fax, in-game referee, and discussion board requests for help, technical support, or restitution.

In this article, low resource consumption specifically addresses the service resources that design can influence, such as server bandwidth, processing power, temporary memory, and disk space usage.

1. Case analysis

A fundamental process of all software design includes average and worst case analysis of the algorithms employed. Let us also consider the often omitted third case: the best case. In addition to software performance, let us evaluate a design's ability to achieve the five objectives stated above during the worst case, average case, and best case scenarios.

The center of our analysis is the user population, the players. The importance of the extreme cases, best and worst, is magnified in PSW games. The user population will usually gravitate, with alarming speed, to the optimal solution of a given subsystem. In the worst case, the old adage of Murphy's Law applies with force: Anything that can go wrong will go wrong. Alternatively, when users optimize, anything that can go right will go right. Any shortcuts, intentional or not, are exploited, and then user communication accelerates optimization. Like an ant trail to honey, players will hone in on the optimal path for character advancement. Therefore, the fundamentals of worst case, average case, and best case analysis underlie PSW game design.

2. Fault tolerance

In good design, PSW games should be evaluated for fault tolerance, especially as regards the technical infrastructure. Many user complaints are generated due to network issues: most predominantly latency and disconnection. What causes these errors? First, the PSW network is only as stable as its weakest connection. Often this is the client's connection to their Internet service provider (ISP). Sometimes the server's ISP can fail as well. There are few twitch-based designs among the popular PSW games in the U.S., because latency, and not player reflexes, often determines the victor. A good design, however, may minimize user frustration during this kind of worst case scenario. As well, a design may prevent users with the best network speeds from having an advantage over those users with minimal connection stability.

The concept of fault tolerance stays with us as we consider the other enabling elements of MMP game design.

A good game design functions well during both of the extreme cases of usage: minimum and maximum. Whether servers are flooded with users, or devoid of them, the game experience should continue to meet design objectives for the users present.

Common problems include overcrowding, blockage, scarcity, and disabled features. For example, if a user can block the travel of other users, then in the worst case scenario, that blockage will inevitably occur. Common solutions to blockage include providing wide lanes for traffic, the ability to physically push other users out of the way to allow passage, and referee intervention. Because of the design objective to minimize service calls though, referee intervention should be a last resort. However, the ability to push brings up a complication not usually encountered in non-PSW games: interdependence. For example, in *Toontown*[1], occasionally a user pushed another user into a dangerous area. As another example solution to blockage, *Ultima Online* allows a user to pass through another user.

On the other end of the scale, minimum usage can also create problems. Some features require multiple users to activate them, so during minimum usage, sufficient users may not exist to play, or may not cooperate. A complete game design addresses this. As an example, ships in *Puzzle Pirates* require a crew to operate. If there are insufficient human crewmembers to sail a ship, then, as an alternate to a player crew member, the user may hire computer-controlled crew members.

3. Fairness

Players demand fairness. No player wants to invest time and money in a game to be beaten by a cheat. Now, in the real world, elements of unfairness do occur at every turn, even in games such as sports. Players may not like this, but they are resigned to the fact. However, because PSW games are constructed worlds, appearing to the player much like a divine creation[2], users hold designers up to greater standards of perfection. The exploitation of an unfair advantage, or cheating, is an element of worst case player behavior. An effective design should endeavor to eliminate most opportunities to cheat.

Clever monitoring and analysis of performance statistics can detect possible cheating during live game operation by tracking statistics about the rate of advancement, victory ratio, and other conditions of success in the game. This data also helps a designer balance the game in accordance with the players' ideal of fairness.

Worst case and best case analysis take on different connotations for analysis of fairness. If there is a non-deterministic result, such as a random number analogous to rolling a die, then sooner or later, the worst case of this result

[1] Goslin, Mike, Joe Shochet, and Jesse Schell. "Toontown Online: Building Massively Multiplayer Games for the Masses" in *Massively Multiplayer Game Development*, Ed Thor Alexander. Charles River Media, Inc., 2003. (3-19).

[2] Bartle, Richard. *Designing Virtual Worlds*, New Riders Group, 2003

occurs. In a PSW game, this probability becomes a statistical inevitability for a segment of the player population. If a user can miss an attack swing ten times in a row, sometimes it happens. This miniscule probability, over the lifespan of a massively multiplayer game, becomes inevitable. And the more players, the more often these small "worst case" events occur. Planning ahead for them, and balancing these best and worst cases of the game mechanics, can offer a more satisfying play experience.

4. Replayability

Unlike single player games, PSW games are never beaten. That is, a player should not reach a state in which the player feels there is no further goal to reach. Instead, the content of the game should be reusable. Since players return to the world often, good game design provides a method of entertainment that does not usually rely on novelty. Most games do this by respawning opponents and resetting missions. Some maintain the freshness of the experience by varying the content. As an example, *Anarchy Online* randomly generates details of its missions.

5. Reversible consequences

A fundamental characteristic of any game, not just PSW games, is a safe environment for experimentation. Many PSW games retain both safe experimentation and event permanence by making most actions reversible. Any item lost should be replaceable somehow through normal game mechanics, such as by repeating game actions which generated the item in the first place. However, designers often reserve some actions as irreversible, such as the choice of a character class. For irreversible events, the situation in which the event occurs, and the consequences of the event, should be fair. If a class is chosen for the lifespan of the character, then fair class balance constitutes a design objective.

Another problem related to a lack of rollbacks includes garbage collection. If items are generated into the world (such as via treasure generation), and can be left by players in common areas, then over time, garbage will accumulate. Mechanisms must exist, like janitors, that will sweep up the garbage. Some game designs disable littering.

6. Asset appreciation

Because the user's data persists over playing sessions for an indefinite term, playing a PSW game is sometimes compared to an investment. The items and attributes that define the status of the player constitute the player's assets. Accumulating the value of these assets encourages customer renewal, since a strategic player makes a decision to participate based on the expected return on investment (ROI). By exhaustively collecting data on the game and sharing analysis to develop a kind of collective intelligence, the player population can assess the ROI with a fine degree of accuracy. Events that completely nullify asset value are the most damaging. This can occur by accident, such as during an archive failure, or by design, such as permadeath[3].

Macro-economic balancing of asset appreciation often requires continual maintenance. Balancing during live service also requires short-term asset appreciation. When an item is reduced in efficacy or efficiency by design, known as "nerfing", that asset has been depreciated. In the worst case scenarios, the affected player does not recognize that a nerfed asset is being balanced for the fairness of all players. That player will publicly complain that the nerfed asset was maliciously depreciated to alienate the segment of players relying on the asset.

Many instances of nerfing can be avoided while balancing is performed by employing a Dutch auction algorithm. Simply put, a Dutch auction begins with an asset at a low value, or high cost. Then the value of the asset increments, or the cost decrements, until asset performance statistics cannot detect an imbalance.

7. Asynchronous Interaction

The primary activity on the Internet is e-mail, which is a form of asynchronous communication. PSW games also offer the opportunity for asynchronous interaction. As defined, the game continues while the player is disconnected. Yet, the game may also continue to affect the status of the player while not connected. This can be incorporated as a feature. For example, in *Shattered Galaxies*, offline player territories may change ownership. In *Ultima Online*, offline players may sell goods.

8. Conclusion

This introduction has touched on just a few of the considerations that are critical to designing a PSW game, such as cooperative gameplay, player synergy, competitive play, and tolerance for grief play, which is also called harassment, or worst case player behavior. There are many other considerations that are worth covering, which may be discussed in later White Papers. If you would like to assist with writing an additional section, please contact the IGDA Online Games SIG to volunteer, at http://www.igda.org/online/

.

B. Social, Political and Economic Systems

PSW games contain a special ingredient that single-player and session-based multiplayer games do not have: a community that is partially defined by the game itself. When hundreds to thousands of users simultaneously share a space, they create a virtual society. Because many of the users will be returning and interacting again, then not only is the game persistent, but the social relationships are persistent. Over time these relationships solidify into social

[3] Bartle, Richard. *Designing Virtual Worlds*, New Riders Group, 2003

organizations, which constitute a society. Whether or not the designers plan on it, this society will be a huge factor in the success or failure of the game.

1. Social Systems

Social system design is still in an experimental stage whose potential remains unknown. Yet some staple designs have been widely adopted in most of the successful MMPs. This section goes over some of the core elements of social systems, ranging from types of different social groups, to the rituals and communication methods between them.

a) Communication

Conversation is a staple of social interaction. In PSW games, this real-time communication occurs via text messages, known as chat. In fact, some players and developers refer to PSW games as glorified chat rooms. There are three common interfaces for chat: keyboard, menu, and voice. Although voice chat may one day supplement or surpass keyboard chat, major PSW games today primarily employ keyboard chat. The communication user interface must be manageable while supplying a broad set of common chat features such as logging, channels, filters, macros, friend and ignore lists, and so on.

Although most PSW games use macros for stock phrases, menu chat may also supplement keyboard chat. This is a chat system whereby users can communicate via various pre-written phrases. This has advantages in that it allows slow typists to still quickly send complete messages. Occasionally, menu chat is employed as the primary interface, but it is limited to the most basic communications and prevents meaningful conversation. As an example, *Final Fantasy XI* (PS2) and *Phantasy Star Online* (DC) use menu chat with optional keyboard chat, since a keyboard is optional for the console. *Toontown*, through attempting to ensure child-safety, uses a menu chat and actually requires a secret password to communicate with some by keyboard chat.

b) Groups

Groups are temporary cooperative aggregations of users. A typical group ranges in size from two to eight players, which is a good size for unmoderated chat. Groups may be formed in minutes and dissolved as quickly. Often the group members share a common goal. The group temporarily pools some of its resources and agrees to cooperate to achieve the goal.

PSW games vary in pooled resource features and agreement features. Commonly, tradable rewards are automatically fairly distributed, to the extent fair distribution is possible. The group shares a chat channel. Often a group has a leader who can control membership and set the group's agreement options, such as pooling rewards. Some games provide additional interface options, such as displaying the game status of each group member. As an example, *Final Fantasy XI* and *Asheron's Call* provide health and other status about group members. This assists group cooperation during play.

Some game designs incorporate group missions; for example:, *Anarchy Online*, *Phantasy Star Online*, and *Dark Ages: Online Roleplaying*.

Other types of cooperation exist, such as trade skills. Most PSW games, however, have their primary method of cooperation be via attacking monsters. As an interesting anthropological note, players usually refer to this part of play as "hunting", and many anthropologists believe that human cooperation evolved as a means of hunting large animals.

c) Organizations

An exclusive and persistent group or organization is usually called a guild. The term “guild” has been used historically so often that it has come to mean a semi-permanent player-created organization, regardless of the game's theme, but other names are sometimes employed to match the theme of the world: guild, clan, monarchy, faction, etc. Many guilds take on particular themes beyond normal gameplay, some associated with the game's theme, and some even counter to the game's theme. Guilds commonly pool resources as well, having a guild treasury of tradable goods that various members have donated and guild officers distribute. Often guilds are serially exclusive relationships: a character can only belong to one guild at one time. Guilds have a political structure of guild officers with authority over other members. Minimally, there is a guild leader who can recruit and ban members. There may be other roles as well. A good guild system tolerates loss or turnover of leadership.

Guilds may form alliances with one another. These alliances may be hard-coded into the world or player-created. In a sense these are larger guilds, although the alliance may have less restrictive membership requirements.

Non-exclusive organizations, sometimes just called “organizations”, are catch-alls for permanent player associations that do not fit into the guild structure. As in real-life, one may belong to many separate organizations simultaneously. For example: in *Dark Ages: Online Roleplaying* a player may simultaneously be a member of a village and a religion. There is little implied cooperation or sharing of resources among members of an organization; however, the organization becomes fertile ground for the creation of groups and sometimes an organization's members will gravitate toward one or more particular guilds or alliances.

Organizations may also be defined by the class, species, or semi-permanent set of features of the character. Although features such as a class or species exist in both single player games and multiplayer games, in a PSW, the class and species constitute a loose organization to which a user may belong. As an example, a cleric in *EverQuest* or *Dark Age of Camelot* has a unique service for group members of another class, such as fighters (who need the cleric's "raise from dead" talent when the fighter falls in combat). This affects social rituals.

The members of a class often speak as a single voice, and thereby constitute a special interest group, which is a type of political organization (covered in more detail below under "Political Systems")

d) Rituals

As in real world societies, rituals exist to demarcate the stages of life and degrees of status in the persistent game. New players become assimilated into the community. Many PSW games have features beyond help files that organize experienced members to become mentors to newer players. In *Asheron's Call*, one joins an allegiance by gaining a specific mentor. In *Dark Ages: Online Roleplaying*, one gains a class by having a player in that class guide the new player.

