

Postmortem: McMillen and Himsl's *The Binding of Isaac*

By Edmund McMillen

How did a game destined for failure become a cult smash? In an article originally written for [Game Developer magazine](#), Edmund McMillen (Super Meat Boy) discusses how he added religion to *The Legend of Zelda*, mixed it with a roguelike, and came out with a surprise hit.

On paper, there is simply no reason for a game like [The Binding of Isaac](#) to have become as huge as it has. It makes no sense -- and this is coming from the person who believed in it the most. I knew *Isaac* was special, but if you asked me to bet on whether *Isaac* would sell over one million copies in less than a year, I would have bet against it.

You see, *The Binding of Isaac* was made to clash against mainstream games -- it was designed to be a niche hit at best. I had hoped it would gain some minor cult status in small circles, kind of like a midnight movie from the 1970s. From any mainstream marketing perspective, I designed *Isaac* to fail -- and that was my goal from the start.

When I started working on *The Binding of Isaac*, I was still haunted by the end of *Super Meat Boy*'s development, and [the hoops we had to jump through](#) to get there. I wouldn't say *Super Meat Boy* was "selling out," but it was the closest I was going to come to it when it came to playing by the rules to make sure that we could sell the game that consumed two years of our lives (and all of our money).

After *SMB*, I no longer had those worries -- I could afford to take a bigger risk and fail, if I felt like failing. I wanted to make something risky and exciting now that the financial aspects of that risk were gone. And I wanted to really push my limits to get back to where I had come from -- a place where there were no boundaries, where I could create anything without worrying about making a profit.

The Binding of Isaac started in a weeklong game jam. Tommy Refenes (*Super Meat Boy* co-developer) was taking a vacation, so I decided to do the game jam with Florian Himsl, who programmed a few of my previous Flash games (*Triachnid*, *Coil*, and *Cunt*). Florian is the kind of guy who is up for anything; he wasn't worried about his reputation, and was basically down with whatever I wanted to do in terms of content. This was good, because I had two clear goals when I started designing *Isaac*: I wanted to make a roguelike game using the *Legend of Zelda* dungeon structure, and I wanted to make a game about my relationship with religion.

Both goals were challenging but very fun to design, and after seven days we had something that was turning into a game. It seemed too good to pass up, so we continued working on it in Flash (using ActionScript 2). At this point in the process, I wasn't thinking about how we were going to sell this game (or *if* we were going to be able to sell the game at all!); it was just a challenge we both wanted to finish.

We finished *The Binding of Isaac* after about three months of part-time development. We released it on Steam, and it was selling okay; for the first few weeks, the game was averaging about 100-200 copies a day, eventually stabilizing at about 150 a day after a few months. By this point, the game had already exceeded my expectations, but five months after release something very odd happened. Our daily average started to climb. 200 copies per day turned into 500 copies, then 1,000 copies, and by the seven-month mark *Isaac* was averaging sales of more than 1,500 copies a day and climbing. I couldn't explain it -- we hadn't put the game on sale or anything, so I was clueless as to why sales were continuing to grow.

Then I checked out YouTube, and I noticed that fans of the game were uploading Let's Play videos constantly -- over 100 videos every day, each getting tons of traffic. *Isaac* had found its fanbase, and that base was growing larger and larger. Not bad for a game that was meant to fail!





What Went Right

1. Roguelike Design

The roguelike formula is an amazing design plan that isn't used much, mostly because its traditional designs rely on alienatingly complicated user interfaces. Once you crack the roguelike formula, however, it becomes an increasingly beautiful, deep, and everlasting design that allows you to generate a seemingly dynamic experience for players, so that each time they play your game they're getting a totally new adventure.

I wanted to combine the roguelike formula with some kind of real-time experience, like *Spelunky*, but I also wanted to experiment more with the traditional role-playing game aspect of roguelike games *Crawl* and *Diablo*. Fortunately, using the basic *Legend of Zelda* dungeon structure as the game's skeleton made it easy to rework almost all the elements of a traditional roguelike formula (procedurally generated dungeons, permadeath, and so on) into a real-time dungeon crawler format. Almost every aspect of the game seemed to fall perfectly into place with little effort.

