

An indie-style experiment at a AAA studio: Insomniac's Slow Down, Bull

 gamasutra.com/view/news/259163/An_indiestyle_experiment_at_a_AAA_studio_Insomniacs_Slow_Down_Bull.php

This postmortem, written by current indie and former triple-A dev [Lisa Brown](#) tells the story of the development of Insomniac's Slow Down, Bull -- an indie-style small game made by a big, well-known developer.

Insomniac Games has a reputation for always being willing to experiment. Whether it's trying to blend game genres, evolving a proven gameplay mechanic or branching out into a new platform, that spirit is something I've admired for a long time.

In the summer of 2013, mid-production on [Sunset Overdrive](#), we tried a different kind of experiment, and I was thrilled to be involved. The premise: How far could one person take a prototype before needing to roll a team onto the game? Could we also make a great game with a small team and shorter timeline than our typical big budget console games?

When building the prototype for the pitch that ultimately became [Slow Down, Bull](#), I started with a few mechanics constraints. First, I wanted to make a game with constrained input, only two buttons. Second, I wanted to try a game where your input stopped movement instead of caused it.

Eventually, this prototype turned into Insomniac's first small PC game and first foray into the realm of open development. *Slow Down, Bull* is an action collecting game about a stressed out, overachiever bull named Esteban who just wants to collect beautiful things, but is constantly worried that he isn't doing well enough. It became a charming little game wherein we partnered with Starlight Children's Foundation to give roughly half the net proceeds to the charity.

It was definitely a bit of a wild experiment for us in a number of ways, and we learned many things along the way.

What Went Right

1. Long prototyping phase

Because the whole initial process was a bit of an experiment, we spent a long time with just me working on the prototype alone, doing all the coding, art, animation, sound, telemetry, and playtesting. It was roughly four months of intense iteration on the prototype before putting something together for a broad company playtest to be greenlit.

After we made the decision to go ahead with the game, but before the full team rolled on, we spent some additional time pitching the project to potential partners amidst some extra experiments on the prototype. Do note that this wasn't a continuous timeline (the studio hibernates for a brief time during the winter holidays), but even still it may seem like a long time to stew on a single prototype.

However, I feel like this was one of our strongest decisions, as the rapid prototype iteration and consistent design log documentation meant that we had a strong, coherent prototype that made production with the entire team move swiftly once they came on board. We were able to iterate through a ton of different experiments, many of which were discarded failures, but which paved the path for the strongest mechanics in the game (the bullcatcher, the possum, and even the cat all were birthed out of a long line of experiments.)

Some of the discarded prototypes included a red light/green light mode, a mode in which you had to collect pickups in predetermined order, a pickup that increased your stress the longer you held it, a mode where you had to steer on a specific path, and a thief who stole decorations that you had to charge into. While these were ditched for not being particularly fun, they helped clarify what WAS fun and distinct about the steering and stress

mechanics, and thus the time spent on trying them was incredibly valuable.



2. Developer Streams

When *Slow Down, Bull* was greenlit, we made a plan pretty early on to do live developer streams on Twitch, as part of the experiment effort. Every week from June to November 2014, the team, with help from our in-house community/marketing team, streamed ourselves working on the game.

Sometimes it was me building a level, or the programmer implementing an enemy behavior, or the artist creating environment assets, or our composer creating a song. During the streams the team also answered audience questions about game development and generally chatted about process.

I feel like the streams were really positive on several fronts. It unified the team under a common weekly event (streaming yourself building the game is pretty stressful, so it bonded us together!) It also connected us with a curious audience of fans. Some of our favorite moments were letting our stream audience come up with names for the characters in the game, or meeting people at events like PAX who had been coming to the streams since the beginning, and who expressed how grateful they were for being shown “behind the curtain” of the development process.

Production-wise, doing the weekly streams actually made the team work in a very efficient way. We felt accountable to show progress to our stream audience every week, and so were calculated in how we scheduled and scoped the game. The fact that the streams proved successful on a production pipeline front was a bit of a surprise, since they were a fair amount of work to wrangle every week.

