

Project Progress Report

So far I have not implemented any of the code. I believe that the implementation phase shall be rather straightforward. I am more concerned with the theoretical phase which involves precipitating a conceptual framework to structure the project, and so I've been focusing my energy during these first 2 weeks on research and diagramming.

I'm trying to use this final project deadline as an incentive to develop a tool that I will actually use for my own purposes and enjoyment. My only prompt when beginning this project was essentially to make some "Rhythm Engine" to conduct a personalized system of live, interactive, gestural, expressive or generative music. Since beginning this project, my conceptual understanding of how this "Rhythm Engine" will actually work has evolved significantly.

Since Max and Ableton automate much of the actual music playing, the artist becomes more of a conductor that directs the playing of their orchestra in the same way that a developer writes code for his machine but actually does little of the labor himself. I want to leverage this power to transform generative music into a "system of conduction" that is responsive to user interaction. (As I understand it, this is the primary incentive within the algorithmic composition community.) Instead of a MIDI keyboard, for example, in which a note has a one-to-one correspondence with a user's finger gestures, this "Rhythm Engine" will use the user input to augment a composition of generative rhythms that is being automatically played by Ableton.

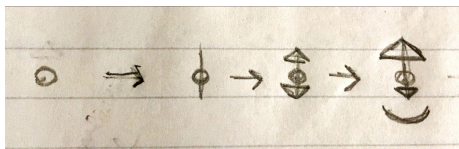
Precipitating a conceptual framework for this project involves describing a "phenomenology of music", by which I seek to abstract out the core components of music. I believe that music describes an emotional state, a state in motion, a system of emotional parts in flux and in relation to one another, a particle system with quantities and qualities. I wanted to find a way to use visuals to control sound in an audiovisual synesthesia. The Jitter package in Max is perfect for this. The Jitter package will be used to visualize a physical system that is subject to user input. The state of this physical system is what determines the music that will subsequently play.

So there will be a visuo-physical system with objects, collisions, forces and textures which is almost like a visual projection of the song itself. I'm drawing inspiration from a lot of works shown in this class, including Tom Johnson's *Rational Melodies*. Now I must answer 2 questions:

1. How will the user interface interact with the Jitter system?

The Jitter system is a constellation of objects that are either stable (at equilibrium) or unstable (far from equilibrium). An example of a stable system is one object orbiting another at regular intervals; this motion will ideally continue forever and never change. An example of an unstable system is one object falling towards another. These objects have inherent energy (potential and kinetic), so they are able to move without user input. This is part of the definition of generative music.

The user interface is a collection of operations, like *Add*, *Subtract*, *Expand*, *Constrict*, *Excite*, *Depress*, *Emphasize*, *Suspend*, *Cleave*, *Fuse* and other actions that alter the state of the system. The specifics of each operation are yet to be determined.



2. How will the Jitter system interact with Ableton?

I still need to think about this more.

The point of this project is not to make something and be done with it. Rather, it's to create a basis for an engine that is able to evolve past Music 147. The Jitter objects, user operations and musical objects are all just instances of a class, and more can always be made to make the engine more complex. This project is simply to create the infrastructure to support the connection between these 3 classes of objects, and furthermore, to enable me, the user, to generate a rhythm that can always be made more complex.