

Interactive Sonic Arts
Etude 05
Kieran Maraj

Jerk-Based Triggering

One of the new behaviours I've implemented this week is a stylus-jerk based triggering of percussive content. I've attempted to map sharp changes of direction on the x-axis to playback of a simple kick drum, and quick changes of direction on the y-axis to randomized sample playback from a small collection of breaking, snapping, and shattering sounds. I spent a lot of time trying to condition the requisite signals and onset detectors properly, and I'm still not happy with the sensitivity. It often feels that gestures that, to me, feel like the exact repetitions will sometimes trigger and other times won't - and when they trigger at the moment I want and allow for the precise musical shape that I'm intending, it's the most satisfying feeling. But when it takes 2,3,4 attempts to get a sound to fire, it's incredibly frustrating. I'm definitely planning to spend some more time here to get the triggering to be more consistent.

Whenever that happens, I have a few ideas for extending this behaviour. One thing I can hear in my mind's ear is a percussive hit leading into low bassy tones. I can imagine replacing the sampled kick with a synthesized one and implement some sort of control that allows me to extend the release time of the kick so that it becomes a tail. Another approach could be to simultaneously trigger sample and low FM synth - I'm imagining a bass voice that begins with a kick trigger and then potentially takes off on its own, with its own behaviour.

Another extension I'd like to make is to pass the triggered percussion through some sort of rhythmic network of delays in order to extend gesture, extend the timeline, and also to potentially create some sort of rhythmic pattern. I'm not sure exactly how I would go about doing this, but I can imagine measuring some quality of the input gesture or mubu stored gestures in order to set delay times, similar to the multilayer perceptron to delay mapping I currently have working.

Stylus Running Memory and Recall

The recall of short-term memory gestures to one of my granular modules is something I'm really enjoying playing with. It's really satisfying to play with, especially when longer gestures are recalled and I start to recognize the shape. The implementation is quite simple, just using a pair of colls - one to store all stylus data over the last five minutes, and one to store the indexes of all no contact to contact transitions. I like how simple this implementation is - it feels elegant, but I imagine in order to develop the behaviour beyond just random recall it's going to get more complex, with storage of gesture lengths and feature extraction, as well as the ability to query against those features. I can imagine myself spending a lot of time fleshing out this behaviour.

Machine Learning Algorithms

As we've discussed, I'm making use of several different machine learning algorithms for different purposes. Over the past week I've developed a new mapping that uses a multilayer perceptron to map stylus position and features (velocity, acceleration, and jerk) to granular synthesis parameters. My intention here was to create a mapping that felt surprising, and I was hoping that the use of the stylus features would lead to unexpected and novel behaviours, but the end result feels fairly predictable, and I feel like I've gotten a sense of pretty quickly. Not to say that it's a bad mapping - I can certainly get some interesting things out of it that I wouldn't

be able to with my other mappings, but the sense of surprise I was looking for isn't really there. But, again, I can imagine myself spending a good amount of time with this trying to create other, more dynamic mappings. One idea I've had is to take a small, low resolution sampling of stylus location points (say 10 points every second) and treat these points as a geometry. The points, and/or their features like their centroid, distances, contraction, etc, could be used as input into the ML algorithm, and the moving, trailing nature of this input data might result in interesting behaviour.