

映像の操作によるリアルタイム楽曲制御

小林和彦（正会員）

関東学院大学 人間共生学部 共生デザイン学科

Real-time music control by changes in images.

Kazuhiko Kobayashi (Member)

College of Interhuman Symbiotic Studies, Kanto Gakuin University

kobaya4 @ kanto-gakuin.ac.jp

アブストラクト

本稿は、ゲームエンジンである Unreal Engine4 を用い、作者の操作に応じて映像を変化させて、楽曲をリアルタイムに制御する手法の研究について述べたものである。具体的には、発音するオブジェクトを 3DCG の空間内に配置し、作者の操作によって空間を移動する事で音の時間的な組み合わせ方を変化させる。そして、この手法を映像作品の制作に活用するためのモチーフや表現方法、プログラムについて検討するため、1 点の試作映像と 4 点の映像作品を制作した。本稿ではこれらの作品の詳細について述べた上で、本研究の手法が作品の制作過程に及ぼす影響を明らかにする。

キーワード：3次元コンピュータグラフィックス、音楽、ゲームエンジン、リアルタイムアニメーション、

Abstract

This paper is describing a method of using Unreal Engine 4 to control music according to the change of images by artist operation. Specifically, a sound playing object is placed in a 3DCG space. Artist is operating a camera to moving a 3DCG space, thereby sound combination is change. For using this method on work production, I examined motifs, expression methods, programs. In a research created one prototype video and four works. And, clarify an influence on this method to production process of work.

Keywords: Three-dimensional computer graphics, Music, Game Engine, Real-time animation,

1. はじめに

近年, Unreal Engine4 (以下, UE4と略) [1]やUnity[2]といったゲームエンジンと, 高性能なグラフィックボードの普及によって, リアルタイム3DCGアニメーションの制作は身近なものとなった. また, ゲームエンジンはゲーム制作のために設計されているので, 鑑賞者の操作に反応する双方向性を持った作品制作も可能となる. 加えて, UE4はコードの記述ではなく, ノードを接続する視覚的なプログラミングであるため, 筆者のようにプログラミングの専門技術を学んでいない者であっても, 操作可能なリアルタイム3DCGのシステムを構築できる.

筆者はこれまで, 音と映像のタイミングを同期させた映像作品を複数制作している. fill up [3]では, まず音楽制作ソフトであるLogicにて作曲を行い, その楽譜での発音時のタイミングを動画の時間軸であるフレーム番号へと変換, フレーム番号に基づいて動画制作ソフトであるAfterEffectsでオブジェクトに動きをつけている. この作品では, 完成している楽曲に合わせて映像を制作している. 一方で, delusional sculpture [4]では, 3DCG制作ソフトであるMayaでボールが落下する映像を先に制作し, ボールが板に衝突する瞬間のフレーム番号を基にLogicで音を配置している. この作品では完成している動画に基づいて楽曲を制作しているため, ボールが板に衝突するタイミングは不変であり, 楽曲を後から変化させる事は難しい. このように, 筆者のこれまでの作品制作では, 時間軸を揃えて音と映像を個別に制作し, AfterEffectsなどの映像編集ソフトで一つの動画ファイルに結合する事で, 音と映像を連携させていた.

これに対してUE4では, プログラムによってオブジェクトの動作に合わせて音を出力したり, オブジェクト自体をランダムに出力したりと, 音と映像を同時に制御する事が可能となる. これは, 個別に制作した音と映像を組み合わせる筆者の従来の制作方法とは大きく異なるため, UE4のプログラミングと双方向性によって表現の幅を広げられると考えた.

本研究は, 映像として表示された空間を移動する事で楽曲をリアルタイムに制御する手法を検討し, 映像作品の制作に活用する方法を明らかにする. 具体的な方法としては, 発音するオブジェクトを配置した空間をUE4によって構築し, 作者が操作するカメラに連動したオブジェクト (以下, トリガーと略) との接触によって発音する. 筆者はトリガーの位置や移動速度などを変化させる事によって, 複数の音の組み合わせ方を変化させて楽曲を制御する.

そして, 上記のUE4のシステムを用いた映像作品の制作を通して, 本研究の手法に適したモチーフや表現方法, プログラムなどについて検討する. また, 作品ごとに筆者の操作によって移動可能な範囲やカメラアングルを変える事で, オブジェクトの発音時の視認性を高め, 空間把握をしやすい画面構成はどのようなものであるかの検討も行う.

2. 従来手法との比較

2.1 映像から音を生成する先行事例

映像から音を生成する手法としては, Virtual ANS[5]やPhonoPaper [6]のように画像の色情報や輝度情報を元に音色を生成するものから, 近年ではImaginary Soundscape[7]のように画像の持つ意味と音の持つ意味を結びつけるものが挙げられる. これらの事例では音単体や環境音を作り出す事を目的としていて, リズムやメロディを構築する本研究とは異なる方向性であるといえる.

画像から楽曲を生成する手法としては, mupic[8]のようなアプリや, Musical World Map[9]やBirds on the Wires[10]といった映像作品のように, 被写体の形状や配置に音符を割り当てたものがある. いずれの手法も時間軸の無い静止画をベースとしたものであり, リアルタイムに映像と楽曲を変化させる事を目的とした本研究とは異なる方向性であると考えられる.

