

# 音楽づくりの学習材《ステージ》の捉え直しと改変

## — プログラミング教育のための学習材化に向けて —

寺内 大輔<sup>1</sup>

### 要約

本稿は、筆者が2016年に開発した音楽づくりのための学習材《ステージ》を、プログラミング教育のための学習材として位置づけるための論考である。まず、筆者が2020年に提案した〈演奏行為のアルゴリズムを構築する活動〉をふまえ、《ステージ》を用いた活動が促す思考と〈プログラミング的思考〉との共通性を検討する。次に、プログラミングの基本的な要素のひとつ「分岐」と、プログラムを明快に構築するための手続き「命令の統合」を盛り込んだ改変による新たな学習材《ネクストステージ (Next stage)》の試作過程を示す。最後に、2つの学習材とプログラミング教育のねらいとの関わりを整理するとともに、評価のあり方およびカリキュラムとしての位置づけの可能性を示す。

キーワード: プログラミング教育, プログラミング的思考, 小学校音楽科, 音楽づくり, 学習材 (教材) 開発

## 1. はじめに

令和2年度から小学校段階のプログラミング教育が必修化された。教科を問わず、「各教科等の特質に応じて」実施されることとなっており（文部科学省2020：p.9）、そのねらいとして次の3つが示されている。

①「プログラミング的思考」を育むこと、②プログラムの働きやよさ、情報社会がコンピュータ等の情報技術によって支えられていることなどに気付くことができるようにするとともに、コンピュータ等を上手に活用して身近な問題を解決したり、よりよい社会を築いたりしようとする態度を育むこと、③各教科等の内容を指導する中で実施する場合には、各教科等での学びをより確実なものとする（文部科学省2020：p.11）

筆者は、このことを踏まえ、小学校音楽科におけるプログラミング教育のための新たな学習材（「教材」という語を用いることもできるが、本稿では「学習材」という語で統一）の開発に取り組んできた。その一環として、筆者は、2016年に開発した音楽づくりのための学習材《ステージ》（寺内2016）をもとに、新たな学習材《ネクストステージ》を試作した。前作《ステージ》は、もともとプログラミング教育を意識して開発されたものではなかったが<sup>1</sup>、そこにはプログラミング教育に関連

させ得る構造、とりわけ、筆者がかつて提案した〈演奏行為のアルゴリズムを構築する活動〉としての特徴をそなえていることに気づいた。〈演奏行為のアルゴリズムを構築する活動〉については、寺内（2020）ですでに論じているため本稿では割愛するが、簡略に説明すると、音楽を〈音による構造物〉として捉えるのではなく、〈演奏行為の集積〉として捉え、そのアルゴリズムを構築することによって音楽をつくる活動を指す。

本稿では、前作《ステージ》をプログラミング教育的視点で捉えなおすとともに、新たな学習材《ネクストステージ》の試作過程を示す。また、両学習材のプログラミング教育のねらいとの関連性、評価のあり方の可能性、カリキュラムへの位置づけの可能性についても考察する。

## 2. 《ステージ》の概要と〈プログラミング的思考〉との共通性

本稿で取り上げる学習材《ステージ》は、その方法自体が〈演奏行為の集積〉としての性格が強いものであり、それゆえに〈演奏行為のアルゴリズム〉の構築との親和性が高いと考えられる。本章では、《ステージ》の概要について述べるとともに、その活動が促す思考と〈プログラミング的思考〉との共通性を検討する。

<sup>1</sup> 広島大学

## 2.1. 《ステージ》の概要

《ステージ》は、3～5名程度のグループによる音楽づくり活動の方法を学習材化したものである。全貌は、「実践者のためのインストラクション」として、寺内(2016)に収録しているほか、筆者のウェブサイトでも公開している。必要に応じて参照されたい<sup>2)</sup>。

紙と付箋紙、筆記具を用いる。メンバーは、各自「やってみたい表現」「得意な表現」を付箋紙にメモする。歌や楽器のような音楽的な表現だけでなく、踊りや身振りといった身体的な表現、朗読や演技のような言葉を伴う表現など、どのような表現でも取り入れることができる。各自の「やってみたい表現」「得意な表現」が書かれた付箋紙を「表現カード」と呼ぶ(図1)。なお、表現の名称はメンバーが自分(または自分たち)で命名する。

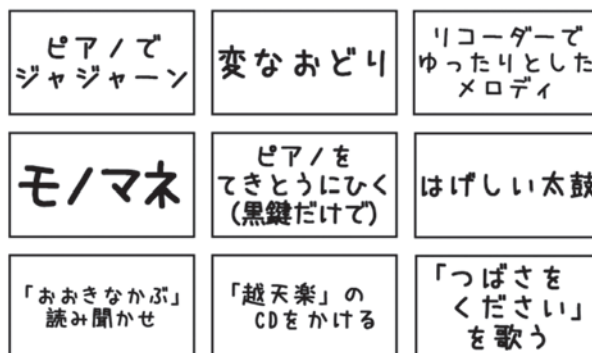


図1 表現カードの例(寺内 2016 : p.59)

次に、「表現カード」を使用する順序や組合せについて話し合い、全体の楽譜をつくる。この楽譜を「演奏シート」と呼ぶ(図2)。最後に、実際に試奏しながらブラッシュアップしていく。この時、必要に応じて、「→」、「ここまで」などの指示を直接書き込むこともできる。

