

リズムゲームをモチーフとした 作曲支援アプリケーションの開発

秋元 和久^{1,a)} 矢澤 櫻子^{2,b)} 星野 准^{1,c)}

概要: 多くの音楽的知識を必要とする作曲行為は音楽的知識の乏しいユーザーにとって魅力的ではあっても敷居が高いといえる。オブジェクトと音を対応させ、オブジェクトの配置から様々な音列を生成し音楽的知識の乏しいユーザーでも直感的に作曲を体験するシステムは数多く考案されてきた。本研究では「直感的にわかりやすく」「アプリで手軽に作曲を楽しめる」システムを目指す。画面のタッチ操作で音を生成するゲームという作曲支援の形を取り、スマートフォンなどのアプリで実装することでより身近で手軽に楽しめるようにした。今回は、アプリの開発及びプロトタイプの評価について報告する。

Development of A Support System for Music Composition using Rhythm game as a motif

KAZUHISA AKIMOTO^{1,a)} SAKURAKO YAZAWA^{2,b)} JUNICHI HOSHINO^{2,c)}

1. はじめに

作曲とは様々な長さ・高さ・強さ・音色の音を組み合わせる行為である。作曲において音の組み合わせ方には一定のルールがあり、複数の音から成る音には不協和音という不快な音が存在し、作曲のための音楽知識を用いて不協和音にならないようにする必要がある。作曲のための専門知識を習得することは難しく、作曲を始めることを断念する他ないユーザーは多い。作曲に関する音楽的知識の乏しいユーザーに対し、専門知識を必要とする部分を計算機にアウトソーシングすることで作曲やメロディを生成する研究が盛んに行われてきた [1]。一方で、作曲行為をエンタテインメント化することで、作曲自体の魅力を高め、幅広いユーザーを作曲に惹き付けるようなシステムも開発されている [2]。

本研究の目的は、作曲の経験が無いユーザーに作曲に興味を持ってもらい、作曲を楽しんでもらうため、音楽創作

活動において必須である作曲の専門知識と、作曲に興味を持ち作曲を楽しむためのエンタテインメント性を高いレベルで両立させることである。これを実現するために、「直感的にわかりやすく」「手軽に作曲を楽しめる」ようにするためにリズムゲーム (予め決まったリズムに沿ってユーザーがアクションを起こすゲーム) をモチーフとした作曲支援システムを提案する。このシステムで、作曲に関する音楽的知識が必要となるせいで手を出しにくい部分をゲームとしてエンタテインメント化することにより、作曲知識が無いユーザーでも、ゲームを楽しむだけで作曲が出来るようになる。

本論文では、2章で提案システムに関連する研究を紹介し、3章で提案システムの概要を説明する。また、4章で提案システムを実装した様子を記し、5章で実装に対する考察と課題を述べる。最後に、6章でまとめと今後の展望を述べる。

2. 関連研究

ユーザーに作曲に関する専門知識を要求せずに、あるオブジェクトと音を対応させることで音列を生成する作曲支援の研究は様々なものが行われてきている。現実世界に

¹ 関西学院大学理工学部情報科学科
² 筑波大学大学院知能機能システム専攻
a) aga29006@kwansai.ac.jp
b) sakurako@music.iit.tsukuba.ac.jp
c) jhoshino@esys.tsukuba.ac.jp

においてオブジェクトと音を対応させ、オブジェクトの配置によって音楽を生成するシステムがある [4][5][6]. The MusicTable[4] は特に子供を対象とした作曲支援であり、キューブを置く場所によって楽曲を製作することができる。具体的には、AR でキューブを置く座標を取得し、x 方向を音が鳴るタイミング、y 方向を音高に対応させて音を生成する。キューブを操作することで、楽器や音長の変更もできる。reactTable[5] は、The Music Table のような基本操作のみならず、音を鳴らすキューブにエフェクトを掛けるキューブを接続することで、音に変化をつけることができる。reactTable は実際にライブパフォーマンスでも用いられている。cubie[6] は、ユーザーがアルファベット 8 文字を指定された場所に置くと、その 8 文字に対応した音が順番に鳴る。その 8 文字はログとして残り、その後 8 文字のアルファベットが 8 回入力されるまで音列を生成し続ける。アルファベットには、音だけでなくテンポの操作やエフェクトを意味するものがある。適当にアルファベットを並べて生成される楽曲を聞いて楽しむこともできれば、cubie のシステムを利用して複雑な楽曲を制作することもできる。reactTable と cubie は、音楽的知識の乏しいユーザー、そうでないユーザーでも音列を生成し、作曲の体験をできるシステムを目標としている。

