



Volume 2, Issue 2
Winter 2007

Letter from the
Editor

Bookworm
Adventures
Postmortem: The
Good, The Not Bad &
The Ugly

Breaking Ranks:
Hacking
Tournaments,
Ladders, Contests,
and Multi-Player
Games

Slingo Quest
Postmortem

Q&A with Richard
Wienburg, Softwrap
Inc.

Improving
Communication with
Your Sound Designer

Top 10 Data Review
for 4th Quarter of
2006

Windows Vista and
Casual Games
Development: What
developers need to
know

Download a PDF Version
of this issue.

Subscribe to the IGDA
Casual Games SIG
Mailing List

Archives:
Fall 2006
Winter 2005
Summer 2005

Letter from the Editor

Welcome to the Winter 2007 issue of the IGDA Casual Games Quarterly.

We've got another great collection of articles in this Winter 2007 issue of the IGDA Casual Games Quarterly. Rob Carroll gets to the bottom of all the Vista questions in an interview with Microsoft's Kim Pallister. PopCap's Tysen Henderson and Jeff Weinstein take us behind the scenes of Bookworm Adventures. Funkitron's Dave Walls turns in an excellent post-mortem on Slingo Quest. Jason Akel contributes another informative interview from the DRM world, this time with Softwrap's Richard Wienburg. Steven Davis explores security issues related to online games. Kane Minkus breaks down the terms and processes of game sound designers and composers. Finally, James Smith returns with more useful highlights from the past quarter's worth of market data.

Do you have ideas for content for the Casual Games Quarterly? We are always open to ideas for articles, regular columns, interviews, and more. Would you like to get involved with the Casual Games SIG? Email us or join the mailing list using the link on the left of this page.

Thanks!

Wade Tinney
Editor - IGDA Casual Games Quarterly
Partner, Game Designer - Large Animal Games
Chapter Coordinator - IGDA New York City
wade[at]largeanimal.com

Table of Contents

[Letter from the Editor](#)

[Bookworm Adventures Postmortem: The Good, The Not Bad & The Ugly](#)
By Tysen Henderson and Jeff Weinstein

[Breaking Ranks: Hacking Tournaments, Ladders, Contests, and Multi-Player Games](#)
By Steven B. Davis

[Slingo Quest Postmortem](#)
By Dave Walls

[Q&A with Richard Wienburg, Softwrap Inc.](#)
By Jason Akel and Richard Wienburg

[Improving Communication with Your Sound Designer](#)
By Kane Minkus

[Top 10 Data Review for 4th Quarter of 2006](#)
By James C. Smith

[Windows Vista and Casual Games Development: What developers need to know](#)
By Robert Carroll and Kim Pallister

Volume 2, Issue 2
Winter 2007

Letter from the
Editor

Bookworm
Adventures
Postmortem: The
Good, The Not
Bad & The Ugly

Breaking Ranks:
Hacking
Tournaments,
Ladders, Contests,
and Multi-Player
Games

Slingo Quest
Postmortem

Q&A with Richard
Wienburg,
Softwrap Inc.

Improving
Communication
with Your Sound
Designer

Top 10 Data
Review for 4th
Quarter of 2006

Windows Vista and
Casual Games
Development:
What developers
need to know

Download a PDF
Version of this issue.

Subscribe to the
IGDA Casual Games
SIG Mailing List

Archives:
Fall 2006
Winter 2005
Summer 2005

Bookworm Adventures Postmortem:

The Good, The Not Bad & The Ugly

By Tysen Henderson, Art Director/Producer, PopCap
and Jeff Weinstein, Software Engineer, PopCap



January 9, 2007

Developer: PopCap Games
Release Date: December
2006
Platform: PC
Engine: PopCap Framework
Dev Time: (Concept to
Release) 30 months
Core Team:

- Tysen Henderson, the producer
- Jeff Weinstein, the programmer
- Bill Olmstead, the animator
- Stephen Notley, the writer



Bookworm Adventures (BWA) marks the return of everyone's favorite lexicographical annelid. Help Lex the Bookworm on his adventures through three unique worlds of literature, whomping mythical foes, collecting treasures and gems, and encountering a cadre of friendly companions. Test your wordsmithing prowess by forming the most polysyllabic verbiage possible, utilizing the elemental power of gem tiles and an arsenal of artifacts.

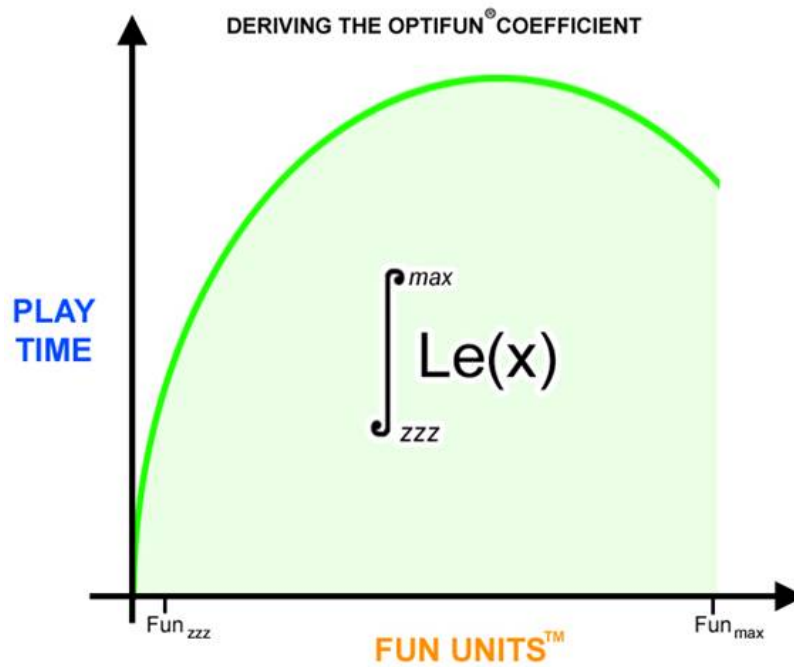
Or in summary, spell words, beat baddies, power up.

For an inside look into the making of Lex's adventure, read on!

DESIGN

The Good: The Fun Curve

One design success of Bookworm Adventures is its unique blend of casual gaming and role-playing elements. While definitely being a casual game, it is accessible and appealing to two disparate audiences: casual word gamers and traditionally hardcore role-playing gamers.



100 Super Computers and 100 days later...

Early in the process we understood that marrying the worlds of casual word games and role-playing games came with inherent risks and rewards. If done well, Bookworm Adventures would excite and entertain both audiences, offering each an exciting, unique experience. If done poorly it could alienate and frustrate both as an awkward monstrosity. No one likes awkward monstrosities. It was therefore crucial that the core mechanics, added layers of gameplay and user interface of the word and role-playing aspects of the game were as easy to learn as the ABCs. In this effort we feel we have done exceptionally well.

The mechanic of forming words in BWA is actually simpler than in the original Bookworm, establishing an intuitive foundation. The restriction of letters needing to be adjacent was done away with completely and very early on. Another simplification was the general rule that larger words always garner greater rewards than smaller words, whether resulting in direct damage, gem tile creation, or bonus round rewards. This allows a given player to focus on only spelling. They can ignore treasure-selection and potion-usage strategies and still successfully progress through the game. The turn-based structure allows players to play at a more relaxed casual pace.

On the flip side, a role-playing gamer can instantly grasp the importance of proper gear for certain scenarios and the power of an emergency heart-refill or a well-timed damage boost with the chugging of a color-coded beverage and feel right at home with BWA. With one set of treasures and an invigorating quaff of a power-up potion the game becomes one of quick, hard-hitting polysyllabic domination. With another set of treasures and an eye on the heart meter, the game becomes more deliberate and centered around defensive play and survival. Once a strategy is selected, an "rpg'er" can then focus on verbosity and forming the lengthiest word they can imagine (or even stumble upon some they didn't know).

Introducing each layer of gameplay was also done with care. The most potent portions of that process were a succinct integrated tutorial, carefully-revealed interface and reiterative arrow tips. The first several chapters of Lex's adventure guise the tutorial in layers of fun and humorous banter. As new gameplay is introduced the tutorial wording is succinct and spoken as character dialog. In tandem, the corresponding bits of interface are gradually revealed, so the player only sees the Level Up Bar after it is taught to them and needs not worry about it until Level Up points can even

be gained. Finally, detailed arrow tips pop up pointing directly at UI elements and punctuate each lesson with expandable detailed descriptions.



Left: Terse. Right: A much more grammatical and expanded descriptive format.

The opening experience of the game enriches the player with each added layer of game mechanics, treating them as new abilities rather than more bars and meters to stress over, which empowers the player rather than simply educating them.

The Not Bad: Maximum mega-monsters, minimum megabytes

So the game is designed to be fun and accessible. In fact, it is so fun that players hope the game will never end. Gulp.

Designing the rules for the world Lex would adventure was a challenge all by itself. Compounding this was the intended scope and richness of that world. Our first adversary for Lex was a more serpentine incarnation of our favorite gorgon, Medusa. In bringing her to life we soon came to realize that using traditional techniques, the world we promised Lex would either have to be a disappointing half-handful of monsters or a prohibitively large behemoth of a download.

We thus designed a proprietary animation system, "PAM animation" that loosely resembles both the interpolated motion of Macromedia Flash and that of a marionette to realize our goals. Each animation would be made of several separate bits and appendages (lower arm, upper arm, hand, shoulder, etc.), be sequenced by hand and formatted to a .pam file, and would then be interpreted and smoothly animated by the game engine itself.

Instead of 430KB for a monster with 5-6 animation filmstrips, we were looking at a tenth of that, for a fully animated creature composed of 20-40 art assets.



10 Monsters for the price of one! Call now while supplies last!

In accomplishing the feat, many design sacrifices had to be made. Any player of BWA will attest that early on the enemies appear or sound similar. Each one animates in a completely unique ways, but something is oddly familiar. For example, a cyclopean shepherd is reinforced by a cyclopean warrior, and in turn followed by a cyclopean chieftain. Further on one encounters countless variations of ravenous wolves, screeching eagles and roaring lions. Reusing and repurposing core monster art soon became a mantra of level design and plot tweaking. Another tactic for stuffing the game to the gills was recombining individual pieces of monsters. Greek mythology lent itself well to this, as throwing the head and wings of the aforementioned eagle on to the body of a lion meant a whole new creature, a gryphon, which we could give both eagle screeches and lion roars. Very quickly we formed a library of thousands of limbs, torsos, heads, weapons and armory and hundreds of flexible-use sounds, and we used this library to create each and every monster in Bookworm Adventures.

