Music Generation in ArtToMusic

Rafael De Smet

February 22, 2017

1 Music Generation

With the help of the library Beads[1], the music is generated based on the graphical analysis. This library is just a helpful tool to produce sounds. It doesn't know anything about (good) musical patterns, rhythm or harmony. The only thing it does is generate a sounds, which is determined earlier on by the graphical analysis.

1.1 Rhythm And Harmony

An integral part of music is the rhythm So I decided to use the edges of the image to determine the rhythm of the music. This is a work in progress. Later the entropy of the analysis of an image will be added to this determination.

Without a melody, there is no music. Any melody of a song is based on the rules of harmony, which notes sound good when played together, which don't? Which notes make up a chord? These kind of rules are the subject of harmony.

The program works with premade chord progressions. These are enumerations of a number of chords in a certain order which creates a melody. For example, the chord progression I-II-V-I is very familiar once you hear it. This means we play the first chord of the key we are in, then the second, then the fifth and the first one to end.

[1] http://www.beadsproject.net/

Based on how much of certain colors there are in the image we are analysing, we choose a different chord progression to work with. If there is a lot of red in the image, the program chooses the I-II-V-I progression, for instance. Other dominant colors lead to other chord progressions.

1.2 Comparison of music generation techniques

1.2.1 Beads

As mentioned before is the Beads library, developed by Ollie Bown with the support of the university of Melbourne, a very handy tool to generate music. Beads uses the concept of unit generators (UGs) as the core of the library. As described by Evan Merz, the author of *Sonifying Processing: The Beads Tutorial*, a unit generator is a "building block of the audio signal". One unit generator takes care of one function in the generation of music.

A simple example of a unit generator is a guitar players distortion pedal. The clean signal of his guitar enters the generator and a distorted version of the signal is produced. Beads works as a series of unit generators (or guitar pedals).

The Beads library has multiple of these UGs, such as an envelope filter, a gain, a waveplayer, etc. The most important UG is the AudioContext. This is the main UG where every other UG plugs into. Without this there is no music but it doesn't create any sound by itself, it just acts as a connection to the computer's hardware. The audio comes from other UGs.

Beads has two main ways of creating audio.

Samples You can load multiple samples in to use later on. These samples can be anything you wish and have to be on your machine to be able to load them in the program. Once you have found the file and loaded it in via the SamplePlayer, you can use it exactly the same as any other UG, connect it to the Gain UG to determine the volume or pass it through an EnvelopeFilter, etc.

Waves Another technique is to use audio waves. The most simple form of audio is a sine wave that produces one tone. Beads has implemented five

different kind of waves, each called a Buffer. These buffers consist of an array of values in the range [-1, 1]. Each buffer is based on a different function.

- SINE: based on the sine function
- SAW: based on the saw function
- SQUARE: based on the square function
- TRIANGLE: based on the triangle function
- NOISE: based on a random variable to determine each value in the buffer

Using the Beads library we can create our own music simply by passing the right input parameters to the different UGS to get the desired music.

1.2.2 Wolfram Tones

Desgined by the mathematician Stephen Wolfram, WolframTones is an online application to generate music. The user can choose from multiple styles of music and the application generates a new piece of music. The way this works is based on a discovery of Stephen Wolfram himself. In the early 1980s Wolfram was researching "one-dimensional cellular automata". He discovered that a set of very simple rules can create a very complex situation.

His experiments started with a row of cells, each black or white. The set of rules, determined before the execution, decide which colour every cell on the next line will get and so on. Figure 1 shows a possible set of rules. This rule makes a new cell black if either of its neighbours was black (on the row above) and makes the cell white if both its neighbours were white.

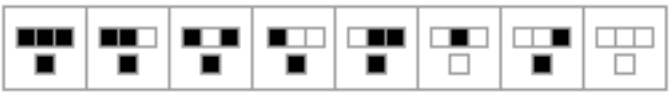

Figure 1: Rules for automata

This gives the result in figure 2. In this case the result is very well structured and organized. Wolfram discovered that there are 256 of these simple sets of rules, based on eight individual rules. Not every one of these 256 sets gives nicely structured results.

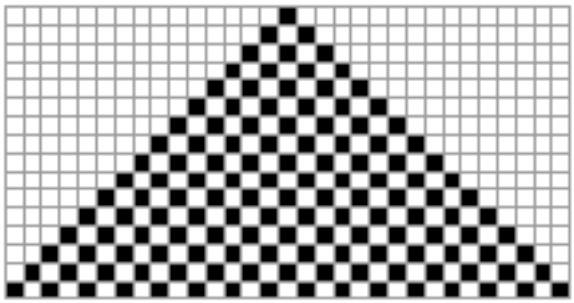

Figure 2: Result automata

Music Wolfram used his automata to create music. Let's say we used one of the 256 sets of rules to create a result pattern. We can take a swatch through this pattern of 15 cells wide. When we flip it on its side we can treat it as a musical score. Figure 3 shows the swath from a resulting pattern and figure 4 shows this swath as a musical score.