Let's start by looking at the *Legend of Zelda* dungeon and resource structure -- it's simple, and really solid. Keys, bombs, coins, and hearts are dropped in various rooms in the dungeon, and the player needs to collect and use these resources to progress through each level. In *Isaac*, these elements were randomly distributed and not required to progress, but I included them to add structure to the experience.

I also pulled a lot from *Zelda*'s "leveling structure," where each dungeon would yield an item as well as a container heart to level up the character and give the player a sense of growth; in *Isaac*, each level contains at least one item, and the player can get one stat-raising item by beating the boss. These items are random, but still designed in a way that made it so your character would have some kind of physical growth as you progress through the game.



I approached the roguelike design from many different directions with *Isaac*, but at its core, what made *Isaac* different than most roguelike games (well, aside from its visuals) was how I dealt with the difficulty curve. Instead of using traditional difficulty settings, I simply made the game adjust to players as they played, adding increasingly difficult content to the game as they progressed. This made *Isaac* feel longer, richer, and gave it the appearance of a story that writes itself. Using this design also allowed me to reward the player for playing and playing well, with more items that would help aid in their adventures and keep the gameplay fresh and exciting.

Once the player finally overcomes Mom, they usually assume the game is over, but instead get a new final chapter, six new bosses, a new final boss, and new items that shuffle into the mix. When the player beats the final chapter, they unlock new playable characters and items, and when they beat the chapter with each new character, they'll unlock even more content that makes the game even deeper still.

With *Isaac*, my goal was to create "magic." I wanted players to feel like the game was endless and alive, that the game had a mind of its own and was writing itself as they played. I remember the original *Zelda* having this feeling of magic and mystery. You weren't sure what things did until you experimented with them, and you had to brainstorm with your friends and put all your findings together in order to progress. I felt like since I was referencing *Zelda* so much in *Isaac's* core design, I should also complement it with the feeling of mystery I felt it had back in the day.

2. Uncensored, Unique Theme

I strongly believe that game enthusiasts want what they haven't seen yet, and that adult gamers should be treated like adults. Some people might argue that *Isaac* isn't "mature" when it comes to its content, but those people would be ignorant fools! When I designed *Isaac's* story and overall theme, I went in wanting to talk to the player about religion in a manner I was comfortable with -- that is, with dark humor and satire.

A lot of the content in *Isaac* is extremely dark and adult. It touches on aspects of child abuse, gender identity, infanticide, neglect, suicide, abortion, and how religion might negatively affect a child, which are topics most games would avoid. I wanted to talk about them, and I wanted to talk about them in the way I was comfortable with, so that's what I did with *Isaac*.

I'm not saying everyone who played *Isaac* did so because they cared about these themes, or that they even understood why they were in the game, but I strongly believe that this adult conversation I dove into with *Isaac* is what made the game stand out to people and kept them thinking.

I grew up in a religious family. My mom's side is Catholic, and my dad's side is born-again Christians. The Catholic side had this very ritualistic belief system: My grandma could essentially cast spells of safe passage if we went on trips, for example, and we would light candles and pray for loved ones to find their way out of purgatory, and drink and eat the body and blood of our savior to be abolished of mortal sin.

As a child growing up with this, I honestly thought it was very neat -- very creative and inspiring. It's not hard to look at my work and see that most of the themes of violence actually come from my Catholic upbringing, and in a lot of ways I loved that aspect of our religion. Sadly, the

other side of my family was a bit more harsh in their views on the Bible; I was many times told I was going to hell for playing Dungeons & Dragons and Magic: The Gathering (in fact, they took my MtG cards away from me), and generally condemned me for my sins.