The consequences of this method were both good and bad. On the downside, many monsters became repetitive, relying on color shifts and accessories for their "unique" look. Certain monsters were cut out of the cast due to their low reward versus high added asset ratio. Creating a gigantic bronze foot and several booming, clattering metallic sounds so that Lex could combat Talos was simply out of the question. On the upside, Bookworm Adventures boasts over 150 vocal monsters, a score of characters including our hero, three-layer parallaxing backgrounds and flurries of fire, poison and ice — all for under 20mb. Each monster introduces themselves, reacts to Lex's multitudes of attacks, unleashes their own assorted bag of tricks and quite satisfyingly crumple to the earth upon defeat, releasing their treasures for the cheerful wormy hero to collect. The final product is an epic world of words that spans three core 'books' of literature and offers the player hours of enjoyment.

The Ugly: Nerdcreep!

Being avid gamers ourselves, each of us had played several RPG titles before taking up the reins of BWA. It was natural for all of us to want to add our favorite parts of each RPG we'd ever power-grinded through and to add a slew of complex interactions, character statistics or skills and reward systems to quench the inner drooling RPG nerd. While we largely succeeded in keeping the hard core out of our casual game by ship date, many design ideas were thought up, meticulously implemented and then later completely stripped out and burned along the way, increasing development time

and adding needless complexity.

At one point in the game, Lex also gained gold from his foes. The gold commodity was initially intended as a secondary reward system for combo-type play (spelling many words of a certain category or length repeatedly) to solve the problem of how to reward players with spelling large words that "overkill" a monster. The absence of points seemed to beg for this feature.

Another example was "gem alchemy". Often a player would use multiple low-level gem tiles in a single short word to deliver a fatal blow. In doing so, the player would be left with a single lonely gem tile, and often never access the highest tier gems. To solve this, combinations of gem tile values would determine a much better gem to be generated.

The result was a set of cumbersome reward systems that offered little added fun or depth and foreshadowed highly complex instructions and descriptions in the creation of the tutorial. Soon each would be torn out, leaving behind an elegantly simple set of rules.

While prototyping these complex design ideas defined a hard line that we did not want BWA to cross, some of the ideas were conceived of purely because they fit the RPG genre, and were fueled innocently by the power gamer lobe of our brains. Design lesson hammered in: never add complexity for complexity's sake.

PRODUCTION

The Good: Power of the People

The best production asset BWA had going for it was the team itself. Each member's level of dedication and combination of overlapping talents and skills made division of tasks, workflow and rounds of iteration clear-cut and efficient. As evidenced in the credits, multiple names appear in multiple roles. On a given day the artist might rebalance values in a Lua file, the programmer might design a special effects sequence, or the writer might come up with a new character/monster design. With significant overlap and a tight-knit team of friends, production of BWA allowed for more experimentation, with quicker iterations of ideas at key times. When it didn't seem like a long grind, it almost seemed like play. Also, team members could switch from role to role both to finish a necessary task more rapidly, and also to combat monotony and burnout. Everyone also lent heavily to the plot and story, each person adding their own humor and nuance to the game, evident in the whimsical, goofy characters Bookworm Adventures exudes. Supporting the core team was a great QA staff who came to know the rules, balancing and scope of the game as well or better than the core team, and offered their own insights to improve the game. Beyond that was a legion of PopCap Beta Program testers, friends and family to realign all of our objectivity as the game's development marched on. Reaching the end of the project in the haggard, frazzled burnt-out states we were in, the core team surely might have been a tad more insane without the adaptability and capability of wearing so many hats (sometimes many at once) and without the support of so many.

The Not Bad: Proprietary Tools

You've heard of PAM animation, have you? You may have heard of Lua along the way as well. (If not, skip down to the Engineering section to get your fill.) To tackle the problems of BWA's huge scale, new technology was adopted, modified, supported and even invented from scratch. As production went forward the tools also had to be developed, supported, and revised. This often led to hang-ups in the workflow and some frustrating desires to throw dev boxes right out the window into traffic.

While PAM animation meant we could reuse art assets,

rapidly design and develop animations and effects, a constant barrage of asset referencing errors, hiccups and visual synching problems existed as well. While Lua meant rapid game balancing, monster implementation and copy editing, any minor flaw in syntax or logic resulted in hard locks and game crashes on run-time with no warning. Sometimes it was unclear whether a personal error had been made, or if we had reached some limitation of the technology itself. Hunting down and fixing these errors often took as much time as usage of the proprietary tool had supposedly saved. It was as if we were not creating just Bookworm Adventures, but also building a box of tools and bug testing them at the same time.



You too can Frankenstein up your own monster from vast libraries of limbs!

While the creation and use of new technology and tools added a significant amount of time to BWA's cycle, it also made the game possible. It is not guaranteed that the tools will be used any time soon on other projects because of the grief they cause, (and lingering mental scarring), but the shipped title seems to have justified the means.

The Ugly: Reeling in the Whale

Coming in at just under 30 months, Bookworm Adventures was no small production. In its original design doc, Bookworm Adventures was five unique books with ten chapters and six to seven monsters each, and even had skeletal descriptions for an infinite-play mode. If the grand dream was going to happen at all, the scythe would have to be reared and cuts were going to have to be made. But even the final project was immense, and to put it nicely, "ambitious". Whether the game would have been just as successful with less detail, less content or less layers of complexity is hard to say, but the precedent for the level of quality was set early, and would have to be kept to for the remainder of the development. With hindsight perhaps a slightly lower bar and slightly less grandiose visions of BWA might have been advisable. Perhaps we simply didn't realize how massive of a game we were really creating until suddenly we were a year and a half into production with the end still well past the horizon.

Another production point which has had the spotlight blazed upon it by recent press is the financial cost of BWA. The most common number publicized is in magnitude of 700k US\$, which at first glance is quite a large number for any casual game, even one of BWA's large scope. While Bookworm Adventures might have been a more manageable undertaking for a group of 15-20, it was in fact a whale of an undertaking for a core team of one producer/designer/artist, one programmer/designer, one full-time animator, and one

writer. With a little math (700,000 / 4 Core members / 2.5 years), you arrive at a less astronomical number of 70k per core team member. Even these numbers do not take into account other miscellaneous factors, but one begins to understand what the number truly means.

Further, smaller team sizes have always been key to the tight creative processes in the PopCap studio, as you might have picked up from the "Good" section above, and as many in the game industry can attest to, doubling or tripling team sizes does not necessarily halve or third the production. While we could have used another set of sane stable hands at times, especially during the long final content grind to deal with more repetitive tasks, touching every aspect of BWA personally gave us a definite sense of creative satisfaction, ownership and assurance of productive quality that is often lost with larger team sizes.

ENGINEERING

The Good: Nifty New Technology!

We knew from the start that the game was going to have a large amount of monsters, all with varying attacks, defenses, weaknesses. We also knew that we were going to have treasures and items with various properties and that everything had to be able to interact no matter how things were combined. We also needed a way to rapidly make code changes to be able to prototype ideas and test out complex effect sequences without the tedious cycle of compiling, loading, getting to the right point and setting up the right situation - only to discover that things needed to be tweaked yet again. Because of this, we decided to use the scripting language, Lua. Lua allowed us to develop a more modular game that could be easily expanded and tested without needing to deal with recompilation.

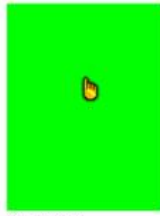
We heavily modified the Lua interpreter to support many advanced features. The first thing we did was to change the syntax to be almost identical to C/C++ since the BASIC-like coding style was irksome, and to add in sorely missed operators such as +=, *=, ++, etc. We then added a bunch of glue code so that on the C++ side we could easily expose methods that could be called from Lua, and vice-versa. Furthermore, we made a series of incredible performance enhancements to the execution speed and memory management of the interpreter. We greatly decreased table access times and improved the performance of the garbage collector. And to aid in rapidly designing code we added a complete state synchronizer that allowed you to modify any Lua code and reload it while the game was running. This meant that if, say a monster's attack algorithm wasn't what you wanted, you could tweak it in real-time and see the changes instantly as you refined it, without having to shut down, recompile, get back to that same point, etc.

But probably the coolest and most useful technology we developed was an integrated debugger for Lua that ran inside Visual Studio 2005. With this, we had a full featured visual debugger using our native IDE that could modify/examine variables, set breakpoints with conditional expressions, walk up the call stack, and be able to trap execution errors right on the line they occurred.

The modular design of the game meant that more complex monster attacks, defenses, resistances, etc. could be created easily. By effectively chaining together small chunks of "effect" code (each "effect" did a small but useful feature, like stun, or cause burning over time) we could create more interesting and varied attacks, as well as create treasures and items that could enhance, nullify, or respond to any type of effect. In addition, since the code for the enemies and the effects was kept simple and well documented, non-programmers could easily modify/create content and experiment with tweaking parameters to quickly balance the game, without needing a developer to implement things.

Built on the existing PopCap Framework, we were able to leverage rapid prototyping as well as rely on the stability of a code base that's been deployed on millions of computers over the past 6+ years.

This is where you'd see a bunch of books and select one to play. For now, it's ugly. Because it's placeholder. Click that rectangle. It's a book.



Book2 text

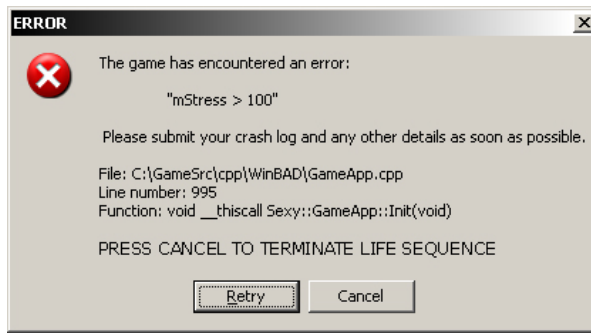
Sometimes too rapid.

The Not Bad: Nifty New Technology.

At various stages of the development process, new technology was created that provided some pretty useful features. As mentioned above, we used Lua extensively from the start. Mid-way through we developed the PAM animation tool, and towards the end, we finally added a packed file format that not only hid the game's assets, but also boosted disk access performance as well. The downside to all this? Bookworm Adventures was the guinea pig for all this new technology. That meant dealing with frequent problems which could detract from other pressing tasks, like making the game. It also meant a certain amount of pervasive fear as the game became larger and more complex: was all this technology going to be stable, or were we going to find lots of weird little issues once it was deployed on thousands of different hardware and software configurations? We introduced a lot of newness, which meant that our QA team needed a substantially longer testing period to verify stability in addition to our normal Framework stress testing.

The Ugly: Nifty New Technology?

