

# プログラマブルな 音楽制作ソフトウェアの制作

九州大学大学院 芸術工学府 博士課程1年

松浦 知也

2019/04/20 未踏IT人材発掘・育成事業 二次審査

me@matsuuratomoya.com

# 自己紹介

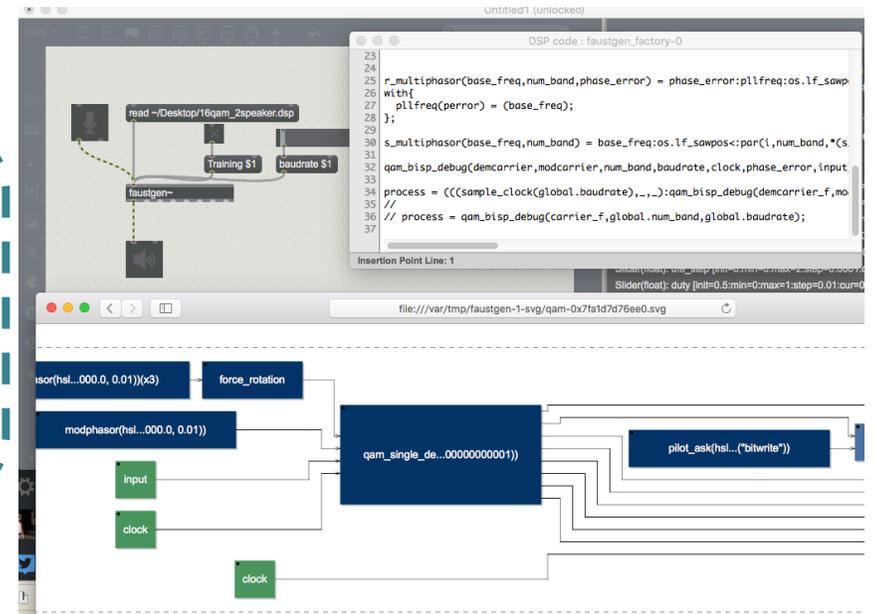
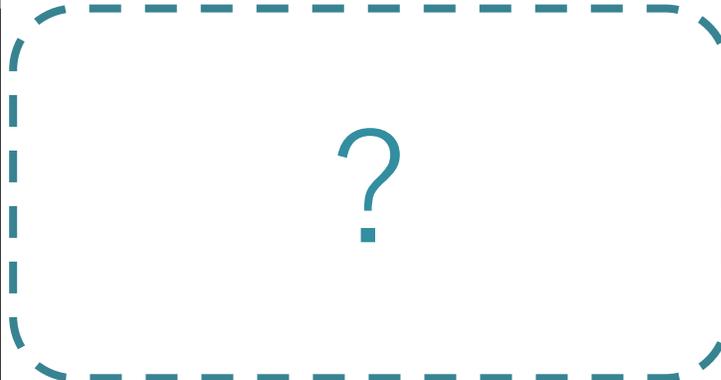


2018/08/18 蓮沼執太フルフィル「フルフォニー」  
での自作電子楽器「Exidiophone」演奏

## 松浦 知也

- Soundmaker：音楽を作るためにそれを生み出すシステムを作るアーティスト
- プログラミングを用いた展示作品（サウンドインスタレーション）からアナログ回路を設計して自作電子楽器まで
- 作品制作自体を研究として行い、学会発表も行う

# 要約



既存の音楽制作ソフトウェア  
(DAW)

ソースコードの編集ツールとしての  
音楽制作ソフトウェア

既存の音楽プログラミング  
環境

## 背景

音楽制作ソフトウェア(DAW)

音楽プログラミングの現在

個人的モチベーション

視覚表現における先行例

## 制作内容

ソフトウェアの構造

エディタのイメージ

言語仕様の構想

実装に用いる技術

## 目標

9か月間で作るもの

世の中への出し方

目指す/目指さないもの

現状/懸念事項

# 背景

音楽制作ソフトウェア(DAW)

音楽プログラミングの現在

個人的モチベーション

視覚表現における先行例

# 制作内容

ソフトウェアの構造

エディタのイメージ

言語仕様の構想

実装に用いる技術

# 目標

9か月間で作るもの

世の中への出し方

目指す/目指さないもの

現状/懸念事項

# Digital Audio Workstation(DAW)

Avid Protools



<https://www.avid.com/pro-tools/features>

Ableton Live



<https://www.ableton.com/ja/live/>

マルチトラックレコーダー、シーケンサー、ミキサーなど実際の機器をベースにしたワークフロープラグインによる拡張ができるが、楽曲そのものをプログラマブルに生成できるものはほとんどない

