「教育臨床総合研究23　2024研究」

コンピュテーショナル・シンキングに基づく小学校低学年を対象としたプログラミング教育の実践

A Study of Practice of Programming Education for Early Elementary School Students Based on Computational Thinking

山　川　大　輝*　　深　見　俊　崇**
Daiki YAMAKAWA　　Toshitaka FUKAMI

要　旨

小学校プログラミング教育においては，低学年を対象とした実践が例示されていない，プログラミング教育を通じて育成すべき能力が明確でないといった問題点が存在する。本研究では，これらの問題点を踏まえ，小学校低学年を対象とするコンピュテーショナル・シンキングの育成を目指した２時間の授業を開発・実践した。その結果，児童の自己評価からは肯定的な成果が確認できたものの，児童の個人差への対応，問題解決の手順の理解度を確認するための評価方法といった改善すべき点も明らかとなった。

〔キーワード〕　小学校プログラミング教育，プログラミング的思考，
コンピュテーショナル・シンキング，小学校低学年

Ⅰ　はじめに

現代社会において，コンピュータは日常生活の様々な場面で活用されている。文部科学省（2020）の「小学校プログラミング教育の手引（第三版）」では，このような社会の到来を踏まえ，「コンピュータをより適切，効果的に活用していくためには，その仕組みを知ることが重要」（p.1）と指摘されている。コンピュータは人が命令を与えることによって動作する。この命令がプログラムであり，命令を与えることがプログラミングである（文部科学省 2020, p.1）。2017 年に改訂された小学校学習指導要領から，「児童がプログラミングを体験しながら，コンピュータに意図した処理を行わせるために必要な論理的思考力を身に付けるための学習活動」（文部科学省 2017, p.22）としてプログラミング教育が新たに導入されることになった。

2017 年改訂の小学校学習指導要領に例示されているプログラミングに関する学習活動は，理科第６学年の「A 物質・エネルギー」における電気の性質や働きを利用した道具があること等をプログラミングを通して学習する場面，算数科第５学年の「B 図形」における正多角形

*島根大学大学院教育学研究科
**島根大学教育学部

の作図を行う学習，総合的な学習の時間の情報に関する探究的な学習の３つである（文部科学省 2017）。これら３つの学習活動のみが小学校学習指導要領においては例示されており，低学年を対象としたプログラミング教育の実践例は示されていない。

諸外国に目を向けると，プログラミング教育が盛んに行われている英国においては，「コンピューティング」のナショナルカリキュラムにおいて，キーステージ１からキーステージ４までプログラミング教育で教えるべき内容が示されている（Department for Education 2013）。このキーステージ１には５－７歳の児童が含まれる。すなわち，英国においては，幼児期・低学年からプログラミング教育を実施することで，発達段階が進むにつれてより高度なプログラミングの学習活動を行うことができるようカリキュラムが構成されている。英国をモデルとして，小学校低学年からプログラミング教育の土台をつくる実践を提案するのが本研究の主題である。

ところで，日本における小学校プログラミング教育は，プログラミング的思考を育むことが目標とされている（文部科学省 2020, p.9）。文部科学省（2016）の「小学校段階における論理的思考力や創造性, 問題解決能力等の育成とプログラミング教育に関する有識者会議（第３回）議事録」では，このプログラミング的思考について「いわゆる『コンピュテーショナル・シンキング』の考え方を踏まえつつ，プログラミングと論理的思考との関係を整理しながら提言された定義である」と述べられているが，この記述からはコンピュテーショナル・シンキングをどのように踏まえたのかについて読み取ることができない（赤羽 2023, p.1）。本研究では小学校低学年を対象としたプログラミング教育の実践を提案することが目的となるが，提案する実践については，コンピュテーショナル・シンキングの考え方を踏まえ，児童がいかにその能力を獲得するのかという目的を明確にした実践を提案していく。

Ⅱ　日本における小学校プログラミング教育

小学校プログラミング教育が導入された目的として，「子供たちに，コンピュータに意図した処理を行うよう指示することができるということを体験させながら，将来どのような職業に就くとしても，時代を超えて普遍的に求められる力としての『プログラミング的思考』などを育むことであり，コーディングを覚えることが目的ではないということを明確にした」と説明されている（文部科学省 2016）。このように，日本のプログラミング教育では，プログラミング的思考を身に付けることが重要だと考えられている。ここで挙げられたプログラミング的思考とは,「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」（文部科学省 2016）と定義されている。

次に，小学校プログラミング教育の学習活動を確認する。小学校段階におけるプログラミングに関する学習活動は，①学習指導要領に例示されている単元等で実施するもの，②学習指導要領に例示されてはいないが，学習指導要領に示される各教科等の内容を指導する中で実施するもの，③教育課程内で各教科等とは別に実施するもの，④クラブ活動など特定の児童を対象として，教育課程内で実施するもの，⑤学校を会場とするが教育課程外のもの，⑥学校外での

プログラミングの学習機会の６つに分類される（文部科学省 2020，p.23）。文部科学省（2020）が公開している「小学校プログラミング教育の手引（第三版）」に掲載されている実践事例がいずれも高学年・中学年の児童を対象としたものであり，低学年の児童を対象としたものではない。そのため低学年を対象としたプログラミング教育の実践を以下に紹介する。

