

コンピュータサウンド教育のためのインタラクティブ環境について —Max, SuperCollider, Processing, SonicPi, およびMozziの検討—

長嶋 洋一†

†静岡文化芸術大学デザイン学部 〒430-8533 静岡県浜松市中区中央2-1-1

E-mail: †nagasm@suac.ac.jp

あらまし メディアアートに関連したデザイン教育の場において、ビジュアル要素と並んで重要な要素の「サウンド」をデザインする教育のための教材として、4種のインタラクティブ・サウンド処理環境について整理検討した。映像作品やコンテンツBGMなどのサウンドに既存のネット上フリー素材を安易に利用する学生が多いなかで、自己の感性を頼りにオリジナルのサウンド生成を求めたり、インタラクティブ作品の「サウンドセンサ」(状況分析)として周囲音響のリアルタイム解析を目指す少数の学生もいる。サウンド処理のためのインタラクティブ環境を活用すると、アルゴリズムによっては先入観を超えた新規なサウンド創生のアイデアの可能性もある。本稿では、Max(Max8)、SuperCollider(Version3)、Processing(Minimライブラリ)、Sonic-Pi、という4種のリアルタイム・インタラクティブ・サウンド処理環境を例として、簡単な事例紹介とともにその特徴や発展性などについて整理する。関連して、外部ハードウェアを用いずにArduino内部処理のみでの単体サウンド生成を実現したMozziについても紹介したい。

キーワード Max, SuperCollider, Processing, SonicPi, Mozzi

About Interactive Environments for Computer Sound Education —Consider Max, SuperCollider, Processing, SonicPi, and Mozzi—

Yoichi NAGASHIMA†

†Shizuoka University of Art and Culture 2-1-1 Chuo, Naka-ku, Hamamatsu, Shizuoka 430-8533 Japan

E-mail: †nagasm@suac.ac.jp

Abstract Four types of interactive sound processing environments were examined as educational materials for designing "sound," which is as important as visual elements in design education related to media art. While many students easily use existing free materials on the Internet for sound for video works and content background music, there are a few students who seek to generate original sounds based on their own sensibilities or to analyze the surrounding sound in real time as a "sound sensor" (situation analysis) for interactive works. Utilizing an interactive environment for sound processing, there is potential for novel sound creation ideas that transcend preconceived notions, depending on the algorithm. In this paper, I will take as examples four real-time interactive sound processing environments: Max (Max8), SuperCollider (Version3), Processing (Minim library), and Sonic-Pi, and will organize their features and development potential, along with brief case studies. In a related topic, I would like to introduce Mozzi, a stand-alone sound generation system that uses only Arduino internal processing without any external hardware.

Keywords Max, SuperCollider, Processing, SonicPi, Mozzi

1.概要

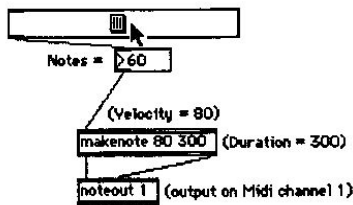
メディアアートに関連したデザイン教育を進めている。昨今のコンピューティング・パワーの向上によって、グラフィックやアニメーションや3Dレンダリングなどのビジュアル要素のデザインは飛躍的に容易になってきているが、同様に重要な要素である「サウンド」をデザインするための道具も充実してきている。映像作品やゲーム等のコンテンツのBGMやSE(効果音)などのサウンドについても、Mac付属「Garageband」のような「超お手軽作曲ツール」やネット上のフリー素材を安易に利用する学生が増えている。その一方で、古典的なサウンド・シンセサイザの歴史から「楽音合成」「音響処理」のテ

マで自己の感性を頼りにオリジナルのサウンド生成を求めたり、インタラクティブ作品の「サウンドセンサ」(状況分析)として周囲音響のリアルタイム解析の活用を目指す少数の学生もいる。このサウンド処理のためのプログラミング環境を活用すると、インタラクティブなインスタレーションやパフォーマンスを実現することができ、アルゴリズムによっては先入観を超えた新規なサウンド創生のアイデアの可能性もある。本稿では、教育のための教材として、Max(Max8)、SuperCollider(Version3)、Processing(Minimライブラリ)、Sonic-Pi、という4種のリアルタイム・インタラクティブ・サウンド処理環境を例として、簡単な事例紹介とともにその

特徴や発展性などについて整理する。関連して、外部ハードウェアを用いずにArduino内部処理のみでの単体サウンド生成を実現したMozziについても紹介する。

2.Max8

デザインの世界では3D-CGソフトの“3ds Max”[1]と間違える学生も少なくないが、ここでの“Max”[2]は全くの別物で、より歴史の長い「visual programming language」で、フランスの国立音楽音響研究機関IRCAMで開発された試作版Maxに筆者が触れたのは発表前の1989年であった[3]。その数年後に筆者が共立出版「bit」別冊「コンピュータと音楽の世界」の中でMaxを紹介[4]したバージョン“Max2”のMaxパッチは、25年後の最新バージョン“Max8”[5]でもそのまま走る。ちょっとバージョンが上がると過去のデータが使えなくなるソフトウェアがあまりに多い現代、貴重な知的財産として世界中の研究者/音楽家が築き上げてきたMaxパッチの上位互換性がこれほど維持されつつ機能拡張が続いているのは驚異的である。なお本稿では紙面の関係でMax8については簡単に述べるしかないので、本稿のベースとしてWeb上で公開しているページ[6]や本稿の高解像度版PDF[7]や、末尾の「参考文献/リンク等」に並べたURLを参照されたい。



