

2022年度 卒業論文

音楽によるレースゲームの
コース自動生成に関する研究

指導教員：渡辺 大地 教授

メディア学部 ゲームサイエンスプロジェクト

学籍番号 M0119077

加藤 思元

2023年2月

2022年度 卒業論文概要

論文題目

音楽によるレースゲームの
コース自動生成に関する研究

メディア学部

学籍番号：M0119077

氏名

加藤 思元

指導
教員

渡辺 大地 教授

キーワード

音楽、自動生成、レースゲーム、リズムゲーム、ノーツ、リズム

ゲームのジャンルとして様々なものが存在するが、その中にリズムゲームとレースゲームが存在する。リズムゲームとは、リズムや音楽に合わせてプレイヤーがアクションをとることで進行する形式のゲームである。リズムゲームには、どのときにアクションを起こすかのタイミングを示唆するために必要なノーツというものがある。ノーツが流れているものを譜面と呼ぶ。リズムゲームは誰にでも馴染みやすい「音楽」を主な題材として扱っており、遊び方が感覚的に解りやすく、同じ曲でも簡単なステージや難しいステージを作れるため、広い層に人気のゲームジャンルとなっている。レースゲームは、プレイヤーが車やバイクなどの乗り物を操作し、競争的に他のプレイヤーやNPCに対して走行するゲームのことである。リズムゲームとレースゲームを組み合わせ、音楽に基づいてレースゲームのコースを生成するゲームがある。このようなゲームは、プレイヤーが自分の好きな音楽とレースゲームを同時に楽しむことができる。代表的なゲームに Riff Racer、Music Racer がある。しかし、この2つの非常に人気のあるゲームジャンルを組み合わせで、このようなゲームを好きなプレイヤーはあまりいない。本研究では自動生成系レースゲームを開発し、被験者にプレイ後にアンケートに答えてもらうことで、自動生成系レースゲームのプレイヤー数が少ない理由を探る。本実験で使用したゲームはノーツ、コースの変化と音楽との関連性を感じ取ることがあまりできなかった。

目次

第1章	はじめに	1
1.1	背景と目的	1
1.2	論文構成	3
第2章	先行研究	4
2.1	コース変化について	4
2.2	ノーツ生成について	5
第3章	手法について	10
3.1	概要	10
3.2	基本ルール	11
3.3	ノーツ生成	12
3.4	コース変化	14
第4章	評価と分析	19
4.1	実験について	19
4.2	実験結果	19
	4.2.1 音楽とコース変化の関連について	20
	4.2.2 音楽とノーツ生成の関連について	22
	4.2.3 自由記述について	24
4.3	考察	25
第5章	まとめ	26
	謝辞	28
	参考文献	30

目次

2.1	編集インターフェース	7
2.2	編集画面	7
2.3	payload の種類	8
2.4	イベントの種類	9
3.1	提案手法	11
3.2	ノーツの生成イベント	13
3.3	ノーツ生成のタイミング	14
3.4	curved world のサンプルシーン	15
3.5	左への変化	16
3.6	上への変化	17
3.7	下への変化	17
3.8	右への変化	18
4.1	音楽とコース変化について	20
4.2	音楽とコース変化についての平均値	21
4.3	コースの変化頻度について	21
4.4	コースの変化頻度についての平均値	22
4.5	音楽とノーツ生成について	23
4.6	音楽とノーツ生成についての平均値	23
4.7	ノーツの流れてくる頻度について	24
4.8	ノーツの流れてくる頻度についての平均値	24

第 1 章

はじめに

1.1 背景と目的

ゲームのジャンルとして様々なものが存在するが、その中にリズムゲームとレースゲームが存在する。音楽ゲーム [1] とは、リズムや音楽に合わせてプレイヤーがアクションをとることで進行する形式のゲームである。リズムゲームには、どのときにアクションを起こすかのタイミングを示唆するために必要なノーツというものがある。ノーツが流れているものを譜面と呼ぶ。リズムゲームは「音楽」を主な題材とし、遊び方がわかりやすく、同じ曲でも簡単なステージや難しいステージを作ることができ、幅広い層に人気のゲームジャンルとなっている。レースゲームは、プレイヤーが車やバイクなどの乗り物を操作し、競争的に他のプレイヤーや NPC に対して走行するゲームのことである。一般的には、コースやトラックを走って、最短時間でゴールするのが主な遊び方である。多くのレースゲームには、さまざまな車種やカスタマイズオプションがあり、オンラインプレイや多人数対戦などの機能も備えていることが多い。リズムゲームとレースゲームを組み合わせ、音楽に基づいてレースゲームのコースを生成するゲームがある。このようなゲームは、プレイヤーが自分の好きな音楽とレースゲームを同時に楽しむことができる。代表的なゲームに Riff Racer、Music Racer がある。しかし、この 2 つの非常に人気のあるゲームジャンルを組み合わせて、ゲームの情報をまとめているサイト「steamdb」 [2] から、このジャンルのゲームをプレイする人は非常に少ないことが分かった。本研究では自動生成系レースゲームを開発し、被験者にプレイ後にアンケートに答えてもらうことで、自動生成系レースゲームのプレイヤー数が少ない理由を分析した。また、「自動生成系レースゲームの人気がない理由はコース、ノーツと

音楽の関連性が薄い」と仮説を立て、実験を行った。

音楽に基づいてゲームコンテンツを生成するゲームを開発するために、音楽ゲーム譜面を自動生成する研究がいくつか行われている。リズムゲームの譜面の自動生成については多くの先行研究があり、機械学習を用いた研究やキー音抽出による譜面の生成などが多く行われている。大淵ら [3] の研究には音楽の音のの情報から機械学習を行い、任意の楽曲から譜面を自動生成する手法を提案した。ノーツの自動生成には、香川ら [4] によるフレーズを抽出する頻出頻度を利用した重要音抽出法を提案し、MIDI 形式の楽曲に対して自動的に譜面データを生成する手法を提案した。都丸 [5] の研究では、MIDI 形式の楽曲データから音楽ゲームの譜面自動生成する手法を提案した。メロディのリズム、ドラムのリズム、ビート、ドラムの盛り上がりによって難易度を変更する。Liang ら [6] の研究では、ディープラーニングを主な手法として、音楽素材からコンテンツを自動的に生成することを目的として研究を行った。Donahue ら [7] の研究では、RNN と CNN アーキテクチャを利用し、リズムゲームの譜面を自動的に生成することが成功した。柴崎ら [8] の研究や福永ら [9] の研究は機械学習を主な手法としてリズムゲーム譜面の自動生成を行なった。どの研究でも事前に作成した譜面データを学習データとして必要としている。辻野ら [10] の研究ではダンスゲームデータの中にある難易度の特徴を取得して、グループ化することで分析を行った。斉藤 [11] の研究ではリズムゲームのノーツが流れていく向きに応じて実際に難易度の分析を行った。川口 [12] の研究では機械学習を使用せずに動画と音声データから譜面の自動生成を行った。

本研究では、ゲームエンジンである Unity[13] を利用して自作ゲームを作成した。Unity のアセットである Koreographer[14] を利用してノーツの生成場所と生成するタイミングを設定する。イベントの設定には専用のエディタを使用して、ビートに合わせてノーツの生成場所と生成するタイミングを指定する。そして、ノーツの生成場所に応じてコースが変化する。