Other rituals may also exist within the framework of the game. For example, many games incorporate weddings as an important ritual. The married status of a player usually has little or no impact on actual gameplay, but substantial resources will still be devoted by players in order to organize one. In *Ultima Online,* counselors, who are community volunteers, officiate a wedding[4]. The mechanics can be tied to the rituals. In other games, such as *Dark Ages: Online Roleplaying*, a marriage is performed only by the priest class, which gives that class a social role. In the Simutronics games, such as *DragonRealms*, players will even pay $100 USD or more in order to get married in grand style.

Rituals can also be important at a higher meta-level. There are rituals to maintain an organization, which can demarcate significant events in the life of a user or the organization's operation. As an example, *Underlight*'s users have new member induction ceremonies. As another example, *Dark Ages: Online Roleplaying* has features for hosting and participating in one of the fictional religion's rituals. Some implemented religious rituals include the admittance, promotion, and removal of members, as well as hosting a weekly service.

e) Events

General public events bring players together to share a space and purpose. These range from in-game plot points, markets, and hunting parties, to weddings and plays. Most events are organized by community leaders and facilitated by community relations.

Events are often tied to the organizations in the game, such as a guild-sponsored event or other organizational event. Sometimes events are bundled together with new features. Some events are tied to the game's storyline, while some celebrate a holiday or an anniversary. *The Kingdom of the Winds*, an early graphical PSW game, offers gag items for Halloween and commemorative items on its anniversary. *Asheron's Call* adds new features and changes the landscape while continuing a storyline loosely themed around the US holiday that it coincides with.

2. Political Systems

Just as persistence of users creates a society, a society creates politics, which is

[4] Kim, Amy Jo. *Community Building on the Web: Secret Strategies for Successful Online Communities*. Peachpit Press, 2000

the maintenance or disruption of power and privileges that a social organization holds over its members and nonmembers. Political systems of some sort are an unavoidable consequence of a persistent society.

In a healthy political system within the game, players accept the authority of those with political power. They may disagree with or antagonize public figures, yet they adhere to the consensus of fair politics and the rules of the political game as it is.

a) The Politics of Community Relations

Game balance during operation of a PSW game is one part game analysis and one part political analysis. From a player organization point of view, game developers can be seen as a power structure within the game's political system.

Today's MMPs design around community relations. Developers leverage guilds and other organizations for service efficiency in designing manageable persistent worlds. Community relations personnel will often facilitate in-game society and politics by speaking with alliance, guild, or organization leaders. Communications could range from organizing events, to discussing upcoming planning changes or additions to the world. When problems arise, community relations staff will be the intermediaries to explain the problems and/or collect customer feedback on possible fixes.

For example, if the game developers determine that a wizard spell is too powerful, and decrease its effectiveness, chances are that the entire wizard's guild will buzz with anger that they have been unfairly targeted for "nerfing" (decreasing of their abilities). Savvy customer relations personnel will anticipate which groups that an upcoming change will most affect, and work with the appropriate organizations in order to set expectations. Representatives from the community, appointed or self-selected, provide an interface layer between the general customer and the game developer. If community relations can work with the wizard guild leaders, to persuade them why an upcoming change is good for the game as a whole, those guild leaders can be very useful in talking to other members of their guild and passing the word along, to prevent player dissatisfaction. In other words, the in-game political power that those leaders have acquired, can be useful to the game developers when controversial changes are needed.

b) Political Game Design

Political systems can also be incorporated as part of game design. For a few examples: *A Tale in the Desert* implemented several ritual features for the passage of laws and acquiring political offices. *Achaea* implemented city and guild wars, and multiple types of structures of government. *Lineage* implemented sieges and taxation. *Shattered Galaxy* implemented factional wars. And *Dark Ages: Online Roleplaying* has both cooperative and competitive political features, such as voting, taking public office, promotion, demotion, and player-versus-player political attacks.

3. Economic Systems

Fundamental to the definition of economics is the notion of allocation of scarce resources. In PSW games there is no (real-life) food, no physical land, and no physical objects within or directly produced by the game. Yet in every successful PSW there exists a thriving economy based on artificially created scarcity. Players buy, sell and trade resources that are invented by game designers and generated according to their algorithms. In order to provide an economy that real-world people can relate to, some developers have used analogies to a stock market but where many of the items traded can also be used as features of the game.

a) Trading

Users trade resources within most PSW games, and a secure interface for game item transaction is critical to retain user trust. An effective PSW promotes its users to trade through convenient and secure trading features and by providing a constant need for new items.

Trading centers, or markets, will spontaneously occur around aggregations of users who are not directly engaged in play. Some PSW games can offer game systems to enable these trading opportunities, such as via market spaces or auction chat channels. In Simutronics' *GemStone,* towns have "merchant quarters", where players can purchase a building as a shop, and offer their own items for sale to other players. At last count, hundreds of players had set up their own shops, and were offering thousands of items for sale via that game mechanism. As another example, *Final Fantasy XI* created auction houses, reminiscent in interface of eBay. These shops also become gathering points for socializing and other types of cooperative play. Some PSW games incorporate trading via events and other social systems. In *Dark Ages: Online Roleplaying*, a player politician may open and operate a fairground, which can have registered merchant boots, a theater, and other forms of entertainment.

Regardless of whether or not formal game systems are provided for it, players are almost guaranteed to find some way to share or trade resources. And where there is trading, there will usually be those "worse case scenario" customers who seek to perpetrate fraud. For example, one common cheat technique is for Player A to offer to sell an item (a magical sword) to Player B for a certain amount of gold pieces. Player B may trustingly hand over the gold, and then Player A will simply run away or disconnect without handing over the sword. Player B's response to this will usually be to immediately complain to customer service and demand that they track down Player A and get their money back. In a small game, this may work. In a large game, this is not an effective use of customer service time. A better way to handle this is to have a game system that allows both players to see what is being traded ahead of time. For example, in *Puzzle Pirates*, both players enter a trading interface where they simultaneously display what they are offering. Only when they both click on "Ready" does the trade take place, and then goods are delivered in both directions at the same time.

Online in-game transactions share the properties of e-commerce, and as such the transaction system in a healthy PSW economy should be similarly secure and user-friendly. A single bug could create a counterfeiting problem that would quickly collapse the economy. Having an obvious interface will cut down on scam trading, by clearly communicating the terms of the transaction to all parties before it is executed.

b) Fashion

Because players interact, the clothes or other personal accessories of a user comprise a fashion identity. Fashion is a critical part of almost every PSW game's economy. The rarer the accessory, generally, the more social prestige is displayed by its position, often in spite of what is generally agreed upon as good taste, or lack thereof. Similar to fashion are luxury or nostalgic items, which provide no game benefit but denote social status or have sentimental value. Even bugs can have extreme value. Just as an upside down airplane on a stamp increases its value among collectors, a rare but buggy item in the game could have a high perceived value among in-game collectors..

Often, fashion is related to social status or game role. In many PSW games, only users with particular game abilities can wear a particular set of clothing. This helps a user identify who might be able to provide a service, such as healing or protection. As an example, in *Dark Ages: Online Roleplaying*, a rogue class is the only class that may perform a haircut and hair dye. Each member of the class is easily identified by the rogue-only costumes. An item with no game effect, such as an ineffectual pet or familiar, is also essentially fashion. Yet this fashion is important and highly valuable, as in-game and out-of-game auctions have demonstrated.

c) Utilities

An item with a game effect has utility, which is the ability to satisfy the needs of a user. It is arguable that any valuable item has utility equal to what someone is willing to pay for it. However, for the sake of this brief discussion, we refer to utilities as items that affect gameplay. PSW games thrive on tradable utility items, which can increase a player's rate of advancement. In a fantasy setting, these will be things such as weapon, armor, potions, scrolls, and so on. Each game balances who has access to which utility, the efficacy of a utility, and the maintenance and lifespan of the item. One way to handle balance is by limiting the possible player profession or level that can use an item.

For example, make powerful swords only usable by players who are level 20+. Many PSW games incorporate a way for players to create their own utility items via the game design. This can also be referred to as "trade skills" or "crafting systems". This can be further expanded by limiting the creation of certain types of utilities to certain types of professions. For example, perhaps only wizards can create wands, but they need a potion to do it, and only apothecaries can create potions. This encourages cooperative game play, as players require the help of other players in order to create the utilities that they want.

d) Faucets and Drains

Most PSW games design a faucet and drain model of the economy. Currency and items are created in numerous ways, and then they are consumed by the system or destroyed. The most common faucets are monsters, created by the server and slain by the users. Players thereby gain a quantity of game currency and/or a set of reward items (also called loot). Trade skills can also create items, or a mine may generate a certain amount of ore per day, which the players can harvest.

Items are then removed (drained) from the game in a variety of methods, which, in good game design, should roughly parallel the rate at which they are generated into the game. Some items are consumed by combat (firing ammunition uses it up) or by trade skills (using up a potion to make something, or combining ingredients to create a suit of armor). Items can also be designed to age and decay at a pre-determined rate, a technique which is often unpopular with players but most effective at ‘turning over’ the economy.

Currency may be drained by selling goods from system-operated shops, or various taxation schemes, such as a tax on products made in crafting. Rent on in-game property is another effective currency drain.

These faucet and drain techniques aid designers in balancing the supply of items and money during development. Once the service becomes live though, the player population may become keenly attuned to the exact rate of generation and decay of much-desired items. At that point, altering the rate of a faucet or drain can result in public backlash; especially if a drain is increased, a faucet decreased, or the effect is felt more harshly by one organization than others in the community. To encourage customer retention through gameplay, the ratio of faucet to drain is usually set in the players' favor in most of the successful PSW games, yet not so much as to rapidly inflate prices or promote hoarding.

In other words, it is important for designers to decide ahead of time how to balance supply. If a game starts with a system whereby all clothing lasts only a month and then needs to be replaced, players will probably adapt to that as the norm, since their expectation is set accordingly. However, if a game starts with all clothing lasting forever, and then a change is made that causes all clothing to start decaying and disappearing after a month, a great deal of customer dissatisfaction is going to result.

4. Conclusion

The subsystems of society, politics, and economy are interwoven, affecting each other. Each requires maintenance and new development to renew interest in the community. Each also affects the game and the game can affect each. A shortsighted modification to a game mechanic can agitate political organizations or destabilize the economy. Design with foresight can empower the users to co-exist in political harmony, and can promote a thriving game economy with a low rate of operator intervention.

.

C. Design to Minimize Griefing

Grief play typically refers to players using game systems or chat to diminish or destroy the game playing experience of other players. This runs the range of using chat to harass other players to using game systems (or bugs) to block, cheat or otherwise stop other players from enjoying the game. Typical grief tactics include blocking desirable game spawns, pulling high level monsters to low level areas, giving invalid information to unsuspecting players and much, much more. A real world example of griefing might be someone who talks on a cellphone in a movie theater. Whether they're doing it through ignorance, or because they are maliciously enjoying the ability to ruin the experience for other patrons, they are causing a detrimental effect on other customers' enjoyment of the product.

The principle of grief play follows from a worst case analysis of player interaction. The rule of grief is: if it can be done, it will be done. It doesn't matter if the behavior is not rewarded or even punished, it will be done. Some players simply do not play the game according to its designers' intentions; instead, they are playing their own game of how many other players can I hurt, and/or how much can an individual victim be injured. As the population of players increases, the spectrum of player interaction widens. Even in a benign population, several instances of grief play are inevitable. Keeping in mind this rule of grief, a PSW designer must examine every game system in the light of how it might be abused.

Let us categorize PSW systems to understand where grief play may be prevented. These categories are: finite and independent; finite and interdependent; and infinite and interdependent.

1. Finite and Independent

For these systems, preventing grief play impairs no playing styles. As an example, *QuizQuiz* is a highly grief-tolerant quiz show and casual game suite with massive simultaneous logins, persistent users, and an in-game economy based solely on fashion. Each point at which grief play could occur was omitted. During beta testing, reports came of user scams. One scammer deceived a naive player into trading a worthless item for a much more valuable item. In one sense, this is the fault of the naive player and not the fault of the system, which worked just fine. It was the player's decision to trade. However, a redesign of the trading interface, which made the unbalanced trade painfully obvious and inserted an additional confirmation dialog, drastically cut down on the number of scams.

2. Finite and Interdependent

In these systems, preventing grief play does impair some playing styles. *The Kingdom of the Winds*, a long-running Korean RPG targeted at casual players, has gone to great lengths to reduce the chokepoints where grief play may occur. For example, items dropped on the ground in *The Kingdom of the Winds* were being stolen. To prevent this, any item placed on the ground was prevented from being picked up by anyone else except a group member, and in the trading interface the exact name of the item appears, along with its durability. As another example, during combat some otherwise non-involved players would jump in to attack a nearly slain monster to steal credit for the killing, a practice called kill-stealing. To prevent this, developers implemented a heuristic to determine who had done the most damage, and once a monster had been significantly damaged, no player outside the attacker's group could harm the monster. *Toontown* and *Puzzle Pirates* go one step further by isolating combatants in an independent subsystem. In doing so, each game had converted an interdependent subsystem into an independent subsystem.