まず，「小学校を中心としたプログラミング教育ポータル」で公開されている実践のうち低学年のみを対象としたものは２点のみである。これらの実践はいずれも，②学習指導要領に例示されてはいないが，学習指導要領に示される各教科等の内容を指導する中で実施するものに該当する（表１，表２）。これら２つの実践は，いずれもプログラミングを行うことが目的ではなく，各教科での学びをより確実なものとするための学習活動としてプログラミングが設定されている。表１の実践の目標は「自分が表現したい場面を必要な助詞を選択して文に表すことを通して，文の中における主語と述語の照応関係や助詞の正しい使い方について理解することができる」であり，表２の実践の目標は「リズムや音楽の仕組みに興味・関心をもち，音楽づくりに進んで取り組む，リズムの面白さを感じ取りながら，どのように音楽をつくるかについて思いをもつ」，「リズム譜に親しみ，簡単なリズムを演奏したり，反復を生かしたリズムをつくったりすることができるようにする」と設定されている。いずれの実践においてもあくまで主たる目的は，文法理解や演奏など教科の内容を身に付けることに焦点化されている。だが，プログラミング教育によって育むことが期待される能力とはプログラミング的思考である。これら２点の活動はプログラミング的思考の定義を踏まえた，動きや音を組み合わせて意図したものに近づけるというプログラミングの活動を行ってはいるが，その活動によってプログラミング的思考が育まれたのかという部分については着目されていない。

一方，山田・小野田（2018）の実践は，算数科・生活科の合科的提案である。算数科では筆算について学ぶ際に位を縦に揃えて書く，一の位の計算をする，十の位の計算をする，の３点が共通して提示される。これらをフローチャートに見立てた授業を行い，生活科ではプログラミング教材を使用し，全６時間の単元で授業を計画した（山田・小野田 2018, p.62）。活動の内容としては，握手の順番を考える，お風呂掃除の順序を考えるといったように生活場面のアルゴリズムを単元の前半で考え，後半では前半で考えたアルゴリズムを実際にロボットにプログラミングする活動が行われた（山田・小野田 2018, p.62）。ただ，この活動は小学校高学年においてプログラミングに取り組むために，低学年段階で素地を作ることを目的としており，プログラミング的思考については言及されておらず，授業に対する評価項目も意欲的に取り組めたか，間違えても最後まで取り組めたか，プログラミングの基本的知識に気付けたかといったことを確認するアンケート調査に留まっていた（山田・小野田 2018, p.61-62）。

日本においてプログラミング教育がどのように捉えられているのか，そして日本でプログラミング教育がどのように行われているのか，具体的な実践例を確認した。日本において，プログラミング教育はプログラミング的思考を身に付けるために行われるものであるが，小学校低学年を対象としたプログラミング教育の実践においては，各教科での学びをより確実なものとするための学習活動としてプログラミングが設定されているため，獲得する能力や達成されるべき目標としてプログラミング的思考が設定されていない。

表１　小学校低学年を対象としたプログラミング教育の実践事例１

対象学年・対象教科	小学校第２学年・国語
対象都道府県・学校名	東京都・品川区立台場小学校
ICT 機器の利用環境	学校所有のタブレット型端末１人１台利用
教材タイプ・使用ツール	ビジュアル言語・Scratch
単元や題材の目標	自分が表現したい場面を必要な助詞を選択して文に表すことを通して，文の中における主語と述語の照応関係や助詞の正しい使い方について理解することができる。
学習の内容	本時では，一つ一つの助詞の意味を捉えた上で，自分が意図した場面（条件）に応じた助詞を選択していく Scratch のプログラミング体験を学習に取り入れることで，主語と述語の照応関係や助詞の正しい使い方について理解することができるようにする。本時の指導に当たっては，まず，学級全体で「ねこ（ ）ねずみ（ ）おいかける。」の文を例に助詞の役割に気付かせるとともに，Scratch の基本操作について説明し，学習の導入を行う。ここでは，助詞の「が」または「を」を選択して文に当てはめ，１文字入れ替わるだけでも場面（イラスト）が変わってしまうことに気付くようにする。次に，「わたし（ ）ボール（ ）かご（ ）なげた。」の文に，助詞の「は」「を」「に」を選択して当てはめるプログラミング体験を行う。これにより児童は，どのように順序立てて文字を組み合わせれば自分が表現したい場面（イラスト）になるのかを試行錯誤していくことになる。さらに，自分が作った文をイラストとともにペアやグループで伝え合うことにより，友達の表現との共通点や相違点に気付き，それぞれの主語と述語の照応関係や助詞の正しい使い方について考えることができるようにしていく。
掲載 URL	https://www.mext.go.jp/miraino_manabi/content/290.html

表２　小学校低学年を対象としたプログラミング教育の実践事例２

対象学年・対象教科	小学校第２学年・音楽
対象都道府県・学校名	大阪府・大阪市立茨田東小学校
ICT 機器の利用環境	学校所有のタブレット型端末２人１台利用
教材タイプ・使用ツール	ビジュアル言語・Scratch
単元や題材の目標	リズムや音楽の仕組みに興味・関心をもち，音楽づくりに進んで取り組む。 リズムの面白さを感じ取りながら，どのように音楽をつくるかについて思いをもつ。 リズム譜に親しみ，簡単なリズムを演奏したり，反復を生かしたリズムをつくったりすることができるようにする。
学習の内容	まずは，リズムカードを見ながら手でいろいろなリズムを打ち，カードに書かれたリズムを実際に音に出して確かめる。しかし，２年生の子どもたちにとって，リズム譜をみてすぐに正確なリズムで演奏することは難しいため，リズムカードを組み合わせていく段階で Scratch を用いる。Scratch による再生を耳で聴くことによって，聴きながらリズムを確認したり，より面白い音楽になるように何度もやり直したりすることが容易となる。さらには，次に児童が自分でリズムを演奏する際の，範奏として活用することができるという利点もある。また，Scratch 上で同じカードを反復させる「くりかえし」の機能を活用することで，音楽の仕組みとしての「反復」を意識することにもつながり，音楽の学習をさらに深めることができる。
掲載 URL	https://www.mext.go.jp/miraino_manabi/content/265.html

