Postmortem: Smudged Cat Games' The Adventures Of Shuggy

PRINT

By David Johnston

[*In this revealing postmortem of The Adventures of Shuggy, Smudged Cat Games' David Johnston reveals how working with the best of intentions and inspiration can have a bad commercial result when the cards fall in the wrong places creatively and business-wise.*]

The Adventures of Shuggy was released on Xbox Live Arcade on the 15th of June 2011. I had started work on the game *Shuggy* would become back in the beginning of 2007; what followed was four and a half years of various ups and downs.

It was developed using Microsoft's XNA framework which enabled me (as a lone developer) to get something up and running really quickly. The game was entered into Microsoft's Dream Build Play competition in 2007 and made it into the top 20.

Sadly, it didn't get one of the four top prizes that were issued that year, which would have led straight to an XBLA contract, but it did gain enough publicity that a few publishers were interested, and a deal was signed with Sierra the following year.

Things were going great until Activision Blizzard took over Vivendi Games (the owner of Sierra) and decided to wind down Sierra's operations.

After a long period of uncertainty, the publishing contract was terminated at the end of 2008, leaving me looking for another publisher. Eventually a new deal was signed with Valcon Games at the end of 2009, allowing me to complete development and finally get the game released.

Smudged Cat Games didn't exist prior to the development of *Shuggy*, but I'd worked on various personal projects over the years, which gave me the experience I needed to complete a game and get it out the door.

I've only actually spent one year working in the "proper" games industry, generally finding the lower pay and longer working hours a turn off compared to working as a general software engineer. I'm still the only employee of Smudged Cat Games; everyone else involved in the project was brought in as a contractor.

What Went Right

1. Simple Control Scheme

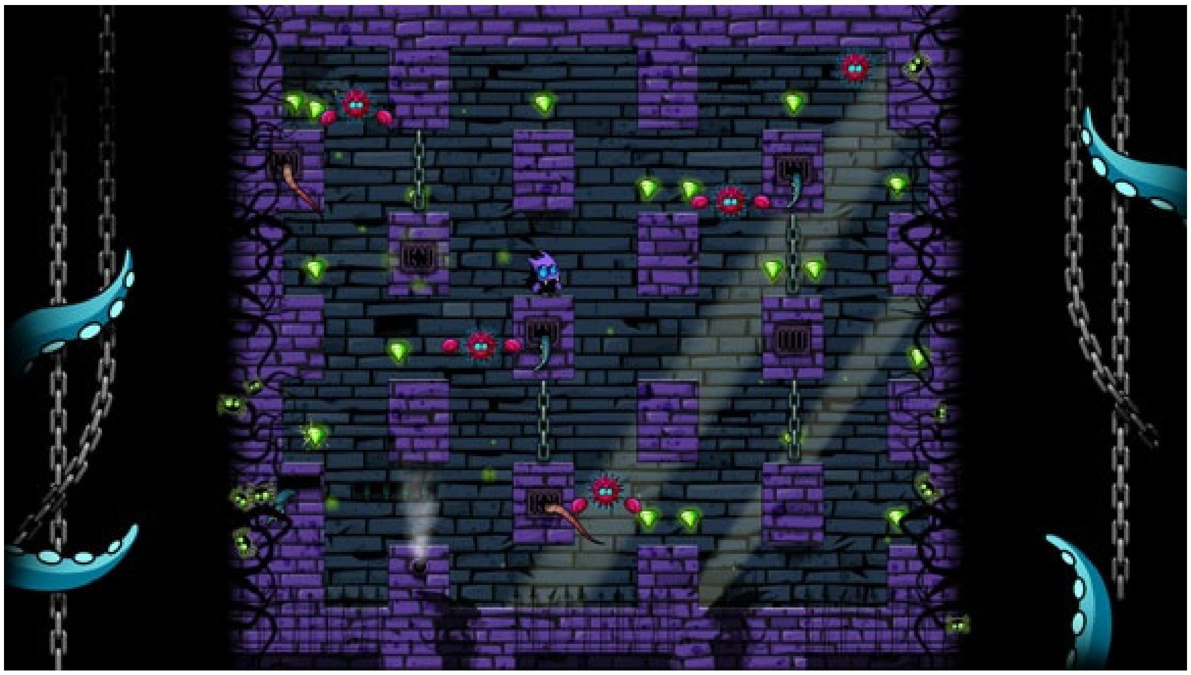
Something I decided quite early on in *Shuggy*'s development was that the control scheme should consist of simple left, right, jump, and action buttons. I quite often find myself getting into a video game, but then things come up, and it's a while before I can get back to playing again.

