

WIZARD 101

THE IDEA WAS SIMPLE: TAKE A SMALL TEAM AND QUICKLY CREATE AN MMO THAT WOULD APPEAL TO KIDS AND FAMILIES WHO LOVE WORLD OF WARCRAFT, COLLECTIBLE CARD GAMES (CCGS) LIKE POKEMON AND YUGIOH, AND CERTAIN POPULAR WIZARDS. J. TODD COLEMAN, JAMES NANCE, AND JOSEF HALL'S VISION OF THE GAME AND ITS LOVABLE CHARACTERS WAS THE RIGHT IDEA AT THE RIGHT TIME: THEY IDENTIFIED AN UNDERSERVED MARKET, AND DEVELOPED A GAME FOR IT.

Our core team was made up of the usual suspects—promising aspirants and industry veterans from Origin, Midway, and beyond. Josef Hall (senior director of software engineering), Todd Coleman (creative director), and I were founders of Wolfpack Studios, and still proudly wear the scars from SHADOWBANE.

At the time of writing this article, we're almost one year after launch and the response has been phenomenal. A fun game with interesting differences to what's currently available has made for a compelling combo, but it's still gratifying to hear from parents and kids who love the game and educational to hear from those who don't. I've personally learned a lot from this project, and I'm pleased to have the opportunity to share some of that.

WHAT WENT RIGHT

1) RIGHT IDEA AT THE RIGHT TIME. At the time we started the project, Disney's TOONTOWN was the only massively multiplayer online game on the market that

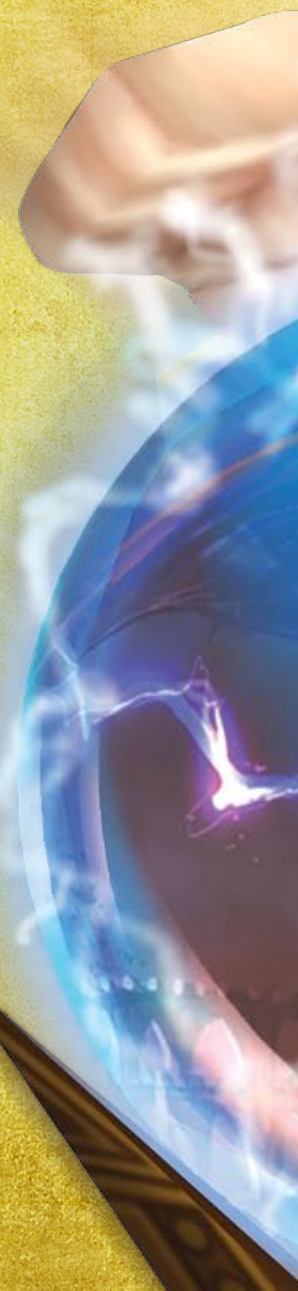
catered to the kids' demographic. Our goal was to address this audience with a product that had more depth.

The market was ripe for a new game, and Todd Coleman's idea of wizardry and CCGs based in a fantasy world was a wonderful framework for a game. Josef Hall, proud parent, made the connection that the kids' and tween market was wide open.

It's always fun to work on an original IP, and we honed the vision through countless brainstorming sessions. The decision was also made to make the game very story driven, with an emphasis on the player being the central character and hero in an epic magical adventure.

Players enter the game as new students recruited by the headmaster to combat a deadly magical threat to all of creation. The idea quickly grew beyond the scope of a school for wizards, and encompasses many different magical worlds. The concept of our different schools of magic mapped perfectly onto types of cards for the CCG.

CONTINUED ON PAGE 24





CONTINUED FROM PAGE 22

We tried to keep the story simple and easy to understand by incorporating a fun cast of characters with an old-school hero's journey. We also wanted to make a game that would appeal to families; Pixar's ability to appeal to parents and kids alike with movies like *Cars*, *Toy Story*, and *The Incredibles* was our shining example.

2) SCOPE, SCOPE, SCOPE. We had a team of fewer than fifty developers at our peak, and an aggressive schedule. We wanted to deliver a top-quality game in under three years. The veterans among us were skeptical, but determined. The only way it worked was to keep the game design tightly scoped; we knew feature creep would kill the project.

