公立はこだて未来大学
2025年度
システム情報科学実習
グループ報告書
Future University Hakodate 2025 System Information Science Practice
Group Report

プロジェクト名
クリエイティブAI
Project Name
Creative AI

グループ名
公立はこだて未来大学
2025年度
システム情報科学実習
グループ報告書
Future University Hakodate 2025 System Information Science Practice
Group Report

プロジェクト名
クリエイティブAI
Project Name
Creative AI

グループ名
音響班
Group Name
Audio Group

プロジェクト番号/Project No.
4

グループリーダー/Group Leader
1023184 原耀良 Akira Hara
グループメンバー/Group Member
1023042 大瀧智元 Tomoharu OTaki
1023179 野本藍里 Airi Nomoto

指導教員
村井源 中田隆行 吉田博則

Advisor
Hajime Murai　Takayuki Nakata　Hironori Yoshida
提出日
2025年1月21日
Date of Submission
Generally.21，2025

概要

音響班は本プロジェクトで，プレイヤーのゲームの世界への没入感を高めることを目的とし，そのためにゲーム内で使用するBGM，効果音を準備することを目標とした．また，AIを用いた活動の主要な手法として，ゲーム内での特定のフィールドのBGMについて，LSTMモデルで8小節のメロディを生成させ，それを参考に作曲を行った．

成果物として，森、洞窟の2つのフィールドを作成した．制作にあたっては、3種類の印象の数値を入力すると、数値に対応してAIが生成したメロディパターンを参考にした．BGM・効果音また，全てのBGM・効果音に対してゲーム用の音量調整，フェードアウト処理などを行い，最終的に全てのゲーム内で使用するBGM，効果音の準備が完了した．音源の準備後は，AIが生成したメロディパターンに対して，音高の頻度や変化パターンの頻度についてNgram分析を行った．その結果，AIは音高をあまり学習していないが，音高の変化パターンについては特定のメロディパターンについては学習できていることがわかった．また，この活動では学習データとして20曲を用いたが，BGM作曲の際には森，洞窟の出力結果からそれぞれ36%，53%の割合のノードを参考にすることができた．

（※文責：原耀良）

Abstract

The sound team's goal in this project was to increase the player's sense of immersion in the game world, and to achieve this, we aimed to prepare the background music and sound effects to be used in the game. Furthermore, as a primary method of using AI, we used an LSTM model to generate an eight-bar melody for the background music of specific fields in the game, and used this as a reference when composing the music. As a result, we generated by the AI were used as a reference when composing the background music for the two fields, the forest and the cave. Furthermore, we adjusted the volume and fade-out processing for all background music and sound effects for the game, and ultimately completed the preparation of all background music and sound effects to be used in the game.

(＊Written by: Akira Hara)

目次

第１章　はじめに

1.1　プロジェクトの概要

本プロジェクトでは，人間が有する創造性をサポートする人工知能を実現するという目標を達成するために，RPGゲームを制作した．今年度は「システム」「シナリオ」「視覚」「音響」の４つの班に分かれて活動した．活動の流れとして，前期に各班が目標としたシステム実現のために，データの収集や分析を行った．続いて後期では，実際にゲームに使用するシナリオ，ゲームシステム，画像，音源を準備し統合することで，ゲームの完成を目指した．

（※文責：原耀良）

1.2　音響班の役割

音響班は，今回のプロジェクトにおいて，ゲーム内で使用する音源の準備を行った．主な活動内容として，AIを活用したBGMの作曲，フリー音源を利用したゲーム内BGM・効果音の準備，AIが生成したメロディパターンについて，学習データに含まれる音高の使用頻度，変化パターンについて比較，分析を行った．

（※文責：原耀良）

1.3　AIを活用した活動

本活動では，AIを活用したBGM作曲への取り組みとして，フィールドBGMの作曲を目的とした．当初はすべてのフィールドに対してAIによる作曲を行うことを想定していたが，時間的制約のため，対象を「森」および「海辺」の2フィールドに限定した．AIを用いた作曲手法としては，各フィールドの印象を3種類に分ける．学習時には学習データとなる音源について，1曲ずつ3種類の印象をそれぞれ5件法をもちいて数値化した．その後数値化した印象と音源の主旋律の音高を関連づけ学習させた．出力時には，ゲーム内で出てくるフィールドに合致する印象を数値として入力し，数値に応じてAIがメロディパターンを生成するシステムを採用した．

（※文責：原耀良）

第２章　関連研究，使用する技術や手法

2.1　音楽と感情の関係性

2.1.1　音楽の印象に対する認知

音楽とそれに対して生じる感情の関係性を調べた．以下に論文から該当する箇所を抜粋し，記述する．

- 感情を高揚，親和，軽さ，抑鬱，強さ，荘重の6つの因子のグループ（各グループに6~10の感情）に分け，曲ごとに感じた表現をチェックさせた実験では，曲ごとに音楽的性格を(時に複数)持ち，それが時間ごとに変化することが分かった（Hevner，1936）．
- 楽器を用いた演奏において，演奏者が意図的に感情を意識して演奏した実験では，「無表情」や「悲しみ」は伝わりやすかったが，それに対し「喜び」，「怒り」，「荘重さ」は伝わりづらかった．また，「悲しみ」と「喜び」は混同されやすかった．理由としては，テンポが遅いこと，音高の変動が比較的小さいこと，音量自体が小さいことが考えられる（Gabrielsson & Juslin，1996）．
- 印象に色を適応させた「配色イメージスケール」を用意し，楽曲に対して抱いた印象をマッピングすることで1つの楽曲が，二つの尺度が示す印象を同時に喚起するのかを調べた実験では，「親和」と「強さ」は強い負の相関，「軽さ」と「高揚」では強い正の相関といった，ある程度の指向性のようなものがあった（川野邊・亀田，2009）．

（※文責：大瀧智元）

2.1.2　音楽感情認識のための音楽の特徴

視覚的印象からメロディを生成するAIの実装を目指すにあたり，音楽がどのように人の感情に影響を与えるかについて調査を行った．特に，Panda, Malheiro, & Paiva（2023）を参考に，音楽の各要素と感情との関連性を整理した．音楽は一般に四つから八つの次元で構成され，それぞれの次元が相互に作用しながら感情に影響を与えるとされている（Meyer，1973; Owen，2000）．この論文では，音楽的特徴と人の感情との関係を「ポジティブ／ネガティブ」および「覚醒度（高／低）」という二軸で分類して説明している．