過去に“Max”とタイトルに入れた学会発表を筆者のサイト[8]で調べてみると5件[9-13]しか無かったが、実はプラットフォームとしてMaxを空気のように活用している研究/発表は数知れず、[8]の「論文」というカテゴリ218件(2022年6月現在)のうち約150件の中身にMaxが関係している。筆者にとってMaxとは研究プラットフォーム、作曲/公演の相棒、音響生成実験ツール、ビジュアライザ、など言わば「生存環境」なのであり、詳しくは末尾の「参考文献/リンク等」の[9-65](著者は全て筆者)を参照いただきたい。またルーツが音楽関連であったMaxは音楽以外にもラビッド・プロトタイピング・ツールとして最上のシステムであり、筆者の過去の心理学実験[66-68]、バイオフィードバック療法[69]、学生のインスタレーション作品[70]、IPA「未踏」採択プロジェクト[71]、新・筋電センサシステム[72]、新・触覚センサシステム[73]、ジャミナータ改造プロジェクト[74]、新楽器「電子十二影坊」[75]、新楽器「GHI2014」[76]、海外でのワークショップ[77]などでも活躍してきた。SUAC(静岡文化芸術大学)の「産学官連携」スキーム[78]による受託研究(案件として情報公開されているが詳細は契約により伏せる)においても、「ネット遅延のある音楽セッション」(ヤマハ, 2002)、「筋電センサのジェスチャ認識」(メガチップス[任天堂系], 2009)、「車酔い防止のカーオーディオ」(パイオニア, 2010)、「自動運転車での自動作曲」(トヨタ中央研究所, 2015)、「エクササイズEMSと音楽の融合」(Xenoma, 2021)などのテーマの全てでMaxをベースに試作システムを開発/納入してきた。また、COVID-19のために全国の大学で講義が全てリモートとなった2020年前期(4-7月)には、Max8の開発元Cycling'74社が世界中のMaxユーザ(学生、研究者、教育機関)のために期間限定でライセンス無償解除して、筆者もSUACデザイン学科専門科目「サウンドデザイン」(Max8入門)を完全リモートで行ったが、この教材[79]は健在なので、インストールして30日間は無料体験期間となるMax8を誰でもここで体験/学習できる。

Maxのプログラミングは30年前から変わらず「ウインドウ内にオブジェクトを置いてワイヤで連結する」というスタイ

ルであり、最近では[5]のようにGUI機能が著しくゴージャスになっているものの、基本概念は一貫している。ここでは筆者の開発したMaxパッチの事例を3件だけ簡単に紹介しておくが、参考文献リンク[9-68][72-77]の中でもMaxパッチのサンプルは豊富に紹介している。Maxはプログラミング環境なので、ブラウザも作れるし[81]、市販のゲーム機USBコントローラも使えるし[82]、OSコマンドによってOS標準ツールも活用できる[83]。これ以外のMax関連トピックについては筆者のサイト[84-87]を参照されたい。

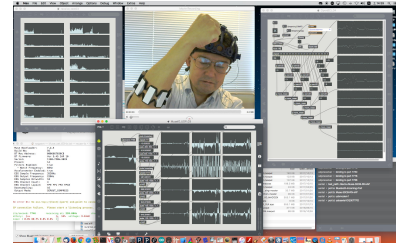


図1

図1は、8チャンネル筋電センサ(+9軸センサ)の“Myo”を本来の片腕使用から拡張ハックして両腕に装着し、額には脳波センサ(+3軸加速度センサ)の“Muse”を装着し、さらに頭部に脳波センサの“OpenBCI”を装着して、それらの生体センシング情報を全てMaxで受け取ってライブ表示している実験の様相である。脳波センシングはBrainインターフェースとして世界的に注目されているが、開眼しているとノイズが大きいために、音楽演奏パフォーマンスのための「楽器」としては大きな困難がある。

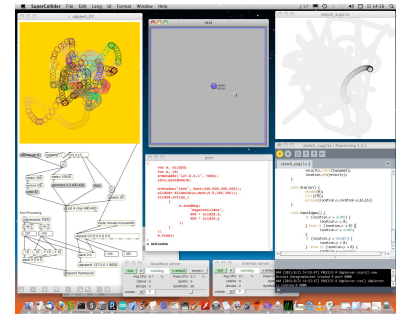


図2

図2は、まだMaxがMax5だった10年以上も昔の実験(国際会議出張中の会場でプログラミング)であるが、同じパソコン内にMax5とProcessingとSuperColliderの3つのソフトウェアを同時に起動させて、カリフォルニア大学バークレー校の研究機関CNMATが開発したプロトコルOSC[80]によって連携させるという実験の様相である。SuperColliderの2次元グラフィカルインターフェース上でマウスアイコンを動かすと、その操作情報がOSCでMax5に飛んでMaxの「lcd」オブジェクト内に描画し、さらにOSCで操作情報がProcessingにも飛んで残像を含むライブ・グラフィックを生成しているが、同時に3つのソフトウェアで関連したサウンドを生成することも容易である。またこれは1台のパソコンで実験したが、OSCによってネットワーク接続された他のパソコンに分業することも容易である。

図3は、トヨタ中央研究所からの受託研究として開発したもので、自動運転車が外界に向けている多種多数のセンサからの情報を受け取って、道路状況や周囲の風景などに対応してリアルタイムにBGMを自動作曲するシステムの試作版として開発したMaxパッチの例である。このシステムは実際にトヨタ中研の実験車に搭載されて筆者も一般道路を走行しての実験に同乗したが、音楽的に妥当な自動作曲システム[53]としての開発の道筋を提示したものの、具体的なセンサのパラメータと生成される音楽に対する感情的な評価との対応については今後の大きな課題も見出した。

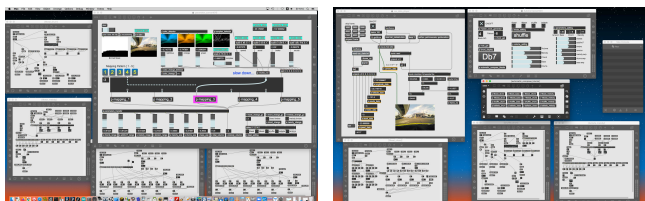


図3

3. SuperCollider (Version3)

SuperCollider [88]とは、1996年にJames McCartney氏が個人として発表した、クライアント-サーバ方式のリアルタイム音響処理システム/アルゴリズム作曲環境である。音響処理部分をサーバとして別途に走らせておいて、クライアントからはOSC経由でパラメータを転送するという優れた構成になっている。開発者のJames McCartney氏は後にAppleにスカウトされて、MacOSのサウンド・コア部分の開発を担当することになったためにSuperColliderのメンテナンスから離れて、現在ではGitHub上のオープンソースとして公開されている [89]。筆者の過去のComputer Music作品公演記録 [90]を調べてみると、2001年のドイツ演奏旅行の際に公演した作品 [91]ではバージョン2のSuperColliderを使って簡単な生成音響グラフィック表示を行っていたが、クライアント側は図4のように既存の自在なGUIを活用することが出来るようである [92]。

