

## 小学校1年生におけるプログラミング授業の実践

メタデータ	<p>言語: Japanese</p> <p>出版者: 静岡大学教育学部附属教育実践総合センター</p> <p>公開日: 2018-02-28</p> <p>キーワード (Ja): 小学校, 1年生, プログラミング, プログラミング的思考, Scratch</p> <p>キーワード (En):</p> <p>作成者: 山崎, 智志, 室伏, 春樹, 紅林, 秀治</p> <p>メールアドレス:</p> <p>所属:</p>
URL	<a href="https://doi.org/10.14945/00024675">https://doi.org/10.14945/00024675</a>

# 小学校 1 年生におけるプログラミング授業の実践

## Implementation of Programming Lessons for the First grade in Elementary School

山崎 智志 \*  
Satoshi YAMAZAKI

室伏 春樹 †  
Haruki MUROFUSHI

紅林 秀治 †  
Shuji KUREBAYASHI

### 概要

プログラミング学習を通して、小学校 1 年生に「プログラミング的思考」を育成することは可能なのかを確認することを目的に授業を行った。プログラミングソフトは、Scratch を使用した。授業の様子と授業後のアンケートをクロス集計し分析した。その結果、プログラミングの学習は、「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み合わせが必要か」や「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」ということを児童に考えることを可能にすることがわかった。また、小学校 1 年生のプログラミング学習では、「わかった」という実感よりも「楽しい」と実感させることが「またやりたい」という気持ちを高めていることもわかった。さらに、角度の知識がなくても試行錯誤的な作業により対象物の回転の角度とパラメータの関係把握したり、条件分岐の考え方を理解できたりする可能性も示唆され、小学校 1 年生の段階でも「プログラミング的思考」が育成される可能性が高いことがわかった。

キーワード：小学校，1 年生，プログラミング，プログラミング的思考，Scratch

## 1 はじめに

プログラミング教育とは、「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』などを育成するもの」とされている<sup>1)</sup>。ここでいう「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み合わせが必要であり、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、記号の組み合わ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」とされている<sup>1)</sup>。

学習指導要領（2017 年公開）では、小学校段階において、この「プログラミング的思考」の育成に力を入れる内容となった<sup>2)</sup>。技術的な内容は一部にとどめるとされているが、実施例<sup>1)</sup>には「プログラミングを体験しながらそのよさに気付く学びを取り入れていくことなどが考えられる」と記されており、プログラミングを体験させることによる効果は期待できる。

現在日本では、中学校技術・家庭科（技術分野）と高等学校情報科においてプログラミング教育が行われている

<sup>3)4)</sup> が、小学校においては、ほとんど行われていない<sup>5)</sup>。しかし、諸外国においては、既にプログラミング教育を含む情報教育の低年齢化が進んでおり、英国では小中学校（5～16 歳）でのプログラミング教育を 2014 年から必修としている<sup>6)</sup>。そのため、日本においても、小学校 1 年生からプログラミング教育を実践できる可能性は十分にある。加えて、小学校 1 年生でプログラミング的思考が育成できることが確認できるならば、小学校の全学年でも同様なことが言えるのではないかと考えた。

そこで、本研究では、小学校 1 年生がプログラミング学習を通して、「プログラミング的思考」を育成できるのかを確認するため、米国マサチューセッツ工科大学 (MIT) が開発した Scratch（スクラッチ）を使用し、小学校 1 年生を対象にプログラミング学習の授業を実践した。

## 2 授業の題材について

### 2.1 小学 1 年時におけるプログラミング教育

学習指導要領には、プログラミング教育についての系統的な指導や発達段階に応じた習得規準などが現状において示されていない。

一方、大森らは小学校段階におけるプログラミング教育を Computational Thinking の基礎を修得することを核とした教育カリキュラムを提案している<sup>7)</sup>。また、ベネッセコーポレーションはプログラミングで育成する資質・能