2.2 音と同期した映像表現の先行事例

音に合わせて映像が出力される表現としては, Audio Architecture [11]のように楽器音と同期したオブジェクトを表示していくものや, MUSIC PLAYS IMAGES X IMAGES PLAY MUSIC[12]のように演奏に合わせてリアルタイムに映像を出力するものが挙げられる.

これらの事例に共通するのは, 楽曲の形は先に完成しているという点で, Audio Architectureは録音された音源に基づいて映像が制作され, MUSIC PLAYS IMAGES X IMAGES PLAY MUSICは楽譜に基づいて演奏が行われている. ミュージックビデオであるAudio Architectureでは音楽が表現の主体であり, 楽曲の魅力を引き出すために映像による表現を付加している. MUSIC PLAYS IMAGES X IMAGES PLAY MUSICは楽譜が表現の根幹にあり, 生演奏と同期した映像演出によってリアルタイム性のある視覚的表現を構築している. このように, 列挙した事例では音楽が表現の主体である.

これに対して本研究では, 映像を操作する事で音の組み合わせを変化させていく点に差異がある. 原曲をそのまま演奏する先行事例に対して, 原曲とは異なる楽曲へと変化させる事を目的とした本研究では, 音楽表現の方向性としては大きく異なると考えられる.

2.3 リアルタイム生成を行う先行事例

前項の方向性を発展させた事例として, 音楽や映像をリアルタイムに生成していくものがある. 楽曲をリアルタイムに生成していく事例としては, Alex McLean (Yaxu) live on DOMMUNE tokyo[13]のように, 即興でコードを記述するライブコーディングによってシーケンサーを制御するものがある. この事例では, プログラミングを使用してリアルタイムに楽曲を変化させていくという点で本研究と類似しているため, 差異を詳細に考察する. 当該事例では, コードの記述によってシーケンサーに具体的な演奏指示を行い, 作者の狙い通りの楽曲を生成する事が可能である. そして, 生成された楽曲を確認しながらコードを書

き換える事で、作者が理想とする楽曲を構築していく。一方、本研究ではプログラミングを作成した筆者自身であっても、楽譜となる映像を見た段階で演奏結果を完全に予測する事は困難である。本研究はクオリティの高い楽曲を構築するのではなく、映像の操作によって筆者自身の想像とは異なる表現を生み出す事を目的の一つとしているので、方向性としては異なるものであると考えている。

本研究と同様にゲームエンジンを用いて映像をリアルタイム生成する事例もある。Roland Groovebox MC-101 live performance at Channel #21 (with Unity visualizer) [14]は、ループシーケンサーから出力されるMIDI信号を利用してゲームエンジンのUnityを制御し、リアルタイムに映像を出力している。この事例では、タイトルの一部にUnity visualizerとあるように、楽器演奏によって音が出力され、映像は演奏に応じて変化するので、映像をコントローラーとする本作とは異なる方向性であると考えられる。

2.4 双方向性を持つ先行事例

リアルタイムに映像と音が出力されるという点では、Rez Infinite[15]やChild of Eden[16]などのゲームが挙げられる。これらのゲームでは、照準を目標物に合わせてロックオン操作を行い、複数の目標物をロックオンしてからボタン操作を行って高得点を獲得する事が目的の一つとなる。ロックオンやボタン操作を行う際に、BGMに合わせた効果音を出力する事で、鑑賞者は音楽を演奏している感覚を得る事が出来る。また、VOXQUARTER 2016[17]のように、ゲームとしての進行方法はコマンド操作による行動決定としながら、コマンド操作によって楽曲を構成するフレーズが変化する事で鑑賞者が楽曲を能動的に構築していく事例もある。

これらの事例では、鑑賞者の操作によって楽曲が破綻しないように、BGMや発音される音を作者側で調整している。このため、作者の意図した方向性とは全く異なる楽曲を鑑賞者の操作によって構築する事は出来ないが、音楽の知識や技術の無い鑑賞者であっても演奏を行っている感覚を得る事が出来る。一方、本研究は演奏速度を極端に変化させる事によって、場合によっては無音状態が数秒間継続する範囲まで、音の組み合わせを変化させられるという点に差異がある。本研究で制作した作品の一部は、筆者自身が操作に習熟しなければリズムやメロディとして成立しないものもあり、この点は列挙した事例とは大きく異なるといえる。

鑑賞者の操作と得点が密接に関連するゲームとしては、SOUND VOLTEX[18]やBeat Saber[19]などが挙げられる。これらは画面内を移動するオブジェクトが、操作タイミングを示す線などに接触するタイミングに合わせて鑑賞者が操作を行うと、オブジェクトが発音して楽曲が演奏される。これらのゲームに共通するルールとして、鑑賞者の操作のタイミングが正確であれば得点は高くなり、タイミングがずれている場合は操作ミスとして得点が低くなるか無得点となり、一定回数の操作ミスを行うとゲーム終了となる。鑑賞者が高得点を獲得するためには、楽譜を記憶して演奏の正確性を高める事が必要となる。