I try returning to the game, but end up spending too long trying to pick up the control scheme again. I'll end up playing for a confused hour; it feels like more of a chore trying to get back in the rhythm again, so I never get round to finishing it. With *Shuggy*, it's easy to play a few levels, leave it for a week or so, and then come back to play a few more.

Moving around and jumping in any platform game comes pretty naturally, so the only thing you need to remember in *Shuggy* (or figure out when you play the game again) is that the action button is mapped to the right trigger. Obviously the function of the action button changes dramatically depending on the mechanic of the level, from speeding up time or rotating the level to swinging on a rope -- but there's only ever one button you need to try giving a push.

I guess I was lucky that the different game mechanics I came up with were fairly easy to implement using just a single action button. It did limit me to only using a single mechanic in each level, so if there is ever a sequel I might break this rule, but for this game, the mechanics were good enough to stand on their own without throwing everything into the pot at once.

There are some levels where it might have been nice to be able to use more than one action, such as speeding up time when you're playing a rope level, but overall I think the simplified control scheme works well and really helps improve the accessibility of the game.



2. The Game Features Time Travel

Time travel seems to be a favored game mechanic now with *Braid* and *P.B. Winterbottom* having done well over the past few years. The time slip levels featured in *Shuggy* were some of the first ones to be added to the game -- well before *Braid* and *Winterbottom* were released, I hasten to add.

I've always been interested in time travel as a mechanic. I actually made several games in the past that featured the same time travel mechanic which appears in *Shuggy,* where you go back in time to encounter past echoes of yourself.

The first time I remember making a game with time travel was way back in 1998, with a game called *Groundhog* on the Acorn Archimedes. I then made *Timeslip* which was originally developed for the PlayStation Net Yaroze back in 1999, and featured on a cover disk of the Official UK PlayStation Magazine. I recently ported it to the Xbox 360 and released it as an Indie Game. After that, I released a PC game through my own website called *Knight Time* which allowed you to move back and forth through time rather than force you back after a set time.

I love the fact that in the time slip levels you're your own worst enemy. Most of the levels are fairly simple to complete if you just stay aware of the fact that you'll need to avoid bumping into your past self somewhere down the line.

I've seen some forum posts complaining that a time travel level is too hard, but it's really only has hard as you make it Remember before you charge in a door that's just opened, that you'll probably be stood on the other side of it at some point, wanting to come back. I also find it helps to use the clock and make note of what time it is at certain key moments so you know what you're going to be doing next time round.

When I made the other time travel games, I found it does limit what you can implement. If you travel back in time, then everything needs to happen the same way each time round. It stops you having enemies with random attack patterns and ones that chase the player. It becomes difficult to implement objects that the player can pick up and put down.

If they pick up an object at 20 seconds into the time slip, put it down somewhere that activates a door, travel back in time and pick it up at 10 seconds into the time slip, then what should happen? The past version won't have the object to pick up, which means the door won't open. If the player has gone through that door at some point, then he'll now be going through a closed door. In *Shuggy*, I had to avoid those situations in the time travel levels, but because Shuggy isn't all about time travel they could feature in some of the non time travel levels.

---

3. The Variety of Mechanics

The main, unique selling point of the game is undoubtedly the many different game mechanics featured throughout. When I started development, the intention was to make a platform version of a *Wario Ware* game, where every single level was unique and lasted about 20 or 30 seconds. As it developed, it became clear that some of the ideas were certainly worthy of a few levels, and the levels that required more time to solve were more interesting than the really short ones.

A lot of the original level mechanics were dropped as they didn't really work, and I focused more on the interesting ones such as rotating the levels, time travel, and rope swinging. I had a couple of levels that featured playing cards which Shuggy had to pick up and form poker hands with. They just involved a lot of picking up and dropping -- not that interesting!