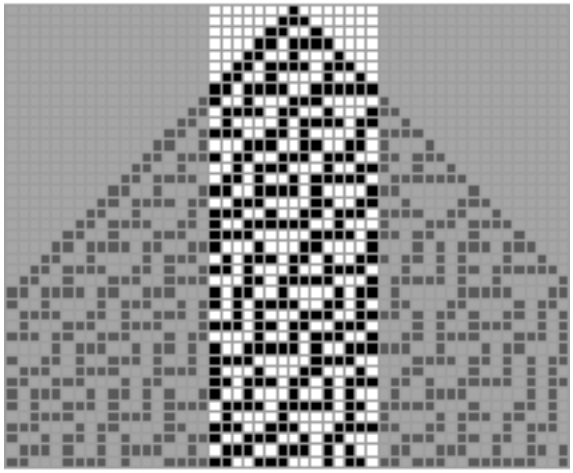

Figure 3: Swath through pattern

Now we can assume that time runs across the page from left to right and every black cell represents a note played. Wolfram has developed his own language and uses various of his own algorithms to form music out of these cellular automaton patterns.

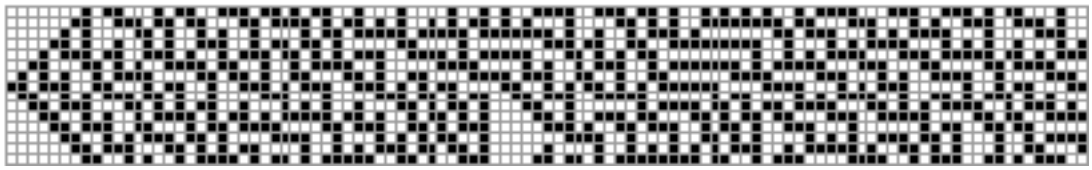

Figure 4: Musical score

The number of rules isn't limited by eight individual rules. We can determine the color of the next cell by looking at his five upper row neighbours, instead of three. This gives a much bigger number of results (around 4 billion) and much more interesting results to use in the music creation.

2 Chords and progressions

A crucial element of music is the use of chord progressions. The most common example of this is a twelve bar blues. This type of music is characterised by a strict pattern. Every song that uses a twelve bar blues follows the same chord progression.

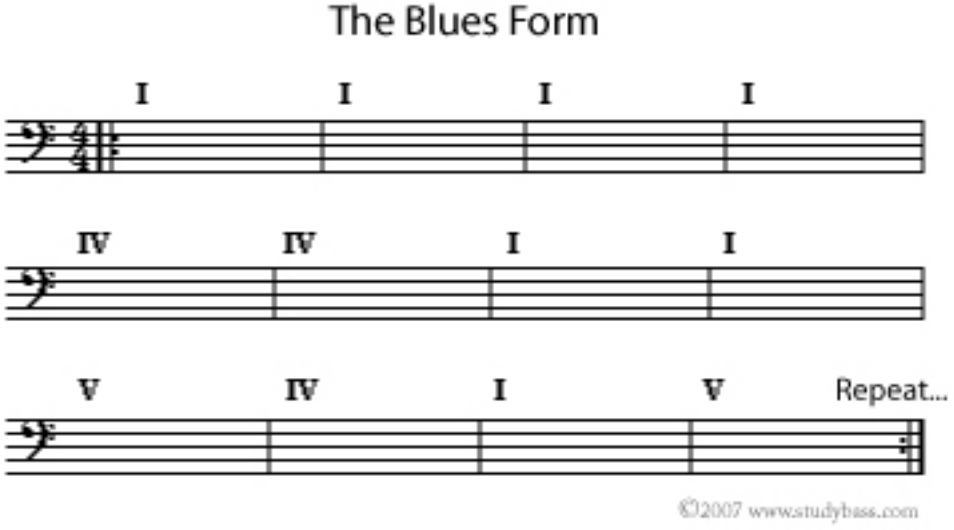

Figure 5: Twelve bar blues

In figure 5 you can see the structure of a twelve bar blues, very simple yet very recognisable. The roman numbers denotes which chord there will be played, relatively to the key of the song.

There are many other kinds of progressions, such as the I-II-V-I chord progression, or any other combination. Most popular songs use multiple chord progressions, one for the verse, one for the chorus and so on.

In ArtToMusic I will be using some standard chord progressions and some more unusual ones. There are a few reasons I choose for the use of chord progressions. Firstly, it is the most common way to write music. Secondly, it facilitates the generation of music in code. Several progressions are known to the ArtToMusic program and can be played easily, given the key of the song.

A few of these standard chord progressions are the following:

- I-II-V-I: happy, vibrant
- I-V-VI-IV: neutral
- I-IV-I-V: very happy, upbeat

To make matters more interesting, I will use some famous numbers from mathematics, such as π, ϕ and e. These irrational numbers can be seen as a series of digits. These digits can be seen as chords in a progression. For example, the first five digits of π are 3, 1, 4, 5, and 1. When we see them as chords we get the following chord progression: III-I-IV-V-I, which seems a very interesting progression.

Analogous to the example we can use the digits of the other numbers and create many progressions based on the graphical analysis.