\* 掛川市立和田岡小学校  
† 静岡大学

力の評価規準の試行を行っており、低学年における到達目標を公開している<sup>8)</sup>。

大森の対案する教育カリキュラムでは、小学校段階を3つのステージに分けており、小学1年時においては擬似プログラミング言語を用いたアンプラグド教材を用いることを提案している。アンプラグド教材とは、コンピュータを利用せずにコンピュータの動作原理やプログラミング的思考を学習するものである<sup>9)</sup>。大森らの提案では、複数の色を組み合わせることで表現された命令(擬似プログラミング言語)をロボットに読み取らせる体験を提案しており、コンピュータにおける逐次処理などの基本処理、分解、評価(デバッグ)、トライ&エラー、複数解の容認を学習内容としている。

ベネッセコーポレーションが公開している評価規準では、プログラミング的思考を「動きに分ける(分割)」「記号にする(抽象化)」「一連の活動にする(一般化)」「組み合わせる」「振り返る」「論理的に考えを進める」と整理している。たとえば大森らの基本処理や分解は「動きに分ける」「論理的に考えを進める」の内容、評価やトライ&エラーは「振り返る」の内容につながることから、プログラミング教育に対する共通性が存在しているといえる。

しかし、大森らの提案は教具としてロボットが必要であり、多くの小学校で実践するには費用面に課題がある。また、ベネッセコーポレーションの評価規準はあくまで試行版であり、今後の実証が待たれている。そこで、小学校で既に配備されているPCを利用したプログラミング教育の実践を実施し、プログラミングの思考が育成できるか検証を行うこととした。

## 2.2 ビジュアルプログラミング言語

ビジュアルプログラミング言語は、グラフィカルな図形やアニメーションを用いてプログラムを記述する言語である。視覚的にプログラミング可能なため、子供向けに開発されたものが多い<sup>10)</sup>。前述のScratchがこれにあたるが、他にもSqueakやVISCUITなどがある<sup>10)</sup>。

## 2.3 プログラミングソフト「Scratch」

Scratchの実行画面を図1に示す。森らはScratchについて以下のように分析している<sup>5)</sup>。

ScratchはMITメディアラボが開発した教育用プログラミングソフトである。インターネット上で無償配布されている。ブロック型コマンドを組み合わせることで、視覚的にプログラミングできる。コマンドは、多言語に対応しており、日本語でのプログラミングも可能である。

また、国内において小学生を対象とした関連書籍が複数販売<sup>11)12)</sup>されており、ブロックに記載される言語も全て平仮名表記が可能である。そこで本実践においては、利用するブロックをスクリプトの「動き」のカテゴリ内にある「○歩動かす」「○度回す」に限定することで小学校1年生においても実践が可能であると考えた。

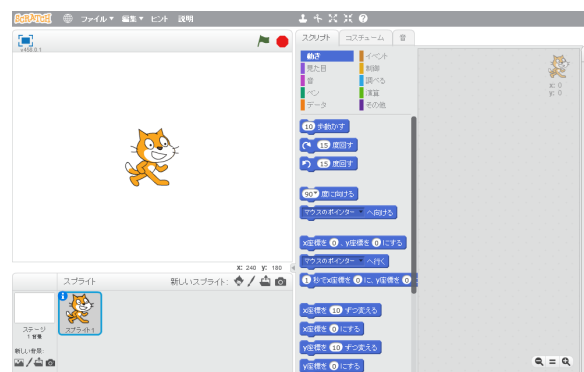


図1 Scratchの実行画面

## 3 授業実践

### 3.1 授業の概要

本実践は、掛川市立和田岡小学校1年1組28名(男子14名、女子14名)を対象として、2016年9月から10月までの期間、生活科の授業で行った。授業は、同校のPC室(児童用35台、教師用1台のタブレットPC(OS:Windows10)を設置)で、山崎(学級担任:T1)と支援員(T2)の2名で行った。

### 3.2 児童の実態

授業を行う前に、1年生の児童(28名)が、家でPCに触れる機会がどの程度あるのか、調査を行った。質問は、「1. 家にPCがあるか」「2. 家でPCを使ったことがあるか」「3. (使ったことがある児童は)どのような使い方をしたか」の3項目で行った。質問1と2の回答をクロス集計した結果を表1、質問3の回答結果を表2に示す。

表1 質問1「家にPCがあるか」と質問2「家でPCを使ったことがあるか」の回答(N=28)