We had some leeway in that our target market was fairly free of competitors, so we were able to initially cut features that, in another market, we wouldn't be able to ship without. Anything that wasn't deemed critical to the core game experience was deferred or cut. Guilds, crafting, mounts, player housing, auction houses, grouping, player-versus-player combat—all these things could wait until after launch. We've added most of them in the year since the game opened, but we never would have shipped on schedule if we'd tried to do it all.

Our internal milestones were built around core features taken from the overall game design. We'd identify one or two major features to finish and polish, estimate them, and let those estimates determine the milestone duration. Then we'd fill out the milestone task list with smaller features as time

and resources allowed. Repeating this process in bursts of roughly eight to twelve weeks allowed us to focus on a few features at a time. We drew only from the master design, which kept the scope of the game from growing too much over time.

We didn't launch with as much content as we'd have liked, but you never do. We elected to ship with four major adventure areas and quickly added a fifth area three months after launch. This was a small enough amount of content to allow us to manage the work, but still enough to provide several hundred hours of game play. We recognize that content is vital in MMOs, but you still have to launch the game!

3) PROTOTYPE AND ITERATIVE DESIGN. The idea of a turn-based MMO collectible card game for kids was a bit risky, to say the least. We knew that the card game combat was our core unit of gameplay, so we had to get it right.

Our initial prototype of the combat system consisted of hand-drawn cards (art courtesy of game visionary Todd Coleman), some ten-sided dice, and colored glass beads (for power points and health). We spent hours playing the game against each other (there were no monsters initially), changing card values as we went with a quick erase and pencil scratch iterative approach.

The second prototype was on the computer, with a client and independent server—a multiplayer version with 2D cards and data stored in tables for easy iterations and balancing. Limited A.I. for computer controlled opponents

came later, and served as the basis for our full monster A.I. system.

The critical part of this early work was to see if the basic core gameplay was fun, and to refine the combat rules. Those rules evolved into our current combat resolver. Prototyping was critical to our later success; locking down core gameplay early allowed us to focus on other elements of the game instead of going through multiple project restarts we couldn't afford.


This iterative approach to development was applied to all new systems, though not to the same degree. Each time a new system was brought online, we'd get it functional as quickly as possible and try it out. Feedback was gathered from anyone and everyone in the company, and incorporated.

As the game's development progressed, we also took the opportunity to focus test. Art direction, pricing model, story elements, characters, combat—almost everything was put in front of kids and parents at some point during production. We listened to the customer, and reaped the benefits.

4) DIGITAL DOWNLOAD AND FREE TRIAL. There was great debate about whether to go retail or direct download, adopt a free-to-play model or give the standard free 30 days. Those of us with shipped MMO game experience were more comfortable with a traditional approach, but our company founder Elie Akillian maintained that digital download was the best way to get our game into the hands of the casual masses. He was right.



GAME DATA



PUBLISHER KingsIsle

DEVELOPER KingsIsle

NUMBER OF FULL-TIME DEVELOPERS Approximately 40 at peak

NUMBER OF CONTRACTORS Roughly 30 including QA

LENGTH OF DEVELOPMENT Six months pre-production, 36 months of full development

RELEASE DATE September 2, 2008

SOFTWARE Microsoft Developer's Studio, G++, 3D Studio MAX, Adobe Photoshop

TECHNOLOGY Gamebryo, Miles, Open Dynamics Engine

PLATFORM PC/Online



As a new, independent studio, we didn't have the pull of a big studio that is able to demand shelf space and end caps. Our game was fun, but no one had heard of us. The obvious answer was to let the game sell itself, and the best way to do that was to let people try it for free.

Going with direct download had many challenges, however. Even now, we're constantly concerned with download size, since it's a major barrier to getting into the game. Each game update is scrutinized and pared down so that we aren't increasing the download to a new user.

