Dream Machine is an always-on, procedurally-generated beats machine

REAKTOR Library hero James Peck, aka memorostic, and Daniel Bereen of Space Agency explain how they brought this unique audio-visual project to life.



James Peck should be a name familiar to both REAKTOR heads and hip-hop fans alike. Whether creating ensembles such as the wildly popular – and free – **SYNS Audio Organization Suite** (still his most downloaded ensemble of all time), or producing music, sample packs and the MPC gems for his Youtube channel as memorostic, he's been a key part of the REAKTOR community for a long time. He nevertheless maintains a strong human element. Dream Machine feels like the logical culmination of his talents, and fresh from the launch of Dream Machine No. 01, we sat down with James and Daniel to discuss the ins and outs of this Guinness World Record-sweeping project.

This is a pre-recorded clip, but you can check out the current live stream at dreammachine.live.

Before we get into the specifics of Dream Machine, could we hear a little about who you both are and what you do?

James: I'm James Peck aka memorostic. I'm a producer, songwriter, sound designer, maker/tester/consumer, a whole bunch of stuff. Most recently I've embarked on a project called Dream Machine, which is a 24/7 procedurally-generated radio station.

Dan: I'm Dan, a buddy of James. We met back in high school. My background is in creative direction, and I run a studio called Space Agency. James showed me the engine he was building in Ableton (Live), and all the cool characters he was doing, and we basically sat around and put together this Dream Machine thing. I help out with the planning, integrations, collateral materials, socials, partnerships and strategy, as well as working with James on new versions of ideas.

Where did the initial idea for Dream Machine come from?

James: It started with the concept of being able to procedurally generate beats. I was making another project a couple years ago called **Twitch Plays The Synth**, it's a custom Reaktor synthesizer you could control with Twitch chat commands. Like, you'd type "!play" and it would play a note. I'd also built a drum machine, took a bunch of samples and drum loops, put everything in the same key, and randomized them. From there it turned into something that doesn't use samples at all. We're generating MIDI notes, generating the mix elements, all the production is happening in a similar fashion. I was also doing 3D modelling at the time, and I was making these little creatures.

Dan: Like evil Crazy Bones.

James: I was putting them on Instagram for fun, and Dan's like, "Yo, you should combine your 3D dudes with your loFi generated thing." Next thing you know Dream Machine is born. I had to figure out the visuals to the music in the stream as well, so I had to invent Unreal Engine, make Ableton talk to Unreal Engine, and then get that streaming 24/7. It turned into this massive project, it was crazy. Every two to three months, there's a new look and a new visual. But it all starts with the music, and that's 100% Abla, Nav for Live devices, Native Instruments plugins, all sorts of stuff.



What did getting Dan on board bring to the project?

James: The audio engine and the Unreal Engine, Dan married those two and brought the concept to life. A project like this, you need other people involved, and Dan's been the other guy planning what to do with it, because there's a question at the end of the day, when you're up against hours and hours of streaming, "well what are you going to do with it?" Dan's finding creative ways to bring Dream Machine not only to people on the internet where it lives, but to things in real life.

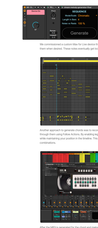
Could you go through how Dream Machine is put together and how each of its constituent parts are talking to each other?

James: On the front-end: I do like have a little Mini that runs Ableton, and beside it is a gaming PC with an Nvidia graphics card, and that has an audio interface, a Steinberg UR22 MK2, Ableton sends audio and MIDI from the Abla Mini out from one of the interfaces, and the other interface is connected to the PC where all the audio goes into OBS and Unreal Engine, which is the Unreal Engine Engine. So that's how they're connected.

In terms of the software that's running and how it works, it's essentially just an Ableton Live session. Everything happens in Ableton. From the MIDI clip generation, to the samples that are loaded in, to the plugins that we map or mix and live using.