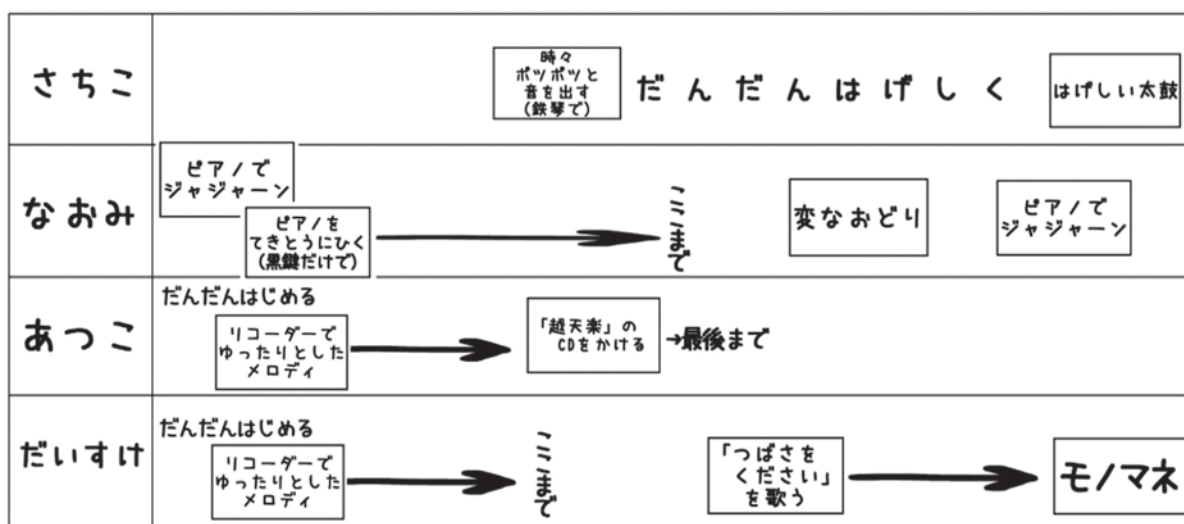


図2 《ステージ》における4名での演奏の場合の演奏シートの例(寺内 2016 : p.60)

以上が、《ステージ》の概要である。

次節では、この学習材を用いた活動が促す思考と〈プログラミング的思考〉との共通性を検討する。なお、前述のように、《ステージ》は、〈プログラミング的思考〉を働かせるということとは別の教育的意義が意識されたものであったが、本稿では、〈プログラミング的思考〉との関わりのみに焦点を当てて論を進めることとする。

## 2.2. 〈プログラミング的思考〉との共通性

《ステージ》は、コンピュータを使わない—いわゆるアンプラグドな—活動であるが、その活動にはプログラ

ミング教育のねらいのひとつである〈プログラミング的思考〉を働かせる過程と共通する過程が含まれていると考えられる。

〈プログラミング的思考〉は「自分が意図する一連の活動を実現するために、どのような動きの組合せが必要であり、一つ一つの動きに対応した記号を、どのように組み合わせたらいいのか、記号の組合せをどのように改善していけば、より意図した活動に近づくのか、といったことを論理的に考えていく力」とされ(文部科学省2020 : p.9)、そのイメージとして図3が示されている。

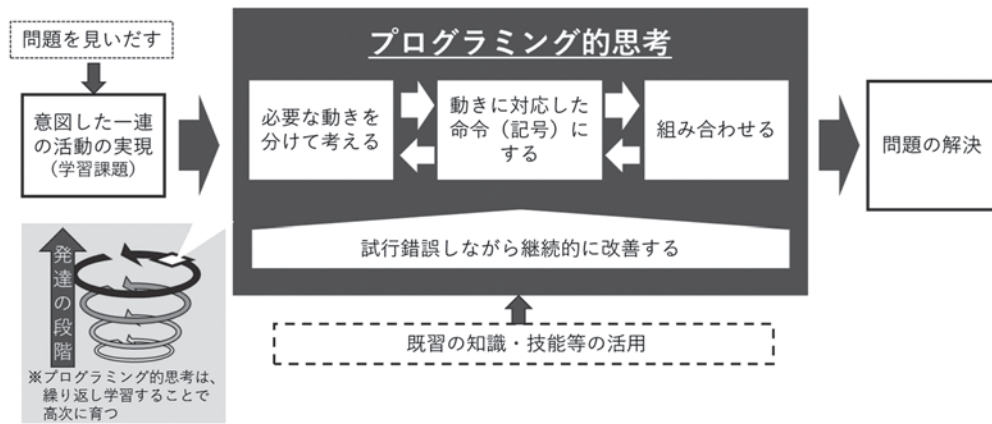


図3 〈プログラミング的思考〉を働かせるイメージ(文部科学省2020 : p.16)

《ステージ》の場合、「自分が意図する一連の活動」がすなわち「つくろうとする(演奏行為の集積としての)音楽」である。そして、「一つ一つの動きに対応した記号」にあたるものが「表現カード」,「記号の組合せ」を視覚的に示す機能を持ったものが「演奏シート」である。

より細かな過程との関わりについても確認しておきたい。〈プログラミング的思考〉を働かせるイメージである図3には、「必要な動きを分けて考える」、「動きに対応した命令(記号)にする」「組み合わせる」という3つの思考を、「既習の知識・技能等」を活用し、「試行錯誤しながら継続的に改善」しながら往還することが示されている。《ステージ》では、メンバーが自らの「やってみよう表現」「得意な表現」をもとに「表現カード」を出し合うところからはじまるが、これは音楽をつくるための素材—すなわち「必要な動き」—となるものであり、同時に「既習の知識・技能等」にあたるものでもある。さらに、一つひとつの「表現カード」に「モノマネ」「はげしい太鼓」といった名前を付けることは、「動きに対応した命令(記号)にする」ことにあたる。次に、「演奏シート」を作成したり、実際の演奏をとおして修正したりする活動は、それぞれ「組み合わせる」「試行錯誤しながら継続的に改善する」ことにあたる。

