

創作ノート

音楽制作ツールとしてのモバイルアプリ《食べれ音2》の制作 CREATION OF THE MOBILE APPS TABELEON2 AS A MUSIC PRODUCTION TOOL

森脇 ほのか

Honoka MORIWAKI

九州大学芸術工学府

Graduate School of Design, Kyushu University

中村 滋延

Shigenobu NAKAMURA

九州大学大学院芸術工学研究院

Faculty of Design, Kyushu University

概要

「音楽」とは、漢字で「音」を「楽」しむと書く。しかしながら、音を聞いて楽しむことは聴力があれば誰にでもできるが、音や曲を作ったり、演奏したりするには技術や才能が必要となり、誰もが楽しむということは難しいのではないかと筆者は考える。そこで、誰でも気軽に音づくりや曲作り、演奏を楽しむことができるツールとして筆者が制作したのが、モバイルアプリ《食べれ音》である。本稿では、筆者が別途研究として制作した、拡張現実感をインターフェースに用いた音楽演奏ソフトウェア《AirRythem-エアリズム-》をさらに進化させるためのツールとして《食べれ音》を発展させて作成した《食べれ音2》の紹介を行い、システム等の解説を行う。

The word “music” means “enjoy sound” in kanji. Anyone who can hear can enjoy listening to sound. However, to produce a musical tone or sound requires skill, and not everyone can enjoy it. Therefore, I created the mobile apps “Tabeleon” as tools by which anyone can enjoy making tones and sounds, playing music, and so on. In this paper, I will describe “Tabeleon2”, which I developed for the music performance software “AirRythem”, which is an augmented reality interface described in another paper. In addition, I will describe the system.

1. はじめに

DAW ソフトなどの普及により、実際に楽器をもっていない人でも作曲や演奏が行えるようになった。しかし、それらの作曲ソフトや演奏ソフトを用いたとしても、実際に作曲や演奏をするには、知識や技術・才能が必要となり、音楽の経験がない人が作曲や演奏をするためには、学習・習得のために相当な時間が必要とな

る。そこで筆者は、音楽の経験がない人でも、気軽に音楽を作って楽しむことができるソフトウェアを制作しようと考えた。先行作品の事例としては、しくみデザイン《KAGURA》[1] や Propellerhead Software の《Figure》[2]、成瀬つばさの《リズムシシリーズ》[3]、的場寛の《Overbug》および藤岡定の《cubic》[4] などが挙げられる。

本稿は、筆者が制作した《AirRythem-エアリズム-》と《食べれ音》についてそれぞれ解説するとともに、その2作品を組み合わせることで制作中の《AirRythem-エアリズム- Ver.3 (仮)》のために改良を行った《食べれ音2》の作品解説論文である。

2. 過去の制作①《AIRRYTHEM-エアリズム-》

2.1. 作品概要

《AirRythem-エアリズム-》は、2013年12月から制作を行ってきた音楽演奏ソフトウェアであり、インターフェースにARの技術を取り入れている。ARとは、現実世界にアノテーションを重畳表示して、現実世界の意味を拡張する技術のことであり、ここでいう「アノテーション」とは、文字や画像だけでなく、音声・触覚・嗅覚などの情報も含んでいる[5]。実際に、ARとして嗅覚を扱った《Meta Cookie》[6]などの作品もあるが、一般に広く知られているARコンテンツは画像の表示を主としているものが多く、ARと聞くと、現実世界に3DCGモデルが重畳表示されている様子を想像する人が多い。そこで、筆者はARによって擬似的に音の可視化と音の配置を行うことで、音に重きを置いたARコンテンツを制作しようと考え、本作品の制作を行った。

《AirRythem-エアリズム-》では、専用のマーカーをカメラに認識させることで、あらかじめ設定した音楽ファイルのループ再生をしたり、マーカーの配置され

た位置に応じた音を鳴らすことができる。また、マーカーを回転させることで対応した音楽ファイルにかかるフィルタの中心周波数を変化させたり、音量を変化させることで音楽を演奏することが可能になっている(図1)。



