



## **2006 Casual Games White Paper**

**The IGDA Casual Games SIG**

**[www.igda.org/casual](http://www.igda.org/casual)**

## FORWARD

While much progress has been made over the last several years, the casual games space is still a very nascent space. Every day new players enter the market, new business models are created, and new customers are reached. Through this rapid change and growth, we have been fortunate to have an extremely open and sharing community. Everyone realizes that the market is big enough, and the risks are low enough that freely sharing knowledge and information is in everyone's best interest. Nowhere is this more evident than in the white paper that follows. It contains in-depth information about the market, the business models in use, issues relating to production and design, significant technologies used, and a full survey looking at the various means to bring games to the market.

We have tried something new with this white paper, by creating it on the IGDA's wiki. A wiki allows a community of people to easily contribute to a common resource. You are reading the results of the first step in this process. This paper has been created over the past several months by the community getting updates as minor as simple format updates, and as significant as complete restructuring of entire sections. All of this has been done through a user's web-browser, and became immediately visible to other contributors on the IGDA website. The next phase is just beginning. All of this content will continue to live on the wiki where we hope it can continue to be updated and expanded on by you, the casual game community. As you read this document, if you notice something that is incorrect, or something that you can add to or expand upon, we encourage you to logon to the wiki, and make those corrections so the rest of the community can benefit. You can access the wiki version of the white paper here:

**[http://www.igda.org/wiki/index.php/Casual\\_Games\\_SIG/Whitepaper](http://www.igda.org/wiki/index.php/Casual_Games_SIG/Whitepaper)**

All of the information seen here has been given freely and openly by the casual game community. It is our hope that with this knowledge, everyone will continue to learn and improve or methods for bringing great games to the world. Unlike the traditional "core" game space, our market is practically limitless, and we have only begun to scratch the surface, and by sharing our best practices, we will enable the entire industry to improve and continue to grow our market at the breakneck pace it has been.

While the content has been given freely, **this paper is not free**. If you learn something valuable, we expect you to contribute back to the community in kind. Whether by updating the wiki with your own knowledge and best practices, or simply adding to the discussion on the mailing list, you are expected to contribute. Your cost for accessing this wealth of knowledge is to share your own knowledge back out to the community, and to assist others in the same way you have been assisted.

Finally, we'd like to extend a heart thanks to all of the contributors who have participated in the creation of this resource. It is your willingness to share knowledge that enables us to provide valuable content by and for, the developers in this space.

With best regards,

Margaret Wallace  
Co-Editor, Casual Games White Paper  
Co-Founder/CEO  
Skunk Studios  
margaret@skunkstudios.com

Brian Robbins  
Co-Editor and Chair, IGDA Casual Games SIG  
Director of Games  
Game Trust, Inc.  
brian@gametrust.com

## CREDITS

### White Paper Editors

<b>Co-Editors</b>	<ul style="list-style-type: none"> <li>• <b>Margaret Wallace</b> CEO, Skunk Studios</li> <li>• <b>Brian Robbins</b> Director of Games, Game Trust, Inc.</li> </ul>
<b>Section Editors</b>	<ul style="list-style-type: none"> <li>• <b>Greg Mills (Market Overview)</b> Director of Premium Games, AOL</li> <li>• <b>Gabe Zichermann (Market Overview)</b> VP, Strategy &amp; Communications, Boonty</li> <li>• <b>James Gwertzman (Business Models)</b> Director of Business Development, PopCap Games, Inc.</li> <li>• <b>Juan Gril (Production &amp; Design)</b> Producer, JoJu Games</li> <li>• <b>Peter Glover (Technology)</b> VP, Games &amp; Products, Atom Entertainment, Inc.</li> <li>• <b>Steven DeBenedictis (Publishing)</b> Director of Marketing, SolidWorks Corporation</li> <li>• <b>Steve Meretzky (Skill-based Gaming)</b> Chief Game Designer, Floodgate Entertainment</li> </ul>

### Contributors by Section

<b>Market Overview</b>	<ul style="list-style-type: none"> <li>• <b>Hugh de Loayza</b> Oberon Media</li> <li>• <b>Peter Glover</b> Atom Shockwave</li> <li>• <b>Jay Moore</b> GarageGames</li> <li>• <b>Matt Walsh</b> R/GA</li> </ul>
<b>Business Models</b>	<ul style="list-style-type: none"> <li>• <b>Kris Alexander</b> Sr. Media &amp; Entertainment Service Line Manager, Akamai</li> <li>• <b>Tom Edwards</b> Geographer &amp; Geocultural Consultant, Englobe Inc.</li> <li>• <b>Peter H. Friedman</b> Proprietor, Certified Public Accounting firm</li> <li>• <b>Cassandra Willard, Esq.</b> Partner, Law Firm of Franklin &amp; Willard, Orlando</li> </ul>
<b>Production and Design</b>	<ul style="list-style-type: none"> <li>• <b>Bryan Bouwman</b> HipSoft</li> <li>• <b>Heather Chandler</b> Red Storm Entertainment</li> <li>• <b>Kenny Dinkin</b> VP/Executive Producer, Playfirst</li> </ul>

	<ul style="list-style-type: none"> <li>• <b>John Driscoll</b> Director of Product Development, Slingo</li> <li>• <b>Nicholas Fortugno</b> gameLab</li> <li>• <b>Jane Jensen</b> Oberon Media, Inc.</li> <li>• <b>Eric Lamendola</b> Slingo</li> <li>• <b>Daniel Prigg</b> Real Networks</li> <li>• <b>James C. Smith</b> Reflexive Entertainment</li> <li>• <b>Jim Stern</b> iWin, Inc.</li> <li>• <b>Mike Sweeney</b> Lead Game Designer, Slingo</li> <li>• <b>Michael Sweet</b> Creative Director/Partner, Audiobrain</li> <li>• <b>Jessica Tams</b> SkillJam</li> </ul>
<p><b>Technology</b></p>	<ul style="list-style-type: none"> <li>• <b>Steven B Davis</b> CEO, IT GlobalSecure Inc.</li> <li>• <b>Brad Edelman</b> Chief Technology Officer, PlayFirst, Inc.</li> <li>• <b>Arthur Humphrey</b> CEO and Lead Designer, Last Day of Work, LDW Software, LLC</li> <li>• <b>Andy Phelps</b> Assistant Professor, Rochester Institute of Technology</li> <li>• <b>John Say</b> President, CEO, Say Design, Inc.</li> <li>• <b>Mattias Stridsman</b> Flash Programmer, Say Design, Inc.</li> <li>• <b>Michael Welles</b> Director, Say Design, Inc.</li> </ul>
<p><b>Skill-based Gaming</b></p>	<ul style="list-style-type: none"> <li>• <b>Justin Beckett</b> Founder / former CEO, SkillJam Technologies</li> <li>• <b>Peter Blacklow</b> VP / Chief Marketing Officer, WorldWinner</li> <li>• <b>Brian Mahoney</b> Manager of Customer Experience, WorldWinner</li> <li>• <b>Allison Rynak</b> Director, PR and Communications, WorldWinner</li> </ul>

<b>TABLE OF CONTENTS</b>
--------------------------

<b>FORWARD.....</b>	<b>2</b>
<b>CREDITS.....</b>	<b>3</b>
<b>TABLE OF CONTENTS .....</b>	<b>5</b>
<b>I. INTRODUCTION .....</b>	<b>6</b>
A. BACKGROUND AND PURPOSE .....	6
B. AUDIENCE AND SCOPE .....	6
C. DEFINITIONS .....	6
D. WIKI.....	6
E. DISCLAIMER.....	7
<b>II. MARKET OVERVIEW.....</b>	<b>8</b>
A. INTRODUCTION.....	8
B. WHAT ARE CASUAL GAMES.....	9
C. VIDEO GAME AUDIENCES: hardcore, core and casual gamers .....	9
D. COMMON INDUSTRY TRENDS AND MARKET FORCES .....	11
E. WEB DEMO/DOWNLOADABLE GAMES.....	12
F. WEB AND COMMUNITY-BASED GAMES.....	17
G. SKILL-BASED GAMES .....	19
H. ADVERGAMING.....	26
<b>III. BUSINESS MODELS.....</b>	<b>33</b>
A. INTRODUCTION.....	33
B. UNDERSTANDING THE VALUE CHAIN .....	33
C. BUSINESS MODELS .....	37
D. CONTRACTUAL ISSUES .....	40
E. CONCLUSION .....	43
<b>IV. PRODUCTION AND DESIGN .....</b>	<b>44</b>
A. INTRODUCTION.....	44
B. DESIGN CONSIDERATIONS.....	44
C. PRODUCTION ISSUES .....	68
D. CONCLUSION .....	78
<b>V. TECHNOLOGY OVERVIEW .....</b>	<b>79</b>
A. INTRODUCTION.....	79
B. TECHNICAL CHARACTERISTICS OF CASUAL GAMES.....	79
C. BASE DELIVERY TECHNOLOGIES.....	85
D. SECURITY AND CHEATING.....	93
<b>VI. PUBLISHING .....</b>	<b>102</b>
A. INTRODUCTION.....	102
B. SURVEYING THE COMMUNITY.....	102
<b>CLOSING.....</b>	<b>115</b>
<b>VERSION HISTORY .....</b>	<b>116</b>

## I. Introduction

### A. Background and Purpose

The Casual Games industry is still in its formative years. Numerous players continue to enter the market, business models are constantly changing, and new customers are found every day. This paper has been created to help define the industry, and give everyone a base level of understanding of all aspects of this space. It is not intended to go into minute detail on any one subject, but should provide broad understanding to the major factors involved in the casual game industry.

### B. Audience and Scope

This white paper is specifically focused on providing valuable, hard-to-come-by information to small to medium-sized casual game developers. This white paper should also prove interesting to anyone who wants to learn or read about the casual games industry. It includes a general market overview, discussion on business models, production and design issues, technologies, and a survey on the methods used to bring games to market.

### C. Definitions

While pure definitions are difficult to come by in this space, for the purposes of this paper, we have generally tried to abide by the following definitions:

**Advergame:** A web or downloadable game where the primary objective of building it is to deliver advertising messages, drive traffic to web sites, and build brand awareness.

**Casual Games:** Games that generally involve less complicated game controls and overall complexity in terms of gameplay or investment required to get through game.

**Downloadable game:** A "small file" game, typically less than 25MB, that is downloaded from a web site or peer-to-peer network and installed on a user's computer, where it runs as a standalone executable with or without Internet access. The current business model dictates that these games often have a trial mode, with the option to purchase the full version for unlimited play. Examples of downloadable games can be found on almost every online gaming site or games channel on the major portals. This category does not include demos of video games or PC traditional large-format game titles that are primarily sold through retailers. This category does include titles that are primarily available for download, even if the game is additionally distributed on CD-ROM.

**Hardcore, Core (Traditional) Games:** Games developed for and delivered on a dedicated game console (set-top or handheld) as well as CD-ROM or DVD that generally involve more complicated game controls and overall complexity in terms of gameplay or investment required to get through game.

**Skill game:** A web game played in a tournament format, in which an entry fee is paid to compete and money or prizes are awarded to the most skilled player or players. Elements of luck have either been eliminated or greatly reduced in the game.

**Web game:** A game launched via a web page with no prior installation of software required. This category does not include games that are downloaded to the user's hard-drive and run outside of the web-browser, but it does include games launched from a web page that might require/installation of a general or custom ActiveX control. Common examples of this are the Flash™, Shockwave™ and Java™ games found on thousands of websites, as well as C++ games delivered via a custom ActiveX control.

### D. Wiki

This paper has been developed, and continues to live on the IGDA's wiki. The wiki allows for the community to easily and continuously update and contribute to the content. We encourage you to reference the wiki to see update information, and to update the wiki yourself if there is something you can add or see the need to correct. The full wiki URL for the white paper is:

[http://www.igda.org/wiki/index.php/Casual\\_Games\\_SIG/Whitepaper](http://www.igda.org/wiki/index.php/Casual_Games_SIG/Whitepaper)

### ***E. Disclaimer***

---

This work was created and written by volunteers on behalf of the community at large. The white paper content is based on the individual input of the contributors, and does not necessarily reflect the opinions or policies of the companies at which the individuals work. There may be inaccuracies and information that has become outdated since this white paper was originally written. The information was obtained from publicly available sources, including company websites, company annual reports, SEC filings, news sites dedicated to games and analyst reports (with express permission).

This information is intended strictly for informational purposes. If you include it in a business plan or any business process, you are responsible for its use and any successes or failures resulting from this report.

If your company bio is incorrect, or if you feel we've missed something, we apologize! All content for the paper continues to live on the IGDA wiki, and we encourage you to correct the information there.

Reproduction of this document in whole or part may be done without written approval but must reference this document as a source and display one of the following URLs as the location to obtain the full report: [www.igda.org/casual](http://www.igda.org/casual) or [www.igda.org/wiki/index.php/Casual\\_Games\\_SIG/Whitepaper](http://www.igda.org/wiki/index.php/Casual_Games_SIG/Whitepaper).

Thank you!

## II. Market Overview

### A. Introduction

---

The casual games market has become a big business, attractive to investors and entrepreneurs alike. Over the past few years, casual games have gone from a cottage industry to one of the highest growth areas in the overall video games sector. The casual games industry has grown from almost nothing in 2002 to well over \$600 million in 2004 in the United States (US). By 2008, industry experts anticipate that the market will surpass \$2 billion in the US alone. For the past several years, thousands of casual games have sprung up all over the Internet, as well as on mobile, console, and even in boxes at brick-and-mortar retail outlets.

When you are in the business of casual games, you are reaching virtually all demographic sectors. Women in their forties comprise the typical casual game player – but so do men, teens, kids, college students, seniors and international audiences. Even hard core game players take a break every now and then to play free online poker games and online pool.

In December 1999, Media Metrix reported that 54.1 million people played a PC game, with 36.5 million of the total players playing Solitaire and other Windows-bundled games such as FreeCell. Perhaps this explains why men and women over the age of fifty comprise of the largest age group playing online Bridge on the America Online (AOL) game service at any given point in time. Without a doubt, more people from around the world play casual games than any other kind of video game.

Although Internet access and broadband adoption have played a significant role in the growth of the casual games sector, it's the game industry's embrace of online distribution for casual small file web and downloadable games that has been the major driver. By leveraging the Internet as a unique platform for the creation, monetization and promotion of gaming content, the industry has created a new channel that both drives and is driven by consumer demand. Today, casual games are among the stickiest, most sought-after content online, and major game sites routinely top Internet site rankings. Interestingly, the major form of online game content is no longer game reviews, but it's the games themselves. Whether played in a web browser or as a fully featured download, casual gaming has moved well beyond the realm of niche into the mainstream.

Additionally, the ease of online distribution has made casual games eminently accessible to non-traditional gaming audiences, particularly female players who may have never even considered themselves "gamers" per se. This overall accessibility, combined with uncomplicated gameplay and risk-free business models such as try-before-you-buy, has been the key to the success of the casual games industry.

Despite the ubiquity of casual games on a variety of gaming platforms, however, this white paper mainly focuses on casual games found on PC and Mac platforms that are played as online or downloadable games. For more information about the presence of casual games on mobile gaming platforms, please refer to the International Game Developer Association's (IGDA) 2005 Mobile Games Whitepaper ([www.igda.org/online/IGDA\\_Mobile\\_Whitepaper\\_2005.pdf](http://www.igda.org/online/IGDA_Mobile_Whitepaper_2005.pdf)) or refer to the IGDA's Mobile Games Special Interest Group (SIG) website ([www.igda.org/mobile](http://www.igda.org/mobile)).



## **B. What Are Casual Games**

---

The term "casual games" is used to describe games that are easy to learn, utilize simple controls and aspire to forgiving gameplay. Without a doubt, the term "casual games" is sometimes an awkward and ill-fitting term – perhaps best described as games for everyone. Additionally, the term "casual" doesn't accurately depict that these games can be quite addictive, often delivering hours of entertainment similar to that provided by more traditional console games. To be sure, there is nothing "casual" about the level of loyalty, commitment and enjoyment displayed by many avid casual game players – just as there is nothing "casual" about the market opportunity and market demand for these games.

## **C. Video Game Audiences: Hardcore, Core and Casual Gamers**

---

A recent AP-AOL study indicates that forty percent (40%) of all people in the United States turn off their televisions in favor of playing some type of video game (Source: "Poll 4 in 10 Americans Play Video Games" ([www.mercurynews.com/mld/mercurynews/business/technology/14526158.htm](http://www.mercurynews.com/mld/mercurynews/business/technology/14526158.htm), May 8, 2006). It is estimated that almost one hundred million people in the US will play a computer game this year alone and yet not all gamers are alike. Within the games industry, players are often assigned to one of three broad (and overlapping) categories:

**Hardcore Gamer:** Gamers who typically play high-action, extremely competitive games that require a greater degree of involvement or dexterity in order to progress.

**Core Gamer:** Gamers who typically play games with a steeper learning curve or games that require some level of deeper involvement or complex tactical challenges.

**Casual Gamer:** Gamers who play games for enjoyment and relaxation rather than games with steep learning curves or requiring high levels of commitment or involvement.

While these audience segments are not mutually exclusive, these distinctions are useful for understanding the different types of video game experiences and play patterns that typify each group.

### **1. Characteristics of Casual Gamers**

For the purposes of this whitepaper, we are focusing on casual game audiences. To some extent, though, the typical casual gamer is hard to define, especially since casual games are played by so many kinds of people from around the world. Despite the limitations imposed by these categories, here are ways in which casual games audiences are generally conceived of in terms of demographics, their favorite genres, and what's understood as their typical gaming patterns:

#### **Demographics**

Generally skews older (35+), though casual game players can be found among college students, teens, school-aged children as well as seniors.

While the gender break-down of casual game players can vary greatly from genre-to-genre and even from game-to-game, the largest audience remains women aged thirty-five to fifty.

The proliferation and popularity of casual games has greatly contributed to an explosion of women on the video game scene. Women comprise forty-three percent of all video gamers, according to a 2005 survey conducted by the Entertainment Software Association, up from

thirty-eight percent in 2003 (Source: Pioneer Press ([www.twincities.com/mld/twincities/living/14502122.htm](http://www.twincities.com/mld/twincities/living/14502122.htm)), May 7, 2006).

Furthermore, a recent study conducted by the Consumer Electronics Association (CEA) puts forth that there are more women gamers than males in the twenty-five to thirty-four age range, with the average age being around thirty years. In this age bracket they say sixty-five per cent of women play video games compared to only thirty-five percent of men. (Source: New York Times ([www.nytimes.com/2006/04/17/technology/17drill.html?ex=1147233600&en=4ee4dae89714ecea&ei=5070](http://www.nytimes.com/2006/04/17/technology/17drill.html?ex=1147233600&en=4ee4dae89714ecea&ei=5070)), April 17, 2006.)

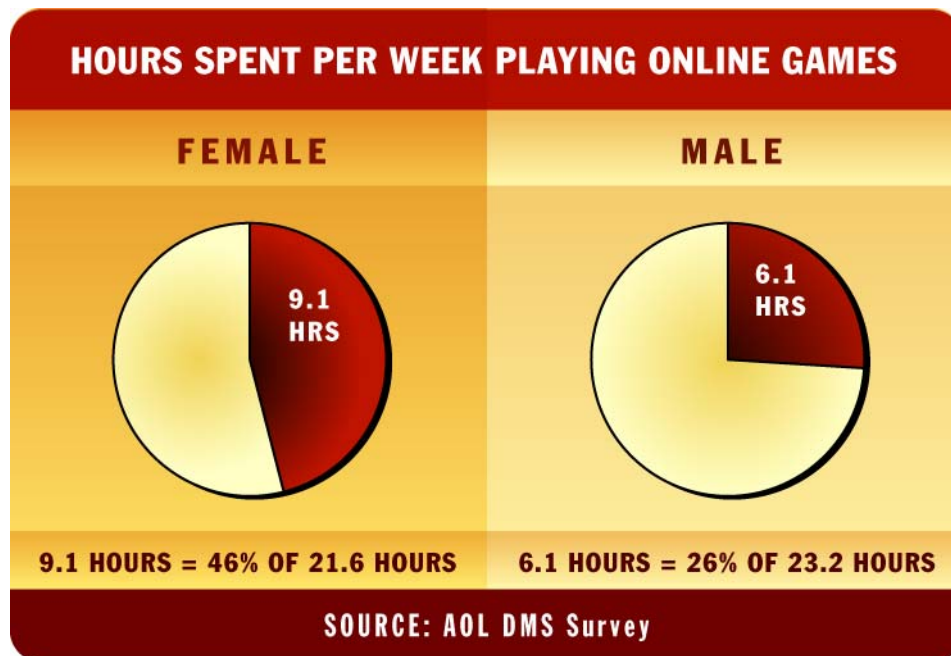
While the typical core gaming audience is male and aged eighteen to thirty-four, casual gamers tend to be both women and men between the ages of thirty-five and sixty-five, with a slight demographic skew towards women.

Gradually, the number of men playing casual games is increasing, but today's market reality requires a focus on female players.

Women represent the largest category of consumers for these games, although they may also be buying these games for their husband, children, household or even as gifts.

### Typical Gaming Patterns

The play patterns of each audience segment vary dramatically, with some female players over the age of forty spending upwards of nine hours per week playing online games. These long online play periods are often divided into small game sessions.



Pick up and drop games multiple times per day.

Relaxation, diversion, socialization and community are key drivers for casual game play as opposed to overt competition.

This average play time is much lower than hardcore gamers' title play commitment near release, but is much higher than the mass-market retail gamers' average play time.

## Favorite Genres

- Puzzle Games
- Card or Solitaire
- Light System Management
- Casual Action

## Primary Points of Access

- Downloadable PC & Mac versions to play offline
- Online play
- Increasingly other gaming platforms (e.g., mobile, console)
- A wide variety of users now see the Internet as a primary entertainment medium, and casual games comprise a main staple of the entertainment value of the Internet. As the audience for online content grows, so does the amount of money spent in the space.

## Responding to Audience Needs and Demands

Keeping the play and consumption patterns of this worldwide mass audience in mind, successful casual games must:

- Seem accessible to players with varying levels of familiarity and dexterity with computer controls;
- Engage players who may not be familiar with various game genres
- Attract players by offering easy-to-learn games that are inviting and generally non-violent;
- Interact with players who are accustomed to user interface conventions from the traditional retail market.

Clearly, growth in the market shows that gaming is no longer reserved for traditional gaming audiences and that it is only a matter of time before the right casual games spark interest for the mature consumer at large, both men and women alike. When this happens, casual games will truly move from mass market potential to mainstream reality.

### ***D. Common Industry Trends and Market Forces***

---

While there is no disagreement about the tremendous potential of the casual games industry as a whole, its component segments have widely differing growth rates, accelerators, inhibitors and key participants.

The following table describes some of the key market forces at play that impact all of these game types one way or another:

Market Force	Impact	Explanation
Increased Investment in Casual Games	High	Publishers, Retailers, Aggregators and Developers have significantly increased their investments in the space over the past few years. A large number of new entrants into this space, including traditional console makers, are driving the market forward. Costs for production also continue to rise.
Consumers Spend More Time Online	High	There is a continuing shift of consumers who spend more time online as compared with other forms of passive entertainment, including watching television or listening to music. Increased broadband penetration gives players easier access to casual game content.
Consumers Increase their Comfort with Gaming	Moderate	As non-gamer consumers increase their gameplay capabilities, they become progressively more comfortable with gaming, increasing the future likelihood that they will try a gaming product from a trusted source.
Developers Push Forward New Genres or Themes	Moderate	The development of new and unique game types, modes of play and genres will increase the addressable market opportunity. Examples include adult or mature themed titles (Leisure Suit Larry) and Core/Casual combinations (Ricochet).
Core-Casual Crossover	Moderate	Increased game availability, regardless of genre, raises consumer awareness and willingness to try games online. This helps attract new players to the casual games space.

The following sections will explore the definitions, trends, accelerators, inhibitors and future potential of each vertical profiled in this White Paper, namely:

- Core and Casual Downloadable Games & accompanying Web Versions
- Web and Community-based games (games played only online)
- Skill-Based games
- Advergaming

## ***E. Web Demo/Downloadable Games***

---

### **1. Overview**

The core and casual downloadable games sector grew over one hundred percent (100%) from 2003-2004, reaching over \$115 million in North America alone (Source: US Online PC Gaming Forecast & Analysis, 2004-2008: Growth Continues. IDC, Schelley Olhava, Dec 2004). The industry caters primarily to a mass-market audience of consumers that have little overlap with traditional game buyers. Although numbers for core and casual downloads are not separated in the IDC forecast, most industry experts believe casual games comprise the majority of this revenue today. An explosion in the number of titles, distribution points and marketing investment is driving growth in this sector. The casual game audience is relatively untapped both in terms of potential consumers and volume of downloads.

The downloadable games market is a North America-centric business today with less than twenty-five percent (25%) of revenues coming from other territories. Global sales are increasing quickly with top English-language games selling well internationally – even to non-English speaking consumers – as a result of the games’ more casual and intuitive play pattern. Additionally, leading channels are increasingly focused on developing localized distribution channels and pursuing localized content, which will be a key revenue driver in 2006 and beyond as evident by the recent purchase of Zylom in Europe by RealNetworks.

Downloadable games are typically offered in try-before-you-buy (TBYB) mode and/or as part of a subscription service, often with a free web game demo that is available to provide cross-marketing to the TBYB version and/or as an advertising revenue generator in its own right. The free online version usually offers consumers unlimited game play but with a restricted feature set, while downloadables typically offer the entire feature set but only one hour or less of trial play, forcing users to purchase the game for unlimited, offline gameplay.

Distribution of these TBYB games is carried out by major portals filling the “retail” role (Yahoo! Games, RealArcade, MSN Games, Shockwave.com, AOL, and Pogo), developer sites (WildTangent, PopCap, GameHouse) and aggregators providing game channels to numerous third-party websites (Oberon Media, TryMedia/Macrovision, Boonty). Core games are distributed mostly by major gaming portals (Gamespy/IGN, Gamespot), Peer-to-Peer Networks (KaZaa, Grokster) and distributors (TryMedia/Macrovision, RegNow).

The casual download market was historically devoid of publishers, with game developers distributing directly to the portal “retailers”. While developers still distribute directly to these portals, recent growth in the market is fueling the emergence of publishing entities -- enabling more successful developers to even start publishing third party titles as well. The upside to this trend is that this new breed of publishers provides industry experience, funding and distribution to content developers. However, developers who work through publishers will often lose control over the distribution and ownership of their intellectual property (IP) creations – something they’ve traditionally held onto up to this point.

A full-service publisher should ideally provide technology infrastructure, focus testing, quality assurance testing, and marketing and PR services for its titles/developers. We expect this trend towards the emergence of publishing entities to continue as the market matures and conditions become less friendly to new developers self-funding and self-distributing their content.

The core and hardcore downloadable market is dominated by the major traditional publishers, such as Activision and Electronic Arts (EA), with some notable exceptions. Independent, large developers have had some recent successes in self-distribution, including the breakthrough of Valve’s Half-Life 2 and Team17’s Worms. Online core game publishers like GarageGames are also evangelizing the opportunity for independent developers to produce games explicitly in the downloadable format. Although there is some disagreement, most industry observers expect a healthy online market to develop for both niche core games and franchise add-ons that will bypass the retail channel completely. The greater penetration of broadband and faster connection speeds will greatly reduce download wait times and increase consumer acceptance.

With relatively low development costs ranging from under \$50k to over \$200k (and rising), developers face a unique set of opportunities and threats in the casual games market. On one hand, low development costs improve the chance of a positive return on investment. On the other hand, low development costs increase the volume of new titles entering the

market, resulting in greater royalty and distribution competition. In contrast, the development requirements for core games are almost identical whether retail or online distribution is being used.

## 2. Casual Genres

The top-selling casual genres today are Puzzle, Mah Jong, Sudoku, Casual-Action games, Word games, plus Card & Board games. “Fun” and “accessible” are key attributes that determine market potential for casual games. However, it is important to read the genre list with the understanding that there is no set common definition in the mind of the casual game consumer. The casual nature of the games provides a huge opportunity for developers to innovate and explore with new creative styles and types. Some representative examples of downloadable casual games include:

Game Genre	Examples
Puzzle	Magic Match, <a href="http://www.oberongames.com">www.oberongames.com</a> (Oberon Media/Codeminion) Mystery Case Files: Huntsville, <a href="http://www.bigfishgames.com">www.bigfishgames.com</a> (Big Fish Games) Tropix, <a href="http://www.gamehouse.com">www.gamehouse.com</a> (GameHouse/Robot Super Game)
Mah Jong	Mah Jong Adventures, <a href="http://www.skunkstudios.com">www.skunkstudios.com</a> (Skunk Studios) Mahjong Escape – Ancient China, <a href="http://www.mahjongescape.com/">www.mahjongescape.com/</a> (Playtime) Mah Jong Quest, <a href="http://www.iwin.com">www.iwin.com</a> (iWin)
Word Games	Big Kahuna Words, <a href="http://www.reflexive.net">www.reflexive.net</a> (Reflexive) Pat Sajak’s Lucky Letters, <a href="http://www.playtonium.com">www.playtonium.com</a> (Adveractive/Playtonium) Super Wild Wild Words, <a href="http://www.gamehouse.com">www.gamehouse.com</a> (GameHouse)
Casual-Action	Cake Mania, <a href="http://www.sandlotgames.com">www.sandlotgames.com</a> (Sandlot Games) Diner Dash, <a href="http://www.playfirst.com">www.playfirst.com</a> (PlayFirst/ gameLab) Feeding Frenzy 2, <a href="http://www.popcap.com">www.popcap.com</a> (PopCap)
Card & Board	Ancient TriPeaks, <a href="http://www.toybox-games.com">www.toybox-games.com</a> (Toy Box Games) Hotel Solitaire, <a href="http://www.zylom.com">www.zylom.com</a> (Zylom) Sudoku, <a href="http://www.gamedesire.eu">www.gamedesire.eu</a> (GameDesire)

At this point in time, most casual games are consumed as either downloadable or online games. Downloadable games are principally single-player experiences today. The free web demo versions that upsell the premium downloadable games are also usually single player experiences that offer very limited replay value. The goal of the free web demo is to upsell a player to download the offline version of the game, though they often generate significant ad revenues in their own right.

Web demo games are not built for evergreen online game play. This is very evident by comparing simultaneous game player numbers on online gaming services that offer web demo games and online games that are built for true online game play experiences like Poker, Hearts and Yahoo! Games Pool. Some companies have added community elements to web demo games such as high scores, chat, and award challenges in order to try to increase web demo game usage and try to further monetize the web demo game time with advertising and sponsorships. It will be interesting to see if downloadable casual games start offering multiplayer experiences and more robust online game play like true online single and multiplayer games that are built for online play as its main focus.

### **3. Business Model Considerations**

Try-Before-You-Buy is a central characteristic of downloadable games, although 2005 has seen a resurgence around advertising revenues generated through online game play. To a large extent, however, developers and distributors of games make money only when offline versions of their games sell. While on the surface this is no different than packaged goods sold at retail, the reality is that marketing and promotion of casual games only drives free trial downloads – and this is associated with some bandwidth cost. Therefore, it is the sole job of the game during the trial period to create an experience that is rewarding enough to drive a purchase or conversion to sale. As mentioned in the preceding section, companies are trying to integrate advertising into the web demo as well as the free trial period of the downloadable game in an effort to monetize free game play time, however developers often do not benefit directly from the generation of these ad revenues. A movement is underfoot in the casual games industry to get online retailers to share their advertising revenues with publishers and developers in order to support the creation and further monetization of their game brands.

The core gaming market offers PC downloadable games that most people are familiar with because PC game publishers spend millions of dollars advertising and building their brands. The challenge is that PC publishers to date have largely used online distribution to generate incremental revenue from PC box games that have been in the retail channel for at least several months. This trend is, however, changing. By contrast, the majority of casual downloadable games have no brand recognition, and developers lack the resources to spend significantly on advertising or marketing. Therefore, those casual developers without a strong web presence in their own right are very dependent on portals and gaming sites to promote their games.