これに対して本研究には得点の概念が無く、オブジェクトの

発音操作は自動で行われるために操作タイミングの正確性は不要で、筆者の操作は空間の移動によって楽曲を変化させる事を目的としている点が異なる。

3. 本手法での提案

本研究では、音符となるオブジェクトの配置された空間を楽譜として見立て、その映像を見ながら筆者がトリガーを操作して移動し、オブジェクトと接触する事で発音する手法（以下、本手法と略）によって、リアルタイムに楽曲を変化させる事を目的とする。具体的には、楽曲を構成する要素であるリズム、メロディ、テンポ、楽器の種類を、トリガーの位置や移動速度の操作によって変化させる。

上記の要素のうち、リズムとメロディについて補足する。筆者は前述したfill upの他に、「monochrome」「still phantom」などの映像作品のBGMを作曲しているが、あくまでも映像作品が表現の主体であって音楽は補助的な役割として用いられていて、筆者自身は音楽制作を主とした作家ではない。また、本研究においては映像の変化によって楽曲を変化させる事を目的としているので、本稿では楽曲のクオリティについては言及せず、音楽中辞典[20]での定義を参考として、最低限のリズムやメロディを構築し、これらを映像の操作によって変化させる事を本研究の目的とする。楽曲の変化を作り出す詳細な手順は、各作品の項目にて述べるが、例えば複数の単音が連続で演奏される事で構築できるメロディやリズムは、トリガーの移動によって音の組み合わせを変化させる。また、テンポはトリガーの移動速度によって変化させている。

そして、本手法のリアルタイム性や双方向性を映像作品の制作に活用する方法を検討した上で、本手法を作品制作に取り入れる事で得られる効果を明らかにする。これらの目的を達成するため、本研究では1点の試作映像と4点の映像作品の制作を行った。本稿では4点の映像作品の制作手順や作品の特性、問題点と改善点、先行事例との差異の考察などについて述べる。なお、試作映像はUE4の技術的な検証が主体となるため、本稿では詳細を割愛して概要のみ述べる。

試作映像[21]は、日本映像学会第42回大会での研究発表 [22]の中で上映したものであり、ランダムなタイミングでランダムな音高(音の高さ)の単音を繰り返し発音するオブジェクトを、3DCG空間に配置している。音は距離に応じて減衰するので、筆者の操作によってトリガーを移動させてオブジェクトとの距離を変化させる事で、聞こえる音の組み合わせをリアルタイムに変化させた。試作映像によって本手法が実現可能である事を確認できたので、映像作品の制作に着手する事とした。

作品制作にあたって、メロディとリズムの定義を改めて確認する。メロディ(旋律)は、「音の連続的な高低変化が、リズムと組み合わせられ、あるまとまった音楽的表現を生み出しているもの」[23]とされている。したがって、メロディを成立させるためにはリズムを成立させる必要がある。西洋音楽におけるリズムは「一定の時間的間隔をもって刻まれる拍によって、楽曲のリズム構造が律される事が多い」[24]とされている。

これらを踏まえて、まずはドラムキットの楽器音を発音するオブジェクトを一定の周期で空間に配置して、リズムを構築する手法を検討する。

4. 制作例

4.1 テンポと楽器構成の変化

本項で取り扱う映像作品の Roll way[25]は、日本映像学会第43 回大会にて作品発表[26]を行ったもので、サイズが幅 1,280 画素、高さ 720 画素、フレームレートが 29.97fps、作品時間が 2 分 40 秒である。

試作映像によって音と映像のリアルタイム表現の基礎的な仕組みは完成したので、Roll way ではドラムキットの楽器音を発音するオブジェクトを一定の間隔で配置してリズムが成立するように空間を構築し、映像作品としてまとめる事を目指す。

まず、作品としての空間や時間軸を構成するためのモチーフについて検討する。試作映像ではオブジェクトが静止したまま音を出力していたため、発音を視覚的に認識する事が難しかった。この問題を解決するため、Roll way ではトリガーがオブジェクトを通過した際に発音した事を表現できるモチーフを選定する。筆者は道路を自動車で行っている時に、目の前を通過していく白線やトンネルのライトが一定のリズムで流れていると感じる事がある。この体験を踏まえ、道路を五線譜に見立て、白線やガードレールなどを音符に見立てたモチーフとして選定し、トリガーである自動車が行き進む事でリズムが構築される方向性を選択した。楽曲を視覚的に表現するという観点からも道路と楽譜にはいくつかの類似点があり、本手法を表現するモチーフとして適していると考えている。まず、道路は進行方向が決まれている、楽譜は左側を起点にして右側へ時間軸が流れていくので、どちらも進行方向が一定である (図 1)。

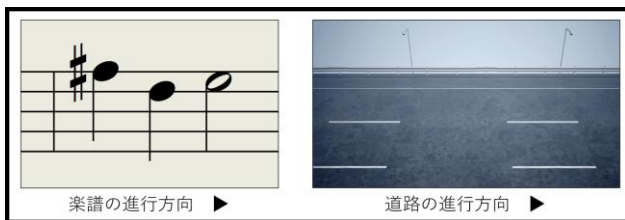


図1. 楽譜とRoll wayの道路を横から見た場合の進行方向。

また、複数の楽器を使用した楽曲を音楽制作ソフトで表示すると、楽器や波形ごとにトラックが割り当てられる。この点は道路の車線と類似していて、車線ごとに楽器を割り当てるという形で表現に組み込めると考えた (図 2)。