Ⅲ　コンピュテーショナル・シンキングとプログラミング教育

これまで述べてきた通り，日本におけるプログラミング教育では，プログラミング的思考の獲得が重要視されているが，「いわゆる『コンピュテーショナル・シンキング』の考え方を踏まえつつ，プログラミングと論理的思考との関係を整理しながら提言された定義である」（文部科学省 2016）と補足されている。このように，プログラミング教育が獲得を目指すプログラミング的思考のベースとなった考え方がコンピュテーショナル・シンキングと捉えることができる。ところが，この有識者会議でプログラミング的思考がコンピュテーショナル・シンキングを踏まえたものであると指摘されたものの，どのように踏まえたのかについては言及されていない（赤羽 2023, p.1）。林（2018）も同様に，プログラミング的思考とコンピュテーショナル・シンキングとの関係性について「有識者会議でコンピュテーショナル・シンキングに対する詳細な検討が行われたという開示記録はなく，プログラミング的思考という用語がどのような踏まえ方によって導出されたのかは明示されていない」（p.165）と問題を指摘している。

ここで，コンピュテーショナル・シンキングとは何かを確認する。今日展開されているコンピュテーショナル・シンキングをめぐる言説は Wing（2006）のエッセイから広がったものである（林 2018, p.166）。このエッセイについては，中島（2015）による日本語訳が存在するため，これを基に内容を確認していく。なお，この訳ではコンピュテーショナル・シンキングを「計算的思考」と表現している。以下の説明では，コンピュテーショナル・シンキングを計算的思考として表現する。

計算的思考について，Wing（2006）は以下のように説明している（中島 2015, p.584-585）。

1．計算論的思考は計算プロセスの能力と限界の上に成立しているもので，計算の主体が人間であるか機械であるかは問わない。
2．計算論的思考は，コンピュータ科学者だけではなく，すべての人にとって基本的な技術である。
3．計算論的思考は問題解決，システムのデザイン，そして基本的なコンピュータ科学の概念に基づく人間の理解などを必要とする。
4．計算論的思考とは再帰的に考えることであり，並列処理であり，命令をデータとし，データを命令とすることである。
5．計算論的思考とは巨大で複雑なタスクに挑戦したり，巨大で複雑なシステムをデザインしたりするときに，抽象化と分割統治を用いることである。
6．計算論的思考とは予防，防御，そして最悪のシナリオからの復帰という観点を持ち，そのために冗長性，故障封じ込め，誤り訂正などを用いることである。
7．計算論的思考はヒューリスティックな推論により解を発見することである。

Wing（2006）が提案したコンピュテーショナル・シンキングについて確認してきたが，コンピュテーショナル・シンキングに対する端的な定義の記述をしておらず，コンピュテーショナル・シンキングとは何であって何でないのかを列記した外延的措定をするに留まっている（林 2018, p.167）。

これらを実践レベルで考えるため，英国においてコンピュテーショナル・シンキングがどの

ように捉えられているのか，それに関連して情報教育のカリキュラムはどのように編成されているのかといったことを確認する。英国に着目した意図としては，①情報教育が進んでいること，②コンピュテーショナル・シンキングが情報教育の中核として設定されていることの２点が挙げられる。①について太田ら（2016）は，「英国は 1980 年のマイクロエレクトロニクス教育計画から世界の情報教育をリードしている」（p.198）と述べており，2013 年に従来の教科 ICT に代わる新しい教科である，コンピューティングのナショナルカリキュラムが告示され，2014 年から実施されたと説明している（太田ら 2016, p198）。ナショナルカリキュラムは，５歳から実施され年齢に応じて主題の内容が変化する。先に述べた通り，英国では幼児期・低学年段階から系統的な情報教育のカリキュラムが編成されている。②については，コンピューティングのナショナルカリキュラム（Department for Education 2013）において，その学習の目的として，「質の高いコンピューティング教育は，コンピュテーショナル・シンキングと創造性を使って世界を理解し，変えていくことができるようにする」とコンピュテーショナル・シンキングが明示されているのである。先に述べた通り，日本におけるプログラミング教育はプログラミング的思考の獲得を目指すが，それはコンピュテーショナル・シンキングを踏まえたものである。このことから，情報教育の中核にコンピュテーショナル・シンキングが存在するという点において英国と日本は共通していると言える。だが，情報教育が盛んであり低学年を対象としたものも存在しているといった点から，英国ははるかに先行していることは間違いない。

英国では，コンピュテーショナル・シンキングを，「複雑かつ雑然で，一部しか見えない現実問題を，知能のないコンピュータだけで問題に取り組めるような指示に変換する，いくつかの知的能力の集合」と定義している。コンピュテーショナル・シンキングの能力は，単にコンピュータを使った問題解決以外に人間だけで問題解決する場合にも応用できることを示している（太田ら 2016, p.198-199）。また，これらの能力の集合を論理的推論（Logical reasoning）として，抽象化（Abstraction），デコンポジション（Decomposition），アルゴリズム的思考（Algorithmic Thinking），評価（Evaluation），一般化（Generalization）の５つの能力としてブレークダウンしている（太田ら 2016, p.199）。

