

# Au-thello: ゲーム「オセロ」をモデルにした 音楽演奏コントローラ

松村智弘<sup>†</sup> 中村滋延<sup>††</sup>

今までに、数多くのミュージックシーケンサーが発表されているが、専門性が高く誰にでも簡単かつ有効に操作できるものではない。そこで筆者は一般に広く知られたゲームである「オセロ」をモデルにシーケンソフト<Au-thello>を制作した。本稿では<Au-thello>について解説し、加えてその制作過程で得られた問題や成果から、専門性の高さゆえに敬遠されがちなシステムにゲーム性を付加することの利点や問題点に関して論述する。

## Au-thello: A Reversi-like Interface for Musical Performance.

TOMOHIRO MATSUMURA<sup>†</sup>  
SHIGENOBU NAKAMURA<sup>††</sup>

Nowadays, many music sequencers are commercially available. However, these are usually designed for professional users and may be difficult for amateurs to use. I have developed Au-thello, a software sequencer modeled on the well-known game Reversi. This paper introduces Au-thello and discusses the modeling of complex systems on games—a concept that is likely to be rejected because of the high degree of expertise required. In addition, I discuss the problems encountered and the results obtained during the development of Au-thello.

## 1. はじめに

本稿は筆者の作品<Au-thello>（オーセロ）の制作過程について解説・考察するものである。

<Au-thello>はオセロゲームを題材にしたインタラクティブアート作品である。鑑賞者がマウスを使ってオセロゲームをプレイすると音響が再生される。オセロの駒ひとつが再生される音ひとつと対応している。また駒はダイアルになっており、再生される音響のパラメータを変化させることができる。鑑賞者がゲームを進めていくほどに再生される音響も複雑になっていく。

ミュージックシーケンサー等の音楽演奏支援ツールは今までに数多く発表されている。しかし、ボタンやスライドバーの多さや操作の複雑さにより、操作に専門的な知識を要求するものが多い。例えば、ヤマハの<TENORI-ON>は16x16個のLEDボタンを操作することで“音楽の知識がなくても視覚的・直感的に作曲/演奏することが可能”なツールである。<TENORI-ON>は複数の演奏モードやレイヤーを実装することで豊かな音楽演奏を可能にしているが、そのシステムの複雑さは初心者にとってはわかりにくいものになってしまうという面もある。

<TENORI-ON>に限らず、こういったシステムのわかりにくさの原因は主に2つあると考えられる。1つ目は用意された入力スイッチが多いため、「何を入力すればいいかわからない」ことである。2つ目はシステムの内部構造が複雑で「入力に対する出力結果がわかりにくい」ということである。以降本稿ではこの2つを克服しようとするものを「わかりやすい」と呼ぶことにする。

筆者の作品制作コンセプトは「誰にでも気軽に触ってもらえる作品であること」である。そのために作品が満たすべき条件は数多く挙げられるが、そのうちのひとつ大切なものは「わかりやすい」ということだろう。<Au-thello>はオセロゲームのルールを操作に持ち込むことで、「わかりやすい」シーケンソフトの制作を目的として設計された。

本稿では<Au-thello>について解説すること、加えてその制作過程で得られた問題や成果から、<Au-thello>にオセロゲームを導入したことの利点や問題点を明らかにすることを目的とする。更にこれより、システムにゲーム性を付加することの利点や問題点に関して可能な限り一般的な結論を得ることを試みる。

<sup>†</sup> 九州大学芸術工学部音響設計学科  
School of Design, Kyushu University

<sup>††</sup> 九州大学大学院芸術工学研究院  
Faculty of Design, Kyushu University

## 2. システムの構成

### 2.1 使用機材

- ・ パーソナルコンピュータ (Mac OS X)
- ・ 入力用マウス (左右ボタンがあるもの)
- ・ 「Max 5」 Cycling '74
- ・ 「Processing」 Casey Reas, Benjamin Fry

### 2.2 プログラム構造

<Au-thello>はインターフェース部と音響再生部から成る。インターフェース部は Processing で書かれている。演奏者が操作するのは主にこのインターフェース部となる。対して、音響再生部は Max で書かれている。ここでは音響の生成と再生が MIDI によって制御されている。両者は osc メッセージの通信により接続されており、音響再生部における各パラメータの値はインターフェース部から受け取った値に応じて変化する。

