

〈ICT教育：小学校 総合的な学習の時間〉

「プログラミング的思考」の育成を目指す学習指導の工夫

—総合的な学習の時間における段階的なプログラミング体験を通して—

うるま市立彩橋小学校教諭 高江洲 義 邦

I テーマ設定の理由

令和2年4月に全面実施を迎える小学校学習指導要領（平成29年告示）において、プログラミング教育が導入される。その背景には情報化の進展による社会や人々の生活の変化があり、将来の予測が難しい社会において、子供たちが将来どのような職業に就くとしても、普遍的に求められる力として「プログラミング的思考」を育むことが必要とされている。これを受けて各小学校では、児童がプログラミングを体験しながら「プログラミング的思考」を身に付けるための学習活動を各教科等において計画的に実施することが求められているのである。

文部科学省は、プログラミング教育の必修化に向けて「プログラミング教育の手引き」を公開し、プログラミング教育の導入の背景や意義について説明している。また、総務省や経済産業省と連携して「未来の学びコンソーシアム」を設立し「小学校を中心としたプログラミング教育ポータル」を通じて実践事例を発信したりするなど、プログラミング教育の全面実施に向けた支援策も講じている。そこでは、各教科等における実践事例が紹介されており、全国各地の小学校に共有されることで、プログラミング教育の円滑な実施が期待されている。

前年度、本校第4学年（当時）の児童を対象に、プログラミング教育の視点を取り入れた授業を実施した。先行事例の多いCSアンプラグド（コンピュータを使わないプログラミング教育）の手法を活用し、算数科と体育科で授業を行い「プログラミング的思考」の育成を試みた。算数科ではわり算の筆算の手順を、体育科では走り高跳びの動きを、一つ一つ細分化して付箋紙に書かせた後、どの順序で並べればうまく実行できるかを考えさせ、うまく実行できない場合は、どのように改善すればよいかを考えさせる活動を行った。それらの授業を通して、課題解決に向けて順序よく考えたり、うまくいかない場合には問題点を見つけ、その問題がある部分を修正していこうとする児童の姿が見られるようになった。これらの児童の姿に、プログラミング教育の効果を実感した。そして、各教科等におけるプログラミング教育は、従来行われてきた学習の中にも実践の可能性を見出すことができ、また、CSアンプラグドの手法を用いることによって導入しやすくなると考えられる。

プログラミング教育の実施について、小学校学習指導要領では、児童がプログラミングを体験することが求められている。また「プログラミング教育の手引き」には「プログラミング教育全体において児童がコンピュータをほとんど用いないということは望ましくないことに留意する」と明記されている。これらのことから、CSアンプラグドの手法のみに止まらず、コンピュータを用いたプログラミングも重視されていることが分かる。しかし現在、コンピュータを用いたプログラミングの学習は、実践事例が少ない。そのため、どの教材でどのような授業を行えば、児童に効果的に「プログラミング的思考」を身に付けさせることができるかを検証する必要があると考える。

そこで本研究では、総合的な学習の時間の「情報」の学習に、コンピュータを用いたプログラミングの体験学習を取り入れる。単発的ではなく、段階的にプログラミングを体験することで、児童が効果的に「プログラミング的思考」を身に付けることができると考え、本テーマを設定した。

〈研究仮説〉

第4学年総合的な学習の時間において、チュートリアル・ドリル型から自由型へと段階的なプログラミング体験をさせることによって、効果的に児童の「プログラミング的思考」を育成することができるであろう。

Ⅱ 研究内容

1 実態調査

(1) 対象 うるま市立彩橋小学校 4年1組17名

(2) 方法 アンケートによる児童のプログラミングの学習及び「プログラミング的思考」に関する意識調査

(3) アンケートの結果と考察

質問項目に対して、児童は「とてもあてはまる」「あてはまる」「あてはまらない」「まったくあてはまらない」の中から1つを選択し回答している。なお、調査を実施した時点において、プログラミングを体験したことのある児童は一人もいなかった。

① プログラミングの学習に対する意識

「これからプログラミングを学びたいですか？」という質問に対し、「とてもあてはまる」「あてはまる」と肯定的に答えている児童は、17名中12名(71%)だった(図1)。それらの児童のうち10名は、パソコンの操作が好きであり、得意であるという児童だった。それに対し、「あてはまらない」「まったくあてはまらない」と否定的に答えている児童は5名(29%)おり、そのうち、パソコンの操作に関する意識として、「パソコンを操作することが好きではなく、得意ではない」という児童が2名、「パソコンを操作することは好きであるが、得意ではない」という児童が2名、「パソコンを操作することが好きで、得意である」という児童が1名いた。