- メロディは，連続する音高（ピッチ）によって構成される．ピッチが高いほど驚きや怒り，恐怖などの高覚醒感情と関係し，ピッチが低いほど悲しみや優しさ，退屈などの低覚醒感情と関係する．また，ピッチ変化が大きいほど活動的・幸せと感じられ，小さい変化は怒りや退屈，嫌悪感を与える（Juslin & Laukka, 2004; Gabrielsson & Lindström, 2011）．
- 複数の音を同時に鳴らす和音構成（ハーモニー）は，感情の質に強く影響する．メジャーコードや協和音はポジティブな感情と関係し，マイナーコードや不協和音はネガ

ティブな感情，特に不安や悲しみ，緊張を生じさせる（Juslin & Laukka, 2004; Gabrielsson & Lindström, 2011）．
- テンポが速いほど覚醒度が高まり，幸せや興奮，恐怖といった感情が引き出されやすい．一方，テンポが遅い音楽は平和や夢のような印象を与える．リズムの規則性も影響し，規則的なリズムは落ち着きや平和を，不規則なリズムは不安や興奮を引き起こす（Juslin & Laukka，2004; Gabrielsson & Lindström，2011）．
- 音の強弱の変化（ダイナミクス）も感情に影響を与える．大きな音や急激な変化は緊張や恐怖，怒りといった高覚醒な感情と関連し，静かな音や小さな変化は穏やかさや悲しみなど低覚醒な感情と関係する（Juslin & Laukka，2004; Gabrielsson & Lindström，2011）．
- 音色は楽器の種類や音の立ち上がり／減衰に関連する．振幅エンベロープが丸みを帯びた音（例：ピアノ，ギター）はポジティブに，シャープな音（例：ドラム，木琴）はネガティブに感じられる（Juslin & Laukka，2004; Gabrielsson & Lindström，2011）．また，スペクトル重心が高い音（明るい音色）はポジティブ，低い音（暗い音色）はネガティブとされる（Wu，Horner，& Lee，2014a，2014b）．
- スタッカート（Juslin & Laukka，2004; Gabrielsson & Lindström，2011）やビブラート（Juslin & Laukka，2004; Dromey，Holmes，Hopkin，& Tanner，2015）といった演奏技法も，聴き手の感情に影響する．具体的な演奏スタイルにより，同じ旋律でも異なる印象を与えることができる．
- 音楽のテクスチャー，つまり音の重なり方は聴き手の感情に影響を与える．モノフォニー（単音）はポジティブな感情や幸福感を喚起しやすいことが示されている（Kastner & Crowder，1990; Webster & Weir，2005）．一方，ホモフォニー（主旋律と伴奏の組み合わせ）は，モノフォニーよりもさらに幸福感を強く引き出すとされる（Gregory，Worrall，& Sarge，1996; McCulloch，1999）．
- 音楽の構造も感情に関係する．繰り返しの多いシンプルな構成はリラックスや安心感を与え，変化に富んだ複雑な構成は緊張感や攻撃性など，よりダイナミックな感情を喚起する可能性がある（Balkwill & Thompson，1999; Imberty，1979）．

以上のように，音楽は多様な構成要素を通じて感情に影響を与えることがわかっている．この知見は，AIによるメロディ生成において，どのような音楽的特徴がどのような印象を与えるかを考慮する上で重要な基盤となる．

（※文責：原耀良）

2.1.3　音楽的構造の感情表現に及ぼす影響

JuslinとSloboda （2001） による『音楽と感情の心理学』第3節では，感情を捉える代表的な方法として，カテゴリー的アプローチ，次元的アプローチが紹介されている．

カテゴリー的アプローチは、怒り・悲しみ・喜びといった生得的で普遍的な「基本情動」を出発点とし，そこからさまざまな感情状態が派生すると考える立場である，このアプローチでは、各情動が互いにどのように異なっているのかを明確に理解することができる．

一方，次元的アプローチは少数の次元上，例えば「快ー不快」「覚醒度」「力強さ」に基づき情動を同定する．なかでもラッセルの円環モデルは情動反応の本質的な部分をうまくとらえ

ている．ラッセルの円環モデルの特徴としては，感情を「感情的評価 (快ー不快)」と「生理的反応 (覚醒度)」の二次元を用いており，音楽作品の中で生じる情動表現の連続的な変化をとらえることができる．

第5節では，音楽の構造の中の様々な要素が聴き手によって知覚される感情の表現について実験的な研究を概観している．音楽がどのように感情的に知覚されるかを把握するため，「自由な現象記述」，「研究者から与えられた記述用語，形容詞，あるいは名詞の中からの選択」，「記述用語が問題の音楽にどの程度よく当てはまるのかの評定」の三つの手法が用いられている．例えば，Huber（1937）は音楽経験のある聴取者に短いピッチパターンを聴かせ，知覚された表現を自由に記述させた． この実験では，気分の印象，人間の特性の印象，感情的に色のついたアナウンス，動きの印象，様々な内的イメージについての記述が見られた．

また，Hevner（1935）による研究では，多数の感情表現語を円形の布置「感情表現語円環」を用い，聴取者は各楽曲に対して適切だと考えた用語にできるだけ多く印をする形式で実験が行われた． 実験には調性音楽の短い曲を選び，長調の曲は短調でも演奏し，旋律の方向，和音，リズム，テンポ，音域などの要素を操作し，原曲を変化させた． その結果，判断に最大の影響を及ぼす要素はテンポ，調，音域，和声，リズムであり，旋律の方向は影響が非常に小さいということが明らかになったことが知られている． また，「悲しい」「重苦しい」感情は短調，低音域，ゆっくりなテンポであるという特徴が見られ，「嬉しい」「明るい」感情は長調，速いテンポ，単純な和音であるという特徴が見られた． なお，このような実験の結果は，使用する楽曲の選定や文脈に大きく依存するため，相対的・文脈的に解釈する必要があるとされている．

さらに，Watson（1942）の研究では，音楽の専門家に対し，複数の楽曲のそれぞれに対してふさわしい形容詞に印をつけてもらうという実験を行った． 評価は音の高さ (低ー高)，音の大きさ (小ー大)，テンポ (遅ー速)，音色 (美ー汚)，ダイナミクス (急激な大きさの変化なしーあり)，リズム (規則的ー不規則的) を五段階尺度上で評定した． その結果，高いピッチと速いテンポ

