GAMASUTRA
The Art & Business of Making Games

PRINT

Postmortem: Gaijin Games' *BIT.TRIP BEAT*

By Alex Neuse

[In a postmortem of the acclaimed WiiWare title, Gaijin Games' Alex Neuse looks at the creation of abstract action game BIT.TRIP BEAT, detailing what went right and wrong in the making of the original WiiWare title.]

BIT.TRIP BEAT was Gaijin Games' debut title, coming out less than a year after the team was assembled. At a breakneck pace, we tackled all the things that go along with setting up a company, designing and developing a new IP, finding and working with a publisher, and releasing a game.

While the three Gaijin team members had worked together before, we had never tried anything quite like this. Alex had tried to start Gaijin Games years ago after leaving LucasArts, but it never took off. Mike had been part of many small start-ups, but none had turned into his "home". And Chris was eager to get his feet wet as an independent developer, bringing a passion for process and efficiency to the team.

This postmortem allows you to peek behind the curtain a little bit to see how it all worked out.

What Went Right

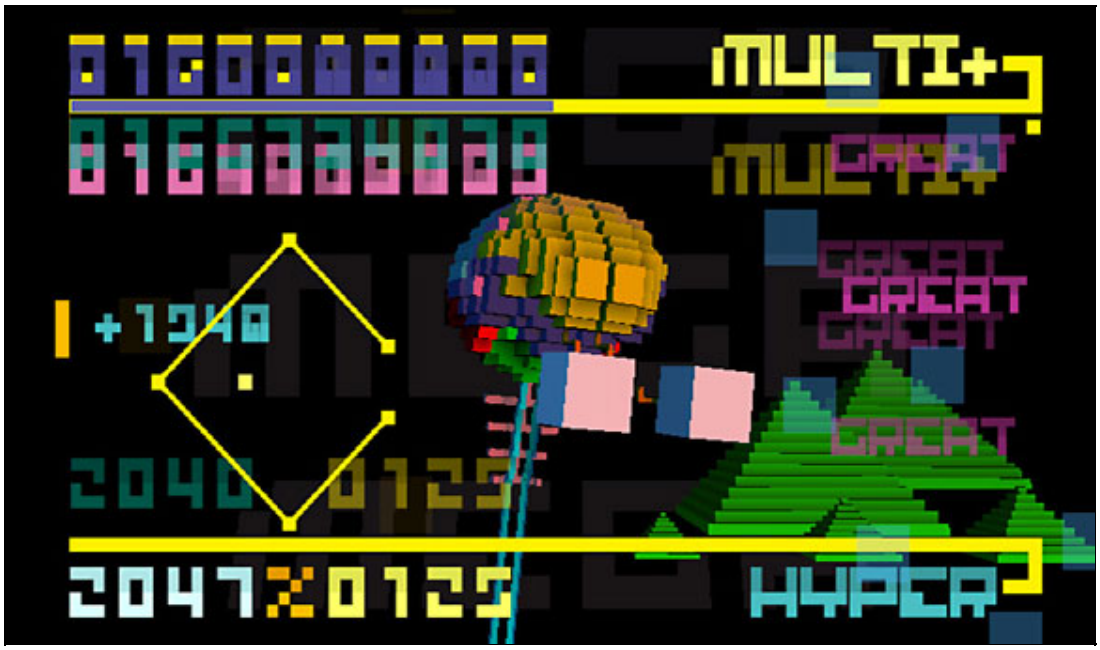
1. We set gameplay and controls as our foundation, letting the game tell us what platform to release on.

I've heard it said that American/Western developers work from the TV out, and that Japanese/Eastern developers work from the controller in. As our company name implies, we have unlimited respect and admiration for Japanese developers, even though we recognize what we are -- gaijin. However, that did not deter us from trying to work from the controller in.

I knew that I wanted to make a series of classic-inspired games and as a big fan of "paddle" games, I knew where to start. However, lacking a spinner controller (insert rant here), there really isn't a good way to control a paddle game. I've played countless paddle games with D-Pads and Analog Sticks, each being an exercise in frustration.

In our prototyping phase, we had the bright idea to try out the Wii Remote's Motion Sensor. The moment we controlled the paddle using the Wii Remote, it was crystal clear what platform we needed to develop this game for.

Working from the controller in paid off, because if the player can't control the game well, what's the point?