3. Infinite and Interdependent

For many of these systems, grief is impossible to eliminate because the varieties of possible disruption are limitless. Chat falls into this category, as there are practically an infinite variety of messages that may be created with an alphabet and sufficient space. As an example, *Toontown*, which bills itself as a family-friendly game, has identified and removed several points of potential grief play. Chat filters have been employed in several PSW games, but these filters were unable to prevent children from receiving offensive or lewd messages. Even a menu of selectable safe words was unable to prevent this. The *Toontown* designers addressed this by settling on a menu of stock phrases. In doing so, the designers had converted an infinite system of communication into a finite system. However, menu chat limits meaningful communication, which is a fundamental draw of PSW games. To attempt to still allow meaningful communication, a password chat system was designed, which required a password for each player to be exchanged out-of-game before they could chat freely

4. Category Analysis

Examining the above categories, it becomes clear that the best design to minimize grief play is to keep a game finite and independent. However, though many systems could be redesigned as finite and independent, redesigning all systems as such would defeat many other design objectives and might take away much of the fun of the game. In other words, small levels of grief potential may inject risk into the game, but they are often necessary in order to keep the game fun. Many players claim that part of the lasting appeal of *Dark Age of Camelot* is the realm versus realm combat, which is a highly interdependent system. *Dark Age of Camelot* designers severely restricted the player-versus-player opportunities to prevent certain griefing opportunities. To engage in PvP, a player has to be at a high-level, in a particular player-versus-player zone, and can only attack designated enemy characters within the same level range. *Puzzle Pirates* similarly has crew versus crew combat. In it, however, the opposing crew appears almost exactly as non-player characters (NPCs) and almost all communication with these players is prevented. This retains the excitement, without allowing potentially offensive trash-talking, at least, during sea battle.

5. Political Systems and Griefing

Political systems in which a player may adversely affect another player are particularly tough forums in which to prevent grief play. There is a fine line between the use and abuse of power. Even the player population cannot agree on where this line lies, so certainly a single deterministic algorithm cannot satisfactorily distinguish between use and abuse of political power. Checks and balances become critical in PSW politics.

As an example, *Dark Ages: Online Roleplaying* has player-driven political systems. The political power centers are decentralized and limited. In a worst case scenario, a burgess of a village may unjustly and maliciously exile a number of citizens in a day, but only from the village. In this case, gameplay is not seriously affected, since other villages exist, and most gameplay can be performed outside the village. Yet the restriction of how often and how many exiles may occur had to be severely limited, because after years of operation the worst case did indeed occur.

Related to political systems are the justice system and the administration of community volunteers. For example, if two players are involved in a feud, they may attempt to use customer service as artillery in what amounts to a drama game, so one player might report another player for harassment. Manpower resources may then be drained while trying to sort through whether or not a staff-worthy event actually took place.

Sadly, another potential source of grief play comes from within, as Game Masters (volunteers or paid contractors), and even employees, can become sources of grief. Preventing these dangerous situations in the design can save the whole team from many accusations. For example, if game staff is empowered to create replacement items for customers, it is wise to put in logging mechanisms that create a permanent record of who created what, when, and for whom. This will make it less likely that a rogue staff member will go around creating high value items for their friends, and if it does happen, then the tracking mechanisms will make it easier to identify who was doing it, and the extent of the damage.

6. Summary

Though there are many different possibilities for grief play, it is important to not become paralyzed by worst case analysis. Many great game systems can be used for harm as well as the enjoyment they are intended to bring. The key to successful design is to recognize the potential harm of various game systems, and minimize that potential by design prior to implementation. This awareness can also serve the team well if a system goes into the game and is then abused more than expected or in different ways than expected. A savvy design team with grief potential in mind can more easily come up with a fix or prevention mechanism.

Worst case analysis of player behavior is an essential design procedure. Remind fellow developers of this whenever one says, "They won't do that." because some player certainly will do that action, particularly if your game has a large population.

D. Player-Created Content

Many designers consider player-created content to be one of the Holy Grails of massively multiplayer online games. A few games such as *Second Life* and There.com have pioneered major elements of player-created content, but we have yet to see a top selling game handle this in an integrated way.

For discussion purposes, let us exclude player profiles, trade skills, and home decorating as player-created content, although they certainly are forms of it. We also exclude content created by volunteers or contractors, who are an extension of the MMP staff. Thus player-created content refers only to open-ended art, objects, and interactivity introduced by unsupervised players.

The first thing to realize is that Sturgeons Law (90% of everything is crud), is alive and well for player-created content. Though there are some spectacular exceptions, the majority of players, even with streamlined and robust tools, tend to produce content which is poor in quality.

Another hurdle has to do with player motivation. Not all players will want to create their own content. An analogy in sports terms might be that most people are content to watch a game, but aren't going to go out and play it for themselves.

The conventional wisdom in the developer community is that though many players may express interest or energy in creating content, that at the end of the day, less than 1% will be capable of creating *good* content.

As a metric from the single-player world, the most popular PC game of all time is *The Sims*. Lead Designer Will Wright, in analyzing the player-created content associated with the game, found a dazzling range of creativity. At a keynote speech given by Wright at the 2002 Game Developers Conference, Wright shared the following data: Hundreds of fan websites had sprung up around the internet, with new content available for download. This content ranged from "skins" of new graphics for characters, clothing, or environments, to screenshots arranged into elaborate storyboards. When analyzing the quantity of players involved in the most creative endeavors, Wright found that of 3.5 million casual players, 700,000 of those players were "collectors" who would browse to Sims fansites and download new material. 16,000 went to the trouble of becoming storytellers and capturing scenes to publish independent storylines. 2,000 were artists who created new visual content, 800 created fansites, and 30 went as far as programming their own tools. So to get an idea of the small percentage of players who became storytellers, 16,000 out of 3.5 million comes out to less than half of 1% of the customer base. And that's without even taking into account whether what was created was high quality or not.

In the multiplayer game space, examples of poor player-created content might range from cosmetic (clashing or dull colors), editorial (text on signs or documentation being poorly-written or full of typographical errors), navigational (areas are poorly connected, with some having "no exit" dead-ends that trap players), frustrating game play (combat areas or puzzles that are "insta-death" areas where unwary players are simply killed or stripped of items), or unbalancing game play (an area that allows players to easily claim large amounts of gold or powerful items, thereby unbalancing the larger game economy).

Because of all of these hurdles, most companies are reluctant to invest in creation content tools, which therefore further reduces the number of people who try the tools.

In order to make player-created content work, it is important to find and cherish the tiny percentage of players who can contribute valued content. This section attempts to discuss ways that player-created content can be incorporated into PSW design.

1. Policies

A design document should specify the qualifications to become a content creator, and how often that content may be created. It also specifies the editorial strategy, such as the approval process to publish content. For example, in *Second Life,* content is reviewed and rated by other players.

Even with unsupervised content creation, review mechanisms of some sort will probably be necessary, and standards clearly delineated at the start. For example, given enough flexibility, it is inevitable that some worst-case scenario player will eventually attempt to create a pornographic corner of the world. The design document should therefore indicate what standards are expected of player-created content, what methods will exist for requesting editorial changes, and what consequences may be implemented for content which is in clear violation of standards, good taste, or even local laws.

Designing a player content creation system is difficult, in part, because player-created content elevates the player to the status of level designer. There are some similarities to the mod communities of multiplayer games, such as *StarCraft*, *Half-Life*, and others, where players have the tools to create new content. However, the massively multiplayer game vastly expands the interdependence of the game systems and players. In a PSW, one player's content has the potential to upset game, economic, or political balance. This loss of service quality directly impacts customer retention. Therefore, preventing system instability, player harassment, duplication and instant victory conditions is critical to successful PSW service. The Policy document should anticipate these possibilities and have clearly delineated guidelines to address them.

2. Functionality

Player-created content is interdependent with the rest of the game systems and therefore may have drastic consequences. Player-created content empowers the players. The level of dependence of other game systems determines the amount of power the player possesses, since each qualifying player has the choice to create or not at any given time. If the game does not work without the player-created content, then in the worst case scenario, the game becomes unplayable. Likewise, if a form of created content may disable a game feature, then in the worst case that game feature is disabled.

Designing what may be created by an unsupervised player is a delicate balance of creative expression and system stability.

3. Case studies

a) Text MUDs

Although the concept of player-created content may seem radical, it is as old as text multi-user dimensions (MUDs, also known as Multi-User Dungeons). In many MUDs, an advanced user may create new content. In a freeform MUD the content generation rules were only bound by whichever political powers that administrated the MUD, which could range from a corporation to a single student running the game from their bedroom. The danger always existed for some user to generate content that might break the game or otherwise disrupt the user experience. As an example, a new room that provides a constant supply of easily defeated opponents would become the dominant form of advancement, making the rest of the areas obsolete options for a strategic player. As another example, a new room that traps the player would become not just a danger, but also a tool for griefers. A griefer could trick several inexperienced players into entering the trap. A lot can be learned from researching prior successes and failures in text MUDs.

Big budget PSW games today, though, are multimedia. This raises the bar of requirements for creating content and thus the tools have become more expensive to create. As an example, *Furcadia* offers player-created areas which have few game features inherent to the basic structure. The player uses a simple level editor and scripting language to design his or her own interactive spaces. Players have created several isolated, interactive regions, which are linked to the staff-created world.

b) Second Life from Linden Labs

As a case study of player-created content, let us consider the graphical PSW game *Second Life*, which provides thorough content creation features. Object data and world status stream from the server to the client. Through this technology, Linden Labs is able to edit any object in the *Second Life* world. Fundamental to *Second Life's* design, Linden Labs has integrated robust object editors into the user client software.

A player, while still logged into the world, can create a new object using a 3D modeler. The player constructs the geometry, textures, and other graphic and audio properties of the item. The item may be as small as a ring or as large as a mansion. A tour through *Second Life* may show a replica of the Sphinx, a hockey rink, a dungeon, a casino, a sky lounge, a shooting gallery, and much more. However, the same tour may also show a giant toilet, a suburban sprawl, and many works in progress. The player may add rich interactivity through a JavaScript-style scripting language. For example, the casino and the shooting gallery do work, although their entertainment value is low. Since the avatar of the player is also an object, the player may customize his or her own appearance in an infinite number of ways. The costumes players have created are amazing and varied. Some include super heroes, anime characters, monsters, animated vegetables, and giant plush toys.

This creativity is interrelated to the business model; to create permanent structures the ‘resident’ must rent land from Linden Labs (or sub-lease from another player). The problem was, and to some extent still is, that players were creating too much low quality content. Besides the rental fee, other features attempt to increase the quality of content. Users may review and rate each other's content. Creative players may earn back their land-rent investments by earning money when other players spend time with their content, or by selling replicas of their digital objects.

4. Legal issues

The legal issues with player-created content revolve around content ownership, copyright and trademark infringements, editorial control, and harassment. The developer must answer the following questions: Who will own the player-created content? If you own it, will you give some compensation to the creator? Will it be linked to how popular it is? Will you share the copyright on the content? Will you allow the player to retain copyright? Again, what will their compensation be and how will be it be tallied? These are all serious issues.

If the previous warnings have not scared you off, then you are in the minority of online game producers. If you still plan to include player created content, the authors recommend that you seek legal consultation, especially from attorneys that specialize in intellectual property law.

5. Summary

The legal issues involved, along with the design issues, mean that most online game producers have been reluctant to encourage player-created content. Those that have encouraged it though (and even a few that haven't), have found a small percentage of customers who have produced entertaining and high quality accomplishments. It is worth noting that Linden Labs experienced a surge in quantity and quality of user-created content when they changed their license agreement to give players full copyright ownership of their Second Life creations.

From a business case, the early attempts have not yet significantly expanded the massively multiplayer audience, but we believe that player-created content certainly has this potential.

E. Designing for the Masses and Designing for

the Few

Within the massively multiplayer scope, one size does not fit all. Some designs work well on a large scale but fail on a small scale, and vice versa. The size of the player population, the demographics, and the psychographics of the player population determine what designs will please those players. Matt Mihaly, founder of *Achaea*, a successful niche text MUD, said that there is no such thing as a perfect game. The fitness of a design depends on the player. The optimal design for one psychographic may be a recipe for churn for a different psychographic.

Given the large numbers of simultaneous players, there is bound to be diversity in the population. However, not all massive multiplayer games contain equal masses. The most popular PSW games to date have gravitated toward a lowest common denominator of design objectives; whereas, some successful niche games have survived closer to the edges of design space to capture specialized demands.

Some game features, including most personalization, typically do not scale well, that is they have exponential costs as population increases. With good scalable game design, then as the player population increases in size, the core game design still functions properly. However, some other game features may not scale, which means that they will begin to cause problems as the game population reaches a certain size. For example, during *EverQuest's* beta phase, its web site boasted plans for GM-run quests and events. By launch though, these plans had been abandoned. As enticing as it sounds, the math breaks down. Personal interaction between staff and players, on a one-to-one basis, is a clear class of features that do not scale. This is a surprisingly similar analogy to peer-to-peer architecture. The number of connections grows factorially. However, the growth being measured for a personal service is the time required by the service operators.