図1. 《AirRythem-エアリズム-》

2.2. システム概要

《AirRythem-エアリズム-》は、画面の処理を Processing[7]、音響の生成を Max 6[8]で行っている。まず、Processing側でPCに内蔵されたカメラからの画像を取得し、どのマーカーがどの位置に配置されているかの解析を行う。そこから、音響の生成に必要な情報を、OSCを介してProcessingからMax 6に送信し、Max 6で音の生成を行う。最後に、鳴らしている音の音量などの情報を今度はMax 6からProcessingに送信し、それらの情報をもとにProcessingで各マーカー上に配置する3Dモデルの生成を行い、カメラから取得した画像に重ねて画面に表示している。

《AirRythem-エアリズム-》では、演奏するために16個のマーカーを用いており、マーカーは役割に応じて5色に色分けされている。カメラから取得した画像は一度グレースケールに変換してから解析に使用しているため、この色分けはシステムの処理に影響はなく、演奏者がマーカーの役割を認識して演奏を行う為に行った。各色のマーカーの数と役割を図2に示す。

	黒	緑	赤	オレンジ	青
個数	1	6	1	2	6
認識時	電源を入れる	音楽ファイルを再生する			四角の座標に対応した音が鳴る
回転時	音量を変えられる	フィルタの中心周波数を変えられる			-

図2. 各マーカーの役割

黒色のマーカーは、音量を操作する為のマーカーで、このマーカーのみでは音が鳴らない。システム上音楽が演奏されている状態にある時(黒色のマーカーが認識されていて、かつ他のマーカーが1つでも認識され

ている時)は「PLAY」と書かれた画像が表示され、演奏されていない状態にある時(認識されているマーカーが黒色のマーカーのみの時)は「NO SOUND」と書かれた画像が表示され、マーカー上空で時計周りに回転する。このマーカーを右回転させると全体の音量が大きくなり、表示されている画像のサイズが大きくなる。逆に、左回転させると全体の音量が小さくなり、表示されている画像のサイズも小さくなる。また、認識されているマーカーの数が増加すると、「PLAY」画像を修飾するための円の数が増加してゆく。このように、演奏者は黒色のマーカー上に配置されている画像を見ることで、現在の演奏の状態を知る事ができる。

緑色のマーカー・赤色のマーカー・オレンジ色のマーカーは、認識されている間、あらかじめ定められた音楽ファイルをループ再生する。緑色のマーカーはリズムカルで忙しい音楽、赤色のマーカーはドラムのキック音、オレンジ色のマーカーはメインのメロディに対応している。黒色のマーカーが認識されていない時、音量は0になり、音楽は聴こえなくなる。緑色・赤色・オレンジ色のマーカーは回転させることによってバンドパスフィルタの中心周波数を変化させることが可能になっている。マーカーを右回転させると中心周波数が高くなり、左回転させると低くなる。また、認識されているマーカーに該当する音楽の音量をリアルタイムで測定しており、その音量に合わせてマーカー上に表示している直方体の数を増減させ、レベルメーターの様な動きをするようにした。

青色のマーカーは、マーカーの4角の座標のX値が時間軸に、Y値が音の高さに対応しており、その値によって鳴る音とタイミングが変化する。マーカーの4角が対応しているいずれかの音が鳴るタイミングで、マーカーの中心から立方体群が放出し、放出される立方体の色は4角それぞれの座標によって変化するようにした。

それぞれのマーカーについて、音が現実付加されていると感じやすくするために、マーカーの位置によって音のパンを振り分け、右側に配置したマーカーの音は右から、左側に配置したマーカーの音は左から聞こえるようにした。

2.3. 《AirRythem-エアリズム-》の考察

《AirRythem-エアリズム-》の制作を通して、音をメインに取り扱ったARコンテンツを生み出した。マーカーを配置したり、回転させるといった簡単な操作で音楽を演奏することができ、音楽の経験がない人でも音楽の演奏が可能になっている。しかし、《AirRythem》には以下の3つの課題がみられた。