One level didn't rotate around the Z-axis like all the normal rotating levels, but around the Y-axis, so the level was a bit like a 3D cube. It was a crazy idea and didn't really work out so got dropped from the final game. The diversity in the levels made it nice and easy to just drop certain mechanics if they weren't working out, and replace them with something else.

The different mechanics came about through a variety of reasons. Allowing the levels to rotate and having gravity different for each object was a major part of the engine design, so I did that first, rather than trying to retrofit it later on -- which was a wise move. All the objects in the game are coded in such a way that they don't care which direction their gravity is; it's all handled by the base class used for all objects. It meant I could add new objects without needing to consider that they might be rotated at some point and their gravity would change.

After that, I implemented the time travel effect, which came easily thanks to my experience in the genre. The different jumping effects were simple enough to add, and just involved tweaking a few calculations here and there. I enjoyed implementing the "teamwork" levels where another character helps you in the level.

They were quite easy to implement (simply record button presses while I went from one position to another, and then play the appropriate recording back) but really make it look like someone else is controlling the other character. Probably the last mechanic to be added was rope swinging, which led to quite a few interesting levels, and the inclusion of the cogs you can wrap the rope round.

4. Skippable Levels

I'm really happy with the way players can progress through the game. From the beginning I wanted players to have a choice of levels to pick from at any given time. I hate that feeling of progressing through a game only to get completel stuck on a specific level. It normally leads to giving up and then you miss out on playing a load of content.

A few reviewers have commented that there isn't really a difficulty curve in *Shuggy;* some levels in the first few areas are much harder than some of the levels in the last area. It was a deliberate move, though. It means players who aren't as skilled should always be able to find easier levels to play; ignore the difficult ones and you can still reach the end of the game.

Meanwhile, players who are after a bit more of a challenge can start tackling more interesting levels early on, rather than wading their way through loads of levels they don't find as enjoyable. That wouldn't have been possible if it wasn't for the way the levels unlock as you progress.

5. Remote Working

Due to the nature of the way the game was developed, most of the team have never actually met each other. Being an indie project and having no budget at the start meant that everyone was working remotely from their own homes.

After coming up with an initial prototype of the game myself, I knew I needed someone to take care of the visual side of things. I posted on an indie gaming forum and got in contact with Chris from Imbri Design, based in Australia, who ended up doing all the in-game graphics. We've only met face-to-face once, at the 2007 Gamefest conference, where the Dream Build Play winners were announced.

Bennet Aldous at Fat Cat Comics, who did the comic book cutscenes, is an old friend of mine based in the UK. Pompom Games and Sonic Source who worked on the Live support and sound effects respectively are also UK-based, but I've only met them a few times. I've still never met Jesse Hopkins, who composed the music for the game and is based in the U.S.

Despite having little face-to-face contact with the other people working on the game, we managed to pull everything together and get a finished product out there which I think is quite an achievement. I was concerned at the beginning (particularly as more people were brought on board) that working remotely was going to pose some significant problems but it really didn't cause any major upsets.

We had a good revision control setup that enabled us all to work on the code and assets at the same time, and I coordinated what everyone was up to. I don't believe any of the other team members have actually spoken to each other!

What Went Wrong

1. It Took Too Long to Release the Game

I believe the single biggest problem was how long it took to get the game into people's hands. Development started at the beginning of 2007, and the version of the game entered into Microsoft's Dream Build Play competition wasn't vastly different from what was finally released in June this year.

Seeing games such as *Braid, Lazy Raiders*, and *P.B. Winterbottom* released, which use mechanics featured in *Shuggy* was upsetting. I knew my chance to stand out kept slipping away. Not only that, but there has been an understandable assumption that I stole the mechanics of those games for *Shuggy*. Particularly upsetting for the time travel / past echo mechanic, as I'd released *Timeslip* back in 1999.

I can't help but wonder how the game would have been received had it been released sometime in 2008, but as a lone developer there was no way I could fund an XBLA release myself. It was unfortunate that Vivendi was taken over by Activision Blizzard, as it (eventually) decided to terminate the *Shuggy* contract, but it was something I had an infuriating lack of control over.

2. The Trial Doesn't Sell the Game Enough

The demo, along with most other elements of the game, went through a few changes over the course of the development. I initially implemented a trial similar to what ended up in the final product, containing the first few levels of the game from the dungeon area.