2. We stayed in our comfort zone.

Having just come off of a Nintendo DS project, and having worked closely with Nintendo in the past, the decision we made with the control scheme reinforced our desire to minimize risk and work with systems that we already knew well.

Migrating our knowledge-base from the DS to the Wii was a lot easier than it would have been to switch to the 360 or PS3. The tools, SDK, and relationships we'd been working with were all familiar and easy for us to ramp up on.

Because of this "comfort zone" familiarity, we were up and running on our target platform about a week after starting pre-production. This also helped at the end of the project, having gone through Nintendo's Lot Check numerous times before.

From beginning to end, the comfort zone was a huge contributing factor to the game's success.

---

3. We chose a publishing partner who worked well with us.

Another way we stayed in our comfort zone was by choosing to work with a publisher. Of course, we wouldn't see the financial returns or the fame that we might otherwise see were we to self-publish, but it was the business model that we were most familiar with and one of our goals was to get a game out fast to start building our brand. Working with a publisher fulfilled that goal.

Our publisher, Aksys Games, was one of the best publishers I've ever worked with. Above all else, their team is nice. But beyond that simple friendship aspect, they had no interest in owning our IP or tinkering with our product during the course of development.

The final product that you play is the exact game we set out to make. A lot of publishers like to tell independent developers what to do and what not to do, but we didn't run into that with Aksys. And whatever price we paid by working with a publisher was worth it.

Most developers view publishers as sinister entities that only want to string developers along, never quite allowing them to hit success and keeping them coming back for more advances that will never be paid off. Not ours. I got the sense from Aksys that they truly wanted us to succeed.

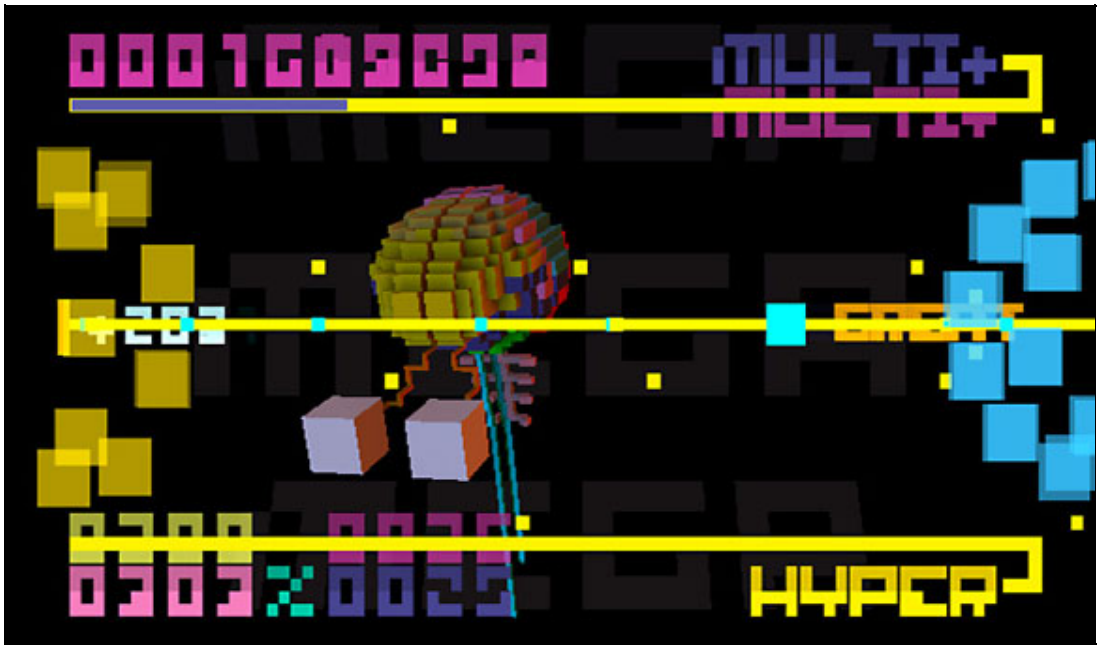
Our publishing partner gave us the reins and said "We trust you to make us all a good game."

4. We designed and developed a game within our scope and budget.

Even though we were very happy with the publishing deal we struck, the budget was more modest than we'd wanted, and the timeframe shorter than desired. However, because our publisher believed in us, and because we believed in ourselves, we decided to move forward and embraced the challenge of designing a game to fit the schedule/budget.

