GAMASUTRA
The Art & Business of Making Games

PRINT

Postmortem: Q-Games' *Pixeljunk 4am*

By Rowan Parker

A reprint from the September 2012 issue of Gamasutra's sister publication Game Developer magazine, this article is a postmortem of Q-Games' IGF-nominated interactive music title, PixelJunk 4AM [YouTube trailer]. It includes new commentary from the developers especially for this Gamasutra version.

You can subscribe to the print or digital edition at GDMag's subscription page, download the Game Developer iOS app to subscribe or buy individual issues from your iOS device, or purchase individual digital issues from our store.

At the end of 2010, right after finishing *PixelJunk Shooter 2,* Q-Games president and founder Dylan Cuthbert pulls me aside for a chat.

"So we've kind of got this music visualizer using the PlayStation Move called *lifelike* on the back burner," he says. "You should make it happen."

I asked for some more details about the project. "Well, there's music," Dylan said, "And there's a PlayStation Move. Off you go."

PixelJunk 4am released in spring 2012 on PSN. It's not so much a *game* in the strictest sense of the word -- it's a Move-exclusive audiovisual composer, where all your performances are broadcast live around the world on PSN.

Music is created using the Virtual Audio Canvas, which is an actual physical 3D space carved out in front of the player. It contains more than 190 sound samples, a wide variety of DSP effects, and the ability to dub loops into your own unique groove. We also released a free Live Viewer, allowing anyone on PSN to stream performances live and give real-time feedback.

It's easy for a game designer to say, "Hmm, this boss is still a little easy. It should breathe more magma!" Knowing when a *4am* event is finished, on the other hand, starts venturing into the realm of music production -- and there were no other similar games to use as points of reference. The control scheme is completely unique to *4am,* and the experimental social gameplay we included also seemed pretty far-fetched at the time when we were developing.

Normally, those are the points at which someone high up usually says, "Stop smoking so much and make something solid!" Yet throughout development, Dylan supported every new crazy idea by saying, "If it's fun, put it in!" Our U.S. publisher Sony Santa Monica was also super supportive despite the amount of wild new design that *4am* was pushing. The faith (and massive balls of steel!) of these fine people is what ultimately allowed us to release a unique experience that will hopefully be remembered fondly for a long time by its players.

What Went Right

1. Fearless Hardware Experimentation

Going into the project, I didn't exactly have a glowing perception of the Move -- it had always felt gimmicky to me. On the first day, though, we played around with some of the SDK samples and were surprised to discover that the Move actually seemed pretty robust and ripe for creating some wild stuff with tricks that neither the Kinect nor the Wii could do. (It occurred to me at the time that it might not be the Move's fault that it wasn't being used for crazy new stuff, but that people weren't making games for it to do crazy new stuff; since we started on 4am, we've seen some other notable Move experiments, which warms my heart greatly!).

With a skeleton team of two programmers and one designer, we prototyped at least 12 different control methods for *4am*, all utilizing the Move to control music in space. Some of the control schemes varied from casting musical "spells" in the air to replicating an eight-way arcade stick and inputting *Street Fighter* commands in the air.

Regardless of how ludicrous each idea seemed, though, the only test that mattered was whether we could ask, "Can I make music, and is it fun?" and honestly answer yes. Dylan supported our willingness to experiment without regard for whether something felt "normal" or "gimmicky," all the way up until we found *4am's* distinctive Virtual Audio Canvas. Without that support to continue experimenting, we might never have pushed the Move enough to find the Virtual Audio Canvas we have now. It would have been a hell of a lot easier to just implement a menu and pointers, but it wouldn't have been *4am*!

2. Discarding Old Tools

When I picked up the *lifelike* project in January, there were already legacy tools and an editor. Unfortunately, this editor was largely unusable (I'll explain why later) and we needed to make a quick decision regarding how to move forward. It was primarily developed and tested on a PC, and the PS3 version wasn't optimized.

GameMonkey (our scripting language) was running computationally intensive generic behaviors with a vast array of tweakable properties. The framework gave us flexibility, but came at the cost of unoptimized generic code that was incurring a lot of processing time. Even though the artists and designers were only able to build rather generic-looking visualizers, our frame rate was still suffering.