# 音楽プログラミングの現在

- 柔軟な時間スケールの記述（シグナル/メッセージを区別しない）
- オーディオ処理をC++などより低レベルの言語に依存せず書ける
- コードの即時実行と反映：ライブコーディングによる楽器演奏としてのコーディング



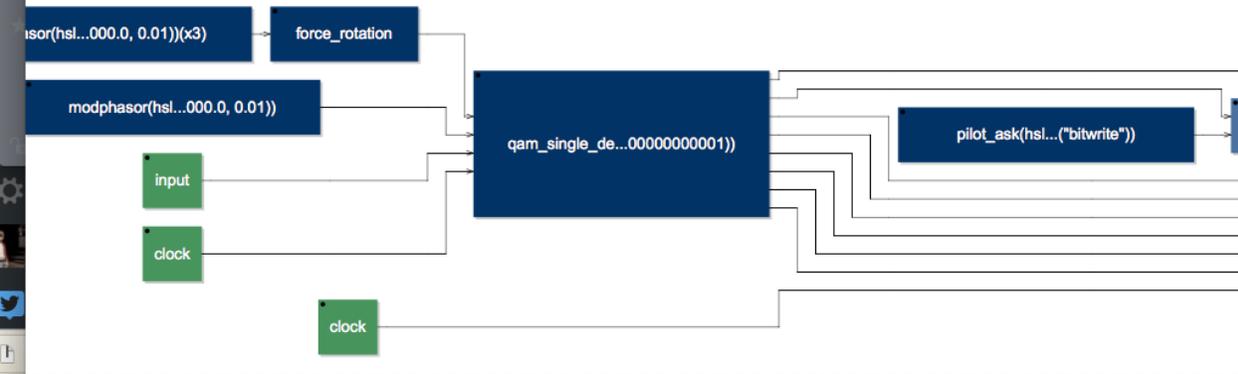
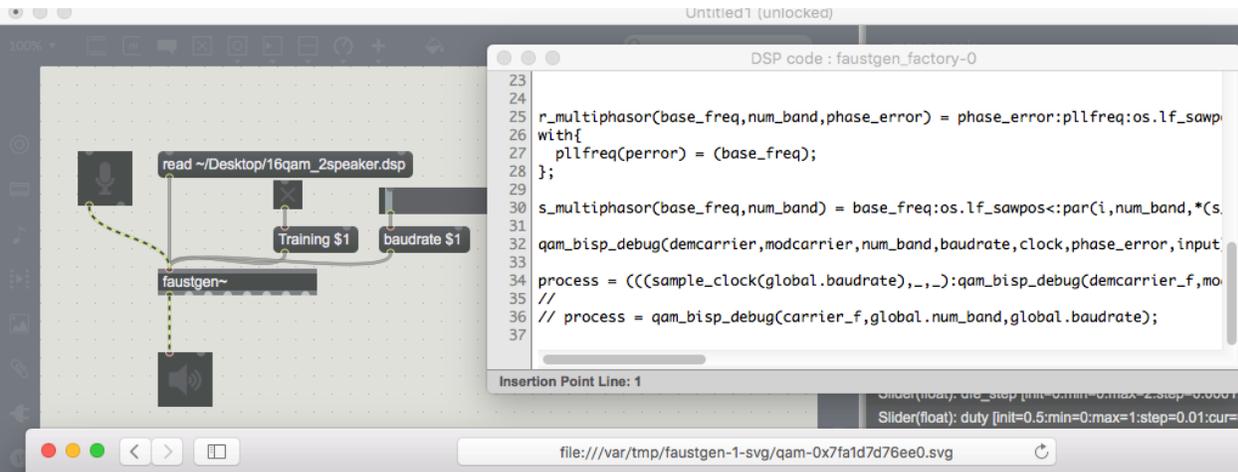
# Faust (GRAME,2002~)