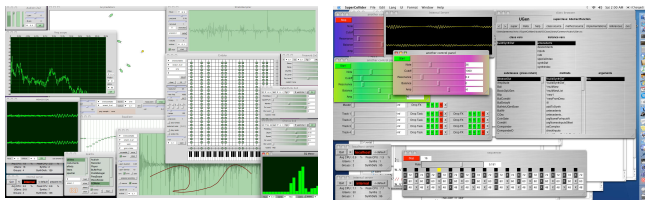


図4

SuperColliderのプログラミング言語は、「Smalltalkに似た動的型付け、ガベージコレクション、単一継承」のオブジェクト指向・関数型言語で、LispやCプログラミング言語に似た構文を持っている [88]。各種の音響処理体系がある中で、世界中に根強い(少数の)SuperCollider信者が活動を続けているが、象徴的なトピックとなったのは、2009年の「SuperCollider 140」 [93]というオープンソース・プロジェクトであった。SuperColliderのプログラムは「半角英数記号」だけで記述されており、スペースやタブや改行は人間が見やすく理解するためのものであって、全部びっしりと詰めても構わない。そこで、当時のツイッターの「140文字制限」という制約を逆手にとって、「140文字だけでどれだけSuperColliderで面白いサウンドを生成できるか」という世界的な大喜利大会となった。このサイト [93]には全部で22個の、たった1行に詰められたSuperColliderプログラムが置かれていて、その下にはこれをSuperColliderサーバに送ると生成されるサウンドが聞けるmp3 Playerが置かれている。びっしり詰めてあるので、まずはこのソースをテキストエディタにコピーして、見やすいようにインデント付きで改行して「解体・解説」してみると、SuperColliderの持つ驚異的な能力を体験することが出来る。筆者が「SuperCollider日記」 [94]としてSuperColliderと遊んでみたのは2011年とだいぶ昔になってしまったが、この中身は現在でも通用するテキストとなっている。

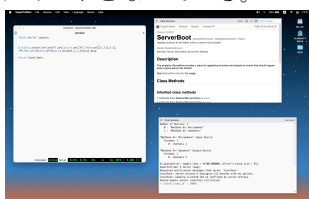


図5

本稿のために「思い出し」として、新たに最新のSuperCollider3.12.2をダウンロードして実験してみたのが図5であり、色々なGUIでデコレーションを志向しない限り、SuperColliderの画面はこのようにシンプルである。最初に「Server.local.boot;」という行を指定して「シフト+リターン」で送るとサーバが裏で起動して、あとは所望のプログラム(ここではsc-140の最初のもの)を指定して「シフト+リターン」で送ると面白いサウンドが鳴り出す・・・というものである。サウンドを止めるには「Command+ピリオド」が必要、というちょっと癖のあるインターフェースなのだが、まあ慣れてしまえば気にならなくなる(久しぶりだと忘却している)。

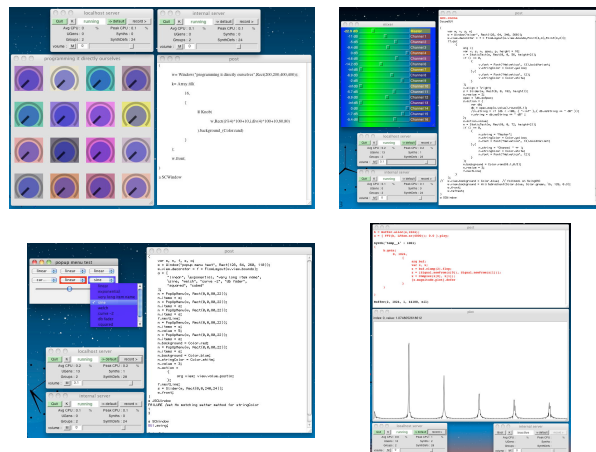


図6

上にあるのは全てSuperColliderに関するスクリーンショットであり、筆者の「SuperCollider日記」 [94] (part1からpart7まであり)から発掘したものである。MaxとProcessingと連携させたProcessingについては、前節「Max8」にある。

4. Processing (Minim)

Processing [95]は2001年に登場した、グラフィックライブラリを伴ったJavaベースのIDE(統合開発環境)であり、最初は「可視化」に優れたツールとして活躍した。しかしコンピューティング技術の発展と共に単なるVisualizeツールから発展を遂げて、前節「Max8」で述べたようにOSCに対応してMaxと繋がったり、さらにサウンド領域でもずんずん発展して、各種の開発環境との仁義なき戦いに参入している。オープンソース文化(ダウンロードに際して「寄付のお願い」があるが、懐が寒い学生は無視することも可能)のために、世界中のアカデミック領域で広く普及していて、特に無料の教育ツールとして発展途上国で強く支持されている。Java言語で記述するProcessingは、一見するとArduino IDEとそっくりのプログラミング環境であるが、これはOpen Source Communityの証明だろう。Javaで記述した「スケッチ」をIDEの「Run」で実行するだけでなく、過去にWebブラウザ内で走る「Javaアプレット」というコンテンツ(ちょうどFlashで作ったswfファイルがFlash Playerプラグインの入ったWebブラウザのウィンドウ内で走るのと同じ)という書き出しも可能だったのはProcessing1.5までである。JavaもFlashも、セキュリティ的にWebブラウザの世界から完全に排除されてしまったので、現在ではこの機能は事実上消滅しているが、2018年5月13日の時点ではFirefoxもChromeもNGだったのに対してSafariだけは走っていたらしい [96]。本稿執筆に際して調べてみると、これまでの安定板Processing3.5.4(2020年1月)からメジャーバージョンアップとしてProcessing4.0beta8(2022年4月)というのがApple Silicon対応まで含めて登場していたが、betaが付いているので本稿では3.5.4について述べる。