## 3. 作品の制作過程

<Au-thello>は試作品を含め、3つの作品を経て現在に至る。ここでは順を追って3つの作品を解説し、制作過程で得た課題や成果を記す。

### 3.1 作品 1: 試作品

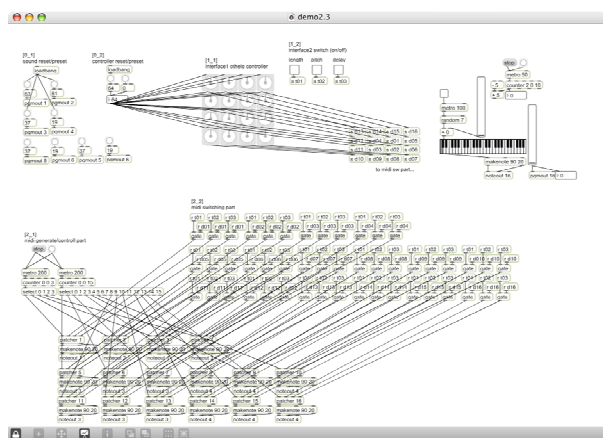


図 1 <Au-thello>の試作品 Max のパッチ画面  
Figure 1 <Au-thello> on a trial as a Max project.

Max のプロジェクトとして試作品を制作した。Max で 4x4 個の Dial オブジェクトを並べた。各オブジェクトにはそれぞれ 1 個の音を当てはめ、Dial の値によって音の大きさを変化させた。

実際にプログラムを操作して得られた感想は Dial の配列が視覚的に面白いことと、Max の MIDI 音声で十分な音響が再生できるということだ。一方で課題として、配列の配置を活かしていない事、Max 上での高度な操作には限界がある事が分かった。

### 3.2 作品 2: <Au-thello>

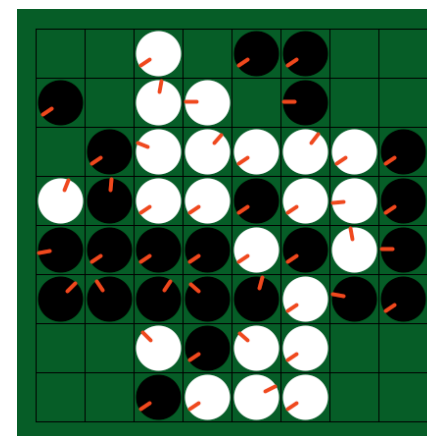


図 2 <Au-thello> スクリーンショット  
Figure 2 <Au-thello> Screen view of GUI.

Max 及び Processing を用いて制作し<Au-thello>と名付けた。「配列より連想される一般的に広く知られたゲーム」としてオセロゲームを題材にし、そのロジックを操作に組み込んだ。それに伴い Dial は 8x8 個に増設した。Dial の値によって音の大きさ、音の高さ、発音タイミング ( $-64ms < \Delta t < 64ms$ ) を変化させた。駒の反転に応じて白い駒には丸い印象の音を、黒い駒には角ばった印象の音をそれぞれ当てはめた。

オセロのロジックの導入により、Dial に「白と黒」という新たなパラメータが付加され、「反転」という変化が生まれ、その変化が再生される音響に影響するようになった。また、鑑賞者による操作は制限された。例えば、いきなり隅のマスに駒を配置するようなオセロのルールに反する操作は不可能になった。さらに、Dial が 8x8 個に増えたことで、全体として再生される音響の中でどの Dial がどの音を発しているのかが分かりにくくなった。

### 3.3 作品 3: <Au-thello 2>

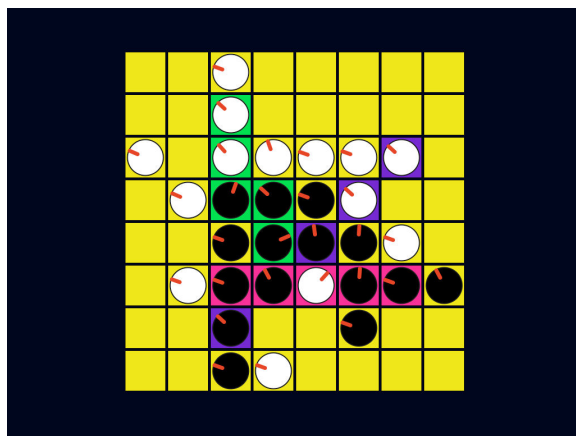


図 3 <Au-thello 2> スクリーンショット  
Figure 3 <Au-thello 2> Screen view of GUI.