1. 音楽をループ再生することがメインになっているため、演奏可能な音楽が限られ、飽きやすい

2. どのマーカーがどの音源に対応しているのかわかりにくい
3. 演奏中にマーカーを操作していると、操作中の手がマーカーを隠してしまい、音が止まってしまうことがあるため、操作がしにくい

3. 過去の制作② 《食べれ音》

3.1. 作品概要

《食べれ音》は、2014年8月に筆者が制作したモバイルアプリケーションである。タップで音虫を配置し、配置された音虫をカメレオンが食べることによって音楽を演奏することができ、画面の処理を openFrameworks[9]、音響の生成を Pure Data[10]で行っている（図3）。

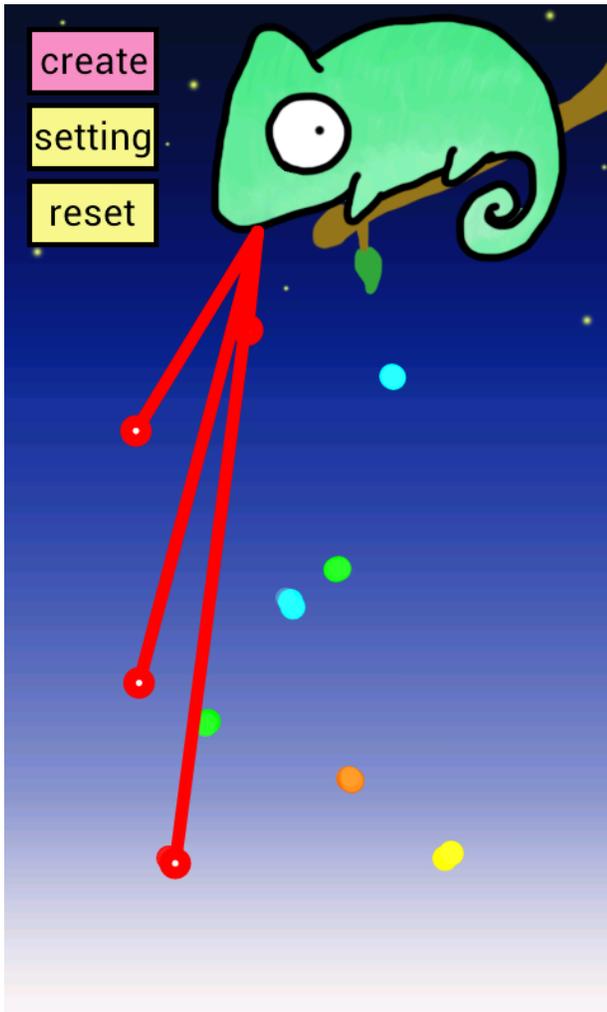


図3. 《食べれ音》

3.2. システム概要

《食べれ音》は、openFrameworks で画面の描画を行い、アプリに組み込んだ Pure Data のパッチから音を鳴らしており、スマートフォン単体で動かすことが可能である。

演奏者は、スマートフォンの画面をタップすることで音虫を配置する。カメレオンは、画面内にある音を左から順番に食べるようになっており、食べる際に音が鳴るようになっている。2秒で画面内の音虫を一通り食べ終え、それをひたすら繰り返すことによって音楽を演奏する。色虫の座標の縦方向は、音高に対応している。一度配置された音虫は、再度タップすることで移動が可能になっている。

音虫は最大で6種類配置することができ、画面上部の「Setting」と書かれたボタンをタップすると、それぞれの音虫の色を変えることができるようになっている。音虫のRGBの値のうち、Rの値とGの値の差をFM変調のモジュレータ周波数、Gの値とBの値の差をFM変調のモジュレータ振幅に対応させることで、色を変えると音色が変わるようにした。

3.3. 《食べれ音》の考察

《食べれ音》では、ユーザーができることは音虫の配置と音虫の色の変更だけであり、できることが少なすぎて、「演奏している」という実感は湧きにくい作品となっていた。しかし、拍という概念を特に設けていないにもかかわらず、ひたすら同じ音列を繰り返すことによって、音楽に聞こえるようになっており、演奏者は何も考えずに適当にタップするだけで音楽を作曲することが可能になっている。