All this new technology wasn't without its pitfalls. Lua was a constant source of extreme anger and frustration. The Lua debugger was highly experimental and didn't become stable until several months before launch. This meant frequent cases of the debugger reporting invalid results, crashing, and just generally exhibiting irritating and untrustworthy behavior. In addition, some pretty hefty changes were made to the interpreter mindstream. The most dangerous one was changing Lua's array indexing to be 0-based instead of 1-based (in an attempt to stem the confusion from constantly switching between Lua and C++ code). Up until 6 months before launch the game still had sporadic off-by-1 errors that caused untold problems and instability. The interpreter would also sometimes do unexpected things that took huge swaths of time to track down. Furthermore, since the syntax of the language changed as we added/modified features, the game's coding style frequently mixed original Lua syntax with C++ syntax. But we weren't about to throw out tens of thousands of lines of source code just to beautify things.



Must...not...go...insane...

Speaking of source code, the game was immense. The game wound up being over 150,000 lines of code spread out over 400 files (and no, that doesn't include any of the dialogue. That alone was 10,000 lines). That sometimes made management difficult. In the early days of the game, once we moved out of the prototyping phase to a more stable phase, we rewrote the code base several times to try to design a scalable architecture that wouldn't turn into a horrific mess. That was obviously quite time consuming.

Once the game was sufficiently tested, we started implementing our DRM system, which was also undergoing a rewrite to be more stable and more expandable. This caused its own set of hassles as it was hard to debug the DRM process and testing was time-consuming. Add to that the fact that we wanted to test out 4 different marketing modes with various features to find the best converting one, and complexity crept in at a crucial development time.

As we approached the end of the game's development time, the impending doom of Windows Vista appeared on the horizon. We decided to make the game, the PopCap Framework, and the DRM Vista compliant. All with a beta operating system and hardware that didn't have any drivers. And yet, it was manageable enough to be handled by a single programmer.

Happy Ending:

Thirty months, 150,000 lines of code and 4,500 images later, Bookworm Adventures was realized. In the process we gained +1 wisdom, increased our stress resistance value, and got our nerd on in a totally casual way. We are all extremely proud of Lex and BWA, and are especially pleased by its reception and success. In the first 45 days of its release Bookworm Adventures made more revenue than any other PopCap title in the same amount of time (Yes, including Bejeweled 2!). With those numbers and other more intangible measures of success, we think we can safely say that our "Good" choices — and even the "Not Bad" ones — have more than made up for our "Ugly" ones.

Relevant Linkage:

If you're interested in Bookworm Adventures, would like to try it out yourself, or have any questions regarding the PopCap Framework, feel free to follow the links below.

Bookworm Adventures: <http://popcap.com/launchpage.php?theGame=bwa>
PopCap Framework Development site:
<http://developer.popcap.com>

Breaking Ranks: Hacking Tournaments, Ladders, Contests, and Multi-Player Games

Steven B. Davis, CEO, SecurePlay (<http://www.secureplay.com/>)

You've made your game, designed it carefully... considered cheaters, hackers, avoided exploits and engine problems, and yet once your game goes "live" online everything falls to pieces.

Welcome to the world of Game Service attacks.

Cheaters and hackers are increasingly attacking the "game around the game" – not the game itself, but the other features of the online service: tournament rankings, ladders, reputation systems, contests, and anything and everything that is part of an online game service.

This paper will discuss a number of these attacks and suggest some countermeasures. The paper will not review traditional computer security weaknesses, but rather the problems specific to being an online game service – weaknesses that can occur even if you have done everything right with encryption, database protection, access control, and firewalls. Unfortunately, there are few standard solutions to these problems as the security weaknesses often are closely tied to a specific game service's business and operational model.

Lobby Attacks

Before players enter a game, they use a lobby service to set up matches – either with opponents of their choosing, or, for tournaments, based on algorithms and procedures provided by the game service.

Tournament and Lobby Spiking – While randomized match ups are theoretically strong, it is an interesting question whether teammates, or opponents for that matter, could collude to enter the matchmaking lobby within a narrow time window and thus increase substantially their chance of being matched together. After all, if ranked games are being run continuously, there are going to be inevitable times when the ranked game lobby is going to be relatively empty. Or, even with a relatively popular game, highly synchronized lobby entry can overwhelm the randomization process. The larger the team, the more effective this tactic will be. A weighting system that adds an anti-correlation component (to ensure that players haven't played together before) and a measure that considers how many games someone has played (to address disposable identities, see below) to the tournament score could help discount the effect of team play. Another strategy may be to allow players to play multiple games at once (this strategy works better with thoughtful, as opposed to "twitchy" games).

Boosting - Once players have been able to match up with whom they wish, either because of an open lobby, a ladder type system, or spiking, they can then "boost" the rank of a designated player or group of players. It is probably a good idea to model your tournament structure against a boosting strategy to see how many entries it would require to be effective. The other goal may be to place better in a less competitive tournament – becoming the "best of the worst" in a junior or amateur tournament rather than having to fight and likely not win in a more seasoned competition.

Tournament & Ladder Game Play Attacks

There are certain attacks that can occur against a game because it is being played in a tournament or as part of a ladder ranking system. These are not really attacks on the game but on its context. While Brain Age may be a great single player game, it would fail utterly in an online

competitive environment. This problem also exists for many puzzles and trivia games. Puzzles may be able to be solved algorithmically and trivia games are vulnerable to **Dictionary Attacks** and

“Swarming” to find the right answer. Finally, there are a number of games with optimal or strong strategies which only are a problem in competitive play.

Collusion – players cooperating together when it is forbidden by game rules. This is a problem for multi-player games in general, but can be even more problematic when tournaments or rankings are involved. For example, collusion in a two-player game is meaningless unless there is a multi-player ranking system that can be attacked.

Bots – software that automates or aids play is a widespread problem in computer games. These applications are not cheating the game itself, but are attacking the multi-player game experience. Bots that uniquely target tournament or multi-player play are **LossBots** that can boost the rank of their sponsoring player. There are also **WinBots** that use optimal or strong strategies and **BoostBots** that do not play the game, but act as player aids.

Game Service Provider Problems

Game service providers don't like to think of themselves as a source of problems, but players certainly do. The most important asset a game provider has is his reputation. In order to avoid damaging PR, game service providers should be prepared for accusations from disgruntled players. Also, the industry should cooperate to establish solid standards and practices since the problems of a single company could snowball to damage the industry as a whole.

Payment Abuse / Rake Abuse – If the game service has payments involved, there are opportunities for payment abuse. A game company that shaves a nickel here, a penny there, and a quarter somewhere else can easily and stealthily earn undeserved revenues. Game providers should provide clear payment tables that are always available to players and full and detailed accounting records for the player's review. It would be optimal to provide an independent audit on the player's platform, but this is not always practical. An outside auditing firm in support of well-documented processes and procedures and other measures can help build a reservoir of trust.

Bias – Because of the nature of the games that they are offering, game providers often have insider knowledge that would give a favored player a real advantage in a game. Also, if there are games played vs. the provider, there can be tax advantages to reducing apparent winnings by colluding to lose to a cooperating player.

Action Hands – In games where variable payments are involved, the game provider may be able to make the game more “interesting” and hence increase payments. In this case, the game provider doesn't really care who wins – just that there is more activity.

The “Game Over” Game

One part of the game code is the responsibility of both the game service provider and the game developer – the Game Over game code. Networked games can end for a number of legitimate reasons, but also for illegitimate ones. Dropped connections and computer failures are too common to be ignored and punished. Game developers and providers also need to be concerned about players abandoning a game to avoid a loss (**Stat Guarding**). However, malicious players can abuse the “game over” logic, and even the game abandonment code to their advantage. For example, depending on how the **“Game Over”** logic is implemented, a malicious player may be able to force the game to end when they have an advantage or to use their preferred score as the authoritative source for the game. Players may abuse the **Game Abandonment System** to make it look like the other player has abandoned the game – and trigger the

game score system to punish them accordingly. The ideal approach is to periodically build a "certified game state" that can be used to replay or finish the game at a later date.

Players can also attempt to **report false scores** and **delay reporting** of undesirable game results to manipulate the ladder or tournament system. A final problem is when players make **side wagers** on game results. This is not something that a game operator can handle directly, but it is an issue that they should be aware of. Unfortunately, there are no magic bullets for this problem – games need to be examined on an individual basis.

Identity Problems

While identity is not really necessary for a high score service (see Security High Score Games - http://www.igda.org/casual/quarterly/2_1/index.php?id=3), it becomes much more important once one moves to a richer online game service. The simplicity of developing a casual game and hosting free games becomes substantially more complicated. Once the identity system is undermined, the remainder of online game service is compromised, as seen in the previous sections.

Invalid Licenses/IDs – Both paid and unpaid games often use a license key or platform ID as part of their identification system. For performance, storage, and business reasons, these keys are sometimes not issued and validated individually, but by an algorithm. Malicious players can steal keys, duplicate them, or break or duplicate the authentication algorithm. There are ways to ensure the security of license keys and recover from compromises – but the techniques are very game service specific.

"Alt" IDs – Free online game services often permit, or do nothing to stop, the creation of multiple identities. Second Life has had particular problems with multiple accounts since it allowed them. Players can use these additional identities to increase their chances of winning or boost their rank with **Lossbots**. Positive incentives can be used to encourage honest registration of identity such as prizes or awards programs.

Outsourcing – Players sometimes recruit or hire other players who are good at a game to play for them to boost their score. This is offered as a service, like gold farming, for several massively multi-player online games like World of Warcraft, but it has also been reported for ladder systems for casual games. There is not much that can be done about this.

Game Save Sharing – Some games save files or other persistent information to be stored locally. This data may be exchanged with other players to boost statistics or otherwise enhance play. This has occurred with the Xbox 360 to boost Achievements in the Xbox Live service. If such files need to be supported, they can be cryptographically tied to a specific platform or user account.

Conclusion

Game service providers are moving to provide richer player experiences to complement their games. These richer game play systems, such as tournaments, ladders, and reputation systems, bind the players to the service and keep them playing. Players do not have to attack the individual games to undermine these value-added services. They can attack the entire game service fabric.

The short history of online gaming has shown that players are willing to hack and cheat just to get the high score on a small web site for a simple Flash game. Player reputations and rankings increase the rewards for successfully exploiting these services. Fortunately, the malicious player's ego trip usually results in the rapid, usually gleeful, disclosure of the attack. The game changes once again as game service providers support real rewards and prizes – hackers and cheaters will stop sharing and publicizing their exploits – putting a much greater burden on the game service

Slingo Quest Postmortem

By Dave Walls, founder and president, Funkitron, Inc

Letter from the
Editor

Bookworm
Adventures
Postmortem: The
Good, The Not Bad &
The Ugly

Breaking Ranks:
Hacking
Tournaments,
Ladders, Contests,
and Multi-Player
Games

Slingo Quest
Postmortem

Q&A with Richard
Wienburg, Softwrap
Inc.