ただし、《ステージ》の活動過程には、図3と合致していない点が2点ある。1点目は、《ステージ》では、「自分が意図する一連の活動」にあたるイメージが活動のはじめに明確化されているわけではないことである。文部科学省による〈プログラミング的思考〉の定義では、「自分が意図する一連の活動」がはじめから明確にイメージされており、活動はそれを実現するために行われるように読めるが、長山(2019)も指摘しているように、音楽づくり活動では、「自分が意図する一連の活動」に相当する〈つくる対象としての音楽〉—すなわち、できあがる音

楽—は、活動過程において柔軟な変更・修正を伴いながら次第に明確化していくものである。そのため、文部科学省(2020)が示している「思い付きや当てずっぽうで命令の組合せを変える」のではないこと(p.15)についても、音楽づくり活動では必ずしもその限りではない。《ステージ》の活動においても、むしろ「思い付きや当てずっぽう」—すなわち、結果の予想の確実性が低い発想—をも積極的に試しながらつくっていくことができる。

2点目は、《ステージ》における活動の出発点が、「表現カード」に表される各自の「やってみよう表現」「得意な表現」であることである。それゆえ、図3にある「必要な動き」は、すでに「表現カード」として音楽をつくる最小単位となっており、あらためてこれらを「分けて考える」必要はない。

以上のことから、《ステージ》が促す思考は図4のようなイメージになると考えられる。

続いて、プログラミングを支える基本的な要素との関連についても確認しておきたい。一般的に、プログラミングを支える基本的な要素とされているのは、「順次」「反復」「分岐」の3つだが(文部科学省2020 : pp.13-14)<sup>3)</sup>、「演奏シート」には「表現カード」を実行する順序が示されているように、「順次」が織り込まれている。「反復」については、「演奏シート」に直接書き込むことによって示すことができる。例えば、図2の例では、同じ表現の継続を示すために「ここまで」「最後まで」という書き込みがあるが、同様の方法で「反復」を示すことも可能である。他方、不確定な結果をもたらす事柄について、何らかの条件を設定して命令(記号)を分岐させることを意味する「分岐」については、《ステージ》には含まれていない。

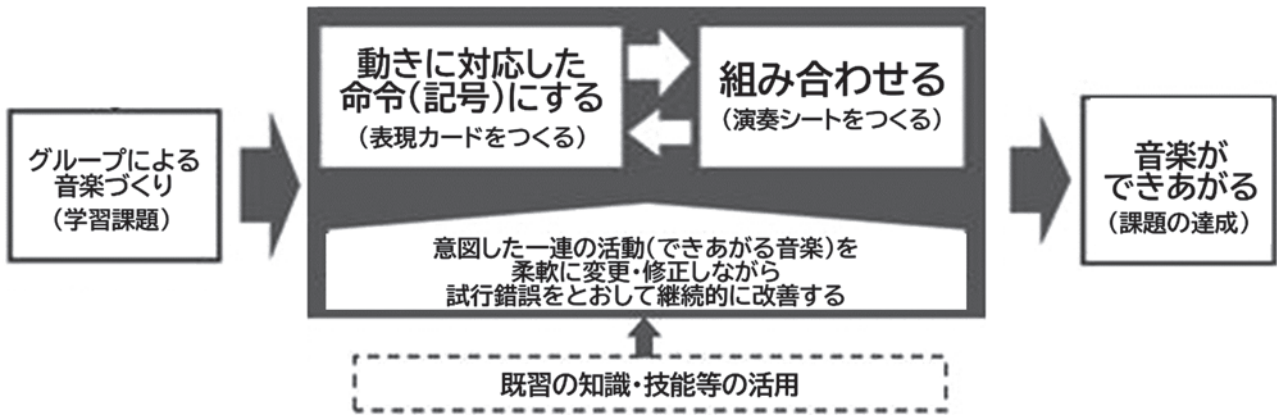


図4 《ステージ》が促す思考のイメージ

### 3. 《ネクストステージ》の試作

本章では、《ステージ》の新たなバージョンである《ネクストステージ》の試作の方針を示す。なお、《ネクストステージ》の全貌は、筆者のウェブサイトで公開しているため、必要に応じて参照されたい<sup>4)</sup>。

今回、《ステージ》の改変において意識した点は、プログラミングを支える基本的な要素のひとつである「分岐」と、プログラムを明快に構築するために便利な手続きである「命令の統合」を盛り込むことである。

#### 3.1. 「分岐」

《ネクストステージ》では、「演奏シート」作成時に「分岐」を取り入れることができるようにした。例えば表1のような可能性が考えられる。

表1 「演奏シート」作成時に「分岐」を取り入れる例

条件設定の対象例	「分岐」を取り入れる具体例
演奏中の演奏者の判断	演奏者 A が耳を触るサインを出すか出さないかによって演奏者 B の行動が異なる
結果が不確定的な表現	演奏者 A が表現「けん玉」を行った後、それが成功したかしなかったかによって演奏者 B の行動が異なる
聴衆の反応	演奏者 A が表現「モノマネ」を行った後、聴衆から大きな笑いが沸き起きたか起きなかったかによって演奏者 B の行動が異なる
偶発的な出来事	演奏するとき太陽が出ているか雲に隠れているかによって演奏者の行動が異なる