・抽象化：問題を単純化するため，重要な部分は残し，不要な詳細は削除する。
・デコンポジション：問題や事象をいくつかの部分に，理解や解決できるように分解する。
・アルゴリズム的思考：問題を解決するための明確な手順で，同様の問題に共通して利用できるものである。
・評価：アルゴリズム，システムや手順などの解決方法が正しいか，確認する過程である。
・一般化：類似性からパターンを見つけて，それを予測，規則の作成，問題解決に使用する。
（太田ら 2016，p.198）

続いて，コンピューティングのカリキュラムについて確認する。英国では教員向けの教材の提供や研修を実施している Computing At School（CAS）によって小中学生が学ぶべき情報教育の内容を学習項目ごとに段階的に示した Computing Progression Pathways が策定され，初等・中等段階で普及している（岩崎・掛下 2018，p.10-11）。これは学習の到達段階を８つの発達段階で定義したものであり，各学校の裁量に委ねられているものの，レベル１－５は

Grade 1 － 6，レベル 6 － 7 は Grade 7 － 9 レベル 8 は Grade10 － 11 で実施することを推奨している（太田ら 2016，p.199）。岩崎・掛下（2018）も同様にこの Computing Progression Pathways について「小学校ではレベル 1 － 5 が標準とされている」（p.11）と述べている。また，この Pathways の学習項目は，アルゴリズム，プログラミング・開発，データ・データ表現，ハードウェア・処理，コミュニケーション・ネットワーク，情報技術の 6 つであり，それぞれの到達目標は先に挙げたコンピュテーショナル・シンキングの 5 つの能力である抽象化，デコンポジション，アルゴリズム的思考，評価，一般化で分類されている（岩崎・掛下 2018，p11）。以下に，この Computing Progression Pathways の内容のうち，低学年が対象であるレベル 1 を記述する（表 3）。なお，コンピュテーショナル・シンキングの 5 つの能力の表記は，抽象化（AB），デコンポジション（DE），アルゴリズム的思考（AL），評価（EV），一般化（GE）となっている。

表 3　Computing Progression Pathways のレベル 1 到達目標

アルゴリズム	私は，アルゴリズムが何かを知っていて，記号を用いて簡単なアルゴリズムを表現できる（AL） 私は，コンピュータが正確な指示を必要とすることを理解している（AL） 私は，エラーを避けるために正確に実行することができる（AL）
プログラミング・開発	私は，ユーザーが独自のプログラムを書くことができることを理解している。また，テキストに頼らない環境（プログラム可能なロボットなど）で簡単なプログラムを作成することで，それを証明することができる（AL） 私は，プログラムの実行・チェック・変更ができる（AL） 私は，プログラムが正確な指示に基づいて実行することを理解している（AL）
データ・データ表現	私は，デジタル内容が多くの形式で表現されていることを理解している（AB）（GE） 私は，デジタル形式の違いを知っていて，かつ，デジタル形式によって情報伝達の方法が異なることを説明できる（AB）
ハードウェア・処理	私は，コンピュータには，プログラムを実行しないと，何もできないという側面があることを理解している（AL） 私は，デジタル装置上の全てのソフトウェアは，プログラムにより実行されていることを理解している（AL）（AB）（GE）
コミュニケーション・ネットワーク	私は，ウェブブラウザを用いて，ワールドワイドウェブからコンテンツを見付けることができる（AL） 私は，安全かつ，規律正しく，オンライン上でコミュニケーションする重要性と，個人情報を守る必要性を理解している（EV） 私は，コンテンツについて関心が生じたり，連絡したりしたい時に，何をしたらよいのかを理解している（AL）
情報技術	私は，指導者の指導の下で，ソフトウェアを使いながら，必要なファイルやフォルダ名に関するデジタルコンテンツを創造したり，保存・取り出しをしたりすることができる（AB）（GE）（DE） 私は，人々はコンピュータと相互作用し合うことを理解している 私は，校内にある技術の活用を共有できる。 私は，授業以外で，共通利用できる情報技術を理解している（GE） 私は，自分の役割を話したり，その役割を改善するために変化を加えたりすることができる（EV）

（磯部ら 2019，p184–185，Mark Dorling（2022）より作成）

Ⅳ　小学校低学年を対象としたコンピュテーショナル・シンキングに基づくプログラミング教育の実践の提案

本研究では，英国におけるコンピュテーショナル・シンキングの定義やComputing Progression Pathwaysの内容を踏まえ，小学校低学年を対象としたプログラミング教育の実践を提案する。

英国のカリキュラムを踏まえると，小学校高学年や中学校でプログラミングの活動を円滑に行うことができ，より高度なプログラミング教育の実践を行うという発達段階の見通しを念頭に置く必要がある。それを踏まえるならば，低学年であっても児童が実際にプログラミングを行う学習活動を実践として行うことが望ましい。そのような学習活動を行うにあたって，Computing Progression Pathwaysで該当する項目はアルゴリズム，プログラミング・開発の２点である（表３）。また，これらの到達目標はコンピュテーショナル・シンキングの５つの能力のうち，アルゴリズム的思考に関係するものである。そこで，実践の目標については，アルゴリズム，プログラミング・開発を設定し，獲得を目指すコンピュテーショナル・シンキングはアルゴリズム的思考とする。