Improving
Communication with
Your Sound Designer

Top 10 Data Review
for 4th Quarter of
2006

Windows Vista and
Casual Games
Development: What
developers need to
know

Download a PDF Version
of this issue.

Subscribe to the IGDA
Casual Games SIG
Mailing List

Archives:
Fall 2006
Winter 2005
Summer 2005

Developer: Funkitron, Inc.
Publisher: Funkitron, Inc.
Release Date: Nov 2006
Platform: PC
Engine: Guru (Funkitron's Proprietary Casual Games Engine)
Dev Time: 10 months
Team Size (full time): 4
Team Size (total): 8
Budget: \$100,000
Number of Slingos Achieved During Development: 10,000,000



About 4 years ago, Funkitron released Slingo Deluxe, a download version of the hit online game Slingo developed by Slingo, Inc. For those who don't know, Slingo is a game that combines slots and bingo, has been around for 10 years, and was one of the first games to connect with what is now termed the "Casual Game Audience".

Slingo Deluxe surprised all of us by doing extremely well (it was released back in the time when downloadable games based on a free online game still seemed a bit crazy), and last year we decided it was time for sequel. After many countless hours of brainstorming, we came up with the amazingly fantastic name: "Slingo 2"!

But then regrouped, put our heads together a second time, and "Slingo Quest" was born.

The goal for the product seemed simple at first, but turned out to be fairly complicated.

Slingo Deluxe took the online version and simply made it a downloadable game. But with Slingo Quest, could we take the Slingo brand and make it into a true casual style game? Could we update the game, but still maintain the classic play? Could we expand the audience, but not lose the core players? Could we get people to play Slingo on four cards at once!?

We hoped we could find the answers to these questions as we headed off into the development of Slingo Quest.

The Evolution Of The Concept

Initially, we really did just start off with a "Slingo 2" style concept. Simply update the game's graphics and then add some other games to it, like 5 Card Slingo, and ship it. But, it just didn't feel right or quite enough for a new product. We really wanted the game to have some depth to it so it would be consistent with the current game design of casual games today and really be something a Slingo Fan would find new and challenging.

But how to take a beautifully small game like Slingo and make it bigger?

Slingo Meets Funkiball

In our previous game, Funkiball, we did an interesting level progression where new features were added and combined in various ways throughout each level. So every new level became almost a brand new game to itself as you played along. Plus, we had a star system in Funkiball whereby each

of the 5 stars represented a different game play goal. The challenge was to replay each level to see if you get all 5 stars in each one. Could this concept that worked so well in a brick busting game be applied to Slingo game play?

Slingo Boss Levels???

We did have some ideas floating around for different types of boards. They usually began with what I call "wouldn't it be cool questions" For Slingo they were: "Wouldn't it be cool if you could play 4 cards at once like they do in bingo parlors?" and "Wouldn't it be cool if you could play a big board, like 7x7?" Thus Slingo Multiboard and Slingo 7x7 where created.

Slingo Inc. also had this very cool game mode on their web site called "Pirate's Fortune". In it, you played against another player and took turns playing each spin. "Wouldn't it be cool if you got to play the Pirate in this game?" Pirate's Fortune joined the mix.

Additionally, we did have another game mode which sounded great: "Wouldn't it be cool if you could play Slingo in a Collapse like manner" – but in practice just didn't seem to work. We may revisit this one later on and see if we can make it live up to its promise.

So we had these interesting new Slingo game modes, and when we applied them to the structure we had in Funkiball, they became the final levels in each section... "Boss Levels"! We were very happy to be able to apply the Boss premise to a simple game like Slingo.

I See Matches...

So we had a level structure, with boss levels. But what do we put in them? How do we make the progression interesting?

This is when we put on our powerup thinking caps. We looked at what was hard in the game, for example getting Devils, getting a big score, finding matches... and created some powerups to help deal with these things. So Devils became blocked by Devil Protection, getting big scores was aided by Multipliers and finding matches was helped by my favorite powerup, Slingo Vision, which allows you to see all the matches you can make on the card. This one is very helpful on the 4 card version of the game.

Slingo and the Star System

We had our levels, and our powerups, now it was time to level design.

Instead of just having the levels add things randomly, we decided to have a progression where each island either featured a new game play element, or, as in later levels, started to combine them in interesting ways like we did in Funkiball. This allowed people to go through the levels, discover new things, and then keep on discovering new things as game play elements started to be combined. One person remarked, this is like having 60 Slingo games in one and this statement turned out to be very true. Every "level" can be played as a game itself, complete with trying to get all the stars and finally trying to get the top score on the top ten lists (there is one for each level).

Wait, Don't Ship, There's More!

The final thing we added was Slingo Stamps which you collect for great achievements. I always liked this kind of feature in games and was glad we were able to squeeze it into Slingo Quest at the last minute. It added that extra layer of depth for true Slingo fans.

Design Is Finished

In developing a game, you would always like to just sit down and write up the design doc that describes the game as a script defines a movie. But, it's usually the case that you have to feel your way around to determine what the game wants to be. It is a very iterative process that involves more than just writing design docs, but actively playing and testing what you are coming with.

With Slingo Quest, it did take us some time to figure out exactly what the game should be, and this was actually done while we were actively developing the game, but luckily,

design and execution came together nicely in the end.

What Went Right

Great Partnership With License Holder

During the development of Slingo Quest, we had a fantastic working relationship with the folks at Slingo Inc. and worked together closely during every step of the development of the game. Being in lock step with them really helped keep the development schedule on track as well as insure that all of the new aspects of the game worked for their brand and audience.

Successful Sequel

We set out to make a game that could offer something more than just a 'new paint job' on the classic game. We wanted to make a product that could offer a brand new challenge to the Slingo fan. So, if you had already played Slingo online a million times and even already owned Slingo Deluxe, Slingo Quest would offer you a new challenge. The new game elements and design we put in added new challenges to the game, and thus helped it become successful as a sequel.

Core Game Play, Expanded

Sometimes when you start fiddling with a game, you can take the brand too far away from its core appeal. I think with Slingo Quest, we were able to expand the game play in a way that, even though there are all these brand new game elements, you can still play the game as you always had. The things that make Slingo what it is (spinning, matching numbers, getting Slingos) are still there but there are now more fun things that surround this core game play.

Quality Graphics and Music

Early on we decided to use an artist that was part of the Slingo Inc. team. It was thought that by having her creating art that the Slingo folks could see immediately, it would cut down on back and forth time. This definitely proved to be the case. But the biggest bonus was that the art she created really was fantastic and brought a truly unique and fun style to the game.

The second factor was music. Our composer really put on his 'happy fun hat' for the game and cranked out some catchy tunes that really stick in your head but don't annoy. This is very hard to do, and I'm amazed that he nailed it rather quickly.

Core Audience and Beyond

Slingo at its core is a fairly simple game, but we wanted to add some factors to the game that would expand the audience a bit. So though the game is still good simple fun, we've added some new game features, like the star system, that adds some additional challenge for people. This has turned out to expand the audience for the game as well as given it the depth that a casual product now needs.

What Went Wrong

Amazingly, not a whole lot went seriously wrong in the development of Slingo. The team really did an excellent job and it was a smooth development, well about as smooth as you can get in the crazy world of game development. But I've dug down and come up with these "What Went Somewhat Wrong" items.

Blind Programmers

Try as they may, some of the programmers just didn't get the magic of Slingo. I imagine this may happen frequently in casual games. Most times in the game industry, you are the

audience for the game you are making, but in the casual space, frequently you are not. This can sometimes prove challenging for a programmer, like writing a romance novel when you don't particularly like romance novels. Thankfully, the programmers worked around this and dealt professionally with feedback like "Trust me, it's fun!" and "If you like it, something must be wrong!"

Game Got Big

The game design, which called for new features, cards and power-ups, spread out over the course of 50 levels and 10 bonus levels, along with Slingo Stamps for achievements and a star system for re-playability quickly took this simple game and made it a fairly big thing. All of these features added up to a great game, but in the course of a month, the game exponentially became something that took more time to test than we planned on. Luckily, we decided to start testing early on, which helped us tweak game play sooner than later. Then, we made use of some of the core Slingo players as part of a beta group to further help us tune things.

Release Pressure

When dealing with a brand like this, there is a pressure to hit the holiday season with it. Though a brand new title may benefit from being away from the holiday rush, with a known brand like Slingo, it really helps to have it be part of the holiday buying season. This put a bit more pressure on the team as we worked towards having the game ready to go in October, which of course turned out to be November. But we were happy to at least have the game out before Thanksgiving.

Moving Target Engine

We are constantly updating our game engine and frequently this happens while a game is in development. Unfortunately, this always causes some problems but luckily nothing that held up the game from releasing. But building a house, while still designing the nails, is always a tricky thing.

Getting Smooth Play

The original Slingo had a delay when you matched items and then made Slingos. The person was forced to wait while the game processed the match and then any Slingos achieved. This created pauses in matching and made things feel clunky. We determined that we wanted play to be smoother and not have these delays. Unfortunately, we started working on this late in the project and thus had to do some major rewriting to get it to work. But in the end, it was worth the extra effort because the player can now make matches quickly and the Slingos will follow behind. The game now feels extremely smooth to play.

Conclusion

Dealing with a game and brand like Slingo is harder than it looks, especially when coming up with a sequel. Add too much, and you've lost the core audience. Add too little, and the game won't be different enough from previous ones and you've lost sales. The trick is to find a way to extend the game so that it builds upon the brand, extends the play, expands the audience and in turn, sells great.

The teams who worked on Slingo Quest, both at Funkitron, Inc. and at Slingo, Inc. were both top notch, and I'm happy to say that they really nailed it.

And now we are all looking forward to making Slingo Quest II, except maybe those 2 programmers who still don't get it... :)

Bio

Dave Walls is founder and president of Funkitron, Inc, one of the first casual game companies.

At Funkitron, Dave has designed such hit games such

Scrabble Deluxe, Scrabble Blast, Boggle Supreme, Slingo Deluxe, Poker Superstars II and the just released Slingo Quest.

Before starting Funkitron, Dave Walls was Creative Director at Hasbro Interactive and oversaw the design of such games as Monopoly II, RISK II, Scrabble II, Game of Life, Boggle, Frogger PSX, Centipede PSX, and NERF Arenablast among others.

Funkitron can be found at www.funkitron.com and has a casual game store at www.blueribbongames.com.

Dave has also started a wiki dedicated to casual games at www.casualgamewiki.net and invites game makers to add their games to it.

Q&A with Richard Wienburg, Softwrap Inc.

