

17 自動作曲システム

野呂 茂洋

指導教員 菅野 研一

1. はじめに

近年コンピュータが身近になりパソコンを用いて曲を作ることが出来るようなソフトウェアを簡単に入手できるようになった。それに伴い楽器を演奏できない人でも、楽器を演奏することなく曲を作ることが出来るようになった。その結果、趣味で作曲を行うような人も現れたが、曲を作るためには音楽についての知識は必要不可欠であり、本人の感性や経験の問題などもあり素人が曲を作るということが難しいという状況に変わりは無い。

2. 研究概要

2.1 目的

本研究では、音楽についての知識の有無に関わりなく誰もが簡単に曲を作ることが出来るような自動作曲を行うツールを作成することをめざす。

過去に自動作曲を卒業研究のテーマとした作品がいくつかあるため、音楽理論について学びながらフリーソフトウェアである Pure Data を用いてプログラムを組んでいく。具体的には Pure Data により自動生成した曲情報を MIDI データとして出力、DAW で演奏する、という流れを想定している。

2.2 開発環境

今回の研究で使用する開発環境は、以下に示す通りである。

表 1 開発環境

OS	Windows10
IDE	Pure Data
使用言語	Pure Data

2.3 Pure Data について

Pure Data (Pd) は、1990 年代にミラー・パケッ

ト氏が開発した音声、ビデオ、映像処理のためのビジュアルプログラミング言語である。ミラー・パケット氏が主に開発したが、Pure Data はオープンソースプロジェクトであり、多数の開発者が参加している。また、創作できる一例を挙げると以下のとおりである。

- ・シンセサイザー
- ・サンプラー
- ・リズムマシン
- ・オーディオエフェクター

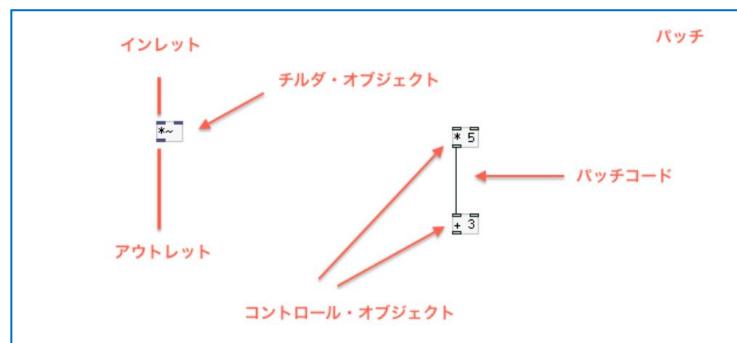


図 1 Pure Data の各名称

2.4 MIDI について

Musical Instrument Digital Interface の頭文字をとったもので、電子楽器の演奏情報をやり取りするための世界共通のデジタルインターフェイスの規格。MIDI に対応した電子楽器や機器には必ず MIDI 端子があり、MIDI 端子を MIDI ケーブルで接続することで、簡単に演奏情報を別の電子機器に伝えることができる。

3. 音楽の成り立つ条件

乱数で音を出すだけでは音楽にはならない。メロディー、コード、リズムなどがバラバラだと、とても不快な音楽を奏でることになる。そこで、テ

ンボ、リズム、コード、ベース音などに様々なルールを導入する。

4. 自動作曲

4.1 リズムについて

今回、音の長さを以下のように決めた。

$$\text{♩} = 1000 \quad \text{♪} = 500 \quad \text{♪} = 250$$

図 2 音の長さ

これは 4 分音符を 1 拍としたときの比率で、8 分音符は 1/2 拍、16 分音符は 1/4 拍という計算で求めることができる。これらを組み合わせて拍を作ることとする。