Political systems are another example. The design of a political system has to take into account worst case analysis, especially griefing. As the population size increases, the spread of the distribution of players increases, so a larger spectrum of player behavior exists. All other things being equal, a larger population will mean the potential for more extreme types of grief play. The meanest of the mean will tend to be worse as size increases, even if the average player is equally friendly. Even things as simple as name generation, which is a very small player customization, have costs that become exacerbated with increased population size. While a small market can police its names and uphold high standards, a larger market game, or a larger demographic spread, has usually used automation. As an example, *Toontown* filters and monitors player name generation to protect the experience of their young players.

Niche market games can offer a higher degree of personal content, and doing so can be a unique selling point of the game. For example, one carryover from face-to-face role-playing games is role-playing, which is pretending to behave as the character in the fictional world. In the Simutronics games (www.play.net), staff routinely spotchecks the game and awards experience bonuses to those players who engage in good roleplaying. *The Kingdom of the Winds* and *Dark Ages: Online Roleplaying*, in the U.S. portion of their market, have offered epic events affected by small sets of the player population. However, those same games cannot offer this responsiveness in Korea, where over 60,000 players may be logged on simultaneously to one of the games.

Styles of gameplay can differ more widely in a smaller market game, which permits innovation at a lower risk. Niche markets support puzzle games (*Puzzle Pirates*), freeform civilization games (*A Tale in the Desert*), streaming player-created sprawls (*Second Life*), warm and fuzzy atmospheres (*Furcadia*), classic Japanese console games (*Dark Ages: Online Roleplaying*), or real-time strategy campaigns (*Shattered Galaxy*). The diversity and depth of each of these games in their own direction extends beyond the center of mass that a mass market PSW game often has to remain near.

Design objectives can become homogenous for mass market PSW games. *Lineage* and *BnB*, each possessing over a hundred thousand simultaneous players, are extremely different genres of games, yet both share common design objectives. Each is relatively easy to start, when compared with the preceding examples of niche designs. *BnB* is a *Bomberman* derivative, with a massive lobby where up to 300,000 simultaneous players have logged in simultaneously. The design is optimized for short, quick bursts of enjoyment. It is extremely streamlined, to the point it is not massively multiplayer, but it has many MMP characteristics.

In summary, the scale and scope of the market will open some doors but also close just as many doors for PSW game designers. All features of a successful game serve the particular player population and their quirks. As the population size increases, extremes of behavior may increase, so possibilities for quirks and deviations of game design may diminish. For a niche market, a cleverly designed system that exclusively appeals to a rare set of playing styles may succeed. But the same system may fail to meet the needs of an audience which is larger by two or three (or more!) orders of magnitude.

V. TECHNOLOGY

Persistent worlds require technology beyond that usually deployed in non-PSW games. The basic architecture of client-server development increases complexity, requiring the diligent and careful separation of user-side code on the client (with associated issues of security) and server-side central processing of game events and player actions. Thrown into the mix are the databases required to store the world state and the ancillary tools and web services required to support customers.

A. Standard Technology

At the most basic level, persistent worlds require servers, disks, memory and networks. This section provides an introductory taxonomy of the standard technologies used in creating and running persistent worlds.

1. Server Hardware Systems

There are two types of computing systems that are common in the implementation of persistent worlds: large single instance, or monolithic, systems, and collections of small systems called grids.

a) Monolithic

A monolithic system is a single system that is responsible for managing all aspects of the world. The physical hardware may be modular, but the unifying characteristics of a monolithic system are a single operating system instance viewpoint presented to the game. Contemporary server systems can easily have CPUs by the dozen, physical memories by the terabyte, and storage well beyond 100 terabytes.

The benefit to the monolithic approach is the simplistic conceptual development model and relatively low administrative costs. The drawbacks to using this system for large games are hardware cost and availability. For large numbers of players with any kind of sophisticated player interaction, very large monolithic server systems would be required. The cost of a monolithic system sized appropriately for a large game is well beyond the budget of most studios. In addition, the size of a monolithic system tends to be fixed; one can add components to the game or hardware, but in general the system has to be taken off line to do so, leading to the issue of availability. If the monolithic system goes down, then all game play stops, and if it goes down unexpectedly, some level of game state will likely be lost.

b) Grids

In abstract terms one can think of a grid as a bunch of systems with some infrastructure software that coordinates and manages the way in which the systems interoperate. Each system, or node, in the grid is only responsible for a small part of the game world at any one time, and the grid infrastructure is responsible for managing communications between the nodes. Since each node in the grid is only responsible for a small part of the world, the nodes themselves can be relatively inexpensive. Thus the benefits of a grid are that one deploys many low cost machines, which give simple scalability while also providing resilience to individual system failure. The drawbacks include system complexity, management overhead, and operational costs.

The low cost benefit is derived from the nature of the grid nodes themselves, which can be low cost commodity computers (PCs). In addition, a properly designed world can be launched on a small number of nodes and scaled up as demand grows. The resilience benefit comes from a feature of grid infrastructure that allows nodes to drop out of the grid in the event of a failure. The online scalability benefit is the ability to add nodes to the grid to increase its capacity. A well designed world, running on top of a mature grid infrastructure, can scale nearly linearly as nodes are added to the grid.

The system complexity drawback comes from the complex programming model needed to take advantage of a grid.

c) Storage

Storing the state of a large persistent world puts great demands of performance and reliability on the system's disk drives. There are many types of disk drives on the market today, with even more file systems, and most vendors support open standards. The matrix of solutions can be mind boggling. To reduce the complexity, consider grouping storage by type of attachment, then pick the type of attachment most suited to your hardware platform and performance requirements. There are basically three types of drive attachments provided by storage vendors: direct attach, network attached, and storage area network.

(1) Direct attach

Most familiar is direct attach, whereby a disk or array of disks is connected via low level protocol, drive controller card(s) and cabling. SCSI and Serial ATA (SATA) are examples of direct attach systems. Benefits of this type of storage are low cost, and high performance. The low cost comes from the small number of components connecting disk to server. At a minimum all that is involved are a controller (often included on the motherboard), cables and the drives themselves. The popular drive types of SCSI and SATA both have relatively high performance. The drawbacks of direct attach disks are that they are directly attached to one server and are generally not shareable to a cluster without assistance of technologies such as CIFS, NFS, or ISCSI. It may be very inexpensive to load up one or two servers with direct attach disks, but when the system starts to expand, money can be lost due to underutilization. Another drawback of direct attach is that when the server attached to the disks goes down, it takes all or part of the world with it.

Another problem with direct attach storage is that it can be difficult to manage just what goes where. As the world grows, it becomes harder and harder to manage the growth of the game. Parts of the world may become tied to specific servers, or groups of users' data get stored on some systems but not others.

(2) Network attached

Network attached storage (NAS) provides a low cost, but comparatively low performance, solution to the problems of direct attach storage. Some NAS vendors try to make their systems seem like magic. There is no mystery; network attached storage is simply a bunch of disk drives connected to a file server. Anyone that has set up SAMBA and NFS on a Linux box has created a NAS server. On the other hand, NAS vendors have an opportunity to highly customize the operating system running on the server, and because they buy their disk drives in bulk they can pass the savings on to their customers. It is often less expensive to buy a packaged NAS solution from a leading vendor than assembling it from parts yourself. In addition, most NAS solutions come very close to meeting the promise of "plug and play". Many of the benefits of NAS are derived from simply breaking the direct attachment between disk and server. Since a server is attached to its storage over the network, if the server crashes, the storage is still there, and can be accessed by any other server. Redundancy is available, at the expense of doubling the amount of hardware used to act as your NAS solution.

The big downside to network attached storage is the fact that your entire disk I/O is going over your IP network. The game communication gets intermingled with storage requests between server and disk drive, and all this goes over the fairly limited bandwidth provided by traditional IP networks. Most people who implement NAS often run NAS on its own network; this addresses some of the performance collision problems, but adds cost.

(3) Storage Area Networks

Storage area networks (SANs) provide the benefits of network attached storage and the performance of direct attached storage but at a premium price. Storage area networks typically use fiber optic connectivity and SAN protocols to connect intelligent storage controllers to any number of systems. Because the network connectivity is separate from general IP traffic there is no data intermingling issue. In addition to the high performance of the fiber optic connectivity, the SAN protocols have been designed with bulk disk I/O in mind. The downside to SAN is the expense in the connectivity. Each server gets a special SAN network interface, often called a Host Bus Adapter. Then there is the expensive fiber optic cabling and custom switching equipment required to connect multiple hosts to a SAN controller.

From a performance and reliability standpoint SAN is an attractive option, but current pricing puts this technology beyond the reach of nearly all developers.

2. Backups

There are many different types of backup techniques, such as online storage, offline storage, off-site storage, and near line storage. One of the least complicated methods of backups is simply to send the data over the local network to another PC. In other words, it's now cheaper to do backups disk to disk, rather than disk to tape. For the cost of a tape drive and a few tapes, it is possible to buy a NAS device and simply backup over the network. This makes an excellent solution for "short term data" or "incremental" backups made during development, and of the world in live production. For production source code, and world snapshots though, it is still recommended to have either a DVD burner, or some spare hard drives.

3. Networking

Networking equipment such as hubs, switches, and routers have become routine commodities. In general a good network administrator will be able to build and grow an adequate system, and any co-location facility can typically provide bandwidth. Ultimately, the game's user experience will be what drives network requirements. Factors to consider include the amount of bandwidth (usually measured in megabits/second), acceptable amounts of packet loss, what kinds of traffic mix will be required (web, file download, interactive move updates, etc) and so forth.

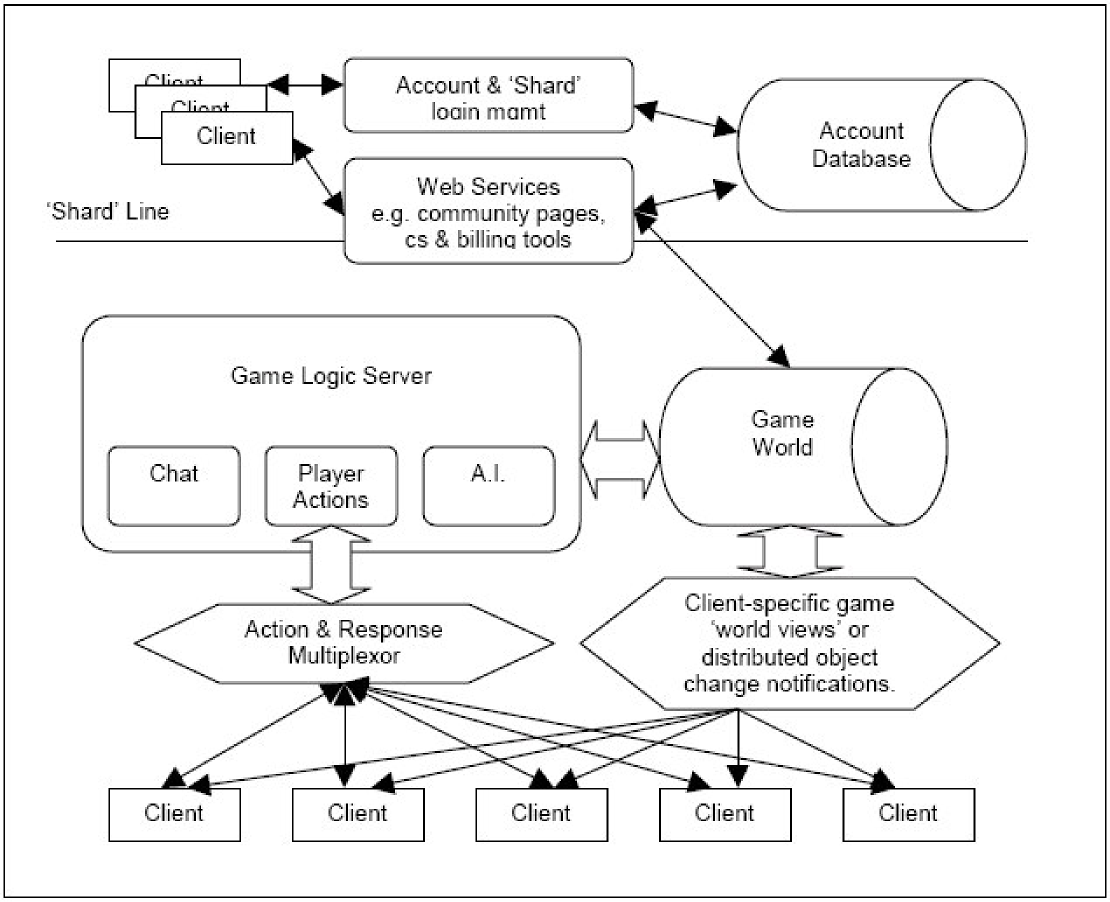
4. Caching

Network caching is a great boon to the Web, but it can cause undesired and unpredictable behavior for a game world. Many ISPs deploy caching within their infrastructure. Since caching could cause game synchronization issues, our advice is not to use HTTP protocol to transmit game-sensitive or time-sensitive data.