Viral marketing can also help online titles succeed. Whether it's by word-of-mouth or explicit viral features, the pass-along rate of a title can have a significant impact on its revenue. A highly viral title may experience anywhere from seven (7%) to seventeen (17%) percent pass-along sales under ideal conditions. This number will be highest if viral marketing is built into the game, peer-to-peer distribution is used, and the anti-piracy technology allows for revert-to-trial capabilities.

Industry conversion rates to sales vary dramatically from channel to channel, and region to region but a good rule of thumb for an average game is around a one percent (1%) conversion from download to purchase, with top performing games converting at significantly higher rates during different periods of a game's lifecycle. Although the majority of game sales tend to take place in the first twelve to sixteen weeks after a game has launched, mass-market, and especially casual games, tend to have more "evergreen" sales patterns, often generating significant revenue for years. In the following sections of the white paper content strategies will be outlined to improve overall conversion rates.

## **4. Key Players in the Casual Games Space**

### *4.1 Major Online Retailers*

AOL Games: [aolsvc.aol.com/onlinegames](http://aolsvc.aol.com/onlinegames)

Big Fish Games: [www.bigfishgames.com](http://www.bigfishgames.com)

MiniClip: [www.miniclip.com](http://www.miniclip.com)

MSN Games: [zone.msn.com](http://zone.msn.com)

Pogo: [www.pogo.com](http://www.pogo.com)

RealArcade: [www.realarcade.com](http://www.realarcade.com)

Shockwave.com: [www.shockwave.com](http://www.shockwave.com)

WildTangent: [www.wildtangent.com](http://www.wildtangent.com)

Yahoo! Games [games.yahoo.com](http://games.yahoo.com)

### *4.2 Aggregators Providing Retail Game Channels to 3rd Party Web Sites*

Boonty: [www.boonty.com](http://www.boonty.com)

Oberon Media: [www.oberongames.com](http://www.oberongames.com)

Spill Group: [www.spillgroup.com](http://www.spillgroup.com)

Trymedia/Macrovision: [www.trymedia.com](http://www.trymedia.com)

Zylom: [www.zylom.com](http://www.zylom.com)

### *4.3 Publishers*

GameHouse: [www.gamehouse.com](http://www.gamehouse.com)

GarageGames: [www.garagegames.com](http://www.garagegames.com)

iWin: [www.iwin.com](http://www.iwin.com)

MumboJumbo: [www.mumbojumbo.com](http://www.mumbojumbo.com)

Oberon Media: [www.oberongames.com](http://www.oberongames.com)

PlayFirst: [www.playfirst.com](http://www.playfirst.com)

PopCap: [www.popcap.com](http://www.popcap.com)

Reflexive: [www.reflexive.net](http://www.reflexive.net)

### *4.4 Developers*

GameHouse: [www.gamehouse.com](http://www.gamehouse.com)

gameLab: [www.gamelab.com](http://www.gamelab.com)

Hipsoft: [www.hipsoft.com](http://www.hipsoft.com)

LargeAnimal: [www.largeanimal.com](http://www.largeanimal.com)

MumboJumbo: [www.mumbojumbo.com](http://www.mumbojumbo.com)

PopCap: [www.popcap.com](http://www.popcap.com)

Reflexive: [www.reflexive.net](http://www.reflexive.net)

Sandlot Games: [www.sandlotgames.com](http://www.sandlotgames.com)

Skunk Studios: [www.skunkstudios.com](http://www.skunkstudios.com)



## **F. Web and Community-based Games**

---

### **1. Overview**

Web games are games that can be executed from within a web browser without the need of external (.exe) installers. Common examples include Flash, Shockwave and Java games found on many game websites, as well as custom C++ games delivered via an ActiveX control.

Community-based games can be loosely defined as those game experiences where the game is built for true online game play and can be both single- and multi-player games. Because portals are aggregators of audiences, community becomes an important way for users to define themselves and for portals to define and target their services accordingly. It is also widely understood that community elements within games are an extremely powerful retention tool: audiences feel that they have an investment in their community personae. Unlike web demo games that upsell downloadable games, web and community online games not only offer community elements integrated into the game but the games themselves are also robust and offer long periods of game play. Single-player web demo games, in contrast, tend to have low game duration and retention rates.

There are a number of business models that rely on web games, many of which are complimentary to each other and will be discussed in greater detail in the business model section:

Game Type	Description
Advertising	The most common examples are web games deployed on destination portals surrounded by advertising units. These units are either sold by a company's direct sales team or purchased from an advertising network. Revenue increases with a game's traffic popularity, the surrounding units' sell-through rates and the units' market values.
Sponsorship	These are games developed with deep integration of sponsor branding. Sponsored web games are deployed either on destination portals or on the sponsor's websites. Some of these sponsor websites have enough traffic and game selection to be considered a portal in their own right. Some developers offer turnkey solutions that include games, website design, hosting and management. For more information, see the Advergaming section.
Subscription	Many community-based gaming services generate significant revenue from end-user subscription fees, usually in the \$5-10/month range.

IDC estimates advertising and sponsorship revenues in online gaming at over \$200M in 2004, growing to \$500M by 2008. Although web games do not account for all this revenue, a significant portion can be attributed to the traffic generated by web games.

### **2. Top Genres**

Web game genres vary as widely as every other game platform. And like other platforms, genre popularity varies between age and gender groups.

Action / Arcade, Sports, Strategy, Role-Playing Games (RPG) all tend to skew to a younger and more male audience.

Board / Card, Casino, Puzzle, Action Puzzle, Word typically command an older but gender equal player base.

Certain Puzzle and Action Puzzle titles have mostly female audiences. All of the above genres can be created as multiplayer, but today this is typically only seen with Card/Board, Word, and Casino games.

### **3. Characteristics of Successful Games**

Features that are typically associated with community-based gaming include: chat, points based rewards systems, prizing, persistence features, tournaments, ladders, message boards and friend/buddy lists.

The types of games typically found on community gaming services range from single-player to group and competitive games with persistent identity. Most services also offer cash competition options, allowing users to play a ladder or tournament for prizes, otherwise known as skill-based gaming. On the whole, most community gaming services focus on casual games and generate their revenue from subscription fees and advertising. Although not totally dissimilar to persistent-world MMOGs, community-based games differ in that their focus is almost exclusively on casual games/gamers and the game play experience is built for quick and short game durations. The games exist primarily online – however there are various examples of sites with minimal downloadable/retail product crossovers that include things such as prizes and points based rewards systems.

Many portals develop their own community-based game titles, although they are often based on existing game concepts such as Hearts, Backgammon, Mah Jong and Go. These simple titles, typically delivered as Java applets (but increasingly moving away from Java and towards Flash), provide little room for third-party developers to bring licensed content to the experience because they are more expensive. However, there is increasing potential demand for such applications.

Because the major portals are also key distributors for downloadable, skill-based and ad-supported games, there is some inherent conflict in the revenue mix they pursue. Community-based games, with their internal development, sunk costs and recurring revenues are a huge attraction for most portals. One other added advantage to sites including community-based elements is that they can add significantly to the amount of time users spend on the site. For example Jupiter/Media Metrix reports that in 2004, casual gaming portals had the highest time-spent per user online of any type of site. As portals increase the amount of time spent on their site, they also increase the opportunity for users to subscribe, download or view advertising – all of which increase their revenues.

However, recent trends suggest that more and more portals are moving toward web content that supports downloadable games without the inclusion of community elements. This trend is commoditizing sites that do not invest heavily in community-based games. There are also instances where portals will ask for joint development of community-based web games in support of downloadable offerings. The advantage of this model is that developers can leverage users' familiarity with existing community-based games to increase the sales of the corresponding downloadable. The development of community-based titles typically requires the integration of APIs that will allow the game to integrate fully with the community services offered on its intended distribution sites. Providing such services and APIs is seen as the key way for portals to maintain their unique identities and to continue to add value as development shifts more and more to outside developers and publishers.

#### 4. Key Players

The following list represents the most highly trafficked websites that publish web and community-based games. Many of the sites listed also offer other game content (casual downloads, game reviews and other editorial, console previews, cut-scene movies, etc.).

Addictinggames.com ([www.addictinggames.com](http://www.addictinggames.com))  
AOL Games ([aolsvc.aol.com/onlinegames](http://aolsvc.aol.com/onlinegames))  
Coffeebreakarcade ([www.coffeebreakarcade.com](http://www.coffeebreakarcade.com))  
Freearcade.com ([www.freearcade.com](http://www.freearcade.com))  
GameDesire.com ([www.gamedesire.com](http://www.gamedesire.com))  
GameHouse ([www.gamehouse.com](http://www.gamehouse.com))  
Games.com ([www.games.com](http://www.games.com))  
Grab.com ([www.grab.com](http://www.grab.com))  
iWin ([www.iwin.com](http://www.iwin.com))  
iWon Games ([games.iwon.com](http://games.iwon.com))  
Jigzone ([www.jigzone.com](http://www.jigzone.com))  
KewlBox ([www.kewlbox.com](http://www.kewlbox.com))  
Kraft Entertainment ([www.candystand.com](http://www.candystand.com))  
Lycos Network Gamesville ([www.gamesville.com](http://www.gamesville.com))  
MiniClip ([www.miniclip.com](http://www.miniclip.com))  
MSN Games ([zone.msn.com](http://zone.msn.com))  
My Way Games ([games.myway.com](http://games.myway.com))  
Pogo.com ([www.pogo.com](http://www.pogo.com))  
PopCap ([www.popcap.com](http://www.popcap.com))  
RealArcade ([games.realarcade.com](http://games.realarcade.com))  
Runescape ([www.runescape.com](http://www.runescape.com))  
Sandboxer.com ([www.sandboxer.com](http://www.sandboxer.com))  
Skilljam.com ([www.skilljam.com](http://www.skilljam.com))  
Shockwave.com ([www.shockwave.com](http://www.shockwave.com))  
Slingo ([www.slingo.com](http://www.slingo.com))  
Trygames ([www.trygames.com](http://www.trygames.com))  
Uproar Network ([www.uproar.com](http://www.uproar.com))  
Yahoo! Games ([games.yahoo.com](http://games.yahoo.com))

#### G. Skill-Based Games

---

##### 1. Overview

The skill-based gaming industry has had about seven years of operating history, led mostly by privately owned firms striving to achieve or maintain profitability in a competitive

market. FUN Technologies, who is 51% owned by Liberty Media, was already the owner of SkillJam, one of the two leading U.S.-based skill game sites. During 2006, FUN acquired WorldWinner, the other U.S. leader, and is in the process of merging SkillJam and WorldWinner. King.com, a private company based in the United Kingdom, is the skill-based gaming leader in Europe. The U.S. and European companies are entering each other's territories of dominance to try to be the worldwide leader.

In this context, a "skill-based game" is a web game played in a tournament format, with each player paying a cash entry fee to play, and with a cash or merchandise prize going to the winner or winners of the tournament. Skill-based games are named accordingly because the outcome of each competition is based on the players' ability and performance, with any elements of luck either eliminated or greatly reduced. This is critical, in order to be legal and avoid falling under anti-gambling statutes.

The laws governing skill-based gaming in the U.S. are on a state-level; there is no applicable federal statute. Even with scoring based on factors of skill rather than luck, skill-based games are not legal in every state. The exact states where skill-based gaming is proscribed vary from provider to provider, based on the readings of each state's law by their respective legal counsels. The most significant states where skill-based gaming is not permitted, on which all skill game providers appear to agree, are: Arizona, Florida, Iowa, Louisiana and Maryland.

Although there is nothing inherent about skill-based gaming that makes it applicable to casual games but not applicable to non-casual games, to date all economically successful skill game sites have been those that offer casual games as the basis of their tournament offerings. There have been several companies that have attempted to use the same economic model with more hardcore offerings, such as first-person shooters, but have failed and subsequently shut down.

There is a dearth of accurate market data publicized on this nascent industry. As is typical of most early stage market assessments, the limited amount of publicized skill gaming market data has generally been restricted to "top down" macro economic extrapolations. Although this document contains macro economic research, it also includes "bottoms up" micro economic estimates based on empirical operating information.

As a sub-category of the overall online gaming sector, the skill-based gaming industry will likely experience similar growth. In 2004, there were roughly 20-30 companies that could be categorized as skill-based game providers. Today, that number has decreased to fewer than a dozen companies, which generated roughly \$100 million in revenue in 2005. With a compound annual growth rate of between 25 and 35 percent, skill game tournament entry fees should exceed \$1 billion by the year 2009.

Although still not a household term, skill gaming continues to experience increasing consumer awareness. Arguably the biggest contributors to this increased awareness have been portals that have developed an appreciation for the skill gaming business model and transcended their initial regulatory concerns. Currently, the network of portals partnering with at least one skill game service provider includes Yahoo, AOL, Pogo, MSN, Lycos, RealNetworks, BoxerJam, Shockwave and Miniclip.

Further contributions to consumer awareness of skill gaming are expected to come from the large online casino and sports book operators who view skill gaming as a logical (and more

regulation-friendly) extension of their core business. Additional consumer awareness will be derived from the increasing number of mobile and iTV skill game applications.

## **2. Characteristics of a Successful Skill-Based Game**

The characteristics of a skill-based game include:

- Outcome not based on luck
- Cheat-resistant
- Short-playing time
- Highly replayable
- Narrow scoring ranges

These characteristics are described in-depth below:

### **Outcome not based on luck**

In order to be legal in most states, a player's score in a skill-based game must be substantially based on skill, and not on factors such as luck and randomness. Skill-based game creators have employed a variety of techniques to turn otherwise luck-based games into skill-based games. Example: in a game where the initial setup of the game would have a huge impact on the scoring – such as the shuffle of cards in a card game – all players in a tournament could receive the same shuffle.

### **Cheat-resistant**

Because money is at stake based on the result of a tournament, the impetus to cheat in skill-based games is far higher than in other online game offerings. Games offered by successful skill-based game providers need to be designed and implemented in ways that make it impossible for even fairly technically-savvy players to cheat. Companies employ a variety of mostly proprietary techniques to foil and/or catch cheaters, which are discussed in more detail in the Technology Overview section of the whitepaper.

### **Short-playing time**

Experience has shown that games with a short playing time have generated the most money for skill-based game providers. The reason is that, like slot machine players in a casino, skill-based game revenues come from a relatively small number of players (relative to the entire population of casual game players on the web) who tend to play repeatedly. A short game length means a greater number of tournament entries per unit time. For example, chess – which seems like it should be a natural for skill-based gaming – has not been economically successful, in part because of its fairly long game length. Furthermore, skill-based game players are predominantly women, more than half between the ages of twenty-five and fifty-four. Given these players lead busy lives, and many of them are moms with even less free time, they enjoy games that can be played in short ten to fifteen minute bites, or even less.

### **Highly replayable**

Again, because skill-based games need to be played many times by each player to be commercially successful, games whose design and/or amount of content makes them highly-replayable without starting to seem repetitious have an advantage. Examples of highly replayable games would be a solitaire, in which the millions of potential card shuffles make the game different with each play; a Boggle-style game, in which the layout of the letters adds variety from play to play; or a jigsaw puzzle game, in which a large supply of different shaped pieces and images keeps players playing.

### **Narrow scoring range**

Players are mostly likely to keep playing in skill-based tournaments when they win, or at least lose by a small margin. So devising a scoring system for a game that keeps a fairly narrow margin between the most-skilled and least-skilled players will promote replayability and avoid discouraging newer, less-skilled players.

### **3. Player Ranking**

In order to give less-skilled players a chance to win, successful skill game sites employ a ranking system of some type to match players of similar skill level into tournaments. Thus, very skilled players play only against other very skilled players, players of moderate skill play only against other players of moderate skill, and so forth.

To be legal, the outcome of online cash competitions must be based primarily on skill and not luck. Thus, a player's score range in a certain game will tend to fall within a narrow band. If all players were allowed to compete in the same competitions, a handful of the best players would win every challenge. This would discourage the losing players from playing on the site, which would, in turn, hurt even the best players by narrowing the field of potential competitors. However, the task is a lot harder than it sounds.

One issue is how to measure the skill of a player. If it is done by average score, a player can "game" the ranking system by deliberately scoring low in low-prize tournaments in order to be able to easily win high-prize tournaments. Thus ranking systems are based on complex formulas that take into account many factors, such as win-ratios, lifetime winnings, average scores, and number of cash competitions played.

Another issue is how to handle new players. If new players start out at a middle rank, but are not very good, they will lose all of their initial competitions, get discouraged, and leave before the site has collected enough information to give them a more suitable rank. But if new players start out at a low rank, and are already skillful in the given game, they will "eat up" the competition until their rank adjusts to a more suitable level. The typical way to deal with this problem is to make a new player's rank much more volatile than a veteran player's rank, in order to move them more quickly to a suitable level. Another strategy is to lock beginner players out of high-prize tournaments, until their rank has been more solidly ascertained.

A final issue is how to use a rank once it is established. A rank can be used as merely a guideline, in an environment where players choose their own opponents, such as in a lobby-based matching system. Or, a rank can be used invisibly to funnel players into competitions of similarly-ranked players. But what does "similarly-ranked" mean? If defined too narrowly, there are too few players to form a competitive community. If defined too loosely, an unfair environment exists wherein certain players have a slim chance of winning. This is a delicate balance which requires constant monitoring and hard decisions.

Ranking systems work best when there are a lot of players being matched and a lot of tournaments in which to match them. This allows skill game providers to create narrow margins of skill within a given tournament, increasing the percentage of time that a lower-skilled player can win. Thus sites with high traffic volume will inherently be able to provide a more "fair" experience for their players.

### **4. Types of Online Skill Game Competitions**

#### **4.1 Limited Entry competitions (also called "Fixed Entry" or "Matched" competitions)**

The identifying characteristic of a Limited Entry competition is that it contains a pre-established number of competitors; generally between two and 10 players. The competition ends when the final player signs up. Thus, while the number of players and the final prize (or prizes) are known in advance, the ending time is not. A Limited Entry competition might remain open for only a few minutes, or for days, depending on the

number of players on the site, and the popularity of the game in question. Players rarely enter a Limited Entry competition multiple times.

#### *4.2 Progressive competitions (also called "Progressive Prize" or "Progressive Cash" competitions)*

These are tournaments where the ending time is set in advance, but there are an unlimited number of participants. Players are encouraged to enter multiple times, and generally do. Progressives generally last for a period ranging from one day to one week. The prize (or prizes) rise with each entry, so while the end time is known, the final number of players and the final prize(s) are not. Progressives in popular games frequently attract hundreds or even thousands of participants.

#### *4.3 Unlimited Entry competitions (also called "Guaranteed Jackpot" or "Guaranteed Cash Pot" competitions)*

These tournaments are similar to progressives, in that the ending time is established in advance, but the number of entrants is not. The difference is that the prize (or prizes) is fixed, rather than rising as the number of entrants rises. The prize is often in the form of merchandise rather than cash, such as consumer electronics, jewelry, gift certificates or travel packages.

#### *4.4 Head-to-Head Challenges*

These are two-player matches played in real-time, generally in a turn-based game, such as chess or pool. Players select their opponents via some type of game lobby. If a ranking system is used, it is merely as a guide to help players choose their opponents, rather than a dictatorial matching system.

#### *4.5 Ladders*

A ladder is a permanent, or at least ongoing, competition composed of many sub-competitions. Players are arranged on the ladder from best to worst, based either on a preliminary "seeding tournament" or a metric such as rank or lifetime winnings in that particular game. Once qualified, a player can challenge another player who is above them on the ladder. Typically they are only permitted to challenge those who are several "rungs" above them. If the challenging player wins, or if the challenge is ignored, the two players swap positions on the ladder. If the challenger loses, both players retain their positions. Thus, players attempt to battle their way to the top of the ladder. In addition to the prizes for each individual ladder challenge, there may be prizes for the top player or players, as well as the bragging rights the leaders gain from holding the top spots.

#### *4.6 Brackets*

A bracket is a series of two-player competitions, with winners playing each other in a progressively-narrowing field, until a final competition decides the winner of the bracket. Because of the structure, the number of participants must be a power of two -- i.e. 16, 32, 64, 128, and so forth. Therefore, a bracket involves some type of pre-competition registration period, in order to fill the fixed number of competition slots. Players have a set period of time, such as 24 hours, to play their bracket game. If the game is not played, they forfeit the game and their opponent moves on. Like progressives, brackets can be competitions with many players and large prize pools. Unlike progressives, however, which reward a single stellar performance; brackets reward players who maintain a consistently good level of play.

#### *4.7 TopThis!*

This is a one-player competition. A player is given a target score based on his or her own past performance in the game. The player then attempts to beat that target score. If the player beats the target score, a prize is awarded. This type of competition is not offered to brand new players because of the need to collect a history of player scores in order to establish a relevant stretch target score.

## CASE STUDY Royal Flush: Designing an Original Skill-Based Game

Written by WorldWinner

In recent months, poker has continued to ride a wave of popularity. Televised poker tournaments fill key timeslots on ESPN and the Travel Channel, and web sites devoted entirely to poker continue to thrive. With consumer demand for poker at an all-time high, WorldWinner's design team set out to translate the unique flavor of poker into a single-player skill-based game.

The following goals were laid out for the design:

1. Format the game as a single-player experience, in which the player gets a score based on his/her performance.  
Create a game that new players can quickly learn.
2. Limit the play time for a single game to less than five minutes.
3. Include as much of the look and feel of poker as possible.
4. Minimize the "luck factor," as is required for the game to legally qualify as skill-based.
5. Create a fun and compelling game that will entertain players on an ongoing basis.

### Rules of the Game

After initial brainstorming sessions, WorldWinner decided the first four goals would be met by designing the game as a form of solitaire in which the player plays through a deck of cards, one at a time. The player scores points by arranging the cards according to pre-determined rules. The game was dubbed Royal Flush, and a simple set of rules was created, borrowing game mechanics from poker wherever possible.

In Royal Flush, players begin with a deck of 52 cards, a stack of four "Raise" chips and four rows with space for five cards each. They have four minutes to play through the deck of cards, solitaire-style, to create nine different five-card poker hands. Just like in real poker, players can raise some of their hands to earn extra points or fold the hands that don't look promising. The game ends when either all 52 cards have been flipped over or the timer reaches zero. If the player finishes the deck before running out of time, a time bonus is awarded.

By allowing the player freedom to create four hands at once, a large amount of strategic options are made available, and the element of luck affecting the game is far less significant than with regular poker. Additionally, the luck factor is further reduced in actual tournament play because each player receives a deck of cards arranged in the same exact order.





### **But Is it Fun?**

With five of six goals accomplished, the most challenging part came last: evaluating the level of fun, and determining whether the game would keep players entertained.

WorldWinner's programmers created a functional alpha version of the game, and the designers spent several days play-testing.

A problem with Royal Flush immediately became apparent. The player had so much freedom to create poker hands that the order of cards in the deck had too minimal an impact on the outcome of the game. One winning strategy was to simply sort the cards into four rows by suit, scoring points for a flush with each five-card hand. No points would be received for the higher-ranked poker hands, but a player could apply this strategy with such speed that the time bonus made it worthwhile.

After a bit of trial and error, a solution was found by borrowing an idea from the dice game Yahtzee. The rules were changed so that each ranking of poker hand could only be made once. After the first time, no further points could be scored for the same hand. The goal of the game was now to make nine different poker hands one time each, and the order of the cards became much more important to the strategy of the game. To obtain a good score, the player needs to constantly evaluate his opportunities and look for his best chances of creating the top scoring hands.

### **A Royal Success**

A final, polished version of the game was created, and on March 2, 2006, Royal Flush was launched on WorldWinner.com. The game has been well received by the WorldWinner player community, quickly rising to become the third most popular game on the site during its first week, and it continues to perform strongly with over 12,000 cash competitions played each day. As marketing efforts get underway, WorldWinner hopes to introduce the game to poker fans everywhere as a fun and legal way to put their poker skills to use.

## **5. Key Players**

Key players in the skill-based gaming industry include (listed alphabetically):

Arkadium ([www.arkadium.com/](http://www.arkadium.com/))

Game Account ([www.gameaccount.com/](http://www.gameaccount.com/))

GameDesire ([www.gamedesire.eu/](http://www.gamedesire.eu/))

Gameduell ([www.gameduell.de/](http://www.gameduell.de/))

Game Trust ([www.gametrust.com/](http://www.gametrust.com/))

Greentube ([www.greentube.com/](http://www.greentube.com/))

King (MidasPlayer) ([www.king.com/](http://www.king.com/))

Madwin ([www.madwin.com/](http://www.madwin.com/))

Moneygaming ([www.moneygaming.com/](http://www.moneygaming.com/))

SkillJam ([www.skilljam.com/](http://www.skilljam.com/))

WorldWinner ([www.worldwinner.com/](http://www.worldwinner.com/))

## **H. Advergaming**

---

### **1. Overview**

Advergaming continues to become a larger niche within the casual games industry. Advergaming is loosely defined as “the use of games to deliver advertising messages, drive traffic to web sites, and build brand awareness.” Advergaming is now used to sell everything from cars and shoes to servers and soda, and cover most major industries, companies and product types. Branded games can be found in instant messaging applications, banner ads, rich media placements, mobile devices, interactive billboards, traditional websites and pop-ups.

The core premise of advergaming is a simple value exchange: the user gives a brand their eyeballs for three to five minutes at a time, and the brand gives them an entertaining game (usually for free) with embedded marketing messages touting the product, service or industry. Advergaming is distinguished from in-game advertising and product placement by the fact that they are designed explicitly for the purpose of marketing a brand.

Advergaming is one model for cutting through the clutter of today’s ad-saturated environment. The average American child today is exposed to forty-thousand ad messages each year and American adults are often exposed to thousands of ads per day in one form or another. The advantages of advergaming reside both in the amount of time a user is willing to spend with the brand as well as in the stronger associations created through effective gameplay.

With syndicated advergaming ranging from \$10,000 to \$35,000 and custom-built games with budgets from \$25,000 to \$500,000, these small games can be cost effective, attractive alternatives to traditional integration into console games.

Unlike the rest of the casual game space, advergaming has a much more limited distribution network. These types of casual games primarily receive their distribution in one of three ways:

#### **Microsites**

Custom sites typically developed around the game, or the messaging surrounding the particular ad campaign’s branding. These sites are usually hosted by the brand, and need to drive revenue in and of themselves. The best examples become virally spread, and can receive millions of visitors.

#### **Branded Portal –**

These are portals that are built and paid for by a brand. One of the longest running examples of this is Candystand.com which has a large number of games supporting Wrigley brands. Other examples include Nabiscoworld.com and Postopia.com.

#### **Traditional Portal -**

There are a couple portal sites that also offer advergaming content. Within the past year Shockwave.com has had several advergaming built for various clients.

A special case exists for entertainment companies with sites like Nickelodeon’s www.nick.com, DisneyBlast.com and CartoonNetwork.com. The games found on these sites

serve a dual-purpose of advertising and promoting the brand, while also being an end product for site visitors to enjoy.

There is also a growing trend of integrating games and advergaming into larger, multi-channel campaigns such as Nike's multi-channel advergaming campaign on [www.NikeGridiron.com](http://www.NikeGridiron.com). Interestingly, Nike created both a web-based version of MVE that ran on the site, and a stand-alone arcade version for select NikeTown retail stores in the US.

The number of specialist advergaming developers has grown alongside increased interest in the sector. Shops like Blockdot, Freshgames, Fuel Industries, Powerful Robot, and Skyworks now also market themselves as experts in the advergence industry.

Traditional advertising agencies are also taking a more active role in advergaming, with some opening entire divisions devoted to marketing in the video gaming world. Examples include Starcom Media Group's Play division, as well as The Bounce Interactive Gaming Group at Young & Rubicam.

Although advergaming is a relatively new field, its growth has been extraordinary. By combining the major strength of gaming (user immersion) with the capital of advertising, this sector of the industry is poised to continue its significant growth rate. Venerable industries such as food, entertainment, automotive and financial services have all had significant success with advergaming, propelling developers like Skyworks, Blockdot and Freshgames forward. The increased attention to games from all sectors of industry, notably the advertising agencies, promises that advergaming will form an increasingly large percentage of the marketing mix.

## **2. Top Advergaming Genres**

Traditionally there have been two styles of advergaming, and the developers who create them. On the low-end, are quick reskins of existing games and game mechanics with logos incorporated into the experience. These are generally very cheap to build for the developer, and have somewhat limited returns for the advertiser. An example of this would be a typical match-three style game with the pieces replaced with pieces of candy. Most casual game developers can potentially reach the types of advertisers who are looking for this style of simple advergence. However, the pay-off may or may not be worth the effort and the market for this is shrinking as advertisers realize the limited value of such offerings.

On the other end of the spectrum are custom built game experiences tuned to promote an advertiser's brand and marketing message. These games are much more time consuming to build, often reaching budgets of over \$100K with a few having budgets in the \$1 Million+ range. However, this type of game is not something a traditional casual game developer has the ability to deliver. These games are only financed by the largest corporations and typically go through their ad agencies. In order to reach these agencies, a developer must focus on advertising, and realize that their main function will be that of a marketer or ad agency, not a game developer. Additionally the sales cycle for these projects is extremely long, often lasting twelve to eighteen months between initial contact and closing the deal, with an additional six to twelve months of development.

That said, there is room within the advergaming space to expand out into genres that are otherwise not profitable in the casual game space. The goal of an advergence is to provide a fun play experience while promoting the positive aspects of a brand. They are meant to be

played, but do not have to be commercially successful. This means that advergames can expand into genres that generally do not sell well, such as racing, or retro-arcade, while focusing on fun gameplay that accentuates the brand.

Genre	Description	Examples
Retro Arcade	There is a certain nostalgia for games of the past, and often they are considered chic and trendy. Some marketers attempt to capitalize on this “retro chic” movement with new takes on old games.	Nestle - WONKANoid Breakout ( <a href="http://www.wonka.com/games/WONKANOIDBREAKOUT.asp">www.wonka.com/games/WONKANOIDBREAKOUT.asp</a> )
Puzzle	Puzzle games are well established, cheap to make and easy to re-skin for multiple clients. With simple mechanics and rules, they are very safe interactive options for brands, but usually offer limited messaging immersion opportunities compared to other options.	Lifetime Television – Place the Face ( <a href="http://www.candidemedia.com/placetheface/">www.candidemedia.com/placetheface/</a> )  Post Cereals – Postopia ( <a href="http://www.postopia.com/games/gamepage.aspx?siteGameID=343">www.postopia.com/games/gamepage.aspx?siteGameID=343</a> )
Racing	Racing games are another genre that can easily be re-skinned for multiple clients. They also incorporate simple mechanics, and are obviously attractive to car companies looking to create a virtual experience for potential buyers.	Uniroyal – Fun Cup Fun Cup ( <a href="http://80.237.207.52/funcup/index_fm.php?fileLanguage=eng">80.237.207.52/funcup/index_fm.php?fileLanguage=eng</a> )
Strategy	Strategy games come in many different forms and levels of complexity, but this flexibility allows marketers to make strategy games as simple or complex as they like. Deeper gameplay often allows more messaging to come out.	HBO – Fate: the Carnivale Game ( <a href="http://www.hbo.com/carnivale/games/index.shtml">www.hbo.com/carnivale/games/index.shtml</a> )
Trivia	Trivia games can be used to educate the user about the brand or product. They are also easy to update and re-skin.	Hawaiian Punch – You Don't Know Punchy ( <a href="http://www.hawaiianpunch.com">www.hawaiianpunch.com</a> )  Lifesavers – Bet Your Lifesavers: The 90s ( <a href="http://www.candystand.com/games/gamepage.aspx?gameid=190">www.candystand.com/games/gamepage.aspx?gameid=190</a> )
Traditional Sports	With the massive popularity of sports-themed video games, marketers consistently look to them to connect with audiences. The sub-genre of Extreme Sports allows marketers to align with hipper, younger audiences.	Coca Cola – Championship Run 2006 ( <a href="http://www.champrun.com">www.champrun.com</a> )  Coors Light - Trauma Tour ( <a href="http://www.coorslightraumatour.c">www.coorslightraumatour.c</a> )

		om) LG – Action Sports Championships Thrash N' Burn Trailer Park ( <a href="http://www.shockwave.com/sw/content/thrashtrailer">www.shockwave.com/sw/content/thrashtrailer</a> )
Role Playing Game (RPG)	The entertainment and car industries have developed a few RPGs. The narrative structure allows them to create a more complex mood, interaction and story line for a more immersive experience.	Altoids - Curiously Strong All Night Long Game ( <a href="http://www.altoids.com/index.aspx?area=game&amp;sitegameid=87&amp;relationid=405">www.altoids.com/index.aspx?area=game&amp;sitegameid=87&amp;relationid=405</a> ) BMW – Cool Flame, The Game ( <a href="http://www.be-the-first-one.com">www.be-the-first-one.com</a> )
Micro Games	Often companies want something small to accompany their campaigns. These short micro games can accomplish this for a fraction the cost of other games.	Mini – Mini Convertible jump game ( <a href="http://www.miniusa.com/crm/mini_entrance.jsp">www.miniusa.com/crm/mini_entrance.jsp</a> ) Sony – Armchair Games ( <a href="http://www.thearmchairgames.tv">www.thearmchairgames.tv</a> )
Multiplayer	Multiplayer games can create a sense of community around the brand, enhance viral distribution of the game and create a loyal user base. However, there are many performance issues, and corporate liability prohibits most un-censored communication during the game.	Red Bull – Flakes of Fury ( <a href="http://www.redbullbigair.com">www.redbullbigair.com</a> ) Rogue Pictures – Cry Wolf ( <a href="http://www.crywolfgame.com/CryWolfGateKeeper">www.crywolfgame.com/CryWolfGateKeeper</a> )

### 3. Key Players

#### 3.1 Advertisers

The Internet Advertising Bureau (IAB) reported in its 2005 year end report that total Internet advertising revenues grew to a record of \$12.5 billion in 2005. Consumer advertisers continue "to represent the largest category of Internet ad spending, accounting for fifty-one percent (51%) of the 2005 full year revenues." Within the consumer related categories, "Retail represents forty-seven (47%) percent of 2005 full year revenues followed by Automotive at twenty percent (20%), Leisure (travel, hotel & hospitality) at 14 percent, Entertainment (music, film & TV entertainment) at ten percent (10%) category, and Packaged goods at five percent (5%)."