One of our better features is our ability to stream the game to the user. This was a huge technical win for us, and basically means that we can deliver game content to the player just before they need it. We have a small initial download that allows the player to create a new character. While character creation is taking place, the game is downloading the tutorial—while the player is in the tutorial, we are downloading the starting area. Although most players will never notice, it means they don't have to incur a giant download to start playing the game.

Finally, our server architecture needed to be scalable and robust. A free-to-play or free trial game with millions of players coming through needs to be able to handle the load without turning away potential players!

Digital download is a hard road, but considering the millions of players who have given WIZARD101 a try, it was the right choice.

5) MIN SPEC AND SMART TECH. Who still has a GeForce2 in their rig? Who actually uses the integrated video chipset that comes with the motherboard? Who still has less than a gig of RAM? Millions and millions of casual users and kids on hand-me-down machines, that's who.

We did exhaustive research early in the project to try and determine what min spec would allow our target market to play the game, and it was pretty scary. Our research indicated a much higher min spec than what we chose, because all data at the time came from gamers, who typically have much more powerful machines than casual users. We took a gamble and went with a much lower min spec, and it really paid off.

We set a tight budget on polygons and texture sizes for every piece of art in the game, and created our areas to support a fixed number of players so that we could limit the load on the graphics card. Clever use of portalling and other tricks in world building allowed us to hide high-polygon pieces of art, and restrict how many concurrent combats a player could see on screen (another big poly hit).

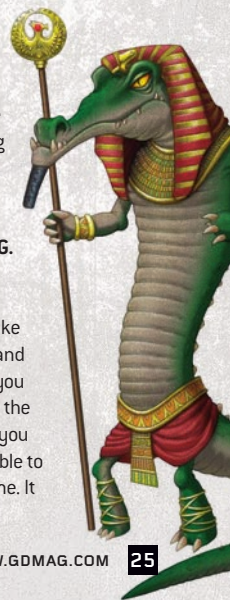
The programming team was very careful about how much data is kept in memory, and spent a lot of time optimizing the code to use

minimal RAM and processing power. Additionally, whenever driver problems arose during compatibility testing we wrote code workarounds so that our casual users would not be faced with the daunting prospect of having to update drivers to play the game. We also knew most casual users and kids would be on older operating systems with outdated service packs and drivers, so we went out of our way to support all that.

Finally, we made the decision to go with a very stylized look to the game. The art style is funny, approachable, and casual, but more importantly, the game looks good on low end machines and will age well, since we aren't competing in the realm of hyper-realistic, bleeding edge graphics.

WHAT WENT WRONG

1) MODULAR WORLD BUILDING. With a small team and aggressive schedule, we made a decision early on that we should take a very modular approach to world and level creation. The idea was that if you use generic building blocks and let the level builders snap them together, you can get a lot of re-use and will be able to create more content for less art time. It



didn't really work—custom areas are better and take just about as much time to create.

We came up with a list of “snappable” pieces (L-shape, T-intersection, end caps, boss rooms, etc.) and had the artists create them for each world with texture variations and decorate them with props appropriate to the different areas. The world designers knew the size of each piece and would create vast adventure area maps on grid paper while the art was being created. In this way, we hoped the designers and artists would be able to work concurrently on the tasks. After the art was created, the designers would snap the pieces together, export them, and the artists would go back for a decoration, polish, and lighting pass. Even though the individual pieces of art were excellent, the end result was fairly generic levels that all looked the same and were boring. The more we re-used pieces, the worse the problem became.

The solution seemed obvious—we would create custom pieces that we could drop in among the generic pieces to provide points of reference to the player in the area and break things up. Examples were gardens with statues, hedge mazes, camps with pavilions, and the like. That really didn't work that well either. Even though the custom pieces looked great, we weren't able to create enough of them to make a difference.

Another approach we tried was to make adjacent areas appear very different by changing the decoration and textures between areas. For example, on Wizard City (our starting world) we themed the adventures areas by element (nature,

fire, ice, etc.) and added icicles, snow, pools of lava, and burning trees to the different zones. That helped some, but it served to hide the problem rather than solve it. The areas still felt very much the same.