I wanted *Isaac* to embody this duality I experienced with religion. I wanted it to show the positive and negative effects it had on me as a child - the self-hate and isolation it instilled in me, but also the dark creativity it inspired. The Bible is a very good, creatively written book, and one of my favorite aspects of it is how so many people can find different meanings in one passage. I wanted *Isaac* to have this in its story as well, which is why the game's final ending(s) have many possible interpretations.



3. The *Wrath of the Lamb* expansion

Doing a DLC expansion was never in the plan for *Isaac*; I assumed the game wasn't going to do well, so it wasn't something we really ever even talked about. I had a few pages of "dream ideas" that I wanted to add to the game, but I had to stop working on them and put them on the back burner, since I wasn't sure how much the extra content would matter if the game didn't do well. Six months later, we ended up taking these dream ideas and expanding them into an extra-large DLC expansion.

The *Wrath of the Lamb* expansion added over 80 percent more content to an already-bloated experience -- and people ate it up. 25 percent of the people who purchased *Isaac* also paid for the expansion, and that ratio is going up by the day. We honestly didn't expect to make it, but once I started seeing such a positive, creative fan response, I felt obligated to continue Isaac's adventure.

Honestly, however, the number-one reason why I did the expansion was because my wife Danielle had already 100-percent-completed the game. It was the first game I had designed that she became obsessed with (she's actually playing it right now, behind me, while I'm writing this), and it made me extremely happy to see her fall in love with something I had made. I just had to continue it (also, she wouldn't shut up about wanting more).



4. Circumventing Censorship with Steam

Steam is amazing, and with the *Isaac* release experience I've found another crucial reason: You can use it to sell uncensored and unrated games. This was vital with *Isaac*, because I wasn't going to bother with getting an ESRB rating for a game I wasn't sure was going to sell more than 100 copies. Valve knew the game was weird and could possibly get some backlash, but they allowed it on Steam because they felt it had potential, and I love them for that.

Another huge plus to working on Steam was the ability to constantly update *Isaac* with fixes, updates, and new content. They would upload a new build within the day we submitted it to them, and if we had released it on any other platform this would have been impossible (and probably cost us about \$40,000 to try).

5. *Isaac's* Fans

The number-one reason why you've heard about *Isaac* is its fans. Releasing *Super Meat Boy* and being in Indie Game: The Movie has shown me a wide range of fan types, but *Isaac* fans are just in a league of their own. At the time of this writing, there are well over 30,000 videos of *Isaac* on YouTube, countless pieces of fan art, animations, and plush toys all over the Internet, and over 30 fictional fan blogs where people can ask characters in *Isaac* questions and get in-character responses. It's totally surreal. Something in *Isaac* just spoke to a large group of creative people, and they held him up and ran with him.

Recently, I've been trying to find out how *Isaac* attracted such a creative and dedicated fan base. What is it about the game that spoke to this large group of artistic men and women? I can't ever know for sure, but I strongly believe that something in *Isaac's* theme and story connects to a large number of "creative outcasts."

I made it from the standpoint of a creative outcast; the game is about a creative child who is looked at as "made wrong" by the one person who cares about him, and his only real escape is his imagination. This is a story I could relate to, and it's one I think a lot of creative people latched on to mostly because it's not really a story you see in video games at all.

I am forever in debt to these people. Not only did they get the game to the masses, they also inspired me so much. You guys make me want to continue designing this game forever.



What Went Wrong

1. Shaky Launches

The Binding of Isaac was updated every day for two weeks during launch, and each time we thought we had solved all the issues. (Each time we were wrong.) Luckily, we were able to remove all game-breaking bugs in the first two days, but there were still many smaller bugs left that gnawed at us for a long time.

It sucked to launch with so many issues -- we had save bugs, game-breaking bugs that wouldn't let you complete the game, bugs that would not reward unlocks and achievements, and even some really odd ones that would scramble item clips and cycle through art from the game constantly. It wasn't pretty, and it was even more painful to watch so many upset players posting in the forums about the many issues with the game. (The biggest question, of course, was "Why didn't you test the game?")

The reason we released *Isaac* when we did was because it was done (if untested), and I didn't want to waste any more of my time on something I expected would crash and burn. I was just so worried it would suck that I wanted to get it out and over with.