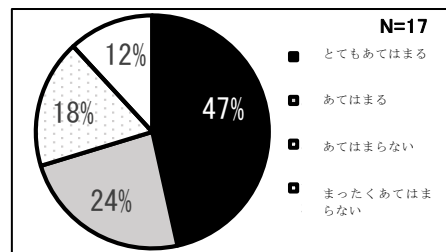


図1 プログラミングを学びたい

これらの結果から、本学級の児童は、17名中15名(88%)がパソコンの操作をすることが好きであり、そのうち操作が得意であると答えた児童の多くがプログラミングの学習に対して肯定的であること、操作が得意ではないと答えた児童のほとんどがプログラミングの学習に対して消極的な意識を持っていると考えられる。

以上のことを踏まえると、コンピュータの操作が難しかったり、複雑で手間がかかるようなソフトウェアを使用すると、操作を得意としていない児童の学習意欲を低下させてしまいかねない。また、操作の指導に終始してしまうことも予想され、本来の目的である、「プログラミング的思考」を育成することにつながらない可能性があるため、教材を選ぶ視点として、操作がシンプルで容易なものであることに留意したい。

以上のことを踏まえると、コンピュータの操作が難しかったり、複雑で手間がかかるようなソフトウェアを使用すると、操作を得意としていない児童の学習意欲を低下させてしまいかねない。また、操作の指導に終始してしまうことも予想され、本来の目的である、「プログラミング的思考」を育成することにつながらない可能性があるため、教材を選ぶ視点として、操作がシンプルで容易なものであることに留意したい。

② 「プログラミング的思考」に関する意識

調査

児童が普段の授業で問題を解いたり、何かを書いたり作ったりする際、「手順を考えて取り組んでいるか」、「間違えた際にどう対処しているか」等、児童の「プログラミング的思考」に関わる意識を知るための質問を行った。

「順序を考えながら問題を解いている」、「順序を考えながら書いたり作ったりしている」児童は、どちらも17名中14名(82%)、そうでない児童は3名(18%)いることが明らかになった。このことから全体的に、本学級では、順序を考えながら物事に取り組む児童が多いと考えられる。また、間違えたり失敗したとき、「どこを直せばいいか探している」児童は、17名中14名(82%)であるが「成功するまでチャレンジできる」と答えた児童は、17名中13名(76%)だった。順序よく処理することや、間違えた際にどこを改善すればいいか探したり、成功するまで挑戦できる児童が全体的に多いことがうかがえる。本研究におけるプログラミングの体験では、児童は順序よ

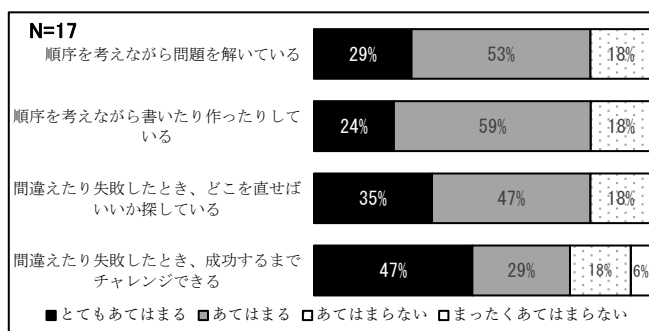


図2 「プログラミング的思考」に関する意識調査

「順序を考えながら問題を解いている」、「順序を考えながら書いたり作ったりしている」児童は、どちらも17名中14名(82%)、そうでない児童は3名(18%)いることが明らかになった。このことから全体的に、本学級では、順序を考えながら物事に取り組む児童が多いと考えられる。また、間違えたり失敗したとき、「どこを直せばいいか探している」児童は、17名中14名(82%)であるが「成功するまでチャレンジできる」と答えた児童は、17名中13名(76%)だった。順序よく処理することや、間違えた際にどこを改善すればいいか探したり、成功するまで挑戦できる児童が全体的に多いことがうかがえる。本研究におけるプログラミングの体験では、児童は順序よ

くブロックを組み合わせたり、試行錯誤しながら、自分の意図した動きをコンピュータに実行させる。この活動を通すことで、より多くの児童の意識を高めていきたい。

2 理論研究

(1) 小学校におけるプログラミング教育について

文部科学省は、平成28年5月に「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」（以下、「有識者会議」という）を設置し、小学校段階におけるプログラミング教育の内容やねらい、必要性等について議論を行った。有識者会議の議論の取りまとめ（平成28年6月16日公表）によると、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』などを育むことであり、コーディング（専門的なプログラミング言語で記述すること）を覚えることが目的ではない」と定義されている。このことから、小学校におけるプログラミング教育は、一部のプログラミングの専門家を養成するためではなく、すべての児童に「プログラミング的思考」を育むために行われるものである。