B. Software Architecture

This section covers the basic elements that are unique to a PSW's software architecture. Please note that we are not attempting to cover the various aspects of PSWs that are common to other games, for example 3D rendering systems.

With the important caveat that PSW games are sophisticated engineering projects with no single "best" or standard solution, the following diagram example presents a functional, highly simplified view of a PSW architecture:


Client
Client
Client
Account & 'Shard' login mgmt
Account Database
Web Services e.g. community pages, cs & billing tools
'Shard' Line
Game Logic Server
Chat
Player Actions
A.I.
Game World
Action & Response Multiplexor
Client-specific game 'world views' or distributed object change notifications.
Client
Client
Client
Client
Client

In general terms PSWs tend to operate by the above architecture, from bottom up:

1. Client-Server, ‘Dumb’ Terminal

The client acts purely as an input device and view to the game world model. Actions are requested from the game server, usually queued by a separate process to the game logic. Responses and confirmations may be returned similarly, or via the client’s view of the game world, which is updated as the relevant section of the game world changes (for example by updates to subscribed local distributed objects, or periodic updates on movements and actions in local 3D space). Note that it is critical for PSW developers to obey the “never trust the client” maxim, and wherever possible to minimize the need for communications between client and server.

2. Shards and Distributed Servers

Distinct “shard” worlds are managed by different server clusters, and each shard is usually split between multiple physical services in large-scale PSWs. Because of the varied techniques used to accomplish this distinction, we have not featured this aspect in the diagram. The most common technique is to locate different world “zones” on separate servers, with limited interaction between players on distinct zones. A more sophisticated technique dynamically balances players in one continuous world between physical servers.

The “Game Logic Server” may accomplish all the tasks associated with managing the game world model and providing ancillary services to players (e.g. chat, NPC AI) or these tasks may be divided between physical servers on functional lines. Processor-intensive AI or cross-zone (or Shard) Chat, for example, might sometimes be better done by separate machine(s), although this can present substantial issues in edge-cases (does chat lag when everyone gossips at once?).

3. Databases

The Game World will usually be held in memory or in a database, or some combination of the two. This varies according to implementation. The most common technique is to commit important transactions (e.g. treasure wins, kills) to an SQL database, but hold non-critical data (e.g. positions, health) in memory to reduce processing overhead.

a) Standard

Standard databases come in a variety of shapes and price ranges and run on both single CPU and SMP (Symmetric Multiprocessor) systems. When the world grows, the method of improving performance of the world's database is to increase the size (meaning memory, CPU, I/O) of the system the database runs on. Full-function open source databases such as MySQL (http://www.mysql.org) and Postgres (http://www.postgresql.org/) are freely available. Commercial databases from Oracle (http://www.oracle.com), IBM (www.ibm.com/db2) and Microsoft (http://www.microsoft.com/sql/) can provide additional integration, support, and performance or scalability solutions, but at a sometimes significant price.

b) Clustered databases

As the size of a game increases, the associated databases grow as well. Simply making the database box bigger may not be an acceptable solution, for practical, technical, or financial reasons. Clustering is one possible solution. There are at least two approaches to increasing the capacity of database systems through clustering: Shared State, and Shared Nothing. In general shared state database clusters use a common store and all database servers have access to all the data. Performance and capacity is improved by simply adding another node to the database cluster. In order to preserve data integrity, sophisticated locking is required in the database, and often in the file system as well. Oracle Real Application Clusters is an example of shared state clustering, and an excellent overview can be found at http://www.orafaq.com/faqrac.htm.

Another approach to performance is to split the data up into chunks and allow each node in a cluster to work on a piece of the data. This so-called “shared nothing” approach can be implemented transparently by selecting a database server that partitions the data and manages queries for you. IBM's DB2 takes this approach to clustering, and an example can be found at http://www.ibm.com/software/data/pubs/papers/linuxcluster/linuxcluster.pdf. This type of database clustering can also be implemented by simply breaking up the world's data into several different databases.

In the past this would require the game developer to know which database system each type of data resided on and then to code that information into the game engine. Content-based routing provides a solution to this problem by allowing a router to be programmed to route queries from the game engine to the specific cluster nodes holding the data. This technique separates the database layout from the application itself, which simplifies the view of the data to the game. This technique can also allow the database layout to change through network configuration without impacting the game engine.

4. Out-of-Game Tools and Web Interfaces

The use of web interfaces for account management is already quite standard, and increasingly games are using web services to deliver in-game information. These services may be aggregated across shards. These applications are usually delivered via standard web server and application server tools, such as apache and a java servlet container such as Tomcat/BEA weblogic, etc.

Customer service tools are included here, although these may be custom client code rather than web services. It is worth noting that billing and other customer service tools are non-trivial, though often neglected until late in development.

5. Directors

The use of a common account login server and “director” to send clients to appropriate shards and/or zone servers is also commonplace. In some designs “directors” remain in between the clients and physical servers, acting as load-balancers and introducing some fault-tolerance.

Other common technology components include a patch download server, along with patching tools on the client.

C. Middleware

As the PSW game space has increased in visibility and market size a number of prospective middleware providers have emerged with the hopes of serving developers and publishers.

There are compelling reasons why middleware might be attractive to a PSW developer:

- Technical problems of creating a PSW engine are subtle and extremely complex. These problems largely lie outside the traditional area of expertise for game developers, and finding engineers and architects with applicable experience is very challenging. The pool of engineering talent with a background in PSW games is, not surprisingly for such a young industry, vanishingly small.
- For most game designs there is a large set of common problems that a single well-designed architecture might support.
- There are substantial infrastructure requirements, and the operation of such hardware and networking infrastructure is outside of the existing core business of most game developers. There are also economies of scale in the provision of such services, such that a middleware and infrastructure service provider might usefully aggregate customers with substantial savings.

That said, the games industry has traditionally been very slow to adopt middleware solutions, and the PSW game space is no exception, for various reasons:

- To effectively develop and operate a PSW game, many developers feel that they must have the in-house expertise necessary to develop one from scratch.
- A developer can expect to support a PSW game for many years, if not decades, therefore reliance on an external provider for fundamental technology (and possibly infrastructure) leaves the developer vulnerable, particularly if the provider is a venture-funded startup that might not reach profitability or might be acquired by a competitor.
- Game developers are especially prone to the ”not invented here” syndrome. In particular this stems from the belief that technology is a key differentiator in the gaming space.

Middleware Providers

There are only a handful of middleware providers currently offering PSW-specific solutions. We are not addressing here middleware providers for games in general, for example 3D engines, physics toolkits and so forth.

At press time no commercial titles have yet been shipped using any of the middleware products available, although a number are supposedly in development. This makes evaluation difficult. Gamasutra published a review of middleware in Jan 2003:

http://www.gamasutra.com/features/20030115/ferguson_01.html

Please note that this is not an exhaustive list, and that some open-source tools exist that may provide a part of the solution for some projects, for example Twisted at http://www.twistedmatrix.org/ and Nevrax at http://www.nevrax.org/

a) BigWorld from Microforte

Microforte is an Australian developer. The *BigWorld* technology comes out of years of largely government-funded research and development and is designed to seamlessly handle very large numbers of simultaneous players. They claim to be able to support 10,000 plus players in a single world environment, distributed over multiple seamless servers.

BigWorld is currently available to developers for evaluation and is reportedly very easy to begin prototyping with. For more information: http://www.microforte.com/

b) Butterfly.net

Butterfly.net is a venture-funded startup with a high-profile strategic partnership with IBM’s Grid Computing initiative. For more information on the Butterfly Grid see http://www.butterfly.net/

c) Zona

TerraZona is the PSW middleware from Zona Research, another venture-funded startup. Interestingly, Zona also offers support services and infrastructure. They can be found at: http://www.zona.net/.

d) GrexEngine

GrexEngine is a startup developing an eponymous “enterprise class” solution for PSW games. A simpler version of their software for single-machine installations is available. See: http://www.grexengine.com/.

VI. PRODUCTION

The successful creation and launch of a persistent world represents an enormous challenge to even well-established game developers. Unfortunately, as the saying goes, “After launch the work **really** begins.” The live game requires an entirely new service-oriented mindset for game developers used to the “packaged goods” retail model, and developers would be well-advised to position themselves with this service model in mind from the outset.

In this section we outline some maxims and approaches to production that will, we hope, help prospective developers along the yellow brick road. As with Design, the field is both too young and too complex to provide an exhaustive guide.

A. Building a Persistent World

So you’ve got money, programmers, artists, designers, producers, testers, customer service folks, and a kitchen sink for good measure. But before any of these people can actually start to work, they need a plan. Not just a plan, but a “Vision”.

1. Preproduction

Preproduction is the planning phase before your juggernaut of a team starts churning out the game. In preproduction, the formal vision of the game is hammered out. The vision documents for the game need to provide two levels of detail. The first level of detail is the 30,000 foot overview. This one to three page document delineates what the game is and is not. It answers the questions that all good detectives and product managers ask: “Where, when, who and why?” Focus is incredibly important at this stage. If the game vision cannot be condensed into a three page summary, then the vision needs to be refined further.

In concert with this high-level overview, a mid-level game overview is prepared as well. This overview is much closer to the nuts and bolts reality of the game. These design documents address the player experience, all gameplay systems, game world layout and purpose, artistic style, and content plan of the game, and function as the roadmap for the team in production as they lay out the specifics of the game’s framework. The documents should be detailed enough to allow for the creation of preliminary schedules for art, engineering and design.

Doing preproduction well is not easy. It takes time, attention, focus and skill. Yet it is utterly important to finish preproduction with an incredibly solid vision of the game. The authors recommend testing the core mechanics of your game during preproduction – via prototyping or even on paper. This is the opportunity to ensure that the heart of your game is fun.

2. Necessary processes and tools

As early as possible, the team should have at least the following tools and processes in place:

- Revision control system for code, art, and documentation
- Bug tracking system
- Web site for project information and status (such as a Wiki)
- Bug lifecycle and terminology
- Automated build system
- Automated build archival
- Build error reporting tools
- Bug submission from within the game client
- QA for the project
- Process for "locking-down" the asset tree during testing and polish phases

3. Production phases

Production is a lengthy process. PSW games typically have development cycles measured in years. In scheduling the production process, adequate time and resources need to be allocated for the various phases of development. When a game enters production, four distinct phases occur. These phases do not occur in exclusion to one another, but instead ebb and flow as the project progresses. These phases are engine development; art and game system development; content development; and game iteration.

a) Engine Development

The first phase is engine building. Before anything else can occur, the game needs a technical foundation upon which to be built. Among the components which need to be built are:

- Graphics engine
- Network model
- Client engine
- Server engine
- Character storage system
- Billing System
- Art pipeline
- World building tools

Do not underestimate the amount of effort that tools will take to build. Without efficient and effective tools, no team can finish a PSW game on-time or on-budget.

b) Art and Game System Development

After the establishment of the game engine and tools, artwork and game-specific code systems become the next priority. Artwork production, now allowed by a functioning art pipeline, can commence in earnest. In addition, all of the game-specific systems, such as the user interface and the combat system can be started on.

c) Content Development

Once the art assets are in place and the majority of game systems are in place, the focus will turn to generating the sheer volume of content for the game. In an ideal world, the code base and the tools will be stable, so that the workflow of the staff creating quests, dungeons, towns, monsters, and items is as efficient as possible.

d) Game Iteration

Once you have the volume of content you need it will certainly need some play testing which will lead to iteration of the game systems and content to make the experience more fun and interesting to the future players. Having a robust period of iteration for the game is critical to maximizing the success potential of a product.

e) Summary

Constant communication is crucial to a smooth production process. Decisions that are made must be documented and then reflected to the team. The vision documents need to be living documents which can be edited during the course of the project. However, as production progresses, changes to the vision documents should become more and more difficult. This “hardening” of the vision documents represents the commitment of the team towards a certain goal for the project. Very few things are worse for team morale and project focus than having the vision for a game change drastically in the middle of production.

4. Change

That being said, change is inevitable. The production team and management need to accept the idea that the end game will vary, in some cases quite dramatically, from the vision initially set forth. Features will be cut. The content focus will change. Project management should plan for change.

5. Shipping Mentality

Each sub-team during the course of the project needs to shift from a development mindset to a shipping mindset. The difference between the two modes is focus. A shipping-focused team does not implement new features, add large extensions to existing features or spend long hours brainstorming new ideas. A shipping mindset is trying to polish existing features, cut extraneous features, and fend off feature-creep. Feature-creep is the tendency to add new features late in the development process, which often leads to bugs, poorly integrated features, and missed deadlines.