図7

筆者が「Processing日記」[97]としてProcessingをざっと調べたのは2011年頃だったのだが、現在のProcessing3.5.4でライブラリを見てみると、図7のようにExamplesのメニューの標準的なライブラリのLibrariesの中に「Sound」というものがあり、Analysis/Effects/Env/IO/Noise/Oscillators/Soundfileというように、ほぼ基本的なサウンド処理が網羅されている。図7のAudioWaveformというスケッチはサウンドファイルをライブ再生しつつ、その信号波形をライブ描画し、BeatDetectionというスケッチはライブ再生している音楽のビート(周期的に繰り返すピーク)を検出して描画し、FFTSpectrumというスケッチはライブ再生しているサウンドのFFT周波数分析結果をグラフとしてライブ描画している。ザッと眺めた範囲では、サウンド処理に関するほぼ全てをこの標準ライブラリですら網羅している。

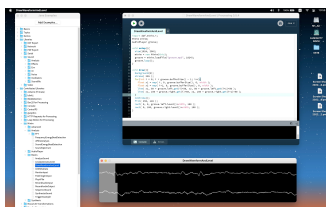


図8

そして標準ライブラリに加えて「Contributed Libraries」という、オープンソースならではの誰かが拡張した新たなライブラリのカテゴリの中にある「Minim」というのが、さらにサウンド関係で強力なメニューを完備させている。図8はそこから一つだけ、もっとも入門編の「Basic」というサブカテゴリの中のDrawWaveformAndLevelというスケッチであるが、より簡単な記述で、ライブ再生しているサウンドの音圧(パワー)とその信号波形の両方をライブ描画している。ここにはさらに別のサブカテゴリとして「Advanced」「Analysis」「AudioPlayer」「Synthesis」というものがある、その全体像としてはさらに網羅的に、かつ効率的にサウンド処理の体系を完備している模様である。大学院生の教材として格好の素晴らしい教育的フリー素材である。

5.Sonic-Pi

Computer Musicの世界では15年ほど前から「Live Coding」というアイデアがブームとなってきた。ライブComputer Musicパフォーマンスにおいては、事前に作られたCDを再生したり固定的シーケンスデータをカラオケ的に再生するのはややダサイ・・・というのが1980年代後半から常識となっていて、各種のセンサを「新楽器」としてライブ音楽生成のパラメータに適用したり、ステージ上での生楽器/歌声サウンドをライブサンプリングしてその場で変容音響を生成するというスタイルが普及している。音楽情報をソフトウェアのプログラムのように記述する「アルゴリズム作曲」の場合には、ソースコード(楽譜)を記述/編集する(作曲)ステップと、このソースを音響生成システムに転送して音響合成(コンパイル)して鳴らすステップとが、交互に繰り返されてきた。しかし「Live Coding」においては、音楽生成記述のコーディングの最中にライブでそれを音響生成する(その画面をスクリーンに投射して聴衆も状況を把握できる)ために、タイプミス/論理的バグの音楽コーディングによって音楽音響生成がステージ上でストップする可能性もある、という緊張感のあるライブパフォーマンスとなる。

本稿で紹介したSuperColliderではコーディングのクライ

アントと音響生成のサーバが分離しているものの、コードをコピーして転送コマンドでサーバに送るというステップが「コンパイル」に相当するので、音楽音響の生成はその段階で停止/分断される。Processingの場合も同様に、音楽関係ライブラリをincludeしたスケッチ(ソース)をIDE内に記述して、「Run」ボタンをクリックすると内部的にコンパイルされて実行されるので、ここでも音楽音響の生成は停止/分断される。一方Maxの場合には、バッチの「編集モード」と「実行モード」とを行き来するので同様に思えるが、実は「編集モード」の最中にも中身の音楽情報やサウンド情報はライブに流れていて、「ステージ上で演奏中にそのMaxバッチを修正しつつ演奏を継続する」というのは1990年代前半から日常茶飯事だった。そしてここで紹介するSonic-Pi[98]とは、SuperColliderやProcessingと同様のテキストコーディング形式の音楽記述システムでありながら「Live Coding」を売りにしている新しいシステムであり、[98]のサイトには多くのライブ・パフォーマンス(コーディングのスクリーン投射を背負ってパソコンに打ち込む姿にはやや違和感もある)の事例YouTube動画がずらりと並んでいる。

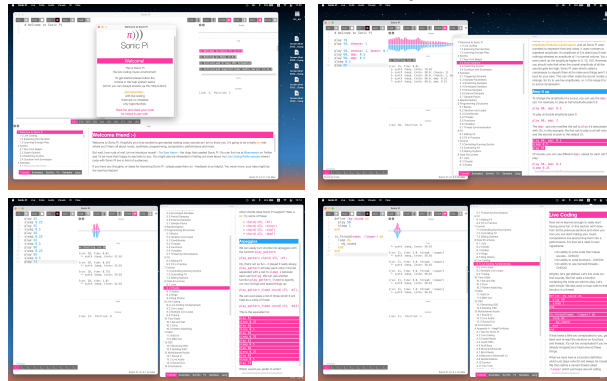


図9

図9は、大学院で筆者に弟子入りを希望し院試に合格したもののCOVID-19のために日本入国を2年間も待たされた院生(留学生)がリモート指導中にSonic-Piを発掘してきて、初めて筆者がSonic-Piを触ってみた日の様子である。後発のシステムであるだけに、ヘルプやチュートリアルが非常に充実していて、あっという間にこのシステムの特性と、その生成する音響の素晴らしさを体験することとなった。



図10

図10は、その留学生がようやく来日を果たして筆者のゼミと一緒に実験した日の様子で、チュートリアルを日本語化しつつ勉強して、さらにMax8からOSC経由でライブ音楽音響生成のパラメータを制御するという実験を進めたところである。全体的にピンクの色彩が満載のSonic-Piというのは、筆者にとって過去にモノクロだったMaxとかグレースケールを基調とする渋いSuperColliderに比べてちょっと違和感があるのだが、若者にとってはコンピュータ音楽を学んだり「Live Coding」に入るための絶好のツールとなりそうな予感がある。