4.2 拍の決め方

拍を決めるために次のような構成図で考える。

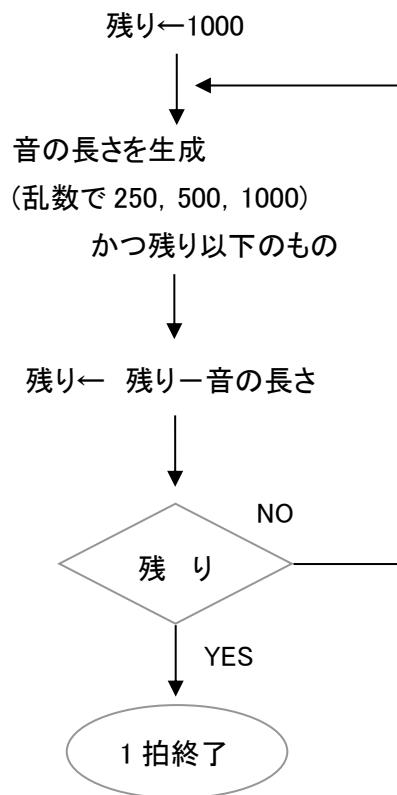


図 3 拍を決めるための構成図

このように考えると次の 5 通りの拍を生成できることが分かる。



図 4 拍のパターン

4.3 コードについて

今回、数あるコードの中から不協和音を除いた次の 11 個にコードを絞った。

C (ド・ミ・ソ), **Dm** (レ・ファ・ラ),
Em (シ・ミ・ソ), **E7** (シ・レ・ミ・ソ#),
F (ド・ファ・ラ), **G** (シ・レ・ソ),
G7 (シ・レ・ファ・ソ), **Am** (ド・ミ・ラ),
Adim (ド・レ#・ファ#・ラ)
B7 (シ・レ#・ファ#・ラ), **Faug** (ド#・ファ・ラ)

各コードに、次にきても不自然でない組み合わせを下に示す。この図を用いて、コード進行を生成していく。

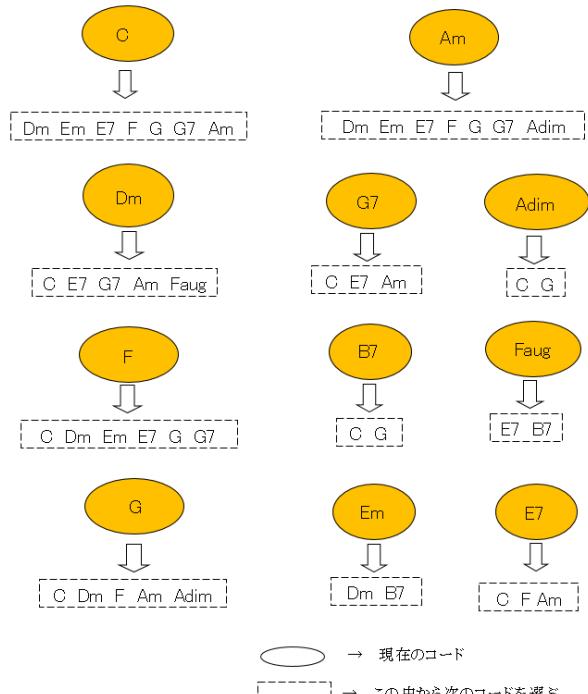


図 5 コード進行

5. プログラム設計

4. 自動作曲からプログラムはリズムを生成させるものと、コードを生成させるものの 2 つのモジ

ユールに分けることができる。そこで、リズムを生成させるものを Rhythm patch、コードを生成するものを Chord Patch とし、プログラム設計を行う。

5.1 Rhythm patch

Rhythm patch は、リズムを生成するプログラムである。音の長さに数字を割り当て、データを扱いやすくした。

表 2 音の長さ

0	1	2
$\text{♪} = 250$	$\text{♪} = 500$	$\text{♩} = 1000$

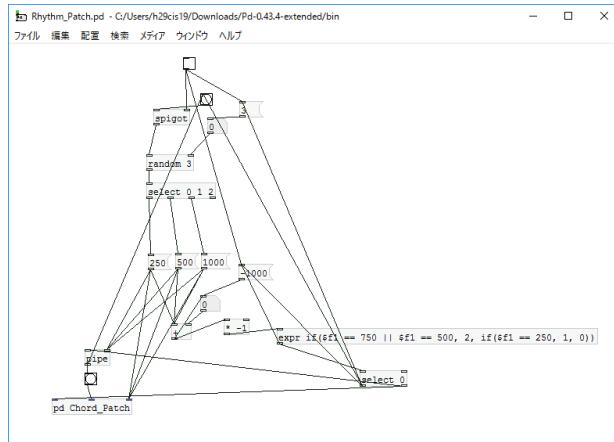


図 6 Rhythm patch

まず、random オブジェクトで最初の音の長さを決める。random オブジェクトとは、bang メッセージを受け取ったときに、0 から指定した数値より 1 小さい値までの範囲をランダムな数値で出力するものである。

random オブジェクトの範囲は以下のように変化する。

表 3 random オブジェクトの範囲

拍の残り数	random オブジェクトの範囲
750	2
500	2
250	1
0	0

生成された音の長さは pipe オブジェクトに入り、リズムとなって Chord Patch に入る。

pipe オブジェクトとは、第 1 インレットに数値を受け取ると、アーギュメントまたは第 2 インレットで指定した msec 単位の時間だけ遅らせて、その数値を出力することができる。

