

Postmortem: Sarbakan's Lazy Raiders



By Ann Gauthier

Settled in inspiring Quebec City, Sarbakan is a fast-growing game developer with over 600 games under its belt. After 10 years of white label servicing and online gaming, leaping into console gaming was as thrilling as it was scary. Sure, we had 10 years of experience in game and IP development, but that experience was generally limited to broad audience, casual online games.

We had but one DS title under our belt, and a budget one at that. What we were doing can only be described as diving into uncharted territory with nothing but a dream. We were working with a new engine (3DVIA's Virtools), a new IP, and a new platform. We knew there would be production delays, that the risk level was high, and that the flow of the project would be hard to predict. Nonetheless, we saw this game as an opportunity to make a statement to the industry: to prove that we could make a high-quality product.

The process of pitching the game to publishers was slow and laborious, but we finally found one -- only to lose it halfway through the project.

When this happened, instead of crying over spilled milk, we rolled up our sleeves and decided to publish ourselves in partnership with Microsoft. With Microsoft's help and expertise, *Lazy Raiders* (then still called by its original title, *Dig it Up*) took on a new twist.



What Went Right

1. The Pitch

For an independent developer, the pitch process is probably the most important. In 12 years, we forged strong partnerships with major web publishers, but none that were directly involved with console publishing.

Our first DS game, *Wordmaster*, was actually an advanced prototype that was closer to a Gold Candidate, and we sold it as-is a few weeks after presenting it to a publisher. This led to signing more deals in the same year, deals that would pave our way into the console market. Our modest DS experience served to prove that we could indeed deliver quality products within allotted deadlines.

This game gave us the confidence to pitch *Lazy Raiders* [[YouTube trailer](#)] at a major game convention in 2008. At that time, we hadn't decided on a platform, so it could easily have been for DS, Wii, XBLA, or cross-platform.

And if there's one thing we learned from our internal pitch process, it's that words and images don't speak loud enough to explain a concept like *Lazy Raiders*. That's why we built a playable prototype. On paper, *Lazy Raiders* looked interesting, but hard to grasp. With the prototype, people went "Ooh! Now I get it!"

The prototype included three complete mazes and some core mechanics, and the visuals were advanced enough to give a good overall idea of what we were shooting for. Before booking meetings with a couple of publishers, our sales

and marketing team mastered the prototype and were thoroughly briefed on the whole game concept.

The publishers' reaction to the prototype was exceptional, and even generated a small buzz around the game at the Casual Connect. When we left, we had netted a publisher, we were keeping the rights to the IP, and *Lazy Raiders* was headed for WiiWare and XBLA.



2. Working With Microsoft?

Lazy Raiders was our first experience with Microsoft, and I must say it was a surprisingly pleasant and educative experience. Our wonderful producer at Microsoft, Scott, provided us with support and counsel.

He pushed us to make the best product possible for their customers, and we asked for more, spurred on by his counsel and encouragement. Scott understands XBLA's audience, both in general and in specific, and helped us hit the bull's-eye with *Lazy Raiders*' target audience.

Microsoft also provided us with all the tools we needed to facilitate the production process, from live meetings to localization and test support. We even got our first usability test for free! That's not even counting the live training we got for services to which we were new, namely the Avatar Awards and Live Party. All in all, working with Microsoft is pretty sweet, and we'd do it again in a flash.

3. Gameplay Experience

Don't you hate it when you play a game and there's so much stuff on-screen that you have no idea what's going on? I know I do. When we finally decided to focus on providing the players with a memorable gameplay experience, the first thing that came up was "we gotta clean things up."

In a game like *Lazy Raiders*, visibility is vital. When the whole maze spins and flips, you have to be able to keep track of where the boulders and dynamite crates are going, which way the walls and rotating rooms are sliding, whether the fire traps are active or not, and, most importantly, where Doctor Diggabone, the Thieves and the Minions are. So we set to make the backgrounds as sober and unobtrusive as possible while maintaining their visual appeal.

We also subdued the music in favor of sound effects. All this voluntary toning down really served to put the focus on the actual maze and enhance the gameplay.

4. Level Design

Early on in the project, our lead programmer hatched a cool level editor. In less than a week, he had it built and debugged. The tool's main objective was to hide the middleware interface so that a non-programmer would be able to create and modify levels without any training.

The editor had the basic functionality to create, move and copy objects. It also allowed the designer to modify object parameters. All changes could be saved and played on the spot to verify the fun factor and the difficulty of the new puzzle.

With this little gem, the level designer could build levels by himself without needing to commandeer a programmer, effectively saving us both time and grief. With it, we created 300 levels, the top 80 of which made their way into the game. It was also the perfect tool to crush those pesky bugs as they reared their ugly heads.

Another tool that simplified our work was RLD (realistic level design), also known as the LOL (log of levels). This dynamic document displays the content of each level and the score values and assigned difficulty rating of each object.

This document serves as a sort of game inventory to validate the entry of each new game mechanic, ensuring a smoother learning curve and creating variety over time. All the data can be consulted in graph form for a better visual understanding of the ramping.

5. Tests, Tests, and More Tests

The internal testing team was included early on in the production process, if only for reasons of proximity. The level designer had been part of the QA team before this project and his desk hadn't yet been moved. This physical proximity, added to the history of teamwork between the level designer and the testers, led to the quick death of countless bugs. They shortly became masters of whack-a-bug.

The further along we were in the production process, the closer loomed Microsoft's certification test. Having by then learned caution, we decided to submit *Lazy Raiders* to pre-certification testing with one of our partners, Enzyme Testing Labs. This allowed us to identify and pinpoint a lot of issues and bugs, most of which were easy to fix.