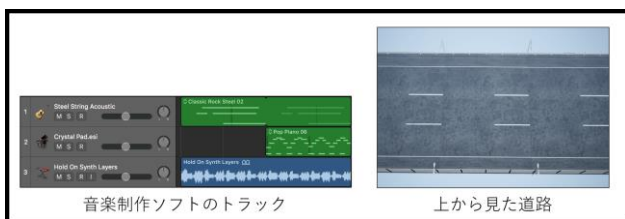


図2. 音楽制作ソフトのトラックと道路の車線の比較。

次に、カメラアングルについて検討する。UE4 を映像作品の

制作に取り入れるにあたって、3DCG をリアルタイムに描画可能な特性を活用したいと考えていたので、自動車の先端にカメラを設置し、道路の手前から奥に進んでいく、奥行きを感じられるような画面構成とした。

Roll way の発音の仕組みと、制作手順について述べる。道路上に白線やガードレール、看板などのオブジェクトを設置して、オブジェクトとトリガーが接触すると発音し、自動車が道路を進んでいくとリズムが構成される。UE4 にはレースゲームを作成するためのテンプレートがあり、アクセル、ブレーキ、ステアリングなどの操作に関するプログラムが組み込まれている。アクセルとブレーキの機能によってトリガーの移動速度を変化させる事で、楽曲のテンポを変更できるようにした。加えて、ステアリング操作によって車線変更を行う事で、発音する楽器を切り替えられるようにしている。

Roll way では、一定の間隔で道路に設置されている白線にはバスドラムの音を割り当て、楽曲全体を通してリズムの土台部分を構成している。白線は全ての車線に設置されているので、走行する車線とは無関係にバスドラムの音は常時発音される。白線によるリズムを一定にする事で、テンポの基準を明確にするとともに、アクセルとブレーキによるテンポの変化を強調する機能を持たせている。車線は 3 本で構成され、左右車線の端に設置されたガードレールと照明には、左右で異なる時間間隔かつ異なる音色のスネアドラムとハイハットの音を割り当ててある。これにより、左車線と右車線の車線変更によってリズムと楽器音の両方が変化する。なお、トリガーの接触範囲は 1 車線のみに影響を及ぼすように調整してあるため、中央の車線を走行する場合はバスドラムの音のみが発音され、右車線と左車線の音が同時に発音する事は無い (図 3)。下図では時間軸をわかりやすくするため、右上方からトリガーを見下ろすような構図としている。また、作品本編では表示されない自動車とトリガーの接触範囲を、図説の一部として表示している。

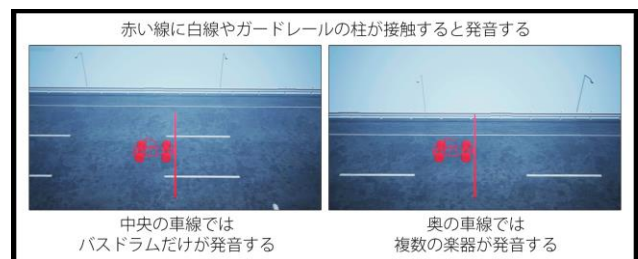


図3. 車線と楽器の配置。

ここまでで述べた基本的な楽器の構成に加えて、例えば右車線の一部には橋脚を支えるワイヤーを設置するなどして、楽器の種類やリズムに変化をつけて、作品全体として単調にならないように構成している。

Roll way では、自動車や道路などのモチーフを用いて、移動速度の変によって楽曲のテンポを変化させて、左右移動によって楽器の種類を変更できるようにした。結果として、筆者による映像の操作によって楽曲を制御するという本研究のコンセプトを作品として具現化する事ができた。

その反面、問題点もあった。作品時間である 2 分 30 秒間走

行を続けられる道路の全長は 6km となり、道路を構成するオブジェクト数は約 6,700 個に及んだ。このため、コンピュータへの負荷が高くなって UE4 の動作が不安定になり、広大な空間で多数のオブジェクトを管理する事自体に作業時間を浪費する頻度が高くなったため、作品全体の流れの構成や、楽器やリズムの組み合わせを試行錯誤する時間を十分に確保できなかった。次に制作する作品では、本手法を具現化した上で、負荷などの技術的な面でも最適なモチーフを検討する必要がある。

4.2 リズムとメロディの制御

本項で取り扱う映像作品の Flow Note[27]は、日本映像学会第 44 回大会[28]にて作品発表を行ったもので、サイズが幅 1,920 画素、高さ 1,080 画素、フレームレートが 29.97fps、作品時間が 3 分である。

まず、本手法での表現に最適なモチーフについて、Roll way で明らかになった問題点を踏まえて再考する。道路のような面積の大きい背景を奥方向に配置すると、コンピュータの負荷が高くなるので、現実世界を再現する方向性ではなく、3DCG の仮想空間の中で抽象的な表現を行う事とした。UE4 には Cascade[29]という低い負荷で粒子を出力する機能が搭載されているので、発光する粒子をモチーフとして選定し、何も配置されていない黒い空間に粒子を浮遊させる事とした。