The following table provides an overview of the key industries investing in advergames.

Industry Sector	Notes
Alcoholic Beverages	Long a target of activist groups, the alcohol industry is always looking for new ways to reach their targets. With gaming being so popular, edgy and viral, it's a natural fit. Coors, Corona, and Budweiser have all recently developed games.
Automotive	Many car companies see games as a way to provide a virtual simulation of what it's like to drive their cars. It allows them to showcase different models, and to provide an exciting, edgy test drive for perspective buyers. Recent high-end games have been developed for Dodge, Lexus, and BMW.
Food & Beverage	<p>With physical product differentiation being minimal for many competitive products in the food &amp; beverage industry, games have become a successful way to speak to specific target markets about their brand message.</p> <p>Whether aimed at an older market with Sprite's retro Game System 600, giving young kids colorful mazes in Teddy Grahams' Park, or targeting Bejeweled fans with M&amp;M's Flip The Mix, this industry sees games as a fun way to compliment the personality of their brands and products.</p>
Hollywood Entertainment	Advergaming offers entertainment companies a way to offer viewers a continued experience of a show and its characters beyond the theatre and a 30-minute timeslot. TV network FX's Lucky's Run Game, for example, allows viewers to explore the characters and environments of the show at their own leisure. They can also experience enhanced plotlines and stories not covered by a given show or movie.
Online (Vertical) Portals	Many web-based companies have viewed games as a way to break through the clutter and entice users to consider their service. The most notable of these is Orbitz, who has used a constant flurry of simple pop-up banner games to drive traffic to their site.
Retail Sporting Goods	Nike, RBK, Adidas, Puma, and others have all entered the web-games sphere. Nike has developed the most games, creating a number of high-end, web-based Wild Tangent games, as well as game experiences for AIM Expressions, and SMS Location Based Games ( <a href="http://www.nike.com/operation6453/">www.nike.com/operation6453/</a> ).
Serious Games & Recruitment Games	<p>Web-based games are not only being used to sell products, but they are also being used to sell and educate about causes. Games are being produced to teach volunteers how to help political candidates with their campaigns (Dean for Iowa Game), to present points of view on major world and political issues (AFKARMEDIA's UnderAsh), and to recruit for the armed forces (U.S. Army).</p> <p>There are also educational games being produced by organizations to educate people about the dangers of salty foods (UK Food Standards Agency's Sid Game), as well as to encourage users to take care of their prostate glands (Prostate Cancer Charity's Journey to Planet Prostate). If the effectiveness of these interactive applications can be proven there will be many more funded in the</p>

	future by these causes.
Telecom	Many of the largest advertisers (in terms of dollars spent) are in the telecommunications industry. There is a lot of money to be made in this industry, and telecom companies are all trying to capture their target markets by any means possible. Many have already attempted to use games (web and console) to correlate their brands with speed, reliability and fun.
Toys	Toys and games have always gone hand in hand. With the inanimate nature of many physical toys, interactive web-based games offer the perfect opportunity to bring these characters and toys to life in the virtual space, jump starting your imagination in the physical space.

Few of these advertisers develop their games in-house. Usually they have their interactive agencies build the games or outsource to a game development company.

### 3.2 Developers

Adgame-Wonderland ( [www.adgame-wonderland.de/](http://www.adgame-wonderland.de/) )

Arkadium ( [www.arkadium.com/](http://www.arkadium.com/) )

Blockdot ( [games.blockdot.com/](http://games.blockdot.com/) )

Casual Game Store ( [www.casualgamestore.com](http://www.casualgamestore.com) )

Denwerk ( [www.denkwerk.de](http://www.denkwerk.de) )

Gamelet.com ( [www.gamelet.com](http://www.gamelet.com) )

3D Groove ( [www.3dgroove.com](http://www.3dgroove.com) )

Fresh Games ( [www.freshgames.com](http://www.freshgames.com) )

Fuel Industries ( [www.fuelgames.com/](http://www.fuelgames.com/) )

Funny Garbage ( [www.funnygarbage.com](http://www.funnygarbage.com) )

GameBrander ( [www.gamebrander.com/](http://www.gamebrander.com/) )

Ganymede Technologies ( [www.ganymede.eu](http://www.ganymede.eu) )

Inludo ( [www.inludo.com](http://www.inludo.com) )

Leviathan Games ( [www.leviathangames.com](http://www.leviathangames.com) )

MindComet ( [www.mindcomet.com](http://www.mindcomet.com) )

Ogilvy Interactive ( [www.ogilvy.com/o\\_interactive/](http://www.ogilvy.com/o_interactive/) )

Panlogic ( [www.panlogicgames.com/](http://www.panlogicgames.com/) )

Powerful Robot Games ( [www.powerfulrobot.com](http://www.powerfulrobot.com) )

Random Media ( [www.randommedia.co.uk](http://www.randommedia.co.uk) )

R/GA ( [www.rga.com](http://www.rga.com) )

Shockwave ( [www.shockwave.com](http://www.shockwave.com) )

SkiveCreative ( [www.skivecreative.com/](http://www.skivecreative.com/) )

Skyworks ( [www.skyworks.com](http://www.skyworks.com) )

Soap Creative ( [www.soap.com.au](http://www.soap.com.au) )

Templar ( [www.templar.com/](http://www.templar.com/) )

Thup ( [thup.com](http://thup.com) )

WM Team ( [www.wmteam.de](http://www.wmteam.de) )



### **III. Business Models**

#### **A. Introduction**

---

As the casual games sector grows and becomes increasingly competitive, it is critical for anyone who has a stake in casual games to fully understand the business models that support the industry. Any successful casual games entrepreneur can tell you that relying on tried-and-true formulas established in traditional gaming sectors will limit your ultimate success in casual games.

Furthermore, the transition of casual games from a cottage industry to one of the highest growth sectors in gaming over the past few years means that new business models are emerging all the time. New players, some bolstered by significant private or venture investment, are continually entering the casual games market. One of the best ways to compete with existing companies in any market is to change the rules – and that’s why there is starting to be widespread exploration of alternate business models. Through greater investment in this sector and consolidation, these new players are aggressively seeking to gain a competitive advantage over existing companies, many of whom may not be able to keep up as 2006 progresses.

Another factor resulting in new models is the increasing globalization of the casual game industry. Companies who are successful in one country are starting to expand into others, and are bringing new ways of doing business with them. This is especially apparent when considering Asian countries like China or Korea, where extremely high rates of piracy have forced companies there to invent new business models.

This Business Models section focuses predominantly on the PC/Mac market, but much of the content of this section applies to alternate platforms as well, such as set-top boxes, handheld gaming devices, mobile phones, consoles, and more.

#### **B. Understanding the Value Chain**

---

A value chain encapsulates all of the various stages or processes that any product goes through on its way to the consumer. It is called a value chain because each stage presumably adds value to the ultimate product or service – or else will eventually be optimized out.

The profit earned at each stage is generally commiserate with the value contributed at that stage, with some stages inherently more profitable than others. Consider the PC business. Most stages of the PC business are highly competitive, and hence not especially profitable. The companies that create motherboards, memory chips, hard-drives, etc are locked in what are essentially commodity businesses. Microsoft, however, has a near-monopoly over the operating system, and therefore captures a huge piece of the overall cost of a typical computer.

In mature industries, the force of competition has typically driven value chains to evolve over time to be very efficient. Companies tend to specialize on a particular stage of the value chain, doing one thing and one thing only, but doing it extremely well.

In new or emerging industries, however, value chains tend to be very fluid. No one knows yet which stage will emerge as the most profitable – and so companies are reluctant to specialize too early for fear of becoming trapped in a very competitive, commodity business.

Consider the movie industry. Originally, movie studios sought to control every stage of the value-chain, from writing their own screenplays, to hiring & managing actors, to developing their own proprietary camera technologies, to producing films, and distributing those films to movie theatres that they themselves owned.

Today, of course, the movie industry value chain is highly decentralized. Entire industries have emerged to service just a single stage of the entire value chain: movie theatre operators, physical film distribution, completion bond financing, visual effects creation, film stock production, etc.

Now consider the casual game industry. The casual game industry is still very new, and so the overall value chain is still very fluid. There has been relatively little specialization, and it is marked by a high degree of “co-opetition” in which companies compete within one stage of the value chain, and cooperate on another. The market continues to change on a monthly basis, and it is difficult to predict exactly what it will look like as it continues to mature.

Nevertheless, it is possible to define a few distinct stages of the current casual-game industry value-chain, even while it is not yet possible to define which of these will end up as the most lucrative.

Most companies who operate today in the casual game space fill more than one of these roles. There are developer/publishers, distributors/retailers, developer/publisher/distributors, and all of the above. Nonetheless, considering each stage in isolation is still helpful for understanding the overall industry and especially in trying to predict the direction it is heading.

### **1. *Developer***

The value-chain starts with a developer creating a new game, and involves the contribution of designers, producers, programmers, artists, sound designers, music composers, and more. The output of this stage is a game, ready to be passed up the value-chain on its way toward the eventual consumer.

One could argue that there are at least two other value-chain stages that feed into this stage – tool providers and sub-contractors. Tools do play an important role in building any game (e.g., Photoshop, Visual Studio), but at least so far today a distinct market providing casual-game industry specific tools has not yet emerged. Many successful developers can and do provide all of their own technology. Also, much of the technology required to create a casual game is available for free over the Internet. Likewise, while many casual game developers outsource parts of their work to other companies (e.g., sound effects, music, art), a casual-game specific outsource market has not yet emerged.

This is typically one of the most competitive stages in the entire value-chain, since there are so many developers eager to make games for a living and there can only be so many “hit” games. The only real way to build value at this stage of the chain is to create and own valuable IP which can then be successfully monetized by the other steps in the chain.

Examples of high-profile casual-game developers include PopCap Games, gameLab, Large Animal Games, Skunk Studios or Sandlot Games.

## **2. Publisher**

In the traditional game industry, the publisher plays a key role in the overall value-chain, working with developers to create games on one side, and then selling and marketing those games on the other.

On the game creation side, publishers generally provide services such as funding, production oversight, quality-assurance (QA), and release management. Publishers typically have an entire portfolio of games under development at any one time, and often license well-known brands or other sources of IP which they then work to turn into marketing games.

On the sales side, publishers typically package games to be sold, localize games for international sales, handle end-user marketing and PR, and work with retailers and distributors to physically get the games out into sales channels.

In the casual game industry, the relative simplicity of casual game development and the existence of efficient online sales channels have historically minimized the need for stand-alone publishers. Most developers self-published their own games, funding their own development and negotiating their own distribution deals. Developers typically paid little attention to marketing or PR, relying on the games to 'sell themselves'.

This is changing, however. As the market matures and competition increases, the role of the stand-alone publisher is becoming more defined. Several online retailers, not interested in working with dozens (or hundreds) of different developers, are limiting the number of partners they work with. Securing prominent placement or promotion is no longer so easy, and managing effectively all of the various channels that exist is becoming a full-time job.

An example of a high-profile casual-game publisher is PlayFirst.

## **3. Distributor / Aggregator**

In the brick & mortar world, the role of a distributor is to physically move product from a manufacturer to the retailer. In some cases, value-added distributors also handle inventory management and even some manufacturing. Distribution is typically not a glamorous business, with relatively small margins.

In the casual game business, however, distribution has emerged as one of the more interesting aspects of the business, even as paradoxically the need for distribution would seem to be minimized given the ease of distributing content on the Internet.

Instead, in many cases distributors are adding value to portals by offering to outsource their game retail operations, operating their day-to-day sales, managing their promotions, and aggregating together many games into a single feed that is much easier for retailers to handle.

On the flip side, distributors add value to publishers by helping them publish their games on hundreds of channels to which they would not otherwise have the bandwidth to manage. A

publisher can hand a game off to a distributor once, and let the distributor handle the process of getting the game out onto many channels. This especially makes sense for smaller sites which are too small for a publisher to justify managing directly, but which in the aggregate still represent significant sales, especially internationally.

Examples of high-profile distributors include Oberon Media and Trymedia (now part of Macrovision).

#### **4. Retailer / Portal**

The final stage of the value-chain is the retailer or portal. In the casual-game industry, these are typically web-sites which consumers visit to try out or purchase games. The ultimate value of a retailer is determined for the most part by the number of users who visit the retailer on a regular basis, since that determines its revenue potential (assuming a fixed average revenue per visitor).

Several years ago, a portal could be successful by just listing a games on its site, and investing in a basic payment-processing system for charging credit cards. Sites generally had the same games and the same prices, and there was little differentiation between the portals.

As the market has matured, sites are starting to compete more aggressively with each other for users, and are looking for ways to stand out. Exclusive content, subscription programs, contests, and community features are all examples of ways in which sites are trying to differentiate themselves.

Services that retailers require include Digital Rights Management (DRM) technology, E-commerce / billing solutions, user account management, and site statistics and tracking. Several companies have sprung up recently to provide these services, and many more are on their way.

Examples of high-profile retailers include Yahoo Games!, RealNetworks, Big Fish Games, and the MSN Zone.

#### **5. Consumer**

No discussion of the value chain would be complete without mentioning the consumer, the ultimate end of the value chain and the source of the money that trickles back to all of the links in the chain.

In some cases the consumer provides that money directly by paying for a game outright. In other cases advertisers provide money on behalf of the consumer by "sponsoring" game-play through ads, product placement, or other means.

#### **6. Various Permutations**

As mentioned earlier, nearly every possible permutation of value-chain roles is being tried in the market today. The most common include:

Developer direct-to-consumer. Many developers operate their own websites and try to sell games directly to their fans. Most developers do not actively promote their sites, and

instead rely on consumers discovering their site after first coming across one of their games on another site. This creates obvious tension, however, between developers and their distribution partners.

**Distributor / Retailer.** Some of the biggest portals operate their own websites, and actively solicit games from developers, rather than outsource that role to a distributor. Likewise, most of the distributors also operate their own websites (though this does cause some conflict with their white-label clients).

**Publisher / Distributor.** Many distributors also act as publishers, a natural step considering that they interact with dozens or hundreds of developers on a regular basis. When a distributor spots a successful developer or a successful game, it can offer to represent and publish that game more broadly.

## **7. Revenue Shares**

Each of the players in the value chain can receive different portions of the revenue depending on a wide variety of factors. For more information on this, including reference revenue share percentages, refer to the survey data included in the Publishing section of this paper.

## **C. Business Models**

---

With an understanding of the casual game industry value chain, it is now possible to discuss business models currently in use. An understanding of the value chain is important, since each model affects companies at various stages of the value chain differently. For example, the “try and buy” model in which consumers purchase games outright accrue more value to the game’s developer than the “subscription” model which accrues the most value to the company who holds the subscription itself, generally the retailer.

It should be noted as well that the business models below, used within the casual games industry to varying levels of success, are not necessarily the only ones possible. Companies in the industry continue to experiment with new models and hybrids. It is important to understand who accrues value within any model (not only direct monetary value, but also things that may be monetizable in the future. e.g. credit/attribution or brand value, customer retention, etc).

### **1. Try and Buy**

The “Try and Buy” model is the simplest model to understand, and is also the model historically associated with the casual game industry. This model is also most similar to the shareware model which has been around for many years. Under this model, consumers may download and play a trial version of a game. Trials are limited in some way, most commonly by time (a one-hour free trial is typical), but trials may also be limited by features, by number of plays allowed, or by some other mechanism.

Throughout play the consumer is typically encouraged to purchase the full version of the game (“up-sell”); once the trial expires the consumer is required to purchase the full version in order to continue playing. In some variants of this model, the player may continue playing the feature limited version indefinitely, in others the player must sit through increasingly annoying “nag screens” encouraging the user to upsell.

Enforcing the trial is generally the task of Digital Rights Management (DRM) software; usually the game is “wrapped” with DRM software, though sometimes the DRM is built directly into the game itself. The DRM software is typically also responsible for handling the up-sell itself, accepting the user’s payment information and handing the transaction off to a back-end payment processing system.

Providing the DRM is generally the responsibility of the retailer or the distributor; the publisher generally hands off an unlocked version of the game to be wrapped along with marketing collateral to help in building the up-sell screens.

Under this model, revenue from game sales by the retailer are generally shared with partners “upstream” with pre-determined royalty rates or revenue shares. A typical developer revenue share is anything from 20%-50%.

One way to measure the success of a game under this model is to look at the “conversion rate”, or the percentage of downloads which get “converted” into sales. Conversion is typically low – anything over 1% is considered okay, anything over 2% is considered great. Game sales can be increased in one of two ways: encouraging more users to download the game, and increasing the conversion rate of those users who do download the game. As a result of this small conversion ratio that many people are looking at alternate means to monetize the consumer in an effort to create a more optimized marketplace.

## **2. Free Web Trial**

A common variant on the “try and buy” model is to offer a free web version of the game which can be played within a web browser. Web games can generally be played over and over again, however they are typically very limited compared to the “deluxe” versions, with fewer features, less content, lower quality sounds & graphics, etc.

A good web game is designed to encourage the user to download the deluxe version. Many users (especially those with slow dial-up modems) will not download a game without first trying the web-game version, though others will only ever play the web games and will never download deluxe versions for a variety of reasons. Some users are simply not allowed to download & install software on their computer (e.g., office workers). Others have been told “never download software on the Internet” and are fearful. Still others simply don’t want to “clutter up” their computers with lots of random software.

Increasingly web sites are looking to monetize users who play the web games without ever downloading the deluxe versions through advertising (see the section on advertising below).

Providing a web game for use ad-sponsored free play, especially in the (currently common) case that advertising revenue is not shared, can be viewed as counter to the interests of the developer/publisher, who's interests are in selling downloads. However, many view it as a successful tactic to getting websites to keep a less-than-stellar selling download game within their catalog, where it might otherwise be dropped. The theory being that the extra 'long tail' downloads justify the cost of developing the web version of the game. Increasingly retailers and portals are starting to share advertising revenue with developers, and this should be discussed as part of any distribution agreement that includes a web version.

### **3. Subscriptions**

Retailers typically report that only a small percentage of their customers buy more than one game under the “try and buy” model. Subscriptions are one way to try and earn more revenue per consumer by creating an ongoing revenue stream.

There are several types of subscription models in wide-spread use:

“All you can eat” The consumer pays a fixed amount per month in return for unlimited play of all games in the program. The user must maintain the subscription in order to continue playing games; when the subscription ends the player’s access to the games also ends. Revenue is generally shared based on game-play starts since it is the only statistic that can be accurately measured.

“Book of the month” The consumer pays a fixed amount each month in return for getting one (or more) games free. Additional games can generally be purchased at a discount.

VIP membership. The consumer pays a fixed amount each month in return for special privileges. This model is most common in Asia, where casual games are often free and monetized through other means (such as item sales, see below).

Models for sharing revenue with developers/publishers whose titles are offered within subscriptions varies between vendors, subscription type, and platform. Common ones are based on game starts/plays (i.e. If a player played nine games of game A, and one of game B, ninety-percent (90%) of the share allocated to developers would go to game A’s developer); or game time (i.e. nine hours logged playing Game A vs. one-hour playing Game B would result in the same split as the previous example).

### **4. Pay-for-play / Ad Sponsored hybrid**

At the 2006 Game Developers’ Conference, WildTangent announced a new business model for games sold through their ‘WildTangent Games Network’ (WGN). This new model is an online version of the classic arcade where you insert coins into the machine for each game play. The WildCoins is a ‘pay for play’ model that is adapted to the existing try and buy model. End users get their first two game sessions for free (the trial period), then they have a choice to spend \$20 and convert to a full purchase, or spend about \$.25 for the next game play, and the next, and the next. Another interesting component of this new business model is that advertisers can buy the WildCoins and sponsor a game session that is then free for the players.

The combination of free play, sponsored play, \$.25 cent play and full purchase creates a payment structure that could possibly yield much higher revenues per game, and closely ties the value of a game as measured by the number of times it is played, with the money it generates. Other companies are also looking into this type of advertising based system, although it is still too early to tell how effective it is at generating additional revenue.

### **5. Tournament / Skill-Based**

This economic model involves players playing a cash entry fee to enter a tournament and play a game. The tournament can be as small as two players, or as large as thousands of players. A player’s score is posted on a leader board, and the winner (or, in larger tournaments, winners) gets a cash or merchandise prize.

The tournament provider makes money by keeping a portion of the cumulative entry fees, rather than returning all the entry fees as prizes. For example, a 5-player tournament with a \$1.00 entry fee might pay a \$4.00 prize to the first-place finisher, with the tournament provider keeping \$1.00.

In order to be legal in the locations where such tournaments are offered, the games must be substantially skill-based, rather than relying on luck or chance; thus the name "skill-based" to refer to this economic model.

Skill-based games are discussed more fully in the Market Overview section of the whitepaper.

## **6. Other Methods**

Other methods seen in the marketplace today are:

- Advertising sponsored
  - Can also be used to pay for nearly every other business model in lieu of cash
- Online ads (web games)
- In-game advertising
- Tournament/skill-based games
  - Cash tournaments vs. free tournaments
- Item-Buy (micro transactions)
  - Clothing / avatar
  - In-game items
- Packaged product / retail
- Alternate platforms
  - Mobile phone / PDA
  - Consoles
  - Set-top box / interactive TV
  - Handhelds

## **D. Contractual Issues**

---

The typical contracts seen in the casual game market are.

Work-for-hire agreement (subcontractor -> developer)

Publishing agreement (developer -> publisher)

Distribution agreement (publisher -> distributor)

Reseller agreement (distributor -> retailer)

End-user-license (retailer -> consumer)

### **1. Key contractual points**

Disclaimer: this section is no substitute for a real lawyer. You should always speak with a lawyer when evaluating contracts.



## **2. Length of term**

Defining the length of the term of the agreement is a very important detail that should not be overlooked in the contract. A typical term can last between twelve months and perpetuity.

Envision what will happen after the contract has expired. Will IP rights revert back to the developer? Will the developer be able to take the game to another publisher? Will the partner have the ability to continue selling the game for a period of time after the contract has expired (common for retail products, it may be called a sell-off period)?

Negotiating an agreement which automatically renews after the initial term will allow a bit of caution as everyone will have an "out" after the initial term has expired. An example of this might include a contract with a 12 month term, which will automatically renew for additional 3 month increments until one party chooses not to continue.

This can work in a developer's favor if the game is an absolute hit. The termination clause would allow the developer to choose not to renew, and then bring the game to another partner for more favorable terms.

This can also work against the developer if the game is a flop, as the partner may then choose to discontinue a distribution agreement after a short twelve months.

## **3. Termination Rights**

Who can terminate?

When can they terminate?

What are the consequences of termination?

## **4. Compensation (revenue sharing, fees, etc)**

- payment terms
- Minimum price per unit
- Gross revenue vs. net revenue
- Definition of net revenue (what's in, what's out)
- Guarantees / royalty advances

## **5. Milestone Descriptions**

Typical development milestones are:

- Project Start

This is typically when the contract is signed, and is only significant if the contract calls for payment on project start.

- Feature Complete

This is where the game's core features are complete, but many of the ancillary elements are still in progress.

- Alpha

At this point the game and ancillary elements should be functional, but will probably still have bugs and graphic / audio pieces missing. All agreed to functionality should be present at this time.

- Beta

The game and all features should be in and nearly finalized. No placeholder graphics or audio should be present, but bugs and further playtesting are still expected at this stage.

- Gold Master / Release

The game is completed and ready to be sent through to distribution for end consumers.

## **6. Trademark usage / guidelines**

## **7. IP rights**

While a brief discussion of IP rights is provided here, there is much more information including an IP Rights White Paper at the IGDA's IP Rights SIG: [www.igda.org/ipr/](http://www.igda.org/ipr/)

## **8. Audit clause**

The audit clause is an essential staple of every distribution and publishing agreement. It gives the right of the developer to access the pertinent accounting records of the publisher/licensor without the need to sue for disclosure. The audit clause, however, is very rarely exercised. While the developer has the legal right to exercise it, it projects distrust of the publisher/licensor - which is clearly harmful to a long-term relationship. Only when developers suspect severe accounting abuses should this be exercised. To avoid needing to exercise an audit, a developer should require the publisher/licensor to provide online or frequent (monthly) reporting by channel including Gross Sales, Gross Revenue, itemized expenses and Net Revenue to developer.

What follows is audit language to include that is bipartisan and widely accepted by publishers/licensors. The five percent (5%) is sometimes seen as ten percent (10%), but negotiate to the more standard five percent (5%).

"Audit Rights. Subject to the confidentiality provisions set forth in Section X, and upon the thirty (30) day prior written request of DEVELOPER, DISTRIBUTOR shall make its books and records directly related to payments pursuant to this Agreement available for inspection during normal business hours by an independent certified public accounting firm retained by DEVELOPER, for the purpose of verifying the payments owed to DEVELOPER, at DEVELOPER'S sole cost and expense. DISTRIBUTOR shall keep records in reasonable detail and supporting documentation consistent with normal industry practices. DISTRIBUTOR agrees to reimburse DEVELOPER on a prompt basis for the cost of such audit in the event such audit reveals that the amount paid to DEVELOPER during the period covered by such audit is less than the amount actually due for such period by an amount greater than five percent (5%) of such amount actually due for such period. Audits shall be limited to one per year. If such audit uncovers an overpayment to DEVELOPER, DISTRIBUTOR shall receive appropriate credits against future sales of the Programs."

## 9. *Geocultural Issues*

One often overlooked aspect of game content development is the aspect of "geocultural" content issues, which can be defined as those content elements - whether in-game or related to the game - that might have potential for cultural and/or political sensitivity in local markets (the US as well as abroad). While such issues are typically more a problem for thematic, non-casual games that employ a generous use of real-world history, symbology, religious themes, geographies and other related aspects, it should be noted that any type of game content - including that of casual games - can be affected.

How does a game developer best manage the potential sensitivities of their content? At least three steps can help:

**Establish proactive processes:** Ensure that the content design and development accounts for the cultural and political implications of the content. While localization personnel/resources sometimes touch upon these topics, it is not a typical focus of most localization efforts and thus may require additional subject-matter expertise.

**Evaluate the "reasonable risks" versus "overt offenses":** Content of reasonable risk is of a type that it could be sensitive but not so much to cause serious repercussions, thus it can be left intact given the local market conditions. Content with overt offense contains well-known sensitivities and should be avoided.

**Post-development process:** Establish clear procedures for responding to and mitigating end-user feedback and/or complaints, in the event that a content element in your content caused a problem in a local market. The key here is to have a clear rationale and defensibility for content that can be easily conveyed.

Again, the great majority of casual games (and any game for that matter) may not encounter such geocultural sensitivities, but in today's world of heightened cultural awareness and interaction as well as the growing ubiquity of games, it's prudent to at least be aware of the potential.

## 10. *Other Significant Points*

In addition to the above mentioned contractual points, a typical contract will contain clauses that pertain to

- Exclusivity
- Territory
- Guarantees / indemnification
- Non-disclosure
- PR / press releases

While these have not been discussed with detail here, they are still significant points that should be considered when negotiating a publishing contract.

## ***E. Conclusion***

---

The Casual Games business is still a very nascent marketplace, and as a result the business models being employed are in an almost constant state of flux. There is a general feeling that there is as much room to improve and innovate on the business side of casual games, as there is within the games themselves.

## IV. Production and Design

### A. Introduction

---

The Casual segment of the games industry changes almost as rapidly as the Internet itself. Technology evolves, broadband usage increases and, every day, more and more people are playing and accessing and even playing their games online. Not only that, casual games are becoming richer and more complex. The evolution in casual game design is finally taking its own path and leaving behind many design rules that applied to core video-games.

In this section, we will examine what it means to design games for the evolving casual games medium and its wide-ranging, international audience. Specifically we will examine the production and design issues facing developers who are looking to make a successful title for this market segment – and focus on the following issues:

- Design Considerations – Designing casual games with your target audience in mind.
- Game Mechanics – Typical game mechanics found in casual games. Key design elements specific to Advergaming, Single-Player Games, or Community Based Games.
- Production Issues – Production and process issues inherent to casual game development.
- Process Differences – Differences between producing casual games vs. other gaming platforms (e.g., console).
- Localization - Effective ways of delivering your games to a world-wide audience.

There are many types of casual games - each with their own design considerations. The table below provides an overview of key design differentiators, typical file size and additional examples of each type of casual game discussed in this section.

### B. Design Considerations

---

Casual games have a unique set of design considerations. Although many important issues in the development of casual games are common to all game design, but the nature of the audience, business model, and context of these games shapes their design philosophy. We know that casual game users play these types of games for three reasons: relaxation, diversion, socialization and competition. These reasons are not mutually exclusive and the best games touch on all of these elements in one way or another.

#### 1. Key Design Elements of Casual Games

##### 1.1 Simple and Meaningful Play with Transparent Rules

At the heart of game design is the idea of meaningful play, the idea that players interact with a game in ways that produce clear and purposeful results. What differentiates games from other forms of media is that players engage the game's system to change it in meaningful ways. This idea of meaningful play is manifested in a number of ways in casual games. First, there is the importance of understanding basic gameplay; players' actions must elicit clear and understandable responses. In a similar vein, it is important that the rules of a game be transparent. Just as the basic interactivity needs to be entirely comprehensible, so the rules of the game should also be at best intuitive and at worst easily

grasped by players. In even the most complex computer games, rulebooks are used only for reference; most of the play is learned through experimentation and in-game instruction. Similarly, a casual game's rules should be intuitive and require no more than a one-screen help or simple tutorial to thoroughly understand.

## 1.2 User Interaction

Many times in casual games much attention and focus has been shifted towards making sure the game is attractive to the eye and utilizing animations and particle effects, however it is essential that users are able to play the games by utilizing visual and audio cues within the game.

Casual gamers will rarely go through a lengthy tutorial, and as we all know manuals are useless, even for videogames. So as a developer don't be discouraged if your game is complex. First try to figure out if you can explain the game through various levels, rewarding the user enough to keep them interested in the game. If you can't break your game into small chunks of interaction, then there could be something wrong in your design for a casual audience.