Ableton's workhorse is Live's Drum Rack, which you'll find in Dream Machine's Drum Kit Library. It's also in each of the system glue loaded.



Some of the MIDI data that drives Dream Machine. You can think of this as a menu of sorts.



This is the kit we've been using for the Dream Machine stream, but we've got around 20 different presets set up! Feed it any seed and a superb musical vibe will be produced, and always somewhere familiar you've been before.



Here's the Kit's architecture. Each of the 16 pads has its own chain, and each chain has its own sample. These make thoroughly good choices to quickly build out ideas.



Another approach to generate chords was to record myself playing hundreds of loops and audio snippets on my MPC, and then I'd chop them up and map them to a MIDI keyboard, and then randomizing your positions in the timeline. The idea is to use interesting melodic ideas that aren't boring.



Here the MIDI is generated for Drum and Bass Kit Library. It goes through randomized processing, and then it's sent to Ableton's Drum Rack. Each pad has its own chain, and each chain has its own processing and sample selection. Rack Macros let us control Reaktor effect FX from Native Instruments' Kontrol S49.



Every pad has its own FX chain assigned to it, which is then sent to the Drum Rack's main bus. Here's the Kick pad's FX chain. You can see it's custom Reaktor instrument, kicker in the chain as effects.



A peek in the Unreal Engine shows what's up in the Ableton session. MIDI notes are sent to custom Reaktor instrument MIDI slots' instantiation, via the MIDI and Ableton, where it's sent it.

Other useful tools for processes music are the Arpeggiator, Randomizer, CV and Velocity modules.

This is one of the session tracks files.

What's been the trickiest thing to get right with this system?

James: The hardest thing to generate is rhythm. It's really easy to generate chords, it's really easy to generate melodies, it's really easy to do stuff like jumping through a bunch of different chord styles and reverb sound slightly elastic, reverbing it would sound random. There's no times where you train would pick up a chord progression or some of the more harmonic structures, but it was lacking a little bit. So the next step was to take a developer off from to help me with that, and that's when we started working with Native Instruments on it. This was a massive leap in the right direction.

There's instances of this device for different parameters to generate notes. There's also an arpeggiator that's adding more notes on top of those notes, randomizing velocity, randomizing length, and then there's randomizing pitch, velocity and note delay. Layered up all these sort of little like bit Lego devices over the course of making this that are very useful for making this work.

The only minimal tweaks because the sounds are so wide, and you can't use a multitrack with all the stuff you want. You can't send a program change to your instance of Kontrol, and it'll play a different patch you have loaded up. You can also control the number of voices and therefore control the memory limit.

Is Kontakt bouncing between patches you've made yourself?

James: I think these are all from the Native stock library. I'm not using any custom Kontakt instruments. I am using tons of my custom Reaktor instruments later in the chain as effects.

Is the audio processing happening within each channel, or do you have busses for that?

James: It's on the core tracks alone, and then it goes to the master channel, where there's a similar bunch more stuff happening there. I have my SHO plugin right at the end of the chain, and this will either get turned on or turned off depending on where it gets randomized to. So if the Filter or Warp wants to change every time there's a new beat, we have to do it's with that as a macro.

I have another type effect I made on the master channel, the MIRROR. It works in a similar way, where it'll turn on or off and there's a few parameters for different types of mirroring. Then there's a Native Instruments Supercharger glue section, and it's a nice glues with parts modulated by the kick instead of these effects is, when you're producing real music, you're making these decisions to somewhat of an intuitive way, but they're also somewhat arbitrary and random, and you're using a guiding principle of: "I'm not going to do some crazy shit, my producing going to do something like this." I'm setting ranges for all these knobs and all these controls, so nothing ever gets too wonky. Essentially, I'm trying to create a system that's kind of idiot-proofed/idiots-to-give-that-extra-character-or-imperfection-or-to.



What other Native plug-ins are you using?