2. Testing (and the Lack Thereof)



At launch, *The Binding of Isaac* had 100 items and five playable characters. 70 percent of the items in *Isaac* stack, and all the item abilities will affect Isaac in some way, so there were so many variables to keep track of that all the testing in the world couldn't have prepared us for launch.

Everything about the game was based on complex variables that multiply with each level you pass. In order to fully test all the variables we had in place, it would have taken hundreds of testers several days of extensive play time to fully debug this little monster -- there were bugs that actually took 100,000+ people four weeks to find due to how buried and rare some of them were.

Also, launching on PC meant launching on 10,000 different PC configurations, so we had bugs that would be caused by antivirus software, clean-up tools, and even some types of keyboard configurations.

The sad fact was that it was the day-one buyers that ended up fully testing *Isaac* for us, and I felt

really shitty about that. A few weeks after launch I put together a free mini-expansion to make up for our shaky launch -- but that, too, was filled with bugs.

3. Performance and Feature Issues with ActionScript 2

The biggest downfall of *The Binding of Isaac* is its performance. *Isaac* was designed in Flash using ActionScript 2; that's what Florian could program in, so those were the limitations we had to work around. Sadly, Flash AS2 is quite outdated, and even with all the amazing work Florian put in, we simply couldn't get the game to run well on lower-end PCs. Flash even had major issues with PCs that used dual-core processors, so even PCs with amazing specs would slow down at times.

If I had known that anyone would have cared about *Isaac*, I wouldn't have made it in Flash at all. Framerate issues aside, Flash's lack of controller support and integrated Steam features really hurt *Isaac*. It pained me to release a game that was lacking features almost all games have. You'd think by now Flash would have added some kind of controller support, but no. Tommy actually wrote an achievement program specifically for *Isaac* so it could award Steam achievements, which was hugely helpful, but I couldn't ever really feel satisfied with the product due to our AS2 limitations.

4. Too Big in Scope for Flash

Aside from the performance issues and AS2's limitations, late in *Isaac*'s development we soon realized that Flash simply wasn't at all made to support a game of *Isaac*'s size. Once the .FLA file rose above 300MB, we couldn't even consistently generate an .SWF file from it without crashing.

This issue almost prevented *Wrath of the Lamb* from coming out at all; we were at a point in development where simply saving the .FLA would corrupt it about 25 percent of the time. Florian would have to restart his PC and save the .FLA in a new folder every time we had to export an .SWF just to test it, and 50 percent of the time it wouldn't work for no apparent reason.

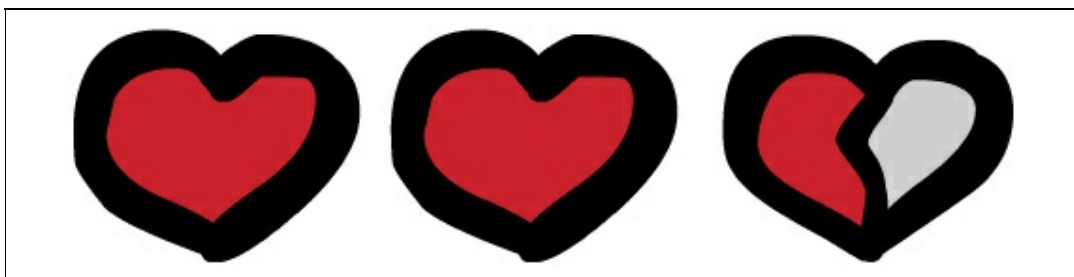
It was quite a horrible experience, and if we could have seen into the future with a crystal ball, we would have simply not used Flash. (Maybe this will be a feature in Flash CS7...)

5. "Blasphemy" and Controversy

Not surprisingly, controversy made a few appearances in *Isaac*'s release year, but not in the way you might think. During *Isaac*'s German retail launch, the German ratings board gave *Isaac* a 16+ due to "blasphemy." That itself didn't cause controversy -- instead, it was the idea that said blasphemy could affect the age rating on a video game.