- 関数型の音響処理言語

- C++やLLVMを介してWebaudioなど、  
複数のプラットフォームに出力

- 外部ライブラリに頼らない音響処理

- 言語の中ではすべてがサンプル単位の  
処理になる



# Extempore (Sorensen,2011~)

```
Sublime Text 2 File Edit Selection Find View Goto Project Window Help So. 13:29

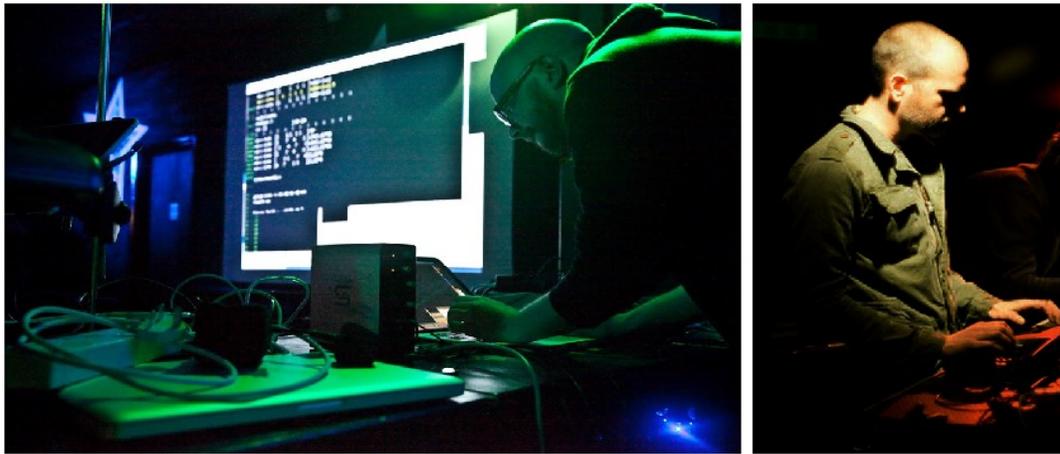
151 (for-each (lambda (p)
152 (let* ((dur1 (* dur (random '(0.5 1))))
153 (dur2 (- dur dur1)))
154 (play-midi-note (*metro* time) *midi-out* p
155 (real->integer (+ 50 (* 20 (cos (* pi time))))))
156 (*metro* 'dur dur1) 0)
157 (if (> dur2 0)
158 (play-midi-note (*metro* (+ time dur1)) *midi-out*
159 (pc:relative p (random '(-2 -1 1 2))
160 (pc:scale 0 'aeolian))
161 (real->integer (+ 50 (* 20 (cos (* pi (+ time dur1))))))
162 (*metro* 'dur dur2) 0))))
163
164 (pc:make-chord 50 70 3 (pc:diatonic 0 '- degree)))
165 (callback (*metro* (+ time (* 0.5 dur))) 'chords (+ time dur)
166 (random (assoc degree '((i vii)
167 (vii i v)
168 (v i vi)
169 (vi ii)
170 (ii v vii))))
171 (random (list 1 2 3))))
172
173
174 ;; add pulse 5:00
175
176 (define pulse
177 (lambda (time)
178 (play-midi-note (*metro* time) *midi-out* 60
179 (real->integer (+ 40 (* 20 (cos (* pi time)))))) (*metro*
180 (callback (*metro* (+ time (* 0.5 0.5))) 'pulse (+ time 0.5))))

Live 176, Column 1 Spaces: 2 Extempore
```

- LISPベースの音声/映像処理言語
- 外部ライブラリに頼らない音響処理
- シグナル/メッセージの区別なし
- ライブコーディング可

# 音楽プログラミングの現在

生成的表現と大衆音楽の接近



Collins, N., & McLean, A. (2014). *Algorave: A Survey of the History, Aesthetics and Technology of Live Performance of Algorithmic Electronic Dance Music*. *NIME*.