次に，具体的な学習活動について構想していく。本実践の到達目標の中にはアルゴリズムに関する内容が含まれるが，アルゴリズムとは，人間が理解できる形で書かれた，問題を解く方法であり，プログラムとは，アルゴリズムを特定のプログラミング言語で実行したものである（Franks 2022）。本研究の実践については，小学校２年生を対象とするが，プログラミングについては教科等に位置づけられていないため，２時間で完結できるものとした。対象児童は，プログラミングの体験がなく，ICT機器の活用も積極的に行われていないことを想定している。実践の目標については以下のように設定した。

（１）問題解決のための手順を理解し，記号を用いて簡単な解決までの手順を表現できる。（知識及び技能）
（２）問題を解決するにあたって，解決までの手順を見いだしその方法について説明できる。（思考力，判断力，表現力等）
（３）問題解決のための手順を表現する活動に進んで関わり，プログラミングの体験を振り返り，プログラミングや問題解決の思考方法を生活や学習に活用しようとしている。（学びに向かう力，人間性等）

先に述べた通り，アルゴリズム的思考とは問題を解決するための明確な手順で，同様の問題に共通して利用できるものである。そのため，全２時間の授業で，児童が問題を解決するための手順を見つけ，それを様々なプログラミングの場面に応用し，問題を解決するまでの手順を考え，記号によってそれを表現できるようになることを目標とした。そして，評価基準については以下のように設定した（表４）。

表４　本実践における評価基準

知識・技能	思考力・判断力・表現力	主体的に学習に取り組む態度
1. 問題を解決するにあたってなぜその手順で解決できるのか理解している。 2. 命令の記号を用いて，単純な分岐のない問題解決の手順を作成できる。	1. 問題解決の手順を考え，自身の言葉や記号などを用いて論理的に表現している。	1. 問題解決の手順について考え，具体的な操作などを通して表現しようとしている。

続いて，本実践の具体的な活動内容とその意図について説明する。プログラミングに触れたことのない児童にとって，いきなりプログラミングを行うと戸惑うことが予想される。そのため，１時間目の導入場面において，児童がプログラミングに触れる前にアルゴリズムについて理解する時間を設定する。低年齢の学習者には物理的なリソースを使ってアルゴリズムを具体的に教える必要がある（Franks 2022）ため，アルゴリズムについて理解するための活動として，とりがぶたを捕まえるまでの手順を考える問題を解くというものを設定した。活動の際にはビジュアル型言語のブロックをイラストにし，これを単純な分岐のない命令を組み合わせた形式で提示する。また，この命令の組み合わせを提示する際に，順次のみではなく反復ブロックも含んだものも示す。

これは，児童が実際に取り組む Code.org の問題で用いる反復の命令ブロックについて，その意図を理解しやすいようにするといったねらいに沿ったものである。Code.org とは児童・生徒を主な対象としてプログラミング教育普及を目的にしているアメリカの NPO 団体であり，2013 年に設立され，ウェブサイトを通じて初心者向けのオンライン教材を提供している（榊原 2016，p.100）。反復ブロックの他に，提示する問題の内容とイラスト，命令ブロックの形は Code の問題と同じものとなるように作成した。これらを同様のものにした意図としては，児童が Code の問題を解く際に，解決に至る考え方は同様であることを示すためである。このような活動に取り組むことで，児童がアルゴリズムとは何かを理解できるようにする。

その後，実践者が Code の解説を行い，児童が Code の教材でプログラミングを体験する活動を設ける。Code を選択した理由としては，低学年を対象としたプログラミング教材が多数存在していること，他の低学年向けの教材と比較して容易であることの２点が挙げられる。今回は Code の教材のうち，Pre-reader Express（2021）の「めいろのプログラミング」を選択した。この教材は，とりに命令を出してぶたを捕まえるといった内容である（図１）。命令ブロックは上下左右のみであり，初めてプログラミングに触れる児童でも，十分にクリア可能だと考えられる。これらの活動を通して，児童が課題を解決するまでの手順を考え，表現できるようにする。

続く，２時間目には，Scratch を用いた課題に取り組む。課題の具体的な内容としては，命令が与えられず停止しているシマウマに対して，命令ブロックを組み合わせて，目的とする動きに近づけるというものである（図２）。この課題は，NHK for School にて公開されている「Why!？プログラミング」の「No. 1　壊れた魚を動かせ」という教材を参考にして，筆者が

作成したものである。これは，何の命令も与えられず停止している魚をプログラミングし，うまく動かそうという内容の教材である。この教材を参考にした理由としては，動きに関する問題が発生するため視覚的に分かりやすく，解決するまでの手順を考えるのに適していると予想されるためである。Scratch は Code と異なり，初心者ではスプライトを意図したように動かすのは難しい。「◯歩動かす」のブロックを「ずっと」のブロックの中に入れなければスプライトはすぐに止まってしまう。「もし端に着いたら，跳ね返る」というブロックを入れなければ，壁で跳ね返ることなくどこかへ行ってしまう。また，回転方法を指定するブロックを入れなければ，壁で跳ね返る際に上下が反転してしまう。このように，特定のブロックを抜くことで分かりやすく問題が発生するため，問題解決の手順を考えるのに適していると考えられる。また，Scratch は Code と比較してブロック数が多く，児童が混乱することが予想される。そのため，事前に使用するブロックを制限して取り組むことにした。また，この課題は Scratch に初めて触れる児童には難しいものであるため，事前に課題が難しいものであること，できなかったとしても問題のないことを強調して児童に伝えることにした。その後，命令ブロックが正しい順序で組み合わされているとりのブロックの順番を確認し，その内容を踏まえて再度プログラミングを行う。この活動が早く終わってしまった児童は,「ライオンを正しくうごかそう」という発展的な課題に取り組む。これは，使用する命令ブロックは先のシマウマ課題と同様なものであるが, 命令の順序が正しく設定されていないという課題である（図３）。この課題では，命令を正しく並び替えるとともに，先のシマウマ課題の内容を再度確認することを目的としている。