は嬉しさと興奮，低いピッチと遅いテンポは悲しみ，音が大きいと興奮，ダイナミックレンジ(音の大きさの幅)が狭いと威厳，悲しみ，平穏などの印象が生まれやすいことが知られている．

ThompsonとRobitaille（1992）の研究では，作曲家に対して「喜び」「悲しみ」「興奮」「退屈」「怒り」「平穏」といった感情を表現する短い単旋律を作らせ，それを中程度に音楽経験のある聴取者が上述の感情尺度のうえで評定した．その結果，意図されていた表現を知覚した．例えば，喜びの旋律ははっきりと調整であり，リズムの変化が大きい．悲しい旋律は短調的または半音階的な和声でテンポが遅く，興奮の旋律は速く音程的な跳躍と高いピッチを持つ．退屈な旋律は音程が一音ずつ変化するような調性的な旋律である．怒りの旋律はリズム的に複雑で半音階的な和声または無調の旋律である．平穏な旋律は調性的で遅く，しばしば一音ずつ変化するような動きと旋律の跳躍が用いられていた，などといったことが知られている．

（※文責：野本藍里）

2.2　作曲

2.2.1　人工知能による作曲

人工知能とそのモデル，作曲に関連する著書（Cope，2019）から考えられることを記述する．

- ランダム性

  いわゆる「ランダム」と言われるコンピュータを用いた乱数生成の手法は，疑似ランダム性である．ランダム性とは，予想可能とするには複雑すぎること，あるいはパターンの欠如，あるいは無関係すぎる振る舞いのことである．創造性というものは一見予想不可能に見えるが，後から考えると合理的な道筋を歩んでいる．その点において創造性はランダム性とは全くの別物である．ランダム性を用いて音楽を生成しても，それに創造性があるとは言えない．

- アルゴリズム

  そもそも人間は臓器の動きや思考においてアルゴリズムの塊である．作曲家も作品に効果を生み出すためにアルゴリズムを使用する．アルゴリズムを用いると想像力が落ちるといったことはなく，想像力ないし創造性がアルゴリズムと競合することはないと考え

られる．なお，遺伝的アルゴリズムを用いて音楽を生成した場合，音高や音域に多様性が見られた．

- プログラミングとマルコフ連鎖

  ルールに基づくプログラミングとは，”直前の音”といった条件に対するif-then-else節の実行をするプログラミングのことである．このような条件的なふるまいはマルコフ連鎖で表現できる．なお，直前の音とさらにその前の音を条件とした2次マルコフ連鎖を用いて旋律を生成するとさらに複雑で面白いものになる．作曲への統計的なアプローチはルールに基づくマルコフ連鎖に似ているが，このルールに基づくプログラミングは疑似ランダムプロセスに由来している．つまり，前述したように創造性に乏しく，結局はプログラマに依存してしまう．

- ニューラルネットワーク

  入力・出力・隠れノードの影響で極端に複雑になっている人工知能モデルは．作曲の面においては，おおよそ同じ方式の曲を何度か入力し，同じ一般的なルールを持つ曲を入力することで学習を進める手法がある．しかし，出力された曲で音楽性が高いと評価されたものはほとんど無い．また，ニューラルネットワークに関連してファジー論理を挙げる．これは曖昧さを含む判断を人が潜在的な「真」「偽」にしたがってランク付けし，最適な選択を行う理論であり，2つ以上の機能をもつ和音に対する機能決定などの複数の解釈がある音楽データにも有用である．MIDIファイルにも用いることができるが，ファジー論理で作曲するプログラムは現状ほとんど存在しない．

（※文責：大瀧智元）

2.2.2　深層学習による自動作曲入門

AIによるメロディ生成を行うにあたり，深層学習を用いた自動作曲の基本的な仕組みについて文献調査を行った．Shin（2024）による著書の内容から，音楽のデータ表現，モデルの選択，および生成手法に関する要点を以下にまとめる．

音楽は，以下の2種類の形式で表現される．

- 記号的音楽：楽譜のように，音の高さ・長さなどを記号で離散的に表現したもの．自然言語処理における文章と類似し，音符の並びを系列データとして扱える．MIDIデータはこの形式に該当し，音楽生成AIの学習データとして広く用いられる．

- 具体的音楽：実際に演奏された音を音響信号として扱うもので，時間・周波数・強さの三軸で構成される．MelスペクトログラムやMFCCなどが使われ，主に音色分析やジャンル分類に応用される．

本プロジェクトでは主に記号的音楽を扱い，MIDI形式から音高系列を抽出し，AIに学習させる．音楽は，音が時間的に連続して並ぶデータであるため，時系列モデルとの相性が非常に良い．そのため，音楽生成の分野では，以下のような深層学習モデルが主に用いられている．

まず，RNN（リカレント・ニューラル・ネットワーク）は，最も基本的な時系列モデルであり，直前の出力を再利用して次の出力を予測する構造を持つ．構造がシンプルで扱いやすい反面，長期的な依存関係の学習には弱く，長いメロディの生成には不向きである．そのため，LSTM（Long Short-Term Memory）のような改良版がよく使われる．LSTMは，情報を長期間保持しやすい構造になっており，メロディ全体の流れを反映した長めのフレーズの生成に適している．同様に，GRU（Gated Recurrent Unit）もLSTMと同様に時系列データに強いが，構造がよりシンプルで軽量であるため，リアルタイムでの音楽生成などに適している．LSTMモデルと比べると，長期間の情報の保持という点で劣る．

より高度なモデルとしては，Transformerが挙げられる．これは自然言語処理の分野で注目を集めているモデルで，自己注意機構（Self-Attention）を活用することで，長期的な依存関係を効率的に処理できる．並列処理にも対応しており，多声部の作曲や大規模な音楽構造の生成に利用されることが多い．Huang et al.（2018）は，Transformer を用いることで長期的な音楽構造を保持した音楽生成が可能であることを示した．

一方，CNN（畳み込みニューラルネットワーク）は，本来時系列データの処理には向いていない．しかし，音声や音楽を画像のように扱う手法（例：スペクトログラム）において，局所的特徴の抽出に有用である．

