

音楽リズムゲームにおけるノーツ自動生成システムの実装と検証

中野凱* 小高知宏** 黒岩丈介** 松田優也** 諏訪いずみ***

Implementation and verification of automatic notes generation system for music rhythm games

Gai NAKANO*, Tomohiro ODAKA**, Jousuke KUROIWA**
Yuya MATSUDA**, Izumi SUWA***

(Received January 15, 2026)

This study proposes an automatic notes generation method for music rhythm games. Music rhythm games are a genre of entertainment-oriented computer games in which players perform actions in synchrony with music or rhythm, and the on-screen "notes" serve as cues for player actions. However, in conventional approaches, notes placement is manually created, which requires considerable time and effort.

In this study, we developed an automatic notes generation method using music analysis techniques, aiming to generate notes patterns that correspond to the characteristics of the music. Furthermore, a playable music rhythm game environment incorporating the proposed system was constructed, and the effects of automatically generated notes on player experience and enjoyment were evaluated. The outcomes of this study are expected to contribute not only to the efficiency of notes creation but also to the overall advancement of music rhythm game development.

Key words : Music rhythm game, automatic generation, music signal analysis, gameplay evaluation

1. はじめに

音楽リズムゲームとは、プレイヤーが画面上に表示されるリズムアイコンに対し、楽曲に同期したタイミングで「叩く」や「弾く」などの操作を行うゲームジャンルであり、一般に「音ゲー」と呼ばれる。また、音楽リズムゲームにおいて使用されるリズムアイコンは「ノーツ」と称される。

現在、音楽リズムゲームにおける楽曲解析およびノーツ

配置は、主に人手によって行われている。制作者は、プレイヤーが無理なくプレイ可能であり、かつ遊戯性の高いノーツ配置を実現するために、楽曲を聴取しながら試行錯誤を重ねる必要がある。この作業には多大な時間と労力を要することが課題として挙げられる。

本研究では、楽曲解析手法を用いて音楽リズムゲームにおけるノーツの自動生成を行い、楽曲の特徴に適したノーツ配置の実現を目的とする。さらに、実際に音楽リズムゲームをプレイ可能な環境を構築し、ノーツ自動生成システムを導入することで、生成されたノーツについてプレイヤーによる評価を実施する。本稿では、特にプレイ時の面白さの観点から評価を行う。

ノーツ生成を自動化することにより、ゲーム制作における制作者の負担軽減が期待される。また、本研究では音響信号から直接ノーツを生成することを目指しているため、楽曲データの詳細な譜面情報が入手困難な場合においても適用可能である。このことから、個人

*1大学院工学研究科 知識社会基礎工学専攻

**Fundamental Engineering for Knowledge-Based Society, Graduate School of Engineering

**知能システム工学講座

**Department of Human and Artificial Intelligent Systems

***仁愛女子短期大学生活科学学科

***Jin-ai Women's College

開発者による音楽リズムゲーム制作支援や、ノーツ生成作業の補助的手段としての利用可能性が示唆される。

以下に本稿の構成を示す。第2章では、音楽リズムゲームおよびノーツに関する基礎的事項について述べる。第3章では、楽曲解析に用いた手法や音響信号の解析方法、ならびにノーツ自動生成システムの構成について説明する。第4章では、本研究で構築したゲーム環境および評価実験におけるプレイ方法について述べる。第5章では、自動生成されたノーツについて、手法および条件ごとの結果を示す。最後に、第6章において本研究のまとめと今後の課題について述べる。

2. 音楽リズムゲーム

本章では、本研究の理解に必要な予備知識として、音楽リズムゲームの概要および関連用語について述べる。特に、音楽リズムゲームにおけるノーツの役割と、代表的なゲーム形態について整理する。

2.1 音楽リズムゲームの概要

音楽リズムゲームとは、楽曲やリズムに同期して、プレイヤーが所定のタイミングで入力操作を行うコンピュータゲームの総称である。一般には「音楽ゲーム」や「リズムゲーム」などの名称でも知られており、俗称として「音ゲー」という呼称が広く用いられている。本稿では、これらを総称して音楽リズムゲームと呼ぶ。

音楽リズムゲームで使用される楽曲は多岐にわたり、ゲーム独自のオリジナル楽曲に加え、J-POP、アニメソング、クラシック音楽、ボーカロイド楽曲など、さまざまなジャンルの楽曲が採用されている。中には、数百曲から千曲近い楽曲を収録するタイトルも存在する。なお、楽曲の著作権や利用規約に関する詳細な議論は本稿の範囲外とする。

一般的なコンピュータゲームにおいても楽曲はBGMや効果音として使用されるが、音楽リズムゲームでは楽曲そのものがゲームの中心的要素となる点が大きな特徴である。そのため、ゲーム設計の過程において、楽曲の構造やリズムに基づいた解析が重要となる。

音楽リズムゲームは、音楽鑑賞としての側面と、操作を伴うゲームプレイの側面を併せ持つゲームジャンルである。視覚情報と聴覚情報を密接に連携させてプレイする点は、他のゲームジャンルにはあまり見られない特徴であり、音楽リズムゲームの大きな特性の一つである。

2.2 音楽リズムゲームにおけるノーツの役割

音楽リズムゲームにおいて、ノーツとは楽曲に同期して画面上に表示されるリズムアイコンを指す。プレイヤーはノーツの出現位置やタイミングを視覚的に認識し、対応する入力操作を行うことでゲームを進行させる。

ノーツが画面上に表示されることにより、プレイヤーは操作すべきタイミングや入力内容を直感的に把握できる。この仕組みによって、楽曲とプレイ操作の同期が成立し、音楽リズムゲームとしてのゲーム性が形成される。