We were left with only three important bugs, which we were free to address, with time to spare, before final certification. In the end, our bug count was extremely low, and *Lazy Raiders* passed certification hands down on the first try. Our testing foresight came at a price, though; 15 percent of the total project budget. It may seem a hefty price to pay, but it was worth it because it spared us from the extra production costs inherent to a last minute bug hunt.

Not all testing was focused on technicalities. In addition to external usability tests, we put in place a system to test and validate the quality of the game in-house. Staff members were asked to play the game and fill in a questionnaire to communicate appreciation on different aspects of the game. These tests were made on every important build to identify and correct major issues before submitting the game to the publisher.



What Went Wrong

1. Me, Myself, and I

As we were venturing into new territory, we did some research. Okay -- *a lot* of research. We wanted everything to be perfect, so we dug up every little fact we could on downloadable console game development and read through dozens of postmortems to try and learn from others' mistakes. Once we'd compiled all that data, we set down ten golden rules that we were sure would lead us to victory.

Confident in our perfect little bible, we followed these golden rules blindly instead of focusing on what was really important: the fun factor. We concentrated on *how* the game should be done instead of *what* it should be.

Also, smug in our perfect process, we failed to listen to the first wave of comments, defending our IP instead of catering to the players' needs. Thankfully, our Microsoft producer steered us on the right track before it was too late; but we'll get to that later.

You might wonder what were the rules that blinded us. Some were artistic in nature, others touched programming and production. Three of them were about the game design proper:

1. Nothing can move by itself, gravity will do the job.
2. Every trap needs a counterpart.
3. Everything needs to be consistent and logical.

Rule #1 survived the contact with reality, but the two others didn't. Rule #2 made the game so complex that new players were overwhelmed; it had to be broken to provide smoother ramping. Rule #3 was broken to give more emphasis to the flip mechanic, so that we could have such things as minions who undergo transformations during flipping. It's not logical, but it improves the game.

2. Cross-Platform and Target Audience

The game was initially cross-platform and designed to be first produced as a WiiWare title, graphic-wise the lowest common denominator, before being exported to XBLA. We all know that Wii and Xbox users don't fall in the same

target audience bracket, so we set out to make a game that would appeal to everyone.

This was a big mistake, as failing to target a niche -- whether wide or specific -- resulted in design hazards that did nothing but dilute the whole game experience. Eventually, we abandoned the whole WiiWare angle, instead focusing on Xbox Live Arcade.

Trying to find ways to make *Lazy Raiders* appealing to Wii and Xbox users cost us both time and money, and ended up a complete waste. But when we decided to focus on XBLA, things picked up and the project revved into gear, gaining both in depth and polish.

3. Communication is the Key

We wanted the game to be so good on all levels that we tried to throw everything in there. Sarbakan has a history of character and storyline development, and we quickly fell into old habits, becoming mired in what we knew, and were good at, rather than giving the players what they wanted.

So in addition to trying to make a fun and challenging game, we developed an intricate background for Dr. Diggabone. Originally, on his quest for a cushy retirement, the old chap was constantly shadowed by his arch-nemesis, the Glamazon, a greedy rival treasure hunter who wanted to take everything he had, and then some. The game was initially planned to be peppered with still scenes of her worming her way into our oblivious hero's life, posing as his maid, his travel agent, his psychologist, etc. in order to learn his trade secrets and the itinerary to his next expedition.

As a result, we ended up overburdening the game with fluff and jokes (and bad ones at that). The game's menu was a chaotic wreck that unsuccessfully communicated what it was supposed to. And if the first thing the players see when they launch the game is a mess, it bodes ill for the rest.

Saying we fixed the menus is somewhat of an overstatement; it was more of a damage control thing. They could have been even simpler and more to the point. After all, menus don't have to be fun; they just need to be easy to navigate and understand.



4. Manage by Day, Work by Night...

Despite our aforementioned delving and research into the production process of other studios like us, and despite our efforts to avoid falling into the same pitfalls they did, we failed miserably. It's like they say: history tends to repeat itself

Our leads are the best at what they do (otherwise, they wouldn't be leads, right?) When they work on the smaller projects that are our bread and butter, they pitch in and are as productive as the people working under them. But in a more ambitious project like *Lazy Raiders*, they ended up managing the team and not doing what they were best at. This cost us precious resources -- people who could have saved us a lot of production time had their talents been put to good use.

After all, a good lead artist or lead programmer can sometimes do in a day what two juniors do in a week. Had we put our best managerial elements in charge of the team and allowed our leads to do what they do best, and what they love doing, we would have avoided delays, costs, and a lot of grief.

5. Time is Money

The original plan called for two multiplayer modes in *Lazy Raiders*, co-op and versus, that would both have been played split-screen.

We had grand ideas and twists: for example, in the versus mode, each trap you disabled would reappear on your opponent's side. In addition, special power-ups would have provided variety and interesting new ways to attack your opponent. The versus mode would have been highly competitive and fun.

Unfortunately, we were already pressed for time because of other problems -- such as the delays caused by time wasted trying to make a cross-platform game and by our management problems -- and choices had to be made.

After talking with our Microsoft producer, we concluded that delivering one superb game mode was better than delivering three half-baked ones. Multiplayer therefore went out the window to allow us to polish the single player experience. Such is the reality of game developing.

Conclusion

Here we are. Despite losing our initial publisher and dropping from a cross-platform title to a single-console one, we finally launched *Lazy Raiders*. Do we regret any of these things?

Nosiree. Losing our publisher allowed us to come into our own, and gave us the confidence we needed to repeat the experience. Switching console allowed us to focus on the targeted audience and give the game more depth, both gameplay-wise and visually.

All the same... that multiplayer mode we wanted to implement was really, really cool. Who knows? Maybe in the sequel.

[Return to the full version of this article](#)

Copyright © 2015 UBM Tech, All rights reserved