Shortly after the controls prototype was finished, we scrapped all the generic behaviors and started rewriting the rendering code with specific visuals in mind. It gave us the freedom to optimize each visualizer independently (scalpel over sledgehammer) and expose only the most interesting elements to the artists. Ditching support for the PC build also let us offload a lot onto the PS3's SPUs, and freed our lone visualizer programmer from the burden of maintaining both versions of the editor. This boosted our frame rate and made the artists feel more empowered to make cool stuff.

We also sat the visualizer programmer right by the relevant artists, in order to quickly iterate the visualizer as they figured out what the artists needed to achieve the look they wanted. This paired-nucleus approach was definitely smoother than a programmer trying to write blanket variables to achieve an invisible goal.

---

3. Renaming from *lifelike* to *4am*

Around July 2011, we realized we had a completely different beast than we'd started with. There was an internal and external need to redefine the project and leave behind any associations to the old one. We couldn't have known how much of an effect renaming to *4am* would have, though. In the initial meetings a lot of names were thrown around that came very close (*PixelJunk: Acid*, *PixelJunk: Groove*) but once it was said, *4am* caught on almost instantly. It encapsulated the identity that we in the office had all come to attribute to the game, and it was the only one we could agree on that would convey that identity to people outside as well.

4am best captured the atmosphere of a club in the early hours of the morning. The name change didn't just throw off the shackles of old design ideas and tech for us; it gave everyone a common word to congregate around. It's also just plain fun to say.

4. A Radio Station on PSN

4am broadcasts all performances live around the world using the PlayStation Network. There is also a Free Viewer available for download, which allows anyone to stream current live performances even if they don't own the game itself. It is essentially a free Visualizer-powered radio station available to anyone with a PS3, where the performers create the content.

Needless to say, this was some uncharted territory both for us and for Sony. Sony's Title User Storage (TUS) service came to the rescue, helping us avoid P2P networking by storing a buffer of each broadcast per user on their servers.

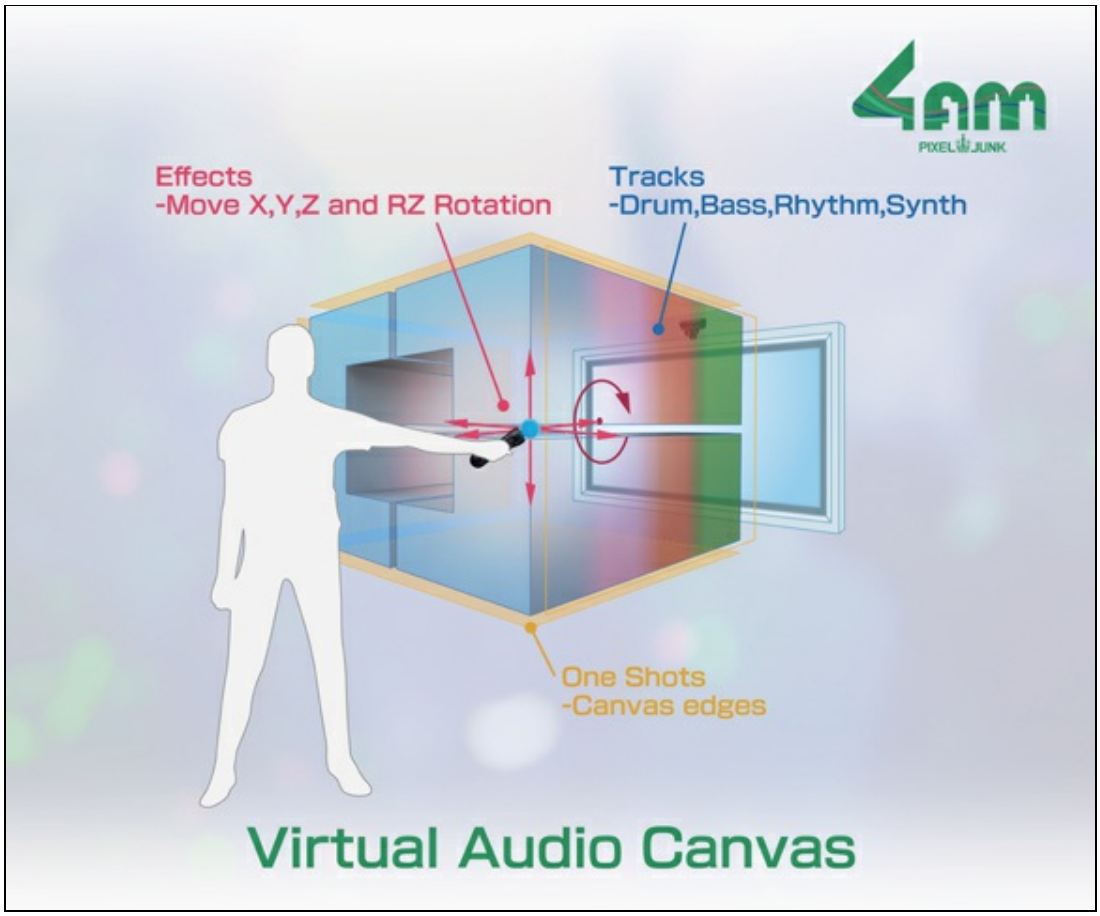
Unlike regular streaming services (UStream, for example), we wanted *4am* to be live *without interruption*, so we couldn't allow for any buffer latency to occur. This meant that the data sent across the network needed to be heavily packed.