This is what makes interface design so important. The concept can be summed up as – “The user should always know what the next step is by just looking at the screen or know what is happening by watching/listening to the game.” It is important to make a game attractive to the eye, but if people cannot figure out how to play – they will stop.

Every game is different, so how to implement the visual and audio cues will always be different. Visually – anything that is important to progressing in the game, such as the “next logical step” should always “pop” in some way, especially if it is something outside of the normal mode of play. If a game is designed correctly, users will quickly become accustomed to how the game should be played. If a user needs to click something additional, they should be prompted in some way. Blinking, throbbing, hi-lights and even an arrow pointing at what to do next are easy visual clues as to what is happening. One trick some developers use is to look away from the screen. When you look back, the first thing that catches your eye should be the next thing you should do (i.e., click on).

Another good feature that developers have used is a “waiting user” prompt. If the user should spend a period of time not doing anything, or not making progress, the game will prompt the user in some way of what to do next.

The interaction between the user and the game should be limited to the computer mouse. The use of the keyboard in a casual game is a big obstacle for users. Since the pace of the casual game is usually relaxed, and the gameplay is easy, adding more than one device to control the game makes people react negatively since having to remember and coordinate keystrokes with mouse movements is something that is hard to do for those who are above 30.

Use of buttons, where possible, should be limited to the left-mouse button. Use of the right-mouse button is actually possible in casual games, but ideally it should be used for an action that is not absolutely necessary. An example is the use of the right-mouse button occurs in Zuma, where players can swap their current ball for another ball of a different color. Some players will never use it, whereas advanced players will have that “expert”\_option.

While designing the game, designers must combine simple controls and an easy-to-read interface that simplifies the number of clicks that the player has to accomplish an action. A good example is the contextual interface in *Plantasia*: where when the player hovers over a bug the cursor icon turns into an insecticide can.

### *1.3 Depth & Complexity*

An important issue is the complexity of games. Considering the normal play periods of casual games (short) and the game experience of the average players (limited), it is not appropriate for casual games to have wildly complex systems that require careful, constant attention and deep strategic thinking. A real-time strategy game with hundreds of units to choose from, or a 3D world with miles of virtual space to explore presents an experience designed to captivate a dedicated user for hours of intense play. However, many people play casual games to take a relaxing break from work or to pass the time with something engaging. This means that casual games should be based on simple core activity that leads to emergent complexity. Initial access to the game should be easy, and the difficulty and engagement of the game should come from doing that same basic activity in increasingly challenging environments. In other words, the game should give the player a simple way of interacting that becomes a rich experience in the game's context.

More complex forms of interactivity, as the real-time strategy game mentioned above, often have steep learning curves that run against the casual interests of the audience. This desire for simple core interactivity has a number of implications. First, a casual audience is generally not interested in memorizing complex macros or commands to understand a game. Thus, casual games are predominately mouse-based, either exclusively or with wholly optional hotkeys. When casual games do use the keyboard, it is almost always limited to arrow keys and a single action key. Thus, the core activity of the game is also fairly simple: clicking on a pair of grid squares to switch two objects, moving the mouse over a deck to reveal a card, or dragging an item from a palette to a specific spot on a game field. The complexity of the game comes from the way that simple interactivity mechanic is used in the context of new levels and available resources (new things to swap, new cards to see, new terrain on the field to negotiate).

In addition, the game should require very limited help to understand. Even more than other kinds of games, casual games should not require players to read detailed instructions or experiment extensively to grasp basic game concepts. Anything that is so complicated that it requires more than a single page of simple help or (at most) a tutorial first level is most likely too complicated for the market. It is also important to keep in mind that many of the tropes that hardcore gamers have internalized are not part of many casual players' vocabularies. This means that what may seem standard conventions to gamers (e.g. WASD for movement, smashing crates to get health) will be lost on a large part of the casual games audience. This is all the more reason why the basic gameplay must provide clear and consistent feedback.

### *1.4 Rewarding Players*

As part of the experience, a rich and varied reward system is very important to casual gamers. The player's main reason to play casual games is to get away from the worries and frustrations of everyday life. Being constantly rewarded is a way to make the player feel good about what they are doing.

Of course, the casual game designer shouldn't confuse rewards with simplicity of play. The game challenge should still be there, but the player should be notified of their good deeds as the progress through any game challenges. Bonus points for specific actions, awarding combos, and themed player ranks have been common reward resources in casual games. Rewards are also related to the optional depth that can be added into the game (see the Optional Depth section for more info).

Overall, increased rewards for successful actions and reduced penalties for early mistakes are key components to making a successful game for the casual games market. The consumers driving this market generally do not see themselves as "gamers" looking for a deep challenge. Often, they are looking for an immediately fun and positively rewarding entertainment experience similar to the benefit one receives when tuning into a trusted primetime TV show that provides a mental escape with a minimal learning curve.

### 1.5 Showing Progress

Rewards will not be relevant if there is no place in the game where progress is represented. Probably one of the most important elements of any game is the score as this is a key motivation factor for many games. If the game is score-centric, anything that either adds to or takes away from the users score should be point blank obvious. Users should immediately know when they have done something good and their score increases, or when something bad happens. Even if the game is not score-centric or there is some other kind of task that is necessary for a user to complete to move ahead in the game, accolades should always be represented with some kind of positive audio cue.

### 1.6 Forgiving Game Play

Casual games should be very forgiving, particularly in the early part of the game. The game should not punish players too harshly for initial mistakes, and should give new players time (whether it be in early levels or beginner's modes) to learn the core interactivity. The grace period can last quite far into the game, with the difficulty only ramping up well into the experience. Consider that the majority of this audience prefers games such as Bejeweled Easy Mode, which is very difficult to lose unintentionally. Difficulty should be less of an obstacle that players must struggle to overcome than a natural growth that matches the player's increasing expertise of the game.

### 1.7 Visuals and Themes

Keeping the audience in mind, the visual design in a casual game could vary depending on the target audience. The game designer should ask first who is going to play the game before deciding on a particular theme or visual style.

Successful casual games have historically relied on themes that that players are familiar with. Games that are based in real life environments (e.g., Cinema Tycoon) give players a sensation of a familiar environment. Other themes that casual players are familiar with have been portrayed successfully in popular Hollywood films – magic (Lord of the Rings) or exotic places (Indiana Jones) are some examples. Contrary to popular belief, sci-fi movies are not as popular across mass audiences. Casual Game designers should stay away from sci-fi themes in their games.



Use of a familiar environment: Cinema Tycoon (Tik Games)

Many downloadable, single-player games take the path of a "realistic" look. But some have started to have cartoon-like looks (e.g., Diner Dash). The key is to not make the player feel they are playing a game for kids. Contrary to puzzle games in arcade machines in the 80s (e.g., Pengo), since most downloads get purchased by an older audience players in general get put off by "cute" characters or visual elements that make the sensation of watching a Saturday morning cartoon. Exceptions to this case tend to gravitate towards those games who have a realistic theme (e.g., Cake Mania) or have UI and supporting graphics with a mature look (e.g., Chuzzle).

When it comes to online web games, the choice of themes and visuals tend to be different. Since the age range of web game players goes across the board, there are all kinds of visual themes used successfully. It will depend on the web site's audience mostly. Sites like Cartoon Network, Nick.com, and to a big extent Neopets target younger players and hence cartoon like visuals are the norm and work very successfully.

Use of bright and shiny colors, appropriately used, is a plus. Even some special effects (i.e.: the use of particle systems to create an explosion of stars when an award is given) have worked well in some games.

### *1.8 Narrative and Characters*

Any discourse on story and character (and by extension, narrative and metaphor) in casual games, ought to probably be prefaced by a brief acknowledgement of the challenges our larger industry faces in brokering the thus-far awkward marriage between interactive entertainment (including console or "core" video games) and compelling narrative (including story telling and character development).

There are numerous books and courses available on story-telling, character development and approaches to narrative structure. Many of these focus on writing for film, television and fiction. There are some on comic book writing, a few for games, but next to none on casual games. When folks do discuss storytelling's dance with video game design, inevitably they bemoan the lackluster job we as game designers have done (to date) of seamlessly integrating story and interactive play. But the common wisdom is that, as a relatively new medium, video games are still getting their sea legs when it comes to sophisticated approaches to story telling, defining a compelling character or, more famously, of moving a player to tears.

The question is, where do Casual Games fit into this larger polemic? Do we operate under the same rules and are we wading through the same challenges as core game designers? Or, are the idiosyncrasies of casual games and their non-hard-core-gamer audience going to liberate the casual game designer and give him or her a uniquely advantageous position at the cutting edge of elegantly interpolating story and interactivity?

#### ***Does a Casual Game Even Need a Story?***

Not all successful casual games have needed story and character. After all, isn't gameplay everything? Who cares about story? Why add characters? Why add story? It costs more, right? And it's hard!

Sure, a game has to have good gameplay or you're sunk. But here are some reasons it may just be worth the effort:



- Artistic merit. If video game-making is an artistic medium, we should be able to use it for all kinds of self expression, especially story telling.
- Story and character are tools for entertaining, for touching people's hearts, for making an emotional impact beyond the mechanics of the game, beyond a high score, a game's interactivity or its puzzle.
- The value of character IP. Additionally, as our industry matures the opportunity to build character I.P. has obvious financial upside. i.e. Lara Croft, Tomb Raider.
- Great Narratives create strong brands. From Charles Dickens' Victorian masterpieces, to today's Sopranos or Lost, serialized fiction is infamous for sucking in its audience and keeping them coming back to find out what happens next. For Casual Games especially, where a thirty to sixty-minute free trial often stands as an obstacle to a purchase, the use of story might just be the holy grail of conversion, motivating users to keep playing.

## ii Picking the right theme

Casual Gamers love Aztec themes, right?

A rich game metaphor provides back-story, defines the play environment, adds a compelling motive to the player's game goals, and gives the user a deeper feeling of immersion into the experience. What's different here about the casual game audience? The story goes that as video games first began to explore narrative themes, the game-making community chose themes that appealed to game makers. Thus the over-saturation of spaceships, robots, archeological tomb hunters, elves and dragons that blast, shoot, swing or roar their way through the game players' imaginative landscape.

Casual games have for the most part trodden the same narrow territory – space ships, aliens, and elves all make their way to the casual game landscape. When you throw in ancient archeological lost city themes like Mesoamerican (Incan, Mayan or Aztec) and ancient Egyptian, you've covered about half the casual games out there. But over the last two years we've seen games emerge that explore broader narrative themes and more everyday characters. We've seen successful games like Insaniquarium, Huntsville Mystery Case Files, Diner Dash, QBeez2 and Granny in Paradise, all borrowing from popular themes in prime-time television.

Will your casual game's story and characters appeal to the casual game market (many of whom are adult women)? If you're thinking about a game with dungeons, bloody battles, spaceships and robots, you might be going down the wrong path.



Film Noir Theme: Huntsville Mystery Case Files (Big Fish)



Granny and kittens: Granny in Paradise (Sandlot Games)

### ***iii Rules for creating a compelling story***

It's worth noting that a good deal of the following may be subjective. Not all stories work by the same rules. Not all stories will benefit from the same approaches. This section therefore may prove more subjective than say, the section on publishing or distribution models, but here goes. As a game designer there are a host of rules to follow or break as you see fit when telling a story in a casual game.

What's so different about casual games? The main difference is the audience. Whereas in a core game, the audience may forgive bad dialogue, poor character development or no story at all in exchange for super cool graphics and hi-tech themes like shooting rampages, battling droids or half-clothed bikini elves, the audience of casual games is made up of many non-hard-core gamers and the bar for entertainment quality is set not by other games, but by television and film.

Here are some often cited oft maligned rules of thumb for story telling in casual game making.

- Immersion. Create the illusion that the player is in the story world. Take care not to shatter that illusion.
- Keep the technology invisible. Don't remind players they are on a computer. Immerse the user in the narrative world.
- Don't break the fourth wall. Try to avoid breaking the fourth wall (the wall between the characters on stage and the audience – thus avoid lines like "Hello there player! Use the mouse and click and drag that item over here!" – or "type on your keyboard")
- Respect the player's imagination. You don't need to tell the user every detail. They will enjoy the interaction more if you let them participate in how the story emerges.
- Allow for closure. Visual theorists call it "closure" when an artist lets a viewer fill in the gaps of a broken circle or on what happens between panels of a comic book. As a story teller, you don't need to tell every detail of your whole story. Allow the user to fill in the blanks with his or her imagination.
- Start as deep in as possible. Start the story as deep into the narrative as you can, allowing the player to fill in the blanks.
- Waste not, want not. With small file sizes and small budgets, typically you can't afford extensively animated cut scenes or tons of casted and recorded character dialogue. But big budgets and big file sizes don't always mean good story telling.
- Less is more. Brutally edit your dialogue. Write a script for scene, then cut it in half – then take a breath and cut it in half again. It's cheaper and it's likely that your story will be more interesting if less is revealed. Most players don't want to read. If your game has a comic book page, can the player get the gist of the story just by looking at the artwork and not reading any text?
- Naturalism. Save time in writing and casting. Go for naturalism in your casting, dialogue writing and directing. People are used to standards set by television and film. No one wants to listen to an overacted set of affected lines read in a deep hoarse throaty manner. No matter how evil it makes your character sound.
- Action is key. Never tell when you can show. Avoid talking heads with lots of exposition. If you need the characters to convey to the player some information, can

you convey it within a dialogue as the players are scaling a wall, or climbing a tree or...? Animation is expensive, can you use comic strips instead of animated cut scenes? If you're going to animate make sure it's interesting – ask yourself if it's really worth animating?

- Rely on Audio. Audio is typically cheaper than animation as a tool for storytelling. Sometimes music, voice and audio sound effects to help immerse, set a mood tell the story. A story is different from a list. If you're going to tell a story in your game, keep in mind that a good story is not just, "first this happened, then that happened, next a third cool thing happened."
- Foreshadow. A good story has narrative structure, offers thoughtful foreshadowing during early stages of the narrative to set up expectations.
- Use a "Third act twist". A good story then plays with or twists those expectations (usually in the "third act" or final act of the game).
- Seamless integration. Maybe the hardest part: Can you weave interactivity and story around each other without making the story feel contrived or tacked on as an afterthought? Playing the game should feel like you're making the story unfold further. Can the story change depending on the player's choices?

#### ***iv Defining Characters: Some starting points.***

Do you know who your character is? Knowing who your character is will make it much easier to write natural dialogue and compelling stories. Take the time to define your characters' back story, their likes and dislikes, family history, strengths and flaws. Their pet peeves. Their quirks and catch phrases. If you do this right, you will know much more about your character than you ever will be able to tell in a small file game but it will make the dialogue flow freely if you can really channel your character and see into their inner world.

Ways to evaluate the strength of a character:

- Do you care about you character? A lot? If you don't care about who your character is, then don't expect the player to care who he or she is!
- Draw on what you know. Don't just add a character because you think you need one.
- Draw on your own experiences and your own world to create more believable characters that people will care about and understand.
- Is your character someone with whom players can identify?
- Is your character aspirational? Someone players might identify with or aspire to be?
- Work with your artist. The character artwork should tell you as much as possible about that character (while avoiding stereotypes).
- Test it! What we think is appealing may not be appealing to our audience. Draw out multiple versions of your characters and run them by potential players.

### Casual Games that Use Characters



Jill from Cake Mania (Sandlot Games)



The QBeez from QBeez 2 (Skunk Studios)



Daphne from Mystic Inn (Big Fish Games)



Flo from Diner Dash 2 (Play First)

### 1.7 Using Sound and Interactive Audio to Enhance Gameplay

Sound can make a powerful addition and perceived added value among casual game consumers. Audio can add emotional context for a player to excite and involve them in the game. Because of the repetitive nature of casual game play as well as typically less narrative content than a console title, audio is an area that allows developers to keep the gameplay fresh and interesting as the game progresses. Audio is typically broken down into three categories:

- Music
- Sound Effects
- Dialog

Most games in the casual game space have at least some basic set of sound effects and music.

The following table illustrates the many areas where sound adds value to a game.

Use of Sound	Description
Game Environment and Narrative Context	Similar to a visual backdrop, music and sound effects can add emotional and contextual elements to the game. Usually this is done through music, but sometimes through sound effects.
User Interface	Typically these are just simple rollover, navigation, and click feedback events that are in line with the overall theme of the game.
Game Interface	These are similar to the UI but are in game event related. This is a good place for some elements of randomization if possible.
Game Feedback	Sound effects usually accompany scoring and reward events, as well as any negative feedback in the game.
Easter Eggs and Surprises	These elements help to engage users over the long term. Surprises with the audio can be an inventive way to keep the game fresh and new with content.
Music as a Gameplay Element	Occasionally the music itself is a gameplay element; this is most obvious in a rhythm-action game, but also could mean that at the end of a level the music branches to warn the player.

Largely due to size and budgetary constraints, the biggest challenge of audio is avoiding repetition in gameplay. Randomization techniques and other adaptive strategies should be used to increase replay ability.

The following table illustrates some examples of additional audio techniques used in casual games.

Technique	Description
Random Music and Narrative Based Music Branching	Music is usually broken into small two- to eight-bar chunks that are placed end to end randomly or in some strategy that fits the gameplay. These branches can be tied to gameplay elements, when you move on to the second part in a level branch to this music seamlessly.
Sound Effects (SFX) Playlists	SFX that either cycle through a playlist or randomly pick from a list is sometimes used for game and UI events.
Split Track Music Playlists	Music can sometimes be broken down in a way that allows the game to change one part while the other continues. For instance having the rhythm split out from the melody might allow you to save not only space but allow the parts to change randomly for more variation in your game.
Real Time Tempo Changes	Increasing the tempo during parts of the game might be an effective way to increase emotional tension, and add excitement to the game.

Additive and Subtractive Music Mixing	Another musical adaptive technique when the music is split out (i.e. melody, rhythm, chords) is to add and remove parts directly based on gameplay.
SFX Customization	Another unique way to deliver personalized gameplay experience is to use things like dialog synthesis to relay the persons score in real time.

With randomization it's important to remember that randomization should be placed in areas where the user expectation is low. Many events need a specific sound effect that never changes because it's so integral to the game experience and to do so would confuse the player. Usage of randomization in key places might confuse the player.

The unique nature of the casual market also allows composers and sound designers with break boundaries with their approach to themes and genres. Since there is typically a shorter chain of approvals in casual game development, it's easy to push through new and interesting genres of music that have never been heard in a game before. Unexpected genres are commonplace in the industry including music from the 50's golden age of TV, modern, abstract, cartoony, early 80s, Star Trek, therimins, whatever. The ability to break out of the normalcy allows a game to find an audience because it is unique and unusual.



Luxor: Amun Rising (MumboJumbo) uses a musical score appropriate to the Egyptian theme of the game.

As with the other areas of game development, technology and size will inform and sometimes limit the amount of things that are possible with audio. Many technologies and middleware have specific constraints as to what is possible with music. See the technology section of this white paper for additional notes on sound technologies.

## 2. Game Mechanics

The contributing factors of distribution, technology, and audience have shaped the current casual content offering. It is for these reasons that the casual gaming industry does not consist primarily of first-person shooters and tactical simulations. Instead, genres have emerged that address the needs of the casual games market.

As mentioned in the introduction, casual games have been largely dominated by a surprisingly small number of game play styles. Puzzle games make up the largest single group, followed closely by a variety of simple arcade games, word games, and classic card and board games. Within these larger genres, most casual games segment further into a handful of specific types.

The following game mechanics are discussed in this section:

- Matching Games
- Finding Subsets I (Puzzle Games)
- Finding Subsets II (Word Games)

- Shape Manipulation
- Casual Sports Games
- Break-Out Variants
- Card and Parlor Games

## 2.1 Matching Games

**Description:** Players are faced with a grid of a limited variety of objects. The objective of the game is to swap, drag, shoot, or transform these objects to create sets of two or more, which then disappear for points. These games often contain "power-up" objects that clear larger parts of the grid or award bonus points for sets including them.

- Examples:**
- Chuzzle (PopCap Games)
  - Luxor (MumboJumbo)
  - The Da Vinci Code (Sony)

Popular Examples of Matching Games:



Chuzzle (PopCap Games)



The Davinci Code (Sony)

## 2.2 Finding Subsets I (Puzzle Games)

**Description:** Players are given a number of objects, a timed end point manifested as a clock, and sometimes a steady increase in the number of objects. The player's role is to find sets within the field objects based on a particular criterion (similar color, shape, etc.). Correctly finding and selecting these sets earns the player points and either delays the end point or advances the user to the goal.

- Examples:**
- 10 Talismans (NevoSoft)
  - Magic Match (Codeminion/Oberon)
  - QBeez 2 (Skunk Studios)

Popular Examples of Finding Subsets I (Puzzle Games):



Magic Match (Codeminiion/Oberon)



QBeez 2 (Skunk Studios)

### 2.3 Finding Subsets II (Word Games)

Description: A notable specific case in this genre is the word game. In this case, the rules of the game's language determine a correct set. Points are often given for correctly spelled words, with greater rewards being credited to players who spell longer words and/or words with rarer letters.

- Examples:
- Acropolis (Gamehouse)
  - Babel Deluxe (Zylom)
  - Pat Sajak's Lucky Letters (U-Click/Playtonium)

Popular Examples of Finding Subsets II (Word Games):



Acropolis (Gamehouse)



Babel Deluxe (Zylom)



## 2.4 Shape Manipulation

**Description:** Players are presented with a empty container divided into different shapes, and a series of pieces that can fit into that container. The player's goal is to fill up the container by picking up, rotating, and placing pieces inside the contact so that no pieces overlap and no empty spaces remain.

**Examples:**

- Mosaic: Tomb of Mystery (Reflexive)
- Puzzle Express (Hipsoft)
- Runic One (Puzzle Lab)

Popular Examples of Shape Manipulation:



Mosaic Tomb of Mystery (Reflexive)



Runic One (Puzzle Lab)

## 2.5 System Management

**Description:** The player is put in charge of a small ecosystem of objects that interact in a variety of ways. The player may add, remove, or alter objects in the system to create particular effects and thus earn points.

**Examples:**

- Cake Mania (Sandlot Games)
- Plantasia (gameLab)
- Fish Tycoon (Big Fish Games)

Popular Examples of System Management:



Cake Mania (Sandlot Games)



Fish Tycoon (Big Fish Games)

2.6 Break-Out Variants

Description: The player controls a paddle, and uses the paddle to ricochet a ball into a set of blocks. The goal is to clear the screen of blocks. Power-ups alter the core game in a few ways including speeding up and slowing down the ball, making the paddle sticky, or increasing the number of balls on the screen.

- Examples:
- Bricks of Atlantis (ArcadeLab)
  - Magic Ball 2: New Worlds (Alawar Entertainment)
  - Shattera (Alexey Saenko)

Popular Examples of Break-Out Variants:



Bricks of Atlantis (ArcadeLab)



Shattera (Alexey Saenko)

## 2.7 Casual Sports Games

Description: Simple sports games that are very forgiving and generally accessible to the widest audience possible in terms of game controls and game objectives.

- Examples:
- Backspin Billiards (Pixelstorm)
  - Redline Rumble 2: Detonator (Atom Entertainment/Richard Smith)
  - Saints & Sinners Bowling (Large Animal Games/Oberon Media)

Popular Examples of Casual Sports Games:



Redline Rumble 2: Detonator (Atom Entertainment/Richard Smith)



Saints & Sinners Bowling (Large Animal Games/Oberon Media)

## 2.8 Card & Parlor Games

Description: This genre includes a variety of traditional playing-card games and parlor game favorites.

- Examples:
- Ancient TriPeaks (ToyBox Games)
  - Mah Jong Escape: Ancient China (Playtime)
  - Mahjong Garden to Go (Pogo)

Popular Examples of Card & Parlor Games:



Ancient TriPeaks (ToyBox Games)



Mahjong Garden to Go (Pogo)

## 2.9 Other Arcade Variants

Description: This genre includes a variety of traditional arcade-style gameplay.

- Examples:
- Cosmic Bugs (Retro64)
  - Gold Miner Vegas Edition (Intermix Media)
  - Tropix (Super Robot Brain)

Popular Other Arcade Variants:



Gold Miner Vegas Edition (Intermix Media)



Cosmic Bugs (Retro64)

While this list is by no means exhaustive, even a glance at the games available in the casual games marketplace reveals a preponderance of games that fit within this handful of categories. There is also a strong degree of overlap among the games within a single category, such that many games have nearly identical mechanics and are only differentiated by narrative context and visual style. In fact, there is a deluge of derivative games in the casual game market that seek to capitalize on the success of a proven mechanic. As the casual game industry grows, new game play, design and genres will emerge and evolve to the next level.

## 2.10 Innovation beyond the Tried-and-True

Innovation in the casual games field must still adhere to the principles of user interactivity and audience expectations as described in the previous sections. The casual gamer is simply a different user group than the hardcore gamer, and the kinds of experimentation and approaches that appeal to the latter may not be successful with the former. Of course, there have been examples of games that have provided different gameplay styles that have also been successful in the casual game market. "Diner Dash" and "Tradewinds" are both games that have previously unseen gameplay for casual games. They both have been commercially successful. Web-only games have seen even more radical experimentation. Games such as Grow ([www.eyezmaze.com/grow/v3/index.html](http://www.eyezmaze.com/grow/v3/index.html)), Samarost ([nlp.fi.muni.cz/~xsvobod4/amanita/samorost/intro.html](http://nlp.fi.muni.cz/~xsvobod4/amanita/samorost/intro.html)), and Squares 2 ([www.albinoblacksheep.com/flash/squares2.php](http://www.albinoblacksheep.com/flash/squares2.php)) all demonstrate the ways Web-only games can push the boundaries of typical play patterns. Thus, there are possibilities with the

constraints provided by a casual audience for new play styles, both as modifications of tried-and-true genres or as more dramatic experimentation.

### **3. Key Design Elements for Advergames**

The challenge for an advergame developer is not only to make a great game, but to design it in such a way that it accurately and positively promotes the brand messaging, and ideals. This will also have to be done on an often extremely tight deadline, with key decision makers completely uninvolved until the very end.

#### **3.1 Brand Messaging**

Brand positioning is the most important part of an advergame for the end client. The developer needs to find out as early as possible what the overall theme, and messaging is going to be for this particular campaign. The game will usually be part of a larger ad-run incorporating banners, TV spots, print ads, and more. Keeping the game on message will be key to the overall success from the client's perspective.

Ideally, the game will be designed around the various aspects of a brand, creating a unique experience custom-tailored to the product or message.

#### **3.2 Scheduling**

Scheduling is often one of the biggest issues affecting advergame developers. The games are generally developed as part of a larger ad campaign, which incorporate ad buys, TV spots, and print runs with very long lead times. As a result the deadlines are typically very hard deadlines and cannot be missed. In the agency world, a development time of two to three months is long, and six months is almost unheard of. This must be considered as the scope of the game is generally more limited by time, than it is budget.

#### **3.3 Campaign Goals**

When creating an advergame it is important to consider the overall campaign goals. While a casual game generally has the goal of earning revenue, advergames fit into marketing budgets, and do not need to focus on revenue. However, they will need to meet other, possibly more difficult, goals.

##### ***i* Large number of plays**

If the goal is to reach as wide of an audience as possible, then distribution and compelling content is the key. The game will need to be extremely easy, allowing the player to play the game effectively in no more than thirty seconds. The most successful games here are generally very short, lasting no more than two minutes, and will also often include humorous elements to encourage viral distribution.

##### ***ii* In-depth Experience**

If the goal is to create a very in-depth brand experience, then the developer will need to focus on depth and variety of gameplay. In these instances, the game will still need to be easy to play, but the extended features will allow for more gameplay variation. It is not unheard of for advergames to be so engrossing that they receive thirty to sixty minute

average play durations. With proper save games, and somewhat episodic content, it is possible to garner up to two hours or more of average playtime across an extended advergaming campaign.

Another great example of in-depth experience is Alternate Reality Games (ARGs) which typically spread a mystery to be unraveled by a massive global player base. More information about ARGs can be found on the IGDA's ARG SIG website ([www.igda.org/arg/](http://www.igda.org/arg/)).

#### **4. Key Design Elements for Casual Single-Player Games**

The challenge for a single player game developer is to keep the player engaged and always wanting more. It is one thing to entertain the player enough to keep him or her interested in playing the free version. It is much harder to get the player so involved in the game that they are willing to pay to continue playing beyond the trial period or pay to upgrade to the "premium version". Even then the job is not done. The premium version had better deliver a good value to the player or the player will become an unhappy customer very quickly. To accomplish this, the game must have depth, measure and show progress, and keep the content seeming fresh for many hours of play.

##### *4.1 Aspirational Fantasy*

Aspirational Fantasy is a design element used as a recourse to immerse the player into the game. It's the environment around the gameplay that makes the player relate to the game. A very good example is the environment set in Aloha Solitaire, where the player sets sail to different islands and plays cards under a palm tree. The sound of a calm sea is heard while playing. This mood set in the game takes the player away from their daily routine, and helps the game to be considered by the player as a "place" to forget about problems.

##### *4.2 Optional depth*

Optional depth, sometimes called Z-Axis of Gameplay or Secondary Strategy, is an element of gameplay which is not part of the core gameplay in the game, but it adds depth to the gameplay by having the player be able to choose different paths to the end goal, or make the user have more than one goal in the game (completion of a task and scoring as many points as possible). Here are some examples:

- In Diner Dash, matching customer colors with seats gives the user additional points. Since points are the end goal in the game, these bonus points help the player achieve the end goal quickly.
- In Feeding Frenzy, catching the bubbles give the player additional points (stars) or elements that enhance the player (thunder = speed bonus). The points in the game help the player achieve higher scores per level, but they don't help to achieve the end goal. The elements that enhance the player helps them achieve the end goal faster.

##### *4.3 Showing Progress*

Most games have a very simple set of rules that define the core play mechanic. Included in that core play mechanic is the definition of a win scenario for the "level". (or round, stage, day, session, match, board, puzzle or what have you) Beyond that, most games have some type of "meta game" that a player progress through each time a "level" is completed.

Having a clear way of tracking and displaying long term progress in the game can be crucial to keeping the player engaged and wanting more than just what is offered in the free trial version. Most games should have some kind of end game scenario or a way to “beat the game.” This should extend way beyond a single game session.

Many games use the simple metaphor of “levels” where you may have to be fifty levels to finish the game. Other games track a character through an evolving career path or story. No matter how it is themed, it is important to show the player a long term goal, and then track through progress towards that goal visually. The simplest way to do this is with a “world map” similar to ones used in *Cake Mania*, and *Feeding Frenzy*. After each level, session, match, chapter, or round the player should see an updated world map to see how far they have come and how much is left to go to get to the next milestone or to the end goal. Progress should be carried over from one game session to the next rather than making the player restart from the beginning each session.

Some games succeed at tracking the player’s personal increase in skill on a high scores lists. These games may require to player to start from the beginning each session and try to get a better score than last time. This is quickly becoming an outdated play mode that is rarely successful in modern casual games. It is usually better to use check point or automatic save games to track a player’s progress along a map through many levels. Of course, these “levels” and “maps” can be completely re-themed and disguised as something entirely different. *Big Kahuna Reef* tracks and rewards progress by showing species of fish that have been discovered, *Chuzzle* makes extensive use of trophies to track progress, *Mahjong Towers Eternity* tracks a player through a series of character class upgrades, and although *Cake Mania* has a world map, many players use their collection of equipment upgrades as the primary way to setting goals and tracking progress. In all these cases, it is not only important that the player advances through a sequence. It is equally important that the way of displaying the progress makes it clear how far along the sequence the player has come and how far is left to go. And it is also good to have intermediate milestones along the way with additional rewards.

#### *4.4 Designing for Upsell of Premium Content*

Game developers who create premium versions of content (played or accessed online) hope that customers will pay money for these titles. To persuade customers to pay money for titles, developers must offer a compelling play experience that customers feel is significantly richer than the free web game demo equivalents. Since players will frequently experience the game via an online version first, and then download and install the game on their PC, there must be significant motivation for a customer to proceed to the downloadable version. Customers are typically motivated by one or more of the following reasons:

##### ***i Deepening Gameplay Over Time***

During the trial period, the gameplay convinces the player that one will want to continue to play this game over and over. Developers want to ensure the customer isn’t questioning, “It was fun for an hour, but will it get any better if I purchase the full game or is it just going to be more of the same?”