3. Team ownership

Besides me, the team for *Slow Down, Bull* was composed entirely of contractors (though some still had some connections with Insomniac -- a former animator gone freelance, a former intern of ours, a QA tester with game art aspirations). Since the game ended up being a highly personal one for me in theme, I knew that it was going to be important that the team felt a connection to the game beyond just being hired hands to produce it.

On the other hand, by the time the team came on there was a strong vision for the feeling of the game, and on a relatively short production cycle (about seven months) we had to be able to hit the ground running. In the end, I felt like we were really successful in creating a strong team bond in which everyone felt creative ownership over the game, which yielded each component of work feeling very personal.

Part of this was finding ways to get the team members to relate to the overarching theme of the game (an overachiever who is stressed out by that which he loves, always worrying that it isn't good enough). Besides being a common struggle of highly creative people, the developer streams ended up bringing everyone together over this theme, as we were all passionate about showing the development process but nervous about showing our unfinished, imperfect work to the public. It made us all relate to the story of Esteban, and was easier to put a bit of ourselves into the game.

In addition, all team members were involved in all discussions of theme and story, even if it didn't relate directly to the work they were assigned. Any time there was a decision to be made about how to present a story element or how to flesh out one of the characters, I made sure it was clear that the entire team was to be a part of those discussions, which took place in our team Google chat.

4. Access to internal resources

The biggest advantage that the large studio setting gave this project was access to resources. During the prototyping phase, other designers were frequently involved in the playtesting process. Studio leads were available to help with questions and processes for leading the project (for example, preparing strong pitches, advice on wrangling contractors, etc.) When we ran into a tricky animation problem, we had a veritable army of highly skilled, experienced animators to consult.

In a way, the act of consulting an expert became a form of delegation, and pure brain-expertise became a resource to be taken advantage of. It almost felt like being in an incubator for creating an indie game, with the process itself being relatively independent, but the fact that help on a complex programming struggle was just a desk or two away.

At one point in the project when thinking about feedback on the collectibles, I polled the entire company on what made juicy feeling pickups, and what made the bolts in *Ratchet & Clank* feel so good. Every department had a different take, of course, and years of experience on that particular piece of polish. Compiling that sort of data was invaluable, and I think a big help on a project with such a small team.

5. Good scoping and production management

We did a great job of keeping the scope of the game down, and making smart decisions about what was critical to include and what to cut. It would have been easy to get caught up in lofty goals about multiple platforms, leaderboards, and squeezing in those last few features because the prototypes were so much fun.

But I think we had a pretty strict, systematic way of addressing cuts, looking clearly at how many resources it would actually take to get things done, and proposing them in such a way that features and cuts would get approved (sometimes it surprises people to learn that even cuts can need approval, and that a cut in a game still creates work, so making those decisions early in our production was important).

I also felt like our team worked efficiently, encouraged by things like our developer streams and also our means of constant communication. Two of our team members were remote and in different time zones, so we made it a habit of always being available in our team Google Hangout, with periodic calls with the whole team. We used Trello for task tracking to good effect, and while we were all on mic after each weekly stream it created a good natural time for reviewing how our work was progressing and if anyone was running into any problems.

We didn't adhere to any formal production process, and what we wound up with was a bit organic, but since it was such a small team this worked out in our favor, and the game was completed on time.

What Went Wrong

1. Not good late telemetry (balancing was off)

We did a ton of playtesting and telemetry in the early stages of the project when everything was coming together, which was super useful for deciding early on what features to keep and to drop and get an idea of if the overall

game was going to be fun. However, towards the end of the project, we got caught under the timeline of finishing the game on time and didn't devote as much time and resources to telemetry and playtesting.

We had some systems design help to balance the meta (figuring out how many decorations would grant how many stamps in each level, how many stamps you need to complete an area, etc.) This was a huge help in the late production when the rest of the contracts had ended and I was laser-focused on bug fixing. As a result the game was way more balanced than it would have been, but most of our playtesters during that time were people at the company who had some familiarity with the game already, and many focused on the first portion of the game.

