Creating Coding School Sapporo 2017

札幌国際芸術祭編

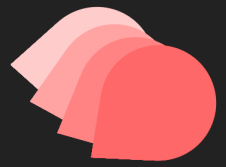
Sonic Piでライブコーディング

金井 謙一

ARTSAT x SIAF LAB.

Petal

Petal

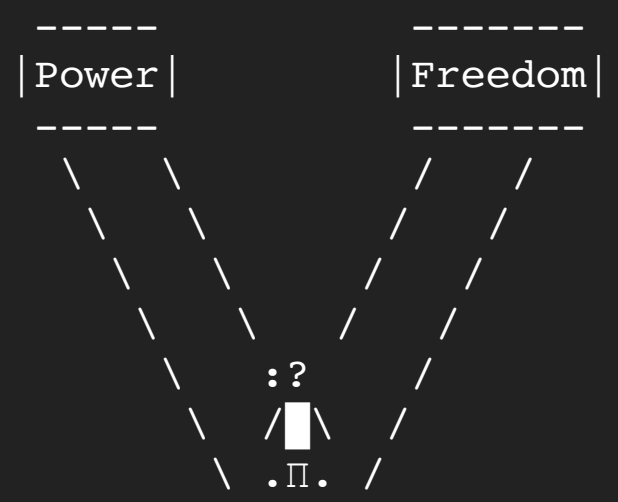


- https://github.com/siaflab/petal
- ***Petal*** is a small language on Sonic Pi with similar syntax to TidalCycles.
- Sonic Pi上に作られたTidalCycles風言語
- space-moereで、短い記述のコードが必要になったのが一番の理由
  - 920MHz帯のLoRa変調SF12モードで150〜300bpsでコードをオンボード側に送信する必要
  - 独自に言語仕様を検討したくなかった

TidalCycles

- https://tidalcycles.org/
- Alex McLeanが開発
- Haskellベース
- a language for live coding patterns.
- an open source live coding environment based off of his 16 years of experience with making algorithmic dance music.
  - 'Alex McLeanの16年にわたるアルゴリズミックなダンスミュージックの制作経験を基にしたライブコーディング環境'
  - https://courses.ideate.cmu.edu/15-104/f2016/2016/09/23/rgriswol-lookingoutwards-04/
- 簡単な記法で複雑な音楽を記述可能

- Sonic Piコミッター Joseph Wilk氏
  - https://github.com/josephwilk/strangeloop2014




```
One path leads to instant power but you are never truely free.
The other leads to complete freedom but you have to build your world
```

- Power: TidalCycles
- Freedom: Sonic Pi, Overtone

Petalのインストール

- https://github.com/siaflab/petal
- Petalのzipファイル(petal-beta2.zip)を解凍(petal-beta2が作成されます)
- 作成されたpetal-beta2フォルダをpetalフォルダにリネーム

- Mac
- petalフォルダをユーザーのホームディレクトリ(/Users/<username>/)配下に、/Users/<username>/petalとしてコピー

- Windows
- petalフォルダをユーザーのホームディレクトリ(C:¥¥Users¥<username>¥)配下に、C:¥¥Users¥<username>¥petalとしてコピー

Hello Petal

- ~/petalが存在する状態で、以下のコードをSonic Piで実行

```
load "~/petal/petal.rb"
d1 "bd"
```

解説

```
d1 "bd"
```

- この例では、petal/Dirt-Samples/bdフォルダ配下の0番目のサンプルを再生しています。
  - petal/Dirt-Samplesに200ほどフォルダがあるので、この中からサンプルを探りつつ音を作っていきます。
- サンプル再生を一定時間で繰り返しています。
  - この繰り返しを、TidalCyclesでは**サイクル**(cycle)と呼んでいます。