図1に、シンプルな音楽リズムゲームのプレイ画面例を示す。図中のディスプレイには、レーンに沿って複数のノーツが一定方向に移動する様子が表示されている。これらのノーツが所定の判定位置に到達した瞬間に、対応する入力を行うことが求められる。

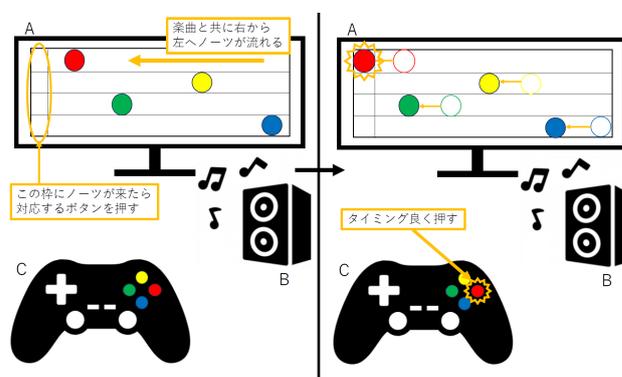


図1 音楽リズムゲームのプレイ画面の例1

ノーツには複数の種類が存在し、それぞれ異なる入力操作が割り当てられる場合が多い。ノーツの種類や数はゲームごとに異なり、少数のノーツのみを使用するものから、多様なノーツを組み合わせたものまで存在する。

ノーツ配置は音楽リズムゲームにおけるゲーム体験を大きく左右する要素である。楽曲のリズムやテンポと適切に同期していないノーツ配置は、プレイヤーに違和感を与え、ゲーム性の低下を招く。そのため、楽曲に適したノーツ配置を設計することが重要となる。

2.3 音楽リズムゲームの例

音楽リズムゲームの具体例として、異なる操作体系や表示形式を持つゲームが多数存在する。基本的なゲームプレイは共通しており、楽曲に同期して表示されるノーツに対し、適切なタイミングで入力操作を行う点に特徴がある。

図2は、タブレット端末でのプレイを想定した音楽リズムゲームの一例である。図1とは異なり、複数の

レーンが配置され、プレイヤーは画面上の対応位置を直接タップまたはフリックすることで操作を行う。

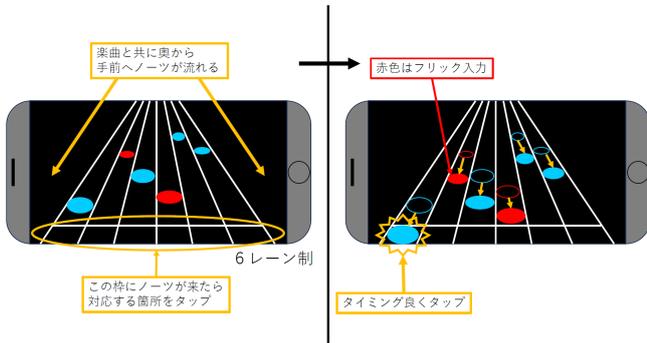


図2 音楽リズムゲームのプレイ画面の例2

また、家庭用ゲーム機やモバイル端末に限らず、ゲームセンター向けに専用ハードウェアを用いた音楽リズムゲームも存在する。例えば、「太鼓の達人」では、太鼓型の入力装置を用いてノーツに対応した操作を行う^[1]。「Dance Dance Revolution」では、床面に配置されたパネルを踏むことで操作を行う^[2]。

このように、音楽リズムゲームは使用する入力デバイスや操作方法が多様であり、それに応じてノーツの種類や配置も変化する。いずれの場合においても、楽曲とノーツ配置の調和がゲーム体験の質を左右する重要な要因である。

3. ノーツ自動生成システム

本章では、本研究で用いたノーツ自動生成システムについて述べる。具体的には、楽曲解析に使用した手法および音響信号からの特徴量抽出方法について説明し、さらに、音楽リズムゲームに適したノーツを生成するために導入した条件付けおよび処理工程について述べる。

3.1 楽曲解析に使用した手法

音楽リズムゲームにおけるノーツ自動生成を実現するためには、楽曲に含まれるリズム構造や音の発生タイミングを把握する必要がある。本研究では、楽曲ファイルに含まれる音響信号を対象として解析を行い、ノーツ生成に必要な情報を抽出した。

主な解析手法はオンセット検出、ビート検出、そして、それらの手法の基盤的な解析手法でのピーク検出である。それぞれを以下に説明する。また、楽曲解析を施した音響信号について、ノーツ生成の要となる特徴量の抽出方法に関しては、その際に使用した手法についても、数式を交えて解説をする。

3.1.1 短時間フーリエ変換

フーリエ変換は、時間領域の信号を周波数領域へ変換する手法であり、次式で表される^[3]。

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t} dt \quad (1)$$

この変換により信号全体の周波数成分を把握することができるが、時間的に局所的な周波数変化を捉えることはできない。音楽信号解析では、音の発生タイミングやリズムといった時間的な変化が重要であるため、通常のフーリエ変換のみでは不十分である。

そこで、本研究では短時間フーリエ変換を用いる。短時間フーリエ変換は、信号を短い時間区間に分割し、各区間ごとにフーリエ変換を適用することで、時間と周波数の両面から信号を解析する手法である。