6.Mozzi

ここまで紹介したMax8、SuperCollider、Processing、Sonic-Pi、はいずれもパソコン上でリアルタイム音楽音響生成するシステムだったが、ここで最後に紹介するMozzi[100]は、「Arduino単体でサウンドする」という領域の新しい潮流である。これまでArduinoでサウンドと言えば、デジタル出力ポートに圧電スピーカを繋いで高速(可聴周波数)でON/OFFする「ブザー音」とか、ADPCM音声情報を必死に出力する(他に何の仕事も出来ない)音声合成がある程度で、あとはArduinoシールド(拡張搭載基板)によって「MIDI音源」・「mp3サウンド生成」などの外部回路を追加して実現するしかなかった。ArduinoシールドはArduino UNOのサイズになっているので、インсталレーション作品としてサウンドを扱う場合には、そのサイズの限界が困難だった。最近ではより小型の「DFR0299」という高性能で興味深い超小型mp3再生ボード[99]も出てきたが、本稿では深入りを避ける。また技術的な背景としては、参考文献[101]を参照されたい。

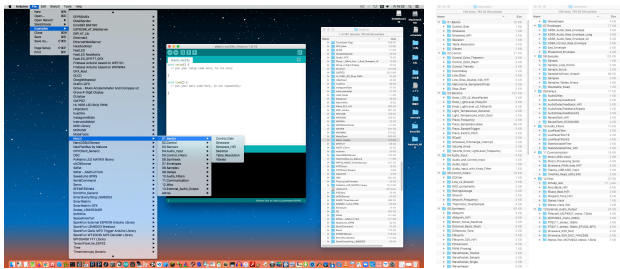


図11

Mozziのサイト[100]を一見すれば分かるように、ここもオープンソース文化の賜物であり、非常に高度な音響信号処理をブラックボックスとしてArduinoで実現するという挑戦的なプロジェクトである。Mozziギャラリー[102]にあるYouTube動画デモを眺めてみるだけで、メディアデザインを志す学生などは大きく惹かれること間違いない。ライブラリは全てGithubに公開されていて[103]、図11のようにArduino IDEのExampleメニューからその一覧を眺めてみるだけで、これまでMaxとかSuperColliderとかProcessing(Minim)とかパソコンベースで実現されてきた音響関係のほとんどの処理をArduinoで実現してしまうという意欲が伝わってくる。

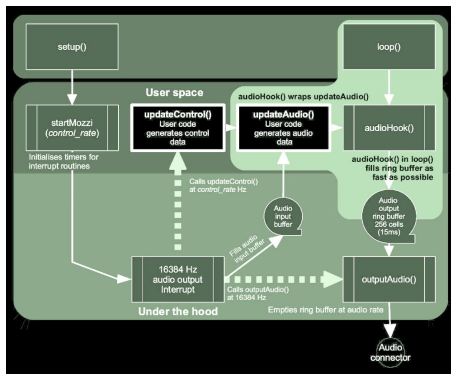


図12

図12は、そのMozziサイトの中の「learn」というページ[104]で解説されているMozziの内部動作の詳細である。普通にC言語でArduinoプログラミングするという世界とは全く異なり、内部的に厳しい割り込みを活用した音響処理エンジンを実装している。サウンドドライバ部分のリアルタイム制御と、ヒューマンインターフェース(センサ等)制御とでは扱うレベルを別にしていて、この発想はMaxと兄弟分である、IRCAMから派生したシステム「PureData」そのものと言える。筆者の独断的な印象だが、このMozziの「中の人」には、間違いなくIRCAMや、スタンフォード音楽研究所CCRMAの関係者がいる、と確信している。

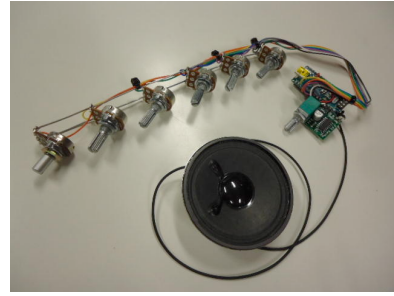


図13

図13は、Mozziを知った翌日に、筆者がArduino NanoにMozziを実装し、小型アンプ(この音量が不要であればArduinoピンからスピーカ直結も可能)と6個のボリュームを繋いで試作した実験機である。このシステムからスタンドアロン(パソコン無し)でどんなサウンドが出てくるかは、YouTubeデモ動画[105]でぜひ確認していただきたい。このMozziは既に筆者が支援するSUACデザイン学科学生のインсталレーション作品として実装されている[70]。

7.おわりに

メディアアートに関連したデザイン教育のための教材として、Max(Max8)、SuperCollider(Version3)、Processing(Minimライブラリ)、Sonic-Pi、という4種のインタラクティブ・サウンド処理環境について紹介した。また関連して、外部ハードウェアを用いずにArduino内部処理のみでの単体サウンド生成を実現したMozziについても紹介した。デザインの学生が本格的/高価なツールを使う一方で、「お手軽無料アニメ制作ソフト」と検索するだけで、Shotcut/AviUtl/Blender/9VAeきゅうべえ/AnimeEffects/OpenShotと6種類も出てきてしまい、かつて精緻な職人芸だった画像処理もスマホのアプリで簡単に出来てしまう時代である。音楽についても、試みに「お手軽ツール 作曲 無料」と検索すると、Garagebandを含めて10種類あるいは16種類を提案するサイトが出てきた。ネット上のフリー素材(画像/ムービークリップ/効果音/楽曲など)を活用する文化も浸透してきて、デザインの学生なのに、肝心の自分のマルチメディア作品に安易にフリー素材を使いつつクレジット表記すれば平気な顔、という驚くべき時代にもなってきた。創造性・創造力とは何なのか、オープンソース文化でSuperColliderやProcessingやSonic-PiやMozziなど膨大な知的世界をフリー公開する原動力は何なのか、改めて考え直したい。ちなみにMax8は米国Cycling'74社が開発提供する「商用ソフト」であるが、CEOでMaxの全てをプログラミングしているDavid Zicarelli氏はIRCAMでComputer Musicを研究していた研究者であり、師匠のMax Mathews博士の名をとって登場したMaxを世界に普及させるために起業したのであり、メンテナンス/開発のスタッフ経費としてのコストのみでアカデミック普及を支援しており、精神としてはこれまたオープンソース文化であることを補足/指摘しておきたい。

参考文献/リンク等

1. Autodesk 3ds Max. https://en.wikipedia.org/wiki/Autodesk_3ds_Max
2. Max (software). [https://en.wikipedia.org/wiki/Max_\(software\)](https://en.wikipedia.org/wiki/Max_(software))
3. Max前夜. <https://nagasm.org/ASL/max02/index.html>
4. アルゴリズム作曲. <https://nagasm.org/ASL/max01/index.html>
5. Max. Cycling'74. <https://cycling74.com/products/max>