これらは現実世界のオブジェクトの操作が直感的でわかりやすい支援である。

これらに対し、画面やボタンをタップするといった動作で音列を生成するシステムの研究も行われてきている [2][3]. TENORI-ON[2] は、1 列目から 16 列目まで順番に、押された状態のボタンに対応する音を生成し、またシステム側で音楽として成立する音列に修正をする。ボタンの状態は 16 列目から 1 列目に戻ったときにリセットされないため、ボタンに変更を加えなければ同じフレーズが繰り返される。一方で、ボタンの状態をランダムに変更する機能を用いることで、同じフレーズを繰り返さないようにすることもできる。上側のボタンは高い音に、下側のボタンは低い音に対応している。LaunchPad[3] はユーザーが音とボタンを対応付けすることを想定して作られたシステムで、ユーザーがボタンに対応させる音を決めることができる。また、TENORI-ON のように 1 列目から順番に再生されるのではなく、ボタンを押したときに音を生成する。

ボタンと音を対応させず、スマートフォンなどの画面をユーザーがタップもしくはなぞり、その位置から音を生成することで音楽を作るシステムがある [7][8]. bloom[8] は、画面を適当にタッチすることで作曲や映像が楽しめるアプリである。画面をタッチすると音がなり、それと同時に鮮やかな波紋が画面に現れる。上側をタッチするほど高い音、下側をタッチするほど低い音を生成する。ユーザーのアクションは自動でループされ、ユーザーがアクションを起こさなくても、bloom が自動で作曲をしてくれる。

musicLine[7] は、画面をなぞると、コンピュータがそれに従って音列を生成する。TENORI-ON や bloom と同様に、なぞった場所が上であるほど高い音、下であるほど低い音を生成する。

それぞれの代表として、reactTable, TENORI-ON, musicLine のインターフェースを図 1, 2, 3 に示す。

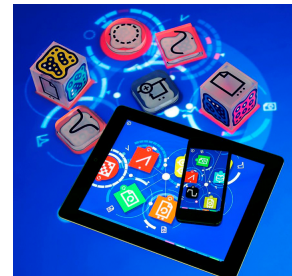


図 1 reactTable



図 2 TENORI-ON

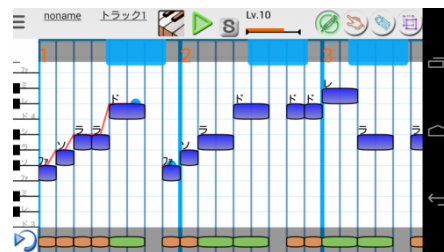


図 3 musicLine

我々は、スマートフォンなどのアプリとして手軽に作曲に触れられるようなシステムにしたいと考えている。このため本研究では「直感的にわかりやすく」「アプリで手軽に作曲を楽しめる」システムを目指す。

3. 提案システム

我々はユーザーがリズムゲームをプレイすることを通じて作曲の経験を得られるシステムを提案する。スマートフォンやタブレットのアプリとして開発することで、手軽に作曲に触れられることが狙いである。具体的には、ユーザーは画面にアプリが指示するタイミングでタッチを行い、タッチをすることで音を生成する。音の高さは画面の位置に対応しており、アプリの指示するタイミングで繰り返す。

返し画面の様々な部分をタッチすることでメロディを生成する。提案システムのゲーム画面を図4に示す。図4の赤い線と青い線を用いてユーザーにタッチするタイミングを指示する。詳細は4章で後述する。