Interview by Jason Akel

[Letter from the Editor](#)[Bookworm Adventures Postmortem: The Good, The Not Bad & The Ugly](#)[Breaking Ranks: Hacking Tournaments, Ladders, Contests, and Multi-Player Games](#)[Slingo Quest Postmortem](#)[Q&A with Richard Wienburg, Softwrap Inc.](#)[Improving Communication with Your Sound Designer](#)[Top 10 Data Review for 4th Quarter of 2006](#)[Windows Vista and Casual Games Development: What developers need to know](#)[Download a PDF Version of this issue.](#)[Subscribe to the IGDA Casual Games SIG Mailing List](#)[Archives:](#)
[Fall 2006](#)
[Winter 2005](#)
[Summer 2005](#)

Note: This interview has been edited for clarity and brevity.

Richard Wienburg (richardw@softwrap.com) is the Vice President of Sales and Product Development for Softwrap and has been with the organization for over 5 years. Softwrap headquarters is based in London.
www.softwrap.com

Jason Akel (jasakel@gmail.com) is an independent marketing and business development consultant to the games industry. Future Casual Games Quarterly issues will include interviews with other DRM providers including Exent and Digital River. The Fall 2006 Quarterly offered an interview with Trymedia.

What has Softwrap been up to in the past few years? The casual games industry heard more about you a few years ago but less so recently.

Our focus in the last few years changed. It started out we were pretty much offering everything that customers requested, distribution, DRM, commerce, etc. Operating a distribution network, we decided, not being one of our key skills, takes up a lot of time for very little return. That's where Trymedia and ourselves diverged. Our key skill is the DRM. We empower the publishers to distribute themselves in their network and on their own web site. [Softwrap no longer operates any third party websites.]

In fact, we've taken a step away from being a games-only oriented company. Our focus has been more on software titles, larger IPs where some of the applications are being sold for 20,000 British Pounds [USD ~\$39,000] as opposed to a game being sold for \$20. Most of the competitors in the games space haven't entered that genre. But that said, our product is applicable to both software and games.

Can you expand on the contrast between Softwrap versus Trymedia and Digital River?

Softwrap is not a distribution company but rather a technology company. We do not compare ourselves to others in the distribution field although we have partners to assist with distribution.

Our focus is on the DRM and this is where we set ourselves apart from all competitors. Trymedia's has a DRM solution to offer their clients but their strength is clearly in their Trygames distribution network.

In order to maximize your sales and reduce piracy, my suggestion would be to use Softwrap's DRM to secure a title and still distribute this through the Trygames portals. This can be done by supplying Trymedia with activation codes to give to customers when they make a purchase. In so doing you ensure that your product will not be found floating around hacker sites yet you are still able to utilize their distribution network.

Digital River offers their Passport system which I repeatedly hear moans and groans about when attending seminars on DRM. It seems that the product has to manually be tied to a title and that only the top level versions which are far more expensive are even worthwhile looking at. In the meantime you still need to design your own second level security to make this somewhat secure.

With Softwrap, everything is done automatically. Wrapping can be done by non-technical employees and no additional code added to the product. Securing your product takes just

a few minutes and not day or hours and this is key as resources are expensive.

Can you give us an overview of the game security and ecommerce services Softwrap offers to casual game developers and publishers?

Softwrap's core competence has always been on the security and the licensing features that go with it. This security does not only protect the publisher from illegal use of their title, but also protects any secured files from reverse engineering and therefore protecting their IP one step further.

Our ecommerce is closely linked to the DRM which ensures a seamless purchase and licensing experience. Once a purchase takes place through our ecommerce either from within the game or directly from the website, the title is automatically licensed. There is no need for the user to wait for any licensing details to arrive via email or to manually insert them. This greatly reduces basic levels of support.

What high-level features do you offer such as subscription offering, shopping cart with up-sell capabilities, around/in-game advertising, subaffiliation, etc.?

Softwrap offers a range of marketing and sales tools. Our most basic tool is our coupon system. This allows for discounts on purchases and cross/up-selling opportunities.

We recently launched a full subscription system with a renewable pricing scheme.

Around and in-game advertising is something we have always been able to offer. We work closely with a company called Eyeblaster who specialize in this field who assist in putting the ad-campaigns together and inject them into the game.

We offer extremely successful affiliate tracking tools to ensure that all affiliate sales are correctly attributed to any particular partner. The affiliates are also able to log into the sales reports and view their generated sales in real-time which helps them feel confident that they are accurately receiving their revenue share.

On DRM, how do various clients work with you? Are they using your DRM out-of-the-box? Are they customizing it and how?

Yes, out of the box. No need for additional security as ours has proven to be more than sufficient in the fight against piracy. Unlike our competitors, hacking has not been an area for concern with our product.

Do you have an offline DRM solution? Offline phone activation support?

We've already been doing that for a couple of years. Our technology is already capable of doing offline sales into retail, OEM or download. You don't need to create different builds for different distribution outlets.

We have both web and offline phone activation support. A publisher can choose one or both.

Can you name some of your prominent casual clients and in what ways they work with you?

Funcom, Sega, Bluefish Media, Reflexive... a full list of partners can be found on our website:
<http://softwrap.com/partners.aspx?section=Testimonial>

How are you working to breach the Asian market? What advantages do you offer developers who want to do so?

At present our DRM services are being used successfully in Asia and have been for quite some time without any hacks being reported.

Our solution including ecommerce has been localized in Japanese, Simplified and Traditional Chinese as well as Korean which is a necessity in the Asian market.

With this all in place, we offer all the tools required for publishers to target the Asian market for online distribution and sales.

Do you have or plan to offer a Mac DRM solution?

This is something we have discussed many times, but with the amount of work still being done on our Windows platform, we don't yet feel the need to cross the barrier.

How should a new developer/publisher sign up for your services and negotiate a contract that may not be your boilerplate?

This is very simple. They should go to our website, launch the Softwrap SDK and register an account. There are no upfront fees and we welcome publishers to test out our systems without tying them into a contract of use.

Once this is done, they should contact one of our business development executives to discuss the services which they require and of course negotiate the appropriate fees. Contracts are kept very simple and we feel that publishers who are not happy with our services should have the choice to continue or move on.

Does Softwrap plan to attend Casualty in Amsterdam (February) and Seattle (July)? GDC (March)?

We will be attending Amsterdam and expect to do the same for Seattle and GDC.

Parting thoughts?

My message would be to all those publishers that have not tried our technology, especially within the last 12 months, to give this a chance. Everyone who has actually seen it at work has commented on the ease of use and fantastic additional features we offer free of charge to assist with their marketing efforts.

If you would like to see a demo, please contact richardw@softwrap.com and we can set this up at a conference or via phone.

Improving Communication with Your Sound Designer

By: Kane Minkus, Somatone

Letter from the
Editor

Bookworm
Adventures
Postmortem: The
Good, The Not Bad &
The Ugly

Breaking Ranks:
Hacking
Tournaments,
Ladders, Contests,
and Multi-Player
Games

Slingo Quest
Postmortem

Q&A with Richard
Wienburg, Softwrap
Inc.

Improving
Communication with
Your Sound Designer

Top 10 Data Review
for 4th Quarter of
2006

Windows Vista and
Casual Games
Development: What
developers need to
know

Download a PDF Version
of this issue.

Subscribe to the IGDA
Casual Games SIG
Mailing List

Archives:
Fall 2006
Winter 2005
Summer 2005

There is a reason none of the guys at SomaTone Interactive Audio become professional Jazz musicians. After years and years of studying our instruments, jazz theory, classical theory, Bill Evans, Miles Davis and practicing our brains out, why not become pro jazz guys? The common consensus around our studios: the jazz guys are in the corner always getting ignored, and who wants to be ignored? We want to make some noise! Do you treat audio or your audio designer for your game like a jazz musician? You would not believe how many sound designers - staff sound designers and production houses alike - express this sentiment. Of course, you may be the exception; you actually love audio and think about it early on in your projects. Perhaps you wake up at night thinking about the coolest melody line, musician or new virtual instrument that would match the game perfectly. Or maybe that is just what we sound guys do. Either way, the following article suggests ways to more deeply understand the world of your sound designer and to communicate effectively. This will help you get the results you are after more quickly and ultimately give your players the impression that audio got the attention it deserves.

One of my favorite teachers at Berklee College of Music used to say, "The sum is made up of the parts! Pay attention to the parts and pieces, from the beginning!" So let's explore the parts of audio that come together to make the whole experience. Some of you might be more experienced with working with music/audio, but if you are not familiar with the digital studio of the new millennium, sequencers, virtual instruments and post production/composition lingo, you could still be in the dark when it comes to really communicating in ways that help the sound designer translate your vision into sound.

The term Sound Designer, especially in the downloadable game industry, has somehow become the common term for the audio professional making both the sound and the music. However, in the audio world, a sound designer is someone who is only designing sounds. This means, most of the time, the sound effects person (although occasionally this could refer to someone designing sounds/ textures for a composer - such as Trent Reznor's sound designer who creates sounds/noises that Trent ends up composing with). Often in the *Advertising Industry* the term sound designer means both the sound effects person and the composer - mainly because in lots of commercials, it is hard to distinguish between general sound scapes as sound effects or music. However, in the *game* industry, there is a clear distinction between composers and sound designers. If you applied for a job at Lucas Arts as a sound designer, no one would expect you to come in with a guitar and start writing music.

So, when talking with your "Sound Designer" - if that is their job title in your company - that person is likely thinking about sound effects and music separately in both the creation and integration - with distinctive language, tools and approaches to both.

-

Tools of the Trade

If you have not been in a commercial sound design/composer studio recently you will be shocked. Long gone are the days of huge console mixing boards and big recording rooms. Most of those studios/facilities have gone out of business (except the ones in LA that are locked into the big film houses or record labels - and even many of them struggle along these days). The technology is so good these days, that you only need two computers (and some folks can just get away with one), a bunch of software and maybe a great microphone. From this setup one can

compose a full orchestra or make a movie sound like a movie (including the score and sound effects). Usually you will see a Mac & a PC (or just one or the other), a small keyboard controller, a nice desk, a set of good pro speakers (one of the most important parts of the studio – *don't forget the sub woofer!*), some sort of acoustical treatment to make the room “mix” ready (maybe a little tracking room to record live instruments/VO), and a refrigerator to keep the audio specialist up all night for your projects and it's crazy deadlines. Yes, there are some guys who love to have more – a real piano, a nice tracking room, a cushy client coach, et cetera, but those are really unnecessary frills to get the job done in casual games. These studios/work stations are commonly referred to as DAW's – Digital Audio Work Stations.

One main tool your sound designer and composer will use is a sequencer or audio design software. The industry standard is ProTools and Logic. Logic only runs on a Mac and it is a general consensus that you want to be running ProTools on a Mac as well. This means your audio specialist is likely to be a Mac user. Be sensitive to that as we Mac users realize that everyone else in the business world uses a PC, especially since we create audio for games that are mostly PC based. Obviously your audio specialist must also know the PC well. Some audio professionals choose to work solely on a PC (especially when they specialize in game audio) because games are mainly created on and for PC users or PC platforms. If they are mainly PC users, they may use a program called Cubase. You don't need to know the details of the programs, but this way you are familiar with their tools. There are some additional specialty programs - like we use Peak to master all our audio at SomaTone (some use Sound Forge on the PC – we will discuss mastering later), Ableton Live for loops and Virtual Instrument hosting (or Acid on the PC – we will discuss virtual instruments later) and Reason is a popular program for some audio professionals.