##### ***ii Higher Quality Experience***

Premium content offline play is usually full-screen and advertising-free, typically with enhanced graphics and sound.



#### *4.5 Using the free trial version to close the deal and make a sale*

With downloadable games, the ultimate goal is to convince the player to purchase the full game. Sometimes this can be accomplished solely with a web version, but usually a player will evaluate a free downloadable trial version of the game before making a purchase. The trial version must be carefully designed to motivate players to purchase. It's a difficult balancing act to make the free trial version compelling enough to motivate the customer to buy the game without giving away so much gameplay in the free trial that the player has had their fill.

##### ***i Content Limiting***

Many games will include the first X number of levels (or the first mission pack) in the free trial version and require a purchase to play additional levels. Typically a player can re-play these demo levels indefinitely. This is how demos of retail CD-ROMs games are usually designed. Some downloadable game developers also use this model for their games. Content limited is especially hard to do well in some kind of games such as action games. These playable games make it difficult for the designer to decide how many levels to give away. Including too many levels in the trial version can leave little reason for the player to buy the full version since the trial is satisfying, but not including enough levels in the demo to give the player an experience of what makes the game fun, can damage purchase potential. It can be easier to make effective use of content limiting in a story driven game or any game where the player previous progress is carried over into the next level.

##### ***ii Feature Limiting***

A common way to differentiate the trial version from the full version is to lock out some features in the trial version. The player must purchase the full version to unlock these special features, such as: Internet high score posting, level editors, expansion packs, or special content such as power-ups or alternate game play modes. By allowing the player full access to many levels, the player can fully appreciate what makes the game fun, but those few locked out features can motivate the player motivation to the purchase.

##### ***iii Time Limiting***

Most large distributors prefer that all their games are limited to a specific amount of play time during the trial period. After sixty minutes of play, the game becomes unusable unless the player purchases the full version. This can be a very effective way to give the customer a full, featured trial play of the game. Players are allowed to experience every feature and play every level they can reach during the trial period. This gives the player a realistic sample of what the purchased game will be like and allows the game designers to show off his best levels to convince the player to buy, without worrying about giving those levels away permanently; the player will only be allowed to play the levels for a short amount of time.

The Digital Rights Management (DRM) systems and wrappers used most portals only allow for time limited demos. It is usually not possible to implement feature limiting or time limiting for the version of the game distributed through these channels because the same version of the game is used for the trial and the full version and it is not always possible to determine if the game is running in trial mode. Designers of downloadable games should take this into account in their game design and make sure the game is a good sixty-minute experience. Some DRMs do provide an optional SDK with an API to communicate with the

DRM so that it is possible to implement content and feature limits. But the sixty-minute trial will always be the lowest common denominator for some of the largest casual games portals.

## **5. Key Design Elements for Casual Community-Based Games**

By definition, "Community" means "A group of people having common interests". As for the casual games space, community is an oft used term which is defined differently depending on who you are talking to. The term "community" has come to include any or all of the following:

### In-Game

- Allowing players to play against each other in the same game room – Multiplayer
- Allowing players to play a game (self paced) in a room with other people
- Displaying a player's score to everyone playing in the room
- Allowing user interaction during game play (full chat or canned messages)
- Announcements of player achievements during game play (amazing feats or events of extraordinary rarity).
- This can be done via Chat or other system message area Tournaments or other incentive-based game play over a fixed period of time

### External (Post Game Community Features)

- Rooms which are set up according to geographic areas or special interests in order to put people in rooms with other people who have a common interest
- Recording of a users score to a Leaderboard or High Score Board
- A player ranking system – usually based on a complicated mathematical formula to determine a player's skill vs. another's. This is generally used to gauge skill for setting up head to head game play.
- Archiving and displaying Top Scores for different periods of time (Day, Week, Month, Lifetime, etc)
- Having a Hall of Fame system to recognize long term outstanding achievements
- A Message Board or Blogging area where players can chat about their favorite games
- A system in which a player can invite another player to compete with them
- Players can win Coins, Points, Tokens, Trophies, Talismans, Badges, Dots or some other kind of award during the course of game play
- Players are rewarded for spending extended periods of time playing games (Longevity award)

Most community based features are based on different ways of displaying (and recording) player related data. Such data as scores, time spent, achievements and other player based information can all be displayed on a web site. Game web sites will generally use a combination of the above features.

No matter what, it's the people playing your game who make the community great. You cannot CREATE a community; you can only foster the growth of a community that is willing to grow.

### *5.1 Head-to-head play*

Competition is the key motivating factor for any head to head game. Most common are your 2 player "standards" such as Chess, Checkers, Backgammon and Billiards. Mahjong games have a renewed popularity in the casual game space these days, however you do not see many multiplayer versions online. Your multi-player parlor style games also appear like Hearts, Spades, Gin, 7/27 and any Texas Hold Em style game. Thanks to the rise in media attention these days, those sites who offer Hold 'Em have seen an upturn in game play. And you have your multiplayer casual games such as Team Bingo, Slingo, any of the multiplayer games at Pogo, MSN, Yahoo! or Games.com. One thing you do not see too much of in the Casual Game space are Cooperative Play games in which players are required to work together to accomplish goals.

### *5.2 Leaderboards and Rankings*

As part of any community based game, an efficient Ranking system is a must in order to keep players coming back. Not to be underestimated, most common leaderboards (like a high-score table) or ladders don't scale well, and a poor implementation can actually hurt the game if players are not able to see that they are progressing.

Sometimes the best approach is not to create a unified ranking, but give players a way of identifying who are the best players in the game. That way they can always find who to look up to.

### *5.3 Viral Marketing*

Word of mouth is becoming an increasingly effective marketing tool. The efficiency of viral marketing is unparalleled because all it takes is to paste a URL in an email or an instant messaging window. Word of mouth promotion can be encouraged in a game by giving players the tools and incentives to do it.

If the game has a virtual economy, incentivize players to get their friends to sign up. However, be very careful about the amount of the incentives as this can break the economy flow. Also, be sure to have security measures in place so to prevent abuse.

Another effective tool is a feature to notify friends about achievements within the game, to help incentivize competition. An example would be a feature that automatically notifies friends that someone beat their high-score.

### *5.4 User to User Communication*

In order to build a strong community of players, users need to be able to communicate effectively. Today, most games stamp chat boxes on all parts of the lobby and inside the game. However, that may not be the most effective solution. Room lobbies have proven to be a nightmare for community administrators, as they are typically flooded with hundreds of messages a minute when the room is full.

In-game chat is effective when the pace of the game lets a player relax and be able to talk. Players don't really talk in real-time games, but the chat window among players of the same team is an effective way to coordinate strategies.

Buddy lists are extremely important in any community based game, since creating ties between users keeps them coming back.

### **C. Production Issues**

---

The lower financial stakes of casual games make it possible to explore a variety of innovative and unusual topics and game mechanics. However, even a small game can go over budget and schedule without proper planning and management. Developers of casual games face numerous production constraints.

#### **1. Process Differences**

The differences between casual games as compared to consoles and other more traditional game platforms (e.g., GBA, PS2, and XBox) are the following:

##### *1.1 Smaller teams*

The smaller project size of casual games usually means smaller development teams in which each team member must wear multiple hats. The game designer or programmer may double as producer; the concept artist, lead artist, and art director may actually be the same person; and everyone on the team will put in time as a QA tester. As a result, management overhead may be reduced considerably and breakdowns in team communication can be less of a risk. However, projects may be more complicated to plan and budget when a single person is doing multiple tasks. While the smaller scope of Casual game projects would seem to make the project management simpler, in reality this simplicity is offset by the fact that most developers of such games need to produce several games concurrently in order to meet their cash flow requirements.

##### *1.2 Working directly with distributors to publish your content*

Although the distributor/developer relationship in the casual space is often likened to the publisher/developer relationship in the traditional retail space, they are actually very different, since distributors in the casual space do not typically share the financial risk of development. The developer assumes all of the up-front risk, while the distributor is typically not involved until much later in the development cycle when the game is nearly complete. However, even at this stage, the distributor can provide invaluable feedback and beta-testing resources to help guide the final tuning and polishing of the game. For casual titles that are being self-funded by the developer, there is often no specific budget, but instead a more flexible number that evolves out of a compromise between the cash-flow reality and a sense of the product that is needed to compete in the marketplace.

Developers working on self-funded original titles may be able to enjoy a more organic, iterative process of creating games. When a publisher's financial oversight is replaced by the developer's, the development process can be more flexible, but developers need to be careful to take advantage of this flexibility rather than fall victim to it. Developers also need to be aware of the additional effort involved in working with distributors. Often, the distributor will require that their branding be incorporated into framing and peripheral

elements of the game. For web games, developers may be required to integrate distributor features for server-based services such as chat, rewards/loyalty points, high scores, etc. For casual games, additional time will also be needed for wrapping the game within the distributor's DRM solution and possibly for producing promotional graphics and copy that can be used on the distributor's website. All of these things can add days to the end of the production cycle and should be planned for by the developer.

For casual games that are being developed to promote a particular consumer brand or product, budgets and timelines are usually immutable and therefore call for a more formalized management approach. Also, a significant portion of the development time is often consumed in incorporating the client's brand identity and getting approval for the brand's representation within the game.

### *1.3 Self-publishing*

Whereas in the traditional videogame space there is a clear-cut between the publisher (i.e.: Activision), and the distributor (i.e.: Target), in the casual space the line is blurry. Online portals behave as the distributors and as the publishers in many cases, and this is a trend which will grow in the next couple of years. On the other hand, when there is not an exclusive distribution agreement with a portal, nothing stops a developer from publishing their game to their own web site. Attracting traffic is a big, and sometimes costly, challenge. Please refer to the business model section of this paper for more information.

### *1.4 Working with Clients on a Work-for-Hire Basis*

In many cases, developers tend to own their Intellectual Property (IP), whereas traditional game developers are often working for a larger publisher and/or with big licensed names on a work-for-hire basis where they do not retain rights to the IP they develop.

### *1.5 Working with a Publisher*

There are good definitions out there on what a Video Game Publisher is, so that is not discussed here. However it is worth mentioning that there are differences between traditional Game Publishers and Casual Game Publishers. One of the reasons is because most Casual Game Publishers have been formed by successful independent developers or by people who left the traditional Game Industry to work on Casual Games.

The financial conditions in a typical contract tend to be more beneficial with Casual Games Publishers than traditional ones. And as the volume of sales of Casual Games increase, Casual Games are going to be a more attractive proposition for the small game developer.

Projects are shorter and milestones are less, so developers move from project to project much more quickly. However, the risk of getting a project killed still exists, so small game developers should be careful not to have all their eggs in one basket and have more than one project at any one time.

With the Internet, there is always the question of the need of the Publisher role. But the truth is that for newcomers is becoming very difficult to get their games on the major portals, so only a few companies today have the infrastructure and the deals in place to get wide distribution. But most of these companies would work with a developer if they believe the game will be successful, and they will offer a developer a much better deal than what a

publishing deal would look like. This type of deal is usually called "exclusive distribution deal", and some times include some type of advance. In this particular case then, the game developer needs to fund most of the development of the game, as the publisher typically will not sign a exclusive distribution deal if the game is not almost complete.

## **2. Project Constraints**

### **2.1 Money**

Games developed for the casual market are typically smaller in scope than retail PC and console games. Project timelines are much shorter and budgets are smaller. Whereas it is not uncommon for a PC or console title to have a development budget of \$5 to \$10 million (US), casual games are typically produced for approximately \$50,000 to \$150,000 USD. Publishers rarely pay advances for original downloadable titles, so the risk usually lies with the developer. Smaller budgets mean smaller teams in which each member must possess a diverse skill set. The implications of this are discussed in more detail below under "Process Differences".

Advergaming typically have a fixed budget that is restrictive relative to the desired deliverable. The clients who outsource these games are looking to get as much out of their budget as possible and may not have a particularly deep understanding of what it takes to develop a game. Skill games usually take the form of lightweight Web playable games and have some additional constraints.

### **2.2 File Size**

File size is an issue wherever digital distribution is concerned, and the mass-market audience targeted by successful games in this space is especially sensitive to large downloads. The most downloaded titles are typically less than fifteen megabytes (15MB), and there is a sharper drop-off at thirteen megabytes. This is particularly relevant to those publishers who are looking to extend the lifespan of titles in their back-catalog; titles that were originally designed for retail distribution are usually quite large in file size. The size constraints for web games are much smaller, with a fifty to five hundred kilobyte initial download being the norm.

### **2.3 Technology**

While hardware support is a concern for anyone developing computer games, it is particularly important for the casual market, where players typically are not interested in buying special equipment to make games run better. Developers should assume that their games are being played on machines that are a few years old and that were probably not even state-of-the-art at the time of purchase. Sound and video cards are not as robust, processing speeds are slower, etc.

Since casual game budgets are not usually large enough to support the development of custom technology, many casual game developers use higher-level middleware such as Director, Flash, Torque, or the PopCap Framework to develop their games. With these environments, developers can produce both Web and Downloadable versions of their games from a single source. It should be noted that reliance on middleware does mean working within the limitations of the latest version of the software. These technologies are discussed in greater detail in the Technology section of this White Paper.

## 2.4 Deadlines

The casual game market is less sensitive to seasonal purchasing patterns than the retail PC and video game industry. While this creates an opportunity to make the games highly polished before they are released, developers should be careful not to invest so much in the polish cycle that the games become unprofitable. Games that are created for a client, such as advergames, typically have a much more rigidly defined schedule, especially if they are tied to an external event, such as a movie launch or product rollout. Seasons and holidays have a small influence in overall sales and traffic. It is better to spread out releases over the year to secure launches with distributors and portals.

## 2.5 Publishing & Portal Needs

Each publisher or portal may have different needs. One way to quickly understand what a publisher or portal desires is by reviewing what content is heavily promoted on their web sites. If possible, the best way is to talk directly to the portal or games site management. You never know how many developers are making another clone, and by the time you submit your game to the publisher, that publisher may have another ten games that are very similar.

Try to always provide something new. Revolutionary ideas can be risky, but very rewarding. At the same time, small improvements to proven formulas tend to be very successful as seen in *Cake Mania* (Sandlot Games).

## 2.6 Builds and DRM Issues

Unfortunately, there is no standard in the way builds are submitted to publishers and portals. Most likely you will have to create ten versions of your game for ten distributors who will each have their own requirements. It's very important to abstract your calls and have separate functions for different DRM calls (see below on the DRM section). It's the same case with Publisher's and Portal's logos.

Some DRM requirements are:

- Wrapper friendly exes (no self modifying),
- Integrating DRM APIs,
- Making games patchable (regardless of multiple wrappers)

## 2.7 Community Management

Once people begin to congregate and because of the freedom / anonymity of the internet, your player's actions need to be closely monitored to ensure that everyone is having a good time. There are several factors you can employ for sites that permit communication between players:

- Have a rock solid Terms of Service – If there are no rules pertaining to your community, then it is mob rule. The TOS for your site should be no different than a software EULA. It should specifically spell out what players are allowed to do, what they are not allowed to do, and what will happen to them if they break the rules.

- Monitor All Communications– Either record it in logs, or allow people an opportunity to report offensive chat. Since people spend most of their time in game rooms playing games, this is where most of the problems will happen. The most common issues are:
  - Vulgarity (Chat that is obscene or sexual in nature)
  - Room Harassment (Scrolling and being obnoxious)
- Personal Harassment (Directing attention at a particular player)
- Use Judicious Caution – Communities that get close will start to share very personal information with each other. For the most part, your staff will not be trained psychologists or have the expertise necessary to deal with serious issues. People with problems should be directed to their local authorities or other experts (suicide hotlines and other such organization's information should be kept handy to pass to members as needed)
- Provide Adequate Support – Every different person has a different person and a different problem. Not all people will be able to fix a problem by reading an FAQ and surprisingly, many people are not computer literate enough to follow the directions. Issues you will experience range from web based issues, to complicated technical problems. Communities are going to require support and you need to be adequately prepared to deal with those issues or people will go elsewhere.
- Listen to your Community!! – Your community has developed because you have great games and have offered them an opportunity to express themselves. They are going to tell you what they like. They are going to tell you what they don't like. Obviously, all suggestions cannot be taken seriously, but the feedback you get from a community is invaluable to making a product better. This is the only way your community will grow and stay healthy.
- ALWAYS have an Email address where people can reach a human being and get a response.

There are several different ways in which you can monitor your community, but the most common are:

- Automated Methods – Various scripts and jobs can be run to automate several community functions. This could include automatic Email support (or a solid FAQ section). This could include a series of server based jobs which monitor logs for illegal scores or offensive chat. None of these solutions, however, deal with the human element which most communities survive on.
- Employ Community Managers – Put people on the payroll, or subcontract to remote staffers who are responsible for facilitating chat, fostering a good environment by reminding people of the Terms of Service and actioning those people who break the TOS. These people need strong training and should be given explicit direction as to the level of power they should employ when dealing with the community. You do NOT want to lose players from a Community Manager on a power trip. They are there to facilitate the community, not police it.
- Utilize Community Volunteers – Sometimes there are people that spend so much time playing community games that you might as well give them something to do. There are many sites which use Volunteers for such mundane tasks message board moderation and chat patrol.



NOTE – Check your local laws to see if there are specific laws about using volunteers to work on your web site.

## 2.8 Summary

Clearly there is no shortage of constraints for the developer producing casual games. However, these constraints can also be viewed as potentially useful design parameters that can help focus the development team (and ultimately the player) on the essence of the game experience they are creating. The key lessons from this section can be summarized as follows:

- Know your customer
- Build the game at the customer's skill level....not your skill level
- Gameplay is king - make sure you have a solid game design
- Differentiation: How is your game better or different from what is in the marketplace?
- Consider building a game with a free web, downloadable and skill-based gaming component. Why leave money on the table but only focusing on one gaming version? Advergaming is probably the one area that is specific.

## 3. Risk Management

The keys to successful risk management are:

- Continuously identify what can go wrong (risks);
- Rank the risks at each stage;
- Create a plan how to handle the risks.

From the very beginning of prototyping to working with distributors after all the work is done, risk management, like any project, big or small, is always a concern. The following section gives an overview for each section and identifies possible hazards that are relatively common in the casual game production process.

The best way to limit your risk is to have a thorough schedule and adhere to it. Concept, proto, alpha, test, beta, test, gold are stages that need to be managed. The main fears are: "Have I spent enough time on any one and/or did I spend too much on any one." Another risk is that at any time during the production process, you may just not be satisfied with the games and may want to stop production wasting time and money for nothing. Finally, you could even end up with a great game but it may not sell. A game not selling well in the marketplace is probably the biggest risk, but let's discuss some of the other possible risks with individual elements and questions that you should be asking yourself before (if ever) you get to that point.

### 3.1 Prototyping

You have a great idea but not sure what to do with it? When does prototyping end and play testing begin? Is this the kind of game that you think is cool or is it the kind of game that you think will sell?

### 3.2 *Play Testing & Beta Testers*

Play testing is just feedback. The most difficult part is interpreting the feedback. Knowing your audience versus knowing your beta testers is critical to judging their comments and making changes to the game. It is critical to gather feedback from one's target audience if designing a word or puzzle game. Conduct user surveys, invite players to your office to play your game and watch them as they play. Don't invite experienced players – but try to find the mainstream casual gamers who will be your primary target audience.

### 3.3 *Production assets*

Did you build that game with the ability to quickly integrate assets, swap and modify with limited risk and time? Waiting for art assets can be a time sink but should not stop the production process.

### 3.4 *Quality Assurance (QA)*

Even before prototyping, you need to ask yourself: Who is the audience and what kind of machines are they using? Who or what is going to do your QA and what are their qualifications? Do they have the right tools to efficiently cover your needs?

- Testing guidelines such as how to conduct a usability study
- Types
- Compatibility testing
- Looking for bugs
- Usability testing
- Play balancing

### 3.5 *Tracking Versions*

Keeping track of which versions has which fixes/art/brands/upgrades might seem time consuming but at the end it will save you time, money and frustration.

### 3.6 *Good Client Relations*

Good communication with your clients will help you not only build a better game but also strengthen your relationship with the client. Your end goal is to ensure the client's expectations and goals are met.

Distribution partners can help you advise you as you create the game so it will appeal to the partner's audience, and being part of the process helps distributors be excited about your title so that when it's finally finished, they are ready (and motivated) to properly market and distribute it. The first time working with a distribution partner might be frustrating as one is learning their requirements but it should be an easier process with future games.

### 3.7 *Technology Choice*

Teams can sometimes go bad, and you have to move the project away to a new team. This is where the technology choice is so important; choosing a technology that only a few

people know how to use limits you in the number of people who can continue with your project. At the same time, choosing a technology that requires development time to create basic elements common in popular frameworks will cause the loss of significant time. Check the Technology Section of the White Paper for more information.

### *3.8 Smaller Project Size, Smaller Teams*

Obviously smaller project size can mean smaller teams. Does your team still have all the basics requirements that it needs to develop the game. Do you have the right artist or developer to do the job? Who will be driving the project if not you? Are you wearing too many hats as the designer/producer/lead developer/business development lead? When teams are small it is obvious that individuals will have to take responsibility for different roles. The question becomes can you manage continuity and capability of the team with the least amount of risk?

## **4. Localization**

Production covers a wide range of localization tasks and contains several areas in which things can quickly get out of control. Production pitfalls can be easier to rectify than technical pitfalls, since they do not need to be fixed by adding new game code. Most production pitfalls can be avoided if the localization process has been thought out thoroughly and planned for accordingly.

### *4.1 Poor Planning*

Poor planning is the number-one cause of difficulties during localizations. As discussed throughout this book, a number of items must be planned for and considered when developing localized versions. For example, localization-friendly code is not something that happens late in the project, it must be planned for in advance.

People often make the mistake of putting localizations on the back burner instead of working on them throughout the production process. Localizations should be an integral part of the production planning because there are many external and internal resources to coordinate. Items such as translations, foreign voice recordings, and linguistic testing all need to be planned ahead of time. These things can't be planned until the developer knows the scope of the localization, which includes what the code can handle, how the assets are organized, and how many assets there are to localize.

### *4.2 Linguistic and Functionality Testing*

Linguistic and functionality testing are time-consuming tasks, and developers have a tendency to underestimate the testing schedule. If the developer has no experience developing localized versions, testing may be thought of as something that can happen on short notice and not take a long time.

Even if the localized versions are using the same code base as the English version, linguistic testing is time consuming. In addition, checking French, German, Spanish, and Italian language assets with four different sets of testers requires a lot of time just to coordinate the bug-fixing process. If the developer is in charge of fixing the linguistic bugs, a lot of time is spent looking over bug reports for all these languages and making fixes.

A general rule of thumb is that it takes a linguistic tester about three to five days to do a first pass on the game, and it takes the developer one day per language to make the fixes. Add in time for making a new build and getting it back to the testers, and the schedule extends quickly.

### 4.3 *Quality of Translations*

The quality of translations is something the developer has no direct control over, but is a problem for localized products in general. If the translator does not thoroughly understand the main concept of the game, appropriate translations will often not be provided. For example, if the game features a character that uses a lot of goofy puns or sarcastic one-liners and the translator does not understand these jokes, the phrases and jokes will likely be translated incorrectly. Therefore, instead of having a goofy character in the French and German versions, the character may appear to talk nonsensically.

Voiceover acting can also contribute to poor quality localizations. If the voice actors and director for the localized voiceover do not understand the context and delivery of the lines, the localized voiceover may not match the game's context. For example, if the characters in the game are human and interact in a realistic environment, the voice acting should match this style. If the voice actors for the localized versions deliver their lines in a cartoon-like, over-the-top fashion, the localized versions will appear less realistic.

Due to cost, travel, or scheduling issues, it is unlikely that the developer will be able to send someone to direct the localized voiceover sessions. However, it is something to consider, especially for high-profile titles. It will be difficult for a non-native speaker to direct voice actors, but if onsite, the developer can assist in the session and provide some context and direction for the lines in general. If possible, the developer should talk with the directors of the localized VO sessions to explain the game and provide examples of how the lines should be delivered.

The developer must provide resources for directing the actors in the localized voiceover sessions. By sending a build of the game to the voice director, the actors can see how their voiceovers will be used in the final product. The final English voiceover files are also helpful because the actors can hear how the line is delivered in English and adapt their line delivery accordingly. The developer can also include context and voiceover direction for each line in the script. Any other resources, such as detailed character notes, pictures of the character, or sample line readings in the appropriate language will also help the performance of the voice actors.

### 4.4 *File-Naming Conventions*

File-naming conventions themselves are not a pitfall; it is the lack of a file-naming convention that can cause problems. Because localized versions require the involvement of so many external people, things can quickly get confusing if there is no standard way of referring to the assets. If the filenames are consistent throughout the project, the developer and translator will better understand what information has been sent.

A file-naming convention for the language assets is also important. If the asset names provide some information about what information is contained in the assets, the developer will have an easier time organizing and tracking what is sent to the translator. For example, a text file named "Game Text" does not provide any information about which part of the game the text is from. If this file is the only text file from the game that needs translated,

then this is not a problem. However, if several text files need to be organized for translation, and they are named "Game Text 1," "Game Text 2," "Game Text 3," and so on, it will be difficult to provide specific context on what the file contains. A good filename is more descriptive, such as using "Mission 01," "UI Text," "Help Text," or the name of the character. Of course, the developer can always open the file to check the contents, but this can get time consuming if there are many of them.

Informative filenames are especially important for voiceover files. Some games contain hundreds of voiceover files, some contain thousands. If they are named in a consistent manner, the developer can tell what the file contains just by looking at the name and will not have to open every file to check the contents.

#### *4.5 Design Pitfalls*

Game design pitfalls are harder to define because game design is subjective. What one person decides is fun to play may be boring to another. However, developers should still take into consideration other cultures and languages in order to design games that appeal to a global audience. Developers do not want people who buy the localized versions to feel short-changed by a game that is very Euro- or U.S.-centric if this is something that can be avoided. However, a global game is heavily dependent on the game's context. As an example, True Crime: Streets of L.A. is obviously a U.S.-centric game based in Los Angeles. Changing the setting for a global audience would take away the game's flavor and context.

#### *4.6 Cultural*

Cultural issues are best identified by having the game design elements reviewed by people from other countries. If the developer works for an international publisher, the company will have access to people in international locations who can provide some basic feedback on how suitable game design and features will be for gamers in their country. They will provide information on cultural taboos, such as language or actions that are considered offensive. They will also know if the game will appeal as a whole to their country and culture. If international resources are not readily available to a developer, feedback can be solicited from international people available locally, such as at college campuses.

#### *4.7 Content*

Actual game content can also create some issues for localized versions. In some cases, the content is not offensive; it is just not the best choice for the localized versions. For example, Shanghai: Second Dynasty is a tile-matching game that contains a set of tiles called "Spelling" in the English versions. The basic premise for this tile set was for players to match pictures of objects with written words, thus the name "Spelling." The artwork for the tiles had the text and pictures embedded into the .bmp files.

This tile set caused some concern when the game was localized into German and Japanese. Both countries wanted the tile set localized into their language and were distressed that a tile set of this nature had been included in the game without a plan for the localized versions. Due to the production schedule, there was no time to design a set of words and pictures for the German and Japanese versions, or to completely redo the tile artwork for each language. Completely removing the tile set was briefly considered, but was rejected because it meant altering the game code and putting the release schedule at risk. In the end, each country decided to just rename the tile set to "English." This was an appropriate compromise since English is something taught to children in Germany and Japan.

Developers must be aware of potential content issues sooner rather than later so they can be fixed before the game assets are in full production. The best way to do this is to gather international input early in the pre-production process and solicit this input through the game's production.

#### ***D. Conclusion***

---

Developers of casual games face a unique set of production constraints, most of which grow out of financial and technical limitations. However, these constraints can be embraced by the optimistic developer and used to guide a more focused development process. This focus is also the first step towards managing the risks associated with game development for this market. Though the budgets are small compared to console and PC games, the risks are still significant for the small developer and require continual vigilance and creative management solutions.

Game design for the casual market demands a great sensitivity towards the less-experienced user and a tremendous respect for the player's time. The casual game audience's level of patience and tolerance toward learning a new game is very low. However, while their level of initial commitment may be low, casual players also have few preconceptions about gaming genres and conventions. There is still plenty of room for innovative core game mechanics that build on these players' experience. Choice of thematic content is critical and has the promise to hold player's attention while they are gently introduced to new game mechanics. Ultimately, players' equity in the game experience should be the goal for any developer and at some level, all of the production and design considerations discussed above feed into that goal.

In the past year, there has not been as much progress in the community based space as many had hoped for. There has also still been very little advancement towards different business models beyond the try before you buy model. This model limits game designers (as most of the production budget is spent on the first hour of gameplay), and there is still not a clear ROI on most advertising funded games (budgets are low, and there is not a big incentive from the portal to the developer as they typically don't share advertising revenue). Micro-transactions are growing, and some games are starting to take advantage of it in the western world. Still, there is a need and opportunity for a back-end company to provide a multiplayer and billing hosting solution for the casual game development community.

The taste of the casual gamer has evolved and is players are beginning to embrace more complex games. The core of casual game design will still be simplicity, so a good balance must always be achieved.

## V. Technology Overview

### A. Introduction

---

This Technology Overview surveys the technical aspects of Casual Games. The first section discusses the range of technical characteristics of Casual Games. The second section discusses specific development tools for Casual Games. The third and final section covers a variety of security and Digital Rights Management (DRM) topics.

### B. Technical Characteristics of Casual Games

---

#### 1. Delivery Models

The delivery model can be defined as the way for the players to reach the game. For some games, players may have to download a program and then install it. Sometimes, users play the game through their web browser, most likely through an embedded plug-in of some kind. Following is a brief review of the most commonly used delivery models.

##### 1.1 Web Browser Games

Web browser games are arguably the simplest form of Casual game (from the user's point of view); most are developed in Flash, Shockwave, HTML, or Java and execute directly in the browser with no need for the player to execute a separate download and installation. Technologies like Virtools, are starting to gain traction in the market as a means of quickly producing professional looking content from within a browser window.

There are substantial security and integrity issues to consider when deploying a Web browser game: for example, it is very difficult if not impossible to save files locally on the users machine (with the exception of "cookies"); there are currently very few technologies that support full screen play (other than maximizing the browser window on the desktop), and; it is unlikely that the developer will be able to block out other applications on the user's machine. In spite of these limitations, users flock to browser-based games because of the ease of use: at most they have to install a plug-in, and at best they simply have to click on a hyperlink to begin playing.

##### 1.2 Downloadable Games

Downloadable games execute directly on a particular native platform, such as Windows XP or Macintosh OS X (although not always). Generally speaking, developers will package all files necessary for execution into a Windows and/or Macintosh installer. Users download and execute the installer appropriate for their operating system and then run the game from their desktop. It is common for developers to release both Web and Downloadable versions of the same game, using the web version to generate awareness and advertising revenue and the downloadable version to generate consumer sales.

##### 1.3 Downloadable Clients with Web-Based Services

It should be noted that even in the downloadable games described above; it is now possible and often appealing to use network services in conjunction with the downloaded client. This is often done for customer support interaction, but is also used for things like high-score lists, brag sheets, player forum links, etc. Even though the game itself exists locked on the

client and the network is not a part of the game play, the Internet is still used to provide experiences that are perceived valuable in relation to the game. This is increasingly the model of choice for downloadable games, as it allows the developer a way to contact the player for a variety of reasons.

## **2. *Playing mode***

The playing mode characteristic of a game covers the number of players involved in a game session and the duration of the game session. While computer game sessions can often require several hours or even multiple sessions of several hours to reach the game end, successful casual games, typically show a tendency to have short simple and repeatable sessions of only a few minutes per session.