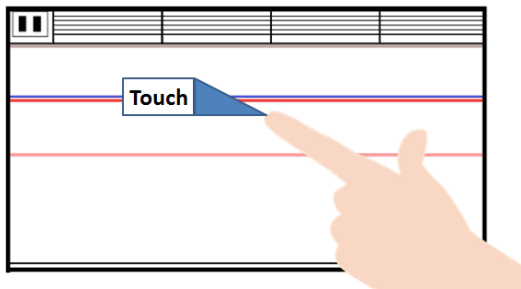


図4 ゲーム画面

提案システムの流れは次の通りである。ゲームの流れをフローチャートとして図5に示す。

1. ゲームを開始し、伴奏を流す。
2. アプリがタッチするタイミングを画面に表示する。
3. ユーザーがそのタイミングで画面をタッチすれば音を生成する。

4. 2と3を繰り返し、伴奏が終了したらゲームを終了する。タイミング通りにうまく画面をタッチできていればゲームクリアになる。

5. ゲームをクリアすると、ゲーム中に生成した音列と伴奏を同時再生することができる。

提案システムは音の配置にタイミングよく画面をタッチするという行為を付加しており、音の配置という動作でもユーザーに楽しさを与えることができる。また、提案システムはゲームクリアの概念を含んでいるので、ゲームをクリアするという目標を達成することで簡単に楽曲が完成する。また、提案システムはリズムが固定で、タイミングよく画面をタッチするだけで作曲ができるので、操作が非常に簡単である。正しい音楽理論を適応する上では、リズムが固定なので音楽理論上不自然なリズムになることがない。また、タッチする場所で音高の高低を選ぶことを繰り返すだけで自動的にメロディが完成するので、理論的に正しいメロディを作ることが出来ないユーザーでもメロディを作ることができる。

作曲支援システムは、ユーザーとコンピュータが共同で作曲を行う、つまり半自動作曲システムであると言える。松原らは、自動作曲システムの要件として次の3つを挙げている [9]。

要件 (A) 新規な音楽を創出できること

要件 (B) 聴衆に馴染みのある音楽スタイルを踏襲していること

要件 (C) ユーザーの制作意図を反映できること

提案システムは、リズムゲームの性質上 (B) を重視し、

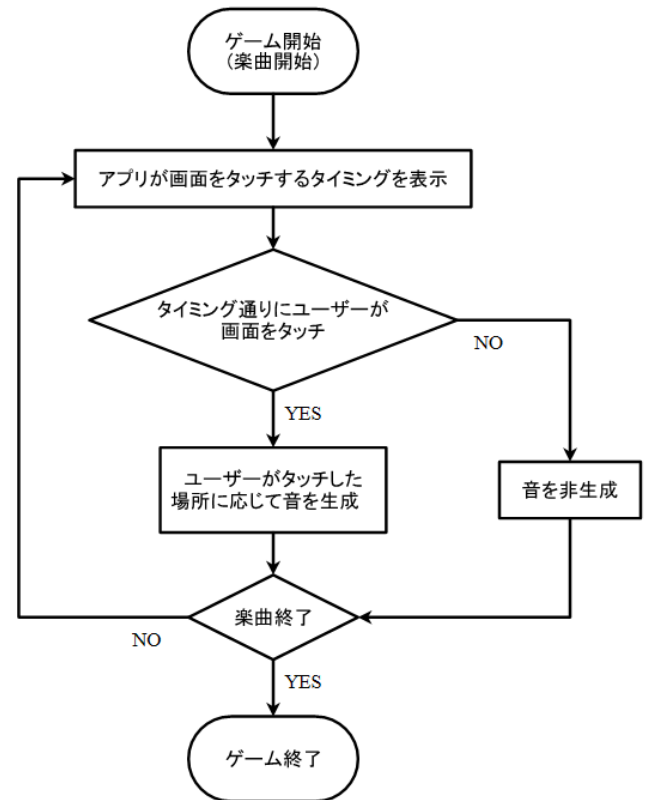


図5 提案システムのフローチャート

同時に (C) の要件を達成することを目標としている。提案システムは、音を上行させたい、あるいは下行させたいというユーザーの意図を反映させることで、(C) の要件を達成しようとしている。