このような「分岐」を「演奏シート」に反映させることで、音楽的には次の2点の効果がもたらされる。1点目は、演奏するたびに異なる結果が生じることによって、音楽の展開のヴァリエティが広がることである。2点目は、演奏過程に不確定的な要素が加わることで演奏者が何らかの要素に即応する場面が増えることである。このことには、児童が即興的に表現するための力を高めることが期待できる。また、プログラミング教育的にも、児童が自然と「分岐」という考え方を意識する仕掛けとなることが期待できる。

#### 3.2. 「命令の統合」

プログラミングには、図3の「組み合わせる」に関わって、いくつかの命令をまとめる手法がある。例えば、ロボットを歩かせる命令の場合「前に2歩すすむ」→「右を向く」→「前に2歩すすむ」→「右を向く」→「前に2歩すすむ」→「右を向く」という一連の命令を、「右回りで小さく一回まわる」と名付けたひとつの命令に統合するといった手続きである。こうした手続きは、プログラミングでは「関数」、「メソッド」、「サブルーチン」などと呼ばれている。それぞれの呼び名の厳密な定義はプログラミング言語によって異なり、小学校のプログラミング教育でよく用いられるプログラミング言語のひとつ「Scratch」では「ブロック定義」と呼ばれている。本稿では、このような手法を、「命令の統合」と呼ぶこととする。「命令の統合」には、プログラムが見やすく、簡潔になるというメリットがある<sup>5)</sup>。

本開発にあたっては、《ステージ》で用いられる複数の表現カードを1枚の表現カードに統合するという発想を導入し、このような統合を「合わせ技」と名付けた。なお、「合わせ技」としての「表現カード」にどのような名前を付けるかは、通常の「表現カード」と同様に、児童

が任意に決める。図5は、「はげしい太鼓」「モノマネ」「ピアノをてきとうにひく(黒鍵だけで)」という3つの表現カードを1つに統合し、「モノマネ祭り」という新たな名前を付けた例である。

このような統合によって、プログラムとしての役割を持

つ「演奏シート」は簡潔になる。《ステージ》、《ネクストステージ》では、プログラムを実行する(演奏する)主体がコンピュータではなく、児童自身であるため、「演奏シート」が適度に簡素化されることは、全体の流れを把握しやすくなるという点で実用的な利点であるといえる。

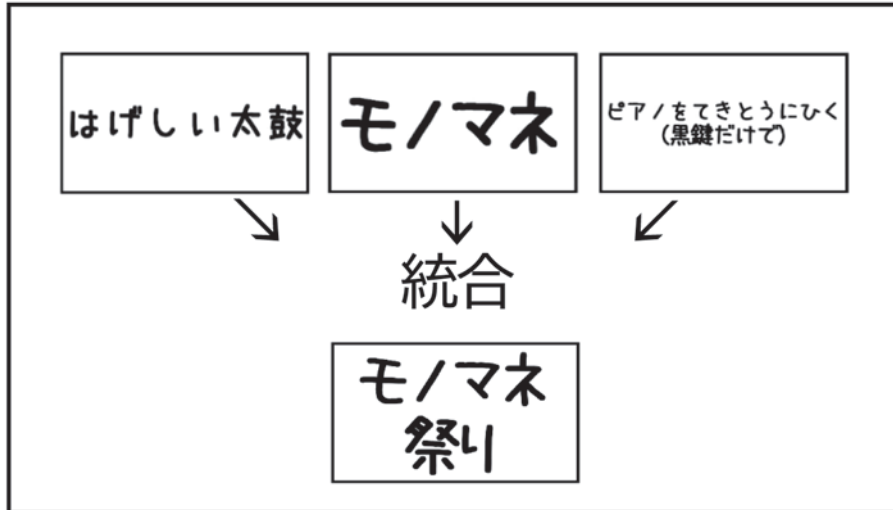


図5 《ネクストステージ》における「合わせ技」の例

以上が、《ステージ》への追加要素である。いずれも、《ネクストステージ》のインストラクションに反映させているが、本稿では、そのなかから、「分岐」と「命令の統合」が取り入れられた「演奏シートの例」のみを示す(図6)。この場合、表現カード「モノマネ」の箇所には「分岐」が、「モノマネ祭り」、「しみり読み聞かせ」、「リコーダーおどり」の箇所には「命令の統合」が、それぞれ反映されている。

なお、前作《ステージ》は、前述したように、小学校音楽科における音楽づくり活動を想定して開発された学習材であったが、これまで、一般を対象とした音楽祭における音楽づくりのワークショップで用いられたこともあった。《ネクストステージ》も同様に、学習材としてだけでなく、数名で音楽づくりを行う際の方法として、幅広い現場で活用可能である。

