

The technology behind DarwinTunes

Here we take a brief look under the hood of DarwinTunes.



The evolutionary algorithm: genetic programming

DarwinTunes is built around an evolutionary algorithm called genetic programming (GP). In GP, you have a population of individuals, each of which contains genetic material that is translated into computer code. The individuals reproduce and mutate, and their reproductive fitness is determined by executing the computer code and comparing the output to the desired output. You may already be familiar with the genetic algorithm (GA), which is similar to GP, but does not the generation and execution of code; instead the genetic material encodes parameters that are plugged into a pre-existing model.

DarwinTunes uses a GP system called PerlGP, which was developed by Dr. Bob MacCallum. It is written in, and evolves, Perl, an interpreted scripting language used for many applications. The genetic material is stored as a tree structure, in common in many other GP systems, because trees are good ways to represent computer code. When the code needs to be evaluated, the tree is flattened into a string of Perl code, which is passed through the interpreter (i.e. it is executed).

DarwinTunes is based very heavily on the evolutionary music system used at Evoelectronica (<http://evoelectronica.com>), also by Bob MacCallum.

Each individual in the population corresponds to a song, and that song is just a piece of Perl code.

Representing a song in Perl

Audio files can be generated easily using a number-crunching library for Perl called PDL (Perl Data Language), since audio is just a lot of numbers (amplitudes) in a row - 44100 per second, to be precise. Stereo audio is just two rows of these numbers (left and right channels). A "row of numbers" is more technically called an array. The Perl code to make a song starts by creating an array of zeroes which corresponds to several seconds of silence (a song in DarwinTunes is

usually a four bar loop, with four beats in a bar). Then several layers of sounds are added, one layer after the other, at different positions in the song (the rhythm), at different pitches (the melody or harmony), and with different sound qualities (timbres). The sounds are made by additive synthesis; simply adding sine waves of different frequencies, phase and volume together, and then shaping them with an "envelope". (At Evoelectronica, pre-existing samples can also be used, e.g. drum sounds.) Sounds can be passed through any number of effects (these are simply function calls) before they are added to the song. Effects include reverb, various filters, pitch shifting and time-stretching. When a sound is added near the end of the loop, it wraps around to the beginning of the loop, thus allowing seamless looping.

When all the layers have been added to the audio, the raw data is saved to a WAV file, and then converted into MP3 using the open-source program lame with some special options to ensure seamless looping.

Loop generation, streaming and the ratings interface

Generating the audio is a computationally intensive task, so it has to run on a dedicated server (not a shared one which serves many other websites). Therefore we run the loop generation on the Amazon EC2 "cloud" service (where you can rent machines by the hour). The Amazon servers have plenty of bandwidth so we also stream the audio from there, using the open-source streaming server, icecast, and a stream source generator called ezstream. When you provide a rating, it is only incorporated by the evolutionary algorithm after approximately one minute has elapsed since the loop was last streamed. This accommodates any network delays that might be occurring and allows several people to provide ratings for the same loop if they are on DarwinTunes at the same time.