4am currently creates an initial one-time-only buffer on load, so if the network connection drops packets during a performer's playback, the buffer shrinks permanently. Fortunately, we managed to keep the network data packet size to a minimum, which meant *4am* has an almost-

negligible rate of packet loss.

The Free Viewer contains all the same assets as the full version, so we only have to send the Move's motion data (gyro and accelerometer) to effectively call all the same functions on the local client, meaning that viewers' PS3s are actually replaying the performer's Move motion data in order to replicate the performance, not simply streaming a video.

We wanted *4am* to be viewable to all PS3 owners, with thousands of people concurrently streaming live performances, and a P2P network setup simply would not have been able to handle that kind of load. It's also nice to know that our unorthodox use of the TUS didn't overload Sony's servers.



5. Support Your Stars

PixelJunk 4am was designed to encourage "performance" in every sense of the word, but we only realized what we'd really made once the live beta started. The live beta was an online stage where players could perform. As people started playing with the live beta, they began to post videos and personal streams of themselves performing in *4am* to build their own followings.

We were humbled to see that players were already enjoying themselves with just the featureless beta, and once we released it, the social effect was compounded. One feature shining through is the auto-shuffle attract mode: While on the main menu, *4am* will seek current live streams in the background and put up a random person playing live as a menu background.

This attract mode also adds to the head count for that performer's crowd. A stable of other features (Facebook, Twitter, hometown, local time, avatar display) were also designed to let players stamp their own identity on an event; identity really lets performers make the *4am* events their own. Since we've launched and gone live, some performers are amassing crowds with thousands of people in the audience. Most people will go their whole lives and never be able to play in front of a live audience of that capacity. I'm thrilled people are experiencing that.

What Went Wrong

1. Lack of Early Direction

In 2010, when the project was still called*lifelike,* it was crippled by a confused artistic direction as to where it should go. If we had a strong artistic vision early on, we could have avoided building features we would end up not using, and we would have built tools that let us get the visual effect we wanted. Since we didn't have that artistic vision, we had to build very generalized tools that produced a rather generic visual style.

In an attempt to make the editor more artist-friendly, we used GameMonkey to expose *alot* of in-game models' behaviors. Unfortunately, we

exposed too many parameters and variables, which took down the frame rate. A great example was the jump behavior designed to make small circles jump in time with the music.

Over time, the jump behavior became bloated with a variety of other requests like interpolating color based on height, rotating on a pivot point, and other small artistic touches. If it so happened that none of them were being used, the cost of one jumping circle could be astronomical for the visual payoff. This was the price though of wanting flexibility in the visuals but never nailing down any one element.

Many of these behaviors were things that could have been (and eventually were) written far more efficiently and optimized for the SPU once we knew exactly what we wanted to make.