James: OT, 70 is hot fire, Passes 02, these are classic recreations of very familiar effects. I have all the Native products and they're very easy to bend, they're very quick to set up, which is great. For bass I have an instance of Kontakt set up with Scarbee Bass Workstation, J Bass, and some my own Operator patches. These are just simple, loFi bass sounds and it's choosing one randomly, and then going through some effects. I dig the Massive X patches.

And what's happening in terms of AI here?

James: There's a few ways I've tried to do it. I've tested every AI tool in the book to create stuff, and settled on controlled randomness for now, until we develop a more AI-driven version. The Reaktor ensemble I made, the Twitch Plays The Synth, that generates data, it sucks. The most successful one I've used is the thing called Splash Pro, which used to be a Roblox app that people could make music with. A lot of the stuff is using neural nets that have been trained on existing MIDI, and I can produce some good stuff, but I haven't found a tool that's been the one. I've been able to make some pretty incredible Max for Live devices that Splash Pro sometimes. I was using a lot of Splash Pro to generate actual MIDI clips and then feeding that into Ableton, and then I've been using a lot of the Native Actions would go to Kraken Max. The problem is eventually you'll run out of clips in Ableton, so my goal is to eventually make a Python device or Max for Live device that generates Ableton clips.

A lot of the time this AI-generated or procedural music, the production is garbage. So I guess I'm trying to take my skill set as a somewhat decent producer, and focus more on mixing. I think Dream Machine' strength is that it's a curated set of sounds. I've gone through and selected each sound by hand and set it up so that it has all these nice possibilities to go to.

The visuals are obviously a big part of Dream Machine. How did that come about, and how do they react to the music?

Dan: Originally I don't even know if we wanted to add the whole thing in Unreal. We were just going to do a simple visualizer in Ableton, and then I was like, I know these things plugged together that we'll need a way beyond any audio responsive where now you can randomly jump between hundreds of cameras each time a beat is generated. Once you had the Unreal Engine in you can apply it in a much bigger way.

James: There's a lot of different things coming from Ableton into Unreal Engine. I have a dummy clip that sends something like, "CC22, message 127" and Unreal Engine picks that up and says, "I have to change the camera to scene number 4." That clip in Ableton has a Follow Action, so now we're using randomized Follow Actions to randomly change the camera every couple of bars. Then there's other messages, things like that. I need Unreal Engine to know when there's a new beat, because when there's a new beat the any colour is going to change, the post-addition is going to change. I find a range where I want the sun to go, and say, "every time this happens, randomise yourself within this range." And then you also kind of know at the end of the song, the chances are it's changing, you're changing all of your post-changes and you're changing your camera. So I'm trying to think of ways to sort of change with your music, you can eventually connect to Ableton with a MIDI signal. I've still thinking of ab lot to do with that.

Did you have any tips for anyone who's seen the Dream Machine project and would like to have a go themselves?

James: I'll copy from a practical standpoint, make every chord you use, just as a clip in Ableton and just set a Follow Action to "Random" or to "Other" and go from there. Build clusters of clips, like I'm Follow Actions, that's a great place to start. And then conceptually, read the Wikipedia for "Reaktor Kraken." It's essentially what I'm doing, which is compositions that can randomness as part of the composition, and then you're kind of like, I'm happy with this range, and then where you can play from whenever you want, or you given like, if this, then. Glass had a piece like that. So there's some cool reading to be done in it. And there's a lot of things right now that are exploring exactly that generation stuff as well. I would just dive into all that.

Check out the Dream Machine stream on the Dream Machine set up itself or at/on/in dreammachine.live. And be sure to check out VHS Degradation Suite and James other REAKTOR creations (MIRROR in particular is a must-have, and it's inspired processes). It's all available for free in the REAKTOR User Library.

Originally created May 8, 2021 on the Native Instruments blog using native instruments and komplete products. Migrated to the Matrix Instruments blog on September 8, 2021.
© 2021 www.native-instruments.com. All rights reserved. Unauthorized duplication is a violation of applicable laws.