粒子をモチーフとしたのは、表現の自由度を高める点でも有用性があると考えたからである。Roll way では道路をモチーフとしたために、移動可能な範囲が奥方向と左右方向に限定されていた。しかし、Flow Note のように何も無い空間に粒子を浮遊させる表現手法であれば、上下にも移動が可能となる。Roll way と同様に左右方向の移動によって楽器の種類を切り替えられるだけでなく、上下方向に移動して音高に変化を付ける事もできる。また、Roll way では現実世界の道路を再現するため、ガードレールや照明を特定の間隔で配置する必要があり、音の時間的な配置に制約があった。一方、Flow Note では光の粒子という抽象表現を用いるため、自由に音を配置する事が可能となる。Flow Note の音高の配置について述べる。空間の上部には高い音の粒子を配置し、粒子の位置が低くなると音も低くなる。複数の音高を配置する事で、音の連続的な高低変化であるメロディを表現できるようになった (図 4)。



図4. Roll wayとFlow Noteの比較。

UE4 を使用してリズムとメロディを構築する技術的な方法について述べる。筆者がこれまでに制作した、音と映像を組み合わせた作品では、楽譜に配置された音符の時間軸を秒数やフレーム数に変換する事で、音と映像の時間軸を同期させる方法

を用いていた。UE4 の時間軸は秒数によって管理できるため、上記で述べた方法を選択する事とした。UE4 での音符の実装方法について述べる。楽譜における「4分音符=120」のテンポ表記の場合、1 分間に 4 分音符が 120 個ある事を示していて、4 分音符 1 個あたりの演奏時間は 0.5 秒となる。また、例えば 4 分音符の演奏時間を半分にした時間である 0.25 秒を 4 分音符に加算すると、付点 4 分音符となる。このようにして変換した秒数を UE4 のプログラムに組み込むと、音符のように時間軸を扱う事ができる。そして、演奏する音符の種類と音高をランダムで選択するようにプログラムして、演奏ごとに異なるリズムとメロディが構成されるようにした (図 5)。なお、UE4 のプログラム作成画面のスクリーンショットでは文字サイズが小さいため、本稿では簡略化した図を掲載する。

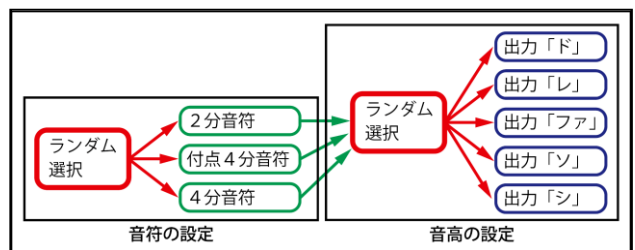


図5. Flow Note の粒子を出力するプログラム。

作品時間と楽器の構成について述べる。Roll way では道路の全長による作品時間の制約があったが、Flow Note では粒子を無制限に出力できるので制約は無い。筆者はこれまで制作した作品の多くで作品時間を 3 分に設定していたので、Flow Note も同様に 3 分とした。楽器の構成については、Flow Note では Roll way の道路よりも左右方向の空間の広さがあるので、楽器の種類をコーラス、シンセパッド、鉄琴、ピアノ、ハーブの 5 種類に増やしている。そして、3 分という範囲の中で楽曲全体の流れを検討し、やや低めのピアノの音から始まり、音高を上下に移動しながらハーブ、ピアノ、鉄琴、シンセパッドと続き、高音のコーラスで作品が終わるように構成した。

予め決められた配列の音の中を、速度を可変しながら移動する事で、楽曲のテンポを制御した Roll way に対して、Flow Note では音を発するオブジェクトをランダムに空間に配置し、筆者の操作によってメロディと楽器構成を変化させる事ができた。加えて、Roll way の技術的な問題を解決しつつ、音の配置の自由度を高めるモチーフとして粒子を選定し、本手法を活用できる表現の方向性を確立できた。

問題点としては、空間が広くなった分、音同士の時間の密度が低くなって無音の時間が長くなるため、メロディとして不自然な音の切れ目が発生しやすくなった。発音時間の長い楽器であれば、次の音の発音までの時間が多少長くても音は繋がりがやすいが、発音時間の短い楽器の場合は、トリガーが積極的に動いて切れ目なく発音させる必要が出てくる。結果として、操作中の画面の録音と録画の作業 (以下、録音と略) を約 100 回再試行していて、楽曲をリアルタイムに変化させるよりも、音同士の時間の隙間が生じないように次の音を探す操作に注力せざるを得ない面があった。この結果から、音をランダムに出力し

たり空間を広くしたりする際は、音同士の密度に注意する必要があると認識した。もう一つの問題点として、Roll way では道路やガードレールなどの面積の広いオブジェクトによって空間の奥行きが表現され、画面内にある要素の位置関係を認識しやすかったが、Flow Note では浮遊する粒子のみで空間を構成しているために、奥行き感覚が掴みにくく、粒子とトリガーの接触を視認する事も難しくなった。

こうした問題はあるものの、前述したように発音する粒子というモチーフは本手法に適していると筆者は感じたため、以降はFlow Note の設計をベースに、音を配置するプログラムやカメラアングルなどを調整する方向で研究を進めていく事とする。

ここまでの研究内容について、先行事例との差異を考察する。メロディをリアルタイムに変化させる先行事例としては、電子楽器のTENORI-ON[30]や、Zenbeats[31]のようなシーケンスソフトが挙げられる。これらの事例では、発音させる音高や楽器を楽譜画面や設定画面を操作して指定する必要があるが、Flow Note ではランダムな音高で粒子が自動的に配置され、操作としては空間の中を左右移動するというインタフェースの形に大きな差異があると考えている。