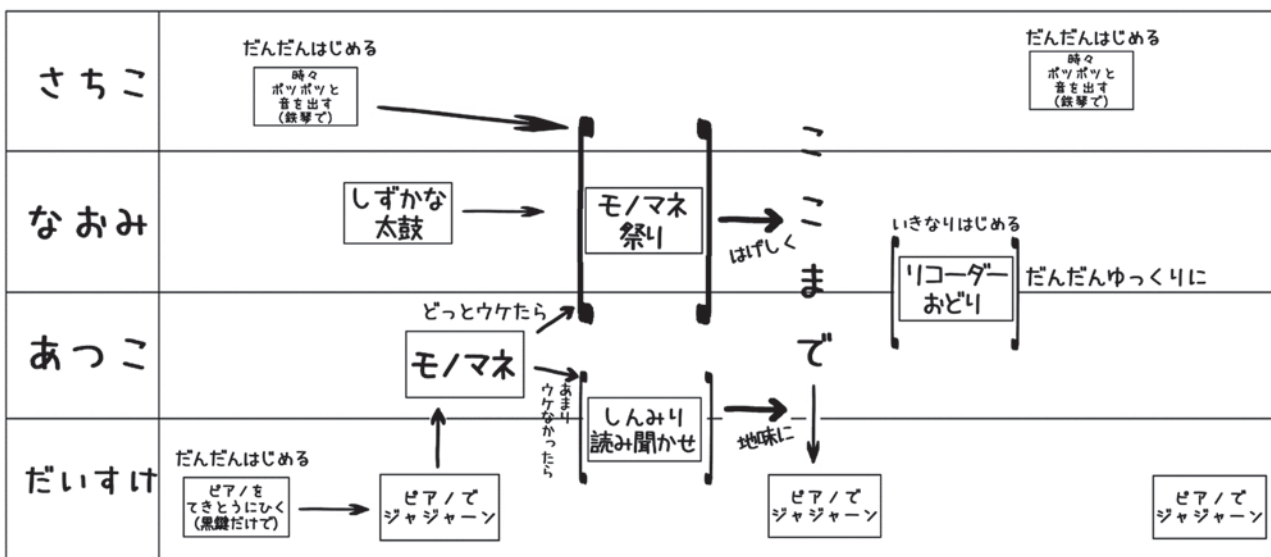


図6 《ネクストステージ》における4名での演奏の場合の演奏シートの例

## 4. 考察

本章では、《ステージ》および《ネクストステージ》と、プログラミング教育との関連性について、3つの視点で考察する。

### 4.1. プログラミング教育のねらいとの関連性の整理

ここでは、本稿冒頭に示したプログラミング教育の3つのねらいのうち、①と②に焦点を当て、《ステージ》および《ネクストステージ》に埋め込まれた構造との関わりを整理する。

まず、ねらいの①として示されている〈プログラミング的思考〉との関わりについて述べる。2.2.において、前作《ステージ》が促す思考と〈プログラミング的思考〉との共通性を整理したが、新作《ネクストステージ》においても、それらには基本的に変わりはない。これに加え、「分岐」については、導入すればするほどプログラム（演奏シート）が複雑になり、結果として働かせる〈プログラミング的思考〉も複雑になる。他方「命令の統合」はその複雑な思考を整理するための重要な役割を担うことになる。その意味で、これら2つの要素の導入は、《ステージ》よりも一層高度な思考を促す装置となることが考えられる。

次に、ねらいの②との関係についても考えてみたい。ねらいの②では、プログラムの働きやコンピュータの特質に焦点が当てられているが、「分岐」「命令の統合」の導入は、このこととも深くかかわっていると考えられる。

「分岐」は、前述したように、プログラミングを支える基本的な要素である。何らかの条件によって行動や状況が分岐することは、コンピュータ・プログラミングに限らず、多くの遊びのルール（約束事）によって生じるアルゴリズムがそなえている要素である。しかしながら、「分岐」は、現在の音楽科で広く行われているプログラミング活動ではあまり取り入れられていない考え方である。それは、現在の音楽科で広く行われているプログラミング活動が、本稿で取り上げたような〈演奏行為のアルゴリズムを構築する活動〉ではなく、「旋律をつくる活動」や「リズム・パターンを組み合わせる音楽をつくる活動」に代表されるような、〈音による構造物〉をつくることを目指しているからであるが、「分岐」という考え方は、そのような活動にも取り入れることが可能である。可能性のひとつとして、速度、強弱、リズム等の諸要素が何らかの条件によって変化するような音楽が挙げられる。例えば、コンピュータ・ゲームにおいて、プレイ時間のタイムアップが近づいた場合にBGMのテンポが速くなる仕掛けは、多くのゲームソフトで採用されている。《ネクストステージ》は、「分岐」を音楽づくりの過程に明確に位置づけ

たことによって、「プログラムの働き」が音楽や遊びにも大いに関わっていることを実感する体験を促すことが期待できるのではないかと思われる。

「命令の統合」は、前述したように、プログラムが見やすく、簡潔になるというメリットをもたらす。このことは、「演奏シート」を整理し、演奏しやすくなることを実感することによって学ぶことができるだろう。加えて、身近にあるプログラムの働きを学ぶことも高い親和性を持つ。例えば、ボタンひとつで洗濯、すすぎ、脱水、乾燥のすべてを行うことのできる全自動洗濯乾燥機は、数多くの命令を統合し、ひとつのボタンによる命令に変えている。このような、身近な生活との関連を取り上げることによって、ねらいの②に関係する学びにつなげることが期待できる。

また、《ステージ》および《ネクストステージ》は、ねらいの②に含まれる「コンピュータには得意なこととなかなかできないことがあることを、体験を通して気付かせること」（文部科学省2020：p.12）につながる特質もそなえている。なぜなら、両学習材は、それらがアンプラグドな学習材であるがゆえに、コンピュータと人間の違いを浮き立たせることになるからである。ここでは、具体的な違いを2点挙げてみたい。