### *2.1 Single-Session Single-player Games*

Playing games is the most fun if other people take part. But other players are not always available if you need them, which led to the invention of single-player games. The goal of a single-session single-player game is usually to make "moves *until one reaches a final state of the game, which results in a win or loss, or a score assigned to that final state.*

Thousand of single-players games have spawn all over the Internet during the last years. While some are simply Web versions of old arcade games or existing bundled game (like Solitaire or Minesweeper), several new completely games emerged. Often, the success of these Web games resides in their simplicity to play and the strong addiction they generate.

### *2.2 Single-Session Multiplayer Games*

Single-session multiplayer games represent the largest category of multiplayer games. Nearly any game that allows two or more players to join play over a network connection falls into this category. Every genre of game is represented in the universe of single-session multiplayer games, including: classic board games such as chess, checkers, and backgammon; card games such as Hearts and Spades; casino games; first-person 3D shooters; and sports arcade games ranging from snowboarding to billiards.

Single-session multiplayer games save little information about the player between sessions. Exception examples include user authentication data, player ratings and casino winnings. These data chunks can be either stored on the end-user's machine or on the server (the latter being a requirement whenever a secure solution is desired).

From a design perspective, successful multiplayer games work best when the social experience is tuned to the audience. This may vary significantly from genre to genre. For example, some 3D shooters could be described as games that "allow teenage boys to trash-talk while fragging each other". Parlor games allow users to leisurely chat without negatively affecting the pace of play. Another important design component of multiplayer games is the lobby system. A successful lobby system allows users ample time to socialize before entering a game. More importantly, it gives users the power to select their opponents.

### *2.3 Persistent World Multiplayer Games (MMOG)*

Massively Multiplayer Online Games (MMOGs) arguably represent the leading edge of Web games development in terms of technological complexity and resource requirements. A



persistent world is called "persistent" because when a player quits the game, the world still operates without him. Even if all players would leave the game at the same moment, or if the server supporting the game would be suddenly disconnected from the Internet for any reason, the world would keep running.

Examples of MMOG games in the CD-ROM space include World of Warcraft, EverQuest and The Sims Online. Graphical MMOG games in the Internet space (as opposed to text-based MUDs) have been steadily increasing in quantity since the mid-1990s, and we expect to see several launch in the next year. They operate as a service over the Internet – players connect via client software to a central server array where the game world is stored and managed. The technical problems of such client-server systems are substantial, requiring specialized and non-trivial engineering and operational (up-time, customer support) capabilities. MMOG games are generally offered for-pay only, with a short trial period to allow potential customers to grasp the game play experience. One can obtain the 2004 Persistent Worlds White Paper at the IGDA web site which was just released in December 2004.

#### *2.4 Skill-Based Gaming*

From a historical perspective, skill-based gaming as we know it today can still be considered nascent in that the concept of skill-based games only began achieving material market awareness in 2001. As a result, the genre can be characterized as being a fairly "early stage" sub-category of Web gaming.

In an effort to convey skill gaming's most relevant technology issues, this section will focus on the three most technology-affected areas of the sector, which in turn drive technological evolution: User Accessibility, Fraud Prevention, and Game Play Expectations.

##### ***i User Accessibility***

The concept of "pooled liquidity" has arguably had the greatest impact on the technological evolution of skill-based games. In theory, the skill-based game service providers with greatest "pooled liquidity" will succeed relative to the competition because of their ability to consistently offer more skill-based games, faster tournament filling, fairer match-making, and larger prizes.

As a result, in an attempt to appeal to a critical mass of users, the most successful skill gaming companies embrace a "least common denominator" technology development approach, whereby the principle technological objective is to provide a user experience that is accessible to the largest number of potential skill gamers. For this reason the most successful skill-based games are inherently quick to learn, quick to play (less than 5 minutes), and readily accessible by the user whose PC may be limited to 266 MHz of processing power and a basic dial-up Internet connection.

The desire to offer the most readily accessible skill gaming platform has inspired an "arms race" within the community of skill-based game service providers. As would be expected (within a relatively early stage competitive technology environment), myriad programming languages are being deployed in an effort to identify the optimum technology solution. The web-based client-side portion of a skill-based gaming system can be built using several technologies, including Shockwave, Flash, Java, C++ and ActiveX Controls. These client technologies, in turn, communicate with back-end systems capable of large transaction volumes, often built on Sun J2EE, Microsoft.NET, or other appropriate back-end platforms.

## ***ii*** **Fraud Prevention**

Game-related fraud represents the single greatest impediment to the large scale acceptance and subsequent mass popularity of skill-based gaming. In skill games, where cash prizes are on the line, the impetus to cheat is higher, and the need to quash both cheating and the perception of cheating is much more important. Skill game providers must earn the trust of players to build a successful business; therefore it is critical that they thwart cheating of any type.

And while the overall incidence of fraudulent activity experienced within skill-based gaming compares favorably to that of Las Vegas casinos (whose incidence of game-related fraudulent activity is in the low single digits), a misperception of the fraud-related risks associated with skill-based gaming could be detrimental.

Of particular note is the fact that the increasing popularity of skill-based gaming will translate into the availability of more and larger cash prize awards. Consequently, the persistence and technological sophistication employed by would-be "fraudsters" will continue to present a challenge to skill-based game service providers. The first lesson in battling cheating is remembering that it is an ongoing battle, requiring a site to spend money on maintaining an experienced anti-fraud team. New techniques are constantly developed; existing techniques are constantly revised and strengthened.

Another important lesson in preventing cheating is not to rely on a single method. If a cheater manages to foil one or two cheat-detection systems, a third will catch him/her. Having just one system is like relying on just an alarm system to safeguard your house; having multiple systems is like have an alarm system, hidden cameras, a trained Doberman, a moat filled with hungry alligators, and a yard filled with land mines.

The most successful skill-based game service providers have been able to effectively minimize game-related fraud by utilizing comprehensive technology-enhanced fraud prevention techniques. These techniques include, but are not limited to, Move Tracking: which allows the skill game service provider to play back and review all moves made by a particular player in a particular contest; Server-Focused Communication Protocols: which reduce the technical vulnerability associated with client-side communication protocols; Event Probability Assessment: which allows the skill game service provider to determine the likelihood of a particular event (i.e., a winning score, the likelihood of a particular move, etc.) based on statistical precedence; and, Strong Encryption Protocols: which securely encrypt client codes and events rendering server communications cryptographically indecipherable.

In terms of detecting "hot spots" for fraudulent game play, a good rule is to "follow the money." Players winning large single prizes, or with a large amount of cumulative winnings, are deserving of a greater level of inspection.

## ***iii*** **Game Play Expectations**

As the skill gaming sector continues to evolve, so will the game play expectations of the skill gaming consumer population. Subsequently, successful skill-based game service providers (like all successful game developers) must contend with the challenge of providing an increasingly more compelling (better graphics, faster processing, better sounds, etc.) game play experience.

However, because of self-imposed technological constraints (derived from the desire to maximize users' accessibility), skill-based game service providers are faced with the conflicting challenge of simultaneously maximizing the game play experience while maximizing user accessibility. Compounding this challenge are the consumers' game play expectations, which have been fueled by the game consoles (Xbox, PlayStation, Game Cube, etc.) and CD-ROM games which, unlike skill games, are not constrained by bandwidth limitations.

Consequently, the ability to deliver a secure and readily accessible skill-based game that has the production values (i.e., 3D graphics, accelerated processing power, etc.) of a console game, equates to the "Holy Grail" for skill-based game service providers and represents the logical evolutionary path for the skill-based game play experience.

### **3. Network Protocols**

A connection or network protocol describes the set of rules that two applications use to communicate with each other. There are thousands of different network protocols in use nowadays, ranging from downloading documents from the Internet to exchanging real-time audio and video. In fact, when two applications communicate with each other, they will most likely use several protocols simultaneously. This section discusses the common protocols broadly used in networked Web games.

Naturally, stand-alone games as described previously do not require any network protocol while played, and generally use Internet standards such as HTTP or FTP in order to be downloaded. Rather, network protocols apply to client-server or peer-to-peer architectures, and to single-session and persistent-state worlds.

#### **3.1 Internet Protocols**

Most hosts on the Internet today use the Internet Protocol (IP) to communicate with each other. IP is a low-level protocol used by routers and hosts to ensure the packets travel from the source host to the destination host. IP hides the transmission path and the recipient host has no way to know if the transmission path included phone lines, LAN, satellite links, or whatever. IP splits the packets into small fragments when they traverse networks links that cannot support large packets, and for reassembling the packets at the other end. Finally, the IP header also includes a "Time-to-Live" (TTL) field that specifies how many network hops may transfer the packet before the packet is discarded. This prevents packets from being accidentally routed in infinite loops around the Internet.

Applications generally never use the IP directly. Instead, they use one of the protocols that are written on top of IP. These higher-layer protocols include services for acknowledgements and retransmission, or support for application port numbers.

The Transmission Control Protocol (TCP) is the most used protocol on the Internet today. It is usually layered on top of the IP and referred to as TCP/IP. This protocol provides the running application with the illusion of a simple point-to-point connection to another application running on a separate computer. Each endpoint can consider a TCP/IP connection as a bi-directional reliable stream of bytes between the source and the destination.

TCP/IP is reliable as it automatically transmits acknowledgements and retransmits data. Furthermore, TCP/IP verifies the integrity of received data using a data checksum contained in the data packet header and both endpoints use a data flow control technique to ensure that the sender does not transmit data packets to the network faster than the network can support or than the recipient can process them. TCP/IP even allows the application to detect when the other endpoint disconnected.

Unfortunately, such reliability comes with a cost. Because of it, TCP/IP must transmit more information in order to accurately describe the data ordering, detect corruption using checksums and transmit acknowledgement or retransmission packets. Moreover, the recipient must receive and accept the entire data stream in the order that the sender transmitted it. Therefore, the TCP protocol handle may arbitrarily hold or buffer transmitted data in order to preserve the packet ordering. Consequently, TCP/IP is not suitable for applications that do not necessarily need a strict ordering and consistency.

The User Datagram Protocol (UDP) is a lightweight communication protocol differing from TCP in three respects: connectionless transmission, best-efforts delivery, and packet-based data semantics. UDP does not establish peer-to-peer connections as the sender and recipient of UDP data do not keep any information about the state of the communication session between them. With UDP, data is sent on a packet-by-packet basis. While TCP used such information to detect packets loss, request retransmission or dynamically adjust the data transfer rate, UDP simply provides best-efforts delivery, making the attempt to guarantee that data is delivered reliably or in order. In addition, the datagrams must not be too big, because if they have to be fragmented, some fragments might get lost in transit.

While UDP might appear at first as too weak to be powerful, it has in fact several major advantages. The first one being its extreme simplicity. As UDP packets do not contain any of the information guarantying reliability used in TCP, they require considerably less processing for both the sender and the recipient. Then, UDP does not maintain the illusion of data stream. This implies that packets can be transmitted as soon as they are sent by the application instead of having to wait in line behind other data stream. Similarly, data can be delivered to the application as soon as it arrives at the recipient host instead of waiting in line behind missing data. Thirdly and finally, as many operating systems impose limits on how many simultaneous TCP/IP connection they can support, UDP/IP seems logically more appropriate for large-scale distributed systems (including MMOGs) where each host can communicate with many destinations simultaneously.

There is however one aspect of UDP/IP that can make it rather unsuitable for some environments. When a socket is receiving data on a UDP port, it will receive packets sent to that port by any host, whether participating to the application or not. This can become a security problem for applications that do not make any difference between expected and unexpected packets. For this reason, some ISPs still block UDP packets, or block UDP in particular port ranges. While this is generally less of an issue than it was a few years ago, any large deployment should still check with major ISPs for UDP support and port ranges.

### 3.2 *Flash Media Server 2*

The Flash Media Server 2 (formerly the Flash Communication Server) from Macromedia (now Adobe) allows Flash and Director developers to implement multi-user solutions for clients. As well as standard data transfer, it includes the facilities for multi-way video and data, offering a rich out of the box solution for content developers. The latest release is 2.01.

Applications will be able to add video conferencing, shared whiteboards, audio and text chat, and other features to their applications in an easy, straightforward manner.

Package and pricing information can be found on Adobe's website.

The Flash Media Server 2 is also offered as a solution for multi-player games, via its ability to also transfer simple packets of data. As most games will be transmitting relatively small data packets, rather than constant streams of video or audio, the maximum number of active connections will be more of a concern than the maximum data transfer rate when considering what edition of the server to purchase. There are certain multi-player features that Flash Media Server 2 doesn't support, such as peer to peer networking.

### 3.3 Director / Shockwave Multi-User Servers Options

Since Director MX, the Shockwave Multi User Server (SMUS) has been depreciated somewhat. Although it still operates with all versions of Director and has been ported to OS/X, development has been frozen on it and the above Flash Media Server is recommended by Adobe as an alternative.

However, a number of Director multi-user game developers still use SMUS for a number of reasons, including the fact that it supports features not available in the Flash Communications Server solution, including the ability to implement peer to peer networking capabilities.

One alternative option to using the SMUS server (but still using the client SMUS Xtra) is the Nebulae Multi User Server, available at [xtras.tabuleiro.com/](http://xtras.tabuleiro.com/), which is a third party implementation of a server system that meets the SMUS spec. Also, a useful set of routines for developing SMUS applications for first time developers can be found at [gomu.rebus.gr/](http://gomu.rebus.gr/).

## C. Base Delivery Technologies

---

### 1. Director / Shockwave

Adobe Director is a tool that can be used for both Downloadable and Web Games. Although the development environment product is named Director, the web player is named Shockwave.

According to Adobe, the Shockwave plug-in has been installed in just over fifty-five percent (55.4%) of the world's browsers. It is available for both Windows and Mac OS X.

Its ability to handle fast bitmap blitting operations has given Director/Shockwave the performance edge over Flash for certain types of 2D, sprite-based games. Additionally, Director's real-time 3D engine and Havok physics, gives it an advantage for developing small, fun 3D games. However, in recent years, Director has fallen behind advancements in 3D hardware. Sophisticated 3D games can still be made in Director, however some developers are turning to other technologies, such as Virtools, to make Web games that can compete with consoles on a technical level.

Shockwave3D is the 3D package that sits inside Director. It offers a vast range of both standard and advanced features such as level of detail, sub-division surfaces, skinning, multi-texturing, particles, the list goes on. Using Lingo or JavaScript syntax, you can create

meshes at run-time, or can use content from all of the major modeling packages using the available exporter plugins which exist for 3D Studio Max, Maya, Lightwave, SoftImage, TruSpace and more. Stand-alone converters are also available. The engine supports Direct3D and OpenGL, and even has a software renderer to ensure that everyone can view the content created in Shockwave.

The real-time physics capabilities offered by Havok, although a subset of their full v1.x package, are quite extensive, and can allow for some very impressive simulations that previously would have been out of the scope of most Director based projects.

Director can be used to develop a Shockwave web demo version of a Downloadable game, as well as the Downloadable game itself. Developers do not need to select a different tool for the Web and Downloadable versions of a game. Director is cross-platform, and can be used to develop Downloadable games for Windows and Mac OS X.

Some developers have reported abandoning Director/Shockwave for 2D applications because its sprite engine is too slow for the demands of their games. However, Director has been used in a number of successful Downloadable games and countless Web games in 2005 and 2006.

In the most recent version, MX2004, Director added the capability to manipulate DVD content, and to access all of Director's features through a new, ECMAScript 1.5 / JavaScript syntax (which is very similar to Flash ActionScript), in addition to the Lingo language the product has used for years.

In 2006, Adobe added Flash 8 support to Director / Shockwave, allowing developers to embed and control assets from the latest version of Flash.

Shockwave has built-in sound support that includes audio compression, streaming, basic volume and panning, music playlist branching, and embedded markers. There is midi and mod music functionality via xtras including the BASS xtra plugin.

## **2. Flash**

Flash has an unprecedented popularity compared to any other web based plug-in. It is available across all majors PC platforms (from Windows to Macs to Linux) and numerous non-PC devices (from consoles to PDA's). Flash offers developers a huge installed client base.

Its core engine is primarily a non-hardware accelerated vector based system, and as such has no game-specific features. Despite these limitations however, many developers have discovered and published various optimization techniques and best practices in regards to code and graphics that have enabled them to produce great results in both visual effects and overall game performance. This has resulted in Flash being used for even the development of deluxe Downloadable games.

With the introduction of Flash 8, Adobe has apparently responded to the requests of the game development community with enhancements that enable much higher performance in the areas of screen-draw and math -- as well as bitmap and filter API's which open up an endless array of code-driven visual effects that allow Flash developers to do things that previously could be done only in Adobe Director.

Its ActionScript language, like any scripting language, can be slow at times, although a number of dis-assemblers are available to allow developers to see exactly what is happening under the hood with the scripts, and some of them have used this knowledge to code their games at this level (almost akin to assembly language coding) to get the maximum performance out of the scripting engine.

However, the ease of dis-assembly makes Flash an unsecured client environment. Without the use of code obfuscating tools (there are a few third-party tool) and techniques, any moderately sophisticated end-user can reverse engineer a Flash movie.

Flash doesn't have built-in 3D support. However, it can draw polygons that can be re-positioned at run-time and some rudimentary real-time 3D environments have been designed and implemented. Some utilities allow for the optimized export of 3D model screenshots and animations from popular modeling packages. These utilities basically take each 3D image and break it up into a vector based image. The more advanced ones break the image up into layers, and try to "tween" between the various frames to help optimize further.

The lack of hardware acceleration within the main player will always result in slower performance than if the hardware were handling the rendering of the scenes, so Flash is not appropriate for certain types of games. However, the popularity, flexibility and portability of the engine is unparalleled. Currently within the marketplace, Flash reigns supreme for creating energizing graphical content through a high level scripting language. ActionScript 2.0 syntax offers a mature, class-based OOP architecture. The recent release of ActionScript 3.0 introduces enhanced structure, faster compiling and dramatic improvements in performance.

Flash is an excellent prototyping solution regardless of the final target platform because much can be achieved with only a few lines of code. An experienced Flash developer can produce a decent prototype of the core mechanics of a game within a few hours, allowing quick review and assessment of a concept in advance of level design and game-play refinement. This helps to easily play-test a variety of derivatives of a particular game mechanic before taking the next development steps. Adding and swapping out static art and animations are trivial tasks in Flash, making even "organic" styles of development to be possible on even tight deadlines and budgets.

One real great benefit of using Flash is the possibility to develop a web version of a Downloadable Game with minimal effort. Whether or not the Downloadable game was developed in Flash, one can easily and quickly produce a smaller and limited demo "upsell" version. Because of the ubiquity of Flash, "upsell" versions of a game reach more users and do not have overhead of custom ActiveX control installation.

Flash has become quite powerful in terms of features over the years, but it is still slower than compiled code. If a game requires heavy calculations (e.g. complicated AI), or has a lot of on-screen action and/or a larger scrolling regions, then Flash would not be a good candidate. For most games, the code execution is fast enough in Flash. The main bottleneck is slow rendering of graphics, especially in full screen (800×600). Fortunately, casual games often do not have a scrolling screen and their graphics are fairly static most of the time. Puzzle games and card games tend to be fairly static and work well in Flash. If the screen area taken up by an animation is minimized, a game can run at an acceptable frame rate on a moderate computer.

Although some devices such as Sony's PSP provide support for more full-featured Flash content (currently up to version 6), Flash Lite, which is the Flash technology developed for mobile phones and consumer electronics devices, has seen rapid adoption, particularly in the Asia and Europe.

Flash has built in sound support that includes mp3 compression, streaming, basic volume and panning, limited music branching, and embedded markers. There is no midi, dls, mod, pitch or dsp support.

### **3. Java / Java WebStart**

Java ([java.sun.com](http://java.sun.com)) began life with the promise of 'write once, run anywhere'. While this is certainly more so than with any other comparable, contemporary language, it hasn't become the panacea once envisioned. On the other hand, Java has become a viable and practical platform.

Java was originally developed to serve as a portable platform for content delivery on interactive set-top boxes. The media group within Sun found themselves grappling with an explosion of portability, compiler, library, and security issues as they attempted to develop atop a variety of commercial hardware platforms, and they invented Java to soothe the pain.

Since those early days, Java has evolved considerably. Java 2 was launched as a successor to the original Java and comes in three 'editions' - each one tailored for a specific class of problems. The original Java has gone on to become Java 2 Standard Edition (J2SE) - a Java platform targeting a desktop environment. The Java 2 Enterprise Edition (J2EE) has emerged as a superset of the J2SE intended for deployment in enterprise and server-oriented environments. Finally, Java 2 Micro Edition (J2ME) has emerged as a slimmed-down Java platform for resource-limited devices (CPU, memory, interface, connectivity, etc.) that still might benefit from the Java platform. J2SE and J2ME are the two most relevant to game development and deployment. J2EE is a set of enterprise-grade technologies not directly relevant (though potentially useful) to game development. Each of these editions have been given extensive coverage in previous papers, and readers making decisions about Java implementations are encouraged to review those notes.

#### **3.1 Extending Java**

Java provides a standard mechanism for platform extension via native code called Java Native Interface (JNI). This makes it practical to use Java as a scripting language while leveraging any existing libraries and allowing you to implement more performance-oriented operations in native code. In particular, the Java 3D API ([java.sun.com/products/java-media/3D/](http://java.sun.com/products/java-media/3D/)) is one such extension that provides advanced 3D APIs built atop existing underlying graphics technologies (OpenGL and Direct3D). While this extension is available only on a limited number of platforms, it is a strong candidate for mid-tier 3D game development (compare to Shockwave w/ the 3D Xtra). Until Java 3D becomes a core component of the Java platform, however, developers will need to ensure that their customers install the Java 3D extensions alongside their Java 3D-based games.

There is also a second development effort in Java circles to promote the use of 3D - the Java Open GL (JOGL) project - which while community run, is now regarded as one of the best opportunities for 3D through Java. More information is available at <https://jogl.dev.java.net/>.



### 3.2 Java Delivery Mechanisms

A variety of delivery mechanisms exist for packaging and delivering Java-based content to your audience. These include:

Applets

The Java Plug-in

Java Web-start

Pre-packaged applications (w/ and w/o the JRE)

The two major delivery mechanisms for Java on the web at this point are the Java Plug-in ([java.sun.com/products/plugin/](http://java.sun.com/products/plugin/)) from Sun and the Java Web Start ([java.sun.com/products/javawebstart/](http://java.sun.com/products/javawebstart/)).

The final area in which Java delivery is becoming critical is mobile game development. You can learn more about mobile game development in the Mobile Games White Paper at [www.igda.org/online/](http://www.igda.org/online/) or from the IGDA's Mobile Games SIG [www.igda.org/mobile/](http://www.igda.org/mobile/)

## 4. Python / PyGame

Python is most often thought of in game development circles as a scripting language for something that controls a lower level C/C++ core. Python in particular has been used in a number of recent projects including ToonTown (Disney), Eve Online (CCP), Blade of Darkness (CodeMasters), Star Trek Bridge Commander (Totally Games), and Earth & Beyond (Electronic Arts).

A new generation of games is springing up using Python in combination with PyOpenGL and/or PyGame ([www.pygame.org](http://www.pygame.org)). PyOpenGL provides a wrapper to the standard OpenGL library and is thus suitable for developing 3D games, PyGame provides a wrapper to the Simple DirectMedia Layer ([www.libsdl.org/index.php](http://www.libsdl.org/index.php)) which uses GL or DirectX depending on platform, and provides cross-platform access to hardware accelerated graphics as well as input devices and other common game-engine requirements.

Libraries exist for Python to do its own networking, and there are now several small downloadable client-server games built with Python/PyGame, with more popping up every day. While most game designers are still looking at Python as a scripting language, its ease of use, its cross-platform capability, and its ability to call C/C++ code when needed make it a strong candidate language for small downloadable games. There are currently no known technologies that play python games natively in a browser.

## 5. PHP + HTML

The Hypertext Pre-processor language (PHP - [www.php.net](http://www.php.net)) is a widely used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML. It features a simple C/C++ style syntax that supports object-oriented mechanisms. PHP excels at its ease of use in building low to complex systems. PHP is completely free with many modules available for most popular Web servers and platforms.

PHP is primarily designed for server-side scripting, therefore it can do anything that a CGI script can do, from collecting data to generating dynamic page content. The difference

between PHP and other scripting languages like JavaScript is that PHP is executed on the server hosting the file. As such, the client only receives the results from this script, without any access to the code that generated the result.

PHP can also be used for command line scripting. This type of usage is ideal for scripts regularly executed using Cron or Task Scheduler. PHP can be used in client-side GUI application as it can output HTML, dynamically generated images and graphics, or even Flash movies generated on the fly. One of the strongest and most useful features in PHP is its support for a wide range of databases, from MySQL, PostgreSQL, Oracle and more via ODBC.

PHP is often selected to create small to medium Web persistent multi-player role-playing and/or strategy games. This is most likely due to its cost, and the ease in which it allows the implementation of a client-server architecture relying on a database system.

These games can be played directly through a client browser written in PHP or through a downloaded client written in another language. Numerous examples of persistent PHP-based games are available through the Internet with the membership ranging from a few dozens to thousands of players.

Among the most successful PHP Web games, is the Legend of the Green Dragon RPG ([www.lotgd.net](http://www.lotgd.net)), developed under the GNU license and hosted on a few dozen servers, each supporting a few hundreds players. Another interesting example is the Star Wars Combine ([www.swcombine.com](http://www.swcombine.com)), a free and non-commercial massively multiplayer Web role-playing simulation game, based on the Star Wars universe, developed by amateurs during their spare time. It gathers over 1,500 active players in a persistent gaming universe.

## **6. *DarkBasic***

DarkBasic has gained a small following over the years primarily as a "first programming language" - but one that is very capable of producing game content. The libraries included with the product allow access to DirectX and provide what amounts to a fully functioning 3D engine. Projects produced with DarkBasic cannot run native in a web browser, but must be downloaded to a client's machine and installed. The engine is currently geared at creating stand-alone 2D and 3D games, and does not have tools for networking capability (although experienced programmers could certainly add them). Nevertheless, the graphical capabilities are well documented and easily employed.

## **7. *VirTools***

VirTools has long been a significant tool in the prototyping of the more traditional games market, from PS2 and Xbox development houses to the big PC Game makers. VirTools also provides a browser plugin and content server to deliver game content. While more expensive than most of the other technologies listed here, VirTools is also one of the most complete - offering near-seamless integration with Maya and 3dsMAX, a "C-like" scripting language, and a wealth of pre-scripted game scenarios, built-in collision detection and even some AI. Unlike several of the other technologies discussed here, it is apparent that VirTools is an engine designed specifically for games and game creation, as opposed to a general purpose tool that can be used as such. One issue to note is the relatively small market penetration of the VirTools plug-in, although that may change as more adopters begin to push content.

## **8. Garage Games / Torque**

Garage Games offers a variety of game engines and game development tools for both 3D and 2D game development. Their most well-known product is the Torque game engine which comes in two flavors: Torque 3D and Torque 2D. An example of a successful Torque 3D release is Large Animal Games' RocketBowl Plus. While Garage Games' engines are not free, the licenses are priced affordably (around \$100 at the time of this writing for the 'Indie License'), with two pricing tiers that depend on the revenue of the licensing company. The Torque engines are very full featured and fairly high-level game engines. While the engines are based in C++, they are designed to be useable without programming in C++ by using Torque's pseudo-C++ scripting language. The engines fully support Macintosh, Windows Desktop and Linux targets, and include sound support. Most recently, Torque has added support for Xbox 360, and has enjoyed some success with LiveArcade titles there. There is also a very active developer community for these products, which can be a significant help to learning the new tool. Common complaints about Torque is that using the engine can involve a steep learning curve, and making modifications to the engine's provided source code can involve considerable digging around.

## **9. PlayFirst / Playground SDK**

The PlayFirst Playground SDK is a C++ API that offers support for hardware accelerated 2D and 3D graphics, UI controls, Music, Sound FX, and Lua scripting. It also includes more game oriented features like user state/preference management, high scores, multiplayer, and a complete game template. Playground games can be deployed as Windows Download, Macintosh Download, and Windows ActiveX games (including hooks for MSN integration). The engine also has full UNICODE support and an embedded vector font engine allowing development for most common user configurations and languages around the globe. The footprint for the engine compresses to just 0.5 MB. It has been used recently in Diner Dash 2: Restaurant Rescue. The SDK is free to developers working closely with PlayFirst.

## **10. PopCap Framework**

The PopCap Framework is a C++ casual game engine and has been used in a number of widely distributed games. It offers a 2D game engine with hardware-acceleration support. The Framework includes many high-level features such as UI controls (buttons, sliders, popup-dialogs) that are very object-oriented and quick to implement. The rendering layer on this framework was designed for the casual user demographic with optional hardware-acceleration support which will fallback to software rendering if it needs to. There is a fairly healthy and growing user base for this engine with active developer forums at PopCap. While this game engine is free, there is an obligation to credit PopCap (including a URL) in any final game product using the engine. There is no direct sound support but the engine comes already integrated with FMOD and BASS sound engines (requires separate licenses). The supported target is only Windows desktop.

## **11. SDL (Simple DirectMedia Layer)**

SDL is an open-source low-level game engine written under the GPU public license. It is more of a rendering layer with a lot of extra functionality and could be a good fit for expanding into a higher-level proprietary game engine. An example of a successful casual game written with SDL is Grubby Games' Professor Fizzwizzle. SDL has good support for multithreading, streaming audio functionality, and supports a very long list of target platforms including Macintosh, Windows Desktop, and Linux.

## **12. GapiDraw**

GapiDraw by Develant Technologies AB is a commercial 2D game engine that is probably better known in the PDA games space than in the desktop game space. It is a very popular choice for Pocket PC games development. The GapiDraw framework does not offer a great deal of higher-level functionality such as UI or controls, although it does offer a full range of fast blitting functionality such as rotating, scaling, alpha blending, additive and blended drawing among others. GapiDraw has also shown itself to be quite robust and stable. An example of a successful casual game release that uses GapiDraw is Sandlot Games' Tradewinds. GapiDraw is one of the few game engines that is capable of simultaneous targeting of Windows Desktop, Palm OS, Pocket PC and Symbian platforms (if you were able to get the graphic assets arranged correctly). There is no support for Macintosh or Linux, nor is there integrated sound support, and it comes with a somewhat higher licensing fee.

## **13. PTK**

PTK by Phelios, Inc. is a commercial low-level 2D game engine. Really more of a rendering layer, this engine could be a good solution for developers seeking a no-frills hardware-accelerated framework that they could build on top of. PTK has support for fast blitting (with rotating, scaling, alpha channel support), font management, basic sound support, file access, and targets both Macintosh and Windows Desktop. The API is very simple with virtually no learning curve. There is no higher-level game functionality such as UI or dialogs. An example of a successful casual game created with PTK is Big Fish Games' Fairies (originally developed by FunPause). Support is rather limited and directly through the developer, with little or no community support presently available. The license fee has a very low price-point and licensees gain access to the full source code, further increasing PTK's usefulness as the foundation of a higher-level proprietary game engine. A version with native support for Intel-Macintosh is reportedly nearing completion.

## **14. Additional Sound and Audio Middleware Technologies**

In some cases when a developer is using a game platform technology that doesn't support sound or the game requires extra needs that the sound API doesn't support, there are several middleware sound packages available. These packages can be especially useful if the game involves sophisticated audio techniques or rhythm action or manipulation.

The most common packages include:

**FMOD** - A commercial package that handles extensive audio, mod music, plugin and dsp capabilities that supports PC, Mac, Linux and all consoles.

**Miles Sound System** - Another commercial package that also handles extensive audio capabilities, including mod, midi, dls, and dsp support. Miles is available on PC, Mac, and consoles.

**BASS** - Another commercial package that offers sound functionality for PC and Mac platforms including mod playback and compression, and sound APIs.

**Audiere** - A open source PC only audio solution that supports mod music playback and sound playback functions.

## ***D. Security and Cheating***

---

As casual games have become a more serious business, security and measures to counter cheating have become more important. Where once a cheater simply boosted his high score, hacked his games with "friends", or disrupted game play and made a nuisance of himself, now these activities can cost a developer potential customers and revenue. Security is a large discipline that spans the entire lifecycle of a product or project and includes both technical and non-technical aspects. This discussion will briefly review the standard IT security technologies that are relevant to networked games, and will focus on those that are unique to this field.