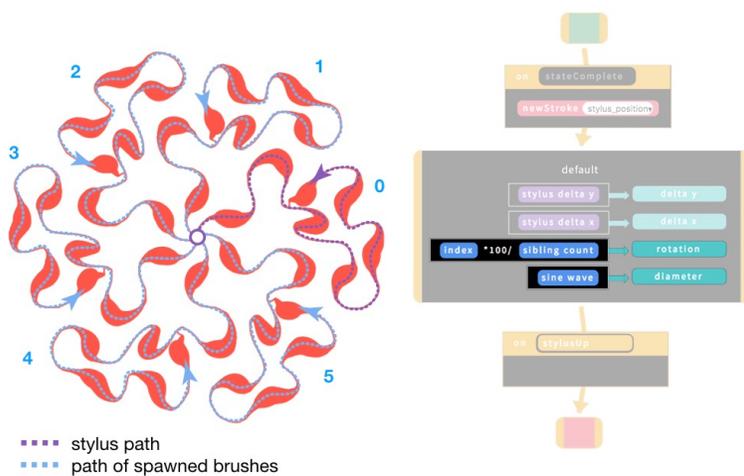


Algorave Tokyo 2018/08/19 at ANAGRA Tokyo  
ChucK+Faustによるライブコーディングで出演

# 個人的モチベーション

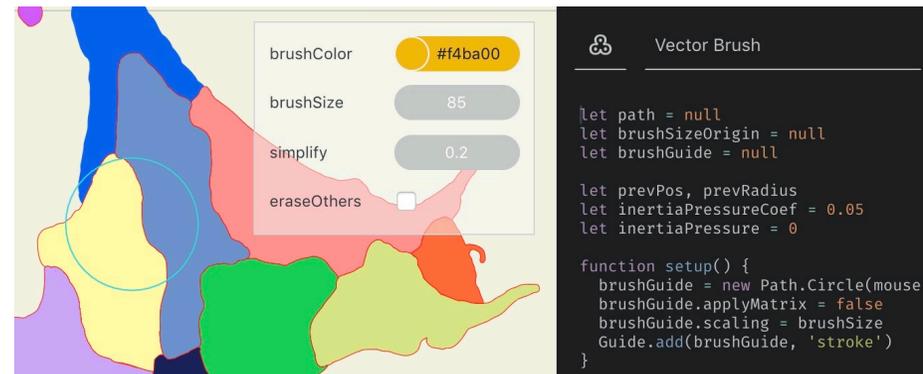
元々ポップミュージックが好きだけどDAW使っているとどうしても曲が作れなくて、プログラミングで音生成したり自作電子楽器作ったりして面白い音素材を作ってみたものの、それをDAWで切り貼りして音楽を作ろうとしてもやっぱり完成せず、そうこうしてる間に自作楽