4. 実装

我々はシステム構築の第一歩として、一つの伴奏に対して指定した5種類の音でメロディを生成するシステムを実装した。伴奏は基本的なコード進行を採用した。伴奏がCメジャーであるので、用いる音はCメジャーペンタトニックスケール (C, D, E, G, A) とした。

4.1 実装1回目

アプリケーションとして実装した画面を図6に示す。また、実際にスマートフォン上で動いている様子を図7に示す。アプリはゲーム開始と共に、事前に用意された伴奏を再生する。伴奏が終わると、初めからやり直すか、ユーザーのアクションによって制作した楽曲を再生するかを選ぶことができる。ただし、作曲未経験者は楽曲の途中から再生しないと仮定し、楽曲の途中からの再生には対応しなかった。

ユーザーが画面をタッチするタイミングを示すオブジェクトとして、リズム線と時間軸線を用いた。リズム線はタッチするタイミングの間隔を示し、このアプリでは赤い横線で表示される。リズム線は一度表示されると動くこと

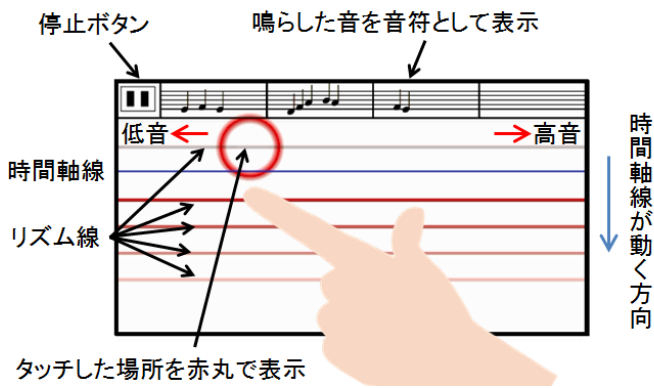


図 6 実装 1 回目

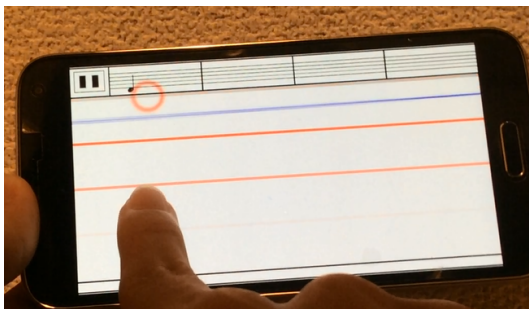


図 7 スマートフォン上の様子

はない。一方時間軸線は時間が進むにつれて上から下へと移動し、移動する速度によって曲のテンポを示す。アプリ上では青い横線で表示される。時間軸線は、一番下に移動すると一番上に戻ってまた上から下への移動を続ける。

リズム線は、直近のリズム線のみを表示するため、時間軸線と重なりそうなリズム線は赤く、そうでないリズム線は白く表示される。時間軸線と重なったリズム線は徐々に黒くなり消える。時間軸線の色は常に青とした。

時間軸線が移動し、リズム線と重なった時をタッチするタイミングとした。リズム線と時間軸線が重なるときに画面をタッチすると音が鳴り、用意済みの伴奏に対して不協和音にならない 5 種類の単音のどれかが選択される。どの音を鳴らすかはユーザーが画面をタッチした X 方向の座標で判定され、画面をタッチした場所が右側であるほど高い音、左側であるほど低い音になる。ユーザーは次の音高を上行させるか下行させるかをタッチする場所によって選択する。この操作を何回も繰り返すことでメロディを制作する。Y 方向の座標は現時点で考慮していないため、リズム線と時間軸線が重なるタイミングで画面のどこかをタッチする必要はあるが、重なる場所をタッチする必要はない。画面をタッチすると、リズム線のどの部分をタッチしたことになっているのか赤丸で表示される。この赤丸は小さい状態で表示され、すぐに大きくなって消える仕様となっている。