本研究は、被験者にプレイ後にアンケートを行い、

1. 音楽とノーツの生成場所の関連性について

2. 音楽とコース変化の関連性について

3. 被験者の感想について

の三つの点に分けて検証を考察した。実験の結果として、「音楽とノーツの生成場所の関連性」を「どちらかというと感じない」と「やや感じる」と回答した人が一番多かった。「音楽とコース変化の関連性」を「少し感じる」と回答した人が一番多かった。ゲームに対する解説が不十分であるため、音楽とノーツとコースの変化との関連性を感じにくくなった。他にも、「リズムゲームとレースゲームの組み合わせの楽しさがわからない」という回答もある。プレイヤーは音楽とゲームコンテンツの関連性を感じることができるが、リズムゲームとレースゲームの組み合わせの楽しさを感じる事が難しいという結果となった。

実験結果から見ると、本実験で使用したゲームはノーツ、コースの変化と音楽との関連性を感じ取ることがあまりできなかった。「自動生成系レースゲームの人気のない理由はコース、ノーツと音楽の関連性が薄い」という仮説を検証するには不十分である。本実験が失敗した原因は、ビートをマークしてノーツを生成するには、ノーツが流れるのを待っている間に音楽とノーツの関連性がなくなってしまうからだと考える。

1.2 論文構成

本論文は、全5章にて構成する。構成は2章にて、先行研究について述べる。3章にて提案手法について述べる。また、4章にて評価と分析について述べる。そして5章にてまとめを述べる。

第 2 章

先行研究

2.1 コース変化について

市販されている自動生成型レースゲームには、様々なタイプがある。一部のゲームでは、プレイヤーが自分でコースをデザインすることができるものがある。このようなゲームでは、プレイヤーは車を操作しながら、自分でデザインしたコースを走ることができる。他のゲームでは、ゲーム内で生成されるランダムなコースをプレイすることができるものがある。このようなゲームでは、プレイヤーは車を操作しながら、ゲームが生成するランダムなコースを走ることができる。

自動生成型レースゲームとは、プレイヤーがプレイするたびにコースが自動的に生成されるレースゲームのことである。このようなゲームでは、コースの変化によってプレイヤーが毎回新しいチャレンジを迎えることができるため、プレイヤーがゲームを繰り返しプレイすることを促すことができる。コースの変化については、ゲームによって異なるが、一般的には次のような方法で行われる。

ランダム生成: コースを生成するときに、決められたルールのもとでランダムにコースを生成する。これにより、プレイヤーが毎回異なるコースをプレイすることができる。代表的なゲームは riff racer[15]、music racer[16] がある。

プレイヤーのパフォーマンスに応じて変化する: プレイヤーがよくプレイした場合、より難しいコースが生成されるようにしたり、逆にプレイヤーが苦手な場合はより簡単なコースが生成されるようにすることで、プレイヤーのスキルに応じたチャレンジを提供することができる。代表的なゲームは Forza Motorsport[17]、DiRT Rally 2.0[18] がある。

ステージごとに変化する: ゲームを進めていくにつれて、ステージごとに異なるコースが生成されるようにすることで、プレイヤーが毎回新しいチャレンジを迎えることができる。代表的なゲームは TrackMania[19]、Trials Fusion[20] がある。

2.2 ノーツ生成について

リズムゲームとは、音楽とともにタイミングを合わせて画面上のアイテムを操作するゲームのことである。リズムゲーム [21] には以下の要素がある。

1. ノーツ: 曲に合わせて画面上を流れるシンボル
2. 譜面: リズムゲームを遊ぶ際に画面に表示される『ノーツ』(アイコン) の順序やタイミングを曲に合わせて配置/設定したもの。
3. レーン: ノーツが流れてくるエリア

音楽には、ビートがある。ビートとは音楽における構成要素の一つである。ビートは音楽のパターンの基礎を成す、一定間隔で繰り返す音である。ビートを活用して、ゲーム内でノーツを生成することができる。具体的には、音楽に合わせてゲーム内のアニメーションを作成したり、受け入れ可能なビートを設定して、それに合わせてゲーム内のノーツを動的に生成することができる。このような仕組みを活用することで、プレイヤーはよりリアルな音楽ゲームを楽しむことができる。音楽のビート情報を分析し、その拍子に合わせて音符を生成する。これにより、音楽を聴きながらゲームをプレイし、音楽のリズム感を感じることができる。ゲーム内で音楽のリズムをマークして音符生成を行う具体的な方法は異なる場合があるが、一般的には以下のような手順が用いられる。

1. 音楽のビート情報を分析する: 音声処理ツールや専用のビート検出ソフトを使用して音楽のビート情報を分析する。

2. ビートに合わせてノーツを生成する: 分析により得られたビート情報をもとに、ビートに合わせてノーツを生成する。これはゲームエンジンやオーディオツールを使って行うことができる。
3. ゲームと音楽の連携: 音楽のビート情報を解析し、そのビートの瞬間に音符を生成する。これにより、プレイヤーは音楽を聴きながら、音楽のリズムを感じてゲームをプレイすることができる。

本研究では Koreography というアセットを使って音楽を分析した。Koreographer は、Unity で使用できる音楽ゲーム向けのアセットである。Koreography を使用することで、音楽のビートに同期したノーツ生成や、音楽に応じた演出やエフェクトなどを実装することができる。Koreographer は Unity 独自のアクティブオーディオプラグインで、ビデオゲームにおける音楽とゲームの同期プロセスを簡単にする。ユーザーは音楽のリズムやボリューム、その他の要素を使ってゲーム内のイベントを動かすことができる。Koreographer はまた、ゲーム内のイベントをゲーム音楽内の異なる点に正確に同期させることができる。図 2.1 と図 2.2 は Koreographer の編集インターフェースである。