In the budget/schedule/scope triangle, they say that in working with a publisher, you usually can only have control of one of those aspects. While this may or may not be true, we took control of the scope and fit it to the budget and schedule. It was challenging at times, and features were cut, but we still told the story we wanted to tell and made a great game entirely within our means.

We had four months to make *BIT.TRIP BEAT* from pre-production to final, and we came in practically right on time.



5. We played our own game a lot.

Holy crap, do I like playing games. I play games a LOT. Over the course of the project, I played *BIT.TRIP BEAT* over and over again. As the designer, that's expected, of course (or it should be). But the other team members also played the game a lot.

I've been involved with many teams over the years and trying to get the team to play the game can sometimes be a headache. But on *BEAT*, we all made a concerted effort to keep our eye on the quality of our product.

Nearly every Friday, we would sit down and do an audit of the game, tearing it apart discipline by discipline. We'd take the feedback back to our individual workstations and spit and polish as much as possible.

Because we played the game often, we never lost sight of what its merits (and shortcomings) were.

What Went Wrong

1. We got lost in do, do, do mode.

Since we had strong time constraints and since we're only three men, we each had to do more than an engineer, artist, or designer would normally do on a project. This led to what I call "do, do, do mode". Basically, you have to keep doing things and keep moving because if you stop to think for even a moment, you can sometimes throw your entire groove off.

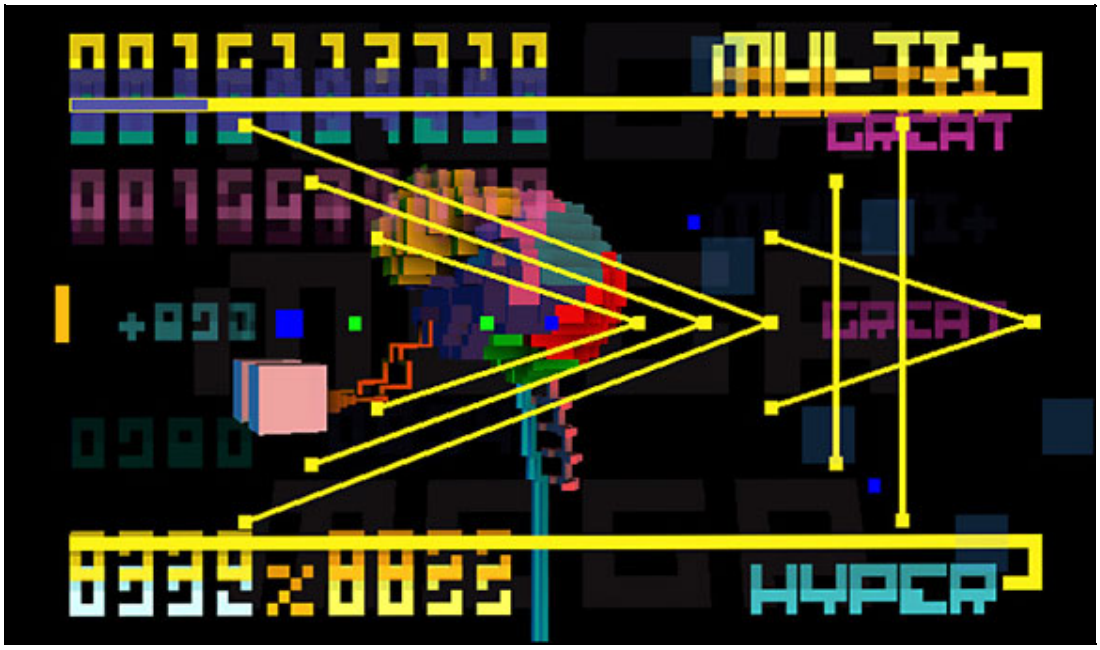
Obviously, it's best to stop, take a breath, and remember that doing something right is ALWAYS better than doing something wrong. But in the heat of the moment it's easy to lose track of the gray area between right and wrong -- because there is one. The "right" way is not always the right way to do something. And sometimes the "wrong" way can be more right than we would ever think.

That being said, sometimes the wrong way is actually the wrong way. One example of how do, do, do mode threw us for a loop was how we moved the player through level two. Because of our camera system, we couldn't animate the camera through the level (or so we thought).

So Mike, our artist, who is not an animator, ended up rigging the entire level as a character and animated it around the camera. This was insanity. A day task of animating the camera through the level became a week task of animating the level around the camera.