<Au-thello>のプログラムを基に制作した。主な変更として、次の2つの新機能を追加した。1つ目は特殊操作を可能にする<Sync 機能>である。<Sync 機能>を使うと鑑賞者は盤面上の Dial の中から自由に複数を選択しグループ化できる。図3で色が変わっているマス（ピンク、緑、紫）がそれぞれグループ化されたものの例である。グループ化された Dial のうちひとつの値を変化させると、同グループ内の他の Dial の値も同様に变化する。<Sync 機能>を使えば鑑賞者は音響により大きな変化を加えることができる。2つ目は演奏補助を目的とする<LED 機能>である。<LED 機能>によって Dial が音を発する際にマスを点滅させ、どの Dial がどの音を発しているのかを示した。

<LED 機能>は Dial と音の対応関係を明確に示した。と同時に、画面の点滅の様子ランダムであるように見えることで、発音のタイミングがでたらめに入力されたものと変わらないように感じられるという面も暴いた。<Sync 機能>は音響により大きな変化を与えたが、グループが大きくなると盤面の大部分を占める Dial の値が同じものに近づいていき、全体の音響が単純になっていくという事が分かった。

## 4 考察

ここでは、<Au-thello>にオセロゲームのロジックが持ち込まれたことにより得られた成果と課題について「わかりやすい」という視点で考察し、<Au-thello>での例から

「専門性の高さゆえに敬遠されがちなシステムにゲーム性を付加することの利点や問題点に関して」可能な限り一般的な結論を得ることを試みる。

### 4.1 作品 1 から作品 2 への考察

ここで初めて作品にオセロゲームが導入される。それにより<Au-thello>にとってプラス・マイナスとなる両方の結果が生じている。

好ましい結果の1つ目に、もともと単なるダイアルだったものが白黒に反転する駒になったことが挙げられる。「白と黒」のパラメータで2種類の音色を操作することで再生される音響が以前より豊かになった。

これをシステムの視点から見てみると、「白と黒」という鑑賞者が扱うべき新たなパラメータが追加されたという事と「反転」という鑑賞者の入力をダイレクトに反映させないような処理が追加された事により、システムがより複雑になっていると言える。ではシステムは「わかり」辛くなったかという点、まだ考察の余地がある。なぜなら、ここでシステムに生じた複雑さである「白と黒」と「反転」はオセロゲームのロジックそのものである。オセロゲームは一般に広く親しまれているゲームであり、白と黒の駒を使うことや挟まれた駒は反転することを含め、ゲームのルールは既に鑑賞者に受け入れられたものである。駒の色が変わることで音色が変わることも白と黒の駒がもともと2人の対峙するプレイヤーを想定していることを考えれば自然なことである。

ここに、ゲーム性を付加することの利点として「ゲームの複雑さをシステムに付加しながらも、鑑賞者に覚えさせる複雑さを小さく抑えることができる」ということが言えるだろう。逆を言えばオセロをプレイしたことがないような鑑賞者にとってシステムは非常に「わかり」辛く、オセロゲームの導入はシステムにとって不利益ではない。

2つ目の結果として、操作できるダイアルの数がゲームのルールにより限られたことが挙げられる。鑑賞者の操作はオセロのルールの制約を受けるようになり、操作の選択肢が絞られることで操作の方向性が示された。

示された方向性とは具体的に次の2つのようなものである。1つ目は、入力を選択肢が絞られたことで、鑑賞者には「何を入力すればいいのか」がわかるようになったことである。全体としては64個用意されたダイアルも、ゲーム開始時点で操作できるものは4つのみであり、新たにダイアルを増やせる場所も4つのみである。単に入力の総数を減らすのではなく、潜在的な数は保ちながら入力を選択肢を絞ることに成功している。2つ目は、入力を選択肢が鑑賞者自身によって徐々に増やされていくことで、鑑賞者は全ての入力を偏りなく使うことができるようになったことである。オセロのゲームを最後まで進めることで鑑賞者は全てのダイアルを必ず使うことになる。