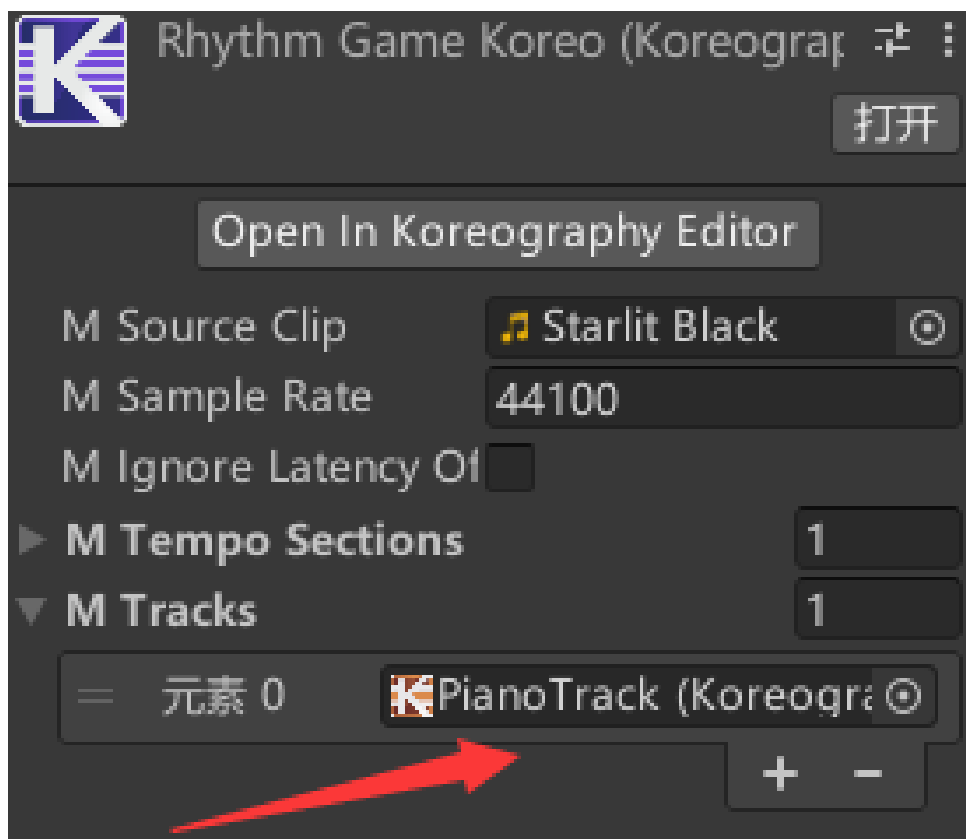


図 2.1 編集インターフェース

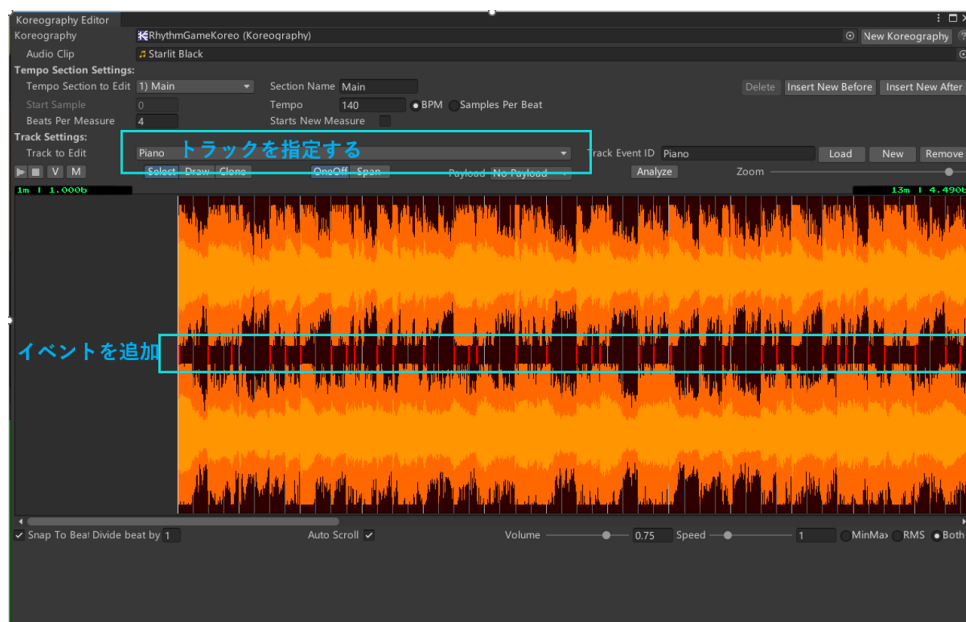


図 2.2 編集画面

イベント発行時に「Payload」と呼ばれるメタデータを指定してオブジェクト側へ渡すことがで

きる。図 2.2 の青い枠の中にある赤の線が payload である。指定できる「Payload」の種類は以下の 7 つがある [22]。図 2.3 は設定できる Payload の種類である。

1. Asset:asset を指定する
2. Int:int 値を指定する
3. Float: float 値を指定する
4. Curve: AnimationCurve を指定する
5. Color: Color を指定する
6. Gradient:ColorGradient を指定する

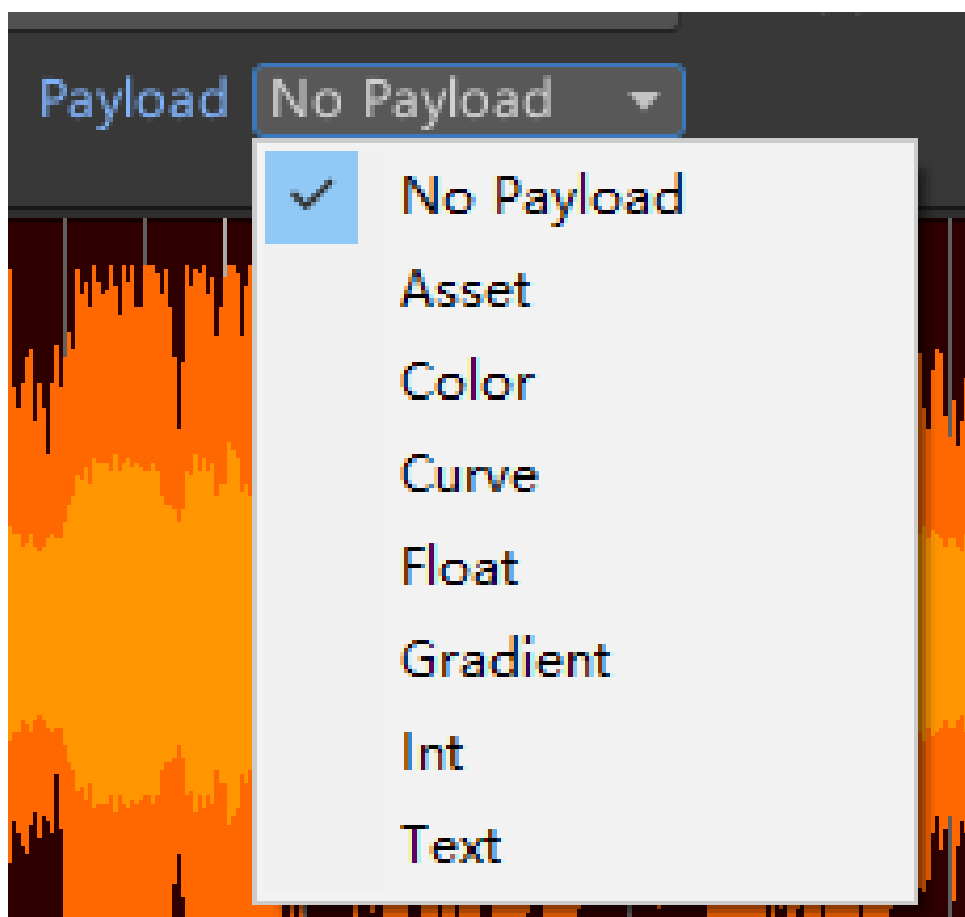


図 2.3 payload の種類

イベントの設定には専用のエディタを使用する。専用エディタで楽曲のファイルを指定するとその波形が画面に表示することができる。イベントの種類は以下の二つがある。図 2.4 は設定できるイベントの種類である。

1. OneOff イベント：開始タイミングのみ指定する-使用する Payload：int,float,Text,Color
2. Span イベント：イベントの開始と終了を指定する-使用する Payload：Curve,Gradient



図 2.4 イベントの種類

第 3 章

手法について

3.1 概要

本研究の手法では三つの部分に分けることができる。第一部分は音楽ファイルの分析、第二部分はノーツの自動生成とコースの変化、第三部分はレースゲームの制作である。本研究では Unity のアセットである Curved world を使ってレースゲームコースの変化を実現する。Koreographer を利用して、ビートに応じてノーツを生成する。図 3.1 は本研究の提案手法である。