授業の最後には，アンケートを配布し，振り返りの時間を設ける。アンケートは「とりがぶたをつかまえるまでのじゅん番をかんがえることができましたか」,「とりにめいれいをだして，ぶたをつかまえることができましたか」,「シマウマをうごかすためのめいれいのじゅん番をかんがえることができましたか」,「シマウマをうまくうごかすことができましたか」の４つの質問に対して，よくできた◎，ほとんどできた○，あまりできなかった△の３段階で評価を行う欄，そして自由に感想を記述する欄を設けた。今回の実践では，アルゴリズム的思考を獲得することが目標であるため，１時間目と２時間目のいずれも問題が存在しており，それを解決していくといった学習活動の内容となっている。振り返りも, 問題を解決するまでの順番を考え，それをプログラミングすることができたかを確認するものとして設定した。

図１　めいろのプログラミング

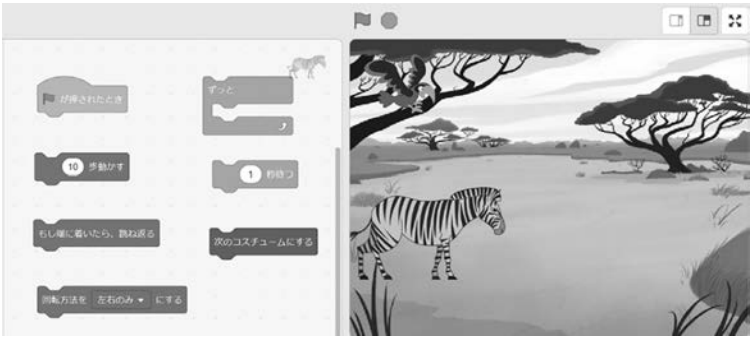

図２　シマウマをうごかそう

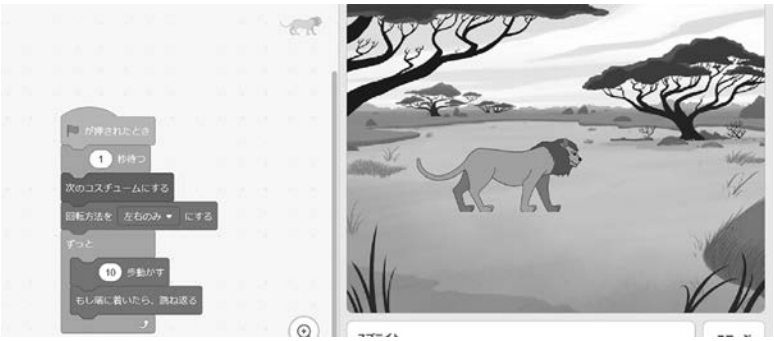

図３　ライオンを正しくうごかそう

Ⅴ　結果及び考察

１．小学校における実践について

本実践は，2023 年 12 月 14 日に X 市立 Y 小学校２年生の児童 25 名を対象として，２時間連続で行った。Y 小学校が短縮授業期間中であったため，それぞれ授業時間は 40 分であった。実践にあたって，大学生３名と大学院生１名が授業補助者を務めた。

２．コンピュテーショナル・シンキングに基づくプログラミング教育の実践の実際

（１）アルゴリズムについて理解する活動

児童はこちらの「ぶたを捕まえるために，とりはどちらに進めばよいのか」，「とりが何歩進めばぶたを捕まえられるか」といった問いかけに対して答えることができていた。また，この際に順次のみではなく，反復のブロックを含む命令の組み合わせも提示し，「左に進むという命令を何回繰り返せばぶたを捕まえられるかな」という問いかけも行ったが，こちらも同様に答えることができていた。

（２）Code「めいろのプログラミング」

問題が分からないという声は少なく，多くの児童が非常に早いペースで問題を進めていた。問題を解く際の傾向については，反復ブロックを使用せず順次の方法で問題を解く児童が多く見られた（図４）。そのため，活動の途中で全体に向けて反復ブロックを使用した方法についても試すように声掛けを行った。そのようにしたところ，反復ブロックの方法を試す児童や，反復ブロックを使用するとブロックの使用数が減り，プログラミングが楽になるといったことに気付く児童が見られた。授業計画の段階では「めいろのプログラミング」の９問目まで解き

終わることを想定していたが，想定していた問題を全て解き終わり，発展的な問題に取り組む児童が多く見られた（図５）。ただ，全ての児童が発展的な問題に取り組んでいたということはなく，１時間目が終了するまでとりがぶたを捕まえることを目指す「めいろのプログラミング」の問題に取り組む児童が１名いた。実践の目標である，問題解決の手順を考え，表現することができたかについては，観察した児童の様子からは達成できていたと考えられる。

（３）Scratch「シマウマをうごかそう」

活動の序盤はとにかく多くのブロックを繋げてシマウマがどのように動くのかを確かめる児童が見られた（図６）。多くのブロックを組み合わせて，並べ方を考える児童の他に，少数のブロックを組み合わせそれぞれのブロックの役割や，特定のブロックを抜いた組み合わせではどうなるのかを試す児童も存在した（図７）。また，児童間でブロックの組み合わせやブロックの役割などを話し合う場面も見られた。ただ，用意されたブロック以外の新たな命令ブロックを追加する，シマウマのスプライトの大きさを変更するといった，指定した課題以外のことに取り組む児童も存在した。このような場合の他に，問題が分からず手を止めてしまう児童も存在したが，試していないブロックの組み合わせはあるのか，それぞれのブロックの役割が分かったのかといった声掛けを行ったところ，問題に再度取り組んでいた。シマウマ課題の活動時間は 20 分程であったが，問題を解けていた児童は少数であった。