Sonic Pi内蔵サンプルの使用

- Sonic Pi内蔵サンプルも使用可能です。

```
d1 :bd_klub
```

音を止める

- Sonic PiのStopボタンを押すと止まります。
- d1の引数を空にすると、d1を止めることができます。

```
d1
```

- :silenceを指定しても同様です。

```
d1 :silence
```

- hushは、全てのループが止めます。

```
d1 "bd"
hush
```

インデックス指定

- bdファルダ内の1番目(0から開始することに注意)を指定してみましょう。

```
d1 "bd:1"
```

- 同じことは以下でも実現可能です。

```
d1 "bd", n: 1
```

音を増やす

- bdを2つ書くと2回鳴ります。

```
d1 "bd bd"
```

- 同じことは、*でも書けます。

```
d1 "bd*2"
```

- /は、再生頻度を減らします。

```
d1 "bd/2"
```

- 特定のサンプルだけ頻度を減らすことも可能です。

```
d1 "bd/4 sn"
```

- []を使って入れ子も作れます

```
d1 "[bd hh] sn"
```

```
d1 "[bd [hh bd]] sn"
```

- *も使って更に複雑にしてみましょう

```
d1 "[bd [hh bd*3]] [sn hh]"
```

- ~は休符です。

```
d1 "~ sn"
```

```
d1 "[bd [~ bd*3]] [sn hh]"
```

ループを増やす

- ループを増やすには、d2〜d9を使います。
  - TidalCycles用語では、**コネクション(connection)を増やす**と言います。

```
d1 "[bd [~ bd*3]] [sn hh]"
d2 ":bass_woodsy_c/3"
```

```
d1 "[bd [~ bd*3]] [sn hh]"
d2 ":bass_woodsy_c/3"
d3 ":ambi_lunar_land/7"
```

- ループを止めるには空にするか、:silenceを指定します。

```
d1
d2 ":bass_woodsy_c/3"
d3 ":ambi_lunar_land/7"
```

- d1が止まり、d2,d3の音が残ると思います。

- 特定のループのみにするには、soloを使います。

```
# d1 "[bd [~ bd*3]] [sn hh]"
solo :d1, "[bd [~ bd*3]] [sn hh]"
d2 ":bass_woodsy_c/3"
d3 ":ambi_lunar_land/7"
```

- d2,d3が止まり、d1の音が残ると思います。

- 全てのループを止めるには、hushを使います。

```
d1 "[bd [~ bd*3]] [sn hh]"
d2 ":bass_woodsy_c/3"
d3 ":ambi_lunar_land/7"
hush
```

テンポ設定

- cps(cycle per second: 1秒あたりのサイクル数)

```
cps(0.55)
d1 "[bd [~ bd*3]] [sn hh]"
d2 ":bass_woodsy_c/3"
d3 ":ambi_lunar_land/7"
d4 "hh hh hh*4 hh hh hh*8"
```

- デフォルト(何も指定しない場合)は、cps=1です。
- わり算で指定する場合は、少数型になるように指定する必要があります。

```
cps(1/2) # 0になってしまう
cps(1/2.0)  # 0.5になり意図した通りになる
cps(1/2.to_f) # 多少長いがこちらも意図通りになる
```

- cpsの他に、以下を用意しています。
  - bps(beat per second: 1秒あたりのビート数)
  - bpm(beat per minute: 1分あたりのビート数)
- それぞれの関係は以下のとおりです。
  - cps(1) == bps(120/60) == bpm(120)

オプション

- gain:, amp:で音量を変更できます。
  - デフォルトは1です。

```
d1 "bd", amp: 2
```

- pan:でパンを変更できます。
  - デフォルトは0(中央)です。

```
d1 "bd", pan: -1
```

- 左から音が出たと思います。

- speed:, rate:で再生レートを変更できます。
  - デフォルトは1です。
  - マイナスにすると逆再生します。

```
d1 "bd", rate: -1
```

- density:でサイクルを速くできます。
  - デフォルトは1です。

```
d1 "bd", density: 2
```

- slow:でサイクルを遅くできます。
  - デフォルトは1です。

```
d1 "bd", slow: 2
```

- density:とslow:の関係

```
d1 "bd", density: 2
```

と

```
d1 "bd", slow: 0.5
```

は同じです。

- stretch:で、サンプルの長さをサイクルの長さに合わせることが可能です。
  - :b - ビートストレッチ(音の高さが変わる)
  - :p - ピッチストレッチ(音の高さを極力変えない)

```
d1 :loop_breakbeat, slow: 2, stretch: :b
```