1点目は、実現できる音楽がコンピュータにはできない表現性を帯びていることである。それは、音楽を構成する最小の素材が、〈音〉ではなく演奏者自身の〈演奏行為〉であることに拠っている。コンピュータは、〈音〉そのものや〈音〉の組み合わせを操作することに長けているが、「表現カード」として表されるような〈演奏行為〉は、それ自体多様な要素が複雑に作用して成立しており、なおかつその「表現カード」をつくった人物とも切り離しがたい性格をもった素材であるため、コンピュータで実現させるには不向きである。児童たちは、《ステージ》や《ネクストステージ》によってできあがる音楽と、コンピュータ・アプリケーションを使ってできあがる音楽との性格の違いを、体験を通して実感することができるだろう。

2点目は、「分岐」の使用頻度についてである。「分岐」は、図6中に示した「演奏シートの例」のなかでごくシンプルなものが1箇所しか使われていないように、あまり多用することができない。もし、《ネクストステージ》で「分岐」を数多く使えば、「演奏シート」が複雑になり過ぎてしまい、「演奏シート」に表すことも、また演奏することも現実的に困難になる。他方、そのような複雑なアルゴリズムは、コンピュータにとっては得意とする処理である。

### 4.2. 評価のあり方の可能性

現在、プログラミング教育の評価のあり方について、

文部科学省(2020)は、プログラミング教育のねらいの①と②を達成すること、教科としての学習をより深いものとする、そして、特に意欲的に取り組む姿やプログラムを工夫する姿を評価することを示している(p.21)。

このことを踏まえ、ここでは、実際の《ステージ》および《ネクストステージ》を用いた活動を想定し、その中から評価の対象となる〈プログラミング的思考〉とかかわる姿を仮説的に挙げてみたい。

まず、実際の演奏と「演奏シート」との関係を意識している姿や、そのために「演奏シート」を修正している姿を挙げる。こうした姿は、プログラミング教育的意義を意識しなければ、さほど重要ではない場合もあるだろう。なぜなら、音楽科の音楽づくり活動では、音楽をつくるのが第一の活動であり、楽譜をつくることは必要に応じて行うこととされている—いわば副次的な活動である—からである。楽譜の役割をもつ「演奏シート」と実際の演奏との整合性の高さにこだわる必然性はないため、「演奏シート」を演奏者が演奏するためのメモ(覚え書き)として位置づけるならば—そのような位置づけも可能である—演奏者たち自身が理解できさえすればよいと考えることもできる。他方、このような姿をプログラミング教育的意義に照らせば、「演奏シート」と実際の演奏との整合性を高めようとするのは、「命令」と「結果」との因果関係を意識している—図3の「動きに対応した命令(記号)にする」—重要な姿であると言える。つまり、授業者は、「演奏シート」を、授業のねらいに応じて柔軟に位置づける必要があるのである。

次に、「演奏シート」をわかりやすく示そうとしている姿を挙げる。特に、《ネクストステージ》では「合わせ技」を効果的に使うことによって「演奏シート」の簡略化をはかることもできる。これは、実現する音楽に照らしてそのためのアルゴリズムの構造を整理している姿—すなわち「プログラムを工夫」している姿—として見取ることができそうである。

続いて、児童が音楽をつくるために試行錯誤している姿についても挙げておきたい。ここで前提として確認しておきたいことは、2.2. で述べたように、音楽づくり活動では、「自分が意図する一連の活動」に相当する〈つくる対象としての音楽〉がつくる過程において次第に明確化していくということである。《ステージ》、《ネクストステージ》ともに、インストラクションのなかには、「より良い表現」を目指すことが示されている。これは、学習指導要領で、音楽づくりの活動が「創造性を発揮しながら自分にとって価値のある音や音楽をつくるものである」(文部科学省2018:p.23)と説明されていることに拠っているが、音楽づくりの過程では、その「自分にとって価値のある音や音楽」にあたる、その時につくろうとする音楽の

特質自体が柔軟に変わることは珍しくない。《ステージ》、《ネクストステージ》においても、例えば、活動当初は「かっこいい音楽をつくろうよ」と考えたとしても、「演奏シート」の作成時に一人の児童が「ここでこの『表現カード』とこの『表現カード』を同時に使えばお客さんから笑いが取れるんじゃない?」ということに気づけば、他の児童も「いいね。ではここからは積極的に笑いをとろうよ」と話が進んでいくように、当初設定した方向性が自然に変わることはしばしば起こり得る。さらに、自分たちの音楽の意図が「積極的に笑いをとる」ということになった途端、「それならこの『表現カード』のほうがいいんじゃない?」という新たな提案をするような場面もみられるだろう。すなわち、ここでの「試行錯誤」は、はじめに意図した目的—前述の例では「かっこいい音楽にする」—を実現するためだけでなく、自分たちの目的を柔軟に変更しながらも、目的と手段との関係を意識して工夫する姿であると言えるのである。

以上、3つの具体例を想定して示したが、前述したようにこれらはあくまでも想定にもとづいた仮説的なものである。今後は、実際の授業実践をとおした一層の検討が必要であると考えられる。

### 4.3. カリキュラムへの位置づけの可能性

#### 4.3.1. 「音遊び」や「即興的な表現」との接続性

筆者が〈演奏行為のアルゴリズムを構築する活動〉を提案した背景には、〈演奏行為の集積〉としての性格の強い「音遊び」や「即興的な表現」の活動と、プログラミング教育との接続性を高めることへの意識があった(寺内 2020)。そのために重要なこととして、「音遊び」や「即興的な表現」を実践させるだけでなく、これらを成立させる構造—すなわち〈演奏行為のアルゴリズム〉—に目を向けさせることが挙げられる。