離散時間信号 $f(t)$ に対する短時間フーリエ変換は、次式で表される^[4]。

$$F(m, k) = \sum_{n=0}^{N-1} f_m(n)e^{-i2\pi kn/N} \quad (2)$$

ここで、 k は離散周波数のインデックスを表す。短時間フーリエ変換では、信号を時間的に切り出すために窓関数を用いる。この窓関数により、非周期信号である音楽信号に対しても、局所的な周期構造を解析することが可能となる^[5]。

3.1.2 ピーク検出

ピーク検出とは、信号やデータ列における局所的な極大値または極小値を検出する手法であり、信号処理やデータ解析の分野で広く用いられている^{[6][7]}。

音響信号処理においては、ピーク検出は音の変化点や強調された部分を抽出するための基礎的手法として利用されており、オンセット検出やビート検出の基盤的処理として重要な役割を果たす^[8]。

ピーク検出は汎用性が高く、多くの解析環境で利用可能である。PythonのSciPyやMATLABなどには、ピーク検出を行うための関数が実装されており、本研究においても既存のライブラリを用いてピーク検出を行った。

3.1.3 オンセット検出

オンセットとは、新たな音が発生する、あるいは音が強調される瞬間を指す。音楽信号においては、振幅やスペクトルの急激な変化として現れることが多い。

オンセット検出は、音響信号処理において音の発生タイミングを推定する重要な手法であり、楽曲のリズム構造の把握に利用される^[9]。オンセット検出によって得られた情報は、楽曲解析やジャンル分類、音楽制作支援など、さまざまな分野で活用されている^[10]。

オンセット検出では、ピーク検出を用いて音響特徴量の変化点を抽出する機会が多い。ただし、ピーク検出そのものは極大値・極小値を検出する手法であり、オンセットと完全に一致するわけではない点に注意が必要である^[11]。

3.1.4 ビート検出

ビートとは、楽曲における規則的な拍の繰り返しを指し、楽曲のリズム構造を形成する基本的要素である。ビートの繰り返し速度はテンポと呼ばれ、通常はBPMによって表される。

ビート検出は、音響信号から楽曲のテンポや拍位置を推定する手法であり、オンセット検出と並んで楽曲解析において重要な役割を担う。ビート検出によって得られる情報は、楽曲分析やジャンル分類、ダンス振付支援など、幅広い応用が可能である^[12]。

本研究では、ビート検出の汎用性に着目し、ノーツ生成における基準タイミングとしてビート情報を利用した^[13]。

楽曲解析の実装には、Pythonの音響解析ライブラリであるlibrosaを使用し、オンセット検出およびビート検出を行った。

3.2 ノーツ自動生成までの流れ

本節では、ノーツ自動生成システムにおける処理の流れについて説明する。ノーツ生成までの工程は、大きく分けて3段階から構成される。すなわち、指定した楽曲ファイルからビートおよびオンセットを検出する検出工程、検出結果に基づいて特徴量を抽出する抽出工程、および抽出した特徴量に条件付けを行いノーツを生成する生成工程である。これらの処理の流れを図3に示す。



図3 ノーツ生成の流れ

3.2.1 検出工程

検出工程では、指定した楽曲ファイルに対してビート検出およびオンセット検出を行う。本研究では、Pythonの音響解析ライブラリであるlibrosaを用いて、楽曲の音響信号からビートおよびオンセットのタイミングを抽出した。

楽曲ファイルを入力とし、オンセット検出およびビート検出を適用することで、楽曲内における音の発生タイミングおよび拍位置が得られる。これらの検出結果は、後続の特徴量抽出およびノーツ生成の基準点として利用される。

検出結果の妥当性を確認するため、ビートおよびオンセットのタイミングを波形上に可視化した。図4は、ビート検出結果の一例であり、視認性を考慮して楽曲の一部分（10秒から20秒）のみを表示している。

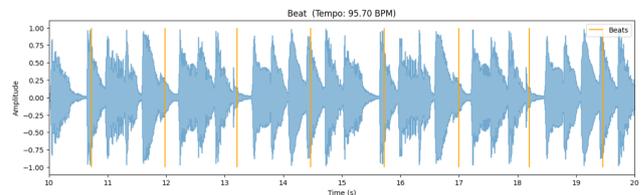


図4 各ビートのタイミング

3.2.2 抽出工程

抽出工程では、検出工程で得られた各ビート時およびオンセット時を基準として、音響特徴量を抽出する。本研究では、短時間フーリエ変換を用いて、これらのタイミング周辺における局所的な周波数成分を特徴量として取得した。

短時間フーリエ変換を適用することで、時間的に限定された区間の周波数情報を抽出することが可能となり、各ビートおよびオンセット時に対応する特徴量を得ることができる。これらの特徴量は、後述する生成工程におけるノーツ種別決定のための入力情報として用いられる。

抽出結果は、各ビートおよびオンセットのタイミングとともに一覧表示され、周波数成分の分布を視覚的に確認できる。図5は、各ビート時における周波数成分の一例を示したものであり、生成工程における閾値設定の参考情報として利用する。