（４）Scratch「ライオンを正しくうごかそう」

ライオン課題については，シマウマ課題と同じブロックを使用して同じ組み合わせを目指すものであるため，先の課題と比較して，分からず手を止める児童は少なく，自分で解き方を考える姿が見られた。ただ，シマウマ課題と同様に本課題に取り組むことなく，用意されたブロック以外の新たな命令ブロックを追加する，ライオンのスプライトを変更するといった課題以外のことに取り組む児童も存在した。このライオン課題は，シマウマ課題と同じブロックを使用し，同じ命令の組み合わせを目指すという，シマウマ課題の内容を確認するという意図から設定された課題であるため，先のシマウマ課題の活動と比較して正しいブロックの組み合わせを見つけ，ライオンを正しく動かすことができた児童は多かった。

図４　順次の方法で解く児童

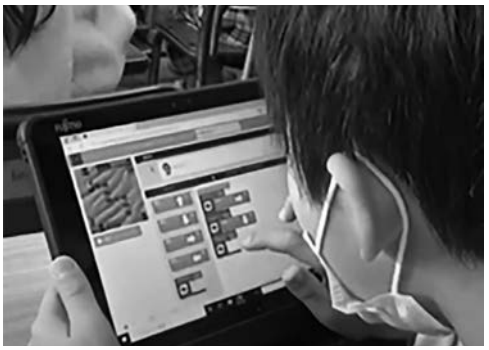

図５　発展的な問題に取り組む児童

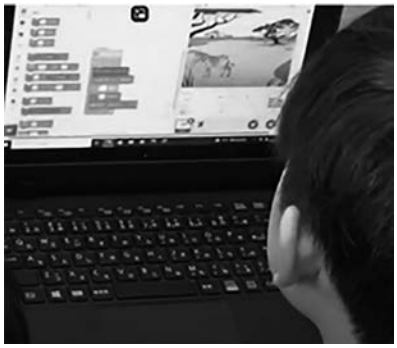

図６　ブロックを組み合わせる児童

図７　特定のブロックを抜く児童

３．小学校低学年を対象としたプログラミング教育実践の成果と課題

本実践の終了時に，児童に対して表５の４つの観点について３段階で自己評価を求めた。◎（よくできた）を３，○（ほとんどできた）を２，△（あまりできなかった）を１として項目ごとの平均値も記述した。

表５　本実践に関する振り返りアンケートの結果（N ＝ 25）

観点	3	2	1	平均
①とりがぶたをつかまえるまでのじゅん番をかんがえることができましたか	18	7	0	2.72
②とりにめいれいをだして，ぶたをつかまえることができましたか	22	3	0	2.88
③シマウマをうごかすためののめいれいのじゅん番をかんがえることができましたか	10	11	4	2.24
④シマウマをうまくうごかすことができましたか	12	6	7	2.20

観点②については平均値が 2.88 と最も高かった。これに対して，観点①も平均値が 2.72 と高かったが，先の観点より低い数値となった。これは，ブロックの順番を考えてから Code の問題を解くのではなく，問題を解きながらブロックの組み合わせを予想する，とにかく様々なブロックの組み合わせを試してみるといった児童が存在したためであると考えられる。２時間目のシマウマ課題については，観点③の平均値が 2.24 と，観点④の平均値 2.20 を上回っていた。これは，課題そのものの難度が高かったためシマウマをうまく動かすには至らなかったものの，動かし方を考えることはできた児童が多かったためであると考えられる。また，２時間目に関する２つの観点③④については，「あまりできなかった」と回答する児童の割合が高かった（観点③ 16%，観点④ 28%）。このように，２時間目に関するアンケートの結果からは児童間の個人差がより顕著に見られた。

アンケートの自由記述欄に記された児童の感想については，いずれも肯定的なものが多かった。特に，「活動が楽しかった」，「面白かった」といった感想が多く見られた。これは，感想の内容を指定せず自由に書くように指示したためだと考えられる。活動に関する感想について，２時間目のシマウマ課題を解けていた児童は少数であったが，楽しいと感じる児童の意見が見られた。これは，Scratch の特性として，特定のブロックが抜けているとスプライトがおかしな動きをしたり，壁に当たっても跳ね返らずどこかへ行ったりしてしまうという想定外の動きが見られるためであると考えられる。この理由の他にも，２時間目は１時間目と比較して，ス

プライトを変更したり，新たな命令ブロックを追加したりするといった，指定した課題以外の内容に取り組む児童もいたため，自由にスプライトを操作するといった行動に面白さを感じた児童もいたと予想される。その他の意見では，タブレットを使用することに楽しさを感じる意見も見られた。これは，対象児童の想定で述べた通り，タブレットを積極的に使用する学習状況ではなかったためであると考えられる。