As a result, early incarnations of the editor on PC were a monstrosity, making it almost impossible to use without an intricate knowledge of each behavior's roots in code. (The original programmer moved on shortly after the prototype, so there are probably still things lurking in there that no one will ever know about now.)

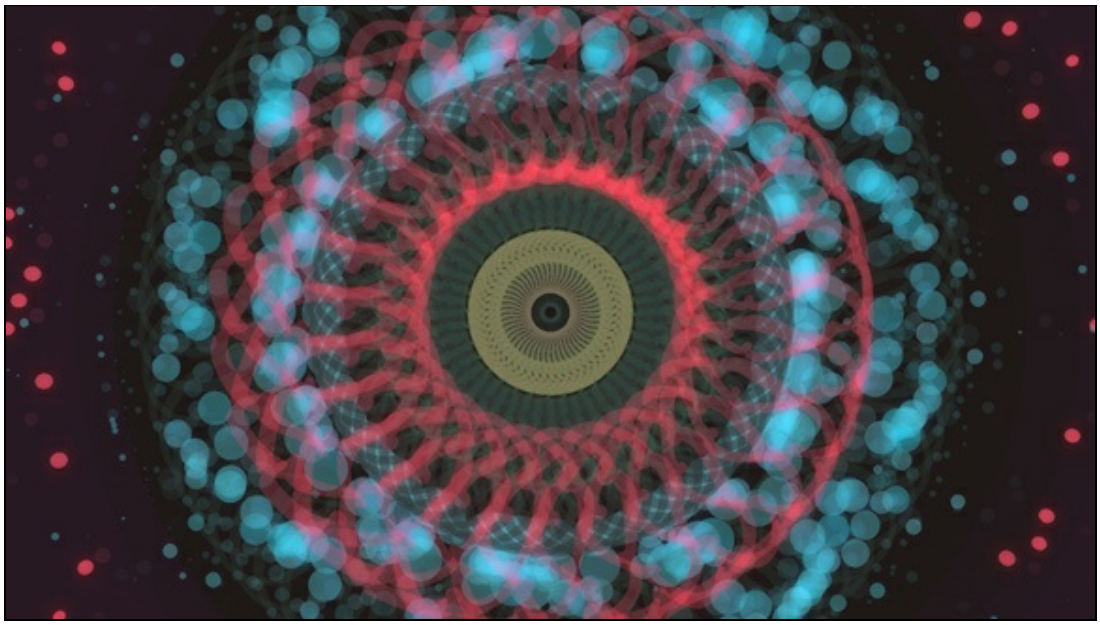
The GameMonkey behavior method of using the editor was largely tossed out following the prototype, essentially binning a vast amount of resources and time.

2. Text Tutorial

We knew that *4am* had so many new gameplay concepts and controls that it was going to be hard to explain how to play it. At game shows and events, we found people picked it up extremely easily, given the tactile nature of the Virtual Audio Canvas (vibration feedback).

Explaining this idea with just text and images, though, was a completely different story. Even after repeatedly refining the current tutorial and running "blind tests" on new people with every iteration, we still haven't really achieved a 100% understanding rate for the controls after one tutorial play-though. *4am* introduces many new control concepts with no easily available mental reference points -- there's nothing like it we can compare it to -- so it's kind of like asking players to read a book and then play a saxophone.

The player can really only learn how to control *4am* through play and experimentation, not a tutorial -- just like a regular musical instrument. However, since we call it a "game tutorial," people expect to understand everything about the game after they're done with it. While it's nice to see many players just "getting in there" and quickly picking up the rest of the controls, it would still be nicer to have a gentler tutorial experience that gives people time to acclimatize first.



3. Artist vs. Player

4am began as a passive music visualizer. When it started coming together as more of a music creation tool, a fundamental conflict arose: Players want to manipulate the music, but the artists want to preserve their original idealized state of a track. When we began experimenting with DSP effects, we quickly found that people like to warp music and make it their own. The dichotomy within *4am* is that we want players to find their own sound, where traditionally produced music aims to have already achieved that for you by the time you hear it. So when making tracks for *4am*, should they be prepped for "finishing" inside *4am* by the players using DSP effects, or should they be ready to go right out of the box?