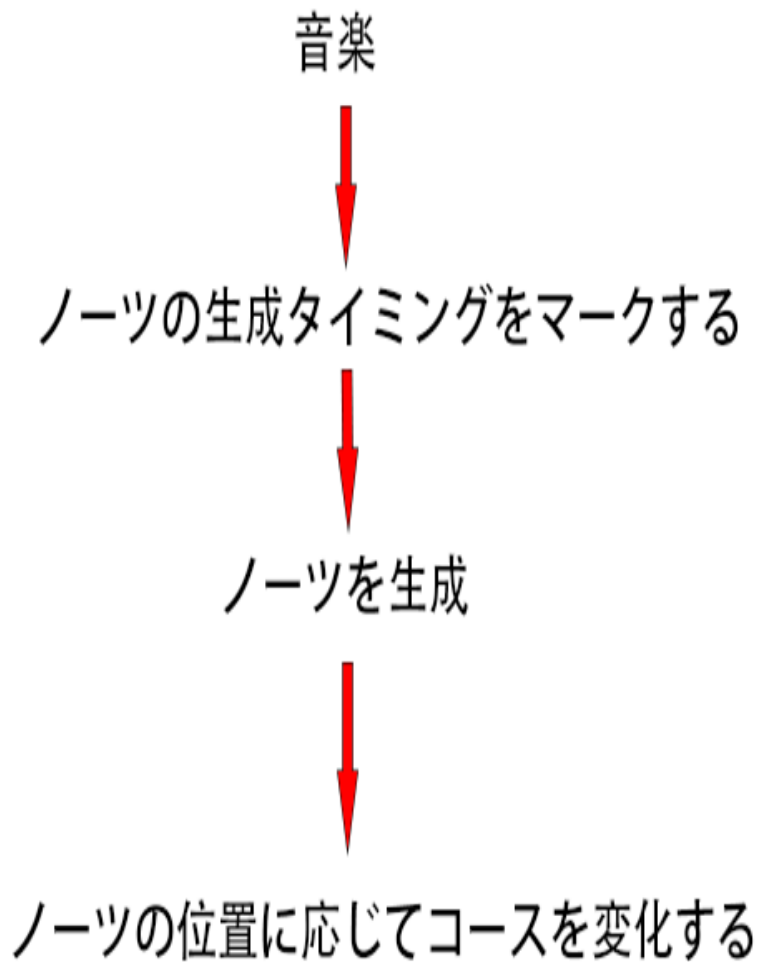


図 3.1 提案手法

3.2 基本ルール

車両の移動は A、D キーを使う。このゲームは 3 レーンで構成されたゲームで、キーボードの A、D キーで操作をするものである。プレイヤーはゲームの中でできるだけ多くノーツを集める。

3.3 ノーツ生成

本研究では、ビートに基づいてゲーム内のノーツを生成する。具体的には、音楽を聴きながら、Koreographerのエディタを利用して、ノーツが生成するタイミングをマークする、マークされた点に応じてノーツが生成する。

ビートに応じてノーツを生成するには、Koreographerのイベントシステムを使用する。Koreographerには、音楽のビートに同期して発生するイベントをリッスンするための「リスナー」と、ビートに同期して実行されるアクションを定義する「アクター」がある。

例えばあるビートに応じてノーツを生成したい場合、まず音楽のビートをマークする。ビートに同期して発生するイベントをリッスンするリスナーを作成する。リスナーがトリガーされたときに、ノーツを生成するアクターを作成する。リスナーとアクターを結びつける。このようにすることで、音楽のビートに同期してノーツが生成されるようになる。

音楽のビートをマークするには、KoreographerのTimeline Inspectorを使用する。Timeline Inspectorでは、音楽ファイルを読み込んで、ビートを示すマーカーをタイムライン上に配置することができる。図3.2でノーツ生成イベントの追加を示す。

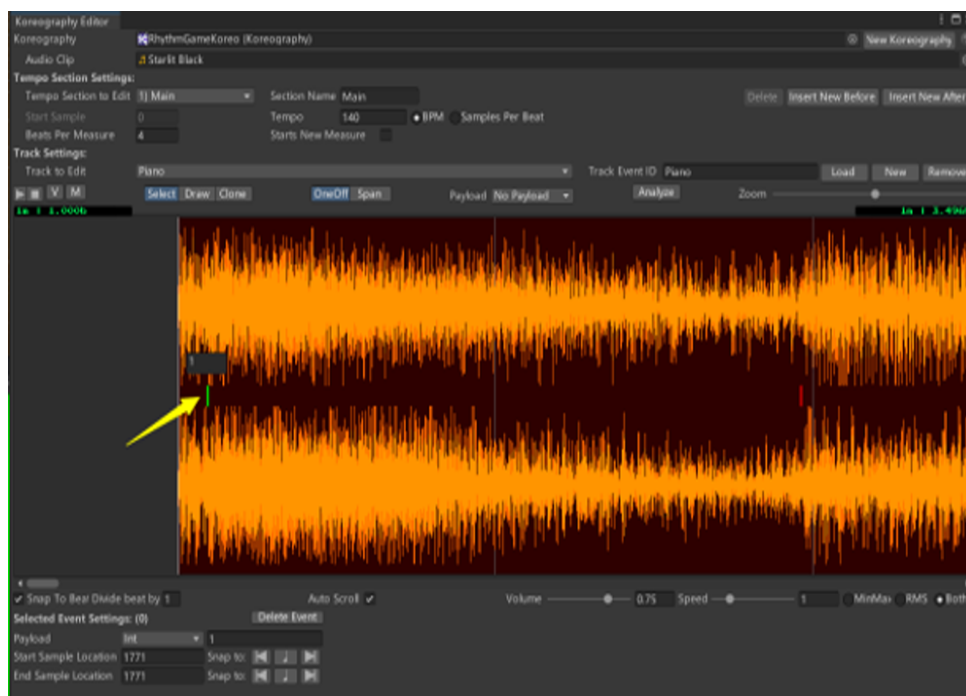


図 3.2 ノーツの生成イベント

ビートに応じてノーツを生成するには、Koreographer のイベントシステムを使用する。下の図の青い丸で囲まれたところはノーツが生成するタイミングである。図 3.3 でノーツが生成するタイミングを示す。