		使ったことが	
		ある	ない
家	ある	13人(46%)	10人(36%)
	ない	0人(0%)	5人(18%)

表1より、「家にある」と回答した児童は23人(82%)であった。しかし、使ったことがある児童は全体の13人

表2 質問3「(使ったことがある児童は)どのような使い方をしたか」の回答 (N=28)

回答内容	人数 (%)
ゲーム	5人 (38 %)
パズル	2人 (15 %)
お絵かき	1人 (8 %)
文字入力	1人 (8 %)
動画	1人 (8 %)
忘れてしまった・未回答	3人 (23 %)

(46 %) で、使用したことのない児童が半数以上いることが確認できた。また、表2より、PCの使い方については、ゲームやパズル、お絵かきソフトが8人(61 %)であることがわかった。以上より、対象とする児童は、プログラミングの経験がないことがわかった。

### 3.3 授業計画

授業では、「プログラミング学習を通して、『プログラミング的思考』を育成する」ことを目標として、授業を行った。具体的には、複数のコマンドを使用しないと解決できない課題を設定した。表3に、授業計画を示す。

プログラミング経験が全くないという児童の実態や、1年生という発達段階を踏まえ、まずは、マウス操作やキーボード入力(かな)の練習から授業を行った。PCの基本操作を学習した後、Scratchを用いたプログラミングの学習を行った。その際にも、まずは基礎的な学習を行い、その後、課題解決的な学習に取り組んだ。

表3のNo.1の授業では、PCを使用するのが初めてという児童もいたため、PCの基本的な使い方から授業を始めた。

No.2の授業では、本校のPCには、スズキ教育ソフトの「キューブきっず」<sup>14)</sup>と呼ばれる教育用ソフトがインストールされているため、その中にある「マウスレッスン」を用いて、「クリック」や「ダブルクリック」などを練習した。

No.3の授業では、プログラミングの学習の際に、キーボードを用いた数値入力を行うため、Microsoft社製ワープロソフト「Word」を用いて、数字の入力と「かな」入力でのひらがなの入力を行い、キーボード入力の練習を行った。

No.4のScratchを用いたプログラミングの学習では、第5時の学習で、「はじめに」で述べた「プログラミング的思考」(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み合わせが必要であり、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、記号の組み合わ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)の育成を目指した授業を行った。具体的には、「スクラッチキャットをステージの左上から右下に動かす」という課題(図2)を教師側から提示し、その活動を実現するために、動きの組み合わせを考えさせた。その際、児童が動きの組み合わせを考え、改善することができるよう、第1時~第4時において、Scratchの基本的な扱い方を学習した。まず、スクリプトの「動き」のブロックパレットにある「○歩動かす」「○度回す」「もし端に着いたら跳ね返る」というブロックをスクリプトエリアに動かし、「スクラッチキャット」と呼ばれるスプライトを動かすプログラムを作る学習を行った。次に、「イベント」のブロックパレットにある「旗がクリックされたとき」と「□キーが押されたとき」というブロックをスクリプトエリアに動かし、マウスやキーボードに反応するプログラムを作る学習を行った。これらをNo.4の第1時~第4時に学習し、前述の第5時につなげていった。

No.5の授業では、プログラムと自分たちの生活を結び付け、プログラムを身近なものとして感じさせたいと考えた。

なお、各授業の後半には、「キューブきっず」にある「お絵かき」を用いて、マウスやタッチペンを使用して絵を描きながら、PCに親しむ時間を設定した。



図2 授業の課題

### 3.4 PCの活用

一人一台PCを使用し、授業を進めた。授業の様子を図3に示す。表3のNo.1~No.3の授業では、最初のうちはPCの使用に慣れていない児童がいたが、授業を重ねていくうちに、マウスやキーボードの扱いに慣れていく様子が見られた。マウスの活用については、ダブルクリックに