平成29年3月に告示された小学校学習指導要領（以下、「新学習指導要領」という）では、第1章総則の第3教育課程の実施と学習評価の1(3)に「情報活用能力の育成を図るため（中略）あわせて、各教科等の特質に応じて、次の学習活動を計画的に実施すること。ア（略）イ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」と記された。新学習指導要領では、有識者会議における「プログラミング的思考」の趣旨に対応させ、「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させるための学習活動」を行うことが求められている。そして、その学習活動を「各教科等の特質に応じて計画的に実施すること」としたのである。

(2) プログラミング教育を通じて目指す育成すべき資質・能力

「小学校プログラミング教育の手引き」では、新学習指導要領に沿って、小学校におけるプログラミング教育のねらいが整理されている。新学習指導要領における三つの柱、①何を理解しているか、何ができるか（知識・技能）、②理解していること、できることをどう使うか（思考力・判断力・表現力）、③どのように社会・世界と関わり、よりよい人生を送るか（学びに向かう力、人間性等）で整理すると、プログラミング教育を通じて、次のような資質・能力を育成するとされている。

- ①身近な生活でコンピュータが活用されていることや、問題の解決には必要な手順があることに気づくこと。（知識・技能）
- ②発達の段階に即して、「プログラミング的思考」を育むこと。（思考力・判断力・表現力）
- ③発達の段階に即して、コンピュータの働きを、よりよい人生や社会づくりに生かそうとする態度を涵養すること。（学びに向かう力、人間性等）

(3) 「プログラミング的思考」について

有識者会議の議論の取りまとめにおいて、「プログラミング的思考」は、急速な技術革新の中で、プログラミングや情報技術の在り方がどのように変化しても、普遍的に求められる力であると考えられている。具体的な定義としては、「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」とされている。

「プログラミング的思考」を身に付けることは、情報技術が人間の生活にますます身近なものとなる中で、それらのサービスを享受するだけではなく、その働きを理解して、自分が設定

した目的のために使いこなし、よりよい人生や社会づくりに生かしていくために必要なことであると考えられている。そのため「プログラミング的思考」は、プログラミングに携わる職業を目指す子供たちのみならず、どのような進路を選択しどのような職業に就くとしても、これからの時代を生きるすべての子供たちにとって共通に必要なとされる力であると言える。

(4) 先行研究における「プログラミング的思考」についてのまとめ

東京都教職員研修センターによる先行研究「児童の情報活用能力の育成ー 小学校段階におけるプログラミング教育の推進を通してー」（平成31年3月）では、児童の「プログラミング的思考」を育むことに着目した理論研究及び授業開発を行っている。その研究の一部において、「プログラミング的思考」の定義を分割することでイメージを持ちやすくし、さらにそれらを児童にも教師にとっても理解しやすくなるという意図で、児童にも伝わるような平易な言葉に置き換えて以下の表1のように示している。

表1 プログラミング的思考の定義の役割と対応する平易な言葉（例）（東京都教職員研修センター）

	プログラミング的思考の定義 (東京都教職員研修センター教育開発課による分割)	対応する平易な言葉（例）
①	自分が意図する一連の活動を実現するために、	「ながれをみとおす」
②	どのような動きの組合せが必要であり、	「うごきにわかる」「うごきをあつめる」
③	一つ一つの動きに対応した記号を、	「おきかえる」
④	どのように組み合わせたらいいのか、	「くみあわせる」
⑤	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、	「かいぜんする」
⑥	といったことを論理的に考えていく力	「かんがえる」「ひょうげんする」

本研究では、これらの対応する平易な言葉を「プログラミング的思考」を育む授業づくり及び活動内容の設定の際に活用すると共に、実践後の検証をする際の視点として取り入れる。

(5) プログラミング教材について

現在、コンピュータを用いたプログラミングの教材は、続々と新しいものが登場している。しかし実際に授業を行う際、どの教材を使えばよいのか迷ってしまう教師も少なくないだろう。利根川裕太（2017）は、ソフトウェア型の教材を表2のように2つの軸で分け、計4種類に分類した上で、児童の発達段階に応じた進め方について考えを示している。

表2 ソフトウェア型教材の分類（利根川）

		課題設定方式	
		チュートリアル・ドリル型	自由
プログラミング方式	ビジュアル（ブロック）	・ Hour of Code ・ プログル など	・ Scratch ・ Viscuit など
	テキスト	・ CodeMonkey ・ CodeCombat など	・ JavaScript ・ Python など