4.3 粒子の密度とカメラアングルの検討

本項で取り扱う映像作品 FN-02[32]は、日本映像学会第 45 回大会[33]にて作品発表を行っているもので、サイズが幅 1,920 画素、高さ 1,080 画素、フレームレートが 60fps、作品時間が 3 分である。FN-02 では、発音する粒子の密度を最適化した上で、発音時の粒子の動きを視認しやすいカメラアングルも検討する。

まず、発音する粒子の密度の最適化について述べる。Flow Note では全ての粒子をランダムに出力していたが、FN-02 ではランダムな出力だけでなく、筆者が予め組み立てたリズムも取り入れている。具体的には、8 分音符などの発音時間の短い音符は、音同士の間隔が大きくなると音に切れ目ができてしまうため、筆者が複数の音を組み合わせるリズムを構築している(図 6)。

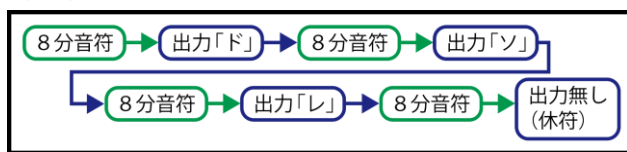


図 6. FN-02 で短い音符の粒子を出力するプログラム。

一方で、2 分音符などの発音時間の長い音符は、粒子の密度が低くても音の切れ目が発生しにくいので、Flow Note と同様のプログラムでランダムに出力している。このように、音符に応じて粒子を出力する方法を使い分けて、FN-02 では音の密度が低くなる状況を軽減できた。音の密度についての問題が解決したので、筆者は録音の際に楽曲の制御のみに注力できるようになった。

次に、発音時の粒子の動きを視認しやすいカメラアングルについて検討する。奥から手前に画面が移動していく Roll way と Flow Note に対して、FN-02 では画面が右から左へと移動する構図にして、トリガーと粒子の位置関係を視認しやすくし

た。カメラアングルの変更に関連して、FN-02 では音高と楽器音の配置を上下方向のみに統一した。具体的には、ピアノの低い音のグループの上にリズム楽器を配置し、その上に 1 オクターブ上のピアノのグループ、その上にシンセパッドの高い音のグループを配置して、上に移動すると音が高くなるのと同時に、楽器も変化していく構成とした。奥行きと上下左右の 3 方向の操作が可能な Flow Note に対して、FN-02 では上下左右の 2 方向へと移動範囲を整理する事で、録音時の操作性を高めた。

問題点として、短い音符の粒子を規則的に配置している事も影響して平面的な構図となり、映像作品としての魅力が乏しくなったと筆者は考えている。この点については、発音時の粒子の視認性を確保しつつ、画面の奥行きを表現できるカメラアングルを検討する事で対処する。問題点はもう一つあり、リズムとメロディを予め組み立てる事で、短い音の組み合わせがパターン化し、リアルタイムに楽曲を変化させるという本研究のコンセプトを損なっている面があった。こちらについては、短い音符の粒子をランダムに配置しつつも、無音時間が大きくなるようにプログラムを調整していく。

4.4 ビジュアルとプログラムの調整

本項で取り扱う映像作品の Floating Notes [34]は、サイズが幅 1,920 画素、高さ 1,080 画素、フレームレートが 60fps、作品時間が 2 分 30 秒である。Floating Notes では、映像として奥行きを表現できるカメラアングルを検討し、短い音符も含めた全ての粒子をランダムに出力しつつも、無音時間が適切になるようにプログラムを改善する。

まず、カメラアングルの検討について述べる。Floating Notes では、トリガーを右斜め上後方から見下ろし、粒子が画面右上から左下に移動しているように見える構図として、画面に奥行きを付加している(図 7)。

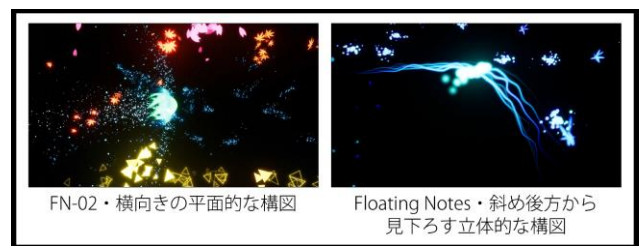


図 7. FN-02 と Floating Notes の構図の比較。

次に、粒子の配置について述べる。音高と楽器の配置を上下の一つの軸に統合した事で操作性を向上しつつ、メロディと楽器構成の変化を両立できた FN-02 の知見を活用して、Floating Notes でも音高と楽器の配置を一つの軸に統合している。具体的には、右に進んでいくと音が高くなっていき、高い音域まで進むと新しい楽器の粒子が混ざるように配置している。楽器の配置は、ピアノから始まり、途中からハーブが混ざりながらピアノが減少し、ハーブからベルに切り替わり、最後はシンセパッド単体の音で作品が終わる。Floating Notes では、音高の配置を変則的に行っていて、ピアノとハーブは低い音を左、高い音を右に配置しているが、ベルとシンセパッドは高い音を左、