《食べれ音》は音楽の経験がなくても適当に操作することで音楽を作曲することができるという点が評価され、International Students Creative Award 2015[11]ではモバイルアプリ部門において、最優秀賞を受賞した。客観的に見ても、その操作の簡易性が評価された結果であるといえる。

また、幾度かイベント等で展示を行ったところ、使用できる6色の色を生かして、音虫で絵を描き、その絵から生成される音楽を楽しむ演奏者の姿も見受けられた。本来であれば、音を作ることを目的に操作を行うアプリケーションであるが、逆に絵を描くことを目的として使用することで、偶発的に誕生した音楽を楽しむこともでき、《食べれ音》の楽しみ方が広がったように思われる。

4. 《食べれ音2》

これまで、音楽の経験がない人でも簡単に作曲や演奏できるソフトウェアを制作することを目指してきた。

《AirRythem-エアリズム-》では、誰でも演奏することが可能であったが、演奏者が任意で作曲を行うことはできなかった。また、《食べれ音》では、演奏者が任意で作曲することが可能であったが、できる操作が少なく、「演奏している」という実感が得にくいものになっていた。そこで、《AirRythem-エアリズム-》と《食べれ音》を組み合わせることで、双方の欠点を補うことができるのではないかと考え、《AirRythem-エアリズム-Ver.3 (仮)》の制作を行っている。具体的には、《食べれ音》で《AirRythem-エアリズム-》の各マーカーに対応する音楽を作曲し、そのマーカーを操作することで音楽の演奏を行っていくというものになる。そこで、《食べれ音》を《AirRythem-エアリズム-》のための音楽制作ツールとして発展させた《食べれ音2》を制作することにした。

4.1. 制作の目的

《食べれ音2》は《AirRythem-エアリズム-》でみられた、演奏できる音楽がワンパターンで飽きやすいという問題を解決することを目的に制作を行った。《食べれ音2》は《AirRythem-エアリズム- Ver.3 (仮)》のためのツールであり、モバイルアプリ単体では動かすことができないというのが大きな特徴である。演奏者は、《AirRythem-エアリズム- Ver.3 (仮)》を用い、従来の《AirRythem-エアリズム-》のような音楽も演奏することができる。付け加えて、手元のスマートフォンで《食べれ音2》を操作することによって、《AirRythem-エアリズム- Ver.3 (仮)》で鳴る音を《食べれ音2》で作曲した曲に切り替えることが可能になるのである。

そこで、《AirRythem-エアリズム- Ver.3 (仮)》の為のツールとしての《食べれ音2》を制作するにあたり、以下の指針をたてた。

4.1.1. 音のクオンタイズを行う

《AirRythem-エアリズム- Ver.3 (仮)》では、既存の音楽ファイルと、《食べれ音2》で制作した音楽を同時に鳴らすことがあると考えられるので、作曲できる音のクオンタイズを行い、既存の音楽ファイルと同時に鳴らしても違和感がないものにしなければならないと思われる。

4.1.2. 音長が任意で変えられるようにする

《食べれ音》では、なる音の長さを変更することができなかったが、より幅広く作曲が行えるようにするために、音長が演奏者により自由に変更できるほうが好ましいと思われる。

4.1.3. 作曲できる音楽にまとまりをもたせる

従来の《食べれ音》ではD2~E6の間の音であればどんな音高の音も出すことができていた。しかし、この方法を用いると不協和音が生じやすくなってしまい、《AirRythem-エアリズム-》の既存の音楽ファイルと同時に鳴らすと違和感を生じかねない。そこで、鳴らすことができる音の高さがペンタトニックに固定することによって、音楽のまとまりを作り、それによって楽曲全体のクオリティの向上を目指す。

4.1.4. 制作できる音色の種類を増やす

従来の《AirRythem-エアリズム-》では、RとGの値の差およびGとBの値の差が音色の合成に使用されていたので、色が違うのに音色は一緒という事態が生じるようになっていた。この問題を解決し、制作できる音色の種類を増やすことで、より幅広い音楽を作曲・演奏することができると思われる。