The result is that the first couple of areas balance out pretty well, but then you get difficulty spikes around Area 3, and the last area of the game has too lax of a stamp requirement to beat. The progression pacing is just off. Doing more thorough pre-release testing and telemetry with external playtesters would have helped iron out the balancing.

2. Streaming and market

Though the developer streams were something I considered a big "what went right" for the project, they came with their own set of mistakes. *Slow Down, Bull* has a very simple control scheme, and in initial prototypes we tested in mouse, controller, and touch. However, we know that in keeping the scope small that we would only be able to target one platform to start. I argued to target PC first, because that's where the audience for dev streams were, and because I felt like a quirky, premium game might find an audience through those streams in a stronger way than the more casual mobile/tablet audience. The hope was to release the game on PC and then follow up with touch interface depending on how it performed.

Though we had a small, consistent group of enthusiastic regulars in the streams, we didn't gain a huge following in spite of pushing the streams regularly to the Insomniac following and weren't able to secure a Twitch front-page push. There were struggles in dealing with how to leverage the streams as a growth tool, much of which was due to some timing challenges (addressed below), but in the end our findings pointed more to the fact that the market was not there after all. Choosing a mobile distribution first, or even a different PC distribution besides or in addition to Steam that might have reached a different audience (itch.io, Humble, etc.) would probably have been a better approach.



Looking back on it, some of it is the usual amount of luck, as it's impossible to tell when a concept is going to

take off, but a big part is being more aggressive about finding your audience early. Showing the game around in various contexts, I found that many people who it resonated with weren't necessarily traditional gamers (for example, traditional artists who loved the look), or they were gamers who spoke specifically about the refreshing nature of the nonviolent approach the game took, or folks who were once overachieving students in liberal arts education who identified with the character.

These were people I encountered on an individual basis when showing the game in informal settings, but finding them earlier would have given better insight about distribution. Do these people play games on PC? If they do, do they get them through Steam? Do they play games on mobile? We could have done more in finding exactly where the audience for this game was, and determining whether or not it was big enough to be sustainable earlier.

Finding these people isn't the same as running a focus test, but rather finding them involves getting the game out there -- showing it at game festivals but perhaps non-games events as well, and talking about the game early in as many outlets as you can. This is why I think this piece of advice is so prevalent for indies -- doing this legwork to find the audience early can inform you about the best platform through which to reach them.

3. Leaderboards are a crutch

This is a lesson learned that I feel can be applied to any game of any size. In early prototyping, within a smaller group of testers who already know one another, leaderboards can be a crutch that draws attention away from what you need to be learning about your game. In the first company playtest of the prototype, I hacked in some rough leaderboards for high scores on each level to see how that felt. The result was that a small number of hyper competitive people had a great time trying to one-up each other for the high scores for each level.

While this was great fun, I think it gave a false sense of where the fun came from in the game relative to the amount of work it takes to actually implement leaderboards in a game "for reals." When we had a final understanding of our timeline and I decided to cut leaderboards up front so that we could actually finish the game on time, there was some response of confusion since, after all, I'd gotten leaderboards up and running within a couple of weeks for the prototype. (However, that "prototype" leaderboard was completely unsecure and barely reliable).

I was conflicted on whether to put this as a "what went wrong" at first because, after all, at the time of the prototype I didn't know if leaderboards could be a thing in the final game and wanted to test them out. However, I wish I'd done that initial playtest without them, so I could have gotten better insights into the fun of the mechanics themselves without the distraction of "it's fun to beat my co-worker to get the high score."

4. Unity's animator (not knowing when to roll our own solutions)

Though the initial prototype for *Slow Down, Bull* was built in Construct 2, we decided to go for Unity for the final game for a variety of reasons (performance issues, a seasoned Unity programmer on the team, a few technical features that were not possible in C2). We decided early on to make use of Unity's animator tool, rather than controlling how and when animations were played in code, mostly because a general look at what it was capable of seemed like it was usable.