After the project was done, we all realized that making a 10 minute change code-side could have allowed us to do this the right way, but since we were pushing so hard and not pausing to consider all options, we unnecessarily gave ourselves more work.

Learning when to "do" and when to "don't" would have helped us to keep the project's velocity higher.



2. Despite trying very hard not to, we fell into old habits, such as crunch.

When we started this project, we decided to do whatever we could to avoid crunch mode, the plague of the video game industry.

Yeah, that didn't go so well.

About two-thirds of the way through the project, we realized just how much work it takes to run a company as well as make a game -- especially if you're making your first game. The newness of everything compounded upon itself and overburdened each of us. Just to stay on top of everything, we found that we had to work more hours. And then more hours. And then more.

Contributing factors to the crunch included do, do, do mode, running a new company, not having a producer, learning the new toolsets, setting up our development environment... The list goes on. Each of these things is necessary for a development studio to run but they have little to do with one specific product. Finding the time to juggle all these responsibilities led to crunch.

Is the project better because we crunched? In this case, I think it is. But our quality of life certainly was not. And a poor quality of life can lead to resentment of one's job among other negative side effects.

---

3. We completely forgot about localization.

Because of the timeframe, we decided early on that we would do no in-game localization. It was not required by the regions that we released in, and as such, we saw the feature as a time-saving tool so long as it was cut. And cut it we did.

However, after completely neglecting any sort of localization pipeline whatsoever, towards the end of the project, as we started doing our Lot Check audit, we realized that error messages needed to be localized.

These are the messages that tell you that the save data is corrupt, etc. This threw a huge monkey wrench into our process, because early on when we decided not to localize anything, we chose a font that had no special characters in it. Furthermore, our engine had no localization support in it.

In the end, we used a different font for the error messages and crammed a localization pipeline into the game and rushed the error messages. Had we thought of this at the beginning, our error messages could have used our in-game font, and since we have so little text in the game anyway, we probably would have just localized the whole thing from the get go.

Forgetting about localization until the very end of the project somewhat diminished the game's accessibility worldwide due to all in-game text being in English only. It also delayed our project by about one week.

4. Our amateur marketing efforts, while decent, petered out prematurely.

While technically not a development issue, this is one of the things that I believe we did very poorly with in terms of timing, and it may have affected our sales in the end.

Being that we had no marketing money and no time to do a "real" marketing campaign, we tackled something within our means and launched a viral marketing campaign. The video was well-received and led to a lot of internet buzz, as did thesubsequent images and our CommanderVideo website. However, for one reason or another, the game's release didn't follow close enough to the high of that buzz.

Our viral campaign died down well before the game was released and people shifted their attention prematurely.

5. Our bosses were a complete afterthought.

In the original design, there were no boss battles. Why there weren't seems silly in retrospect, because as a gamer I love boss battles. Regardless, I chose not to include them in the design.

The more we played the game, the more we realized that we needed something. However, this realization came quite late in the project, after all of our other tasks had already been scheduled.

And due to the small scope of the game, as acting producer, I didn't want to cut anything in the schedule to make time for the new work we'd have to do to implement bosses. So, we piled them right on top of the crunch that we were already doing.

Because we were so far along in the development process, each boss had to be done in roughly two to three days. As you can imagine, that doesn't lead to well thought out design, nor does it lead to enough tuning or difficulty balancing.

In the end, the bosses were rushed, and while I think they ultimately add value to the game, they definitely feel rushed when I play it.

What We Learned

Over the course of *BIT.TRIP BEAT*'s development cycle, we learned our limitations, above all else. We learned how much work we can actually handle -- or rather, how much is too much.

We learned to try harder as a team to find the right solutions to development problems and recognize better when doing something the wrong way is okay.

We also grew stronger in our determination that working with people you like and trust (whether it's in the office or with your business partners) is invaluable.

For the sequel, *BIT.TRIP CORE*, we've already done much better than we did on *BEAT*, but we have also fallen into some of the same traps.

All in all, though, I believe that there is value in both success and failure and I have faith that each game we make will be at least as good as the last, and I hope better.

Game Data

Publisher: Aksys Games

Release Date: March 16, 2009

Developer: Gaijin Games

Platform: WiiWare

Return to the full version of this article
Copyright © 2016 UBM Tech, All rights reserved