6. コンピュータサウンド教育のためのインタラクティブ環境について. <https://nagasm.org/ASL/202208EA/index.html>
7. 本稿の高解像度版. <https://nagasm.org/ASL/paper/202208EA.pdf>
8. Art & Science Laboratory. <https://nagasm.org/ASL/>, <https://nagasm.org/ASL/ASL.html>
9. 生体センサとMax4/MSP2による事例報告. 情報処理学会研究報告(2002-MUS-44), 情報処理学会, 2002. <https://nagasm.org/ASL/SIGMUS0202/index.html>
10. インタラクティブアートの統合的システム・プラットフォームとしてのMax/MSP. DSPサマースクール2002論文集, 2002. <https://nagasm.org/ASL/dspss2002/index.html>
11. Max/MSPとKymaとLabVIEWによる音響処理について. 平成15年度全国大会講演論文集2, 情報処理学会, 2003. <https://nagasm.org/ASL/paper/IPSJ0303.pdf>
12. 基礎心理学実験プロトタイプツールとしてのMax7とウェルネスエンタテインメントプラットフォームとしてのMax7. 情報処理学会研究報告(2018-MUS-120), 情報処理学会, 2018. <https://nagasm.org/ASL/paper/SIGMUS201808.pdf>
13. 音楽情報科学ツール“Max”を用いたメディアデザイン - RFIDの活用例を中心として. 情報処理学会研究報告(2019-MUS-124), 情報処理学会, 2019. <https://nagasm.org/ASL/paper/sigmus201908-1.pdf>
14. マルチメディア・インタラクティブ・アート開発支援環境と作品制作・パフォーマンスの実例紹介. 情報処理学会研究報告(95-MUS-16), 情報処理学会, 1996. <https://nagasm.org/ASL/paper/ss1996.pdf>
15. [広義の楽器]用ツールとしてのMIDI活用. 情報処理学会研究報告(96-MUS-18), 情報処理学会, 1996. <https://nagasm.org/ASL/midi01/index.html>, <https://nagasm.org/ASL/paper/SIGMUS199612.pdf>
16. インタラクティブ・マルチメディア作品“Asian Edge”について(研究ノート). 京都芸術短期大学紀要[瓜生]第19号1996年, 京都芸術短期大学, 1997. <https://nagasm.org/ASL/asian/index.html>
17. Real-Time Interactive Performance with Computer Graphics and Computer Music. Proceedings of the 7th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Man-Machine Systems, IFAC, 1998. <https://nagasm.org/ASL/paper/ifac98.pdf>
18. インタラクティブ・メディアアートのためのヒューマンインターフェース技術造形. 静岡文化芸術大学紀要・第1号2000年, 静岡文化芸術大学, 2001. <https://nagasm.org/1106/news/docs/suac2000.pdf>
19. 新・筋電センサ“MiniBioMuse-III”とその情報処理. 情報処理学会研究報告(2001-MUS-41), 情報処理学会, 2001. <https://nagasm.org/ASL/SIGMUS0108/index.html>, <https://nagasm.org/ASL/paper/SIGMUS200108.pdf>
20. Composition of “Visional Legend”. Proceedings of International Workshop on “Human Supervision and Control in Engineering and Music”, 2001. <https://nagasm.org/ASL/kassel/index.html>
21. “IMPROVISESSION-II” : A Performing/Composing System for Improvisational Sessions with Networks. Proceedings of International Workshop on Entertainment Computing, 2002. <https://nagasm.org/ASL/iwec2002/index.html>
22. Interactive Multi-Media Performance with Bio-Sensing and Bio-Feedback. Proceedings of International Conference on Audible Display, 2002. <https://nagasm.org/ASL/paper/ICAD2002.pdf>
23. GDS Music--- ネットワーク遅延を伴う音楽セッション・モデル. 情報処理学会研究報告(2001-MUS-46), 情報処理学会, 2002. <https://nagasm.org/ASL/GDSM/index.html>, <https://nagasm.org/ASL/paper/SIGMUS200207.pdf>
24. 宇宙人音楽と人体音楽の作曲事例報告. 情報処理学会研究報告(2003-MUS-49), 情報処理学会, 2003. <https://nagasm.org/ASL/SIGMUS0302/index.html>, <https://nagasm.org/ASL/paper/SIGMUS200302.pdf>
25. メディア・アートと生体コミュニケーション. 静岡文化芸術大学紀要・第3号2002年, 静岡文化芸術大学, 2003年. <https://nagasm.org/ASL/suac2002/index.html>
26. Bio-Sensing Systems and Bio-Feedback Systems for Interactive Media Arts. Proceedings of 3rd International Conference on New Interfaces for Musical Expression, NIME, 2003. <https://nagasm.org/ASL/NIME03/index.html>
27. SCANNED SYNTHESISのための身体動作コントロール. AES東京コンベンション2003講演論文集, 2003. <https://nagasm.org/ASL/AES2003/index.html>
28. GDS (Global Delayed Session) Music - new improvisational music with network latency. Proceedings of 2003 International Computer Music Conference, ICMC, 2003. <https://nagasm.org/ASL/paper/icmc2003-2.pdf>
29. 音楽的ビートが映像的ビートの知覚に及ぼす引き込み効果. 芸術科学会論文誌 Vol.3 No.1, 芸術科学会, 2003. <https://nagasm.org/ASL/beat/index2.html>
30. 音楽インターフェースとしてのScanned Synthesis. 情報処理学会研究報告(2004-MUS-55), 情報処理学会, 2004. <https://nagasm.org/ASL/paper/SIGMUS0405.pdf>
31. 欧州のComputer Music研究の状況報告. 情報処理学会研究報告(2004-MUS-58), 情報処理学会, 2004. <https://nagasm.org/ASL/paper/SIGMUS0412.pdf>
32. コンテンツクリエイターのための著作権フリー音楽クリップ生成システム“FMC3”. 芸術科学会論文誌 Vol.5 No.3, 芸術科学会, 2006年. <http://www.art-science.org/journal/v5n3/index.html>, <https://nagasm.org/FMC3/index.html>
33. メディアコンテンツ・デザイン教育におけるコンピュータサウンドの活用事例. 情報処理学会研究報告(2007-MUS-72), 情報処理学会, 2007年. <https://nagasm.org/ASL/paper/sigmus0710.pdf>
34. サウンドの空間的予告による映像酔いの抑止について. 情報処理学会研究報告(2007-MUS-73), 情報処理学会, 2007. <https://nagasm.org/ASL/paper/sigmus0712.pdf>
35. 新しい筋電楽器のジェスチャ・表現の検討について. 日本音楽知覚認知学会2010年春季研究発表会資料, 日本音楽知覚認知学会, 2010年. <https://nagasm.org/ASL/paper/onchi201005.pdf>
36. カオスに対する聴覚的なアプローチ(1). 電子情報通信学会非線形問題研究会(NLP)研究会資料(技術研究報告) NLP2011-158, 電子情報通信学会, 2012. <https://nagasm.org/ASL/paper/201203NLP.pdf>
37. 身体に加わる加速度とサウンドの音像移動に関する心理学実験報告(2/4). 情報処理学会研究報告(2012-MUS-95), 情報処理学会, 2012. <https://nagasm.org/ASL/paper/sigmus201206.pdf>
38. Computer Musicパフォーマンスはこの20年間で進歩したのか. 情報処理学会研究報告(2012-MUS-96), 情報処理学会, 2012. <https://nagasm.org/ASL/paper/sigmus201208.pdf>
39. 聴覚的クロノスタシスに関する実験報告, 日本音楽知覚認知学会2014年春季研究発表会資料, 日本音楽知覚認知学会, 2014. <https://nagasm.org/ASL/paper/onchi201405.pdf>
40. カオスに対する聴覚的なアプローチ(2). 電子情報通信学会非線形問題研究会(NLP)研究会資料(技術研究報告) NLP2014-44, 電子情報通信学会, 2014. <https://nagasm.org/ASL/paper/201407NLP.pdf>
41. 筋電センサのジェスチャ認識に関する新手法. 電子情報通信学会パターン認識・メディア理解研究会(PRMU)研究会資料(技術研究報告) PRMU2015-54, 電子情報通信学会, 2015. <https://nagasm.org/ASL/paper/PRMU201506.pdf>
42. Assembling Music. Proceedings of 2nd International Symposium on Sound and