また，GAN（敵対的生成ネットワーク）やVAE（変分オートエンコーダ）のような生成モデルも活用されており，これらは音楽のデータ分布そのものを学習し，新しい音楽を創り出すことができる．特に，より多様性のある生成結果が求められる場合に有効である．

さらに，音楽の「美しさ」や「調和」といった曖昧な要素を評価するために，強化学習の活用も注目されている．これは，AIが出力するメロディに対して「調和している」「印象に合っている」といった報酬を与え，それを最大化するように学習させる手法である．

本プロジェクトでは，音楽の時間的構造と印象との関係を学習することを目的としており，まずは時系列情報を扱うのに適したLSTMを使用して，視覚印象に基づく音高系列の生成を行った．

（※文責：原耀良）

2.2.3　作曲の基礎知識

Du Boisn（2019）を用いて，音楽および作曲の基礎知識を学んだ．

まず，音楽の基礎として「テンポ」「拍子」「音程」について理解を深めた．テンポは知覚される拍の速さ（頻度）を指し，BPM（beats per minute）はそれを表す単位である．テンポを決める決めることで，誰が演奏しても曲の長さが一定になる．拍子とは，強拍と弱拍の繰り返しによって形成されるリズムのパターンのことである．代表的な拍子には二分の二拍子や四分の四拍子などがあり，例えば二拍子は行進曲，三拍子はワルツ，四拍子はロックなど，それぞれの拍子によって曲の印象が変わる．また，音程とは二つの音の高さの隔たりを指し，一オクターブは八度である．音程には「完全系」と「長短系」の二種類が存在する．

次に，メロディー作りに関連して，「長調」「短調」「和音」について学んだ．長調は明るい雰囲気を持ち，「全音→全音→半音→全音→全音→全音→半音」という音程の並びを持つ．半音とは西洋音楽において最も小さい音程であり，全音とは半音２つ分である．長音階またはメジャースケールとも呼ばれる．一方，短調は暗い雰囲気を感じさせる音階で，「全音→半音→全音→全音→半音→全音→全音」という並びであり，短音階あるいはマイナースケールと呼ばれる．長調と短調を使い分けることで，曲の明るさや暗さの印象を変えることができる．また，メロディーは各小節の拍に合わせて，音の長さを合計しながら作る必要があることも学んだ．

さらに，和音については「和音の展開」と「主要和音」を中心に学習した．和音の展開とは，同じ構成音であっても音の並び方（配置）によって響きが変化するというものである．たとえば，「ミ・ソ・ド」と「ソ・ド・ミ」ではベースとなる音が異なるため，印象も異なる．主要和音とは，調ごとに特に重要な三つの和音を指す．各調には「主音，上主音，中主音，下属音，属音，下中音，導音（または主音）」の七つの音があり（※図を添付），その中でも「主音」「下属音」「属音」を基にした「主和音（トニック）」「下属和音（サブドミナント）」「属和音（ドミナント）」が主要和音とされる．主和音は安定感があり，曲の終止感を生む．

属和音は緊張感を持ち，調の特徴を強調する響きを持つ．下属和音は，主和音と属和音をつなぐ役割を果たす．

これらの知識は，後期に行うBGMやSE（効果音）の作成において実践的に活用した．

（※文責：野本藍里）

2.3　開発環境

AI並びにプログラムを実装するための開発環境として，使用言語はPython，Spyderを用いている．理由として，Pythonには多様なライブラリが存在しており，データ処理や機械学習などのタスクに適しているからである．また，SpyderはPython向けの統合開発環境であり，デバックモードでプログラムの処理の挙動がわかりやすい．そのうえ，変数エクスプローラ画面によって，プログラム実行中の変数の変化が容易に見れるため，学習，開発のためのツールとして適している点も挙げられる．

（※文責：原耀良）

2.4　人工知能の使用

人工知能でメロディを生成するにあたり，使用するモデルの候補を多層パーセプトロン（MLP）を用いたモデルとLSTMモデルに絞った．

- 多層パーセプトロン（MLP）

  英名Multilayer perceptronの略．線形分離可能ではないデータを識別できる非線形的に活性化されるノードの3つ以上の層からなり，ディープニューラルネットワークを構成できる．フィードフォワード型のニューラルネットワークであり，時間的な依存関係や履歴情報を扱う機構を持たない．

- LSTM

  長・短期記憶のことで，英名Long short-term memoryの略．RNNの一種であり，忘却ゲートなどの仕組みを用いることでRNNの課題である情報の長期的保存を可能としており，MLPや従来のRNN（回帰型ニューラルネットワーク）の課題であった時系列データに基づく処理に適している．

今回は時系列が重要なメロディ構築のため，LSTMを採用した．

（※文責：大瀧智元）

2.5　関連科目

本プロジェクトでは，人工知能続論の内容であるRNNモデルやLSTMモデルといったニューラルネットワークモデルの基礎知識を元に，AIの活用方法を検討した．

（※文責：原耀良）

2.6　Ngram分析

2.6.1　Ngram分析の概要

Ngram分析とは，系列データにおいて連続する n 個の要素から構成される部分列である n-gramについて，その出現頻度や確率，傾向を調べる分析手法である．言語処理以外にも音声処理，音楽情報処理，生物情報学などの分野で利用される．（山田，2004）．また，Nが１の場合はunigram，２の場合はbigram，３の場合はtrigram，4の場合はquadgramと呼ばれる．

本プロジェクトでは，unigram，trigram，quadgramを用いてAIに学習させたメロディおよびAIが生成したメロディについて，任意のN個の音の組み合わせが使われている頻度を分析した．

2.6.2 本プロジェクトでのNgram分析方法

本プロジェクトでは，Rを用いてNgram分析を行った．AIに学習させたメロディおよびAIが生成したメロディのノート番号を分析対象とした．

また，Ngramによって得た結果を分かりやすくするための前処理として，ノート番号をアルファベット表記（C4，D5など）に変換した．変換にはpythonを用いた．

（※文責：野本藍里）

第３章目的を達成するまでの手段，手法

3.1　AIを用いたメロディパターンの作成

3.1.1　印象の決定

「森」「洞窟」というテーマに対してどのような印象が適切であるかについて，音響班のメンバーで話し合い，議論を行った．その際，視覚班から提示された各フィールドのイメージ画像を参考にし，画像から受ける印象をもとにいくつかのキーワードを考案した上で，それぞれの場所にもっともふさわしいと判断される印象語を三つずつ選定した．