器でライブとかしてたら評価され始めるようになったんだけど、気づいたらノイズ音楽をやる人だと思われるようになって、いや実は電子音楽とかノイズとか全然好きじゃないし、別に自分がやりたかったのってそういうのじゃなかった気がするんだけど、ってというか**DAWみたいなインターフェースのソフトウェアの中でもっと柔軟にプログラミング的表現が扱えれば全部解決する気がするんですけど・・・**

# 視覚表現における先行事例



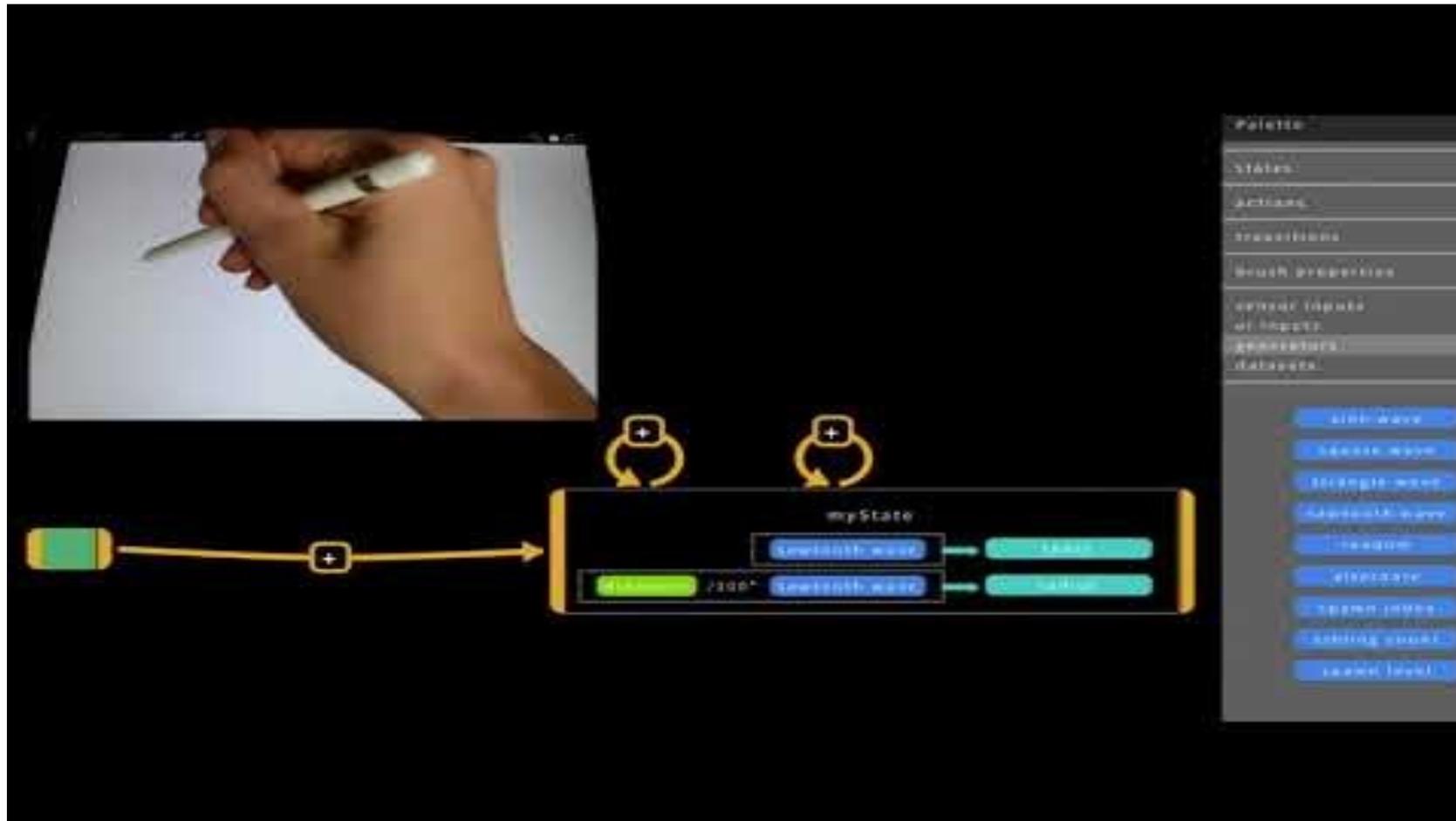
Jennifer Jacobs  
“Para”/ “Dynamic Brushes”