Steps to Creating Audio

Understanding the steps in audio production will help you isolate where a problem is coming from so that you can help correct it:

For Sound Design:

1. Pre-Production
2. Recording Sounds
3. Editing & Sound Sculpting
4. Mixing
5. Delivery

For Composers:

1. Pre-Production
2. Writing
3. Recording
4. Editing/Mixing
5. Mastering

Although some of these steps have the same name, many are very different for the composer and the sound designer.

Sound Designer

Pre Production

This is where the creative conversations take place (we will talk about effective creative conversations in the next article) and reference material is collected. This is a great time to reference another film, game or project that contains sound design that you like or dislike. This is also the time to discuss audio integration and audio engine considerations.

Recording Sounds

The sound designer might need to, or choose to, record original sounds for a palette they will use later. This is sometimes labeled Foley Sessions; sound effects sessions that are related to materials being used to create sounds – footsteps, bags being hit together, rustling of clothing, smashing of things, anything you can think of can be recorded here. Another option is to purchase a massive library of pre-recorded sounds. Be aware that the sounds in a purchased library are rarely finished. A good sound designer will layer sounds, sculpt sounds and use all sorts of audio/editing tricks to get an appropriate sound for a game. At SomaTone, although we have an enormous library of digitized pre-recorded sounds, we still find ourselves recording new material constantly to layer it over existing material.

Editing & Sound Sculpting

This is where the sound design starts to come together. The sound designer will begin sculpting the sound, through editing features on his sequencer. They might use effects (like reverb, delay, flangers, EQ), use fades/crossfades, layer many sounds on top of each other, take a pre-recorded sound and run it through a synthesizer to effect it, and so on.

For example, a sound for a growl for an alien character in a game, might be a combination of a dog growl, a metal door slamming together, stones rubbing together and a harsh synth tone. This is where the creativity of the sound designer really starts to make a difference in the quality of the sounds. There are many tricks that can be employed to arrive at just the right sound. It is pretty much an “anything goes” approach, but a good sound designer has a sense of what should be layered in (both sounds and effects) to make a sound effective, clear and impactful. Good sound designers also know ahead of time what they are looking to hear and how they are going to roughly get there – this eliminates endless tweaking.

Mixing

The sound designer will then mix all the elements of the sound audio, both individually and separately, so that each sound effect has the appropriate levels relative to all the others. These are simply volume (or referred to as Gain) adjustments.

Delivery

Once the sound effects are mixed, the delivery of them should be kept at these levels and they need to be individually “bounced out”, or put into a final form such as .wav, mp3, or OGG format (rather than the sequencer’s format). This makes it easier for the programmer to integrate them and maintain a dynamic sound in the game (instead of “normalizing” all the sound effects – a process of making each sound as loud as it can be). If the sound designer is actually creating environment loops, they might choose to keep the “panning” (sound location in the 3-D space), or deliver a full looping file with all elements integrated. A “Two-track mix” means the final mix of the sound effects in a two-track form; this is your basic stereo track with a Left and Right channel – like a regular musical CD. The two-track mix is usually referred to when the sound effects track is continuous - versus the individual elements being delivered. Another common term is “Sound Effects Stem” – this comes from the film industry where you have just the sound effects separately instead of having them mixed in with the music or dialogue. Sound Effects Stem and Two Track are interchangeable if you are only talking about the sound effects, because some two track mixes can have the music and dialogue “married” or mixed together.

Composers

Pre-Production

Again, this is where the creative conversations take place and reference material is collected. Integration and audio engine considerations should be discussed here as well. In the film world, you would have a “spotting session” here,

which means that the director and composer would get together and go through the whole film to pick out “cue points” – or places where the music should be synced to the film edits/emotional content. This can be done for game as well – even though games are not linear, they can have some linear elements in the early discussions.

Writing

The composer will go to work writing early sketches of the music. Every composer will compose slightly differently, so I will not attempt to layer one process over everyone, but from a macro scale, the composer will have at least two phases – 1) rough ideas or ideas in progress and 2) final compositions (“pre-mixed” or not mixed). Be aware that any composers are concerned about giving rough unfinished compositions to producers for fear that they will not be able to hear the final musical vision from the rough sketch.

Recording

Once rough ideas are started (or if the composer likes to start immediately writing in their sequencer), recording the parts can begin.

The writing/recording process will often start with melodies, chords or drum beats. If a score is going to be really textural or out of the norm, the writing process might start with a concept – such as banging on a piano to create a horror feel. Then the composer begins to layer instruments in the sequencer on different “tracks”. To explain what a “track” is, imagine a racetrack where several cars are all driving. However, each car is restricted to staying within their own lane. This is how a track in the recording world works. You can have one track for guitar, then another track for bass, another track for vocals, and another track for drums. When you play all the tracks back simultaneously, you get a fully orchestrated song. This is why it makes it possible for one composer to play and record all these different instruments separately and then play them back together. Tracks in the DAW are effectively limitless. So you can have 80 tracks if you have 80 different instruments in the orchestra all playing at the same time and recorded on a separate track.

Once upon a time, we had *real* instruments, but now we have samplers that trigger recordings of those instruments instead. And if you have amassed a good sample library of instruments, you can have incredibly realistic sounding instruments from every corner of the world at your fingertips. Sprinkle in some good “programming” (this is what it is called when a composer plays an instrument, like a flute, on a keyboard) and you can create incredibly realistic sounding orchestras, world compositions and beautiful scores. If you are still a skeptic of the quality of sampled instruments go check out “Ivory” from Synthology or Symphonic Choirs from East West. These samples will knock your socks off. If your composer is using cheap or bad samples this will show up by the piece sounding too “midi”, “fake”, “synthesized” or cheap. Feel free to request better samples; there are incredible samples out there for every instrument on Earth. Since composers should be constantly updating their sample libraries, or spending time creating them through recordings, this should be a major reason why you chose to work with a composer or not. You will know right away by listening to the horns, pianos, strings, guitars, etc, if your composer is using high quality samples or not. The instruments should sound like they are played by live players since most good sample libraries are recordings of live players now a days.

As the composer layers each track with new instruments, harmonies, melodies, etc, they are creating their final vision (or trying to find one, depending on how they like to write). We believe a composer should have a clearly defined vision of what they are going for and should be able to clearly explain it, before they even start writing anything. This means you should be able to have meaningful conversations with your composer about where they are headed at anytime during the creative process. By meaningful, I mean you should understand in non-musical terms what the result will be – we will look at this later as it is a communication process in itself. If a composer has lost the vision, it is likely they are lost in general in their composition.

Some producers ask for initial sketches of music to make sure the music is headed in the right direction. A word of caution about this: often musicians are concerned that their finished work will not be able to be envisioned, by someone who isn't them, or isn't a composer.

Let me give you a personal example. When I compose, one of the mix engineers might stop in my studio to hear how things are going. I often get a funny look like "what are you doing man?", and then they leave confused. However, once I get all the parts I envision into the songs, they will come back in and totally get it, with a sigh of relief. With complex orchestral orchestrations, for example, often the composition won't sound like much until all the parts are playing together. You and your composer will have to work out the right time to listen to early sketches. Just remember that you cannot always hear the composer's internal masterpiece in initial sketches. But again, the composer should be able to clearly explain to you where they are going with the sketch.

Editing/Mixing

Once the basic instruments and parts are all put into the song, the composer will move into an editing/mixing stage. Cutting and pasting music like words in a word document, the composer will begin to edit parts together or use editing tricks for a specific effect (often used in electronic music these days, such as stutter or vocal edits).

The mix will also begin coming together here. The mix is comprised of volume levels of all the instruments, the panning (3-D placement of an instrument in the sonic space), reverbs, delays, EQ, flangers & special effects.

The mix is one of the most important and often overlooked components of a song. The mix is where all the parts start to get shaped and molded together into a cohesive piece of music. Mixing involves a musical and technical understanding of compressors, reverbs, delays, frequency response, EQ's, special effects, the big picture, dynamics, automation, de-essers, limiters, expanders, and so on. Most problems, mistakes and issues are created, discovered and fixed in the mix. More often than not, a composer is not a mix engineer and will admit that their biggest weakness is in mixing their music. At our studio, for example, we have a dedicated mix engineer mixing everything that goes out the door. The difference between what comes in from our composers and goes back out to the client after going through a proper mix process can be like night and day.

Without getting too technical, let's look at a few very important parts of the mix so that you can give your composer more specific feedback.

Volume

One basic part of the mix is volume – some instruments will need to simply be made quieter or louder. However, don't forget that your composer has the ability to "ride" the volume, or create dynamic changes in the music. The strings don't need to sit static at the same volume through a passage –either loud or soft. Expect and demand from your composer that the mix sound *dynamic*. This is one of the most forgotten parts of music from young composers. Keep the music and parts moving and creating tension and release through the volume.

Pan

Pan is often overlooked as well. This is where an instrument is placed in the stereo field. You will be amazed how much space opens up in music when instruments are properly panned in the sonic space. Imagine a jazz trio all standing in the same place trying to play. It would sound "crowded", because it is! If the music/mix sounds crowded, or too much, all happening at once, it could be improper panning. Panning can also help separate frequencies that are all competing for space. If a mix is too muddy, try requesting that the panning be addressed a little more.

EQ

This is a secret weapon among mix engineers. A good EQ job can make a world of difference. Make sure that different instruments are not competing for the same frequency space. Make sure that instruments containing lots of mids and highs (in the frequency spectrum), such as guitar and piano, are "rolled off" or have no, or little low end competing with the bass. If your mix sounds muddy, or too bright/harsh, this is the area you want to recommend that your composer address.

Effects

Reverb and Delay are two common effects in music. Reverb gives you the spatial relationship that the instrument has to the room it is in (reverb is actually just a long delay repeated very closely). If you want a dry violin sample to sound like it is in a cathedral, crank up the reverb (or use a little delay). Of course, too much is described as sounding "wet". If you feel the mix needs more dimensionality perhaps individual instruments need more reverb (or delay, depending on the type of music). Or if the mix feels too loose and sloppy sounding, it might need less reverb to tighten up the cohesion of the instruments working together.

Compressors

Compressors work to stabilize the sound in the sonic space, by making it not have as much of a dynamic range. In this instance, sometimes you want to reduce the dynamic range to cause the sound to feel more controlled. If a sound is too wild, or too dynamic, using a compressor will help reel it in.

