IPA

2019 年度 未踏 IT 人材発掘・育成事業 採択案件評価書

1. 担当 PM

竹迫 良範（株式会社リクルートテクノロジーズ 執行役員）

2. クリエータ氏名

松浦 知也 (九州大学 大学院芸術工学府 芸術工学専攻 博士後期課程)

3. 委託金支払額

2,047,600 円

4. テーマ名

プログラマブルな音楽制作ソフトウェアの開発

5. 関連 Web サイト

- 開発リポジトリ URL：https://github.com/mimium-org/mimium
- 開発者個人 Web サイト：https://matsuuratomoya.com

6. テーマ概要

本プロジェクトでは、新しい音楽プログラミング言語“mimium”の設計及びその処理系の開発を行った。mimium は音声信号処理のような低レイヤーの処理から、楽譜のような高レイヤーの制御処理を一つの言語体系の中で完結に記述でき、その上で JIT コンパイラによる高速な処理を実現するものである。

7. 採択理由

本提案は、音楽と音響の両方を表現できる新しいプログラミング言語を設計し、ソースコードのテキストと GUI で双方向に編集可能な音楽制作ツールを開発する野心的なプロジェクトである。電子楽器の発展によりシンセサイザーや MIDI シーケンサを使った音楽演奏が当たり前となり、GS 音源や GM 音源などの標準規格が登場することで、DTM による作曲も行われるようになった。MML（Music Macro Language）で音楽をプログラミングする文化はそこで育ってきたと言える。その後、PC の性能が向上し、GUI ソフトウェアを使ってマルチトラックのオーディオ合成が柔軟にできるようになり、DAW による楽曲編成も一

般的となった。

最近は、アルゴリズムとレイブを組み合わせた Algorave という造語も生まれ、プログラミングと DJ プレイや VJ 表現を組み合わせたライブコーディングによる音楽表現も発展しつつある。たとえば TidalCycles では数十行のプログラムを書いて、ソースコードを直接編集することで音楽の演奏をリアルタイムに変えることができる。本プロジェクトによって開発される新しい楽曲編集システムによって多様な音響音楽表現をリアルタイムに制御できるようになり、最近の DAW による音楽編集では成し遂げられない新しい音楽表現の発展を期待して採択した。

8. 開発目標

本プロジェクトの当初の開発目標は以下の通りであった。

(1) 音楽プログラミング言語仕様の策定・技術調査
音楽プログラミング言語の仕様の策定と、そのための技術調査を行う。本言語では、時間の型を持ち、MIDI のようなイベント処理とオーディオのようなストリームの処理を他言語に頼らず一つの言語体系で記述でき、また冗長性の排除よりもテキストとしての可読性を高め、ソースコードが楽譜のようにも機能することを目指す。
(2) 音楽プログラミング言語の実行環境の開発
(1).で策定した音楽プログラミング言語の実行環境を実装する。本実行環境は、オーディオ処理のためにパフォーマンスが求められることと、LLVM を使用することで異なるハードウェアやプラットフォームでの実行が可能になることから、C++を使った実装を想定する。
(3) ソースコード編集ツールの開発

(1).のプログラミング言語のソースコードを読み込み、その構造を一般的な DAW のように時間軸上に複数のトラックが存在し、その中に時間範囲を指定して配置されたオブジェクトとして表示し、ソースコードと双方向に編集できるソフトウェアを実装する。本編集ツールでは、マウスを用いた波形やリズムパターンのアルゴリズミックな生成を支援する GUI、出力を音声ファイルや MIDI パターンとして書き出す機能などの実現を目指す。

9. 進捗概要

実際に上記の開発目標を達成するため、9 月時点で(1), (2), (3)までのプロトタイピングを実装した。しかし、実装上の技術スタックの問題でパフォーマンス上の課題があることが判明し、実用に耐えないことが発覚したため、途中で開発目標を変更し、(1)と(2)の言語仕様の策定と高速な処理系開発に注力することと

した。新しい開発目標として「インフラストラクチャーとしての音楽プログラミング言語」という以下のコンセプトを策定した。

- 楽譜レベルと信号処理レベルの記述が一つの言語体系で書け、リアルタイム実行できる事
- C や JavaScript といった汎用言語のシンタックスからそれほど逸脱せず、読みやすい言語である事
- 将来的にネイティブアプリケーションとしてだけでなく、ブラウザ上での実行や、マイクロコントローラ上での実行など、多様なプラットフォームで動作する事

リアルタイムでの信号処理を実現することを目指すため、また後々のプラットフォームの拡大を容易にするため、LLVM コンパイラ基盤を導入した。最終的に本プロジェクトで開発した mimium と他の音楽プログラミング言語の特徴を比較したものを表 1 に示す。

表 1：mimium と他の音楽プログラミング言語の特徴比較

	Max	SuperCollider	Extempore	Faust	ChucK	mimium
信号(速い)処理	○(Max)	○(hoge.ar())	○(xtlang)	○	○	○
フルスクラッチ信号処理	△(Gen)	×	○	○	△(拡張)	○
制御(遅い)処理	○(MSP)	○(hoge.kr())	○(Scheme)	×	○	○

mimium が他の言語と比べて特徴的なのは、下記 3 点を全て満たすことである。

- 楽譜ではなく信号処理レベルで音を記述可能
- 信号処理をフルスクラッチで、尚且つパフォーマンスを損なわず記述可能
- 楽譜のような制御処理も同じ言語の中で記述可能