「洞窟」については「重い」「狭い」「緊張感のある」という印象が共通して得られたため，これを基本方針とした．一方，「森」については，「鬱蒼とした」「不穏な」「ざわつきのある」といった印象が強く，これらがもっともその雰囲気をよく表していると考えられた．これらの印象語は今後のBGM制作やメロディー分析において，分類や評価の基準として活用した．

（※文責：野本藍里）

3.1.2　MIDIデータの前処理とデータ収集

学習に使用するデータは，著作権フリーの曲をYouTubeなどから探し，そこから抽出したいメロディーを耳コピで再現し，DAWソフト「Domino」に打ち込み，MIDIデータとして出力したものを用いた．対象とする楽曲は「森」「洞窟」に対しそれぞれデータは二十個ずつ，合計で四十個のデータを収集した．なお，一つの曲から複数のメロディを抽出する場合もある．各データは八小節分のメロディーを基本単位とし，この八小節を二回繰り返すことでデータとしての長さを確保した．

耳コピに用いる楽曲は，学習の精度を高めるため，基本的に四拍子のものを優先的に選定した．ただし，「森」をイメージした楽曲の中に三拍子のものが多く見られる場合には，例外的に三拍子のデータも使用した．

学習に適したMIDIデータとするために，前処理として音高以外の要素をすべて統一した．この処理では，音の開始タイミングを揃え，音の長さはすべて八分音符に統一した．また，強さ

（ベロシティ）や拍子の情報も一定に揃えることで，不要なばらつきを排除し，音の高さ（ピッチ）はC4を基準にしてデータ全体の音域を明確に記録した．

（※文責：野本藍里）

3.1.3　モデルの学習

pythonでLSTMを用いてメロディを学習させる．その流れを以下に記述する．

- 一つのマップに対して3つ印象が存在し，8小節のメロディに対して感じる印象の要素の評定値（不気味な：5，陰鬱な：3，静かな：2　等）とそのメロディをLSTMに入力して学習させる．
- 大規模言語モデル（chatGPT等を予定）にマップを入力し，3つの印象に対する要素の評定値を出力させる．
- 大規模言語モデルに出力させた印象の要素の数値をLSTMに入力し，8小節のメロディを複数出力させる．
- 出力させた複数のメロディを繋げ曲の主旋律として使用し，目的の印象に合わせた曲を作曲する．

（※文責：大瀧智元）

3.2　BGM/SE収集

ゲーム内で使用するBGM9曲および効果音61音は，インターネット上で配布されているフリー音源を使用した．BGMの音源は主にYouTube上で公開されているものを，効果音の音源は主にPixabayなどの音源配布サイトから入手し，それぞれの利用規約を確認したうえで，クレジット表記の条件を遵守して利用した．

また，ゲーム内での使用に適するよう，音量調整やフィードバック処理などの編集を行った．音源の選定にあたっては，ゲーム全体の世界観や各フィールドの印象との整合性を重視し，シナリオ班と協議しながら決定した．

（※文責：原耀良）

第４章　AIの出力したメロディ

4.1　AIの出力したメロディとメロディから作った楽曲

前処理を終えたMIDIデータをpythonでscvデータに起こし，LSTMに学習させた．その後出力させたいメロディの印象を入力し，メロディのMIDIデータを5つ生成させた．出力させたMIDIデータのうち作曲の参考にできそうだと感じた部分を抜き出して作曲し，それぞれ実際に参考にした部分の平均割合を算出した．

（※文責：大瀧智元）

4.1.1　森

ゲーム内における森マップの戦闘背景に対する印象を言語モデルChat GPTに出力させると，その割合は【鬱蒼と：3，不穏な：1，ざわつき：1】であった．その印象の数値をLSTMに入力し，メロディを生成した．生成したメロディのうち，作曲の参考にできると考えたのは【階段状に音程が上昇する】部分と【下降したあとに上昇する】形であり，参考にできたノードの割合は全体の約36%であった．

（※文責：大瀧智元）

4.1.2　洞窟

ゲーム内における洞窟マップの戦闘背景に対する印象を言語モデルChat GPTに出力させると，その割合は【重い：3，狭い：3，緊張感のある：2】であった．その印象の数値をLSTMに入力し，メロディを生成した．生成したメロディのうち，作曲の参考にできると考えたのは【階段状に音程が上昇する】部分と【下降したあとに上昇する】部分，加えて【上昇したあとに下降する】形であり，参考にできたノードの割合は全体の約53%であった．

（※文責：大瀧智元）

4.2　Ngram分析の結果

AIに学習させたメロディおよびAIが生成したメロディをunigramで分析することで，使われている音の頻度を明らかにした．また，AIに学習させたメロディをすべてDマイナースケールに転

調していることから，AIが生成したメロディがDマイナースケールに含まれる音を含んでいるかどうか調べることで，AIの精度を明らかにした．

また，AIに学習させたメロディおよびAIが生成したメロディをtrigramで分析することで，音高の上昇と下降のパターンを調べた．パターンの種類を図１に示した．パターンは上昇のみ，下降のみ，上昇した後に下降する，下降した後に上昇するの4パターンに分類し，それぞれのパターンの使用頻度を調べた．

（※文責：野本藍里）

4.2.1　unigram分析の結果

森のBGMについて，unigramの結果を図２に示した．AIに学習させたメロディはDマイナースケールに含まれる音が84.3%，Dマイナースケールに含まれない音が5.6%，休符が10.1%であった．AIが生成したメロディはDマイナースケールに含まれる音が46.4%，Dマイナースケールに含まれない音が32.7%，休符が20.9%であった．

図２　森のBGMのunigram分析の結果

洞窟のBGMについて，unigramの結果を図３に示した．AIに学習させたメロディはDマイナースケールに含まれる音が88.3%，Dマイナースケールに含まれない音が3.4%，休符が8.3%であった．AIが生成したメロディはDマイナースケールに含まれる音が42%，Dマイナースケールに含まれない音が31.9%，休符が26.1%であった．

