CS3220: Digital audio and applications

Assignment 2

Individual Report

Student ID: 100770005

Section 1 Features

The patch has two distinct modes of operation. First, the user is able to load two different sample files (a guitar and a piano in this case) and modify their volume as well as transition between the two sounds with the use of cross-fading. In the GUI objects colored orange, green and light blue are relevant to the first mode of operation. The second mode of operation is much more technically complex. In the GUI objects colored red are relevant to the second mode of operation. The user is able to generate two different musical patterns. The first being a more calm sound that is determined with microphone input and generates a pattern, within set parameters. The second pattern is produced through the use of a euclidean algorithm set to produce sounds with a more "aggressive" feel. The user is able to use cross-fading in order to transition between the two patterns with the help of a horizontal slider and is able to adjust the BPM with a horizontal slider as well.

Section 2 Research

The project required some special research regarding sound generation through the use of input sounds and mathematical algorithms. In particular rhythm calculations for the euclidean pattern, like those present in pd euclidrythmncalc, are attributed to Mr. Martin Brinkmann due to the fact that they were successfully implemented in Pure Data and as a replacement for a first order Markov chain pattern generator. The ambient pattern generation with the use of microphone input audio was inspired by the paper "Synthetic game audio with Puredata", by Andy James Farnell. These concepts serve as a basic introduction to the world of synthetic audio generation. According to Leonard Paul in his paper "Video Game Audio Prototyping with Pure Data", companies are increasingly growing aware of the role of sound and music in their products. One notable example is the game Spore, by EA. According to Farnell this transition form the use of different sample sounds to a more parameterized, self- generating patterns can be attributed to advantages such as:

- compact, space efficient code able to generate many types of sounds;
- flexible parameterizations makes synthetic sounds seem "alive", unlike repeated samples shuffled randomly.

According to both Farnell and Paul, Pure Data is the preferred tool to accomplish this due to the fact that it is simple to use by audio engineers not proficient in the use of languages like C/C++ and it is efficient in the process of pattern generation.

Section 3 Code

The generation of musical patterns through the use of input audio from the environment was challenging. The most challenging aspect was the detection of sound with the use of a microphone and the adc~ object. In Figure 1 this process is shown. First we use adc~ to pick up surrounding sounds with our microphone. After which we use snapshot~ which converts those sounds into values we can store with the pipe object and be sent elsewhere. These values are sent to a bank of 7 custom osc~ sub-patches, shown in Figure 2 and Figure 3.

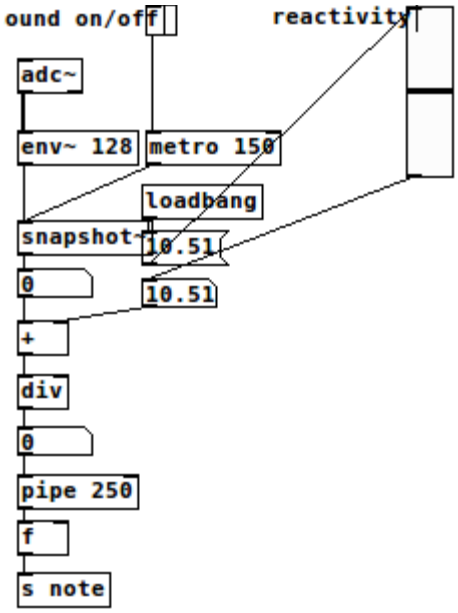

Figure 1.

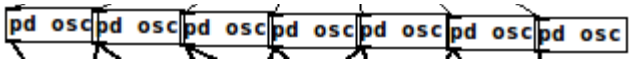

Figure 2.