3.2.3 生成工程

生成工程では、抽出した音響特徴量に基づいて条件付けを行い、ノーツを生成する。本研究では、主に周

ビートのタイミング: 11.39秒, 周波数: 1086.82 Hz
ビートのタイミング: 12.31秒, 周波数: 1296.73 Hz
ビートのタイミング: 13.21秒, 周波数: 891.41 Hz
ビートのタイミング: 14.13秒, 周波数: 960.90 Hz
ビートのタイミング: 15.03秒, 周波数: 1050.89 Hz
ビートのタイミング: 15.94秒, 周波数: 1274.84 Hz
ビートのタイミング: 16.85秒, 周波数: 934.48 Hz
ビートのタイミング: 17.76秒, 周波数: 1170.36 Hz
ビートのタイミング: 18.67秒, 周波数: 1252.01 Hz
ビートのタイミング: 19.57秒, 周波数: 1335.79 Hz
ビートのタイミング: 20.49秒, 周波数: 1063.96 Hz
ビートのタイミング: 21.40秒, 周波数: 1169.59 Hz
ビートのタイミング: 22.30秒, 周波数: 1156.62 Hz
ビートのタイミング: 23.22秒, 周波数: 1310.50 Hz
ビートのタイミング: 24.13秒, 周波数: 869.38 Hz
ビートのタイミング: 25.03秒, 周波数: 972.49 Hz
ビートのタイミング: 25.94秒, 周波数: 1099.93 Hz
ビートのタイミング: 26.85秒, 周波数: 1320.59 Hz
ビートのタイミング: 27.76秒, 周波数: 912.17 Hz
ビートのタイミング: 28.77秒, 周波数: 986.32 Hz

図5 各ビート時の周波数

波数成分に対して閾値を設定し、その値に応じて生成されるノーツの種類を決定した。

例えば、一定値以上の高周波数成分を持つ場合には特定のノーツを割り当て、低周波数成分が支配的な場合には別のノーツを割り当てるといった条件付けを行った。閾値の設定には、抽出工程で得られた周波数分布、図5を参考にし、高周波数帯および低周波数帯を区別する形で決定した。

このような条件付けは、既存の音楽リズムゲームにおいて、楽曲の特徴変化に基づいてノーツ配置が設計されている点を踏まえたものである。

さらに、生成対象とする楽曲の時間範囲を指定可能とし、楽曲全体ではなく任意の区間に対してノーツ生成を行えるようにした。図6に、周波数閾値および時間範囲の設定例を示す。

高周波数の値を指定: 1100
低周波数の値を指定: 800
開始時間を指定: 85
終了時間を指定: 95

図6 各条件設定

以上の条件を設定した後、各ビートおよびオンセットのタイミングに基づいて、対応するノーツが自動的に生成される。

4. ゲームシステム

本章では、本研究で提案するノーツ自動生成システムの有効性を評価するために構築した音楽リズムゲームシステムについて述べる。主に、ゲームシステムの構成および設計方針、ならびにゲームプレイの流れについて説明する。

4.1 システム構成と設計の概要

本節では、音楽リズムゲームにおけるノーツ自動生成を実際に評価するために必要となるゲームシステムの構成と設計について述べる。

本研究では、提案するノーツ自動生成システムを実際のゲーム環境に導入し、プレイヤーによる主観評価を行うことを目的として、独自の音楽リズムゲームシステムを開発した。本システムにより、自動生成されたノーツを用いたゲームプレイが可能となり、ノーツ配置の妥当性や面白さを評価できる環境を構築している。

開発した音楽リズムゲームシステムは、PythonおよびPygameライブラリを用いて構築されている。楽曲の音響信号解析にはlibrosaライブラリを使用し、ユーザーが任意の楽曲ファイル(mp3形式)を指定することで、ノーツの自動生成からゲームプレイまでを一貫して行える点を特徴とする。開発環境にはVisual Studio Codeを使用し、実行環境はローカルPC上での単体実行とした。

ゲーム全体の構成および処理の流れを図7に示す。ゲームプログラムを起動するとタイトル画面が表示され、その後メニュー画面に遷移する。メニュー画面では楽曲選択が可能であり、選択した楽曲に対してノーツが自動生成され、ゲームが開始される。ゲームはあらかじめ設定した一定時間が経過すると終了し、終了後にはスコア等を表示するリザルト画面が表示される。なお、ゲームプレイの詳細な操作方法については、次節4.2で述べる。

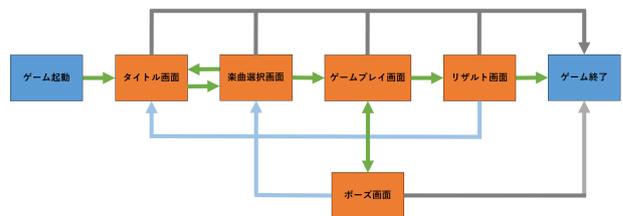


図7 ゲームシステムの流れ

ゲーム中のノーツは、画面上部から下部に向かって縦方向にスクロールし、画面下部に設置された判定ラインに到達する構造となっている。ノーツは6つのレー

のいずれかに割り当てられ、各レーンには対応する入力キー（Q, W, E, I, O, P）が設定されている。音楽の再生と同期してノーツが生成・落下するため、楽曲のリズムに合わせたプレイが可能である。

本システムは、以下の主要なモジュールから構成される。ゲーム速度の設定や楽曲選択を行うメニュー画面モジュール、オンセット検出および周波数解析を行う音楽解析モジュール、解析結果に基づいてノーツの種類およびレーン配置を決定する譜面生成モジュール、背景、ノーツ、判定ライン、スコア表示などを描画するゲーム描画モジュール、ならびに、ノーツが判定ライン付近に到達した際に対応する入力キーが押されたかを判定し、スコアを算出する当たり判定モジュールである。