（※文責：野本藍里）

図３　洞窟のBGMのunigram分析の結果

4.2.2　tirgram分析の結果

tirgramの結果について，表１に結果をまとめた．また，休符を含むパターンは除いていることから，それぞれのパターンの数値の合計が1にならない．

森のBGMについて，AIに学習させたメロディは，上昇のみのパターンが0.134，下降のみのパターンが0.194，上昇した後に下降するパターンが0.213，下降した後に上昇するパターンが0.162であった．AIが生成したメロディは，上昇のみのパターンが0.016，下降のみのパターンが，0.066，上昇した後に下降するパターンが0.016，下降した後に上昇するパターンが0.097であった．

洞窟のBGMについて，AIに学習させたメロディは，上昇のみのパターンが0.187，下降のみのパターンが，0.195，上昇した後に下降するパターンが0.193，下降した後に上昇するパターンが0.18であった．AIが生成したメロディは，上昇のみのパターンが0.056，下降のみのパターンが0.006，上昇した後に下降するパターンが0.082，下降した後に上昇するパターンが0.082であった．

（※文責：野本藍里）

表1　メロディのパターンの頻度

	森	
	学習させたメロディ	出力させたメロディ
	頻度	頻度
上昇	0.134	0.016
下降	0.194	0.066
上昇下降	0.213	0.016
下降上昇	0.162	0.097

	洞窟	
	学習させたメロディ	出力させたメロディ
	頻度	頻度
上昇	0.187	0.056
下降	0.195	0.006
上昇下降	0.193	0.082
下降上昇	0.18	0.082

図４　森のBGMのtrigram分析の結果

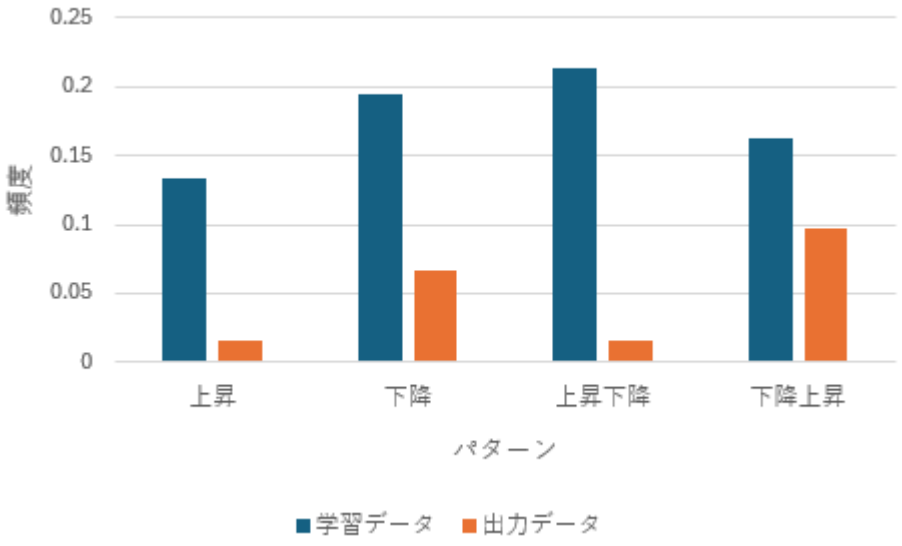
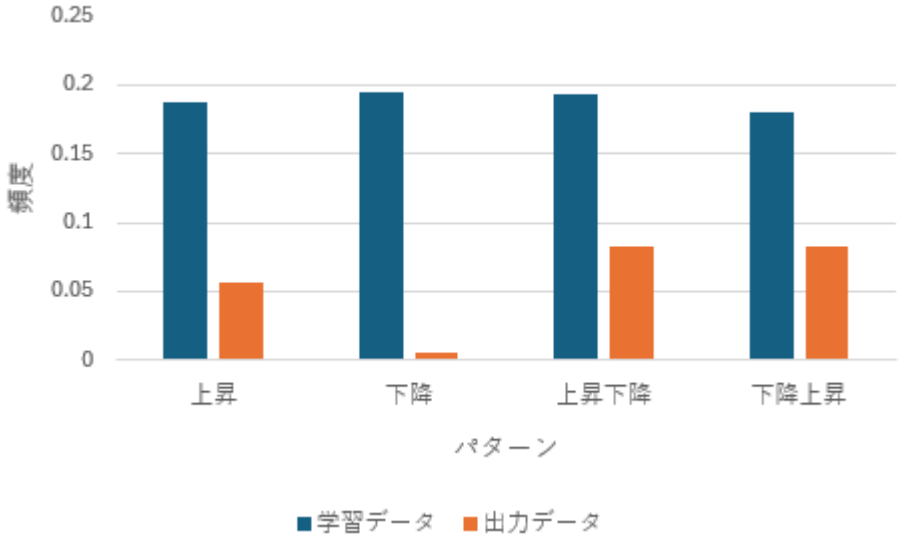


図５　洞窟のBGMのtrigram分析の結果



第５章　考察

5.1　学習させたデータの特徴

森のBGMの学習データ20曲について，テンポ，メロディの特徴，楽器の使われる頻度などの特徴を調べた．

テンポについて，120が7曲，110が3曲，60が2曲であった．加えて，テンポが200，125，115，105，80の曲が1曲ずつであった．

メロディの特徴について，同じパターンの繰り返しを用いた曲が7曲，高い音から低い音へ少しずつ音高が下がっていく特徴のある曲が5曲，音高が上がっていきその後下がる傾向のあった曲が4曲であった．また，特徴の無い曲は4曲であった．

楽器の使われる頻度については，最も頻度の多い楽器が笛，その次がピアノ，その次が弦楽器，木管楽器，鉄琴であった．

洞窟のBGMの学習データ20曲についても，森のBGMと同様に特徴を調べた．

テンポについて，120が6曲，100が4曲，110が3曲，180が2曲であった．加えて，テンポが190，150，115，90，80の曲が1曲ずつであった．

メロディの特徴について，同じパターンの繰り返しを用いた曲が10曲，高い音から低い音へ少しずつ音高が下がっていく特徴のある曲が2曲であった．また，音高が上がっていきその後下がる傾向のあった曲と音高の変化がなだらかな曲が1曲ずつであり，特徴のない曲が4曲であった．