もし、64個のダイアルがいきなり提示されれば、初心者にとってその全てを使い切ることでさえ難しかっただろう。

これをシステムの操作の視点から見てみると、操作に禁則事項が追加されたことで、操作はより複雑になっていると言える。しかし、ここで生じた複雑さはオセロゲームのルールに由来し、これも既に鑑賞者に受け入れられたものである。駒を配置できるマスが決まっていること、ゲームを進めるほどに盤上の駒が増えていくこと等はごく自然のこととして受け入れられる。

ここに、ゲーム性を付加することのもうひとつの利点として「ルールの複雑さを操作に付加しながらも、鑑賞者に覚えさせる複雑さを小さく抑えることができる」ということが言えるだろう。再び逆を言えば、オセロを知らない鑑賞者にとってルールに沿った操作は単なる特殊操作でしかなく、非常に「わかり」辛いものになるだろう。

一方で、好ましくない結果として、オセロの規格に従うためにダイアルの数が大幅に増えたことが挙げられる。上記の通り、ゲームの進行状況によって入力を選択肢は絞られているが、そもそも、音響再生が目的であれば64個ものダイアルが本当に必要であるかは定かではない。ダイアルが増えるということは、入力が増え、「わかり」辛いということに繋がる。実際に8x8個のダイアルのうち、どのダイアルがどの音を出しているのかが分かりづらくなったという結果が出ている。

ここに、ゲーム性を付加する際の問題点として「ゲームの規格に従うために、本来の目的に反する複雑さがシステムに持ち込まれることがある」ということが言えるだろう。もっと少ないダイアルでより効率の良い音響生成ができるのであれば、入力の数はい少ない方が「わかりやすい」システムに繋がるはずである。

#### 4.2 作品2から作品3への考察

ここで作品に2つの新たな機能が追加された。それにより<Au-thello>の演奏ツールとしての能力は向上したが、問題点も見つかった。

まず、<LED機能>である。この機能によってどのダイアルがどの音を発しているのかが一目で分かるようになった。これは前作での問題を受けて追加した機能だが、この機能によってダイアルと出力の関係はとも「わかりやす」くなった。

次に、<Sync機能>である。この機能によって鑑賞者は再生する音響により大きな変化を加えることができるようになった。この機能はマウスによる入力の限界を補う為に考えられた。例えば、もしこのGUIが実物の駒と盤を使って操作できるインターフェースであれば鑑賞者は両手を使って複数の駒を同時に操作できる。そこで、複数の駒を同時に操作する最も簡単な仕組みとして<Sync機能>を考えた。しかし、この機能はオセロのロジックとは無関係であり、システムが「わかり」辛くなる原因になる恐れがある。もし、同じような特殊機能が次々と追加されれば、システムは必ず「わか

り」辛くなるだろう。つまり、新たなアイデアはゲームのロジックに沿った形で実現されなければならない。ここに、ゲーム性を付加する際の問題点として「ゲームのロジックを逸脱するアイデアが排除されてしまう」という仮定が考えられるが、この検証は<Au-thello>にさらなる変更を実験的に加えながら試みたいと思う。

## 5 まとめ

以上より<Au-thello>に関して次のことが言えた。

- オセロのロジックを取り入れたことで、鑑賞者には「わかりやすい」形でシステムをより複雑にできた。
- オセロのルールを取り入れたことで、鑑賞者には「わかりやすい」形で操作を限定したり、特殊な操作を課したりできた。
- オセロの規格に従ったことで、システムの一部は鑑賞者にとって「わかり」にくいものになった。

また、システム一般に関して次のことが言える。

- 一般的なゲームをシステムに取り入れることで、ゲームの複雑さをシステムに付加しながらも、利用者に覚えさせる複雑さを小さく抑えることができる。
- 一般的なゲームのルールをシステムの操作に取り入れることで、ルールの複雑さを操作に付加しながらも、利用者に覚えさせる複雑さを小さく抑えることができる。
- 一般的なゲームをシステムに取り入れる際にゲームの規格に従うことで、システムに望まれない複雑さが持ち込まれる可能性がある。

## 参考文献

- 1) ヤマハ TENORI-ON 日本語公式サイト  
<http://www.yamaha.co.jp/tenori-on/>