xxxxx xxxxxxxx

21m.380 Music and Technology

Florian Hollerweger

10 May 2016

# Synthesized Dice Roller Writeup

I decided to take on the task of synthesizing the sound of rolling dice. This sound is of particular interest in sound design because it is commonly used in virtual renditions of board games. In addition, its inherently stochastic nature prevents game designers from using one sample over and over again, else it would be recognized. A synthesized dice roller would allow the user to play the sound of picking up the dice, shake them for however long they like, and then throw them on the table. The program could also allow the user to use different amounts of dice, allowing for even more flexibility. For these reasons, and for general interest in attempting to model stochastic short sounds, I set out to synthesize dice rolling.

To start, I created a short recording of actual dice being picked up, shook, and rolled on a table. Listening to the recording, I picked out what sounds would be important and which effects would need to be applied to them and outlined them in a requirements specification document.

Next, going off of the requirements specification document I began to create a model. In order to model the sounds, the sounds were first analyzed by ear to see if I could pull any useful information out of them. I also analyzed them by looking at the waveform and its spectrogram. The waveform and spectrogram showed me that they were simply clicks – short bursts of noise. However, my ear told me that they weren't quite just that. Otherwise, the clinks of dice on each other and the clunk of the dice on the table would sound the same. I did some research using Andy Farnell's book *Designing Sound* and found that modelling the resonances of the material in a band passed filter bank would be a good way to shape the clicks.

Using this knowledge, I narrowed down my method selection and chose to use a mix of subtractive and additive synthesis to generate the sound of the clinks and the clunks. These methods were combined with some algorithms for stochastic behavior and for the bouncing of the dice. I implemented these methods using open source graphical programming language PureData.

Lastly, this implementation was compared to the initial recording for fine tuning. The whole process was iterated over to improve things and get the final implementation sounding like the original recording.

There were some bumps in the road. It was difficult to model the sounds involved. After toying around with the band pass filter banks for a while, I think I finally came up with sounds that sound like dice clinking in a cup and clunking on a board. I'm not happy with how the pick up phase worked out. The sliding sounds were very hard to model, and I ended up settling for quiet amplitude modulated filtered noise tuned to the same filter bank as the board. I think if I were to go back and change anything I would go back to analyze, research, and model the sound of dice sliding more thoroughly. Next I'd try to implement the sound of dice in the hand, since that is more common than in a cup. Otherwise, I'm overall happy with the patch. It was a fun project to explore sound design and PureData.

21M.380 Music and Technology: Sound Design
Spring 2016