楽器の使われる頻度については，最も頻度の多い楽器が笛，その次がピアノ，その次が弦楽器，木管楽器，鉄琴であった．

（※文責：野本藍里）

5.2　Ngram分析から分かったこと

5.2.1　unigramの結果から分かったこと

　洞窟のBGMについて，AIに学習させたメロディのパターンの傾向がほぼ等しいという結果が得られた．これは，AIに学習させたメロディに同じパターンのメロディを繰り返す曲が多かったためだと考えられる．

　AIが生成したメロディにはDマイナースケールに含まれる音があまり見られなかった．このことから，AIは音の高さをあまり学習していないということが明らかになった．

　また，AIが生成したメロディには休符が多く含まれるという傾向も見られた．

（※文責：野本藍里）

5.2.2　trigramの結果から分かったこと

　森のBGMについて，上昇を含むパターンの生成があまりできていないという結果が得られた．これは，学習データの中でほかのパターンに比べて上昇のみのパターンが少ないことから，上昇のみのパターンの出力が減ったことから上昇した後に下降するパターンが少なくなったと考えられる．

　洞窟のBGMについて，学習データのパターンの頻度がほぼ等しくなるという結果が得られた．これは，AIに学習させた曲に似たパターンを繰り返す曲が多かったためだと考えられる．また，下降のみのパターンの生成があまりできていないという結果が得られた．

（※文責：野本藍里）

5.3　今後の展望

　本プロジェクトでは，20曲の音源を学習データとして用い，LSTMモデルによって生成されたメロディパターンを基にBGMの作曲を行った．その結果，森のBGMでは生成メロディの約36%，洞窟のBGMでは約53%を参考にするなど，AIによるメロディ生成が実際の作曲に一定程度寄与していることが確認できた．また，N-gram分析の結果から，特定のメロディパターンについては十分に学習が行われていることが明らかになった．

一方で，使用したモデルや学習データの規模には制約があり，生成されるメロディの多様性や楽曲構造の一貫性には改善の余地が残されている．今後は，LSTM以外にもRNNモデルやマルコフ連鎖などの異なるアルゴリズムを用いてメロディ生成を行い，それぞれの生成結果を比較・分析することで，作曲支援における各手法の有用性を検証することが考えられる．

さらに，学習データ数の増加や感情・シーン情報を入力条件として取り入れることで，よりゲーム内状況に適応した音楽生成が可能になると期待される．これらの取り組みにより，人間の創造性を補完・支援するAI音楽生成システムの実現に，より一層近づくと考えられる．

（※文責：原耀良）

第６章　参考資料

Balkwill，L. L.，& Thompson，W. F. (1999). A cross-cultural investigation of the perception of emotion in music: Psychophysical and cultural cues. *Music Perception，17*(1)，43–64.https://doi.org/10.2307/40285496

Dromey，C.，Holmes，S. O.，Hopkin，J. A.，& Tanner，K. (2015). The effects of emotional expression on vibrato. *Journal of Voice，29*(2)，170–181.https://doi.org/10.1016/j.jvoice.2014.06.003

Du Boisn，F.(2019）.*作曲の科学 : 美しい音楽を生み出す「理論」と「法則」*（木村彩訳）. 講談社.

Gabrielsson，A.，& Juslin，P. N. (1996). Emotional expression in music performance: Between the performer's intention and the listener's experience. *Psychology of music，24*(1)，68-91.

Gregory，A. H.，Worrall，L.，& Sarge，A. (1996). The development of emotional responses to music in young children. *Motivation and Emotion，20*，341–348.https://doi.org/10.1007/BF02229231

Hevner，K. (1935). The affective character of the major and minor modes in music. *The American Journal of Psychology，47*(1)，103-118.

Hevner，K. (1935). Expression in music: a discussion of experimental studies and theories. *Psychological review，42*(2)，186.

Hevner，K. (1936). Experimental studies of the elements of expression in music. *The American journal of psychology，48*(2)，246-268.

Hevner，K. (1937a). *The affective value of pitch and tempo in music.* American Journal of Psy-chology，49，621-30.

Hevner，K. (1937b). *The affective value of pitch and tempo in music.* American Journal of

Psy-chology，49，621-30.

Huang, C. Z. A., Vaswani, A., Uszkoreit, J., Shazeer, N., Simon, I., Hawthorne, C., ... & Eck, D. (2018). *Music transformer*. arXiv preprint arXiv:1809.04281.

Huber，K. (1923). *Der Ausdruck musikalischer Elementarmotive. Leipzig，Germany*: Johann Ambrosius Barth.

Imberty，M. (1979). *Entendre la musique: Sémantique psychologique de la musique* [Understanding music: Psychological music semantics]. Dunod.

Juslin，P. N.，& Sloboda，J. A. (Eds.)(2001). *Music and emotion*: Theory and research(pp.56-94).(Masayuki Sato) Oxford University Press. (Original work published 2001)

Juslin，P. N.，& Sloboda，J. A. (Eds.)(2001). *Music and emotion*: Theory and research(pp.124-159).(Kengo Ohgushi) Oxford University Press. (Original work published 2001)

Juslin，P. N.，& Laukka，P. (2004). Expression，perception，and induction of musical emotions: A review and a questionnaire study of everyday listening. *Journal of New Music Research，33*(3)，217–238.https://doi.org/10.1080/0929821042000317813

Juslin，P. N.，& Sloboda，J. (2011). *Handbook of music and emotion: Theory，research，applications*. Oxford University Press.

Kastner，M. P.，& Crowder，R. G. (1990). Perception of the major/minor distinction: IV. Emotional connota参考資料tions in young children. *Music Perception，8*(2)，189–201.https://doi.org/10.2307/40285496

Kawanobe, M., & Kameda, M. （2009）. *音楽作品の感情価測定尺度と配色イメージスケール間*

のマッピング　映像情報メディア学会誌, *63*(3), 365-370

McCulloch，R. (1999). *Modality and children's affective responses to music* [Undergraduate project for Perception and Performance course (Ian Cross，instructor)]. Cambridge.

Meyer，L. B. (1973). *Explaining music: Essays and explorations*. University of California Press.https://doi.org/10.1525/9780520333109

Owen，H. (2000). *Music theory resource book*. (No publisher).

Panda，R.，Malheiro，R.，& Paiva，R. P. (2023). Audio features for music emotion recognition: A survey. *IEEE Transactions on Affective Computing，14*(1)，68–88. https://doi.org/10.1109/TAFFC.2020.3032373

Shin，A. (2024). *深層学習による自動作曲入門*. オーム社.