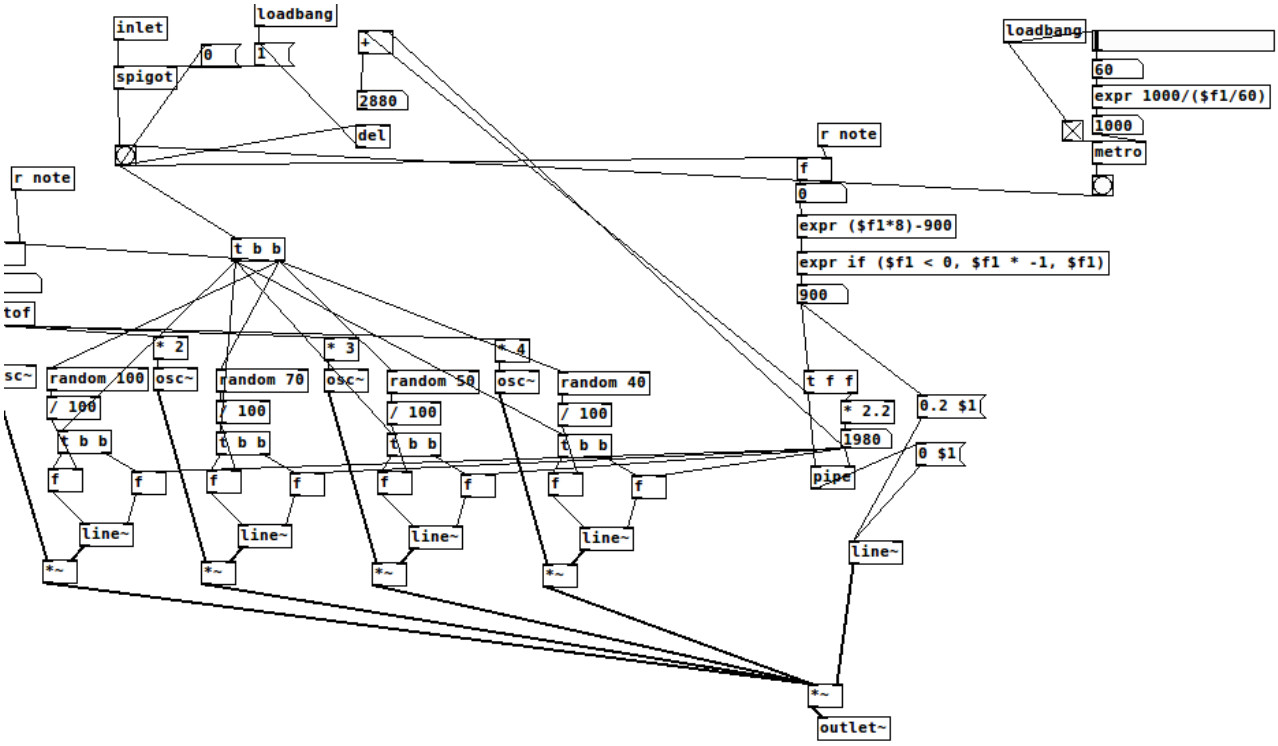

Figure 3.

Here the values we receive from the microphone pass through several osc~ objects use the random function in order to output waves of different frequencies. This is where the BPM control is located for this pattern generator as well, shown to the right above the outlet~ object.

Section 4 Conclusion

The project provided me with greater insight of the creation, implementation and use of different sound pattern generation techniques and their use in the industry by professionals in different fields, in particular the video-game industry. The most challenging concept to understand was the sound generation through the use of a euclidean algorithm. Whilst the concept is understandable theoretically its implementation is not as easy. Further areas of improvement include:

- using a much more sensitive microphone to produce better input sound for the ambient music generation;
- a deeper understanding of sound generation with the use of a euclidean algorithm;
- the implementation of a Markov chain pattern generating algorithm;
- the implementation of a cellular automata pattern generating algorithm;
- audio and sound morphing;

Bibliography

Farnell, A.J. ‘Synthetic game audio with Puredata’ .

Paul, L. ‘Video Game Audio Prototyping with Pure Data’ .