画面の上部に楽譜を設置することで、ユーザーが画面をタッチして音を鳴らすと対応する音符が楽譜に表示され、

ユーザーはどの音高の音が鳴ったのかわかるようにした。また、擬似的に楽譜を書かせることによってユーザーに作曲しているという実感を与えた。

4.2 実装 1 回目に対する意見

上記の実装に対して得られた意見は以下の通りである。

1. 楽譜に現れる音符は上下で音高を表しているのに対し、操作画面では左右で音高を表しており、音高を直感的に操作することが出来ない。
 2. 楽譜に現れる音符を、四分音符以外で表す必要があっても、四分音符で表している。
 3. 楽譜に音符が表示されるのは演奏時のみになっているが、再生時にも音符を表示すべきである
 4. 楽曲が事前にわからないので、ゲーム開始直後はどんなテンポで画面をタッチしたら良いのかわからない。
 5. リズム線の現れるタイミングがわかりずらく、いつ画面をタッチしたら良いかわからない
 6. ユーザーが操作できるのがメロディだけなので、ユーザーが作りたい音楽の種類が反映されない
 7. 楽譜部分とリズム線の現れる部分が一本の線のみで句切られているため見にくい
 8. リズムがランダムになるので次にどんなリズムが来るのかが想定出来ない (一般に、ある曲に対してタッチするタイミングはランダムでなく決まったものになっており、何度も曲を練習することでそのタイミングを覚える)
- 再実装では、ユーザーインターフェースを変更・追加し、意見 1~5 に対応した。

4.3 実装 2 回目

実装画面を図 8 に示す。

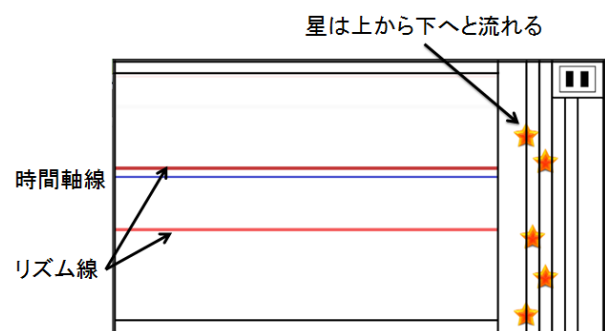


図 8 実装 2 回目

1 の楽譜と操作画面の音高の高低の向きが一致していないという意見に対して、楽譜を上部から右部に移し、時計回りに 90 度回転させた。この変更により音高の高低が左右で表されていることが直感的に理解できるようになった。2 の楽譜に現れる音符の種類がおかしいという意見に対して、音符でなく星を表示するようにしたことで、四分音符

や八分音符の表示処理を回避した。また、ゲームらしさが増した。3の再生時にも音符を表示するべきであるという意見に対して、再生時にも星が表示されるようにした。4の楽曲のテンポがわかりにくいという意見に対して、事前に楽曲がどんなテンポなのかわかるように、曲が開始されるまでの4小節のタイミングを数字で表示した。5のリズム線が現れるタイミングがわかりづらいという意見に対して、リズム線の表示速度を変更した。

4.4 実装2回目に対する意見

再実装に対して以下の意見を得た。

9. 星の色を左右で変更した方がより直感的になる。

10. 楽譜上の五線譜はなくてもよい

今後は上記の意見に対して実装を行う。

5. 考察と課題

5.1 考察

今回、20代男性2人にアプリを体験してもらって意見をもらうことが出来た。ユーザーから意見としては出てこなかったが、アプリの展望として必要と考えられる5点を挙げる。

1点目として関連研究と生成される楽曲に違いがあった。TENORI-ONやbloom, The Music Table, reacTable, cubieはユーザーのアクションによって配置された音列をある一定区間で何度も繰り返し、そこに新しくユーザーのアクションを加えて楽曲を変化させていき、楽曲全体を作り上げるシステムである。こうして作られた楽曲は、ユーザーの意図だけでなくシステムによる介入があるので、新規性がある楽曲を生み出しやすい。それに比べ本システムは、メインの音列のみを操作するため、関連システムの楽曲とは違った楽曲が生成される。提案システムが生成する楽曲に対し、ユーザーが新規性なしと判断するかどうか、もしくは、そもそも新規性がないことは作曲支援システムにとって問題なのかどうかを今後のアンケートで評価する必要がある。