This internal tension had a direct impact on the ranges available to players within the DSP effects. A flanger or chorus DSP might have a wide frequency or feedback range to play within, but by carefully setting a min/max range under the hood, we could safely trim the performer's freedom space to something we knew would produce the kind of sound we wanted.

Unfortunately, some of these ranges are probably still too subtle for the average user to enjoy, since they can be difficult to hear depending on what track is playing. We had to create effects that were audible and interesting to players, but at the same time stylistically acceptable to the artist. In hindsight, these are two completely opposed ideas, so it was always going to be difficult to satisfy both. The player still needs to have fun, but not at the cost of being a passive appreciation tool for the artist's benefit.

---

4. Communicating *4am* to the Media

4am is not a traditional "game." We probably took too long to realize ourselves that it is more of an instrument than a game, and so we slipped into our regular PR habits. The mistake we made was expecting the same response as usual from the traditional game media outlets: Many review sites and journalists clearly didn't know how to approach the title, and either chose to not review because it didn't fall within their definition of a game, or reached in with a lukewarm reviewing hand without really wanting to shake the boat.

This was in stark contrast to the sharp love/hate response we've seen in comments and forums from people. It's validating to know we made something that generates a passionate reaction from traditional gamers and challenges their definition of a game.

Those players who do love *4am* have continued to perpetuate it through word of mouth and online user reviews, which has been a great boon. However, we definitely needed to act earlier and reach out in more lateral PR directions to audio- and music-production-specific sources.

We've also been showing *4am* at a variety of music festivals this year, such as Coachella, Electric Daisy Carnival, and very soon at Lollapalooza, giving festival-goers a cool chill-out option.