4.2. システムの解説

《食べれ音2》は、《食べれ音》と同様に、openFrameworksによって制作を行った。《食べれ音2》では、音はスマートフォンからは鳴らさず、パソコンで起動中の《AirRythem-エアリズム- Ver.3 (仮)》のシステムから音を鳴らすようになるので、今回の制作では、音の生成に必要なパラメータをOSCを介して、パソコンで起動中のMaxパッチに転送し、Maxパッチで音を鳴らすようにした。

前述した、《食べれ音2》の制作指針に沿って、以下のようにシステムを改良した。

4.2.1. 音のクオンタイズ

画面を横方向に16分割し、同じ区間にある音虫を同時に食べさせることによってクオンタイズを実現させた。システム上、2秒間で左端から右端までの音虫を食べるようにしているので、BPM120とした時の16分音符にあたる間隔でクオンタイズされていることになる。

4.2.2. 音長

画面をタップして音虫を配置する際、画面に指が触れている時間の長さで音長が変えられるようにした。音の長さは、①のクオンタイズと大きく関わっているため、 $2(\text{秒}) \div 16(\text{分割}) = 0.125(\text{秒})$ を基準とし、そこから2倍、3倍、4倍の音を鳴らせるようにした。また、その長さに応じて、音虫の長さも変化させることで、音の長さが視覚的にも伝わるようにした。

4.2.3. 音楽のまとまり

音楽にまとまりを感じさせるために、鳴らすことができる音を限定することにした。演奏者は、「ヨナ抜きペンタトニックの5音」・「コードのトライアド3音」・「コードの主音と属音の2音」と、限定される音の数を切り替えることができるようにした。また、ボタン操作で属調や下属調に転調することができるようにした。

音のベロシティについても、強弱をつけることで音楽にまとまりを持たせることができるようにした。音虫を配置する際に、ランダムで4種類のベロシティが振り分けられるようにした。

4.2.4. 音色の種類を増やす

より音色の種類を増やすために、R、G、Bで扱える音のパラメータを変更した。まずRの値では、FM変調のモジュレータ振幅を変更することができる。FM変調のモジュレータ周波数は、《食べれ音2》では一定で600Hzとし、モジュレータ振幅のみを変更することでFM変調を行うことにした。次に、Gの値はキャリアの波形をsin波～ノコギリ波～矩形波となめらかに変えられるようにした。これは、sin波・ノコギリ波・矩形波を同時に鳴らし、3音の合成の比率を変化させていくことで可能にした。最後に、Bの値はデチューンとして、キャリア周波数から少しずらした周波数の音を鳴らすことができるようにした。

また、シンセサイザーで制作した音だけでなく、MIDIからパーカッションの音を鳴らせるモードに切り替えることができる仕様を取り入れた。

4.3. 《食べれ音2》の考察

本章では、前述した《食べれ音2》の制作指針にしたがって、それぞれの変更箇所の達成点と課題を述べるとともに、《AirRythem-エアリズム- Ver.3 (仮)》のためのツールとしての《食べれ音2》の有用性について考察する。

4.3.1. 音のクオンタイズについて

今回の《食べれ音2》では、画面を横方向に16分割することによって音のクオンタイズを実現させた。聞こえる音としては、クオンタイズされているため、違和感がないが、見た目的に、横方向に少しのずれがあったとしても同時に音が鳴らされるため、視覚情報として若干の違和感を感じさせるものになってしまった。ここで、見た目的にも16分割の位置に固定することで、その違和感を減らすことができるが、そうすると《食べれ音》にあった、絵を描くことで音楽が生まれるという面白さが損なわれてしまうという問題が生じる。

音のクオンタイズについては、見た目上違和感のないものにする改良が必要になってくると感じられる。

4.3.2. 音長について

音長を変更することができるようになったことによって、《食べれ音2》で演奏することが可能な音楽に多様性をもたせることができた。しかし、例えば画面の右端に長い音をもってこようとすると、見た目的に音虫が途中で途切れてしまうという問題がある。これを解決するために、画面の右端にいったら左端から音虫の続きが描かれる等の改良が必要となる。