The first publisher felt that players might think the dungeon was all there was, and wanted to get across the fact that there are five areas in the game, each containing 20 or more levels. The trial was changed to levels scattered throughout the game rather than just at the beginning so it was obvious how deep the game was when you played the trial. The other advantage of this approach was that it gave the chance to show off more of the different game mechanics rather than being limited to what was available in the dungeon.

When Valcon Games took over as publisher, the company felt that leading the player around all the different areas might be too confusing and wanted to go back to the first few levels from the dungeon. The idea was that we would still convey that there are many different game mechanics through a couple of videos that pop up as you play through the trial.

I think changing back was a big mistake. Although we've got the videos, I'm not convinced that players really pay much attention to them. When someone is playing a demo, they just want to play, and can easily skip over videos. I've done it myself -- when upsell screens appear as I'm playing a trial, I mash the A button hoping to continue, and don't really look at what's on screen. I should have let the game sell itself, rather than hoping the upsell screens would do it

3. Unclear Target Audience

Since the release of the game I've seen some reviews that really like the graphics and some that don't. However, one thing that most reviewers seem to agree on is that the graphics don't fit the gameplay. A lot of the levels in *Shuggy* are aimed at hardcore gamers who know their way around a platform game, but the graphics make the game look like it's aimed at a much younger audience with less experience.

I worry that a lot of hardcore gamers have disregarded the game, assuming that it's some generic platformer aimed at pre-teens. Similarly, I imagine there have been some younger gamers out there who have downloaded the trial liking the look of the game, but have found that the levels are far too difficult for them.

It's a shame, because I really like the look of the game, and I'm not a fan of the gunmetal grays and rusty browns that tend to dominate games aimed at a more hardcore audience -- but in future, I'd try to keep the look of the game more in line with the audience that it's targeted at.

---

4. Publicity

Shuggy appeared on XBLA on the 15th of June, without warning or fanfare.

I left publicizing the game in the hands of the publisher, and ended up feeling quite disappointed at the amount of publicity Valcon managed to generate. Once we had a release date, I felt we needed to be letting everyone know about the game. I spoke to the publisher as I felt we should be emailing game video and screenshots to as many web sites as possible.

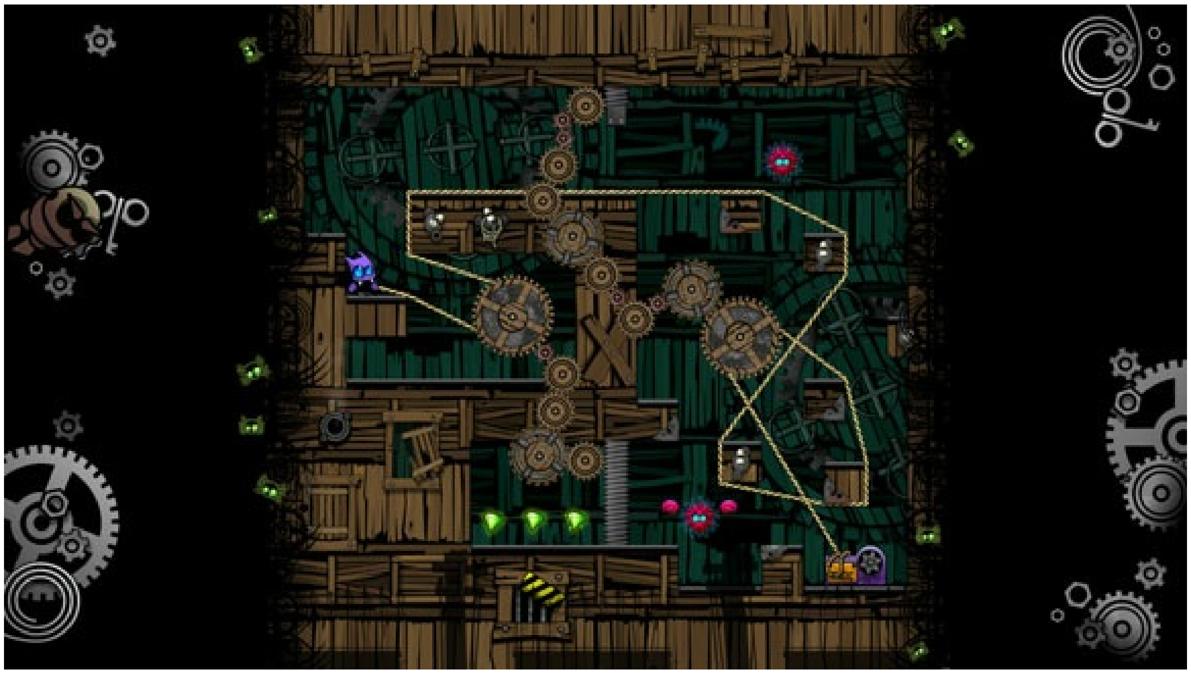
Valcon didn't want me to contact any media directly, as this would be duplicating their efforts, so I left the publicity in their hands. It took ages for the official press release about the game to appear, because apparently Valcon was waiting on it being approved by Microsoft. We even had to wait until the game was actually released before getting any review codes, meaning there were no reviews of the game available on the day of release.