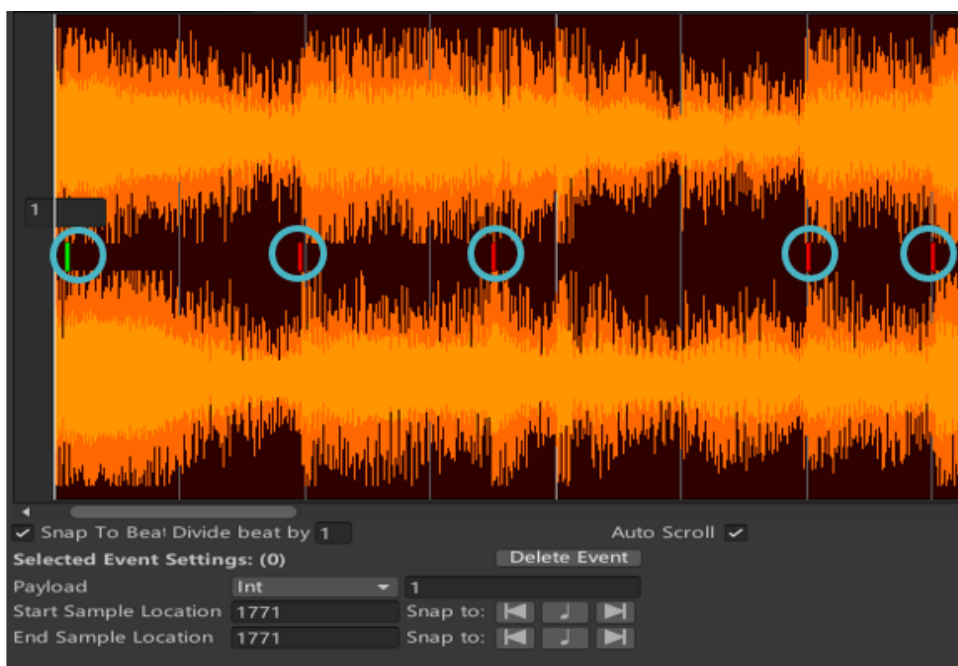


図 3.3 ノーツ生成のタイミング

また、楽曲中のノーツが発生する箇所に整数値を設定する。実行中に指定された場所に番号に応じてノーツを生成する。例えば、数字は 1 である場合は左のレーンでノーツを生成する、数字は 2 である場合は真ん中のレーンでノーツを生成する、数字は 3 である場合は右のレーンでノーツを生成する。

3.4 コース変化

本研究では Unity のアセットである Curved world を使ってレースゲームコースの変化を実現する。Curved World [23] はゆがんだ世界の表現を簡単に作ることができるアセットである。Horizontal(水平)と Vertical(垂直)のパラメータを調整することでコースを上下左右に変化することができる。図 3.4 は curved world にあるサンプルシーンである。



図 3.4 curved world のサンプルシーン

コースの変化についてはノーツが生成された位置に応じてコースが変化する。Curved world を使ってコースの曲がり具合を調整する。4 秒ごとにノーツの位置を確認し、その時点でノーツが生成された位置に応じてコースが変化する。例えば、生成しているノーツが一番左のレーンにあると、コースは左に曲がる。図 3.5 でコースが左に変化するゲームの実装画面を示す。

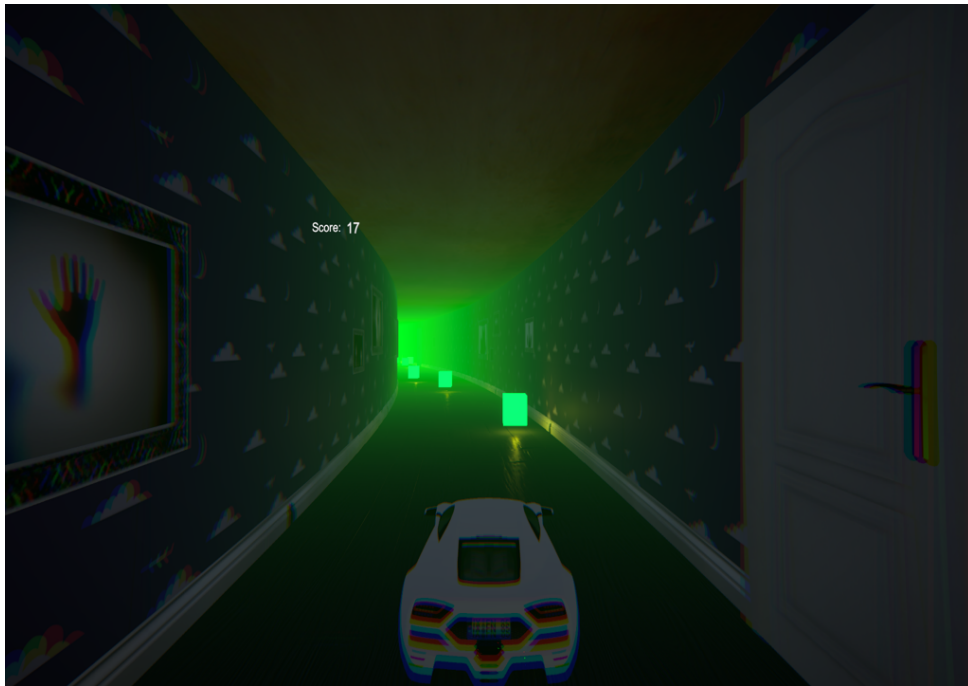


図 3.5 左への変化

真ん中のレーンにノーツが生成されると、1回目は上に曲がる、2回目は下に曲がる。図 3.2 でノーツ生成イベントの追加を示す。図 3.6 でコースが上に変化するゲームの実装画面を示す。図 3.7 でコースが下に変化するゲームの実装画面を示す。

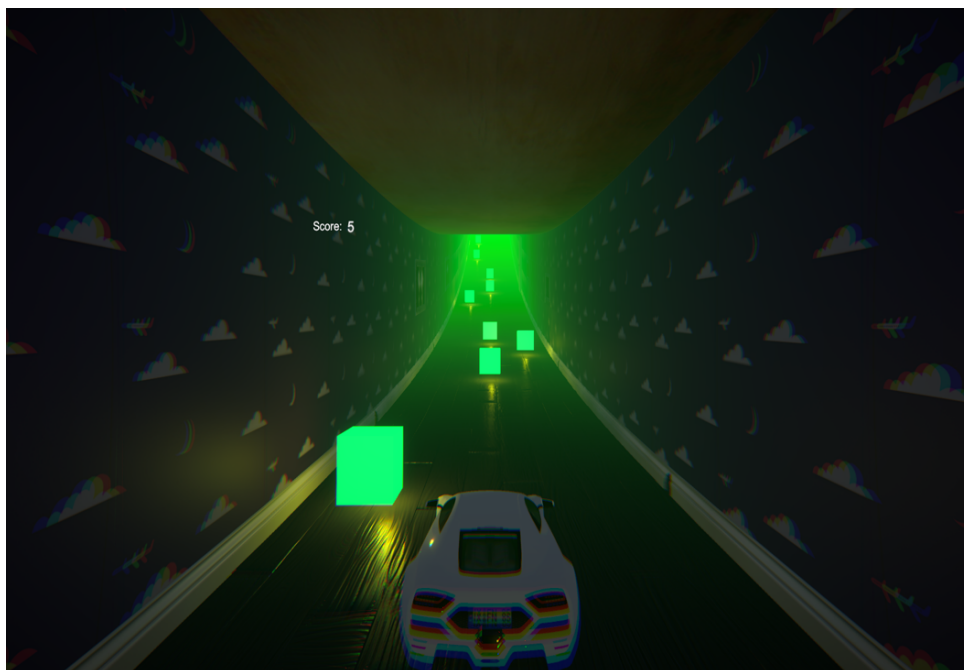


図 3.6 上に変化



図 3.7 下に変化

生成しているノーツが一番右のレーンである場合、コースは右に曲がる。図 3.2 でノーツ生成イベントの追加を示す。図 3.8 でコースが右に変化するゲームの実装画面を示す。