本ゲームシステムは、音響信号から得られる音楽的特徴に基づいてノーツを生成することで、プレイヤーに演奏感を与えることを目的として設計されている。

4.2 ゲームプレイ方法

本節では、ノーツ自動生成システムを導入した音楽リズムゲームについて、その基本的なプレイ方法を説明する。

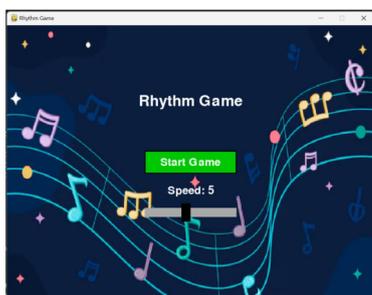


図8 タイトル画面

ゲームを起動すると、図8に示すメニュー画面が表示される。プレイヤーは、この画面においてノーツの落下速度をスライダーにより調整し、「Start」ボタンを押すことでゲームを開始する。

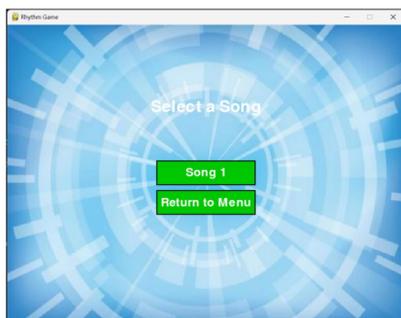


図9 楽曲選択画面

次に、図9に示す楽曲選択画面に遷移し、あらかじめ用意された楽曲の中から1曲を選択することでゲームが開始される。選択された楽曲に対して、ノーツ自動生成システムが適用され、譜面が生成される。



図10 ゲームプレイ画面

図10にゲームプレイ画面を示す。ゲーム中は選択した楽曲が再生され、その音楽と同期して画面上部からノーツが落下する。ノーツは6つのレーンに割り当てられており、各レーンはそれぞれキーボードのQ, W, E, I, O, Pキーに対応している。プレイヤーは、ノーツが判定エリアである画面下部の判定ライン付近に到達したタイミングで対応するキーを入力することでノーツを消去し、スコアを獲得する。

また、本ゲームでは3種類のノーツ画像を用意しており、楽曲中の周波数成分に基づいて異なる種類のノーツが生成される。これにより、音楽の特徴の変化が視覚的にも表現される構成となっている。

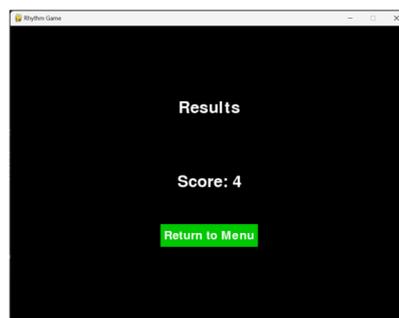


図11 リザルト画面

ゲーム終了後には、図11に示すリザルト画面が表示され、プレイ結果として総スコアが提示される。プレイヤーは、再プレイを行うか、楽曲選択画面へ戻ることが可能である。

なお、ゲームプレイ中に画面左上のポーズボタンを押すことで、図12に示すポーズ画面へ遷移し、一時的にゲームを中断することができる。

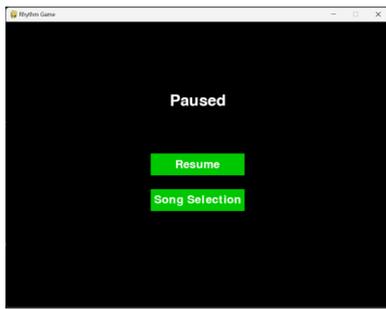


図 12 ポーズ画面

5. 音楽リズムゲームの評価実験

本章では、本研究で提案するノーツ自動生成システムの有効性を検証するために実施した評価実験について述べる。実際にプレイヤーによるゲームプレイを通じて評価を行い、自動生成されたノーツ配置が楽曲に対して適切であるか、また、音楽リズムゲームとしての面白さを有しているかを検証した。

評価はアンケート形式により実施し、ノーツ配置の妥当性、プレイ可能性、難易度、およびゲームとしての面白さといった観点から主観評価を行った。評価項目は以下のとおりである。

- ノーツ生成タイミングの適切さ
楽曲に対して、ノーツが適切なタイミングで生成されているかを評価する項目である。
- ゲームプレイの可否
プレイヤーにとって、物理的または認知的にプレイが困難となるノーツ配置やタイミングがどの程度発生しているかを評価する項目である。
- ゲームの難易度
実際のゲームプレイを通じて、プレイヤーが感じた難易度について評価する項目である。
- ゲームとしての面白さ
ノーツ自動生成システムを導入した音楽リズムゲームが、娯楽としてどの程度面白いと感じられたかを評価する項目である。

本評価実験は、音楽リズムゲームのプレイ経験の有無や経験年数が異なる高校生および大学生、計 21 名を対象として実施した。被験者には実際にゲームをプレイしてもらい、その後アンケートへの回答を求めた。

また、選択式の評価項目に加えて、自由記述式のアンケートも併せて実施し、定量評価では捉えきれない意見や改善点についても収集した。これらの評価結果および考察については、次節にて詳述する。

5.1 ビート検出を用いた実験結果

本節では、ビート検出に基づいて自動生成されたノーツ配置を用いた音楽リズムゲームの評価実験結果について述べる。

図 13 に、ビート検出を用いて生成されたノーツ配置の一例を示す。図において、横軸は時間、縦軸はレーン番号を表している。下段には入力楽曲の波形および検出されたビートのタイミングを示し、上段には各レーンに生成されたノーツの種類と配置を可視化している。この図より、楽曲のリズム構造および音量変化と、生成されたノーツ配置との対応関係を確認することができる。特に、ビート検出に基づくノーツ生成では、一定の周期性を持ったリズムに沿ってノーツが配置されていることが分かる。