Blasphemy isn't something you can define for everyone (what's blasphemous for one religion isn't necessarily so for another), so how could one define something as containing blasphemy? It was a very interesting argument, and I'd be lying if I said that having the first game rated 16+ due to blasphemy didn't feel awesome, but sadly it was this controversy that I believe eventually led to Nintendo's decision not to port *Isaac* to the 3DS.

I remember my wife being worried about *Isaac*'s release, worried that it might offend the wrong people and someone could end up being hurt. I can't say I didn't have some hesitation about this aspect of talking about religion in a satirical and possibly blasphemous way, but I couldn't help but avoid the simple logic that, well, most of those kind of people don't play games. And after over a year, I really believe that's true. (Thank God!)



Isaac Reborn

As of writing this postmortem, *The Binding of Isaac* has sold over one million units on PC and Mac in its first year on Steam, one-quarter of the people who own the main game paid for the *Wrath of the Lamb* expansion, and the interest seems to continue building.

A few months ago I was contacted by Tyrone Rodriguez of Nicalis (publisher of *Cave Story*, *VVVVVV*) and asked about how I felt about remaking *The Binding of Isaac* for consoles. I love consoles as much as the next guy, but dealing with the business end of console development wasn't something I wanted to dive back into at this point.

I told him yes, but I had a few strict guidelines to make sure an *Isaac* remake was perfect. I wanted the game to feature the second planned

expansion that I couldn't do in the Flash version, I wanted it to feature local co-op, I wanted the graphics to be totally remade in 16-bit but still look and feel like the Flash version, and finally, I didn't want to deal with anything when it came to business. Nicalis has agreed to these terms, and development has started on *The Binding of Isaac: Rebirth*.

It's still too early to tell for sure what consoles the game will end up on, but both Microsoft and Sony feel like it would be a perfect fit for their digital platforms, and we have a feeling the new look might soften up a few people at Nintendo for a possible Wii U/3DS eShop release. I'm wary about how the game might control on iPad, but if they can make it work, I'm all for it.

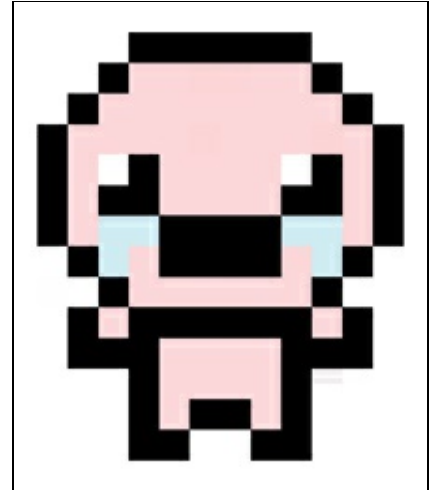
The Binding of Isaac was a huge personal achievement. I was able to talk about something personal and meaningful in a way I felt comfortable with, and I was able to get my feet in the water with the roguelike formula and random generation.

When I started on *Isaac*, my goal was to make a niche cult classic, something with a tiny but die-hard fanbase. What I didn't expect was how large a "tiny" niche audience would actually be.

But what moved me the most is the amount of creativity *Isaac* inspired in others. Every day I read fiction blogs, watch YouTube animations, and look at others' illustrations while thinking about how honored I am to have made something that could have helped motivate so many to create.

The three months Florian and I put into creating *The Binding of Isaac* didn't just pay off with a financial windfall -- it also gave us an eye-opening experience that proves to me without a doubt that people truly want and respect games that are uncensored and risky, and that ask more of the player than most games these days do.

[If you enjoyed this article, please consider [a physical or digital subscription](#) to Game Developer magazine, where you'll find a wealth more exclusive content, and where this postmortem originally appeared as the cover story in the November 2012 issue. An [updated, Retina-compatible version](#) of the Game Developer iOS app is also now available.]



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

Copyright © 2016 UBM Tech, All rights reserved