表3 授業計画（13 時間）

No.	学習内容	時間
1	電源の ON・OFF の仕方等、PC の基本的な使い方を学習する。	1
2	マウスの使い方を学習する。	3
3	キーボードの使い方を学習する。	3
4	【第1時～第4時】（Scratch の基礎的学習） 「○歩動かす」「○度回す」「もし端に着いたら跳ね返る」というブロックの活用方法。 スプライトの動かし方。 「旗がクリックされたとき」「□キーが押されたとき」の活用方法。 【第5時】（「プログラミング的思考」の育成を目指した学習） スクラッチキャットをステージの左上から右下に動かす」という課題に取り組む。	5
5	身近にあるものとプログラムとのつながりを考える。	1

苦戦する児童が多くいたが、個別に対応し、練習を繰り返すうちにできるようになっていった。キーボードを用いた文字入力については、キーボードの文字を探すのに時間がかかる児童はいたが、すでにひらがなを学習した1年生であれば、「かな」入力は十分に可能であった。数字の入力についても理解は早く、表3のNo.4からのプログラミングの学習にもいかされていた。



図3 授業の様子

## 4 授業実践の結果

### 4.1 プログラミング学習の様子

表3のNo.4では、授業計画に基づき、Scratch を用いたプログラミング学習を行った。「○歩動かす」「○度回す」というブロックをスクリプトエリアに動かすことは、理解して、実際にできていた児童が多かった。しかし、「○」に当てはまる数値を適切に入力できず、プログラムを実行すると、「スクラッチキャット」がほとんど動かなかったり、動きすぎて画面からいなくなってしまうことが出てきた。画面からいなくなってしまうことへの対策と

しては、「もし端に着いたら跳ね返る」というブロックを用いて対応した。プログラムの実行については、「旗がクリックされたとき」と「□キーが押されたとき」というブロックを用いた。プログラムを作成する際に、まず一番上に「旗がクリックされたとき」を移動するようにしたので、活用できていた児童が多かった。後半「□キーが押されたとき」の活用の際には、スペースキーを設定したが、あまり活用する児童は見られなかった。

表3のNo.4の第5時には、「スクラッチキャットをステージの左上から右下に動かす」プログラムの作成を行った。この課題を解決するためには、「○度回す」と「○歩動かす」のブロックを組み合わせ、適切な値の入力が必要になる。実際に、児童は、既習事項の「旗がクリックされたとき」「○歩動かす」と「○度回す」、そして「もし端に着いたら跳ね返る」を用いて、プログラムを作成していた児童が多かった。図4に授業で生徒が作成したプログラムを示す。



図4 児童が作成したプログラムの一例

事前に学習した際に、「○歩動かす」→「○度回す」の順でブロックを組み合わせていたため、本時においても、その順で組み合わせる児童が多かった。図4(左)では、1回の実行で左上から右下へ動かすことが難しいため、何度か実行を繰り返してステージの左上から右下へ移動させてい



た。一方で、図 4(右)の「〇度回す」を先にしてプログラムを作成する児童も見られた。このような児童は、一度の実行で動きを表現することができていた。

児童の数値入力に関しては、動きを予測して適切な数値を入力するのではなく、動きを見ながら手探りで数値を変えていた児童がほとんどであった。前述したように、座標や角度の知識が乏しいため、あまり意味をもたない数値を入力している児童も、多く見られた。

プログラミングにおいて意図する活動を実現するためのプログラムは何通りもあるため、自分のプログラムが完成した後に、互いのプログラムを見合ったり、教え合ったり、それを参考に作り直したりする姿が見られた。目指すもの(答え・目標など)は一つであっても、それにたどり着く手段は様々であるプログラミング学習の良さが出た場面であった。また、教え合う際には、国語や算数の授業で活躍する児童とは異なる児童が教え役になる様子が見られた。図 5 に授業の様子を示す。



図 5 互いのプログラムを見合う様子

## 4.2 授業後のアンケート

授業後に、アンケートによる調査を行った。質問内容を以下に示す。

質問 1 PC の授業は楽しかったか。

質問 2 プログラミングの授業は楽しかったか。

質問 3 プログラミングの授業でどんなことが楽しかったか。

質問 4 プログラミングの授業はわかったか。

質問 5 プログラミングの授業でどんなことがわかったか。

質問 6 プログラミングの授業をまたやりたいか。

質問 1・2・4・6 に関しては、3 段階(3:思う(肯定的回答) 2:どちらとも言えない(中立的回答) 1:思わない(否定的回答))で回答するよう指示した。質問 3