図 3.8 右への変化

第 4 章

評価と分析

4.1 実験について

本研究は、被験者に実験用のゲームをプレイしてもらい、音楽とノーツの関連性、音楽とコース変化の関連性を調べるアンケートに回答する。本研究では、11 人のアンケート調査結果を収録した。アンケートの内容について、実験後、音楽とゲームの変化との関連性についてアンケートを取った。アンケートは、「実験前に被験者が類似のゲームをプレイしたかどうか」、「コースの変化について音楽との関連性をどのくらい感じるか」、「ノーツの生成場所について音楽との関連性をどのくらい感じるか」、「ノーツの出現頻度についてどのくらい感じるか」、「コースの変化頻度についてどのくらい感じるか」という 5 項目の質問をしてもらい、音楽との関連性についての質問は「強く感じる」、「やや感じる」、「少し感じる」、「どちらかという感じない」、「あまり感じない」、「まったく感じない」の 6 段階評価で被験者に回答する。ノーツの出現頻度とコースの変化頻度について「多い」、「やや多い」、「やや少ない」、「少ない」の 4 段階評価で被験者に回答する。最後に被験者の感想、改善できそうだった点を調査する。

4.2 実験結果

実験結果については、11 人の被験者のうち、似たようなゲームをした人はいない。コースの変化と音楽の関連性を感じたことについては「少し感じる」という回答が最も多く、合わせて 3 人が「少し感じる」と回答した。「強く感じる」以外の選択肢はそれぞれ二人ずつ回答した。その平均値は 3.09 である。ノーツが生成される場所と音楽の関連性を感じるかについては「どちらかと

いうと感じない」と「やや感じる」が同数で、それぞれ四人が回答した。その平均値は 3.27 である。ノーツの出現頻度については「やや多い」が最も多く、8 人が「やや多い」と回答した。「やや少ない」と答えた人が 2 人、「多い」と答えた人が 1 人、その平均値は 2.91 である。コースの変化の頻度については「やや少ない」という回答が最も多く、計 8 人が「やや少ない」と回答した。その平均値は 2 である。

4.2.1 音楽とコース変化の関連について

音楽とコース変化の関連の実験結果から見ると、「少し感じる」と答えた人が三人で最も多く、「強く感じる」以外の選択肢はそれぞれ二人ずつ回答した。「少し感じる」と答えた人が最も多く、本実験で使用したゲームがコースの変化と音楽との関連性はあまり大きくないことが分かった。図 4.1 は音楽とコース変化の関連性についての調査結果である。図 4.2 は音楽とコース変化の関連性についての平均値である。

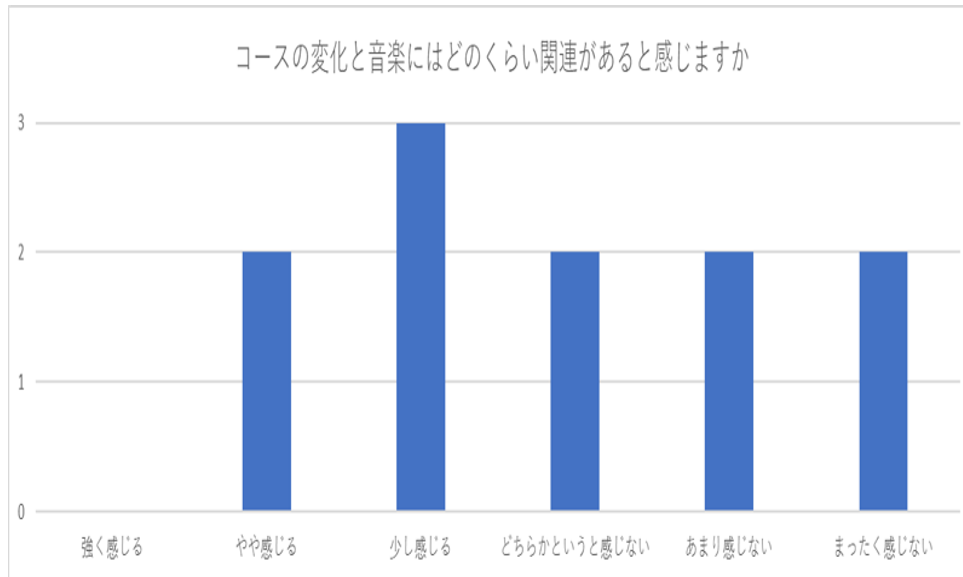


図 4.1 音楽とコース変化について

コースの変化と音楽の関連性について			
	人数	合計点	平均
強く感じる(6)	0	0	3.09
やや感じる(5)	2	10	
少し感じる(4)	3	12	
どちらかというと感じない(3)	2	6	
あまり感じない(2)	2	4	
まったく感じない(1)	2	2	

図 4.2 音楽とコース変化についての平均値

コースの変化頻度について、「やや少ない」と答えた人が9人で最も多く、それ以外は1人ずつであった。実験結果から見ると、音量の大きさによってコースの変化を行うと、音楽の種類によってコースの変化頻度が多すぎたり少なすぎたりすることがある。今回ゲームに使用した音楽は音量の変化があまりなく、コースの変化頻度が少なすぎになった。図 4.3 は音楽とコースの変化頻度についての調査結果である。図 4.4 はコースの変化頻度についての平均値である。

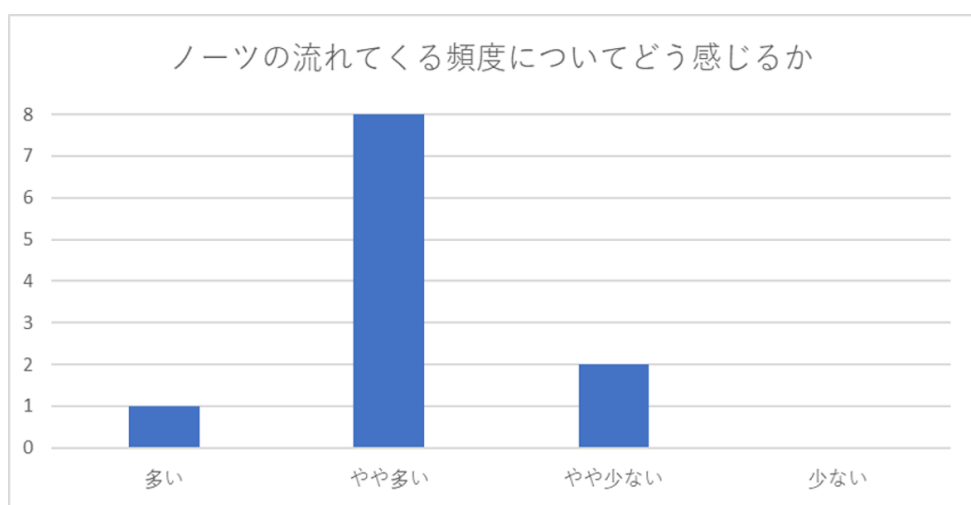


図 4.3 コースの変化頻度について

コースの変化の頻度について			
	人数	合計点	平均
多い(4)	0	0	2.00
やや多い(3)	1	3	
やや少ない(2)	9	18	
少ない(1)	1	1	