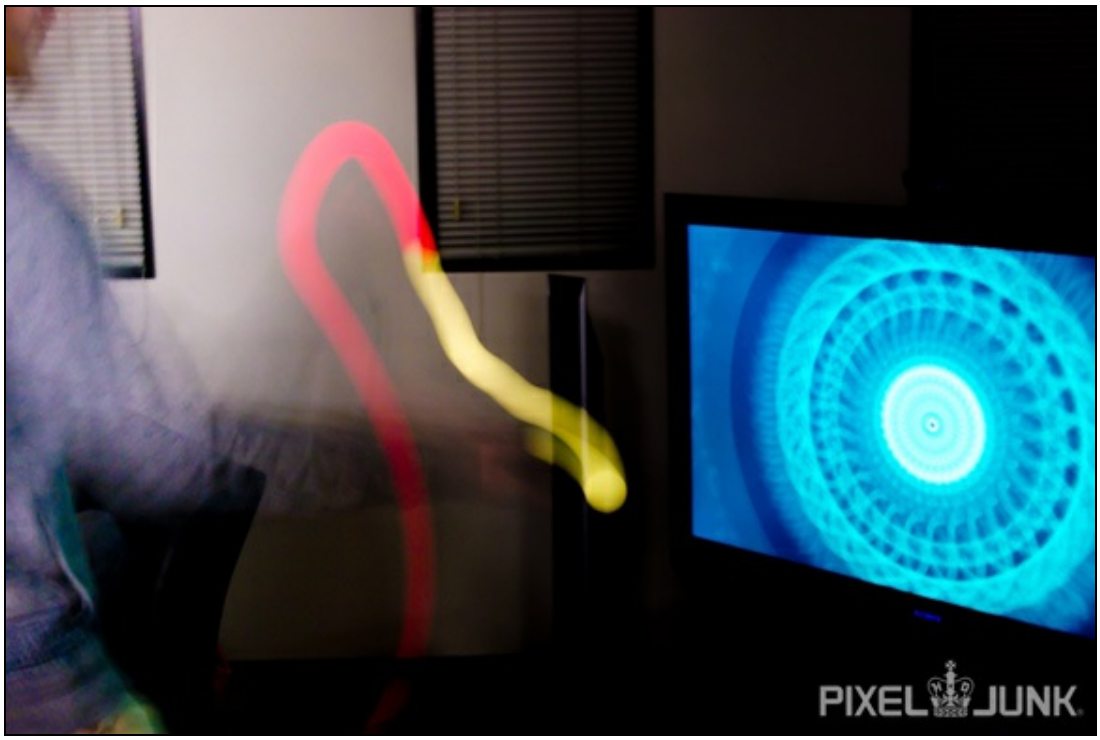
5. Managing Beta Expectations

There was a degree of miscommunication surrounding the beta for *4am*. We were the first beta to have two packages (a Free Viewer and a Full Version), which made it tricky before we'd even started. Of course, both were critical for testing purposes, but the main problem was people mistaking the Free Viewer as a demo and expecting gameplay. In the context of the full title, the Free Viewer is great and has had a huge uptake, but in a climate where betas are free anyway, that value was lost and people expected playable content.

From a developer standpoint, we wanted to use the beta to test the PSN broadcasting system in a semicontrolled environment and get some player feedback to make some last-minute changes. Prior to the beta, the only broadcasting tests we'd been able to perform were between internal QA with Sony, so we didn't know how *4am* and PSN would handle having thousands of people download the Free Viewer and watch.

The great uptake meant we fulfilled these two goals, but I feel that some people may have been burned on their first impression through the beta. The official beta announcement was a combined effort with our publisher Sony and our own official channels, but ultimately we should have taken more ownership of the message and ensured that word got out about how the beta was going to work (by baking an informative screen right into the package, for example).

Unfortunately, with the small team size and time restraints, we really couldn't give the beta the time it deserved. Overall, the beta was successful in satisfying our development requirements, but from a marketing perspective, I feel there may be a few players out there who might never give *4am* a second chance.



Late into the Morning

If someone had asked me to sit down and seriously design a new musical instrument in January 2011, I would have been enthusiastic. I also would have had no idea where to start. (Now, I would probably clutch my knees and rock back and forth underneath my desk.) Yet at the end of our musical-space journey, we now have a new electronic instrument and performance-art platform that people are enjoying online.

I feel amazingly fortunate to have landed in something that was backed by so much faith, and I am grateful that we were given the time we needed to make *4am* and not some corner-cutting "missed opportunity." Since then, we've been working in secret fervor on a new *PixelJunk* IP that will see a return to more traditional gaming roots (but still has that unique *PixelJunk* twist).

Update, April 2013: What's Happened Since

It's almost a year since we released *PixelJunk 4am* and in that time we've learned a lot about player tendencies, the pleasure and pitfalls of pioneering a new kind of game-performance experience, and some lessons along the way. The upside to all the hard work that went into *PixelJunk 4am*, however, has manifested in an 'Excellence in Audio' nomination at the Independent Games Festival during GDC, so it's great to see *4am* recognized for its technical achievements.

Since the description of the category is a little vague, it's difficult to ascertain whether the Excellence in Audio nomination is based on the quality of one's game soundtrack or if it's given to a developer based on the technical merits of a game's audio design. Hopefully it's the latter, because there are very few games out there using technology like *4am*'s positional audio engine.

Ironically, one of the most popular elements of *PixelJunk 4am* -- the multitude of music visualizers that accompany each stage of the game-- has little to do with the audio design at all. These have proven so popular, in fact, that we've created an entirely separate app for PS3, called Visualizer, for people who don't possess at PS Move or PS Eye.

This way, people who've wanted to enjoy the eye-catching visuals of *4am*, but didn't have the means to do so, can now enjoy this music app and use it with their own PlayStation 3 music library. Many people don't know that Q-Games developed the original visualizers for the PS3, along with other things like elements of the XMB. In a way, we're getting back to our 'roots' by releasing this new set of visualizers just as the PS3 makes way for it's new sibling.

PixelJunk 4am has similar origins; the project started off with us testing tech for the algorithms and DSPs used in the visualizers. It's fitting that both *4am* and Q-Games should return to their PS3 origins, full circle.

Data Box

Developer: Q-Games
Publisher: Sony Santa Monica
Release Date: May 15, 2012
Platforms: PSN
Initial Number of Developers: 3
Final Number of Developers: 6 + art and music by Baiyon
Length of Development: 16 months
Development Tools: C++, GameMonkey
Hallucinogens consumed: Minimal

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