Security is intimately tied to the operational integrity of the game. While computer games were once products, in many cases, casual games today are services. This changes the essential nature of the business - games are not just "published", they are "run". Customer service, retention, and acquisition depend more on persistent quality than on flashy packaging and cool graphics. Development and delivery of a game is not the end, it is just the beginning.

The security threat for traditional computer games used to be copy protection. Various proprietary schemes have been created with each one a balance of protecting sales versus annoying customers. Typically, not annoying customers has won out over security techniques - just as with other traditional software, the belief was that a satisfied "thief" today may be a good, paying customer tomorrow. The effectiveness of the available anti-copying techniques also was (and is) an issue.

It is always important to remember - the bad guy does not need to break the security system, they just needs to beat the game.

The following represents a partial list of the types of attacks and common exploits; new attacks are devised against every new game and security technique.

There is unfortunately a dearth of products focused on networked game security. Rather it falls to the developer to defend against the exploits described below when implementing with any of the previously mentioned toolsets. This is unfortunate, and can significantly impact development time and technological implementation. Some known software solutions to these types of attacks are listed in sections below, although each of these will require careful integration with your game and its development process.

### ***1. Common Exploits***

In the following sections, we will review the most common methods used in Web cheating, and the protections against them. This section is based on the observations compiled from the sources listed in the footnotes.

#### ***1.1 Network Game Security Issues***

With the rise of casual games, security has become a much more serious challenge. While the game industry has always been concerned about piracy, new platforms, new business models, and the power and speed of networks have created new security problems for game developers, publishers and operators. These problems rapidly become more serious for networked games since word-of-mouth and long term relationships drive sales and revenue

more than just slick packaging and marketing. Network game operators also face costs associated with customer support and charge-backs that traditional computer game publishers and developers have not had to confront. The discussion that follows will try to establish clearer definitions of game security issues.

## 1.2 Piracy

Piracy continues to be a problem for network games. Digital Rights Management tools have joined older physical media protection and various licensing tactics to protect against unauthorized duplication. Unfortunately, once these systems have been defeated, there is no way to restore the game to a secure state. Web game services do not have this problem as the operation of the service provides some measure of protection against piracy. Web game services do not protect against duplication of legitimate copies of a game - at best they typically prevent concurrent use of a game account. In addition, it is also necessary to protect not only the game itself, but all copyrighted materials contained within the game, any and all music, sounds, artwork, storylines, etc. that are created or derived from copyrighted work. A detailed discussion of these issues as they relate to piracy can be found in our discussion on DRM.

## 1.3 Ghost Servers & Doppelgänger Services

Web games have given rise to a new problem - creating pirate servers and services, not just pirating games. Though this problem is new, it is in some sense worse than traditional piracy. The game server or service operator has often invested substantial resources in the game infrastructure. Whether the game service provider is trying to operate a subscription service or simply providing a quality Web service with marketing and license registration features, the game service is an important business asset. In the US, the BnetD application that emulated Battle.Net is an example. In Asia, the problem has been more serious. Due to widespread software piracy problems, subscriptions and virtual goods are virtually the only viable business strategies for game developers. Unfortunately, pirates have set up alternate game servers and hosting services threatening the subscription revenues of legitimate game developers and operators.

## 1.4 Data Spoofing

One of the simplest means to attack games is to spoof the data passed between participants. This is distinguished from "State Spoofing" described below as altering the local game state. Data spoofing may consist of altering player actions, updates to game state, or any other information exchanged between game players. This can occur at two levels: "on the wire", after the data has been sent; or internally, before the data has been packaged and sent to the remote location. Traditional security tools such as encryption and digital signatures are powerful methods to protect against "wire" based or third party attacks.

Unfortunately for game developers, these attacks are the least likely - game players themselves are the most serious threats to networked games. The malicious alteration of data by a game player can only really be stopped by effective verification of the incoming data itself. This does have serious design implications for networked game programmers. If a game sends raw game state updates to remote players, the ability of the receiving game instance to validate the changed state may be difficult. Data exchanges that map to "player actions" may be easier to validate as they correspond to inputs to the game rules. Chess provides a straightforward example: it is easier to validate whether a Knight can move from

one position to another than to compare the game board before and after the move to determine if the new board could have resulted from the old one.

### *1.5 State Spoofing & Disclosure*

The local game state is inherently vulnerable to attack. It sits on the computer of the potential hacker and, if the game is going to work, the game code has to operate. The reverse engineering of the game state and rules is virtually inevitable. While there are tools to obfuscate code and data, they can only delay the problem. They may also introduce performance and testing issues for the game developer. The best approach is to design game software so that the known game state does not damage the security of the game. This may not be practical for games with rigorous performance requirements and low bandwidth. As discussed above with data spoofing, the alteration or manipulation of the local game state should not permit an advantage against other players. The worst-case scenario should be the cheater simply falling out of synch with the rest of the players.

### *1.6 Net Time Manipulation*

"Time Hacks" have been one of the most persistent problems for network games. By manipulating the apparent lag between players' computers, cheaters can give themselves a performance edge. A hybrid between time hacks and data manipulation hacks is to overload a receiving system with more actions than should be allowed within a given time interval (more moves or shots than should be permitted). Part of the rules validation discussed above should be temporal rules, not just state-based rules. Maintaining time synchronization is a problem for computer games, but actually Web Chess faces a worse problem - trying to control the game clock in a manner that is fair to all players.

### *1.7 Score Spoofing*

One of the simplest ways to encourage participation in Web games is through a shared high score board. Unfortunately, some players will spoof their score to get on the high score board. If there is no interaction between the game application and the game provider, it is very difficult to detect such spoofs. Hackers will reverse engineer the game application to determine what they need to do to submit a high score. If the game is implemented so that it is completely interactive with the game host, such spoofs can be stopped; the problem with this approach is the bandwidth and processing requirements to support this approach.

### *1.8 Tournament Collusion & "Playing with Yourself"*

One of the fastest growing portions of the network game business is offering tournaments. As usual, hackers follow the market. For tournaments, players can collude with other players or create fake player accounts to boost their ranking. This problem is worst for free tournament systems (the cost of registration and play tends to naturally deter this tactic), but could be a concern in games with substantial prizes or other incentives. Though various means can be used to monitor who plays with whom, the best method for official tournament games is probably to randomly assign opponents and change them regularly. This is well proven in the traditional games world for card rooms.

### *1.9 Optimal Play*

The problem of a "perfect player" or "aimbot" is not exclusive to computer games. In some sense, a card counter in Blackjack is exactly the same - a player following the rules of the game who is using all of the information available to him to his best advantage. This problem is more of a game "system" or design issue than a security problem - it would be much better to remove optimal strategies from computer games. It is more fun, after all if there are meaningful strategic choices. A good "aimbot" will be nearly indistinguishable from a good player (barring other cheats, of course).

### *1.10 Griefing*

The in-game abuse of other players has been a problem as long as network games have. These players are carrying out activities that are legal under the rules of the game, but detrimental to the fun and game experience of the other players. This is no less important an issue than cheating or hacking, but it does need to be countered differently. Spawn killing (killing characters where they enter or re-enter a game) and camping (waiting in a location where a high value creature or item will appear) are both examples of the problem. Once again, good game design practices are probably the best mechanism to counter these issues. For example, instead of having monsters created at a fixed location, they could be created through a probability model in an area where an individual or party is roaming. By eliminating the fixed location, the incentive and advantage to camping goes away.

### *1.11 Outsourcing Play*

Recent news reports have discussed MMOG players outsourcing the "operation" of their characters while they work or sleep to other players in Russia and other countries. Players hiring other players to play on their behalf may or may not be a security issue for Web game services today, but it could become a problem if it distorts the experience of the game for other players. Since most Web game services are protected solely by user name/passwords, they do not have a good method to protect against this problem. Also, game service providers must balance the convenience of allowing players to use multiple computers (such as a home PC as well as a laptop) with security requirements. It is interesting to note that the older model of metering gameplay by time would naturally address this issue.

### *1.12 Denial of Service*

The problem of denial of service attacks against games and game servers is shared with all other Web transactions. There is no way to completely stop an attempt to bring down a network service. However, game software should be designed robustly to only accept "good" data from valid sources and generally resist buffer overflow and other traditional attack methods. Centralized commercial game services can use rate filters, intrusion protection systems, and other tools to harden themselves against these attacks.

## **2. Game Software Security Tools**

There are a number of game software security tools. These products attempt to detect and counter cheating software by various strategies. Typically, they are built for client/server games where the server is assumed to be trustworthy. Some current products in this area include HLGuard, Cheating-Death, nProtect's Game Guard, and Even Balance's Punkbuster. Each of these products must be tailored for a specific game and must be updated as new



attacks are found. Since they are a separate software package, these tools can be added in to an existing game.

The server portions of such solutions look for anomalous network traffic that corresponds to various cheating packages. This is very similar to conventional Intrusion Detection Systems (IDS) that have a catalog of signatures that they continuously search for. This catalog is continually updated with new "cheat signatures". The server may also use heuristics to attempt to detect new forms of cheating. Some of these server tools probe the client-side player platforms to determine if the corresponding client security software is in-place and operating properly. The server may also support a registry service for good and bad players.

The client portion of these solutions also is looking for cheating software (sometimes referred to as "warez"). The security client software faces a particularly daunting task as the player typically willingly, if not eagerly, has installed the malicious code. These solutions work similarly to anti-virus software - looking for "warez" signatures. They may also work in a more intimate fashion with the game to make certain types of cheats impossible (Cheating-Death relocates the local position of illegitimate targets to a point where they cannot be attacked). Makers of these security tools are in a continual "arms race" with the cheating community. Cheaters will reverse engineer and circumvent each version of security code and the toolmakers must continually work to keep up.

### *2.1 Game Protocol Security Tools*

By their nature, Web games can be attacked both within the local software and on the network connection between players or between a player client and the game server. Encryption and digital signatures can prevent manipulation of game data and against disclosure. This can be very effective against third parties but, as noted above, may be less effective against a malicious player client.

A different approach is to secure the game transaction between the players. The SecurePlay library does this by implementing game transactions in a manner that is mutually suspicious. This approach means that the participants in the game do not need to trust the other players' software - as long as the transaction is valid, the game is valid. These basic transactions can be combined together to implement the various game rules and provide overall game integrity. The advantage of this approach is that the Web game can be assessed secured entirely through the network transactions between the players (or a player and the game server) without requiring any trust in the remote party. The challenge comes from building these transactions into the game, rather than adding an after-release security tool.

### *2.2 Piracy & Digital Rights Management (DRM)*

Game developers need to be concerned both with the protection of media assets as well as the game itself. Art, music, and other third-party copyrighted assets may be used under license in a game. This discussion covers both general DRM and addresses issues specific to games.

## **3. The Digital Media Life-Cycle**

To understand the role of digital security, one must begin with the lifecycle of digital media from creator to consumer. The process described below highlights the major stages of this

lifecycle and they vary from music to video to films to games, but the essential steps are the same:

**Creation** - the creation of the media in its original format. Content may come from other sources. There is no real means today to provide or mandate appropriate copyright citations or pay royalties.

**Mastering** - the conversion of the original work into a format suitable for distribution.

**Production** - the printing, stamping, and reproduction of master onto physical media.

**Vendor Distribution** - the distribution of media to retailers or end users.

**End-User Distribution** - this stage moves copies of the media to the end consumer or audience members.

**Conversion to User Acceptable Format** - the delivered media is read by a device or application and converted into a format that the user may see, hear or otherwise comprehend.

**User Experience** - the user is able to experience a replica of the original material.

The critical challenge that security systems for digital media face is that an adversary attempting to circumvent the security system must deliver a comparable or identical User Experience. Thus, security measures must only be separated from a "User Acceptable Format" (UAF) as opposed to actually being defeated (Why break down a locked door when there is an open window nearby?). The existence of common, non-proprietary or non-royalty bearing encoding systems such as MP3, JPEG, etc. gives an adversary a means of distributing media in a (UAF) at low or non-existent cost. The adversary's objective, therefore, is reduced to migrating from a secured format to a UAF.

This is not true for many games where the game engine itself prevents creation of a UAF version of the game itself, but not art, sound, or other assets.

#### **4. Current Security Solutions**

The current solutions for securing digital media can be divided into the following general categories. Some systems combine elements of several of these categories (i.e., combining a digital signature with a fingerprint and proprietary encoding):

##### *4.1 Proprietary Encoding*

Proprietary Encoding is the use of distributor-controlled medium for the distribution and reading of digital media. This solution has been used most widely for DVDs. Proprietary encoding enables post processing and a multitude of other security controls to be implemented - whatever the developer of the proprietary encoding system wishes (as also noted with DVDs that include regional controls over and above traditional copyright protections that limit the use of DVDs to certain geographic regions based on configurations of DVD players).

The practical problems associated with proprietary encoding include their limitations on artists and distributors for the production and control of media as well as issues of royalties to the owners of the encoding technology (as witnessed by the recent and ongoing battles over DVD RAM and previously with VHS and Betamax).

The security problem with proprietary encoding schemes is that they are subject to reverse engineering (DeCSS for DVD that allows DVDs to be read and processed in software by PCs with open source tools) and, in the hands of pirates, to the regeneration of media into any form and format desired. DeCSS showed that the reverse engineering of the DVD proprietary encoding system was not difficult, so a pirate would be able to gain access to the program or do the reverse engineering herself and still can. There is a general security principle that must be recognized: any security system that is mass-produced by the millions can and will be broken by the few for the benefit of many. Economics and engineering all favor the adversary. Cable scramblers, DVD encoding, and file protection schemes for games have all shown that virtually any security system can be broken.

The approach of most DRM and other security solutions today is to "buy time". Games and most other media have a very short "shelf-life". By protecting a game or other media against piracy for days to weeks to even a month, many sales can be protected.

#### *4.2 Post-Processing*

Post-Processing is the alteration, encoding, or other modification of digital media during the act of copying, playing, or distribution. DVD is again an example of this type of system where DVD players output a signal that cannot be easily transferred to videotape. This can be used in conjunction with counters and other means to track the number of times that a piece of digital media has been copied and so trigger limits on copies and redistribution (either stopping the redistribution or degrading the copy through subsequent generations).

The security problems with post processing are identical to those with proprietary encoding and fingerprint systems. If the post-processing system can be reverse engineered or bypassed, then a product can be produced to "reset" the media to a pristine state. Or, if the post-processing uses some fingerprinting approach, a "BEFORE" copy can be compared with an "AFTER" copy to identify and remove the security fingerprints. Finally, it is quite possible for an audience member to be able and willing to accept imperfection - the adversary objective is user acceptance, not perfection. It is notable that these security mechanisms are most effective against individual audience members, not large-scale pirates.

#### *4.3 Physics & Engineering*

Certain anti-copying technologies rely on the low-level physical characteristics of physical media and the engineering design of digital media readers. These techniques are subject to the wide range of design and specification of consumer electronics (either causing the security technology to fail or resulting in the media being unreadable by legitimate customers). These solutions do not work at all for downloaded electronic media since there is no standard physical media or writing process. They can be circumvented by using low-level copying or production tools to preserve or bypass these technologies.

#### *4.4 Watermarking*

A "watermark" is information that is embedded in all copies of a piece of digital media. The information is either identical for all copies or divided into large categories. (The most familiar "watermarking" system is the regional encoding used for DVDs.) Watermarks can more easily be engineered to reduce the amount that they degrade a digital media source and are also much easier to produce than fingerprinted systems. (See production challenges under "Fingerprinting".) Watermarks are typically used with proprietary encoding or post-processing systems to limit copying or other use of digital media. Watermarking is much

more a forensic than a digital rights security tool - only special devices can read out the watermark. In this scenario, the source of large-scale pirated copies can be identified by determining the "batch" and source of the security leak. Theoretically, watermarks can be used for digital rights protection. However, the fact that the watermark is common across all copies and that all audience media players will have a copy of the "watermark checker" invites circumvention.

Watermarks can be removed or altered if copies are available with different watermarks. They can also be removed (or inserted, if necessary) if the watermarking scheme is reverse engineered.

#### 4.5 Fingerprinting

These are copy unique identifiers embedded in a piece of digital media. Fingerprints are actually placed inside of the media file - modifying it in small, undetectable ways that ensure that the fingerprint is present without distorting the base media. The digital fingerprint must match the identifier of the media player in order to be used as a digital rights tool. This technology has been proposed in order to track user authentication and limit copying. There are several fundamental challenges to the use of fingerprint systems: production of unique media for every user, ensuring that the "unique ID" of the player cannot be duplicated, and ensuring that the fingerprint cannot be removed.

#### 4.6 Covert Fingerprinting

Like Fingerprinting, Covert Fingerprinting embeds unique identifiers into each individual a piece of digital media. With this technology, customer media readers do not process or identify fingerprints. Rather, media distributors or their agents will scan widely distributed copies and use the covert fingerprints to determine the source of unauthorized copies.

Three challenges exist in the implementation of digital fingerprinting systems. The primary obstacle in the deployment of digital fingerprint systems is the registration of authorized consumers. While authorization can be integrated into an on-line sales process, this system does not mesh well with current traditional sales approaches (such as CDs or DVDs). The other practical problem is the creation of unique fingerprints for each copy of a base product (also technically difficult for traditional distribution). Finally, the main security challenge for digital fingerprint schemes is that fingerprints are easy to detect and alter, especially for large-scale pirates. The following process provides an example of how a pirate may work around digital fingerprinting.

A pirate can buy two legitimate copies of the media (COPY1 and COPY2).

She can then "add" the two copies together:

(COPY1 + COPY2)

Since each has the same base media that has incorporated the fingerprint, she can identify the fingerprint:

$COPY1 + COPY2 = MEDIA + FINGERPRINT1 + MEDIA + FINGERPRINT2 = FINGERPRINT1 + FINGERPRINT2$

Thus, separating the media from the fingerprint

The media components cancel out and the fingerprint can be identified and either 1, garbled so it is ignored or 2, removed entirely. Thus fingerprinting becomes much better at

protecting against casual "fair use" than against large-scale piracy. This approach will work with as few as 2 real copies, so even small-scale piracy is not too difficult.

Because many games support Web play, registration and download systems can incorporate watermarking and fingerprinting into the distribution process with much less effort than for physical media. Web game services themselves are a powerful DRM tool.

#### *4.7 Security Labels or Tags*

Security Labels (or "Tags") are supplementary tags that are appended to a piece of digital media and may also be bound to it by a digital signature (see "Signatures" below). Tags are typically used with proprietary encoding and post-processing systems to limit copying or other use of digital media. They can include simple copy counters, serial numbers, or other identification and use control information.

Tags are easily removed or altered, as they are a distinct portion of a digital media file or stream.

#### *4.8 Signatures*

Signatures wrap a piece of media with a tag that includes additional information but is also derived from the media itself. Signatures are used in combination with one of the other means of protection. The important attribute of signatures is that a signature verifier cannot also create a valid signature.

Signatures are most useful as part of an individual identification scheme and thus have the same problems with registration noted for fingerprints. Also, the signature is additional data that can be removed either directly or via the same type of scheme as described above for removing embedded fingerprints. If a signature is required by the media player (or game engine), then the discussion of attacks on proprietary encoding applies. Once the signature has been stripped, the data can be translated into an alternate format that is freely readable and does not require a digital signature for use. Thus, the attack circumvents the signature and does not defeat it.

As noted previously, because of the complexity of game engines, a signature-based solution, if properly designed, can be effective.

#### *4.9 DRM and Anti-Piracy Solutions*

There are a number of standard commercial anti-piracy solutions. Macrovision and Aladdin Data Systems have targeted the games industry and provide physical media security, software security, and token-based security systems. There are also shareware distributors like eSellerate that include DRM and anti-piracy as well as e-commerce features. There are a growing number of DRM companies that are focusing on the games industry. Most of these companies deliver an integrated distribution, DRM, and e-commerce solution (with try-before-you-buy, limited use, expiration, and other features). Exent, Real Arcade and Trymedia are in this category. Some game companies, such as Valve, with its Steam service, and Blizzard's Battle.net, have created their own solutions. IT GlobalSecure has linked Web game play to anti-piracy and DRM with its SecurePlay Keeper product.

## VI. Publishing

### A. Introduction

---

The casual games market is one of the most dynamic and prospering segments of the computer game industry. This is due to the growth in a variety of critical factors that constitute the structure and marketability of the business including: customer variety, product design, platforms and availability. This growth is not only observable in the form of units sold or revenue generated, but also in the definition of the types of game products that are being produced, and what types of companies are involved in bringing them to market. The information presented in the earlier sections of this paper clearly demonstrate that a "casual" game now exists in a diverse number of ways, and that companies from a variety of traditional and developing game industry segments all include themselves (and justifiably so) as active participants in the growth. What used to be a straight line value chain from product maker to product seller to consumer has now taken on a much more diverse and distributed form where many types of companies are able to bring computer game products to market.

The casual games market has broken this traditional model of how computer game products are brought to market. The change was not immediate or profound, but despite a slow beginning it has evolved into a significant and undeniable influence on the entire industry.

Today, the function of bringing casual game products to the market is not confined to organizations performing the traditional roles of publisher or distributor. The variety of players in this market and their varied roles are explained in the business models section of the paper.

### B. Surveying the Community

---

Given the diverse nature of players in the casual game value chain, a broad cross-section of players in the space were surveyed to learn more about how these various players bring products to market. The survey was open to all companies and individuals and was promoted through the casual games mailing list run by the Casual Games SIG, as well as through direct solicitation of previous respondents. The survey is ongoing and can be found here: [www.igda.org/survey/index.php?sid=13](http://www.igda.org/survey/index.php?sid=13)

For this first round of data collection, more than fifty companies submitted their views of the publishing function in the Casual Game market, more than three times the amount of surveys that were collected from direct interviews of companies in previous white papers.

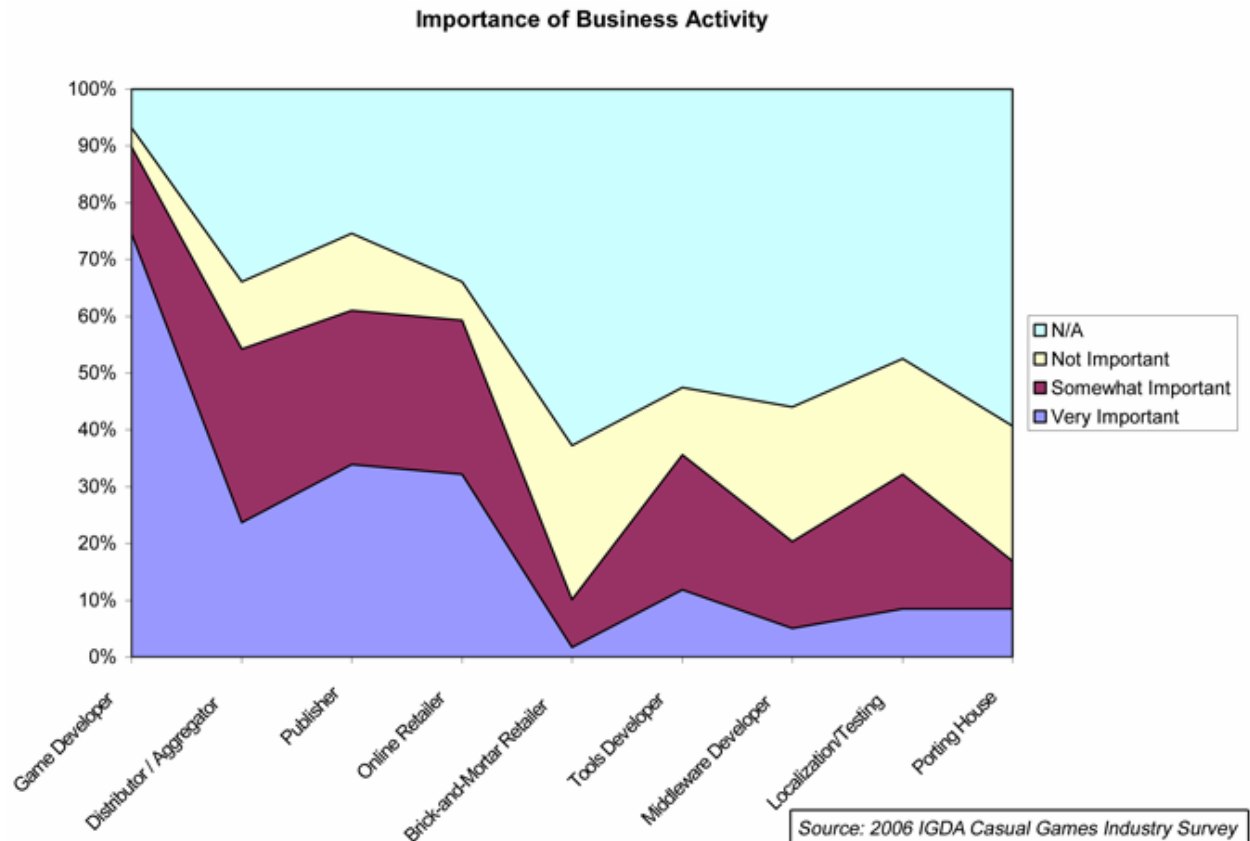
The survey gathered information ranging from business descriptions, to annual revenue, to product types produced to marketing strategies and royalty scenarios. No paper in any previous year has gathered so much data on the quantitative and qualitative matters of this growing market.

This data has been analyzed and is presented below to provide some insight into the publishing process for casual games today.

## 1. Casual Game Company Make-up

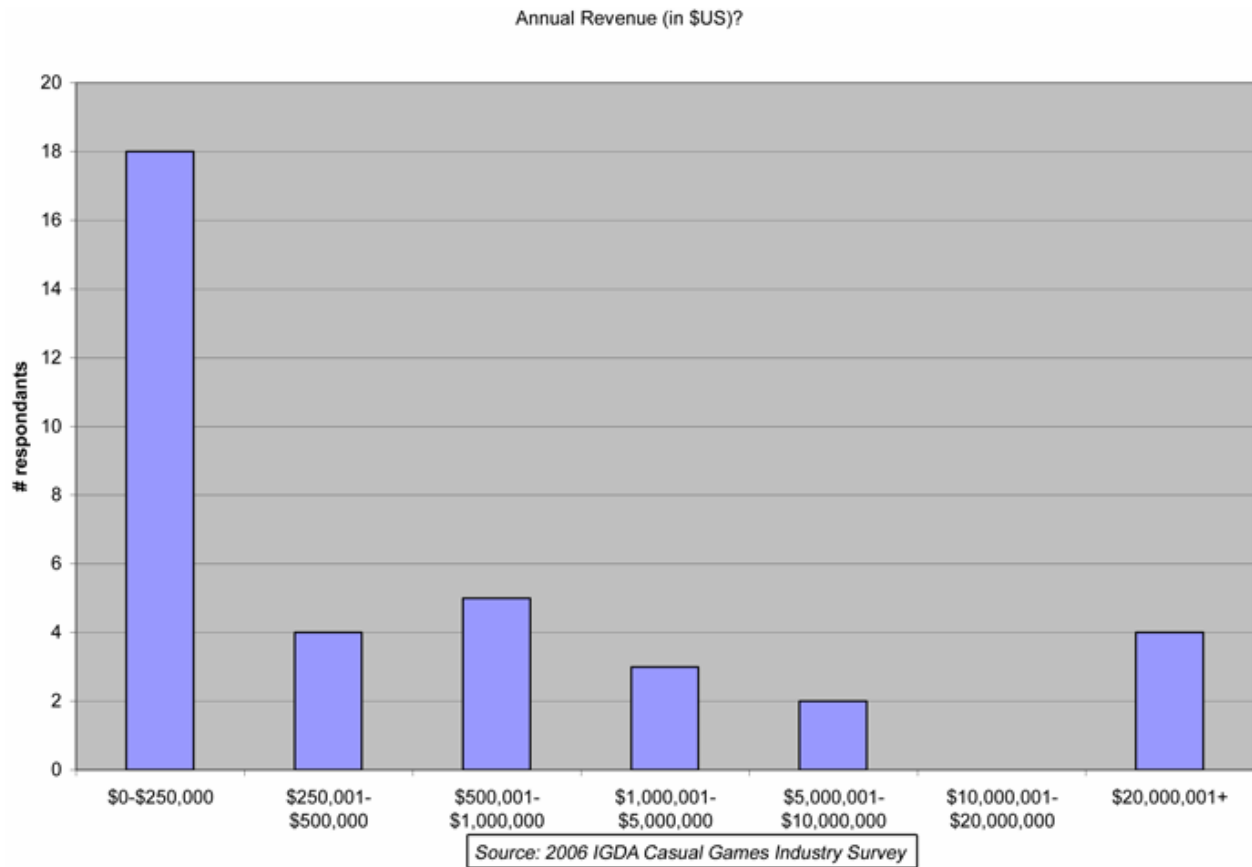
The following information presents statistics and responses that define the functions, size and market segments of the companies that participated in the survey.

### 1.1 Primary Business Functions



While the majority of respondents to the survey were game developers, over fifty percent (50%) still indicated that being a Distributor, Publisher and Online Retailer were Somewhat or Very Important to their business. This reflects the varied roles casual game companies must play in today's market.

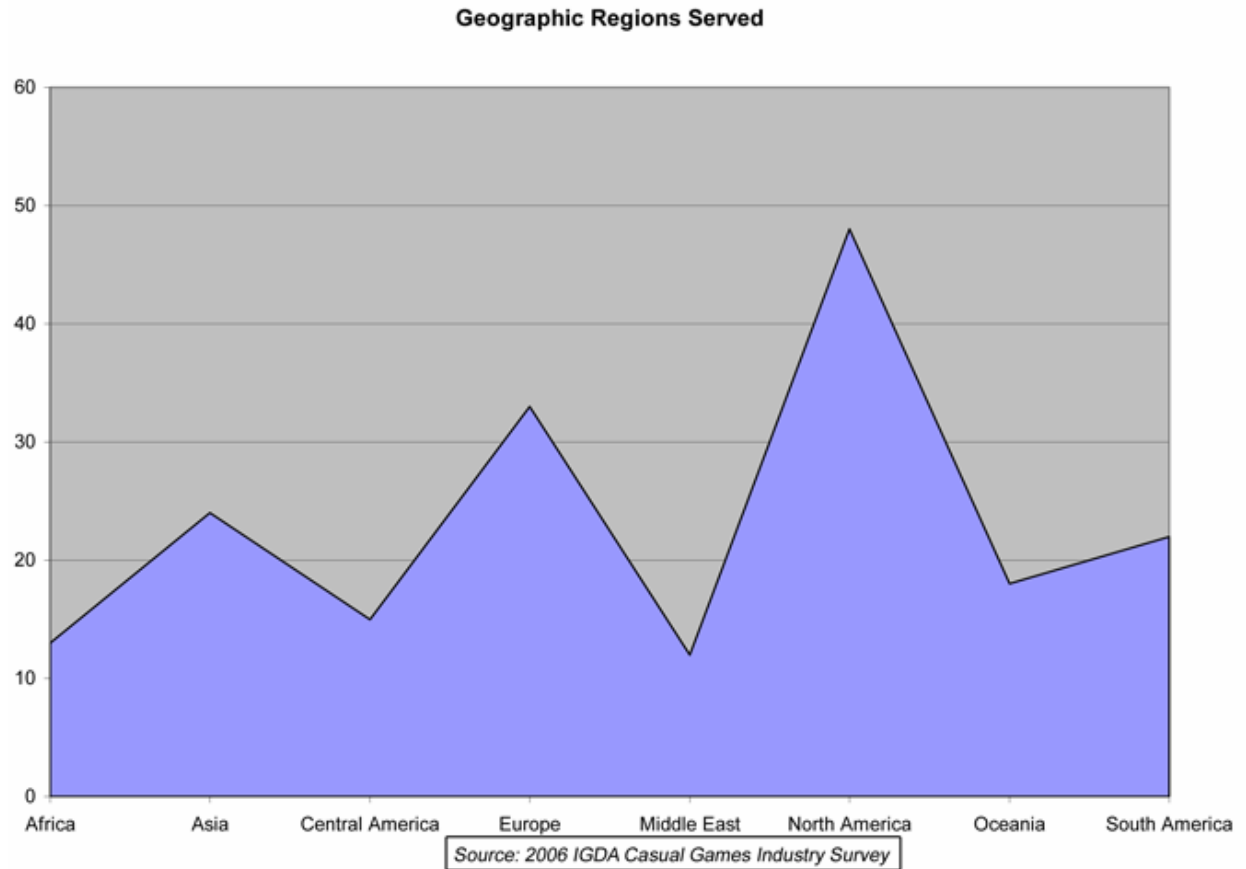
## 1.2 Annual Revenue



The majority of companies who answered this question generate less than \$250,000 in annual income. This shows that the casual game space is still comprised of numerous very small companies with presumably low overhead and small staff.