- Interactivity, SI15, 2015. https://nagasm.org/ASL/paper/SI2015_nagasm.pdf
43. お触り楽器. 情報処理学会研究報告(2015-MUS-108), 情報処理学会, 2015. https://nagasm.org/ASL/paper/SIGMUS201508_1.pdf
 44. 新楽器へのアプローチ. 情報処理学会研究報告(2015-MUS-108), 情報処理学会, 2015. https://nagasm.org/ASL/paper/SIGMUS201508_2.pdf
 45. 内受容感覚バイオフィードバックによる“癒し系エンタテインメント”の考察. エンタテインメントコンピューティング2015論文集, EC2015実行委員会, 2015. <https://nagasm.org/ASL/paper/EC2015.pdf>
 46. 脳波センサ”MUSE”は新楽器として使えるか. 情報処理学会研究報告(2015-MUS-110), 情報処理学会, 2016. <https://nagasm.org/ASL/paper/SIGMUS201603.pdf>
 47. 脳波バンド”MUSE”による心理計測の可能性について, 日本音楽知覚認知学会2016年春季研究発表会資料, 日本音楽知覚認知学会, 2016年. <https://nagasm.org/ASL/paper/onchi201605.pdf>
 48. 皮膚から音を聞く可能性・第2弾”. 情報処理学会研究報告(2016-MUS-111), 情報処理学会, 2016. <https://nagasm.org/ASL/paper/SIGMUS201605.pdf>
 49. Multi Rubbing Tactile Instrument. Proceedings of International Conference on New Interfaces for Musical Expression, NIME, 2016. https://nagasm.org/ASL/paper/NIME2016_nagasm.pdf
 50. Towards the BioFeedback Game --- with Interoception and Rehabilitation ---. Proceedings of the 8th International Conference on Virtual Worlds and Games for Serious Applications, 2016. <https://nagasm.org/ASL/paper/VW-Games2016.pdf>
 51. Bio-Sensing and Bio-Feedback Instruments --- DoubleMyo, MuseOSC and MRTI2015 ---. Proceedings of 2016 International Computer Music Conference, ICMC, 2016. https://nagasm.org/ASL/paper/ICMC2016_nagasm.pdf
 52. 生体情報センシングのバイオフィードバック療法への応用について. 知覚情報研究会・研究報告, 電気学会, 2017. <https://nagasm.org/ASL/paper/IEE2017nagasm.pdf>
 53. 自動運転車のためのリアルタイム作曲システムに向けて. 情報処理学会研究報告(2017-MUS-118), 情報処理学会, 2018. <https://nagasm.org/ASL/paper/SIGMUS201802.pdf>
 54. 生体情報センシングと内受容感覚コミュニケーションの可能性について. 電子情報通信学会ヒューマンコミュニケーション基礎研究会資料(技術研究報告) HCS2017-102, 電子情報通信学会, 2018. <https://nagasm.org/ASL/paper/HCS201803.pdf>
 55. Realtime Musical Composition System for Automatic Driving Vehicles. Proceedings of 2018 International Conference on Entertainment Computing, IFIP TC14 ICEC, 2018. https://nagasm.org/ASL/paper/ICEC2018_nagasm.pdf
 56. PC環境での心理学実験におけるレイテンシとジッタの再検証”. 日本音楽知覚認知学会2019年春季研究発表会資料, 日本音楽知覚認知学会, 2019. <https://nagasm.org/ASL/paper/onchi201906.pdf>, https://nagasm.org/ASL/Latency_Jitter/index.html
 57. 音楽心理学実験ツールとしてのPC環境性能の再検討. 情報処理学会研究報告(2019-MUS-123), 情報処理学会, 2019. <https://nagasm.org/ASL/paper/sigmus201906.pdf>, https://nagasm.org/ASL/Latency_Jitter/test.html
 58. ウェルネス・エンタテインメントのための錯覚体験システム ~ 聴覚やマルチモーダル錯覚を中心として ~. 電子情報通信学会ヒューマン情報処理研究会資料(技術研究報告) HIP2019-87, 電子情報通信学会, 2020. <https://nagasm.org/ASL/paper/HIP202002.pdf>
 59. Jean-Claude Risset 温故知新. 情報処理学会研究報告(2020-MUS-128), 情報処理学会, 2020. <https://nagasm.org/ASL/paper/SIGMUS202008.pdf>
 60. Risset Rhythm 温故知新. 日本音楽知覚認知学会2020年夏季研究発表会資料, 日本音楽知覚認知学会, 2020. <https://nagasm.org/ASL/RissetRhythm/index.html>
 61. インタラクティブな錯覚体験システムの試作報告. 電子情報通信学会ヒューマンコミュニケーション基礎研究会資料(技術研究報告) HCS2020-49, 電子情報通信学会, 2020. <https://nagasm.org/ASL/paper/Hcs202010.pdf>
 62. 脳波センサ”Muse 2”・”Muse S”は新楽器として使えるか. 情報処理学会研究報告(2020-MUS-129), 情報処理学会, 2020. <https://nagasm.org/ASL/paper/SIGMUS202011.pdf>
 63. ウェルネス・エンタテインメントのためのインタラクティブな錯覚体験システムに向けて. 電子情報通信学会ヒューマンコミュニケーション基礎研究会資料(技術研究報告) HCS2021-29, 電子情報通信学会, 2021. https://nagasm.org/ASL/paper/HCS202108_HQ.pdf
 64. 『Real Time Risset Rhythm Generator』から『Live Sampling Risset Rhythm』への道. 情報科学技術フォーラム2021講演論文集, 情報処理学会・電子情報通信学会, 2019. https://nagasm.org/ASL/paper/FIT2021_nagasm.pdf
 65. 新・生体センサシステム”EmotiBit”は新楽器として使えるか. 情報処理学会研究報告(2021-MUS-132), 情報処理学会, 2021. <https://nagasm.org/ASL/paper/SIGMUS202109.pdf>
 66. 音楽的ビートが映像的ビートの知覚に及ぼす引き込み効果. <https://nagasm.org/ASL/beat/index.html>
 67. グロッケン音色の利用に関する考察. <https://nagasm.org/ASL/Glocken/index.html>
 68. 聴覚的クロノスタシスに関する研究. <https://nagasm.org/ASL/Chronostasis/index.html>
 69. バイオフィードバック療法セミナー. <https://nagasm.org/ASL/BFseminar20151012/index.html>, <https://nagasm.org/ASL/BFseminar20160228/index.html>, <https://nagasm.org/ASL/BFseminar20180331/index.html>
 70. SUACインストール. <https://nagasm.org/1106/installation/index.html> (part1-5)
 71. IPA「未踏」開発成果. 情報処理推進機構, 2005. https://www.ipa.go.jp/jinzai/esp/mitoipedia/seika/year/2005_1seika.html
 72. 新・筋電センサシステム「VPP-SUAC」. <https://nagasm.org/Sketching/VPP-SUAC.html>
 73. 新・触覚センサシステム「PAW-double」. <https://nagasm.org/Sketching/PAW-double.html>
 74. ジャミネータと遊ぼう. <https://nagasm.org/ASL/Jaminator/index.html>
 75. 電子十二影坊(Dodeca Propeller). <https://nagasm.org/ASL/12Propeller/index.html>
 76. 新楽器「GHI2014」. <https://nagasm.org/ASL/GHI2014/index.html>
 77. Bio-sensing Platforms for “Wellness Entertainment” System Design. ICEC2018 Tutorial Workshop, 2018. <https://nagasm.org/ICEC2018workshop/index.html>
 78. 静岡文化芸術大学 産学官連携. <https://www.suac.ac.jp/researchcenter/interact/cooperation/>
 79. SUAC「サウンドデザイン(長嶋)」リモート教材. 2020. <https://nagasm.org/1106/sound/index2020.html>
 80. OSC. <https://www.cnmat.berkeley.edu/OpenSoundControl>
 81. <https://nagasm.org/1106/news4/docs/jweb.jpg>
 82. <https://nagasm.org/1106/news5/20170419/>
 83. <https://nagasm.org/1106/news4/tosaya/index.html>
 84. Max6日記. <https://nagasm.org/ASL/max03/index.html>
 85. Max7日記. https://nagasm.org/ASL/Max7_1/index.html
 86. 続・Max7日記. https://nagasm.org/ASL/Max7_part2_1/index.html