に関しては、「スクラッチキャットを動かす」「動きを変えられる」「何度もやり直せる」「いろいろな方法がある」「命令を組み合わせる」の 5 項目の中から、楽しかったことを挙げさせた。質問 5 に関しては、授業で主に用いた「旗をクリックされたとき」「〇歩動かす」「〇度回す」「もし端に着いたら跳ね返る」「□キーが押されたとき」の 5 つのブロックの使い方がわかったかを、それぞれ回答させた。

## 4.3 アンケートの結果

質問 1・2・4・6 の回答結果を表 4 に示す。

表 4 質問 1・2・4・6 に対する回答 (N=28)

	肯定	中立	否定
質問 1	25 人 (89 %)	1 人 (4 %)	2 人 (7 %)
質問 2	16 人 (57 %)	7 人 (25 %)	5 人 (18 %)
質問 4	8 人 (29 %)	16 人 (57 %)	4 人 (14 %)
質問 6	18 人 (64 %)	4 人 (14 %)	6 人 (21 %)

質問 1「PC の授業は楽しかったか」に関しては、PC を活用することの楽しさを感じている児童は 9 割近くいるが、質問 2「プログラミングの授業は楽しかったか」に関しては、肯定的回答が 6 割弱にとどまっていることがわかる。また、「楽しくなかった」と感じている児童も質問 1 に比べて 10 % 程増えていることから、お絵かきやマウスレッスンなどに比べ、プログラミングは楽しさを感じられなくなっていることがわかる。

次に、質問 3「プログラミングの授業でどんなことが楽しかったか」に関して回答内容ごとに分類した。分類した結果を、表 5 に示す。

表 5 質問 3「プログラミングの授業でどんなことが楽しかったか」に対する回答 (N=28)

回答内容	人数 (%)
スクラッチキャットを動かす	16 人 (57 %)
動きを変えられる	23 人 (82 %)
何度もやり直せる	12 人 (43 %)
いろいろな方法がある	19 人 (68 %)
命令を組み合わせる	15 人 (54 %)

質問 3 では、「動きを変えられる」「いろいろな方法がある」ことは楽しいと感じている児童が 6 割を超えているが、それ以外は 6 割未満となっていた。次に、質問 1 と質問 2 の関連性を明らかにするためにクロス集計をした。集計結果を表 6 に示す。また、質問 1 と質問 2 の回答の類似度をコサイン類似度<sup>15)</sup>の値で示した。

質問 1 で PC の授業は「楽しい」と回答した児童のうち、11 人 (39 %) がプログラミングの授業では「中立(どち

表6 PCとプログラミングの授業：質問1と質問2の関連性 (N=28, 類似度 0.91)

		質問2 プログラミングが楽しい		
		肯定	中立	否定
質問1 PCが楽しい	肯定	14人 (50%)	7人 (25%)	4人 (14%)
	中立	1人 (4%)	0 (0%)	0 (0%)
	否定	1人 (4%)	0 (0%)	1人 (4%)

らともいえない)」「否定(楽しいとは思わない)」と回答した。これは、プログラミング学習を通して、楽しさを失う児童が4割近くいたことを示している。その一方で、質問1で「否定(楽しいとは思わない)」と2人の児童が回答したが、質問2では、「肯定(楽しかった)」と回答した児童が1人(4%)いた。この結果から、プログラミングを通して、楽しさを感じることができた児童もいることがわかった。

次に、質問5「プログラミングの授業でどんなことがわかったか」の回答結果を表7に示す。

表7 質問5「プログラミングの授業でどんなことがわかったか」に対する回答 (N=28)

回答内容	人数 (%)
旗をクリックされたとき	21人 (75%)
○歩動かす	17人 (61%)
○度回す	25人 (89%)
もし端に着いたら跳ね返る	23人 (82%)
□キーが押されたとき	17人 (61%)