Ravi Chugh  
“Sketch-n-Sketch”



橋本麦  
Programmable Pen Tool

# 視覚表現における先行事例

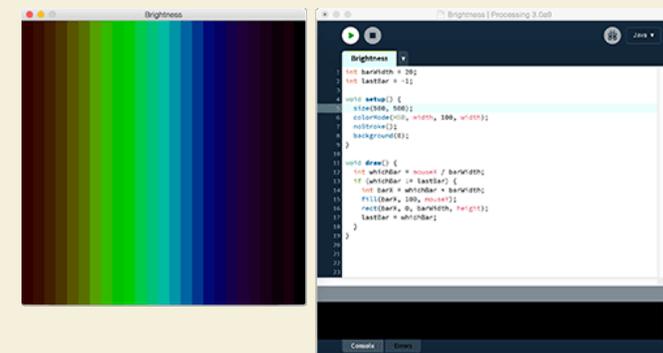
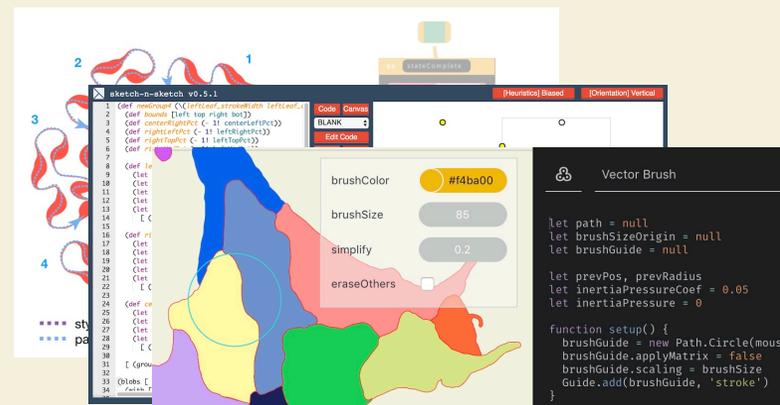
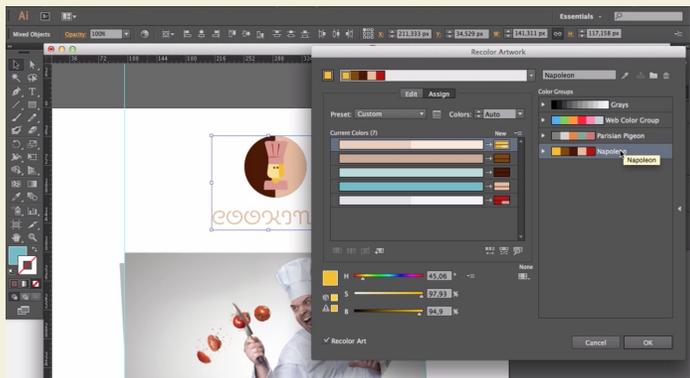
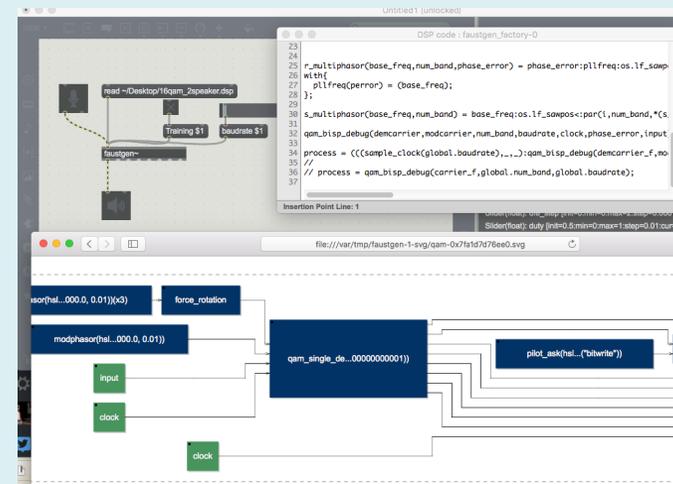