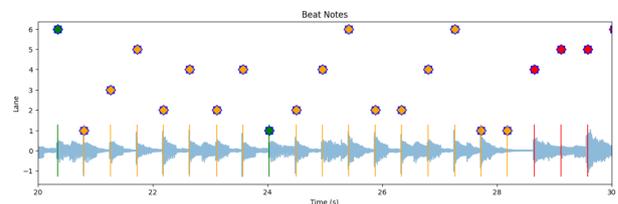


図 13 ビート検出によるノーツ生成位置とタイミング

次に、アンケートによる主観評価の結果を図 14 に示す。

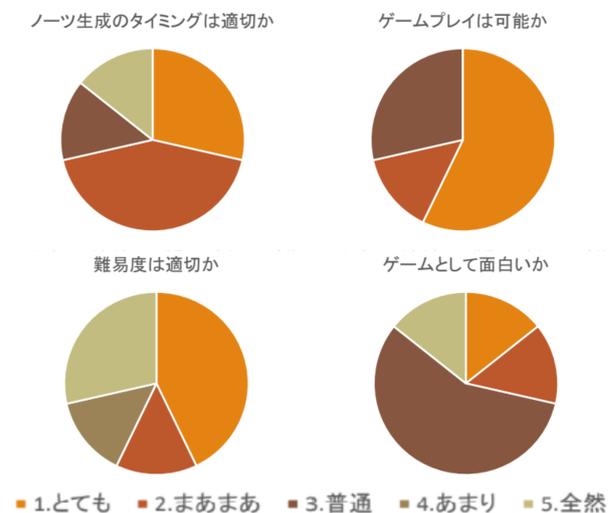


図 14 ビート検出によるノーツ自動生成の評価結果

「ノーツ生成タイミングの適切さ」に関する評価では、「概ね適切である」とする回答が多く、ビート検出に基づくノーツ生成が楽曲のリズムと一定程度整合していることが示唆された。

「ゲームプレイの可否」に関しては、全ての被験者から「プレイ可能である」との回答が得られ、極端に不適切なノーツ配置やタイミングは発生していないこ

とが確認された。

「ゲームの難易度」については、「適切である」とする評価が多かった一方で、一部の被験者からは「適切ではない」とする意見も得られた。このことから、ビート検出のみに基づくノーツ生成では、プレイヤーの熟練度によって難易度の感じ方にばらつきが生じる可能性があると考えられる。

「ゲームとしての面白さ」に関する評価では、中程度の評価が多く見られた。この結果から、ビート検出に基づくノーツ生成は、安定したプレイ体験を提供する一方で、単調さを感じさせる可能性があることが示唆された。

5.2 オンセット検出を用いた実験結果

本節では、オンセット検出に基づいて自動生成されたノーツ配置を用いた音楽リズムゲームの評価実験結果について述べる。

図 15 に、オンセット検出を用いて生成されたノーツ配置の一例を示す。図において、横軸は時間、縦軸はレーン番号を表している。下段には入力楽曲の波形および検出されたオンセットのタイミングを可視化し、上段には各レーンに生成されたノーツの種類と配置を示している。この図より、入力楽曲の波形変化に対応して、各オンセット時刻にノーツが生成されていることが確認できる。

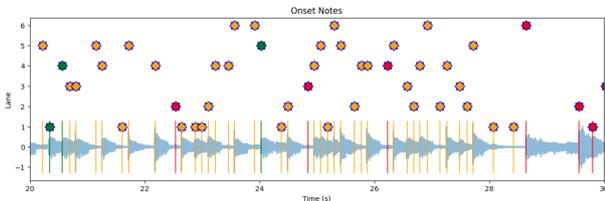


図 15 オンセット検出によるノーツ生成位置とタイミング

次に、オンセット検出によるノーツ自動生成に対するアンケート評価結果を、図 16 に示す。

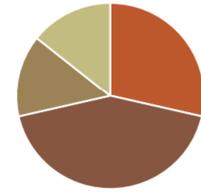
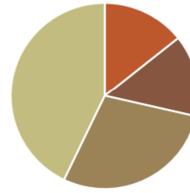
「ノーツ生成タイミングの適切さ」に関する評価では、「あまり適切ではない」とする回答が多く見られた。これは、オンセット検出によって生成されるノーツが、楽曲中の細かな音の立ち上がりに対して過剰に配置される傾向があるためと考えられる。

「ゲームプレイの可否」に関しては、一部の被験者を除き、「プレイが困難である」との回答が多数を占めた。この結果から、オンセット検出のみを用いたノーツ生成では、プレイヤーにとって処理しきれないノーツ密度やタイミングが発生しやすいことが示唆された。

「ゲームの難易度」については、「非常に難しい」と

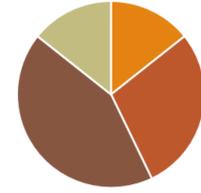
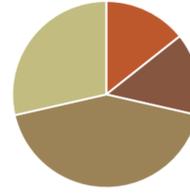
ノーツ生成のタイミングは適切か

ゲームプレイは可能か



難易度は適切か

ゲームとして面白いか



■ 1.とても ■ 2.まあまあ ■ 3.普通 ■ 4.あまり ■ 5.全然

図 16 オンセット検出によるノーツ自動生成の評価結果

する評価が全体的に多く寄せられた。これは、オンセット検出によって短時間に多数のノーツが生成されることが、操作負荷の増大につながったためであると考えられる。

一方で、「ゲームとしての面白さ」に関する評価では、ビート検出を用いた場合と比較して、やや高い評価が得られた。この結果から、操作難度は高いものの、音の変化に強く同期したノーツ配置が、演奏感や爽快感の向上に寄与している可能性が示唆された。