表7より、児童がプログラミング授業を通してわかったと回答した内容として、「○度回す」「もし端に着いたら跳ね返る」と書いた児童が80%を超えていた。「○度回す」に関しては、算数の授業で角度の学習をしていない児童でも、入力した数値により回転角度が変わるという関係性がわかったことを意味している。さらに、「もし端に着いたら跳ね返る」に関しては、実行画面上の対象物の動きに限定したものであるが、プログラムの条件分岐に関する理解を示唆している。

質問4「プログラミングの授業はわかったか」では、「肯定(授業がわかった)」と回答した児童は3割弱(表4)であるが、わかった内容を聞いた質問5では、回答はブロックごとの命令内容を記述していたが、どの内容も6割を超えている。この結果から、それぞれのブロックの活用につ

いてはわかったと感じてはいても、プログラム学習全体として考えると、肯定的回答(わかった)ができない児童が多かったことがわかった。次に、質問2と質問4の関連性を明らかにするために、クロス集計した結果を表8に示す。

表8 プログラミングの授業：質問2と質問4の関連性 (N=28, 類似度 0.78)

		質問4 わかった		
		肯定	中立	否定
質問2 楽しい	肯定	7人 (25%)	8人 (29%)	1人 (4%)
	中立	0 (0%)	5人 (18%)	2人 (7%)
	否定	1人 (4%)	3人 (11%)	1人 (4%)

質問4で「中立(どちらとも言えない)」と回答した児童(16人(表4))のうち、5割(8人)の児童が質問2では、「楽しかった」と回答していたことが表8よりわかった。この結果から「肯定(わかった)」とは言えなくとも「楽しかった」と感じることができていた児童がいたことがわかった。また、表4より、質問2と4に否定的回答をした児童は、質問2では5人(18%)、質問4では4人(14%)であったが、両方の質問に対して「否定(楽しくなかった)(わからなかった)」と回答した児童は、1人(4%)であったことが、表8よりわかった。これは、内容理解がともなっていないでもプログラミングの授業を楽しんでいる児童が3割程度いたことを示すと同時に、プログラミングの学習は、児童にとって興味・関心が高い授業になることを示唆している。

ところが、表4より、質問6「プログラミングの授業をまたやりたいか」に関しては、6割を超える児童が「肯定(またやりたい)」と回答したが、2割を超える児童が「否定(もうやりたくない)」と回答している。そこで、質問6と質問2(楽しかった)と質問4(わかった)との関連性を明らかにするために、それぞれクロス集計をした。表9と表10にそれぞれの集計結果を示す。

表9より、質問2(楽しい)と質問6(またやりたい)の回答は類似度が0.98と高いことから、児童にとって「楽しい」ということと「またやりたい」ということはほぼ同じであると言える。

表10より質問4で「中立(どちらとも言えない)」と回答した児童(16人)のうち、5割以上の児童(9人)が質問6で「肯定(またやりたい)」と回答した。この結果

表9 質問2と質問6の関連性 (N=28, 類似度 0.98)

		質問6 またやりたい		
		肯定	中立	否定
質問2 楽しい	肯定	13人 (46%)	2人 (7%)	1人 (4%)
	中立	4人 (14%)	2人 (7%)	1人 (4%)
	否定	1人 (4%)	0	4人 (14%)

表10 質問4と質問6の関連性 (N=28, 類似度 0.78)

		質問6 またやりたい		
		肯定	中立	否定
質問4 わかった	肯定	8人 (29%)	0	0
	中立	9人 (32%)	3人 (11%)	4人 (14%)
	否定	1人 (4%)	1人 (4%)	2人 (7%)

は、プログラムの理解が伴っていないくても、「またやりたい」と思っている児童が多いと言え、表8が示す結果と同様に、プログラミング学習は理解が伴っていないくても「またやりたい」「楽しい」という思いを持たせる学習であると言える。

## 5 考察

授業の様子から、課題解決的なプログラミング学習の授業においては、「プログラミング的思考」にある「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み合わせが必要」かや、「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」ということを考えていた児童が多く、小学1年生であっても「プログラミング的思考」を獲得する可能性があることがわかった。これらはベネッセコーポレーションの評価基準で示されたプログラミング的思考における「組み合わせる」や「記号にする（抽象化）」といった目標につながるものであり、PCを利用したプログラミング的思考の習得が可能であることを示唆している。また、プログラムに設定する数（パラメータ）の大小に伴って対象の動く距離が変化することを確認できるため、数理的な処理の基礎知識を獲得することができると思われる。