図 4.4 コースの変化頻度についての平均値

4.2.2 音楽とノーツ生成の関連について

ノーツの生成場所と音楽の関連の実験結果について、「どちらかというと感じない」と「やや感じる」の回答数が同数で、それぞれ四人が回答した。実験結果を見ると、「強く感じる」と答えた回答者はいなかった。音楽とノーツの生成位置の関連性は特に大きくないことが証明された。図 4.5 はノーツの生成場所と音楽の関連性についての調査結果である。図 4.6 はノーツの生成場所と音楽の関連性についての平均値である。

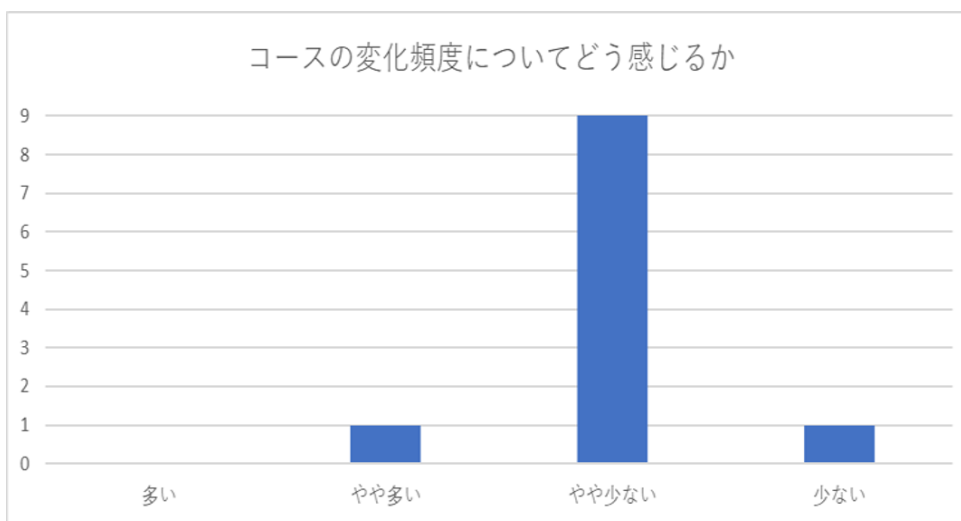


図 4.5 音楽とノーツ生成について

	人数	合計点	平均
強く感じる(6)	0	0	3.27
やや感じる(5)	1	5	
少し感じる(4)	4	16	
どちらかというと感じない(3)	4	12	
あまり感じない(2)	1	2	
まったく感じない(1)	1	1	

図 4.6 音楽とノーツ生成についての平均値

ノーツの流れてくる頻度について、「やや多い」と答えた人が8人で最も多く、「やや少ない」と答えた人が2人、「多い」と答えた人が1人、ノーツの密度を減らす必要があることが分かった。図 4.7 はノーツの流れてくる頻度についての調査結果である。図 4.8 はノーツの流れてくる頻度についての平均値である。

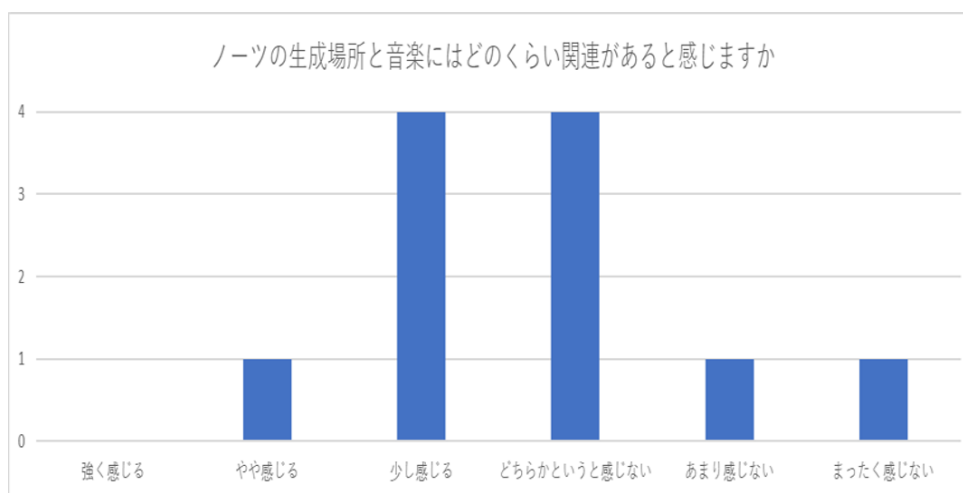


図 4.7 ノーツの流れてくる頻度について

ノーツの出現頻度について

	人数	合計点	平均
多い(4)	1	4	2.91
やや多い(3)	8	24	
やや少ない(2)	2	4	
少ない(1)	0	0	

図 4.8 ノーツの流れてくる頻度についての平均値

4.2.3 自由記述について

二人はゲームの「音量が大きすぎる」と回答した、二人は「ゲームのルールと操作方法の解説が欲しい」と回答した。四人は「ノーツと音楽の関連性があまり感じられない」と回答した。三人は「コースの変化と音楽の関連性があまり感じられない」と答えた。

4.3 考察

ノーツの生成、コースの変化と音楽の関連性についての平均値から見るとノーツ、コースの変化と音楽との関連性はあまり大きくないことが分かりました。コースの変化の頻度についての平均値から見ると、コースの変化が少ないと感じるのは、ノーツが同じレーンで連続して生成される可能性があるため、変化の頻度が少なくなった。ノーツの出現頻度についての平均値から見ると、ノーツの密度を減らす必要があることが分かった。

まず実験結果から、本実験で使用したゲームはノーツ、コースの変化と音楽との関連性はあまり大きくないことが分かった。二人の被験者がゲームの音量が大きすぎるとゲームのルールがわからないという意見をもらった。プレイヤーが自由に音量を調整できるインタフェースを追加し、ゲームの操作方法やゲームルールの説明を追加した方が良いと考える。ノーツの密度を減らし、音楽とノーツの生成とコースの変化のアルゴリズムを改善することで、被験者はそれらの間の関連性を容易に感じたと考えられる。

第 5 章

まとめ

本研究では、「自動生成系レースゲームの人気のない理由はコース、ノーツと音楽の関連性が薄い」という仮説を立て、自動生成系レースゲームのプレイヤー数が少ない理由を探るために自動生成系レースゲームの開発を行った。被験者に実験用のゲームを遊んでもらうことで音楽とゲームの関連性に関するアンケートを回答してもらった。実験結果から見ると、本実験で使ったゲームはノーツ、コースの変化と音楽との関連性を感じ取ることがあまりできなかった。「自動生成系レースゲームの人気のない理由はコース、ノーツと音楽の関連性が薄い」という仮説を検証するには不十分である。本実験が失敗した原因は、ビートをマークしてノーツを生成するには、ノーツが流れるのを待っている間に音楽とノーツの関連性がなくなってしまうからだと考える。アンケート調査の結果から見ると、十一人の中で似たようなゲームをプレイしたことある人はいなかった。音楽とレースゲームの組み合わせがどのような楽しさがあるのか、自動生成系音楽レーシングゲームというジャンルのゲームは比較的マイナーで、その楽しさを理解できない人が多いことを分かった。