低い音を右に配置している。これにより、ハーブとベルの高音を中間点として、楽曲の終盤に向かってベルとシンセパッドの音が低くなるという構成にしている。

プログラムの調整では、短い音符も含めた全ての粒子をランダムに配置する事とした。この際、無音時間が2秒(テンポ120における全休符)を大幅に超過しないようにしつつ、音と音の間隔が0.125秒(テンポ120における16分音符)を下回らないように留意し、リズムとして不自然にならないようにした。しかし、粒子は音高によって空間の左右にランダムに配置されるので、トリガーから離れた位置に連続して粒子が配置されて無音時間が長くなったり、トリガーの前方に粒子が密集して16分音符よりも極端に短い間隔で発音したりする事がある。

そこで、筆者自身が画面に流れてくる粒子を見ながら積極的に速度を変えて発音のタイミングを調整したり、右方向へ移動する事によって左奥にある粒子を回避したりして、音と音の時間的な間隔を調整する事とした。だが、筆者自身の操作のみで、不自然な音の間隔の発生を防ぐ事は難しいので、録音を繰り返しながら楽器や音高ごとの粒子の出力量を調整している。プログラムの修正と録音作業を繰り返していく中で、発音のタイミングの揺らぎによって、筆者でも予測できない音の組み合わせが生み出される事がある。この点については、筆者がこれまで映像制作ソフトなどで行ってきたような制作とは異なる魅力を、作品に付加できたのではないかと考えた。Floating Notesの録音結果全体を楽曲として成立させるためには、粒子の動きの癖を掴んだ上で操作方法に習熟する必要がある。これは、誰もが容易な操作によって音を演奏できる Lenses[35]のような作品とは対極の方向性にあると筆者は考えている。

Floating Notes では、カメラアングルの改善によって映像の立体感と発音時の視認性を両立し、プログラムの修正によってできるだけ自然な形で、リズムとメロディをランダムかつリアルタイムに構築する事ができた。

ここまでの研究内容について、先行事例との差異を考察する。移動によって音が変化する作品としては、in the depths[36]のように、サイン波の干渉による音の変化を、鑑賞者自身が移動しながら体験するものが挙げられる。これに対して Floating Notes では、音符がランダムに出力されていて、筆者の操作によってリズムやメロディがリアルタイムに変化するため、録音ごとに異なる楽曲が構成される点に差異があると考えられる。

一方、ゲームではゼルダの伝説 ブレスオブワイルド[37]のように、鑑賞者の操作するキャラクターの位置によって聞こえてくる音楽が変化するもの[38]や、NieR:Automata[39]のように場面に応じて、同じ楽曲で異なる楽器へとシームレスに切り替えられるもの[40]が挙げられる。このような、位置や場面によって音楽を変化させる手法は、3D空間を移動するゲームでは一般的に用いられている手法である。こうした手法では、予め完成している楽曲の一部のパートの音量などを状況に応じて可変させている。そして、楽曲を構成する音符同士の時間的な間隔は相対的には変化しないため、楽曲自体のメロディが変化するという事は無い。これに対して Floating Notes は、移動速度を極端に変化させて、音符同士の時間的な間隔を変化させて

いる点が大きく異なると考えられる。

5. まとめ

本研究によって、筆者が映像を操作して楽曲を制御する手法を映像作品の制作に活用できる事が明らかになった。4点の映像作品の制作を通して、本手法に適したモチーフ、カメラアングルなどの表現方法、音を制御するプログラムを見つける事ができたのではないかと筆者は考えている。

そして、本研究の最後に制作したFloating Notesでは、ランダムに出力された粒子によって映像の中に楽譜を構築し、映像を見ながら筆者が操作を行う事で楽曲を制御できた。こうした制作手法は、筆者がこれまで取り組んでいた、音楽制作ソフトや映像制作ソフトによって予め完成させた音と映像ファイルを組み合わせる手順と比較すると、映像を見ながらリアルタイムに変化を加えるという点が、大きく異なると考えている。

この他、楽譜の構築はコンピュータが主体となって行い、録音中の操作を筆者が主体的に行う事で、コンピュータによる自動生成と人の手による反応を組み合わせるといった制作方法を実現できた。この点は、本手法が作品の制作方法に影響を及ぼして、表現の幅を広げる上で有用性があったと考えている。

Floating Notesでは、本研究の手法を映像作品へと具現化する事ができた。今後は映像作家としての筆者の立場からFloating Notesの作品全体としての品質を高め、コンテストや映画祭など、社会的な評価を受ける場での発表も視野に入れていきたい。

参考文献

(全ての Web サイトの最終参照は 2020 年 10 月 20 日)