一方で、「記号の組み合わ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」ということに関しては、授業者が求める結果を得ることができなかった。その理由として、プログラムを実行する際に、試行錯誤的に何度か実行を繰り返して、ステージの左上から右下へ対象物（図2）を移動させていた児童がほとんどで、1回の実行で左上から右下へ動かすようなプログラムを前もって考えるというように、根拠をもとに考えるところまで学習展開がなされていなかったことが考えられる。根拠を基に論理的に思考する学習は、発達段階的に考えても小学1年生では難しいため、試行錯誤的な操作を中心とした学習展開になることは仕方ないことであるが、授業の中で「実行は1回」のような制限を課題に設けたり、参考となるプログラム例示したりすることで、ブロックの組み合わせやブロック内の数値を見直しやすくなり、根拠を基に考える学習活動に発展できるのではないかと考えられる。

アンケート結果から、PCの授業よりもプログラミングの授業の方が「楽しくない」と感じている児童が多かった。これは、授業内容が「お絵かき」や「マウスレッスン」等の操作中心の作業的な学習に比べ、プログラミングの学習は操作を伴うものの思った通りの結果すぐ得られないばかりか、思考を伴う作業が求められる学習であることが原因であると考えられる。ところが、小学校におけるプログラミング学習においては、「情報手段に慣れ親しむ」<sup>16)</sup>ことが重要であり、「またやりたい」という意欲を継続させることが求められる。「またやりたい」と感じる要素としては、「わかった」かどうかよりも、「楽しかった」かどうかの影響していることが表10よりわかった。そのような結果を考慮して、「もうやりたくない」という思いを減らすための手立てを、授業の中でさらに工夫していく必要がある。たとえば、配布ファイルに移動目標（ゴール）となるスプライトを準備しておくことで移動順序を考えさせたり、移動の軌跡を表示させて絵を描かせ移動に目的を持たせたりする等の方法である。

また、表7より、児童がわかったと感じた内容の中で「〇度回す」は80%を超えていた。さらに、「〇歩動かす」に関しては60%を超えていた。Scratchの指導においては、「〇歩動かす」については座標の知識が、「〇度回す」については角度の知識があると考えやすいことは当然であるが、小学1年生の児童には、そこまでの知識を求めることはできない。それが、動きを予測して適切な数値を入力するのではなく、授業内で試行錯誤的に、動きを見ながら手探りで数値を変えていた児童がほとんどで、あまり意味をもたない数値を入力している児童も見られた原因である。しかし、多くの児童が「楽しい」と感じながら、これらの作



業を試行錯誤的でも行っていたということは、座標や角度の知識がない児童でも、感覚的にこれらの知識を理解できる可能性を示唆している。プログラミング学習は、算数や数学で習う知識がなくても、試行錯誤的に算数や数学で学ぶ知識を把握していく学習になる可能性があると言える。

同じく表 7 より、児童がわかった内容として「もし端についたら跳ね返る」も 80 %を超えていた。これは、条件分岐の考え方であり、プログラミング的な思考の重要な要素の一つである。たとえ、Scratch で示されているブロックの使い方がわかったという程度であっても、小学 1 年生の段階で条件分岐的な内容が把握できることを示唆している。条件分岐の考え方を把握できるならば、プログラムを実行する前に対象物の動作を予測したり、動作の不具合が生じ時に根拠をもって考えることができるようになると期待できる。なぜならば、移動距離や角度だけで動作を規定した場合、PC 画面上の目測が頼りになるため、プログラミングの確認作業がパラメータの微調整が中心となり、作業そのものが試行錯誤的なものになってしまうが、条件分岐が入ることで、プログラムの確認作業に対象物の動作が一定の条件をもとに代わるというような論理的な思考が加わるからである。本授業実践でだけでは、条件分岐を把握している程度が、論理的な作業を可能にする転位可能な汎用性ある知識にまで定着しているものとは考え難いが、今後詳しくそれに関して検証していく必要がある。