5.3 ビート・オンセット検出を用いた実験結果

本節では、ビート検出およびオンセット検出を組み合わせることでノーツを自動生成した場合の音楽リズムゲームにおける評価実験結果について述べる。

図 17 に、ビート・オンセット検出を併用して生成されたノーツ配置の一例を示す。図において、横軸は時間、縦軸はレーン番号を表している。下段には入力楽曲の波形と検出されたオンセットおよびビートのタイミングを可視化し、上段には各レーンに生成されたノーツの種類と配置を示している。この図より、オンセットとビートのタイミングが一致した箇所では、複数のノーツが同時に生成されており、同時押しノーツとして表現されていることが確認できる。

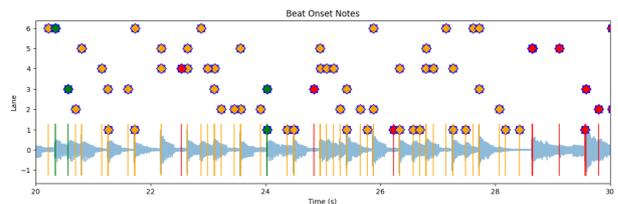


図 17 ビート・オンセット検出によるノーツ生成位置とタイミング

次に、ビート・オンセット検出を併用したノーツ自動生成に対するアンケート評価結果を、図 18 に示す。

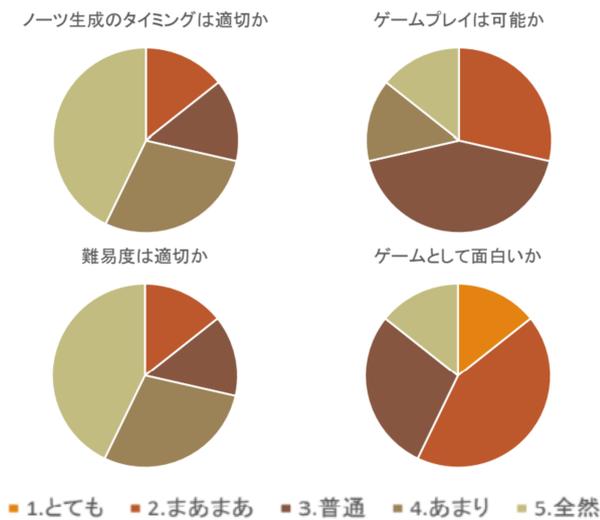


図 18 ビート・オンセット検出によるノーツ自動生成の評価結果

「ノーツ生成タイミングの適切さ」に関する評価では、オンセット検出のみを用いた場合と同様に、「あまり適切ではない」とする回答が多く見られた。これは、オンセット由来のノーツ生成が細かな音の立ち上がりを反映するため、ノーツ配置が過密になる傾向があることに起因すると考えられる。

「ゲームプレイの可否」に関しても、オンセット検出を用いた場合と同様に、「プレイが困難である」とする意見が多数を占めた。この結果から、ビート検出を併用した場合であっても、オンセット由来の高密度なノーツ配置が操作負荷の増大につながることを示唆された。

「ゲームの難易度」については、オンセット検出のみの場合と比較して、さらに難易度が高くなったとする評価が多く寄せられた。これは、オンセットとビートが重なった箇所でも同時押しノーツが生成されることにより、プレイヤーに要求される操作が複雑化したためであると考えられる。

一方で、「ゲームとしての面白さ」に関する評価では、ビート検出のみおよびオンセット検出のみの場合と比較して、最も高い評価が得られた。この結果から、操作難度の上昇と引き換えに、楽曲の構造や盛り上がりを強く反映したノーツ配置が、演奏感や達成感の向上に寄与している可能性が示唆された。

6. 考察

本研究では、ビート検出、オンセット検出、およびそれらを組み合わせた手法の3種類のノーツ自動生成手法について、音楽リズムゲームとしての有効性を評価した。以下では、各手法の特徴と評価結果について考察する。

まず、ビート検出を用いたノーツ生成では、ノーツの生成タイミングに関して「楽曲に対して適切である」とする評価が多く得られた。また、ほぼすべての評価者から「ゲームプレイは可能である」との意見が得られ、操作不能な配置はほとんど発生しなかったことが確認された。一方で、ゲームの難易度に関しては「簡単である」と「難しい」という意見が二極化する傾向が見られた。これは、ビートのみを基準として一定間隔でノーツが生成されるため、プレイヤーの経験値やリズム感によって体感難易度が大きく異なっていたことが要因であると考えられる。また、面白さの観点では、「ビートを刻むだけで単調に感じる」という意見が多く、楽曲の細かな構造変化を十分に反映できていない点が課題として挙げられる。

次に、オンセット検出を用いたノーツ生成では、「ノーツ生成タイミングが必ずしも適切ではない」とする評価が多く見られた。オンセット検出は音の立ち上りを細かく検出するため、ノーツ数が増加しやすく、結果として配置が高密度になる傾向がある。そのため、音楽リズムゲームの経験が少ないプレイヤーにとっては操作が困難であるとの意見が多く寄せられた。このことから、ゲームの難易度についても、ビート検出の場合より「難しい」と感じる評価が多くなったと考えられる。一方で、面白さの観点では、ビート検出の場合と比較して評価が高く、楽曲の抑揚や音の変化がノーツ配置に反映されることで、音楽リズムゲームらしさが向上したと考えられる。