- [1] Epic Games, “Unreal Engine 4”,
https://www.unrealengine.com.
- [2] Unity Technologies, “Unity”, https://unity.com/.
- [3] Kazuhiko Kobayashi, “fill up”,
http://k-kobayashi.info/movs/hd_fillup.html, 2002.
- [4] Kazuhiko Kobayashi, “delusional sculpture”,
http://k-kobayashi.info/movs/hd_delu.html, 2011.
- [5] Alexander Zolotov, “Virtual ANS”,
https://warmplace.ru/soft/ans/, 2020.
- [6] Alexander Zolotov, “PhonoPaper”,
https://www.warmplace.ru/soft/phonopaper/, 2020.
- [7] Qosmo, “Imaginary Soundscape”,
http://www.imaginarysoundscape.net/, 2017.
- [8] 株式会社ディーバ, “mupic”, https://mupic.jp/, 2019.
- [9] John Keats, “Musical World Map”,
https://youtu.be/F6va6tg62qg, 2017.
- [10] Jarbas Agnelli “Birds on the Wires”
http://vimeo.com/6428069, 2009.
- [11] Comelius, “Audio Architecture”,
https://youtu.be/RvNpY-gF_LE, 2019.
- [12] Ryuichi Sakamoto, Toshio Iwai, “MUSIC PLAYS
IMAGES X IMAGES PLAY MUSIC”, 1997.
- [13] Alex McLean “live on DOMMUNE tokyo, 14 Nov 2018”
https://www.youtube.com/watch?v=dIpzU71LAQQ, 2018.
- [14] Keijiro Takahashi, “Roland Groovebox MC-101
live performance at Channel #21 (with Unity visualizer)”,
https://vimeo.com/379562830, 2018.
- [15] Enhance, “Rez Infinite”,
https://www.jp.playstation.com/games/rez-infinite-ps4/, 2016.
- [16] Ubisoft, “Child of Eden”, https://www.ubisoft.co.jp/coe/,
2011.
- [17] 岩本翔, “VOXQUARTER 2016”, https://voxquest.tumblr.com,
https://youtu.be/kuh9FxG7azo, 2016.
- [18] 株式会社コナミアミューズメント, “SOUND VOLTEX”,
https://p.eagate.573.jp/game/sdvx/v/p/top/, 2020.
- [19] Beat Games, “Beat Saber”, https://beatsaber.com/, 2018.
- [20] 株式会社音楽之友社, “新編 音楽中辞典”, 2008.
- [21] 小林和彦, ゲームエンジンを活用した音と映像の表現・
試作映像, https://youtu.be/sEsZ_Lo2OpQ, 2016.
- [22] 小林和彦, ゲームエンジンを活用した音と映像の表現,
日本映像学会第42回大会概要集, pp.49, 2016.
- [23] 株式会社音楽之友社, “新編 音楽中辞典”, pp.368, 2008.
- [24] 株式会社音楽之友社, “新編 音楽中辞典”, pp.742, 2008.
- [25] Kazuhiko Kobayashi, “Roll way”,
https://youtu.be/I46gGymPM1o, 2017.
- [26] 小林和彦, 日本映像学会第43回大会概要集, pp.61, 2017.
- [27] Kazuhiko Kobayashi, “Flow Note”,
https://youtu.be/URVX8LVuQ3o, 2018.
- [28] 小林和彦, 日本映像学会第44回大会概要集, pp.75, 2018.
- [29] Epic Games, “Cascade”,
https://docs.unrealengine.com/ja/Engine/Rendering/
ParticleSystems/Cascade/index.html.
- [30] ヤマハ株式会社, “TENORI-ON”, https://www.yamaha.com/
ja/about/design/synapses/id_005/, 2008.
- [31] ローランド株式会社, “Zenbeats”,
https://www.roland.com/jp/products/zenbeats/, 2020.
- [32] Kazuhiko Kobayashi, “FN-02”,
https://youtu.be/LXbWb_9qdXE, 2019.
- [33] 小林和彦, 日本映像学会第45回大会概要集, pp.53, 2019.
- [34] K Kobayashi, “Floating Notes”,
https://youtu.be/5HtCwAIg4Hw, 2020.
- [35] HUSH, “Lenses”, https://heyhush.com/work/hush-lenses-
interactive-art-installation/, 2016.
- [36] The SINE WAVE ORCHESTRA, “in the depths”
https://www.ycam.jp/archive/works/the-sine-wave-
orchestra-in-the-depths/, 2017.
- [37] 任天堂株式会社, “ゼルダの伝説 ブレスオブワイルド”
https://www.nintendo.co.jp/zelda/index.html, 2017
- [38] 岩泉茂, “【CEDEC2017】「ゼルダの伝説」で、自由度の
高い「オープンエアー」の表現を支えるサウンド”
https://game.watch.impress.co.jp/docs/news/1078827.html ,
2017.
- [39] 株式会社スクウェア・エニックス “NieR:Automata”
https://www.jp.square-enix.com/nierautomata/, 2017.
- [40] 上田雅美 “NieR:Automata 開発ブログ
『NieR:Automata』BGM実装の開発裏話”
https://www.platinumgames.co.jp/dev-nier-automata/article/
155, 2017.

小林 和彦



2002年東北芸術工科大学デザイン工学部情報デザイン学科卒業。
2005年東北芸術工科大学大学院芸術工学研究科デザイン工学専
攻修了。2005年高知情報ビジネス専門学校CGクリエイター科
専任教員。2006年東北芸術工科大学情報デザイン学科映像コー
ス助手。2009年関東学院大学工学部情報ネット・メディア工学
科助教。2014年関東学院大学理工学部理工学科情報学系専任講
師。2017年関東学院大学人間共生学部共生デザイン学科准教授。
現在に至る。3DCGアニメーション、映像の合成による作品制
作を行う。日本映像学会、芸術科学会、会員。