## 6 まとめ

本実践を通して、小学校 1 年生において、プログラミング言語 Scratch を用いてプログラミング学習を行うことで、プログラミングの楽しさを感じたり、実際にプログラムを作成したりできることを確認できた。また、基礎から段階を踏んで学べるような授業計画を組み、課題解決的なプログラミング学習をおこなうことで、小学校 1 年生でも「プログラミング的思考」を育成していくことは可能であるとわかった。さらに、小学校 1 年生に対するプログラミングの学習では「わかった」よりも「楽しい」を重視した学習が重要であることもわかった。

加えて、本実践において、意図する活動を実現するために、プログラムを修正したり、改善したりした活動は、「主体的な学びの過程」と言え、仲間と互いのプログラムを見合ったり、教え合ったりした活動は、「対話的な学びの過程」と言える。このことから、プログラミング学習には、学習指導要領において重視されている「主体的・対話的な深い学び」の実現の可能性が大いにあると言えることがわかった。

今後は、学習に対する児童の思いをより深く検証し、「プログラミング的思考」の育成はもちろん、児童の関心や意欲を継続させる授業計画や手立てについての研究を深めていきたい。また、「プログラミング的思考」やプログラミングの特性に関する理解を深め、各教科における小学校 6 年間の指導計画とその先の中学校・高等学校を含めた体系的なプログラミング学習および情報教育について検討していきたい。さらに、「主体的・対話的な深い学び」の実現に向けて、深い学びにつなげるための手立てについても、研究を進めていきたい。

## 参考文献

- [1] 中央教育審議会：次期学習指導要領等に向けたこれまでの審議のまとめ、別紙 3-2「小学校段階におけるプログラミング教育の在り方について（議論の取りまとめ）」（2016）
- [2] 中央教育審議会：次期学習指導要領等に向けたこれまでの審議のまとめ、別紙 3-1「情報活用能力を構成する資質・能力」（2016）
- [3] 文部科学省：中学校学習指導要領解説 技術・家庭編，教育図書，pp.32-37，（2008）
- [4] 文部科学省：高等学校学習指導要領解説 情報編，開隆堂，（2010）
- [5] 森 秀樹，杉澤 学，張 海，前迫 孝憲：Scratch を用いた小学校プログラミング授業の実践，日本教育工学会論文誌 34(4)，pp.387-394（2011）
- [6] 植松 恵理子：小学校にプログラミングがやってきた！，三省堂，pp.91-92（2016）
- [7] 大森康正ほか 4 名：小型ロボットを用いた小学生向けプログラミング教育教材の開発とその活用方法，日本産業技術教育学会，第 32 回情報分科会（上越），研究発表会講演論文集，pp.29-32（2017）
- [8] ベネッセコーポレーション，プログラミングで育成する資質・能力の評価規順（2017/5/27 版），<https://beneprog.com/terms>
- [9] 利根川裕太，佐藤智：先生のための小学校プログラミング教育がよくわかる本，翔泳社，p.126（2017）
- [10] 深谷 和義，宮地 晶子：小学生向けプログラミング授業のための「プログラミン」利用の検討，日本教育工学会論文誌 36(Suppl.)，pp.9-12（2012）
- [11] 阿部和広ほか 3 名：親子で楽しみながら考える力，つくる力，伝える力を育もう！小学生からはじめるわくわくプログラミング，日経 BP 社，（2013）
- [12] 阿部和広ほか 4 名：親子で楽しみながら考える力，つ

くる力，伝える力を育もう！小学生からはじめるわく  
わくプログラミング 2，日経 BP 社，(2016)

- [13] 文部科学省：幼稚園教育要領，小・中学校学習指導要領等の改訂のポイント (2017)
- [14] スズキ教育ソフト「キューブきっず」：<http://www.suzukisoft.co.jp/products/cubekids6/>(2017年4月16日確認)
- [15] 吉田光男，山本幹雄：教師情報を必要としないニュースページ群からのコンテンツ自動抽出，日本データベース学会論文誌，Vol.8，No.1，pp.1-6(2009)
- [16] 文部科学省：小学校学習指導要領 第1章総則第4の2の(9)，p.16 (2008)