GUI Application



Programming Environment

Sound



Visual

## 背景

音楽制作ソフトウェア(DAW)

音楽プログラミングの現在

個人的モチベーション

視覚表現における先行例

## 制作内容

ソフトウェアの構造

エディタのイメージ

言語仕様の構想

実装に用いる技術

## 目標

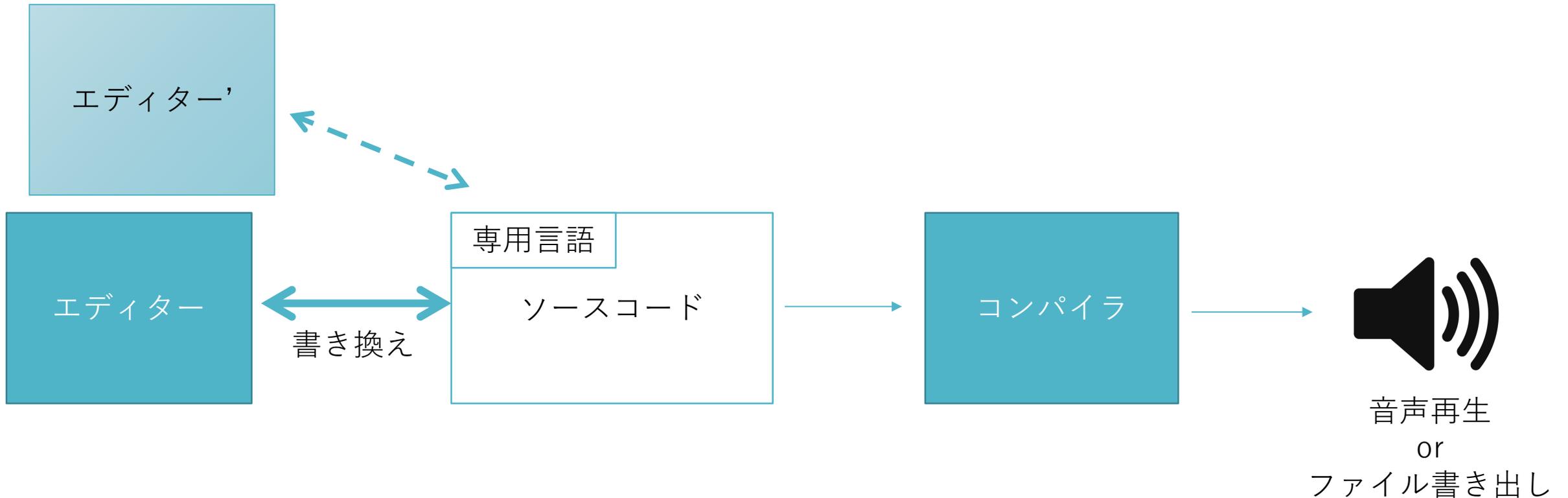
9か月間で作るもの

世の中への出し方

目指す/目指さないもの

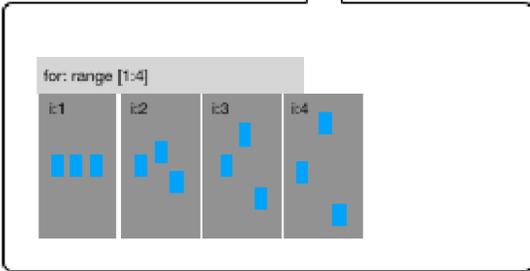
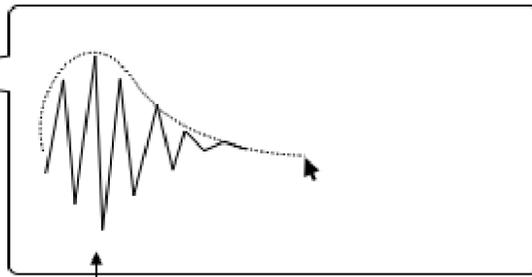
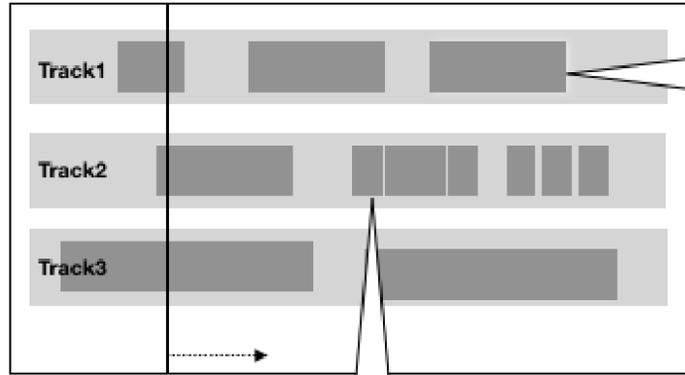
現状/懸念事項

# ソフトウェアの構造



ソースコードを楽譜 $\div$ その音楽作品の記号的定義と考え、それを操作するツール

繰り返しの構造であれば、テキスト上で同じ構造が繰り返されるのではなく“繰り返しという命令”を記述するべき



```
wave = triangle*bezier(10,20,30,20,...)  
range = [t1,t2]  
t1 = 3::minutes+100::ms  
t2 = t1+40::seconds
```

エディター

エディター'