For our latest world, Grizzleheim, we finally made it work. We took a totally custom approach to level building: each area was individually conceived and designed, then hand crafted by an artist. The result was a much-improved visual appeal, and all the areas combined took about as long to make as it takes to create a set of snappable pieces.



2) WE CHANGED THE BUSINESS MODEL CLOSE TO LAUNCH.

Naturally, getting people to give us money for the game was key to our longevity and success as a company, and so the business model was a hotly debated topic during early production. We finally settled on a subscription model that was family-friendly and had a good price point. Fairly close to launch, however, we re-opened the subject for discussion and decided to take a more hybrid approach—we'd allow for both subscribers and micropayment customers. At the same time, we also decided to allow users to play the first part of the game for free. By adding a free trial, we increased the number of players the architecture had to support by an order of magnitude. It's a testament to the scalability of what our engineers built that it was even possible that late in development.

We've seen promising results from catering to users that want to pay us in different ways, but because we chose to offer micropayments fairly

late in the development cycle our implementation was less than ideal. For example, rather than having a micro-payment shop available to the users at the touch of a button, we had to use an automated in-game character as our micro-payment shopkeeper. Players have to find him in-game to be able to make micro-transactions. Additionally, the types and variety of items available for micropayments are limited and not altogether compelling.

Another challenge to using the hybrid approach has been the fine line we have to walk with our users; we want to entice our subscribers to make microtransactions, but we don't want to make them feel like they are getting less value for their monthly payments or being forced to use microtransactions. The approach we've taken is that for every item available in the game for a micropayment, that item is also available in the game by other means—for gold, as a rare monster drop, or as a PvP reward. By doing this we have an answer to our subscribers' concerns about value, but it makes for a lot more data work, is error-prone, and can create game balance issues.

If we had the chance to do it all over again, we would pursue a hybrid business model earlier in development. That way we could have created a much smoother experience and more compelling micro-transaction offerings to the users.

3) USER INTERFACE MISTAKES.

To be frank, our graphic user interface is kind of a mess. The GUI is in many ways the face of the game, and supposed to be the user's best friend. It's one of the pieces of the game that speaks to the overall quality of the product, and ours isn't great.

We failed, at the start, to come up with our user interface language—a bible of rules that should make your GUI elegant and intuitive, if followed. As a result, our GUI is often clunky, crowded, and inconsistent. Sometimes buttons are round with icons; sometimes they are square with words. Sometimes we navigate menus with side tabs; other times it's with circular icons at the top of the page. It is critical that you think through how your players will use the interface, and iterate and polish it until it shines—we didn't do that. Although some of our HUDs (deck configuration, for example) did go through dozens of revisions, without a set of established guidelines the result was inconsistent and unpolished.

On the technical side, for reasons beyond our control, we were forced to make the decision to build our own graphical interface system and have struggled with it ever since. We used a homegrown tool for interface layout that was difficult to use and hard to learn. This means that designers and artists had trouble making (and fixing) the GUI, so it fell to programmers to implement HUDs and fix bugs. As a result, programmers spent more time than they should have fixing interface issues, and



our screens lack the visual polish an artist would have provided.

Additionally, our interface elements don't scale and resize with the game's screen resolution. We support 800x600 at the low end, so you can imagine that the HUDs become so small they are almost unusable at high resolutions.

Lastly, our user interface screens are all static; we don't have the capability to animate them, so they seem to lack polish compared to other games made with Flash-based GUIs.

We're currently in the process of migrating our system and all interface elements to Flash, and will soon share the level of quality many of our competitors display.

4) STATS AND METRICS PROBLEMS. The collection, representation and mining of data related to player activities can provide the developer with the keys to tweaking their product to perfection. The trick is to collect enough of the right data, and to make that data accessible to the right audience. If you fail in any of these respects you're in for some headaches. We have headaches.

We weren't sure which facts were going to be important in understanding the success of our game and how we'd need to slice and dice those facts in order to make decisions. We also could have done a better job of making sure the metrics supported the different groups within our company. For example, marketing and operations may both be interested in unique logins, but may require different dimensions—demographically by week for marketing, peak activity hours for operations. We still struggle with asking the right questions and getting the right answers to the people who need them.