4.3.3. 音楽のまとまり

音高を限定することによって、制作できる音楽にまとまりを持たせることに成功した。これについては、演奏可能な音高群の種類を増やしていくことで、更に演奏できる音の多様性を増すことができるので、最終的にはより多くの音高群を用意しようと考えている。

4.3.4. 音色の種類について

音色の生成方法を変えることによって、《食べれ音》よりも幅広く音色を生成できるようになった。しかし、パーカッションの音を加えたことにより、《AirRythem-エアリズム- Ver.3 (仮)》にあった既存の音楽に存在するビート感といかにして共存するかが今後の課題であるといえる。

以上のことから、《食べれ音2》で生成できる音楽の多様性が増し、《AirRythem-エアリズム- Ver.3 (仮)》のためのツールとして有用であると考えられる。しかし、視覚上の問題はまだまだあり、改善の余地があるといえる。

5. おわりに

《食べれ音2》は、《AirRythem-エアリズム- Ver.3 (仮)》のためのツールとして、《AirRythem-エアリズム-》の問題点であった、「演奏される音楽に限られ、飽きやすい」という問題を解決することができるため、非常に有用である。また、《食べれ音2》単体で演奏できる音の多様性も増し、音楽演奏ツールとしても使用できると考えられる。

今後は、前述した《食べれ音2》の問題点を改善していくとともに、《AirRythem-エアリズム- Ver.3 (仮)》のツールとしてだけでなく、単体での音楽演奏アプリとしても操作できるようにし、AppStoreなどで配信を行い、より多くの人に気軽に音楽を作って楽しんでもらいたいと考えている。

6. 参考文献

- [1] 株式会社しくみデザイン, KAGURA, <http://kagura.cc/>
- [2] Propellerhead Software, Figure, <http://www.propellerheads.se/figure>
- [3] 成瀬つばさ, リズムシシシリーズ, <http://otonoasobiba.hiroimon.com/>
- [4] 中村滋延, 演奏ツールとしてのソフトウェアアート—その創造特性と可能性, 芸術工学研究九州大学大学院芸術工学研究院紀要, Ver13, pp.9-30.
- [5] 佐野彰, AR 入門-短になった拡張現実-, 株式会社工学社, 2010, p26
- [6] Takuji Narumi, Meta Cookie, <http://www.cyber.t.u-tokyo.ac.jp/~narumi/metacookie.html>
- [7] Processing, <https://processing.org>
- [8] Max, <https://cycling74.com/products/max/>
- [9] openFrameworks, <http://openframeworks.jp/>
- [10] PureData, <https://puredata.info/>
- [11] International Students Creative Award, <http://kc-i.jp/award/isca/>
- [12] 橋本直, AR プログラミング-Processing でつくる拡張現実感のレシピ-, オーム社, 2012
- [13] 青木直史, サウンドプログラミング入門-音響合成の基本と C 言語による実装-, 技術評論社, 2013

7. 著者プロフィール

森脇 ほのか (Honoka MORIWAKI)

1991 年広島県生まれ。九州大学芸術工学部音響設計学科卒業。九州大学芸術工学府の修士課程に在学中。「『音楽』とは『音』を『楽しむ』と書く」という言葉をテーマに、音楽の知識がない人でも気軽に音楽を楽しむことができるような音楽演奏システムを制作・研究している。

中村 滋延 (Sigenobu NAKAMURA)

1950 年生まれ。作曲家、メディアアーティスト、音楽評論家、映画研究者。交響曲 5 曲を含む 100 曲以上のクラシック系現代音楽を作曲。また「音楽系メディアアート」という領域を創成し、視覚要素を含むコンピュータ音楽やサウンド重視の映像アートを多数制作。現在、九州大学大学院教授（芸術工学府コミュニケーションデザイン科学コース、芸術工学部音響設計学科）