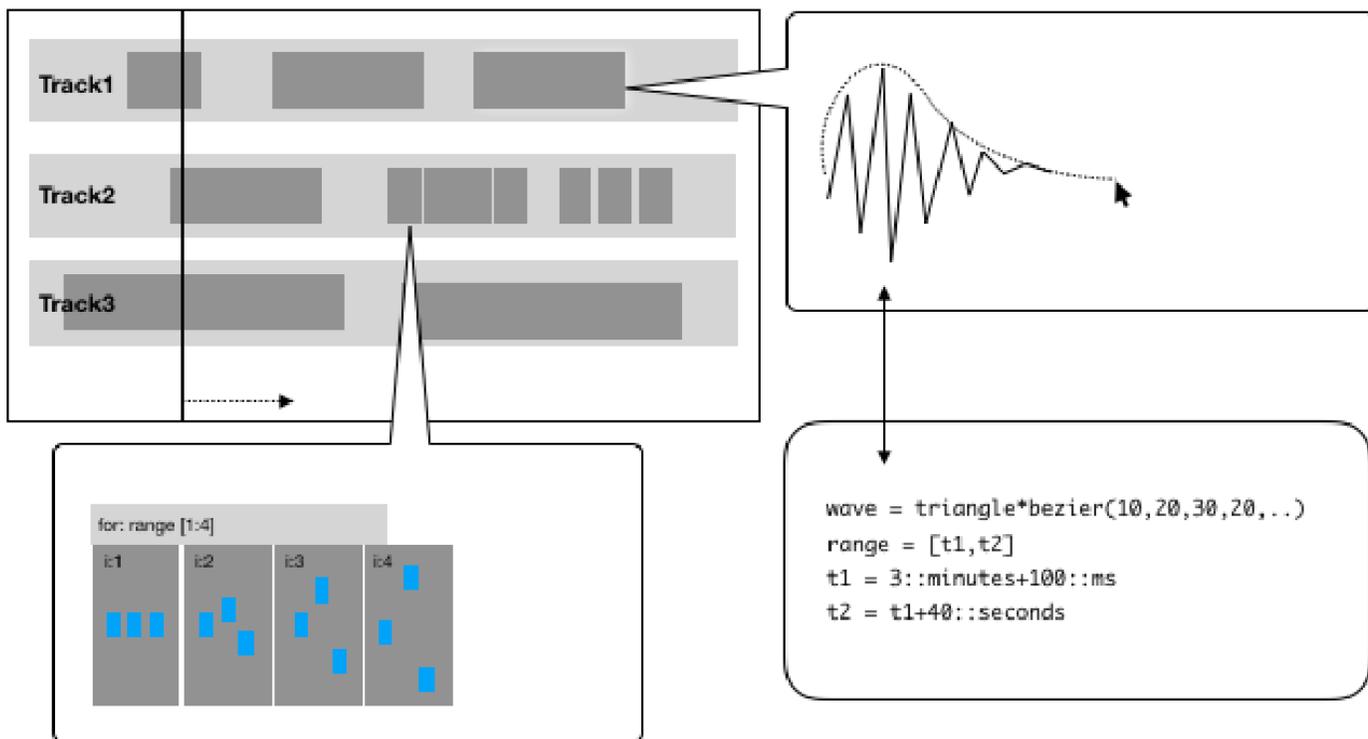
書き換え

専用言語  
ソースコード

コンパイラ



# エディターのイメージ



基本的には、DAWのようにタイムラインベースのエディタ

# 言語仕様の構想

```
Type timepoint [Time t,float v]
```

```
Type region [timepoint tp1,timepoint tp2]
```

```
Type Regeonvec2 [Regeon r1,int i1,int i2]
```

```
Type Note Regeonvec2(i1=pitch,i2=velocity)
```



一般的なプログラミング言語の型 + 時間だけをプリミティブとして用意し、音価や音声波形は言語の中の標準ライブラリとして定義：全ての音楽的要素を時間 & 数値の構造体としたボトムアップ的な定義（Extemporeが近い構造）

その上で、Lisp系よりはヒューマンリーダブルなコードになるように…

# 実装に用いる技術

HTML/CSSによるGUI設計：高速プロトタイプ&カスタマイズ性

一方で、すべてをブラウザで実装するにはWebaudioが不安定

→C++上でHTMLを表示できるライブラリを使用？ (ultraligh.ht)

オーディオドライバのクロスプラットフォーム対応：JUCE (C++用ライブラリ)

# 背景

音楽制作ソフトウェア(DAW)

音楽プログラミングの現在

個人的モチベーション

視覚表現における先行例

# 制作内容

ソフトウェアの構造

エディタのイメージ

言語仕様の構想

実装に用いる技術

# 目標

9か月間で作るもの

世の中への出し方

目指す/目指さないもの

現状/懸念事項

# 9か月間で作るもの

## 最低限：

- タイムライン型のGUIで波形生成→音声ファイル書き出しまで（非リアルタイムでもよい）
- “メタ”トラック機能の実装：変数違いの複数バージョンやトラック自体の配列化
- オーディオデータのインポートや、各種連携・通信機能は後から

# 世の中への出し方

- 言語仕様、コンパイラのソースコード：オープンソース（LGPL）
- GUIアプリケーション部分：はじめはオープンソース、将来的には商用化？
- 自分が使うためのツールであることを最優先にしたうえで、ただソースが公開されているだけの名ばかりOSSでない、長期的なコミュニティ形成を意識

プロジェクト期間

自分で試用・作品制作

ユーザーテスト・勉強会・ドキュメンテーションの充実

# 目指す：

- 生成的な表現をUIの補助で効率的に編集できること
- 音楽の構造をプログラミング言語という形で定性的に表すこと
- それによって生まれる新しい音楽表現を作り出すこと！

# 目指さない：

- 既存のDAWや音楽プログラミング環境を置き換えるものを作ること
  - ▶ 道具にはそれぞれ得意不得意がある
- 音楽制作・音楽プログラミングを誰でもできるもの・簡単にすること
  - ▶ 生成的な表現にはそれ相応の知識がいる/「誰でも」できるは往々にして「誰かの」劣化版でしかない

# 現状と懸念事項

- 今のところ、まだ下調べの段階（UIのライブラリ、コンパイラの実装など）
- コンパイラ制作に技術的課題がたくさん出そうな予感
- 一旦別の言語（Extemporeなど） + Webブラウザでプロトタイピングするべき…？

# ソースコードの編集ツールとしての 音楽制作ソフトウェアの制作

九州大学大学院 芸術工学府 博士課程1年

松浦 知也

2019/04/20 未踏IT人材発掘・育成事業 二次審査

me@matsuuratomoya.com