87. Sketching日記(実質Max8). <https://nagasm.org/ASL/Sketch01/index.html>
88. SuperCollider. <https://en.wikipedia.org/wiki/SuperCollider>
89. SuperCollider (GitHub). <https://supercollider.github.io/>
90. <https://nagasm.org/ASL/YouTube.html>
91. <https://www.youtube.com/watch?v=vtbUT2R2qmE> , <https://www.youtube.com/watch?v=SwXvlhsB8qw>
92. <https://upload.wikimedia.org/wikipedia/commons/f/f0/IxiQuarks.jpg> , https://upload.wikimedia.org/wikipedia/commons/5/51/SuperCollider_screenshot2.jpg
93. sc-140. <https://supercollider.github.io/sc-140.html>
94. SuperCollider日記. <https://nagasm.org/ASL/SuperCollider/index.html> (part1-7)
95. Processing. <https://processing.org/>
96. https://nagasm.org/ASL/Max7_part2_5/index.html#processing
97. Processing日記. <https://nagasm.org/ASL/Processing/index.html> (part1-5)
98. Sonic Pi. <https://sonic-pi.net/>
99. <https://nagasm.org/ASL/Sketch12/fig1/027.jpg>
100. Mozzi. <https://sensorium.github.io/Mozzi/>
101. Arduinoによるハモンドオルガンの実現. 情報処理学会研究報告(2021-MUS-133), 情報処理学会, 2022. <https://nagasm.org/ASL/paper/SIGMUS202201.pdf>
102. Mozziギャラリー. <https://sensorium.github.io/Mozzi/gallery/>
103. <https://github.com/sensorium/Mozzi>
104. <https://sensorium.github.io/Mozzi/learn/>
105. Mozzi FM デモ. <https://www.youtube.com/watch?v=5wUFoPzmQh8>