今後の展望として、まずユーザーがゲーム中に自由に音量を調節できないという問題を解決し、ユーザーがゲーム中に好きな音量を出すことができるようにする。また、ゲームを始める前に、ゲームのルールを理解して、ゲームの操作をより楽しめるようにするための解説が必要である。また、ゲームのルールを追加して、プレイヤーがゲームの中で自分の行動範囲をより理解し、ゲームのルールに違反しないようにすることが望ましい。最後に、ユーザーが自分の好きな音楽を追加できるようにすることが望ましい。また、音楽のリズムの特徴に合わせて自動的にトラックが生成され、音楽のリズムに合わせてノーツを集める。これによって、プレイヤーはゲームをより

自由に楽しむことができ、より良いゲーム体験が得られる。

謝辞

本研究を進めるにあたり、多くのアドバイスや指導をしてくださり、相談に乗ってくれた渡辺先生、阿部先生には大変お世話になりました。ご指導いただいた渡辺先生、阿部先生には、心から感謝申し上げます。

また、手伝ってくださった友人方や先輩方に心より感謝いたします。先輩方からゲーム開発に関する多くの技術を教えていただきました。新しい知識を学ぶことは自分にとって非常に意味のある時間でした。アンケートの回答者数が足りないときは、友人たちが貴重な時間を割いてアンケートに協力してくれました。本当に感謝しています。

最終審査を聞いていただき、アドバイスや質問をしてくださった永田先生にも大変お世話になりました。貴重なご指導とご助言を賜りました。感謝申し上げます。渡辺先生、阿部先生には、研究の着想から、調査、論文執筆まで多くのご指導をいただきました。渡辺先生と阿部先生から有益なアドバイスを多くいただき、最終審査の発表に大きく役立ちました。心から感謝申し上げます。

学部3年から学部4年までの間、就職や研究について多くのことを学びました。特に学部4年で卒業論文に着手する時に渡辺先生の指導によって、論文は大きな進歩を遂げました。渡辺先生は論文の書き方も教えてくれて、論文の修正や添削もしてくれました。渡辺先生のおかげで、卒業論文を順調に完成することができました。本当にありがとうございました。

自分が困難な時期を乗り越えられたのは友達や家族のおかげです。研究の過程で、自分では解決できない困難がたくさんありましたが、友達や家族の助けで最後まで頑張りました。自分を支えてくれたすべての人たちのおかげです。感謝の気持ちはもう言葉では表せません。

最後に改めて、多くのアドバイスやお手伝いをしてくださった渡辺先生、阿部先生、永田先生、私を助けてくれた友達や先輩たちに心からの感謝の意を表します。辛い時期を支えてくれた両親に深く感謝いたします。

本当にありがとうございました。

参考文献

- [1] 音楽ゲーム. <https://ja.wikipedia.org/wiki/%E9%9F%B3%E6%A5%BD%E3%82%B2%E3%83%BC%E3%83%A0>. 参照: 2023.1.15.
- [2] Pavel Djundik. ゲームの発売日とオンライン人数. <https://steamdb.info/tag/8253/>. 参照: 2023.1.15.
- [3] 大輝福永, 景子越智, 康成大淵. キー音を用いたリズムアクションゲームにおける譜面の自動生成. 東京工科大学大学院バイオ・情報メディア研究科メディアサイエンス専攻, 東京工科大学メディア学部, 東京工科大学メディア学部, No. 10, 2020.
- [4] 香川俊宗, 手塚宏史, 稲葉真理ほか. 音楽の重要な構成要素の抽出の提案-音楽ゲーム用譜面自動生成のために. エンタテインメントコンピューティングシンポジウム 2015 論文集, Vol. 2015, pp. 326–333, 2015.
- [5] 都丸英紀. 楽曲からのリズム取得の違いによる音楽ゲーム譜面の難易度表現の研究. 学部卒業論文, 東京工科大学メディア学部ゲームサイエンスプロジェクト, 2017.
- [6] Yubin Liang, 心池田. リズムゲームの上達を支援するコンテンツ自動生成法. 情報処理学会研究報告. GI, 研究報告ゲーム情報学, Vol. 2018-GI-39, No. 11, pp. 1–7, 02 2018.
- [7] Chris Donahue, Zachary C. Lipton, and Julian McAuley. Dance dance convolution, 2017.
- [8] 大地柴崎, 充酒井, 博丸山. ニューラルネットワークを用いた音楽ゲーム譜面自動生成の検討. 第 80 回全国大会講演論文集, Vol. 2018, No. 1, pp. 165–166, 03 2018.
- [9] 福永大輝, 大淵康成. リズムアクションゲームにおけるキー音の自動推定. 画像電子学会研究会講演予稿, Vol. 17.04, pp. 172–175, 2018.
- [10] 辻野雄大 山西良典 山下洋一井本桂右. ダンスゲーム譜面の特性分析とクラスタリングに基づく特徴的な譜面の自動生成. エンタテインメントコンピューティングシンポジウム, 09 2019.

- [11] 齊藤浩輝. リズムゲームにおける譜面ノーツ動作パターンによる難易度分析に関する研究. 学部卒業論文, 東京工科大学メディア学部ゲームサイエンスプロジェクト, 2022.
- [12] 川口雄大. リズムゲームにおけるノーツ数を考慮した譜面の自動生成に関する研究. 学部卒業論文, 東京工科大学メディア学部ゲームサイエンスプロジェクト, 2020.
- [13] Unity Technologies. Unity. <https://unity.com/ja>. 参照: 2023.1.15.
- [14] Sonic Bloom. Koreographer. <https://assetstore.unity.com/packages/tools/audio/koreographer-54639?locale=ja-JP>. 参照: 2023.1.15.
- [15] FOAM Entertainment. Riff racer. https://store.steampowered.com/app/351990/Riff_Racer__Race_Your_Music/. 参照: 2023.1.15.
- [16] AbstractArt. Music racer. https://store.steampowered.com/app/893030/Music_Racer/. 参照: 2023.1.15.
- [17] Turn 10 Studios. Forza motorsport. https://ja.wikipedia.org/wiki/Forza_Motorsport. 参照: 2023.1.15.
- [18] Codemasters. Dirt rally 2.0. https://en.wikipedia.org/wiki/Dirt_Rally. 参照: 2023.1.15.
- [19] NADEO. Trackmania. <https://ja.wikipedia.org/wiki/TrackMania>. 参照: 2023.1.15.
- [20] ユービーアイソフト. トライアルズフュージョン. <https://www.ubisoft.co.jp/trialsfusion/>. 参照: 2023.1.15.
- [21] アイテルハック. 【音ゲー用語集】音ゲー用語を徹底解説!! <https://eiter-hexe.com/music-game-glossary>. 参照: 2023.1.15.
- [22] Karvan. 音ゲー製作には必須? koreographer を使ってみたよ. <https://www.karvan1230.com/entry/2018/08/28/221434>. 参照: 2023.1.15.
- [23] Amazing Assets. Curved world. <https://assetstore.unity.com/packages/vfx/>

[shaders/curved-world-173251?locale=ja-JP](https://www.shaders.com/curved-world-173251?locale=ja-JP). 参照: 2023.1.15.