Our plan for growth underestimated the amount of data we would need to gather and the number of reports we'd need to run. The activities of millions of players add up quickly. We had some issues scaling our data warehouse with the increasing data set, and had to scramble to keep up. Beyond larger and faster disks, we needed a reporting, retention, and aggregation strategy that would keep our data warehouse manageable after a year of data and billions of facts.

It's not that we weren't warned—our unfortunate data expert told us we needed to make smart choices, but in the heat of making a fun game, we didn't listen. A year after launch, we have a mountain of data, and are having to work very hard to be able to parse through it all to see valuable trends and statistics.



5) POOR PLAYER GROWTH STRATEGY. This was a rookie mistake, and we should have known better. MMOs grow over time, and have a lifespan of five to ten years. A smart designer would plan for plenty of room to grow the game and grow the characters along with it. We, however, chose to box ourselves in and make it difficult.

The first basic problem is that we chose to use a percent scale for many of our equipment and advancement modifiers. Accuracy, damage increase, damage resistance, and other attributes lie along on a scale of 1–100 percent (some with caps lower than 100 percent). This means that we have a hard ceiling on how much power we can award the player through the course of the game.

Here's an example: players use power points (pips) to cast spells in combat, and more powerful spells require more pips. Players gain a pip at the beginning of each turn in combat. As players earn power and equipment, they gain the chance to get a double pip. Here's the sad part—the double pip chance is on a scale of 1–100 percent, and we launched the game with players able to achieve near 90 percent. Given that 100 percent is the max, we have very little room to make more powerful equipment or grow the player beyond what they could achieve when we launched. Additionally the equipment upgrades we designed ended up not being very compelling. With only 100 points to grant, we have upgrades that go from 2 percent Fire Resistance to 3 percent—not very exciting.

The second problem is that we didn't give characters very many attributes. We thought that because we were making a collectible card game, we wanted the majority of player power to come from collecting cards and building decks. However, when you only have a few attributes on the character, you don't have many ways to create valuable

equipment, so your loot and advancement options become very limited.

The last major problem related to player advancement was that we didn't launch the game with any true boss fights. Really, the only way we had to make fights more difficult was to increase the health of the monsters, which just resulted in longer fights. Additionally, we added some scripted boss fights after launch, and there was a huge backlash from our player community.

Now that we're well past launch and it's quickly becoming time to increase the power scale, we're faced with some difficult challenges. The prospect of building and testing a new equipment and character development scheme is daunting, not to mention re-balancing thousands of pieces of gear. The anticipated community response alone is enough to make me cringe.

CLASS DISMISSED

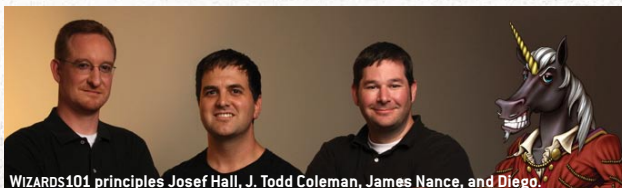
» Some game projects are sprints, some are marathons. An MMO game project feel like sprinting a marathon. We learned just how much you can accomplish with a small, talented team. We learned there is no substitute for good planning, and that polish happens all the time, not just at the end.

The best thing about an MMO is that it doesn't go away after launch, so we can correct some of the mistakes we've made along the way and apply what we've learned in making the pre-launch product to the live product.

By anyone's standards, WIZARD101 is a phenomenal success, and it's absolutely the best project I've ever worked on. There are a few things I'd do differently, and some good lessons learned, but overall it was an immense pleasure to work on such a great game. 🍷

JAMES NANCE is the senior producer for WIZARD101. His career started in 1991 when Nance joined Origin Systems as a QA tester. He was the lead designer on SHADOWBANE and an executive producer at Wolfpack Studios prior to joining KingsIsle. Email him at jnance@gdmag.com.

PHOTO BY TOM HALL



WIZARDS101 principles Josef Hall, J. Todd Coleman, James Nance, and Diego