コンパイラとは別に、開発にメインで使用していたエディタである Visual Studio Code 向けのシンタックスハイライトを作成した（図 1）。これは Visual Studio Code のマーケットプレイスからダウンロード可能になっている。

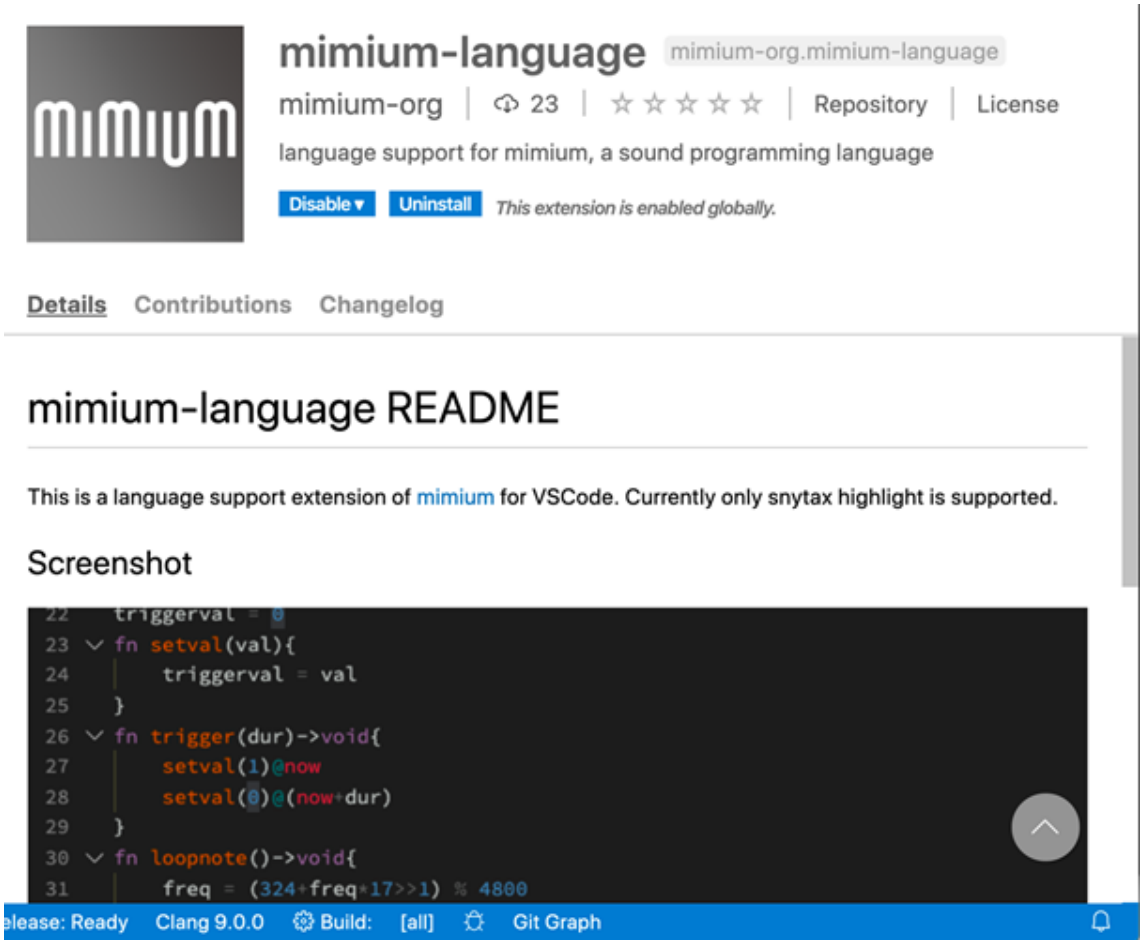

図 1：Visual Studio Code 用の mimium 言語のシンタックスハイライト拡張機能

10. プロジェクト評価

新しい音楽プログラミング言語 mimium をゼロの状態から設計し、C++と LLVM API を用いて高速な JIT 言語処理系の実装を行った。言語仕様も十分考えられたものであり、音声信号処理のような低レイヤーの処理から、楽譜のような高レイヤーの制御処理を一つの言語体系の中で完結に統一的に記述できることを証明した。この言語を利用して、今までの五線譜の枠に囚われない新しい音楽表現が作られることが期待される。

11. 今後の課題

mimium は音をプログラムで生成するための最低限のプリミティブな言語仕様を満たしているが、配列やタプル、モジュールなど言語仕様そのものの拡張の検討が今後の課題である。現状は macOS のみで動作するが、今後幅広いユーザに使ってもらうためには Windows、Linux など別の OS にも対応していく必要があり、さらに GUI による操作ができるようになると理想である。新しいコンセプトを実現している音楽プログラミング言語でもあるため、コンセプトを論文としてまとめ、国際学会で発表することも期待している。