

GB



ology@github

Search this website .

Imitating a Drum Circle

2020-11-01 BY GENE



Ever been to the park and heard a tribe drumming in the distance? This is a [drum circle](#). Upon closer inspection, each drummer has a single instrument (or maybe a set of bongos). Each drummer keeps a steady beat in common and in syncopation with his or her fellow

drummers.

I wondered if this could be simulated somehow? There are of course decisions to be made – parts to this equation... How should the drummers enter? All at once or gradually? How should a drum be chosen or selected? And what phrase(s) should be played anyway?

Enter MIDI-Perl. Here is the program: [drum-circle](#)

Basically it does the following:

1. Use the modules necessary to have a drummer and rhythmic phrases.
2. Define the number of drummers (\$max) as a number supplied by the user (or 5 by default).
3. Define a [MIDI::Drummer::Tiny](#) instance (\$d), with which we will drive everything.
4. Set the possible drums to use (\$DRUMS). In our case this is 15 toms, bongos, congas, cabasa, claves, and wood blocks.
5. Declare an array of phrases (@phrases), which is a list of code-references.
6. Define a 4 bar [Music::Duration::Partition](#) instance (\$mdp), which we will use to generate rhythmic motifs for each drummer.
7. Loop from 1 to \$max, appending phrases to play.
8. Add the phrases to the score such that they are played simultaneously.
9. Write the generated “composition” to disk, named after the program “drum-circle.”
10. Define the subroutine to generate our phrases!

This last part is shown here:

```
1. sub phrase {
2.     my ($p) = @_;
3.
4.     my $drum = $DRUMS[int rand @DRUMS];
5.     while ($seen{$drum}++) {
6.         $drum = $DRUMS[int rand @DRUMS];
7.     }
8.
9.     my $motif = $mdp->motif;
10.
11.     my $phrase = sub {
12.         for my $n (1 .. $d->bars + 4) {
13.             if ($n < ($p * 4)) {
14.                 $d->rest($d->whole);
15.                 next;
16.             }
17.             for my $i (@$motif) {
18.                 $d->note($i, $drum);
19.             }
20.         }
21.     };
22.
23.     return $phrase;
24. }
```

Here the variable \$p is the number 1 to \$max from our loop in step #7 above. Next, an unseen drum is selected at random. Then a *quasi-random* rhythmic motif is generated by [Music::Duration::Partition](#) as defined above. Finally an anonymous subroutine is returned, that just decides whether to play the generated motif or rest for a whole note.

This resting is crucial. Each drummer enters after 4 bars of the previous drummer having played. So, the second drummer enters after 4 bars, the third after 8, the fourth after 12, etc. Having everyone play at once or at random was just not a pleasant musical evolution. So I chose to have them enter one at a time.

Each run of this is different. Sometimes you get a “pleasant musical evolution”, but sometimes you get all toms, or just a weird clash of rhythms. Here is one with 8 voices that is ok:

00:00

00:00

And how about another?

00:00

00:00

Ok, one more!

00:00

00:00

FILED UNDER: MUSIC, SOFTWARE
TAGGED WITH: DRUMS, MIDI, PERL

Epistemologist-at-large

^ Top