もともと、「音遊び」や「即興的な表現」は、〈演奏行為のアルゴリズム〉を意識させることと関わって実践されてきた。学習指導要領解説に「次はこのような約束で表現してみたいなど、児童の提案によって、授業を展開していくことも考えられる」と示されているように(文部科学省2018:p.76)、「音遊び」や「即興的な表現」の〈演奏行為のアルゴリズム〉を成す要素である〈約束事〉を児童からの提案によって柔軟に変更することは珍しくないからである。このことは、〈演奏行為のアルゴリズム〉に目を向けさせるだけでなく、自らアルゴリズムを作り出す発想にもつながる可能性をもつものだが(寺内2020)、ここでとりわけ重要だと考えられる概念が、今回の開発で導入した「分岐」である。「分岐」は、現在の小学校音楽科で広く実践されている「音遊び」や「即興的な表現」においても、「○○さんが△△の合図を出したらリズムを

変える」のような形で含まれることが多い。そのため、児童が「音遊び」や「即興的な表現」の〈約束事〉に変更を加えようとするれば、「分岐」を意識する機会が一たとえ「分岐」という語を知らなくとも一自然と促されることになる。すなわち、今回の開発で取り入れた「分岐」は、「音遊び」や「即興的な表現」とプログラミング活動とを接続するための重要な要素のひとつなのである。

また、《ステージ》および《ネクストステージ》における「演奏シート」が、各奏者の演奏行為を示す楽譜としての役割だけでなく、「音遊び」や「即興的な表現」を成立させるアルゴリズムを視覚的に捉えるツールとしての特質を持っていることも重要である。現在、児童・生徒向けのプログラミング教育の現場では「Scratch」などのヴィジュアルプログラミング言語<sup>6)</sup>が多く採用されているが、この傾向の一因として、プログラムの構造が視覚的に捉えやすいことが挙げられる。「演奏シート」も同様に、〈演奏行為のアルゴリズム〉を視覚的に捉えることによって、「音楽をつくる」だけでなく「音楽を成立させるための仕組み自体をつくる」というメタ的な創造性を刺激し得ることが期待される。

#### 4.3.2. 〈演奏行為のアルゴリズムを構築する活動〉のなかでの位置づけ

《ネクストステージ》は、プログラミングにおける重要な要素である「順次」「反復」「分岐」をすべて反映させながら演奏行為を組み合わせることでできるシステムである。その意味で、《ネクストステージ》は、〈演奏行為のアルゴリズムを構築する活動〉のゴールイメージを実現する学習材のひとつとして位置づけることができそうである。

《ネクストステージ》の実践を、〈演奏行為のアルゴリズムを構築する活動〉のゴールイメージのひとつとした場合、その段階へ至るプロセスについても考えなければならない。このことは、筆者にとっての今後の課題のひとつだが、本稿の最後に、この課題に対する現時点での可能性として、〈演奏行為のアルゴリズムを構築する活動〉全体のカリキュラムイメージを、簡略に示したい。

まず、〈演奏行為の集積〉としての性格が強い音楽じたいに親しむ段階では「音遊び」や「即興的な表現」として様々な活動を数多く経験することが重要であると考えられる。このとき、行為のアルゴリズムを児童に意識させるため、4.3.1.で示した、児童からの提案によってルール（約束事）を改変するプロセス（文部科学省2018：p.76）を積極的に取り入れることが重要であると考えられる。

次に、〈一つ一つの命令に対応した記号〉を意識する活動が考えられる。これは、例えばハンドサインやカードなどを用いた即興演奏の実践<sup>7)</sup>などが考えられる。ま

た、筆者がかつて開発した別の学習材《ひみつのサインをつくれ！/見やぶれ！》（寺内 2020）も、演奏行為を促すサイン—すなわち〈一つ一つの命令に対応した記号〉に焦点を当てた、即興的かつゲーム的要素の高いものである。

続いて、〈演奏行為のアルゴリズム〉を構築することに焦点化した活動を、簡易なものから取り上げていくことが考えられる。例えば、坂入（2017）で紹介されている活動や、前作《ステージ》のようなシンプルで一連の動きを考える活動は、〈演奏行為のアルゴリズム〉構築の初歩的な段階に位置づくと考えられる。こうした活動のうち、《ネクストステージ》のような学習材の体験をとおして、〈演奏行為のアルゴリズム〉構築の手段を拓げていく可能性があると考えられる。

## 5. おわりに

本稿では、学習材《ステージ》を、プログラミング教育を意識した観点で捉えなおし、プログラミング教育に関わる要素をより強調した新たな学習材《ネクストステージ》の試作を示した。また、《ステージ》、《ネクストステージ》のプログラミング教育のねらいとの関連性、評価のあり方の可能性、カリキュラムへの位置づけの可能性を検討した。これらは、〈演奏行為のアルゴリズムを構築する活動〉のための学習材として、まだ数少ない具体例にかかわる成果でもある。

しかしながら、4.2で挙げたように、《ネクストステージ》の評価のあり方を実際の授業実践をとおして検討することは、今後の課題のひとつである。このことは、《ネクストステージ》の、学習材としての妥当性の検証にもつながるものである。