Mastering

Does your composer hand in fully mastered recordings? They should! Mastering is a process of taking the final mixed two-track (or music composition) and running it through one more stage of audio processing. This stage of audio processing will usually contain: multi-band compression (to restrain the dynamic edges of the mixed tracks a bit), limiting (squeezing the music to the loudest point before distortion – this also removes some of the dynamic range) and perhaps a bit of EQ to adjust any levels for the whole track. This stage is what gives it that "radio quality".

When you sit in your car and tweak the knobs on the EQ, you are mastering (or affecting the master recording) the song in your car (of course the song on the radio has already been mastered professionally!). You will notice that each time you mix and then master (or use a compression stage) you are limiting the dynamics of the piece. In order to properly get a professional sound and keep the dynamics of the piece (so it doesn't just sound flat), the mixer/mastering engineer must really know what they are doing. If your composer is not handing in mastered tracks, you will know it because it will sound weak when played against a reference piece of music in the same genre. Also, it will be quieter than other pieces of music played at the same volume on your stereo or computer.

Next Time

This gives you lots of tools to have effective technical conversations with your composer and sound designer. Stay tuned for part two of this article, in which we will explore effective ways to discuss creative content design - you have a vision for your project and need to turn it into audio. I will discuss the use of Laban Movement Analysis (Effort, Space and Shape Modules), creative industry terms for sound designers and composers, how to effectively give feedback, common communication mistakes between producers and composers & how to clearly discuss an abstract idea – words that translate to sounds. Until then!

Author Bio: Kane Minkus

Kane Minkus is one of the founding partners of SomaTone Interactive Audio, a leading music production firm for the Film, Television, Video Game, Online, Advertising and Mobile Industries located in San Francisco. A leader in the audio

community, the SomaTone Interactive Audio team has become known as a cutting edge source for music scoring, sound EFX, mixing, mastering and voice over production. Gaming credits include Medal of Honor (EA), And 1 (Ubisoft), Jurassic Park (Dream Works), Diner Dash 2 (PlayFirst), Mystery Case Files 1 & 2 (Big Fish Games), and Bang! Howdy (Three Rings Design).

[Get Involved](#) | [Contact Us](#)

© 2007 IGDA

This website is maintained by the volunteers of the IGDA's Casual Games Special Interest Group. For information about the Casual Games SIG, go to www.igda.org/casual



Top 10 Data Review for 4th Quarter of 2006

By James C. Smith, Prodecer, Reflexive Entertainment

Letter from the Editor

Bookworm Adventures
Postmortem: The Good, The Not Bad & The Ugly

Breaking Ranks: Hacking Tournaments, Ladders, Contests, and Multi-Player Games

Slingo Quest Postmortem

Q&A with Richard Wienburg, Softwrap Inc.

Improving Communication with Your Sound Designer

Top 10 Data Review for 4th Quarter of 2006

Windows Vista and Casual Games Development: What developers need to know

Download a PDF Version of this issue.

Subscribe to the IGDA Casual Games SIG Mailing List

Archives:
Fall 2006
Winter 2005
Summer 2005

In the 4th quarter of 2006 we saw four play mechanics dominate the top 10 lists of all the casual games portals. Seek and Find games like MCF Prime Suspects and Hidden Expedition: Titanic were the most popular type of game. Restaurant action sims like Cake Mania™ and Diner Dash 2 were tied for 2nd place with slower paced sims like Virtual Villagers and Fish Tycoon. Finally, match 3 games like Luxor 2 and Bejeweled 2 Deluxe are still somewhat popular.

It is possible to estimate which are the most popular downloadable games by keeping track of which games have appeared in the top 10 lists most often on many casual games portals. In each issue of the IGDA Casual Games Quarterly we summarize the top 10 lists for the quarter. Games are ranked on each portal using a simple scoring system that awards points for each day a game is in a site's top 10 list. Ranking #1 in the list is worth more points than #10. We make no attempt to summarize the whole market across all ports but instead list what is popular on each portal.

These three charts list the estimated 20 best selling games from three popular casual games portals. In other words, they summarize the 92 different daily top 10 lists made public by these portals during that quarter.

Real Arcade 4th Quarter 2006				
Rank	Game	Score	Days In Top 10	Peak
1	Luxor 2	448	70	1
2	Scrabble®	322	77	4
3	Paparazzi	273	35	1
4	Mystery Solitaire™ - Secret Island	259	35	1
5	Diamond Detective	246	33	2
6	Slingo® Quest	219	26	1
7	Mahjong Fortuna 2 Deluxe	210	35	2
8	Pat Sajak's Trivia Gems	209	34	2
9	Risk II	206	41	3
10	Ancient Tripeaks II	195	28	1
11	Diner Dash®: Flo on the Go	190	19	1
12	SPY Fantasy	182	28	3
13	MCF: Prime Suspects	164	17	1
14	Word Spiral	161	28	3
15	Abundante!	154	28	3
16	LEGO Chic Boutique	147	28	2
17	Pantheon	133	21	2
18	Jurassic Realm	121	17	2
19	Aztec Ball	113	20	4
20	Family Feud™	112	38	5

For example, Scrabble is estimated to be the 2nd most popular downloadable game on Real Arcade during this quarter because it appeared in the top 10 list published on RealArcade for 77 of the 92 days in the quarter but it never ranked higher than #4 on any days top 10 list. Luxor 2 earned as estimated overall rank of #1 for the quarter because it accumulated a score of 448 by staying very high in the top 10 lists for most of the 70 days that it appeared in the list.

MSN 4th Quarter 2006				
Rank	Game	Score	Days In Top 10	Peak
1	MCF: Prime Suspects	871	92	1
2	Virtual Villagers	568	65	1
3	Big Kahuna Reef 2	387	60	4
4	Tropix	386	50	2
5	Bejeweled 2 Deluxe	303	42	3

6	Travelogue 360: Paris	270	27	1
7	Kudos	219	42	5
8	Diner Dash 2™	190	24	2
9	Mahjongg Artifacts	105	17	4
10	Star Defender 3	87	16	5
11	Galapago	86	13	3
12	Bespelled	84	14	5
13	Rack em Up Road Trip	72	9	3
14	Magic Match	70	14	6
15	Cake Mania™	63	9	4
16	Bookworm Adventures Deluxe	60	6	1
17	Inspiration	60	10	5
18	Jewel Quest® Solitaire	56	7	3
19	Mahjong Escape™ - Ancient China	50	10	6
20	SCRABBLE® Blast!	48	6	3

BigFish 4th Quarter 2006				
Rank	Game	Score	Days In Top 10	Peak
1	Hidden Expedition: Titanic	717	92	1
2	MCF: Prime Suspects	637	92	1
3	Travelogue 360: Paris	426	59	1
4	Luxor 2	317	53	2
5	Virtual Villagers	306	60	3
6	Slingo® Quest	244	38	2
7	Westward™	222	31	1
8	Diner Dash®: Flo on the Go	213	26	1
9	Mystery Solitaire	190	36	2
10	MCF: Huntsville	174	44	5
11	MCF: Ravenhearst	167	17	1
12	Carrie the Caregiver	166	21	2
13	Nancy Drew - Danger by Design	165	35	3
14	Jewel Quest® Solitaire	139	25	4
15	Atlantis Sky Patrol	125	35	4
16	Believe in Santa™	124	24	1
17	Nancy Drew: Secret of the Old Clock	118	25	2
18	Cute Knight	97	24	4
19	Flower Shop™: Big City Break	69	8	2
20	Cash Cow	57	25	6

This next grid is not as easy to read at first glance, but if you take a moment to study it, you will be able to see how each game ranked in 10 popular casual games portals. For example this grid shows that, for the time period sampled here, Luxor 2 is estimated to be the 4th best selling game on BigFishGames.com, the 5th best selling game on GameHouse.com, Pogo's 9th best selling game, and the #1 best selling in Real Arcade. This grid also shows us that Luxor 2 was not a top 30 seller on MSN, Oberon, or Reflexive during the time period sampled here.

Game's Estimated Rank for 4th Quarter 2006										
Game	Big Fish	Game House	MSN	Oberon	Pogo	Real	Reflexive	Shock wave	Try Media	Yahoo
Bejeweled 2 Deluxe			5		17				5	6
Big Kahuna Reef 2			3		21		7		11	
Cake Mania™		4	15	8	4		6	3	6	1
Carrie the Caregiver	12							2		
Delicious Deluxe		11				25				5
Diamond Detective		6				5				
Diner Dash 2™			8		11		13	8	3	2
Diner Dash®: Flo on the Go	8				8	11	14	7	7	20
Fish Tycoon					6		4	1	20	12
Gold Miner: Vegas									1	
Hidden Expedition: Titanic	1			27	2		3		28	13
Kudos	27		7		15					11

Luxor 2		4	5			9	1			5	2	3
Magic Match					3							
Mah Jong Medley			3									
MCF: Huntsville	10	17			1	5			10	16	23	
MCF: Prime Suspects	2			1	13	7	13		2		17	4
Mystery Solitaire™ - Secret Island			8		4	27	4		9			
Paparazzi							3					
Risk II							9		5			
Saints & Sinners Bowling					5							
Scrabble®			1				2					28
Slingo® Quest	6					13	6		8	9		25
Super Collapse! 3			2									
The Poppit! Show						1						
Travelogue 360: Paris	3			6	12	12			17			21
Treasures of the Deep					2							
Tropix			28	4							15	
Virtual Villagers	5	20	2		6	3	24		1	6		15
Westward™	7				15	18			20	4	19	9
Wheel of Fortune											4	

All of these rankings are based on public data collected from the portals web sites and archived on www.game-sales-charts.com. More detailed information can be found using the [Top Games](#) chart on GameSalesCharts. This will allow you to see additional portals, more than just the top 20 games, and more data about each game such as the first and last time it appeared in a portal's top 10 list. The [Day by Day](#) chart will also make it easier to visualize trends over time on any given casual games portal.

This article lists data from 10 web sites and details 3 of them. The GameSalesCharts web site tracks the downloadable games top 10 lists of 20 different casual games web sites. WildGames and Boonty were recently added. New sites are added any time they are suggested.

James C. Smith is one of the founders of Reflexive Entertainment. Tracking the history of the top 10 lists is a part time hobby he started back in 2002. In 2005 James launched www.game-sales-charts.com to make interactive queries of his top 10 database accessible to everyone. You can reach James at James@game-sales-charts.com

Windows Vista and Casual Games Development: What developers need to know.