ここまで，授業中の観察，振り返りアンケートの結果を確認した。最後に，これまでの内容を踏まえ，本実践の改善すべき点を述べる。まず，児童の個人差への対応が挙げられる。前述した授業中の観察において，１時間目の Code の問題では多くの児童が基礎的な問題をクリアし，発展的な問題に取り組んでいた。だが，１人の児童のみが１時間目が終了するまでとりがぶたを捕まえることを目指す問題に取り組んでおり，児童の個人差が見られた。また，２時間目もシマウマ課題を解けている児童は少数ではあるが存在した。課題が解けてしまった児童は，他の児童と話をする，ブロック内の数値を変更するといったように，こちらが指定した学習活動以外のことを行っていた。アンケートの結果からも，観点④については「よくできた」が 12 名と「あまりできなかった」が７名と他の観点に比べて個人差が浮き彫りになっていた。このように，問題を解くペースが早い児童，遅い児童のいずれも学習活動を行うことができるような課題設定と実践の方法を検討する必要がある。

この他の改善すべき点として，評価の方法が挙げられる。今回の実践では，評価の方法としてアンケートを選択した。だが，今回の実践のベースとした Computing Progression Pathways のアルゴリズムとプログラミング・開発の項目には，実際にプログラムを作成するような内容が多く含まれている。そのことを踏まえると，児童が何らかの制作物を作成し，それを基に評価を行うといった評価方法が適切であったと考えられる。アンケート形式の評価方法と比較して，制作物を作成して提出か共有するといった評価方法では，児童がどのように問題解決の道筋を考え，表現したのかといった部分が明確になると考えられる。また，２時間目のシマウマ課題，ライオン課題の際にスプライトを変更する，ブロックを追加するといった児童の存在を挙げたが，制作物を作成するといった評価方法であれば，そのような児童の行為も，学習活動に繋がる可能性がある。

今回の実践では，以上の２点が改善すべき点として明確に表れた。

Ⅵ　おわりに

本研究では小学校低学年を対象として，コンピュテーショナル・シンキングに基づくプログラミング教育の実践を提案することを目的として論を進めてきた。２時間の実践を構想し，実施したところ，授業中の観察からは問題解決の手順を考え，表現する児童の姿が見られた。だが，振り返りアンケートの結果や授業中の児童の行動から，児童の個人差への対応，問題解決の手順の理解度を確認するための評価方法といった改善すべき点も明らかとなった。

プログラミング教育を実施する際には，その実践で何を目標とするのかといった，獲得する能力の明確化が必要である。また，実際に ICT 機器を活用したプログラミング教育は低学年の段階では積極的に行われてはいない。だが，本論で取り上げた英国のナショナルカリキュラムのように，低年齢の段階から実施することで将来のより高度な実践へと繋がると考えられる

ため，様々な形での実践が展開されていくことが必要であるだろう。

謝辞

実践に協力して頂いたＸ市立Ｙ小学校校長並びに２年生クラス担任と25名の児童の皆さん，本実践と準備のサポートをして頂いた大学生と大学院生の皆さんに謹んで謝意を表します。

参考文献

赤羽泰（2023）小学校プログラミング教育における課題とその能力の測定方法の開発．教育デザイン研究，14：1–2

Department for Education, UK（2013）National curriculum in England: Computing Programmes of Study The Statutory Programmes of Study and Attainment Targets for Computing at Key Stages 1 to 4. https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study（2024 年 3 月 15 日最終閲覧）

Dorling，M（2022）CAS Computing Progression Pathways KS1（Y1）to KS3（Y9）by Topic.https://www.computingatschool.org.uk/resources/2014/january/cas-computing-progression-pathways-ks1-y1-to-ks3-y9-by-topic（2024 年 3 月 15 日最終閲覧）

Franks，R（2022）Algorithms Top Tips.https://www.computingatschool.org.uk/resources/2022/february/algorithms-top-tips（2024 年 3 月 15 日最終閲覧）

磯部征尊・大森康正・岡島佑介・川原田康文・上野朝大・山﨑恭平・山崎貞登（2019）初等中等教育段階のコンピューティング／プログラミング教育の目標と学習到達水準に関する日米イングランドの比較研究．上越教育大学研究紀要，39：179–193

岩崎誠・掛下哲郎（2018）CAS ガイドラインに基づく小学校プログラミング教材の提案 :mBot ロボットとアルゴリズム学習用ツールキットの活用．情報教育シンポジウム論文集，2：9-16

文部科学省（2016）小学校段階における論理的思考力や創造性，問題解決能力等の育成とプログラミング教育に関する有識者会議（第３回）議事録.
https://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/gijiroku/1382219.htm（2024 年 3 月 15 日最終閲覧）

文部科学省（2017）小学校学習指導要領.

文部科学省（2020）小学校プログラミング教育の手引（第三版）.
https://www.mext.go.jp/content/20200218-mxt_jogai02-100003171_002.pdf

文部科学省・総務省・経済産業省（2017）小学校を中心としたプログラミング教育ポータル.
https://miraino-manabi.mext.go.jp/（2024 年 3 月 15 日最終閲覧）

太田剛・森本容介・加藤浩（2016）諸外国のプログラミング教育を含む情報教育カリキュラムに関する調査－英国，オーストラリア，米国を中心として－日本教育工学会論文誌，40（3）：197-208

林向達（2018）Computational Thinking に関する言説の動向．日本教育工学会研究報告，16：165-172

榊原直樹（2017）App Lab を用いた初心者向けプログラミング授業の実践－文系大学生を対象としたプログラミング学習の導入について－．清泉女学院大学人間学部研究紀要，14：99-104

Wing, J.M.（2006）Computational Thinking, Communications of the ACM, 49, No.3: 33-35（訳：中島秀之（2015）計算論的思考．情報処理，56：584-587）

山田秀哉・小野田千明（2018）小学校 2 年生におけるプログラミング教育の実践報告．日本デジタル教科書学会 発表予稿集：61–62