B. Testing Persistent World Games

Testing of a PSW game is different than testing of a single-player game. Among the unique problems the game will face are:

- Latency affecting game systems and the feedback the player receives
- Multiple players stressing the server's capacity to handle player input in a timely manner
- Game balance issues -- The game design needs to handle players feeling under-powered compared to other players. Rates of advancement should be checked to see if they are comparable between different classes. If two friends play, one of whom is a wizard and one a warrior, will they still be able to fight the same creatures together after a month of play, or will one class have advanced far more quickly than the other, forcing them to play separately?
- Content accommodation of player surges and multiple players attempting to solve quests simultaneously
- Minimizing the impact of hackers on the game service and other players

1. Malicious Player Behaviors

A very important aspect of testing is the prevention of malicious actions by players. More information about this is covered under "social and political systems" elsewhere in this White Paper. As a summary, it's important to accept as a fact that in any given online game, people will test all the limits, no matter how absurd. These behaviors will go from attempting to duplicate items to repeating certain behaviors very rapidly to try and force unintended results, to sending malformed network traffic to the servers in an attempt to crash the game. Players *will* decode your network traffic. Every single entry point into the game’s code and content needs to be bulletproofed. Hard-earned wisdom from existing game developers is that just because the designers had a vision for how the game "should" be played, does not mean that players will follow that same vision, or even have any respect for it whatsoever.

2. Basic tools

There are several basic tools necessary to testing a PSW game. A bug tracking system is crucial. Bug terminology and the bug lifecycle need to be established as soon as possible. The developer and publisher need to establish regular meetings, either in person or teleconferencing, to review and act on current bugs and problems. Turbine refers to these meetings as “bug triage”. The earlier triage meetings are established, the smoother the process will be when builds crash and tempers rise.

3. Types of Testing

There are a variety of types of testing for a PSW game. Each of these categories requires different skill sets, tools and metrics.

a) Engine testing

Engine testing is the testing of all underlying code systems which make up the engine. This level of testing needs to address questions like: Does the network model handle high network transmission errors? Is the graphics engine compatible with all of the officially supported hardware? Do the game servers handle load efficiently? Performing this level of testing requires network analysis tools, a client hardware configuration lab, and the network operations facilities to test network conditions such as high-load, high-latency, mis-ordered packets, malformed network traffic, and a variety of denial of service attacks. The network layer needs testing on cable modems, DSL, and analog modems connections.

If a developer does not establish tools and logging to assist in the testing and debugging of your low-level engine code, they are dramatically increasing the risk of a massive launch failure.

b) Game systems code testing

Game systems code testing involves the testing and verification that all of the game-specific logic functions as intended. Do the features of the game work? Does the determination of hit resolution roll all of the appropriate dice and check the proper statistics? Does the user interface handle incorrect input gracefully?

c) Content verification

Content verification is the process of ensuring that all of the game's content is both present and functioning. If the designers built a quest which involves the Bunny Orb, then all the steps to retrieve the Orb need to function. Is the Bunny Orb actually in the game assets? Does it look right both when on the ground and when wielded by the player? Does the item have an icon for display in the player's inventory? In the trading screen? Is the quest displayed properly in the quest panel?

One of the largest hurdles of testing content in a PSW is tracking the status of content testing. The sheer volume of content necessitates a highly-organized test case tracking QA group. If testing is performed in a haphazard fashion, then major problems will slip through the cracks.

d) Content quality testing

Testing content is not just a matter of verifying that it exists and functions in the game. Another crucial aspect of content testing involves assuring the quality of the content in the game. Quality defines whether the content is appropriate for an online persistent game. Take the above Bunny Orb quest for example. Just because the Bunny Orb is in the game doesn't mean that testing is done. Can players find the quest after others have completed it? Does the quest handle non-standard actions by the player? Does the quest handle the malicious actions of players trying to "grief"?

4. The Timing of Testing

Testing of a PSW game needs to occur as soon as possible. The more time between the creation of an error and when it is uncovered, the more difficult it will be to correct and the more likely there will be more instances of that same error. If testing waits until the end of the project, then the content team will have to redo significant portions of the game's content. Any content rework means that new content is not being created, which is bad from a player experience viewpoint.

5. Phases of Testing

As the game progresses, there are several different phases of testing that will occur. Processes need to be established for both the QA and production teams for each phase.

The E3 trade show (http://www.e3expo.com) is its own special beast due to the importance it has in the business side of the video game industry. Movies will need to be made. Special gameplay sequences will need setting up. A game client that the press can play with will need to be built and bulletproofed. Ignore the schedule impact of E3 at the project's peril.

Alpha is when the vast majority of the code for the game is in place, but content is still being developed. External players under Non-Disclosure Agreement (NDA) are playing the game for the first time. The production team needs processes in place for rapidly performing incremental builds and then propagating the resulting builds to the Alpha servers. Communication pathways for player feedback need to be established early and also clearly communicated to the players. Player expectations, if not set properly, will lead to bad word of mouth buzz for your game, so it is crucial to clearly and repeatedly communicate what players should expect in terms of responses to their feedback.

It is an extremely bad idea to leave Alpha with missing features. The unfortunate reality of the current online climate is that when a game becomes "Beta", players expect a fully functional and polished game. From a game development standpoint, opportunities for rapidly iterating on gameplay features, game feel and game focus leave as the project ends Alpha. This is a time when the game comes under sustained intense external scrutiny for the first time. The player feedback is invaluable in refining the game. From a marketing perspective, if a game has crashing bugs and only half of its promised ship features, the game will suffer bad publicity and bad word-of-mouth. No game developer wants to see their game lampooned in Penny Arcade for being buggy. That can only lead to lower sales.

Alpha testing requires:

- External bug submission tools
- Private message boards for player direction and discussion
- NDA enforcement resources
- Processes to deliver new builds to the hardware facility

Beta testing is an outgrowth of the Alpha test phase. More players are allowed into the game servers as the network operations center for the game comes online. NDAs are usually applied although they lose their effectiveness as more and more players enter the game. The Beta phase should focus on game stability, engine performance under load, and all of the live game processes.

Beta testing requires:

- All the alpha tools
- Customer support personnel
- Phone support personnel (not required, but recommended)
- Full-time public relations personnel
- Efficient and automated build delivery and propagating

A word of warning: a poorly run Beta phase is a surefire way to guarantee mediocre sales and subscriptions. There is only one chance to make a good first impression so open Beta should only be done when you believe you are completely ready to launch.

Another word of warning: Players in Beta do not play the game the same way they will after launch. In Beta, it's understood that the entire database will probably be wiped clean once the game goes live, so players will be more likely to take chances. For example, players may enjoy killing off their characters over and over again for the fun of it, and they won't care about item or stat loss. Once persistence is ongoing though, players will play the game differently, begin collecting items in earnest, and become much more miserly about their stats and other status indicators. This may surface problems not seen during the Beta period, so it's important to be ready to quickly react to these emergent issues after launch.

C. Launching a Persistent World

Having two hundred thousand people hitting the “Register” button within five hours of each other requires bulletproof software, hardware and networks.

The launch phase is the highest pressure phase of development. If there are problems with any aspect of the game, from the game technology to content to tools to customer support to billing, they will be highlighted at launch.

It is the recommendation of the authors, that the game development team not be allowed to take vacation time near the launch date of a PSW. Instead they should be encouraged to take vacations before this stage even starts.

Unless you are extremely well prepared, and lucky, assume that your game will have problems with billing, either in registration, CD key authentication or web site access, due to the massive influx of customers on Day 1.

Then once players get past registration they need to enter the game. So during testing, make sure that stress-tests are run on both the login services and servers.

After the registration and login hurdles are overcome, then players are in the game, and the bugs, crashes and complaints will escalate.

The first days of launch are also critical in establishing a rapport with the community. A single spokesperson for the company should relate server downtimes and estimated restarts, make important announcements and interact with the community. The customers need a face, or at least a name, to associate with the product. If a spokesperson is not provided, the player community will either elevate someone of their choosing or just assume that the developer doesn't care (or perhaps both). If the developer does not control the message being given to the players, then rumors and misinformation will rule the day. Official and frequent messages are important not just for the players, but for the industry press as well.

D. Live Team Production

Once a game has been launched, it becomes an entirely different beast. The game project transforms. A common mistake made in the early days, was that some companies looked at multiplayer games the way that they had when they worked on single-player games. In the single-player world, the majority of effort goes into the game before it launches, then the game hits the shelves of stores, and the development team goes on to their next project.

In the PSW space, this cycle is reversed. When a game is shipped is when the real work starts. The development team becomes a live team, and the game becomes not a product but a service, whose aim is to entertain players and generate revenue.

1. On game design

Once a game has shipped, the game design calcifies. Players react extremely adversely to any game change which affects their characters in any fashion. If there are two classes and one has the damage of its attacks increase, then it will seem that the other half of the player-base will be complaining about how the developer has made them weak. If the developer's reaction is to increase the speed of the other class's attacks, then it will seem as though the other half of your players start complaining about unfairness.

Constant communication with the players is crucial to maintaining a happy and content player base. Communication methods can be via game documentation, scheduled forums & chats, in-game news updates, message board postings, poll surveys, and a variety of other techniques. Players who feel engaged in the game will stick with it through thick and thin. These players are great for generating a lively and exciting game world. From a business perspective, they improve the profit margin of the game.

Aside from communication, there are three other categories of tasks competing for the live team's time: new content and features, bug fixes and improvements, and game balance tweaks. Players want all of them, however budgets and deadlines say otherwise. Whenever trade-offs are made, the player-base should be informed by your customer relations team.

2. Maintaining and Growing the Game

There are three primary tools which a live team uses to maintain and grow the game: live events, content updates, and expansion packs.

Live events are when developers or other staff enter the game world and lead groups of players through customized experiences. These events might revolve around killing a special monster or exploring a remote dungeon. Many games such as *Ultima Online* and the Simutronics play.net PSWs run live events for their players.

Content updates are when the live team introduces new content at no extra charge to players as part of the game experience. *Asheron's Call* 1 and 2 introduce new content on a monthly basis.

Expansion packs are large amounts of new content and features sold either electronically or through the retail channel. Typically, the new content and features are limited to only those players who purchased the expansion pack. *EverQuest* has had considerable success with expansion packs.

VII. CUSTOMER SERVICE AND RETENTION

Customer Service lies at the heart of successfully operating a PSW game. Players will expect support and prompt service in return for their monthly fees. Game providers who can deliver on expectations will be able to enhance retention and, perhaps more importantly, word of mouth exposure for their game. Failing to deliver on customer service can have numerous negative consequences, although to be fair, some very successful PSW companies have a poor reputation for service amongst their demanding users.

In this section we give a brief overview of the mission and components of PSW customer service.

A. An Introduction to Customer Service

1. The Mission: Convert and Retain

The first awareness that most customers will have of the game will be via marketing or word-of-mouth. Typically new customers are brought into the game via a free trial, and then if the customers like the game (assuming they can get through the hurdles of installation and login), they convert to paying customers. The primary purpose and mission of customer service is to retain customers through the conversion process, and then for as long as possible thereafter. Whether the customer is being assisted with installation, billing or game issues, keeping the majority of customers happy and subscribing is always the goal.

Some elements of customer service are more straightforward than others of course. Technical and billing issues can almost always be brought to a satisfactory conclusion relatively quickly. Game issues can be much more problematic, as they tend to be more subjective. These issues range from difficult to understand game mechanics to customer harassment and even outright scams of one customer by another in-game. Indeed, some customers (grief players) will do more damage to the game environment by being kept in the game world, to the point of actually driving away other good customers. In those cases the challenge of customer service will be to figure out which customers need to receive quality care, and which ones, to put it simply, need to be shown to the door and encouraged to find some other method of entertaining themselves.

Every issue not resolved to the satisfaction of the customer can lead to the customer leaving the service. Fortunately a satisfactory customer service experience can lead to greater customer loyalty. It is highly recommended that game makers pay significant attention to the quality and responsiveness of their customer service.

2. The Politics of Game Changes

In general, players respond to game design changes in similar manners as they do to billing policy changes, corporate policy changes, administrative policy changes, and client security changes. In many decisions of operations, the lines between game and service are blurred into one, or are so interdependent as to make customer service synonymous with player service.

While a public policy primer is beyond the scope of this document, the mention of perennial problems is warranted. The players may respond en masse, as members of their own special interests group, often tied directly to their permanent game roles. As an example, suppose a policy change is implemented in a fantasy PSW that affects the dwarven races more than it does other in-game races. When those players who primarily play dwarf characters perceive an injury, then they will be the most vocal and most likely to leave.

This shifts the role of game balance from mathematics into politics. Service providers can retain customers better by isolating character balances early on. They can also announce upcoming changes. The only thing a PSW game customer hates worse than a necessary nerf (game asset devaluation), is an unannounced and unexplained nerf. Given that the population is huge, the service is guaranteed to have a few vocal players who are the least receptive to change. Good customer service anticipates their needs and mitigates the impact. Please see the section on "Political Systems" for more information on this.