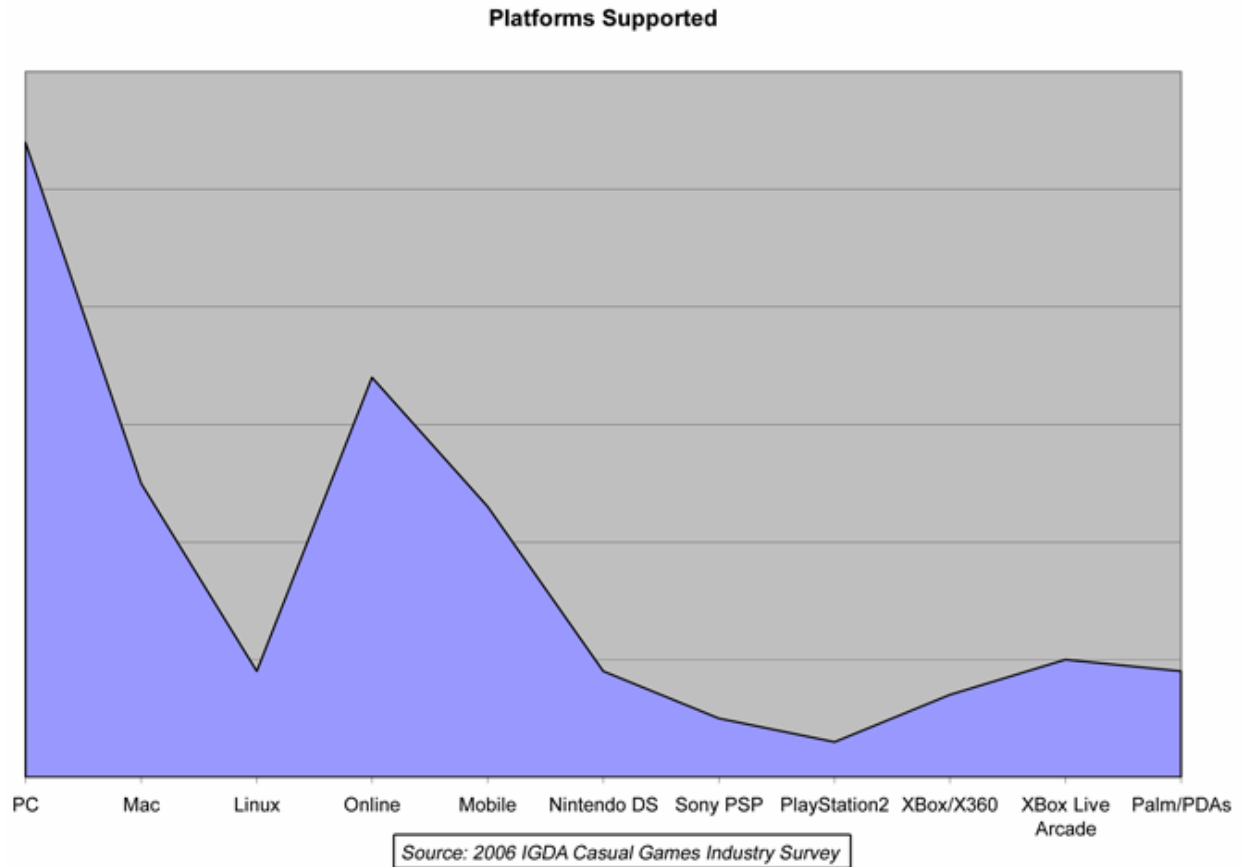


### 1.3 Geographic Regions Served



As expected, most companies target North America, with Europe not too far behind. Somewhat surprising was the strong showing of Asia and South America. This suggests that the casual game market is starting to expand into a truly international marketplace.

#### 1.4 Platforms Supported by Games Produced

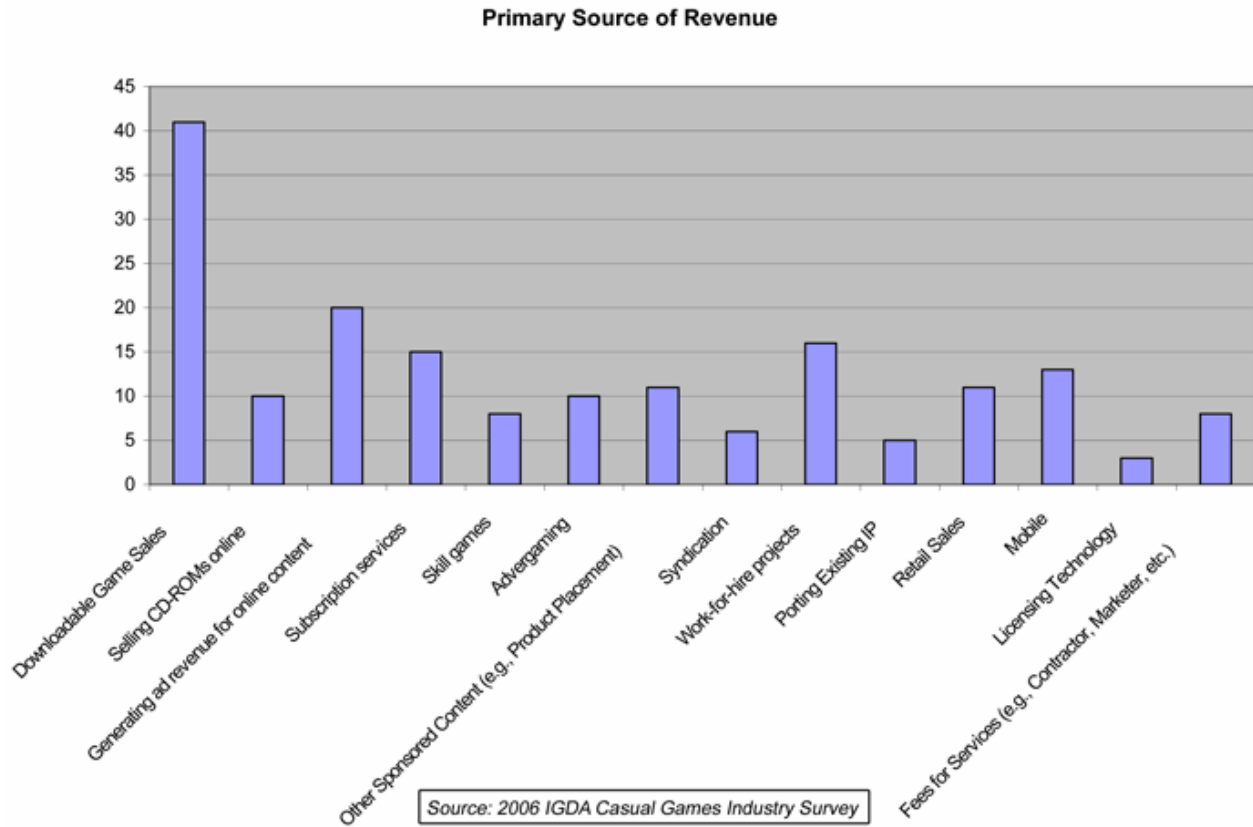


Not surprisingly most companies support the PC platform. Online and Mac also have a decent showing, but all other areas have low levels of support. Xbox Live Arcade has received a relatively large response here, despite it's new entrance to the space. This is probably a result of the success of the initial Xbox Live Arcade titles.

#### 2. Revenue and Performance

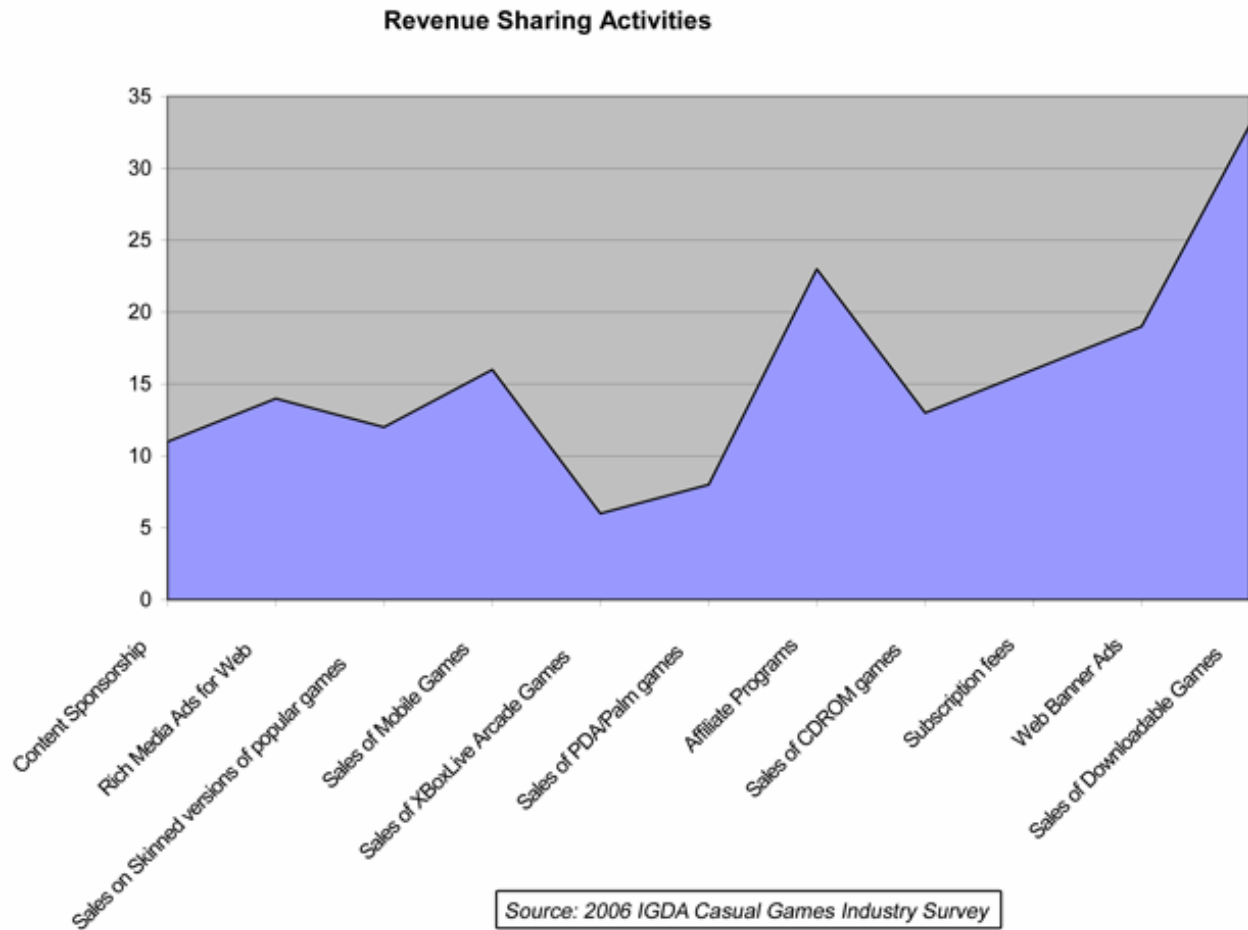
Survey respondents answered questions on revenue generation and product performance giving insight on how money is made, what types of products account for the most sales, and what types of revenue sharing structures they are involved in.

## 2.1 Revenue Generating Activities



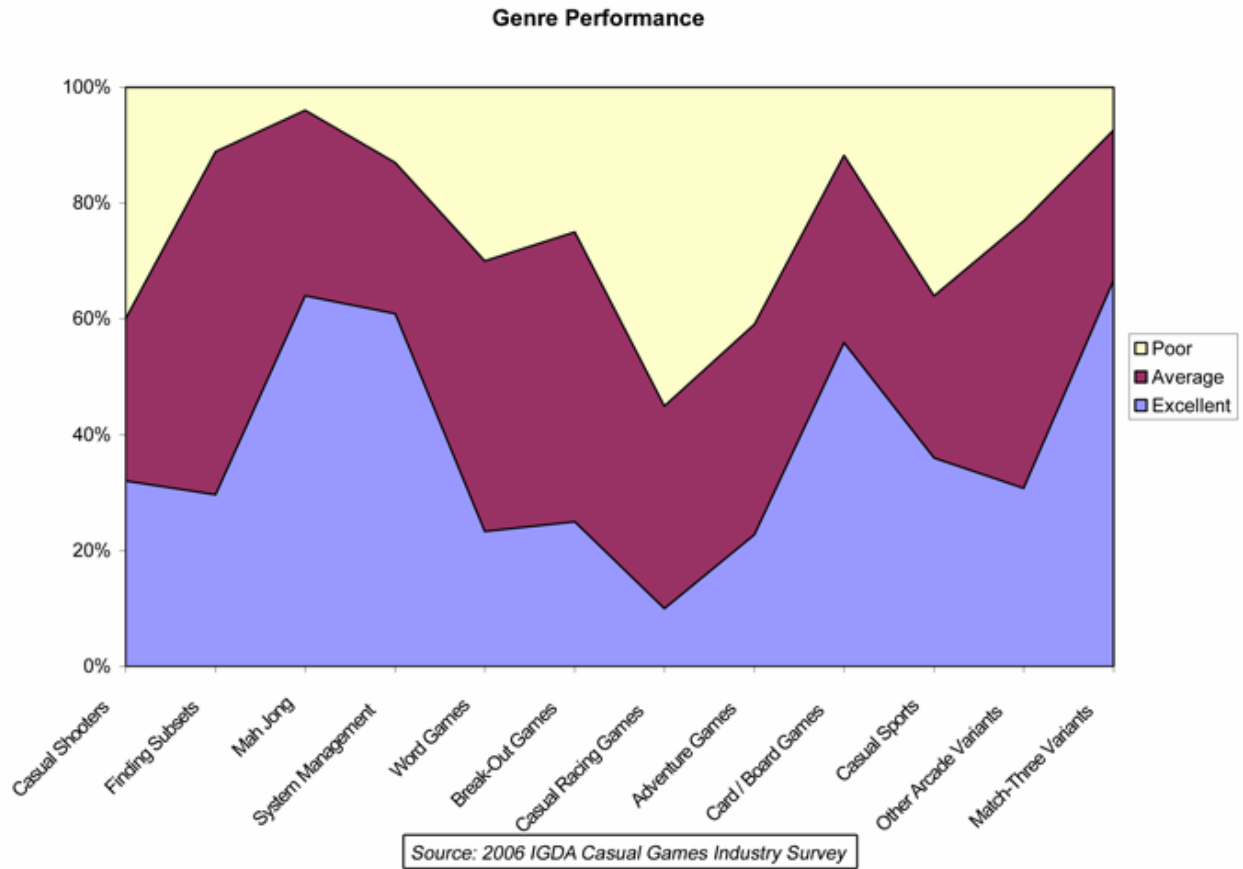
Most companies still see downloadable game sales as their primary source of revenue. The biggest point of interest however is that there is no other overwhelming method of revenue beyond the download. Many companies are trying numerous different things to varying levels of success, and this survey seems to suggest that most companies have a varied source of revenue.

## 2.2 Revenue Sharing Activities



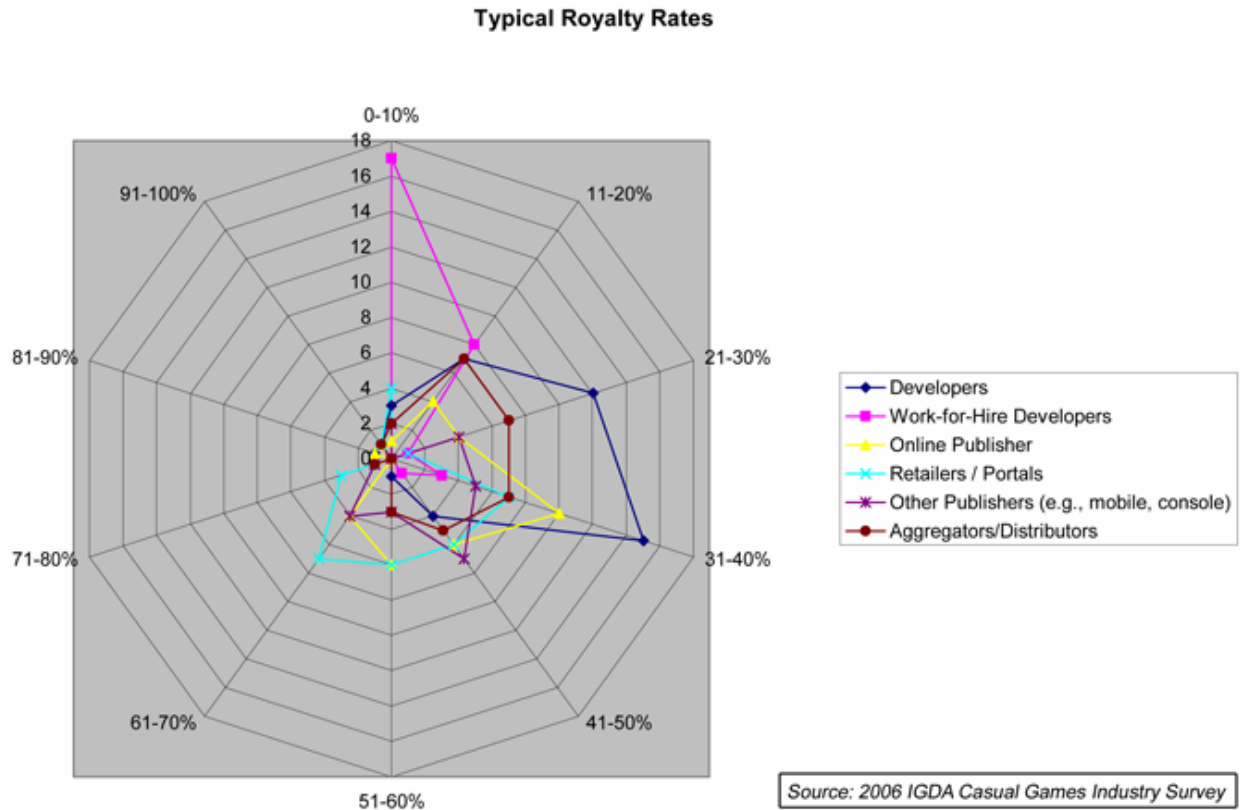
Again not surprisingly, most respondents reported the sharing of downloadable game revenue and affiliate programs. It also appears that a significant number of companies are sharing revenue from Web Banner Ads, suggesting that the trend towards sharing this revenue is becoming more prevalent across the industry.

### 2.3 Sales Performance by Game Category



While Match-Three, Card/Board Games, Mah Jong and System Management all continue to be excellent performers, there are very few types of games that seem to perform very poorly. Only Casual Racing Games has more reports of poor performance than Average or Excellent. This suggests that casual gamers are still interested in a wide variety of game styles.

## 2.4 Royalty Rates



The results from this question about typical royalty rates demonstrate how in-flux the publishing dynamic is today. According to the responses, Developers will typically see between twenty-one percent (21%) and forty percent (40%), Publishers between thirty-one percent (31%) and fifty percent (50%), Retailers between thirty-one percent (31%) and seventy percent (70%), and Aggregators/Distributors between eleven percent (11%) and fifty percent (50%).

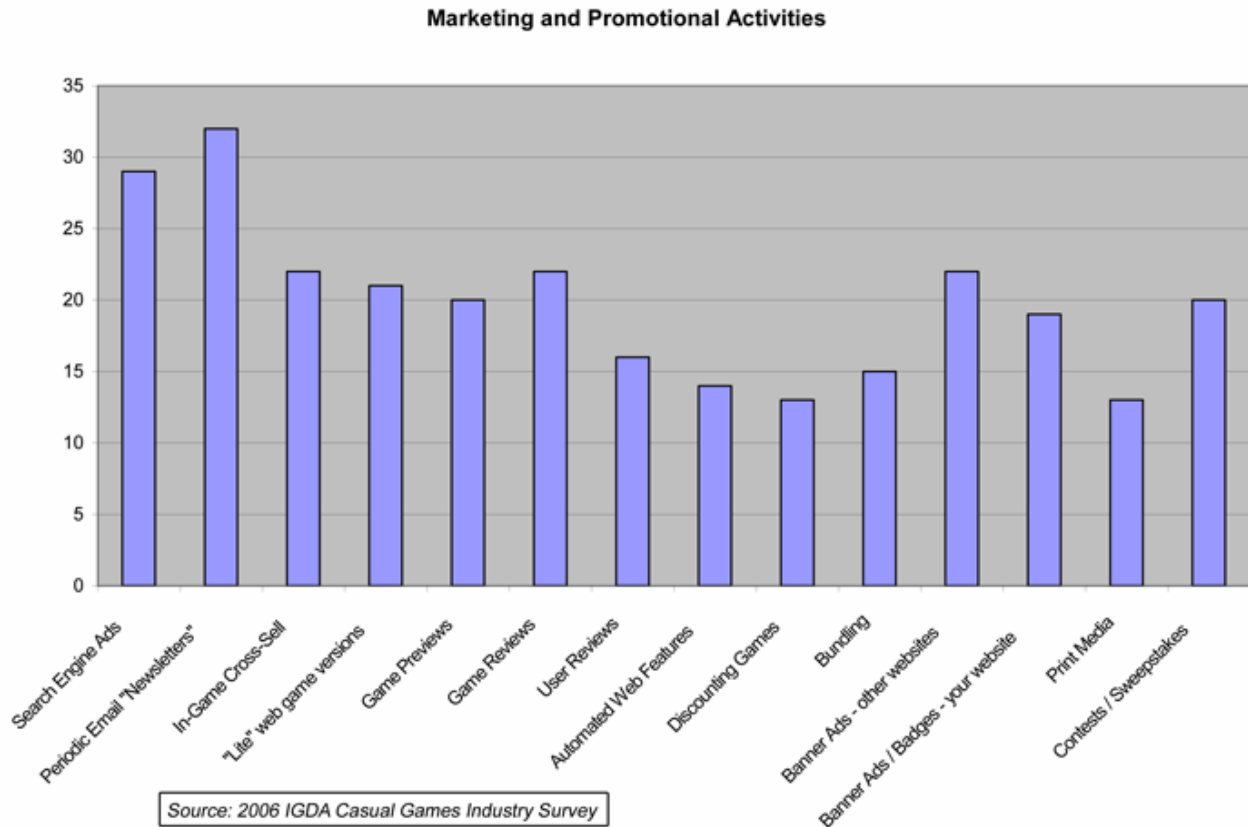
### 3. Marketing

Respondents were asked to define their target markets and how they market their products to them.

#### 3.1 Target markets

Most companies indicated that they target casual gamers. Several respondents further qualified that by age or gender, geographic location (i.e., Latin America), or platform (i.e., Mobile or XBox 360). A couple responses included all gamers within their target markets. Middleware, Tool and service companies also included ad agencies, publishers and developers among their target markets.

### 3.2 Marketing Mix



As seen in previous questions, it appears that there is a wide variety of means currently being used to market and promote casual games. Email Newsletters and Search Engine Ads are the most widely used, but not by a great margin. Discounting Games and Print Media are utilized the least, but still appear to be relatively commonplace methods for promotion.

### 3.3 Marketing Philosophy, Approach or Strategy

While there are a wide variety of approaches to marketing games, and the philosophy and strategy behind it, several trends are seen in the responses. Numerous developers indicated that their marketing is based almost completely on developing strong solid titles, and then depending on word of mouth and growing consumer loyalty and brand recognition. Other developers rely completely on portals and publishers to sell their products, while they focus solely on developing new games.

Others prefer to take a more active role and work to get the word out as widely as possible, and find cost effective ways to promote products themselves.

One publisher's response outlined the value of working delegating the roles of promotion and marketing to a company that is dedicated to it:

*... Our approach is to let the customer drive what we do and get meticulous early in the cycle about the unique selling proposition of the game. Pre-launch activities include: consumer research and testing, market intelligence, naming, positioning, copywriting, trademark clearance,*

*box art design, PR pre-pitching, and channel marketing/sell-in. Within consumer research we have many methods, such as 1:1 game panel testing, usability testing, focus groups, and a formal consumer beta program. The beta program allows consumers to give qualitative and quantitative (behavioral) feedback on a game for fine-tuning and polish. Launch/ongoing marketing activities include: PR, advertising, channel feedback, business model / price testing, customer service, and direct marketing.*

Several responses also indicated the value of attending industry events and conferences to network with peers and make the business contacts necessary to successfully work with all companies in the value chain.

#### **4. Business Development Tactics and Competitive Advantages**

Respondents describe partnership strategies that are a significant part of the unique growth of the Casual Games space. They also explain in their own words the unique skills they possess to compete in the industry:

##### **4.1 Partnerships**

The majority of responses reflected the need for collaboration to be successful in the space. Developers will partner with publishers and distributor/aggregators. Publishers partner with developers and distributors, etc.

Beyond that there is also a significant number of responses that show the open and collaborative community that still exists in this space. Companies indicated that they share projects with other developers, as well as ideas, methodologies and common knowledge about the space. A typical response was seen from this developer.

*We consistently exchange ideas with other developers in the industry. There is enough pie to go around.*

#### **5. Competitive Advantages**

The differences in competitive advantages in this space are as varied as the companies and business models. Most responses related to an expertise or specialty in a niche area of the business, be that particular game styles (like word or solitaire), geographic location (Southern Hemisphere), or target market (corporate training or advergaming).

The wide variety of responses seen here shows once again that the casual game space is still a nascent industry with relatively little amounts of competition among the proven and established companies. Each seems to have their own specialty and expertise even though there is a large amount of apparent cross over in target audience.

#### **6. Companies Surveyed**

Listed below are the companies who had responded to our survey at the time the data was collected and analyzed. This is not intended to be an exhaustive or thorough list of casual game publishers. While we solicited responses from all of the top players in this space, not all of them were able to respond to the survey. However, we do feel that we achieved a broad enough cross section of companies for the data to be valid across the entire industry. Companies were asked to provide their company name, main website, and year of founding to get an idea of the approximate age of the players in this space.



3RD sense [www.3rdsense.com](http://www.3rdsense.com) founded 2001  
Akel & Associates 2005  
Alawar Entertainment, Inc. [www.alawar.com](http://www.alawar.com) 1999  
Artex Studios, Inc. [www.casualgamestore.com](http://www.casualgamestore.com) 2001  
Atari Inc. [www.atari.com](http://www.atari.com) 1972  
Atrativa Latin America S/A [www.atrativa.com.br](http://www.atrativa.com.br) 2000  
Bamtang Games [www.bamtang.com](http://www.bamtang.com) 2002  
BeachWare [www.beachware.com/](http://www.beachware.com/) 1990  
Big Fish Games [www.bigfishgames.com](http://www.bigfishgames.com) 2002  
Boomzap Pte Ltd [www.boomzap.com](http://www.boomzap.com) 2005  
Cj Internet [www.netmarble.net](http://www.netmarble.net) 2000  
Crux Games [www.cruxgames.com](http://www.cruxgames.com) 2005  
Def-Logic Ltd [www.def-logic.com](http://www.def-logic.com) 2001  
Délirus Entertainment [www.delirus.com.br](http://www.delirus.com.br) 2001  
Englobe Inc. [www.englobe.com](http://www.englobe.com) 2005  
Fun Factor Games, LLC [www.funfactorgames.com](http://www.funfactorgames.com) 2006  
FUN Technologies [www.funtechnologies.com](http://www.funtechnologies.com) 2002  
Game Trust, Inc. [www.gametrust.com/](http://www.gametrust.com/) 2002  
Games2Win [www.games2win.com](http://www.games2win.com) 2005  
Golden Goose Game [www.goldengoosegames.com](http://www.goldengoosegames.com) 2004  
Goodsol Development Inc. [www.goodsol.com](http://www.goodsol.com) 1995  
Gridwerx [www.gridwerx.com.au](http://www.gridwerx.com.au) 2003  
Haight, Brown & Bonesteel [www.hbblaw.com](http://www.hbblaw.com) 1937  
i squared learning [www.i2learning.com](http://www.i2learning.com) 2004  
Intermix Media, Inc. [www.grab.com](http://www.grab.com) 1999  
Jolly Bear Games [www.jollybear.com](http://www.jollybear.com) 2004  
KDF Infotech [www.kdfinfotech.com](http://www.kdfinfotech.com) 2006  
Large Animal Games [www.largeanimal.com](http://www.largeanimal.com) 2001  
Logler.com [www.logler.com](http://www.logler.com) 2006  
Mach8 [www.mach8.nl](http://www.mach8.nl) 1999  
Metaplay, Inc. [www.metaplay.com](http://www.metaplay.com) 2002  
Microsoft [www.microsoft.com](http://www.microsoft.com) 1976  
Mofactor [www.mofactor.com](http://www.mofactor.com) 2005  
Moondance Games [www.moondancegames.com](http://www.moondancegames.com) 2003  
MSL Inc [www.mslgaming.com](http://www.mslgaming.com) 2004  
Nanogames Multimedia Developers [www.nanogames.co.za](http://www.nanogames.co.za) 2005  
Pet Tomato, Inc. [www.pettomato.com](http://www.pettomato.com) 2005

PlayFirst, Inc [www.playfirst.com](http://www.playfirst.com) 2004  
Reflexive Entertainment [www.reflexive.com](http://www.reflexive.com) 1997  
Riddler LLC [www.riddler.com](http://www.riddler.com) 2002  
Sabarasa Entertainment srl [www.sabarasa.com](http://www.sabarasa.com) 1996  
Sillysoft Games [www.sillysoft.net](http://www.sillysoft.net) 2002  
SJS [www.desgolf.com](http://www.desgolf.com) 2005  
Skunk Studios [www.skunkstudios.com](http://www.skunkstudios.com) 2001  
Slingo, Inc. [www.slingo.com](http://www.slingo.com) 1995  
Smart Box Design [www.smartboxdesign.com](http://www.smartboxdesign.com) 2003  
TableStar Games [www.tablestargames.com](http://www.tablestargames.com) 2004  
Taparo [www.taparo.com](http://www.taparo.com) 2006  
Tournament Games Inc [www.tournamentgames.com](http://www.tournamentgames.com) 1998  
TrayGames [www.traygames.com](http://www.traygames.com) 2003  
Twitchy Thumbs Entertainment, Inc. [www.twitchythumbs.com](http://www.twitchythumbs.com) 2005  
Vampirosiames Games [www.vampirosiames.com.ar](http://www.vampirosiames.com.ar) 1999  
Wendigo Studios Inc. [www.wendigostudios.com](http://www.wendigostudios.com) 2006  
WildTangent, Inc. [www.wildgames.com](http://www.wildgames.com) 1997  
Word of Mouse Games [www.womgames.com](http://www.womgames.com) 1996

## CLOSING

We thank you once again for your interest in the 2006 Casual Games White Paper. This paper is the result of the efforts of numerous volunteers, and it is only with their continued support that we can provide projects like this to the community.

This paper will continue to live on through the IGDA's wiki. If you have found any errors, omissions, or areas where you can contribute, please make the appropriate changes on the wiki:

**[www.igda.org/wiki/index.php/Casual\\_Games\\_SIG/Whitepaper](http://www.igda.org/wiki/index.php/Casual_Games_SIG/Whitepaper)**

## VERSION HISTORY

6/26/2006 – Initial Release

7/1/2006 – Updates to Skill-Based Gaming

7/11/2006 – Added Steven B. Davis as a contributor to Technology