The animator at the time seemed deceptively useful, but ended up causing us more headaches than anything else in production, and I feel like we should have done a bit more deep-dive testing of the system before committing to it. There were a lot of problems with using the animator for certain things that also had rotation or translation dependencies on code events (for example, Esteban's head was rotated in code in certain circumstances).

We also had a lot of trouble with bugs in more complicated state transitions between animations for certain characters, especially anything that involved any sort of layering (Annette the bullcatcher was notorious for not transitioning properly through her various tell states, even when all the variables were set, due to the unintuitive transition settings in the animators overriding what animation was being told to play in the state machine).

At the end of the project, it felt like we had had to fight and conquer Unity's animator to get it to do what we wanted, rather than it being a tool to help productivity. The more general takeaway here is that, while out-of-the-box tools are amazing in making small scale development fast and accessible, it is still important to take time to do a bit of trade-off analysis and decide if it's better for your project to roll your own tools or methods.

5. Making a little game in a big studio

I mentioned earlier that a big pro to doing a small scale production in a large studio was access to resources, but the big studio environment created some obstacles as well. One big challenge was melding Insomniac's triple-A console and mobile marketing experience with something more experimental in nature.

Slow Down, Bull is a small and quirky game that is incredibly personal to the team who made it, and typically with small indie games the personal slant can be leveraged in marketing the game. However, in a bigger studio with many projects, each game has to fit within the brand of the company and share promotion time with all of the other projects. It's a balancing act. This game had to be seen as an Insomniac game first and foremost, and though we highlighted some of the personal nature through the dev streams, pushing a development story that was very close to the individuals on the team would have been unfair to the rest of the company as a whole.

The end of production fell to bad timing, right during our push on *Sunset Overdrive* and then the Android port of *Outernauts*. It was important for the company to not divide its message across games, so the result was there was a big delay between wrapping up the developer streams and the launch of the game so we couldn't tie launch promotion to the streams. Many times smaller games tend to aggressively stir up word well before launch to pinpoint its audience well ahead of time.

We ended up doing something in between, with regular blog posts, dev streams, and press-driven announcement prior to launch, but with many other projects to manage, we didn't really have the bandwidth to do the aggressive, early audience-seeking promotion that you see in a lot of smaller niche titles.

We ran into a few issues with "small game at big studio" in other places, too, in looking to submit it to festivals. There was a disconnect between the small scrappy nature of development and the fact that it was a game from a large established studio, which meant that it didn't really "fit" in a lot of places that better serve smaller indie teams.

Lest I come off as bashing the marketing team, the reasons for resistance here are perfectly reasonable. When you have a large company with an already established brand, controlling that image and keeping it consistent is paramount, or else a lot of things can go wrong. I just think we weren't anticipating some of these challenges early on when we decided to do the experiment to begin with.

Conclusion

In the end, I feel like we made a solid, quirky game with an unusual mechanic and an incredibly charming, relatable story. The schoolcraft art is beautiful and the music is amazing. Watching streams and Let's Plays of the game and seeing people "get" how the stress of the gameplay resounds with the nature of the story and character has been really rewarding. However, we didn't find an audience large enough to be sustainable through the distribution channel that we chose, though in the end we gained a lot of insight about the challenges of making small games at a big studio.

Will Insomniac try more games like this in the future, changing things up with all the lessons we learned? That remains to be seen!

As for me, the whole experience has resulted in me going indie. It made me realize that to be at my best and happiest was to be making games for the joy of making them, and to be able to make games for smaller audiences that may not be sustainable to a large studio (even if the large studio is making a small game).

The developer streams were such a positive experience for me that I'm focusing most of my work on that -- streaming production on small, experimental game development, educating aspiring game devs and curious fans

- and being able to do so in a context that is more easily suited to smaller, personal games than within a large studio structure that still has to be concerned with market. The challenges of marketing on my own terms and building a community from scratch are ones I feel ready to face after the learning experience of this project.

Lisa Brown was the designer and project lead for Slow Down, Bull, having previously been a designer on Resistance 3 and Sunset Overdrive. She has since made the indie plunge and is off having wild indie adventures, mostly involving streaming game development.