2点目として、提案システムは演奏によって作曲を行う即興性を持つ。その反面、ユーザーは一度入力に手間取ってしまうと、次の入力が上手くいかない傾向があった。これはユーザーの作曲意欲を削いてしまう可能性があるため、ゲームを途中からやり直す機能を強化するなどの対策をする必要がある。

3点目として、フリックや長押しなど普通のリズムゲームにあるはずの操作がない、タッチした座標のY方向が意味を持っていない、というゲームの完成度の低さが挙げられる。これらに対して、機能の追加・改善を行う。

4点目は使える音が5種類だけでありメロディの表現力が乏しいということである。自動作曲システムである[1]や音列に和声を付与するシステムである[10]などを参考

に、より良い音列生成アルゴリズムを実装する。

上記で述べた課題に加え、実装1,2回目に対応できなかった意見が課題として残っている。6の作りたい音楽の種類が反映されないという意見については、伴奏や楽器の種類を増やすことで対応する。7の上部のリズム線がタッチしにくいという意見については、少数意見であったため、今後の検討材料とする。8の次に来るリズムが想定できないのでタッチしにくいという意見については、同じリズムを何度も繰り返すようなアルゴリズムを採用することで対応する。

6. まとめと展望

本研究では、リズムゲームをモチーフとした作曲支援アプリケーションを開発した。開発したプロトタイプを実際にプレイしてもらい、意見をもらうことができた。今後は作曲支援アプリとしてより精度を高めるべく、考察と課題で述べた課題と実装2回目でもらった意見に対する機能の追加・実装を行う。また、アプリケーションの評価として、システムの使用前後でどれだけ作曲への興味に変化したか、楽曲の完成度、ユーザーの意図実現度、作曲の楽しさなどをユーザーにアンケートして確認する。また、作業の効率性や動作速度、操作難易度といったシステムの使い心地についても評価する。上記した評価項目は、他システムとの比較実験を行う。本システムはユーザーに入力を強制するシステムとなっているので、その点がユーザーにどういった影響を与えるかを特に評価する。実験後、実験の結果を踏まえ、更なるアプリの改良を目指す。

参考文献

- [1] 深山 覚, 中妻 啓, 米林 裕一郎, 酒向 慎司, 西本 卓也, 小野 順貴, 嵯峨山 茂樹. Orpheus: 歌詞の韻律に基づいた自動作曲システム, 情報処理学会研究報告. [音楽情報科学], 179-184, 2008-07-30.
- [2] <http://jp.yamaha.com/products/musical-instruments/entertainment/tenori-on/>
- [3] <http://www.h-resolution.com/novation/launchpad.php>
- [4] 牧野 真緒, 大島 千佳, Rodney Berry, 樋川 直人, 西本 一志, 鈴木 雅実, 萩田 紀博: “作曲・演奏支援システム The Music Table”, インタラクシオン 2004, A-09.
- [5] <http://reactable.com/>
- [6] 藤岡定, 中村滋延, 栗原詩子: Cubie-パズル・ゲームをモチーフとした「思考型」演奏ソフトウェア, 情報処理学会研究報告 [音楽情報科学], pp.73-77, 2007.
- [7] <https://play.google.com/store/apps/details?id=jp.gr.java.conf.createapps.musicline>
- [8] <http://www.generative-music.com/>
- [9] 松原正樹, 深山覚, 奥村健太, 寺村佳子, 大村英史, 橋田光代, 北原鉄朗. 創作過程の分類に基づく自動音楽生成研究のサーベイ. コンピュータソフトウェア, Vol. 30, No. 1, pp. 101-118, 2013.
- [10] ベンジャミン ルカ エバンズ, 渚棟方, 哲雄小野. 自動和声付与システム' CMY '. 情報処理学会北海道シンポジウム, October 2014. session-A-03.