3. Emergencies

Another necessary role performed by customer service is response to emergencies. No production practice to date has succeeded in eliminating 100% of programming bugs or design flaws. When a critical bug is exploited, en masse, there is an emergency. For instance, almost any instance of the class of duplication bugs is an emergency. These are situations where due to an unanticipated loophole in the game design, a player can duplicate one or even all of their inventory items. If a player can make something for nothing, especially valuable somethings, then the economy will crash from a flood of counterfeit game items. These kinds of emergencies, if not stopped in minutes or hours, will cause negative chatter, decrease the perception of game quality, negatively affect game play, and ultimately gouge the good faith of the customer base.

4. Solving Customer Problems

If a player gets to the point where they escalate a problem to the game provider, they really want that problem solved. The first impression given by the service personnel is in how responsive they are to the submitted problem. The next thing a player notices is how empathetic the service personnel are and how effective they are at diagnosing the problem. Finally players want consistent and effective solutions to their reported problem. Of course doing all of these activities well requires significant effort and training, yet it is the expectation of our audiences that we will do this well!

B. Customer Service Tools

One of the most costly areas of running a PSW is customer service. Customer service runs 24x7x365 and is people-intensive. Tools that improve staff's ability to service customers quickly and accurately are a necessity, not an option.

It is highly recommended to new developers that they bring an experienced customer service manager into the design and development process long before launch. Launch will be a high stress time when customer service is working overtime to address unexpected issues from a large quantity of customers all trying to get into the product at once. It is not the time to be belatedly figuring out what tools that customer service personnel need in order to do that job.

Some of the tools needed will end up being very specific to your specific game and service, and they need to be highly integrated with the game and service infrastructure. Development of customer service tools should be thought of as an ongoing investment in raising service effectiveness and lowering customer service costs. Good customer service is a good business decision. It will increase the percentage of customers that convert from free trials to paying subscribers.

Good customer service also decreases attrition rates, by addressing customer problems to prevent those customers from cancelling their accounts. This investment in customer service, such as calculating costs per service call, and monthly cost of customer service, can be balanced against the costs of not having customer service. For example, the decrease in revenue from customers cancelling, the cost of acquiring new customers to replace those that have cancelled due to frustration or a perception of low game quality, or the decrease in customer conversion that might occur if there wasn't any customer service to assist new players with login issues.

1. Basic Customer Service Toolkit

Customer Service tools, at minimum, will include:

- Systems for managing users, including changing passwords, examining and changing character data, moving characters on game servers, and so on.
- A method of banning user accounts. Depending on distribution model, it may be possible to effectively ban users by their credit card numbers. Most systems ban by IP address, but that is problematic in the face of dial-up and DSL IP pools. A more robust method is to ban accounts and at the same time taint the machine's 'unique' MAC address. This has been successful for *Puzzle Pirates*, with the caveat that there exist at least five ‘default’ MAC addresses generated by modem users and some network hardware.
- A user record, keeping tickets or logs on complaints by and against the

player, possibly in-game support issues and so forth. This is very useful.

- An automated mechanism which players can use to submit bug or typo reports. For example, in the Simutronics game *Alliance of Heroes* (play.net), a player can use the command "TYPO <message>" to submit a bug report. The report will include the name of the player who issued it, any optional explanation text that they care to include, the time that it was issued, and where the player was at the time that they submitted it.
- An automated mechanism which players can use to submit a complaint about another player. This tool should log the last communications between the complaining player and the one they're complaining about. Complaints will often be of the nature, "So and so just said something obscene to me," so having an automated log of this will be very helpful in catching obscene and otherwise offensive players.
- A ticketing / queue system for handling in-game and out-of-game support requests.
- Cancellation tracking. It is highly recommended that when users cancel, that they are given an exit interview or at least an automated survey where they can indicate why they are canceling. Staff should also be assigned to read these answers carefully.
- Tools for analyzing user behavior (a whole topic in itself).
- Tools for customer service to communicate gathered data to the developers

It's common to have these tools split between in-game commands, custom game or support tool clients, and web services. One would expect the trend in the future to be towards web services.

2. Billing Systems and Billing Support

Inevitably in the process of taking money from customers they will have problems and issues that need resolving. Common cases are:

- Statement Review. Aside from creating an account in the first place, one of the most common things that customers will want to do is check the status of their account in terms of what charges have been placed lately. There should be an easy and secure way for them to do this on the company's website.
- Account Changes. Customers will routinely be changing their credit cards, addresses, phone numbers, email contact information, and so forth. Simple automated mechanisms should exist to allow them to keep their account info updated in a secure manner.
- Cancellations. Most customers will expect to find a cancellation option (or at least information) on the company's webpage. Though, even if a nice easy way to cancel is provided via the web, customers will still try to cancel via other means, such as via email and phone calls. Note: Some

developers say that it is a bad idea to make cancellation too easy, and require for there to be a human in the loop on all cancellations. This gives customer service an opportunity to address whatever problem the customer may be having that prompted the cancellation.

- Refunds. A decision needs to be made on what actions or complaints may warrant a refund or credit. It is common for some customer service departments to take the official line of 'no refunds', yet it is not always economical to argue about the matter with a customer. In many cases, especially with a customer who is upset but with whom there is no prior record of problems, the more effective strategy is to simply issue a refund and move on. As an alternative, issuing a credit for future months may be an even more effective solution, since it provides value to the customer, encourages them to stick around, and costs the developer practically nothing (since they would have lost the revenue anyway if the customer cancelled)
- Card charge problems. If a customer has a situation where they were billed multiple times, or they see a charge they don't recognize, they will likely want to discuss it with someone at the company.
- Requests for free accounts, discounts, etc. This is at the developer's discretion.

a) Phone Support

Most game companies agree that phone support for in-game issues is probably a bad idea, but one exception is in the case of billing support.

The primary method of addressing billing issues should always be via automated mechanisms such as email and web forms. Phone support, if it is offered, should be an additional method of communication and not the primary one. The advantages and disadvantages of offering such support can be considered as follows;

(1) Advantages

- Better new user acquisition: Some customers will only give their credit card to a human over the phone, and will balk at the idea of entering it into a web form.
- Customers on the phone may be more inclined to "impulse buy" if a customer representative can reassuringly explain the advantage of an additional package or feature
- Better fraud resolution. Credit card fraud is an unfortunate fact of modern society. Some of a game's customers will sign up with credit card numbers that are not their own. If a victim then sees an unrecognized charge from a game company on their statement, having a phone number for them to call can be invaluable. Many victims may not have internet access (or tech savvy enough) to write an email, but will call a phone # if they see it in order to challenge the charge. Without telephone contact information, victims are more likely to dispute the charge with their credit card company, which can

take up even more customer service time getting it resolved than would have been taken by a phone operator. As a further advantage, allowing phone contact by victims can be a quick way of warning the game developer that some fraud is taking place, so the game developer can more easily pass along patterns of fraudulent behavior to law enforcement.

- Decreased Cancellations. If a customer goes to the trouble of calling to say that they are going to cancel their account, the customer service rep may be able to identify what problem that the customer is having, and address it, in order to prevent the cancellation. Human representatives also have the option of offering "stick around" credits to try and keep the customer from cancelling: "Well, let me give you two free months. If you still want to cancel after that time, you can, but this way you have some more time to play and see if that's what you really want to do."

(2) Disadvantages

- Having a phone center is an additional expense. Staff must be trained and paid, phone lines need to be set up, and if credit card numbers are being passed around, there may be local insurance or bond issues that need to be covered.
- A certain percentage of customers, especially in a community-intensive service such a multiplayer game, will want to call just to chat or "check on their account", even though they don't actually need any assistance. This can use up resources without generating any positive revenue effect.
- Some customers may be quicker to cancel if they can just pick up the phone.
- In-person phone service can take time. Updating account information over the phone could take several minutes, vs. the few seconds it might take to handle it via an automated system.

Developers need to balance out the cost of a call center, vs. the revenue that it can earn (or prevent from being lost) by making it easier for customers to resolve billing problems. Most large PSW developers have made the decision to have some sort of billing phone support, even if it's only during "bankers' hours", For example, the Simutronics billing center (www.play.net) is open from 1-9 p.m. Eastern, Monday through Friday. The hours are skewed later in the day for two reasons: One is that peak usage will generally be in the evening hours, and the other is to have the best window possible for customers to call in from different time zones.

3. Email Tracking and CRM

Another type of customer service application worth investigating, is an issue-tracking and/or Customer Relationship Manager (CRM) system. CRMs are applications which can coordinate customer queries from multiple sources into a single database. For example, if a single customer is generating queries via email, the message boards, in-game questions, and telephone calls, a sophisticated CRM system would be able to pull all these queries together so that any customer service rep in any of those venues would have access to all of the information at once. Another type of CRM system might be one that would take email-type queries from customers, associate them with game account information, and automatically route the questions via a series of FAQs.

That said, some developers massively over-invest in pricey billing and CRM systems that are needlessly complex. There are other ways to accomplish the same effects with in-house and open-source tools, which can take care of the same customer issues for much less cost.

C. Support Staffing

1. Live In-game Support

The provision of live in-game service is a requirement for nearly all persistent world games. In the traditional fantasy MMORPG worlds customer support representatives are called 'GameMasters', but other titles are coming into common usage or are dependent on the game's genre. For example in the Simutronics game *CyberStrike 2*, the staff are called "CyberOps". In *Puzzle Pirates* they have "OceanMasters", *Anarchy Online* has "ARKs" (Advisors of Rubi-Ka)*,* and many other games simply use the acronym "CSR", for Customer Service Rep.

Most games operate a ticketing or queue system, whereby players petition the support staff for assistance and wait for their 'number' to be called. Unfortunately many games end up with long queues of support issues, such that the player may have logged off by the time their number is called.

There is no doubt that many games have been woefully under-staffed, although often the acute service problems stem from design decisions and/or bugs. PSW operators are well advised to run analysis to identify the top customer service queries, and then work to eliminate the most common issues through engineering changes or alternate communication methods such as a news banner whenever possible.

2. Paid GMs, at home and away

Most large PSW games utilize a substantial full-time support staff of paid GMs. These are often entry-level jobs given to local youngsters hopeful of a job in game development.

One of the advantages of a successful PSW is that it attracts a large audience of loyal players who would adore working for your company. Some games, such as those at the Simutronics play.net site, or Three Rings' *Puzzle Pirates,* recruit support staff from the established player-base. Staffers then work part-time from their own homes, as independent contractors. An advantage to hiring from within the player-base is that the staff members already understand the game very well, which results in substantial savings in training and related issues. When hiring, it is also easier to judge the quality of a potential staffer's work, since their social skills, time available, trustworthiness and skill in online communication will have already been observed via their style of gameplay.

3. Volunteers

Many players will be glad to offer their services to the game for no monetary compensation. Whether a game operator is wise to take them up is another matter. There are a number of major issues:

- Powers. If volunteers are going to usefully solve support issues they will need to have some power, such as ability to manipulate the game environment, instantly transport to different game locations, or communicate to staff or players in ways that normal players cannot. This is dangerous, as some potential volunteers will be attracted to the powers rather than the responsibility, and may not exercise these powers for good. For example, a player who is empowered to replace lost silvers can be very tempted to use the power to provide free silvers to their other player buddies.
- Management. No matter how well organized the volunteer staff, there needs needs to be a paid employee at some point who manages the volunteers and keeps an eye out for abuse. This manager will need resources to accomplish the job, such as tools to track the behavior of the volunteers.
- Training. Volunteers will need to be trained
- Resources. Volunteers will need software tools to do their job.

Overall with volunteers, the best results have been achieved by giving them positions geared towards the helping of new players and increasing the fun in the game world; for example, answering new player questions and running events for players.

From a legal perspective, the area of volunteers fulfilling customer service roles is controversial, as both Electronic Arts (with *Ultima Online*) and AOL (in its chat rooms and forums) have been served with class-action lawsuits. EA reportedly settled their lawsuit recently, to the tune of ~$1,000 per volunteer: http://terranova.blogs.com/terra_nova/2004/04/uo_lawsuit_sett.html

The AOL investigation by the US Department of Labor was dropped, but apparently civil cases are still crawling through court.

The legal issues are not yet clear, but it seems that if no monetary compensation or equivalent is given to the volunteers, and their time is *not scheduled* then one may be on reasonable stable legal ground. If some sort of compensation is given, then the staff members can be classified as paid contractors (see "Paid GMs, above). Nonetheless, because of the ambiguous issues that are not yet resolved, new developers are encouraged to exercise caution when using volunteers.

D. Community Dialogue and Forums

What we typically call community management might be more accurately described as managing communication with the community.

Many players will pay little or no attention to game news, announcements, message boards, and the like. They simply login and play, and if they hear anything about upcoming game changes, it's going to be via word of mouth from other players. Others, however, will seem to almost live in the game forums, hanging on every utterance from a developer. These players will then debate and argue among themselves endlessly, analyzing each developer word and whim. The active and attentive portions of game communities really want to hear from the game and service provider, and they like being kept informed in a straightforward and timely manner. This is not to say they will like all the information conveyed, but they do appreciate getting it. Though the quantity of active players may seem small in comparison to the rest of the customer-base, it is important to keep in mind that the active ones will probably be the heralds that spread information to the rest of the non-news-reading population. So it behooves a good community manager to stay aware of who the community's informal leaders and communicators are, to increase the probability that those informal leaders are communicating the information that the developers want communicated.