今後も、本稿で試みたような、既存の学習材の捉えなおしや新たな学習材の開発を引き続きすすめていきたい。合わせて、〈演奏行為のアルゴリズムを構築する活動〉のカリキュラムについても、本稿4.3で試みたような特定の学習材の位置づけの検討のみならず、〈音による構築物〉をつくる活動やコンピュータによって音を出すことを前提とした活動、さらには音楽科の学習内容との関係性を踏まえた全体像を意識した検討が必要であると思われる。

## 注

- 1) 同学習材の、音楽科としての実践および教育的意義については、寺内・甫出（2017）、Terauchi and Myodo（2021）などで議論されている。
- 2) 「ステージ」（寺内大輔ウェブサイト）、<https://dterauchi>.



com/stage.html (2021年12月11日アクセス)

- 3) 「コンピュータを動作させるために命令(記号)の組合せを考える際には、たとえば、命令(記号)を順序立てたり、条件を設定して命令(記号)を分岐させたり、命令(記号)を繰り返させたりすることなどが考えられます。これらは一般的にプログラミングを支える基本的な要素とされています」(文部科学省 2020:pp.13-14)
- 4) 「ネクストステージ」(寺内大輔ウェブサイト), <https://dterauchi.com/nextstage.html> (2021年12月11日アクセス)
- 5) ただし、プログラムの細部を見えにくくするというデメリットも併せ持っている。
- 6) プログラムを、テキストではなく視覚的なオブジェクトで表すプログラミング言語。グラフィカルプログラミング言語とも呼ばれる。「Scratch」はその代表例のひとつである。
- 7) ウォルター・トンプソン(Walter Thompson)が考案した《サウンドペインティング(Soundpainting)》(1974)、ジョン・ゾーン(John Zorn)が考案した《コブラ(Cobra)》(1984)、L. ブッチ・モリス(Lawrence "BUTCH" Morris)が考案した《コンダクション®(Conduction®)》(1985)などが代表例である。近年では、国内の音楽教育分野でも注目が高まっている(例えば、寺内 2015, 小枝 2020の実践など)。

## 参考文献

- Terauchi, D. & Myodo, H.(2021) Exploring the Pedagogical Possibilities of the Idea of Composition Based on Children's Interests and Strengths, in the Proceedings of The 13<sup>th</sup> Asia-Pacific Symposium for Music Education Research: Exploring Possibilities and Alternatives in a Changing Future, Morijiri, Y., Imada, T., Ogawa, Y.,(ed), Japan Music Education Society, pp. 219-227.
- 坂入優花(2017)「はくの流れにのってリズム遊びを楽しもう」小林祐紀・兼宗進編著・監修『コンピューターを使わない小学校プログラミング教育“ルビィのぼうけん”で育
- む論理的思考』翔泳社, pp.34-37.
- 小枝洋平(2020)「身体の動きを基盤とした即興演奏の可能性—障害の有無を架橋し共創する交流及び共同学習」『音楽教育実践ジャーナル』Vol.18, pp.46-55.
- 寺内大輔(2015)「児童の多様な表現を引き出す即興的表現活動の可能性—ジョン・ゾーン《コブラ》(アマチュア版)の制作と小学校音楽科における実践」『音楽教育実践ジャーナル』Vol.13, no.1, pp.94-105.
- 寺内大輔(2016)「児童の得意な表現を手がかりとした創作活動のための学習材の開発—ジョン・ゾーンの作曲手法を参照して」『初等教育カリキュラム研究』第4号, pp.53-63
- 寺内大輔・甫出頼之(2017)「表現活動のための学習材《ステージ》を用いた授業の開発:小学校における2つの授業実践の振り返りをとおして」『広島大学大学院教育学研究科紀要.第一部,学習開発関連領域』,66号,pp.63-70
- 寺内大輔(2020)「〈演奏行為のアルゴリズムを構築する活動〉の提案—小学校音楽科におけるプログラミング教育のあり方の検討をとおして—」『音楽教育学』49巻,2号,pp.25-32
- 長山弘(2019)「小学校音楽科におけるプログラミング教育のあり方の検討—授業実践事例を手がかりに—」『初等教育カリキュラム研究』第8号,初等教育カリキュラム学会,pp.55-67.
- 文部科学省(2018)『小学校学習指導要領(平成29年告示)解説 音楽編』東洋館出版社.
- 文部科学省(2020)「小学校プログラミング教育の手引(第三版)」,インターネット, [https://www.mext.go.jp/content/20200218-mxt\\_jogai02-100003171\\_002.pdf](https://www.mext.go.jp/content/20200218-mxt_jogai02-100003171_002.pdf) (2020/3/1取得)

## 参考ウェブサイト

寺内大輔ウェブサイト, <https://dterauchi.com>

付記:本研究は、JSPS科研費JP18K13161の助成を受けたものである。

# A Reinterpretation and Modification of the Music-Making Activity in “Stage”: Aiming for Use as Learning Material for Programming Education

Daisuke TERAUCHI

Hiroshima University

## Abstract

This paper is a discussion on how to position “Stage,” the activities for music-making developed by the author in 2016, as learning material for programming education. First, based on the “activity of algorithm construction of performance actions” proposed by the author in 2020, I examine the similarities between the thinking promoted by the activity using “Stage” and “programming thinking.” Next, I outline the process of prototyping new learning material “Next stage” by incorporating conditional branching, one of the fundamental elements of programming, and the callable unit, a simple procedure for constructing a program. Finally, the relationship between the two learning materials and the aims of programming education are summarized. In addition, I considered the possibilities for valuation and positioning as a part of the curriculum.

Keywords : programming education, programming thinking, elementary school music studies,  
music-making, development of learning materials