ランダム

- オプションにはランダムを指定可能です。

```
d1 "bd", amp: "rand"
```

- ループの度に音量が変化していると思います。

- rand - ランダムな少数を返します。
  - rand
    - 0〜1の少数
  - rand 5
    - 0〜5の少数
  - rand -1 1
    - -1〜1の少数

- irand - ランダムな整数を返します。
  - irand
    - 0〜1の整数(0か1)
  - irand 7
    - 0〜7の整数
  - irand -1 4
    - -1〜4の整数

ユークリッド・リズム

- PetalでもTidalCyclesと同様にユークリッド・リズムが指定可能です。
  - 少ない記述量で複雑なリズムを表現できます。

```
d1 "bd(5,8)"
```

- 5をパルス、8をステップと呼んでいます。

- Godfried Toussaint(2005) "The Euclidean Algorithm Generates Traditional Musical Rhythms"
  - 「2つの数の最大公約数を求めるユークリッドのアルゴリズムをリズムに適用することにより、伝統的なリズムを記述可能である。」
  - 上の例では、2つめの数(8)のステップを通じて、最初の数(5)のパルスを等間隔で分散させようとします。
- デモサイトもあるのでとりあえず試すのが良いでしょう。
  - http://dbkaplun.github.io/euclidean-rhythm/

- パルス、ステップの指定の他にローテーションの指定が可能です。

```
d1 "bd(5,8,3)"
```

- それぞれの関係は以下のとおりです。
  - (5, 8) [x . x x . x x .]
  - (5, 8, 1) [x x . x x . x .]
  - (5, 8, 2) [x . x x . x . x]
  - (5, 8, 3) [x x . x . x x .]
- 次のパルスが先頭に来るまで、パターン全体を左にシフトしています。

- 論文内で、「伝統的なリズム」として以下が紹介されています。
  - (25) : A thirteenth century Persian rhythm called Khafif-e-ramal.
  - (3,4) : The archetypal pattern of the Cumbia from Colombia, as well as a Calypso rhythm from Trinidad.
  - (3,5,2) : Another thirteenth century Persian rhythm by the name of Khafif-e-ramal, as well as a Rumanian folk-dance rhythm.
  - (3,7) : A Ruchenitza rhythm used in a Bulgarian folk-dance.
  - (3,8) : The Cuban tresillo pattern.
  - (4,7) : Another Ruchenitza Bulgarian folk-dance rhythm.
  - (4,9) : The Aksak rhythm of Turkey.
  - (4,11) : The metric pattern used by Frank Zappa in his piece titled Outside Now.

- つづき
  - (5,6) : Yields the York-Samai pattern, a popular Arab rhythm.
  - (5,7) : The Nawakhat pattern, another popular Arab rhythm.
  - (5,8) : The Cuban cinquillo pattern.
  - (5,9) : A popular Arab rhythm called Agsag-Samai.
  - (5,11) : The metric pattern used by Moussorgsky in Pictures at an Exhibition.
  - (5,12) : The Venda clapping pattern of a South African children’s song.
  - (5,16) : The Bossa-Nova rhythm necklace of Brazil.

- さらにつづき
  - (7,8) : A typical rhythm played on the Bendir (frame drum).
  - (7,12) : A common West African bell pattern.
  - (7,16,14) : A Samba rhythm necklace from Brazil.
  - (9,16) : A rhythm necklace used in the Central African Republic.
  - (11,24,14) : A rhythm necklace of the Aka Pygmies of Central Africa.
  - (13,24,5) : Another rhythm necklace of the Aka Pygmies of the upper Sangha.

Utility関数

- Dirtサンプル再生
  - dirt_sampleでサンプルのファイルパスを取得できます。
  - これを使って試しにサンプルを鳴らしてみます。

```
sample dirt_sample("if", 0)
```

- Dirtサンプルファイルリスト取得
  - dirt_samplesで指定したフォルダ内のサンプルのファイルパスの一覧を取得できます。
  - これを使って順番にサンプルを鳴らしてみます。

```
dirt_samples("if").each_with_index do |e, i|
  puts i
  sample e
  sleep 0.5
end
```

- ユークリッドリズム確認

```
puts euclidean_rhythm(5, 8)
puts euclidean_rhythm(5, 8, 2)
```