As this is my first XBLA, game I don't know what kind of PR Microsoft could or should do, but *Shuggy* didn't get any attention on the Xbox dashboard. Combined with the lack of pre-release publicity from Valcon, this meant that the game went live without anyone knowing about it.

Shuggy was also released on the same day as the latest *Magic: The Gathering* title, which got a lot of attention and has been really successful. The combination of going out cold and up against such a strong title really didn't work in *Shuggy*'s favor.

Trying to put myself in Microsoft's place perhaps its lack of interest stems from the fact it took so long to release the game. By the time *Shuggy* came out, a few of its more interesting elements had been overshadowed by other titles, so Microsoft may have felt the game was treading over old ground. I don't know any of this for sure, and still think that each new XBLA release should have some kind of attention of the Xbox dashboard.

In hindsight, I wish I'd followed my instincts and taken the matter into my own hands. I'm going to make sure I give any future projects a bit more publicity during development, rather than waiting until they're released. I've learned this lesson the hard way.



5. I Tried to Do Too Much

Like most projects, *Shuggy* started out fairly simple, and then more and more features kept getting added. As well as adding the various different game mechanics, I had the co-op, challenge, and head-to-head modes, as well as some grand plans to have various minigames themed around some of the mechanics. The minigames actually got implemented, but never got polished enough to make it into the final product, after Sierra suggested that there might be a bit much going on.

Since then, I've released the mini-games as separate Indie titles. There's *A Bomb's Way*, based around the floating jump and rotating mechanics which is like a version of the old arcade hit *Bomb Jack*, the game the floating jump was based on. *The Tower: A Bomb's Climb* is based around the zombie game mechanic where the player doesn't have direct control over the character, but can power up his jump to propel him up a never ending tower.

I'm not a big fan of the head-to-head mode; I don't think it's that interesting, and while I quite like the challenge mode, I think it would have been best if I'd just implemented the single player and co-op modes from the beginning... and added support for playing co-op online. It's too easy to get carried away when you come up with an idea for something, but when you're working with such a small team, I've learned that you really need to keep things simple and focus on the best aspects of your game.

The End?

The development of *Shuggy* has been a bit of a headache and a cause of concern for me over the last four years. So much so that it has made me consider if it's what I really want to do, particularly considering how badly the game has sold to date. My wife and I now have a newborn baby (a lot of this was typed out using one finger as I held my baby in the other) so I'm not sure I can face the financial risks that game development brings.

I have another couple of projects in the pipeline that I'd like to see finished, and see how they go, but I'm just not prepared to take the same risks that I took in order to complete *Shuggy*. They'll probably be released as Indie Games. While I will always love playing video games, and dream of crafting my own virtual worlds, I fear this may be the end of game development for me.

Data Box

Developer: Smudged Cat Games
Publisher: Valcon Games
Release Date: 15th June 2011
Platforms: XBLA
Number of full-time developers: 1
Number of contractors: 5
Length of development: 4½ years
Budget: $250k
Lines of Code: 70,000
Key Development Tools: Visual Studio 2008, XNA Framework 3.1, CorelDRAW, SVN
Number of SVN revisions at release: 666

Return to the full version of this article