Compiled by:
Robert Carroll, TableStar Games, IGDA Casual Games SIG
Kim Pallister, Developer Relations Manager, Microsoft Casual Games

A Q&A with the Vista Developers

Compiled by:
Robert Carroll, TableStar Games, IGDA Casual Games SIG
Kim Pallister, Developer Relations Manager, Microsoft Casual Games

In the months leading up to the public launch of Vista, a number of questions have been raised regarding how the platform will affect the casual games industry. In an effort to help answer the most frequent asked questions from the casual games community, we've asked Microsoft and the Vista team to address a number of the questions that people have asked over the last couple of months.

We would like to thank Kim Pallister and the Vista team for taking the time to address these questions.

Vista & Casual Games

Q: What is new in Windows Vista that has to do with games?

A: Gaming was considered a core usage scenario throughout the design of Windows Vista. One of the results is that Windows Vista adds games to the types of media that end users access and manage (similarly to how images, music and video were added in Windows XP). This means that games get their own prominent place on the start menu, and that the OS has some context with which to treat games. This is accomplished primarily via three mechanisms, the Game Definition File (or GDF), an application called the Games Explorer, and support for Rich Saved Games.

Other areas of Windows Vista that are relevant to games include the Parental Controls, usage of Limited User Accounts, Game Controller Support, and inclusion of Media Center and Tablet PC support.

Q: What are the benefits?

A: Supporting the Game Definition File mechanism allows developers to better expose their games to consumers, define context-relevant actions the user can take with the game, and to better build brands around their products by building awareness for developers and publishers bringing games to market. Use of Limited User Accounts allows for increased PC stability. Rich save games allow consumers to better navigate their games experience. Parental Controls give parents tools to control their children's gaming experience on their PCs. Support for the Xbox360 game controller allows for a more consistent controller experience between console and PC.

Q: What are the areas I should pay attention to as a developer?

A: There are four areas developers should pay attention to, in order to ensure their games don't run into problems: Limited User Accounts, Proper Installation and setup, support for Game Definition Files, and Parental Controls.

Installation and Setup

Q: What are the Implications of Limited User Accounts?

A: Every account on the system runs most processes as

"Standard User" (formerly called "Restricted User" on Windows XP). Administrator accounts still prompt the user when running applications at admin level.

Q: Currently many games write config files or save game data into the Program Files area on the computer. In Vista this information should be virtualized and default users will not be able to write to these areas. Depending on what data is being written, users can lose save games or in some cases will not be able to run at all as a non-elevated user after the initial install. What can be done to prevent these problems?

A: One of the most common application compatibility bugs with Standard User (aka Limited User) accounts is an application saving data into its protected install directory or setting registry keys in the protected local machine hive. Windows Vista provides an automatic fix for many of these scenarios called virtualization. With virtualization active, any writes to a protected file directory or registry key are automatically redirected to a per-user location. This makes it appear to the application that the write was successful and it can read that data back, but it is not actually present in the protected location. While this allows many applications to successfully run that would otherwise fail, it does have a number of limits and support implications:

- Since the written files do not actually exist in that location, users won't be able to find them there. There is a "Compatibility Files" button on the Windows Vista File Explorer Window that will help, but it can still cause some confusion.
- Writes to code binary files (EXEs, DLLs, etc.) are not redirected by virtualization as this would be an obvious security vulnerability, so in-place 'patchers' will fail in the absence of other changes or specific application compatibility fixes.
- Since all redirection is per-user, files or registry keys written by one user will not be visible to another.
- The compatibility files are only visible to application EXEs running in the same directory.
- Virtualization is an application compatibility technology not a general solution, so it is disabled in some situations.

In short, virtualization will allow games and applications to save and load data files in otherwise protected locations, but can be unintuitive to the user, developer, and customer support to understand why it works in some situations, but not in others. We do not recommend developers depend on virtualization for their applications, although it is very useful to ensure many legacy applications continue to work with User Account Control enabled.

See the DirectX SDK technical article "User Account Control for Game Developers" (<http://msdn2.microsoft.com/en-us/library/bb206295.aspx>) for more detail.

Q: How will programs that provide games, but aren't games themselves, (e.g. Game Browsers, Game Downloaders, etc) be classified? If a user downloads and installs such a program, will it join the regular applications or the Game Explorer?

A: Only if the developers of such applications choose to define themselves as games and the application declares itself as a game during installation.

Parental Controls

Q: What are Parental Controls, and how do they apply to games?

A: There is a Parental Controls system in Windows Vista that gives users a tool to administer their children's accounts on the PC as it pertains to games, application usage, and Internet access. For games, the tool allows parents to control access to games based on game rating, or if they choose specific details. Parents can also choose to restrict or allow access to specific games, or to restrict gaming during specified times of day.

Q: Will Parental Controls restrict access to my game? How often will this happen?

A: Parental Controls provides parents with a tool to use, if they choose. It is not on by default. In order for it to affect a game due to ratings, three things must happen: First, the parent must set up a separate, limited user account for their child, Second, they must choose to switch on parental controls for that account. Third, they must enable the use of Game Ratings (off by default even when parental controls is switched on). Fourth, in order to be restricted, the game either needs to have a ratings board rating higher than that allowed for the account, or, in the case of an unrated game, the parent needs to have specified that they wish to block unrated games (unrated games are allowed by default).

Q: In what instances will an unrated game be hidden from users?

A: Only when parents have enabled parental controls, and then specified in the settings to block unrated games, will it then block games in the games explorer that are unrated.

Q: What rating services are recognized by Parental Controls?

A: ESRB, CERO, OFLC, PEGI, BBFC, and USK are all supported. The choice of rating system is based up the region that is set during install, and can be changed via the control panel.

Q: How are RP (Rating Pending) games handled? Are all games that are still in testing considered to be unrated?

A: While the rating is pending, they do not have a rating and thus are treated as such.

Q: In what instances do the Parental Controls come into effect? Do all games automatically get brought into the Game Explorer or do developers have to add themselves? Will Vista attempt to classify incoming programs as "games"?

Parental controls for games only affect applications recognized as games by Windows Vista. However, parents can choose to block any windows application via parental controls. Applications get recognized by Windows Vista as games in one of two ways: If they are installed as such, or if they are one of a library of legacy games that Windows recognizes from a legacy database.

Q: Can downloaded games be installed in places other than the Games Explorer (e.g. start menu, icon on desktop, etc). What happens to games on the start menu if blocked by parental controls.

A: Game Developers are open to using the platform as any application developer would. We encourage people to use the Games Explorer, as we are trying to encourage the use of a cleaner aesthetic, but it's the developer's choice.

Games blocked by parental controls will be blocked at the file system level regardless of whether started from the start menu, games explorer, or via clicking on the executable.

Q: Some developers have attempted to support Vista parental controls, but Microsoft provides no API's for a game or game manager to provide their own UI access to parental controls or provide an alternative UI to the game explorer, so a game cannot "adopt" the system and interpret parental controls. Could UI access to the parental controls be made available to developers?

A: Some limited programmatic interaction with the parental controls is possible. More information is available in the Parental Controls Programmers Reference found on MSDN here:

<http://msdn2.microsoft.com/en-us/library/ms711890.aspx>

General Information

Q: *Are there any other issues that developers should be aware of when preparing to make their games work with Windows Vista?*

A: There are a number of items specifically applicable to the casual games business (downloadable games in particular) that are worth noting:

Ratings: Those developers choosing to get their titles rated should be aware that the ESRB is aware of the issues facing casual game developers (cost sensitivity, install & distribution often controlled by a 3rd party, etc). We have been told by the ESRB that they are taking these into account and will address them in the coming months.

Game Definition File (GDF) metadata: Making Windows Vista aware of the information contained in the GDF is something that happens during installation of the game. Since in the casual game space, installation is often handled by technology provided by the distributor or portal, developers should discuss the procedures by which this will occur with those parties.

In the case of Microsoft Casual Games, the installation of games is handled via a DRM wrapper and installer solution provided by our distribution partners. In the case of downloadable games, this is Oberon Media, and in the case of our GameSpring subscription offering, Real Networks. Please contact those parties for questions on their installation technology, or email mcg@microsoft.com for more information.

One area within the GDF likely to cause concern will be URL's around developer and publisher names. Many portals will likely express concern around providing URLs to developer or publisher sites, especially when those sites often sell games, thus competing with these same portals. Any portal or distributor may have it's own policy around how this will be handled. In the case of MSN Games:

- We will not over-ride Developer or Publisher name with our own. We feel strongly that attribution for the title belongs to those that created the game.
- We will not generally ship games with URLs to Developer or Publisher sites. We want to offer a consistent customer experience, and we don't want to send our customers to competing sites.
- MSN Games will instead substitute URLs to MSN Games, with parameters to call up a list of that developer's or publisher's games on the MSN Games site. We are interested in bringing attention to more of your games, on our site.

Learning More

Q: *Some in the Casual games development community feel that the GDC developer day is targeted at the core gaming industry, 360 developers, and large CD-ROM developers, and would not contain much content for casual game developers. Are there any plans for the Microsoft game development team to engage with the casual developer community at industry events?*

A: Many of the sessions that Microsoft presents at such events are general enough to be applicable to all game developers on Microsoft platforms. In addition to this, Microsoft Casual Games presents developer sessions at most of the major industry events. Sessions are being presented at the upcoming Game Developer Conference, and Microsoft Casual Games is holding a Partner Day event prior to the Casual Connect (formerly 'Casuality') conference in Amsterdam. In addition, Microsoft Casual Games will have a developer-focused website in the near future at <http://www.microsoftcasualgames.com/> . In the meantime, general inquiries can be mailed to mcg@microsoft.com.

Q: *Where can developers find out more information about game-impacting technology in Windows Vista?*

A: The best place to start is the latest DirectX SDK, which can be obtained from <http://msdn.microsoft.com/directx/>. Many of the DirectX SDK technical articles address these and

other topics, and can be found on MSDN at <http://msdn2.microsoft.com/en-us/library/bb173039.aspx>.

Developers not familiar with the technical details behind support of Windows Vista should see the relevant documentation on MSDN, found here:

<http://msdn2.microsoft.com/en-us/library/bb173039.aspx>

In particular, "Windows Game Explorer for Game Developers", "Gaming with Least-Privileged User Accounts" are a good starting point.

Another good starting point is the GDC 2006 Presentation entitled "Getting Ready for Microsoft Windows Vista, available here:

<http://msdn.microsoft.com/directx/presentations/>

Interviewer Bio: Rob Carroll

Rob Carroll is the Producer for TableStar Games out of Berkeley, Ca. working on the HeroCard suite of games, which blur the line between CCG's, board games and on-line play. Before joining the team at TableStar, Rob was a designer with Critical Mass interactive, and a producer and game designer for the massively multiplayer on-line game Glympse that was in development with Meta 4 interactive. Before moving to San Francisco to join the team at TableStar, Rob was on the Board of Directors for the International Game Developers Association, San Diego chapter and a member of the Program Advisory Committee for the Game Development curriculum at the Art Institute of San Diego.