ビート検出とオンセット検出を組み合わせた複合手法では、全体的な印象としてオンセット検出のみの場合と大きな差異を感じにくいという意見が多く得られた。しかし、ビートとオンセットのタイミングが一致した箇所でも同時押しノーツが生成される点については、難易度が大きく上昇する一方で、「より音楽リズムゲームらしい演奏感が得られる」といった肯定的な意見も見られた。その結果、3手法の中では最も高い「ゲームとしての面白さ」の評価を得ることができた。

また、ゲームの難易度に関しては、全体を通して「簡単だった」と「難しかった」という評価が二極化する傾向が確認された。これは、評価者ごとの音楽リズムゲーム経験の有無や熟練度の違いが、評価結果に大き

く影響しているためであると考えられる。実際に、音楽リズムゲーム未経験者にとっては評価自体が難しいという意見や、経験者であっても初見プレイでは適切な評価が困難であるという意見が複数寄せられた。

以上の結果から、ノーツ自動生成システムを評価するにあたっては、プレイヤーの熟練度に応じて難易度を調整可能な仕組みが重要であることが示唆された。今後は、ノーツ密度や同時押しの発生頻度を制御することにより、プレイヤーが自由に難易度を選択できるシステムの導入が有効であると考えられる。

7. まとめ

本研究では、音楽リズムゲームにおけるノーツ自動生成を目的として、楽曲解析手法であるビート検出およびオンセット検出を用いたノーツ生成システムを構築し、実際にプレイ可能なゲーム環境を通じて評価を行った。その結果、ビート検出およびオンセット検出を用いたノーツ生成は、楽曲のリズム構造や音の立ち上がりを反映したノーツ配置を実現できる点で有効であることが確認された。特に、ビート検出は安定したプレイ可能性を確保しやすく、オンセット検出は楽曲の抑揚を反映した演奏感の向上に寄与することが明らかとなった。

一方で、評価実験の結果から、プレイヤーの音楽リズムゲームに対する経験や熟練度によって、ノーツ生成手法に対する評価が大きく異なることが確認された。このことから、ノーツ自動生成システムの適切な評価および実用化を行うためには、プレイヤーが自由に難易度を選択できる仕組みや、ノーツ密度・同時押し頻度などを調整可能な機構を導入することが重要であると考えられる。

また、本研究で用いたビート検出およびオンセット検出といった一般的な楽曲解析手法のみでは、楽曲が持つ複雑な音響構造や作曲者の意図を十分に反映したノーツ配置を行うことは難しく、楽曲に対するノーツの適応度には依然として課題が残る。そのため、完全な自動生成のみで高品質な譜面を生成することには限界があり、人手による微調整を前提とした支援システムとしての位置づけが現実的であると考えられる。

今後の課題としては、ノーツ密度の動的制御や難易度推定手法の導入に加え、機械学習を用いた特徴量の自動抽出や、プレイヤーの操作ログを用いた適応的なノーツ生成手法の検討が挙げられる。これらを組み合わせることで、より多様なプレイヤーに対応可能なノーツ自動生成システムの実現が期待される。

参考文献

- [1] 株式会社バンダイナムコエンターテインメント. 太鼓の達人. <https://taiko.namco-ch.net/taiko/>.
- [2] 株式会社コナミアミューズメント. Dance Dance Revolution. <https://p.eagate.573.jp/game/ddr/ddra3/p/>.
- [3] 小倉久和. 技術系の数学-ベクトル解析・複素関数・フーリエ級数・ラプラス変換編-. 近代科学社, 2014.
- [4] 矢田部浩平. 第三回: 短時間フーリエ変換. 日本音響学会誌, Vol. 77, No. 6, pp. 396–403, 2021.
- [5] 小野順貴. 短時間フーリエ変換の基礎と応用. 日本音響学会誌, Vol. 72, No. 12, pp. 764–769, 2016.
- [6] Steven L.Horowitz. A syntactic algorithm for peak detection in waveforms with applications to cardiology. *Communications of the ACM* Volume, Vol. 18, No. 5, pp. 281–285, 1975.
- [7] 周長明, 高木幹雄. ピーク検出に基づいた最適なヒストグラム分割法とその閾値選択への応用. テレビジョン学会誌, Vol. 45, No. 10, pp. 1179–1189, 1991.
- [8] 沼田宗敏, 野村俊, 神谷和秀, 田代発造, 輿水大和. うねり曲線を抽出するローパスフィルタの研究—ダウンサンプリングとG-スプラインフィルタによるガウシアンフィルタの置き換え—. 精密工学会誌論文集, Vol. 71, No. 2, pp. 235–241, 2005.
- [9] 松尾章弘, 原佑輔, 土江田織枝, 山田昌尚. リズム練習支援のためのリアルタイム発音検出手法の検討. 情報科学技術フォーラム講演論文集, Vol. 14, No. 2, pp. 205–206, 2015.
- [10] J.P.Bello, L.Daudet, S.Abdallah, C.Duxbury, M.Davies, M.B.Sandler. A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing*, Vol. 13, No. 5, pp. 1035–1047, 2005.
- [11] Al-Khowarizmi and H.Maulana. The Utilization of Gaussian Filter Method on Voice Record Frequency Noise. *International Conference on Cybernetics and Intelligent System (ICORIS)*, pp. 1–5, 2020.

- [12] 伊藤彰則. 楽曲分割と高精度ビート推定を用いたビートトラッキング. 研究報告音楽情報科学 (MUS) , Vol. 128, No. 9, pp. 1-6, 2020.
- [13] 國枝伸行, 島村徹也, 鈴木誠史. 対数スペクトルの自己相関関数を利用したピッチ抽出法. 電子情報通信学会論文誌, Vol. J80-A, No. 3, pp. 435-443, 1997.