1. Trust

Good communication can enhance trust. In a PSW game, this trust element is important, as the sum total of the players' time investment is often represented by the avatars and objects which they have collected within the game. Anything that causes a player's stats or items to be lost in unexpected ways, can decrease the player's trust in the game.

Player trust can also be diminished or lost by inconsistent or delayed communication from the game's developers. Further, trust can be destroyed if a developer is caught making announcements that are overt "spinning" or, worse, outright falsehood. And just like in a real world relationship, once trust is lost it is almost impossible to regain. In our medium, lost trust leads to lost subscribers.

Gaining player trust is simple in theory but often difficult in execution. A good start, of course, is being careful to always tell the truth when speaking to the customers. This does not mean that it is necessary to tell them every single detail and screw-up, but it is important to communicate in a timely, accurate and straightforward way on all issues relevant to the players.

One of the most common mistakes made in community management is confusing the communication function with marketing, and therefore "spinning" the information given to the players. Players usually see through this manipulation, and it lowers trust in the community manager and service overall. For example, developers will often mis-judge the amount of time that is required to develop a new feature, and will post on the message boards that it will be available "soon". This word will often provoke ridicule from savvy players, since they will have heard it so often. There is even a satirical acronym used by the players for this kind of promise, called RSN (Real Soon Now). To prevent this kind of reaction, developers should be careful not to use the word "soon". They can instead say that a system is being worked on, but they should never announce its release as imminent unless a hard release date has been committed to. And then if such a date is announced to the players, the developers had better damn well meet that date, or risk losing more trust.

2. Consistency

Another important requirement is consistency. Community managers should use standard message formats for particular types of messaging and have daily, weekly and/or monthly communication cycles for particular communications.

Another common mistake is when the in-house developers go into player forums and make announcements, without cooordinating with the community manager (or even worse, participate in flame wars). Incongruous communications from marketing staff, community managers and developers will undermine player trust in the service.

For live products, the Community Manager should have review oversight of all communications for optimal management of the player community. This view elevates the importance of this role, but it can help to prevent further problems down the line.

VIII. CLOSING WORDS AND APPENDICES

The field of persistent world gaming is only just emerging as a real business and entertainment medium after a twenty-year gestation period. The pieces of the puzzle are in place; the audience is ready and growing, getting broadband-connected with PCs and, increasingly, consoles and wireless. Early products in this market have had tremendous $100MM+ success with the early adopter audience. The challenge we now face is to break persistent world entertainment to the mainstream audience, so that it can take its place as the rightful mass entertainment media of the 21st Century and, as Arthur C. Clarke put it, ”Eat Television Alive”.

The authors hope that you’ve found a valuable resource here in our first Persistent State World white paper. We have only scratched the surface of this broad, important and growing subject. However, this leaves plenty of material that we might cover in future iterations. We presently plan to issue an update to the paper in 2005, and we encourage interested contributors to get in touch with the editors at: http://www.igda.org/online/

A. Further Reading

1. Books

Massively Multiplayer Game Development. Ed Thor Alexander. Charles River Media, Inc., 2003.

Designing Virtual Worlds, Richard Bartle, New Riders, 2003

Developing Online Games: An Insiders Guide, Mulligan and Patrovsky, Pearson Education, March 2003

Community Building on the Web: Secret Strategies for Successful Online Communities. Amy Jo Kim. Peachpit Press, 2000

2. Other Online Resources

a) Mud-dev

The MUD Development mailing list is not platform, language or game specific, but concentrates on discussing the design and implementation of any and all MUD servers, systems, and MMOs in general. Many of the issues of PSW design and operation have been touched on over the years, making the archives an incredible resource.

http://www.kanga.nu/lists/listinfo/mud-dev/

b) Gamasutra.com:

"Massively Multiplayer Resource Guide."

http://www.gamasutra.com/resource_guide/20030916/index.shtml

Carpenter, Adam. "Applying Risk Analysis To Play-Balance RPGs" http://www.gamasutra.com/features/20030611/carpenter_01.shtml

Frost, Paul. " Postmortem: The Tools Development of Turbine's Asheron's Call 2" http://www.gamasutra.com/features/20030820/frost_01.shtml

Goslin, Mike. "Postmortem: Disney Online's Toontown" http://www.gamasutra.com/features/20040128/goslin_01.shtml

Kennerly, David. "Better Game Design through Data Mining" http://www.gamasutra.com/features/20030815/kennerly_01.shtml

c) GameSpy

Archives here contain a series of articles surveying the past, present and future of the most famous MMOGs in the U.S.

http://www.gamespy.com/amdmmog

d) TerraNova

Good discussion of online world economics.

http://terranova.blogs.com

e) Raph Koster's Site

Raph Koster maintains a useful set of essays, talk notes and other information on his website:

http://www.legendmud.org/raph/gaming/

f) Richard Bartle's Site

Richard Bartle, creator of the original MUD, likewise maintains a useful library:

http://www.mud.co.uk/richard/

g) David Kennerly's Fine Game Design

Our own David Kennerly has a fine site on Game Design:

http://www.finegamedesign.com/

h) Korea Notes

"CSC540 Network Architecture": Including reports by leading Korean massive multiplayer game developers. Korea Advanced Institute of Science and Technology.

http://cosmos.kaist.ac.kr/cs540/

B. Contributor Biographies

- **Daniel James, CEO, Three Rings Design, Inc.**

  Daniel James is founder and CEO of Three Rings, a San Francisco developer and operator of massively multi-player online games for the mass-market casual audience. Three Rings' first game, *Puzzle Pirates*, combines accessible and fun puzzle games with the depth of a social persistent world. Prior to Three Rings Daniel consulted on game design, toiled for many years on *Middle-earth Online*, and co-founded two profitable UK internet startups, Avalon and Sense Internet. Contact him at psw@djames.org.

- **Gordon Walton, VP and Executive Producer, Sony Online Entertainment**

  Gordon Walton has been authoring games and managing game development since 1977. He currently serves as VP and Executive Producer for an unannounced product at Sony Online Entertainment. Prior to joining Sony Online Entertainment, Gordon was VP and Executive Producer of *The Sims Online* at Electronic Arts/Maxis, and prior to that with Electronic Arts at Origin Systems managing *Ultima Online*(tm). He also served as Senior VP of Kesmai Corporation where he oversaw the development of *Air Warrior*(tm) and *Multiplayer BattleTech*(tm) among other online games. Gordon has owned and managed two development companies and has been development manager for Three-Sixty Pacific and Konami of America, Inc. He has personally developed over thirty games and overseen the development of hundreds more. He has worked exclusively on massively multiplayer games since 1995.

- **David Kennerly, Game Designer**

  David Kennerly developed five massive multiplayer games in the U.S. and Korea, holding titles of producer, game designer, community relations, and technical writer at Nexon. Before joining Nexon, he designed The *X-Files Trivia Game* for 20th Century Fox. He has published articles at Gamasutra.com, in IGDA white papers, and spoken at the MUD Developers Conference. He is the author of the final chapter in a forthcoming textbook on scriptwriting and storyboarding for games. He enjoys discussing game design at kennerly@finegamedesign.com.

- **Nova Barlow, Director of Support, Themis Group**

  Nova Barlow brings experience from the top ranks of the former *Ultima Online* interest program, where she designed and implemented many short and long term quests for a populace of several thousand players, covering both long and short term entertainment needs during her tenure. Under her direction at Themis, she has worked with players to bring a unique experience and approach to deployment of official fiction in *Jumpgate* and wears many hats as researcher, editor, story writer, and consultant. Ms. Barlow is the lead author of the Themis Report 2004 and was a speaker on interactive storytelling and massive multiplayer games at the recent Computer Game Technology Conference in Toronto.

- **George Dolbier, Senior Architect, IBM**

  George Dolbier is a Senior Architect working in IBM's Online Gaming group. Prior to IBM he was a Unix kernel and database engineer for the likes of Informix, Oracle and Sequent computer systems. He was also director of engineering at Thrustmaster/Centerspan, producer of peripherals, components and communications software for the gaming industry.

- **Justin Quimby, Lead Engineer, Turbine Entertainment Software**

  Justin Quimby is a lead engineer at Turbine Entertainment Software. Over the past six years, the Brown University graduate has worked on each of the award-winning video games in the *Asheron's Call* franchise and *The Lord of the Rings: Middle-Earth Online*. He has spoken at GDC and has an article published in Game Programming Gems 4. Currently, he is realizing his childhood dreams of working on *Dungeons & Dragons Online*.

C. IGDA Online Games Steering Committee Members

- **Brian Robbins**

  Brian Robbins is Director of Online Gaming for Fuel Industries, a developer of web applications and games. He has contributed to nearly 100 web-based game titles, and enjoys pushing the limits of the currently available technology. Brian is also an active speaker, having presented at numerous industry conferences. He is very involved in the game development community, contributing to the Online Games White Paper every year since its inception, as well as helping run the Colorado and now Ottawa IGDA chapters. He has a BS in Computer Science, and an MBA from the University of Denver. He can be reached at brobbins@fuelindustries.com.

- **Elonka Dunin**

  Elonka Dunin is General Manager of Online Community at Simutronics Corp. (http://www.play.net), where she has worked since 1990, doing a little bit of everything, from programming to design to management. Born in Los Angeles, Elonka studied Astronomy at UCLA, and then joined the United States Air Force. She is a world-traveler who speaks several languages, and has visited every continent, including Antarctica. An amateur cryptographer, she has also won considerable acclaim with the cracking of "uncrackable" codes. Elonka is a longtime member of the IGDA and has been attending the Game Developers Conference for more years than she can remember. She can be contacted at elonka@simutronics.com, or via AIM at ScreenName “elonka”.

- **Jeferson Valadares**

  Jeferson Valadares is Creative Director and co-founder of Jynx Playware, a 12-person, 3-year-old game development company. He's also on the organizing committee of the Brazilian Workshop on Games and Digital Entertainment (WJogos) and on the Technical Consultative Advisory Board of the Game Design and Planning Graduation Course at the Universidade Anhembi Morumbi, Brazil. A B.S. and M.Sc. graduate, he does research on Artificial Intelligence applied to Computer Games. Amazingly, he STILL manages to come up with time for his favorite non-gaming related activity: reading.

- **Steve deBenedictis**

  Mr. DeBenedictis is the Manager of Product Strategy for new products at SolidWorks Corporation. He is responsible for planning, developing and marketing new consumer applications focused on 3D graphics functionality in the entertainment, creativity and productivity product categories. "Cosmic Blobs " is the primary offering from this business which is a revolutionary new

3D graphics and animation software program for kids. He has a comprehensive background in product management and business planning with technology development and entertainment media companies including Lycos, Akamai, Atari and 3Com. He has worked extensively in the Internet media, data networking and software development industries creating products and online services for electronic entertainment game play and distribution. He can be reached at steve@bumpsriver.com.

- **Jon Estanislao**

  Jon Estanislao is responsible for business development activities at Activision in the wireless, online/broadband, and interactive television gaming industries. Formerly, he was a strategy manager in the Communications, Media and Entertainment Industry practice of Accenture LLP. He specialized in the interactive entertainment industry and assisted clients, including console manufacturers and software publishers, with online strategies, competitive analysis, market entry, financial analysis, and customer registration. Jon has also been a speaker at interactive entertainment industry events, including GDC. Jon has an MBA from the Anderson School at the University of California, Los Angeles (UCLA), a BS in Business Administration from Georgetown University, and a CPA in the State of California. He can be contacted at jestanislao@activision.com.

- **Greg Mills**

  Greg Mills, Director of Programming , AOL Games. He is currently responsible for the casual business and content strategy for the AOL Games group. In addition, he is responsible for all industry and competitor analysis in the online gaming industry for AOL. Before joining AOL, Greg worked at WorldPlay Entertainment and at the 3DO Company in a variety of marketing and business development positions. Greg earned his M.B.A. from Santa Clara University and a B.A. in Economics from Pomona College.

- **John Welch**

  John Welch is the co-founder and President/CEO of PlayFirst, Inc., a developer and publisher of downloadable, try-before-you-buy games for segments outside of the traditional 18-34 male such as women and families. Mr. Welch is a past Chair of the IGDA Online Games SIG and he has been a contributor to its well-received White Papers. In his previous position as Vice President of Games and Product at AtomShockwave Corp. (1999-2004), Mr. Welch was responsible for driving the product strategy and acquiring and developing games for the Shockwave.com and GameBlast brands. Prior to Shockwave.com, Mr. Welch spent time at Sega (1998-1999) and with a consulting company that he co-founded (1997-1999). He holds Bachelor's and Masters degrees in Computer Science, the former from MIT and the latter from the University of Massachusetts. He can be reached via http://www.playfirst.com.