Thompson，W. F.，& Robitaille，B. (1992). Can composers express emotions through music?. Empirical studies of the arts，10(1)，79-89

Watson，K. B. (1942). The nature and measurement of musical meanings. *Psychological Monographs*，*54*(2)，i.

Webster，G. D.，& Weir，C. G. (2005). Emotional responses to music: Interactive effects of mode，texture，and tempo. *Motivation and Emotion，29*， 19–39.https://doi.org/10.1007/s11031-005-4414-0

Wu，B.，Horner，A.，& Lee，C. (2014). The correspondence of music emotion and timbre in sustained musical instrument sounds. *Journal of the Audio Engineering Society*，

62(10)，663–675.

Yamada，T.（2007）. N-*gram方式を利用した漢字文献の分析　立命館白川静記念東洋文字文化研究所紀要*，1，1–23

第7章　付録

下記のコードは、3種類、5段階の印象の数値を入力としてメロディパターンを出力するLSTMモデルのプログラムである．

```
# -*- coding: utf-8 -*-
"""
Created on Wed Jul 23 17:13:31 2025



@author: Otaki
"""



import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM，Dense
from mido import Message，MidiFile，MidiTrack
from tensorflow.keras.layers import Masking
import config



def LSTMmodel():



    # ===== ステップ1: CSV読み込み・前処理 =====
```

```
for i in range(5):
    df = pd.read_csv("pitch_sequences.csv"，header=None，encoding="cp932")

    # 特徴量（例: 森1_2_3_2.mid → 2，3，2）を抽出
    features = df[0].str.extract(r'_(\d+)_(\d+)_(\d+)')
    features = features.astype(int)
    df = df.drop(columns=0)

    # 音高データ取得（127は欠損とみなすため np.nan に）
    pitches = df.iloc[:，:128].replace(127，np.nan).values

    # ===== 特徴量の n 番目が m のサンプルをテストに使う =====
    feature_index = config.feature_index
    feature_value = config.feature_value

    # 条件に合うインデックスを抽出
    test_indices = np.where(features.iloc[:，feature_index] == feature_value)[0]
    train_indices = np.where(features.iloc[:，feature_index] != feature_value)[0]

    # 学習用・テスト用に分割
    X_all = np.repeat(features.values[:，np.newaxis，:]，pitches.shape[1]，axis=1)
    y_all = pitches

    X_train = X_all[train_indices] #一応残してる
    y_train = y_all[train_indices]

    X_test = X_all[test_indices]
```

```
y_test = y_all[test_indices]



# ===== ステップ2: 標準化 =====

scaler = StandardScaler()


# 標準化時は欠損値無視でfitする必要がある(NaNを無視)

pitches_mean = np.nanmean(pitches，axis=0)

pitches_std = np.nanstd(pitches，axis=0)

pitches_scaled = (pitches - pitches_mean) / pitches_std


# スケーリング後，NaNは0に（マスク対象にするため）

pitches_scaled[np.isnan(pitches_scaled)] = 0


# ===== ステップ3: LSTMデータ整形 =====

X = features.values

y = pitches_scaled


X = np.repeat(X[:，np.newaxis，:]，y.shape[1]，axis=1) # (N，128，3)

y = y.reshape((y.shape[0]，y.shape[1]，1))          # (N，128，1)


# ===== ステップ4: モデル構築・学習 =====

model = Sequential()

model.add(Masking(mask_value=0.0，input_shape=(128，3))) # 入力特徴量にマスクを追加するならここ

model.add(LSTM(64，return_sequences=True))

model.add(Dense(1))
```

```
model.compile(optimizer='adam'，loss='mse')

model.fit(X，y，epochs=100，batch_size=8)


# ===== ステップ5: 特定の特徴量に基づき，MIDIを5つ生成 =====


# 出したい印象(特徴量)を指定

input_feature = config.input_feature


#for i in range(5):

  # 同じ特徴を128タイムステップ分複製（1サンプル）

repeated_feature = np.repeat(input_feature[:，np.newaxis，:]，128，axis=1)


# 予測

predicted = model.predict(repeated_feature)

predicted_flat = predicted.reshape(1，-1)

predicted_original = predicted_flat * pitches_std + pitches_mean


# 整形：四捨五入→int変換→休符処理

final_notes = np.round(predicted_original).astype(int).flatten()

final_notes[final_notes < 1] = 0


# ===== 4つ以上同じ音が連続する場合の処理 =====

processed_notes = final_notes.copy()

count = 1  # 連続回数カウンタ



for j in range(1，len(final_notes)):
```

```
    if final_notes[j] == final_notes[j - 1] and final_notes[j] != 0:
        count += 1
    else:
        # 連続が途切れた時，もし4回以上なら間の音を0にする
        if count >= 4:
            start = j - count
            end = j - 1
            processed_notes[start + 1:end] = 0
        count = 1

# 末尾が連続して終わった場合の処理
if count >= 4:
    start = len(final_notes) - count
    end = len(final_notes) - 1
    processed_notes[start + 1:end] = 0

final_notes = processed_notes

# ===== MIDI書き出し =====
mid = MidiFile()
track = MidiTrack()
mid.tracks.append(track)

for note in final_notes:
    if note > 0:
```

```
            track.append(Message('note_on'，note=note，velocity=64，time=0))
            track.append(Message('note_off'，note=note，velocity=64，time=120))
        else:
            # 音を出さずに時間だけ進める
            track.append(Message('note_off'，note=0，velocity=0，time=120)) # または time=120 のダミーイベントだけtime を足す

    filename = f"generated_output{i+1}.mid"
    mid.save(filename)
    print(f"{filename} を保存しました．")

# ===== テスト予測・精度評価 =====
y_pred = model.predict(X_test).reshape((len(test_indices)，128))
y_pred_inverse = y_pred * pitches_std + pitches_mean
y_pred_final = np.round(y_pred_inverse).astype(int)

# 欠損扱いを除外し，正解率を出す
y_true = y_test.astype(int)
mask = ~np.isnan(y_test)
correct = (y_pred_final == y_true) & mask
accuracy = correct.sum() / mask.sum()

print(f"特徴量 {feature_index} 番目が {feature_value} のMIDIに対する正答率: {accuracy:.2%}")
```