5.2 Chord Patch

Chord Patch はコードを選択する部分と、次のコードを生成する部分で構成される。

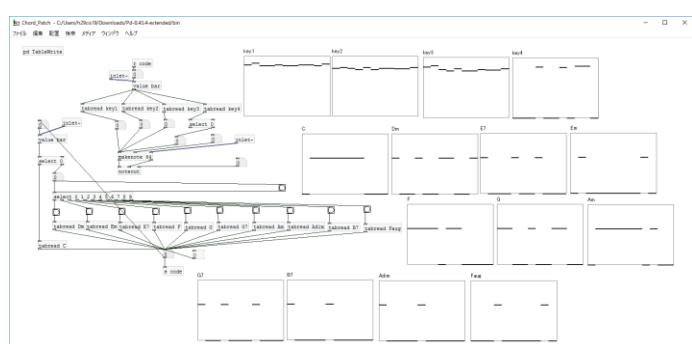


図 7 Chord Patch

使用するコードは 4. 自動作曲でも説明したように C, Dm, Em, E7, F, G, G7, Am, Adim, B7, Faug である。各コードに数字を割振り、データを扱いやすくする。

表 4 コードの割り当て

0	1	2	3	4	5
C	Dm	Em	E7	F	G
6	7	8	9	10	
G7	Am	Adim	B7	Faug	

これらのコードは三和音、又は四和音で構成されている。この音のデータを扱うために table オブジェクトを使用する。table は、一定範囲の整数に対応する整数の組み合わせを記憶し、呼び出すことができるオブジェクトである。4 つの table を使用し、各 table 名を key1, key2, key3, key4 とした。各 table に入る数値を以下に示す。

表 5 各テーブルの内訳

コード番号	key1	key2	key3	key4
0	ド (60)	ミ (64)	ソ (67)	
1	レ (62)	ファ (65)	ラ (69)	
2	シ (59)	ミ (64)	ソ (67)	
3	シ (59)	レ (62)	ミ (64)	ソ# (68)
4	ド (60)	ファ (65)	ラ (69)	
5	シ (59)	レ (62)	ソ (67)	
6	シ (59)	レ (62)	ファ (65)	ソ (67)
7	ド (60)	ミ (64)	ラ (69)	
8	ド (60)	レ# (63)	ファ# (66)	ラ (69)
9	シ (59)	レ# (63)	ファ# (66)	ラ (69)
10	ド# (61)	ファ (65)	ラ (69)	

()はノートナンバー

コード番号が 4 つの table に入り、その番号に格納されているノートナンバーを呼び出す。その後、makenote に入り、ノートオフのメッセージを加え noteout される。

6. 試作品

下の図 8 は本プログラムを組む前に作成した、乱数を用いた自動作曲プログラムである。この試作品はコードを選択する部分が未実装のため、制約の少ない random オブジェクトにより MIDI 信号が決定される。

実際に生成されたものは音楽とは程遠いものだった。

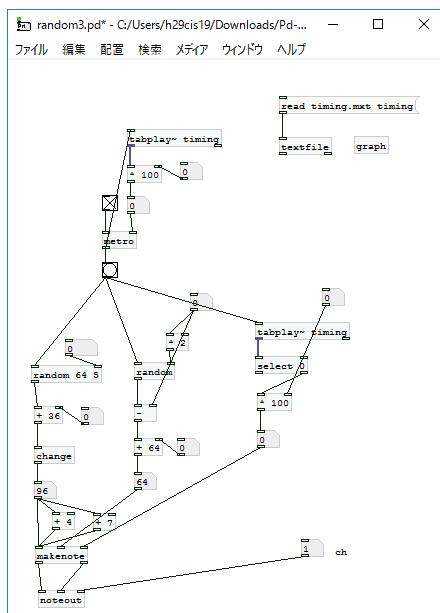


図 8 試作品

7. まとめ

今回作成した作曲プログラムは、試作品として作成した音を乱数で組み合わせたものと比べると、数段音楽的に聞こえる。しかし、当初予定していた MIDI 情報を保存する機能を付けることが出来なかった。Pure Data はフリーソフトウェアのためか、作成したデータを外部に保存する(書き出す)機能が限定的であるため、MIDI ファイルとして保存する機能を実装するのに複雑なパッチを組まなければならず期間までに完成することができなかった。

今後の課題として、本研究を通じ Pure Data を多少扱えるようになったため、より完成度の高いプログラムを組めるように勉強を続けていきたい。そして、今回実装することができなかった MIDI データとして作成した曲を外部へ保存する機能の導入を検討したい。

参考文献

- 1) 柳田久美子:「自動作曲システムの作成」, 卒業研究 (2003).
- 2) 咲山尚幸:「自動作曲システムの作成」, 卒業研究 (2004).
- 3) 美山千香士:『Pure Data—チュートリアル&リファレンス』:株式会社ワークスコーポレーション (2013).
- 4) 「Pure Data Japan Pure Data とは?」<http://puredatajapan.info/?page_id=2>