利根川によると、ソフトウェア型教材は1つ目の軸に見られるように、ブロックを組み合わせでプログラムしていく「ビジュアル」タイプと、言語を入力することでプログラムしていく「テキスト」タイプの二つの方式に分類される。「ビジュアル」タイプは、タイピングや文法による間違いがなく、「順次」「繰り返し」「分岐」といったプログラミングの考え方の学習に集中しやすいという良さがある。対する「テキスト」タイプは、関数の種類が豊富であり、一般的なプログラミング言語のようにテキストを入力することで、より高度なことができるという特徴がある。しかし、タイピングの能力が必要とされることから、すぐに実践することは難しい。また、小学校段階でのプログラミング教育においては、プログラミング言語を覚えたり、プログラミングの技能を習得することがねらいではないため、取り扱うことは少ないと考えられる。

2つ目の軸では、課題設定の形式が分類されている。「チュートリアル・ドリル型」は、コンピュータの画面上で出される課題を、画面上の指示に従って解いていく形式である。そのため、プログラミングの初心者でも比較的指導が容易であること、また、同時に多人数の指導が可能であるという点に良さがある。対して、自由型は数多くのブロックから自由に選び、好きなように並べたり組み合わせたりすることで様々な動きを実現できるという特徴がある。そのため、表現的な活動においては、自由型のほうが馴染みやすいと考えられる。利根川は、①ビジュアル

ル×チュートリアル・ドリル型→②ビジュアル×自由型→③テキスト×チュートリアル・ドリル型→④テキスト×自由型と進むのが、児童の発達段階に沿った進め方であると述べている。

本研究では、コンピュータに慣れていないという本学級の児童の実態を踏まえ、「ビジュアル」タイプの教材である「Hour of Code」と「Scratch」を選択する。進め方は、利根川の考えをもとに、全10時間のうち3時間をチュートリアル型の「Hour of Code」、その後の7時間を自由型の「Scratch」で進めていく。「Hour of Code」は、Web上にあるプログラミング学習サイトであり、40を超えるプログラミングのアクティビティがある。その中から、ここでは「古典的な迷路」というアクティビティを選択する。このアクティビティでは、画面に表示された課題について、自分のキャラクターを指定されたプログラムのブロックを使って動かし、全20問の課題をクリアする。その中で児童にプログラミングの基本となる「順次」「繰り返し」「分岐」を段階的に体験させることができる。また、ブロックを組み合わせる際に、「ドラッグ&ドロップ」等のコンピュータの基本的な操作に慣れさせることもできる。



図3 「Hour of Code」『古典的な迷



図4 「Scratch」

「Hour of Code」で、プログラミングの基本的な考え方と操作に慣れた後、児童は、「Scratch」でプログラミングを行う。「Scratch」は、プログラムのブロックを組み合わせながら、自分が意図する動きをキャラクターに指示することができるという点が「Hour of Code」と共通しているため、移行がスムーズに行えると予想される。「Hour of code」ではコンピュータから課題が提示されるのに対し、「Scratch」では課題が提示されることはない。また、使用できるブロックが多く、限定されていないため、初めのうちはどのブロックを使えばよいか迷う児童も出てくると思われる。「プログラミング的思考」を育むという観点から、組み合わせ方や改善策を考える活動に時間をかけたい。そのため使用するブロックは、児童らと共にどれが必要かを話し合いながら選んでいく。その後、児童は1人1台コンピュータを使用して、意図する動きを実現させるためのプログラムを考えていく。「Scratch」は、自由度の高さから、児童の創造力や問題解決能力を育むこともできると期待できるため、最後の授業では、完成した作品に児童が自分なりに工夫してプログラムをアレンジする活動を取り入れていきたい。

(6) 段階的なプログラミングの体験

本学級の児童の実態を踏まえると、すぐに自由な動きをプログラムさせることは難しいと思われる。効果的に「プログラミング的思考」を身に付けさせるため、プログラミングの体験を段階的に進めていく。ここでいう段階的とは、全10時間の授業の中でプログラムを積み重ねながら作成し、難易度を徐々に高めていくことである。始めの3時間で「Hour of code」を活用してプログラミングの考え方とコンピュータの基本的な操作を身に付け、第4時から「Scratch」を活用したゲーム作りを行う。「Scratch」の体験では、児童は1時間の授業ごとに作成したプログラムをファイルに保存し、そのファイルを次時の学習へと繋げていく。具体的には、第4時にネコが画面上を動き回るプログラムを作成し、第5時はネズミがマウスのポインターに向かって動くプログラム、そして第6時には、ネズミがネコに触れると動きが止まるプログラムを作成するという流れである。計3時間の授業で、ネコとネズミの追いかけっこゲームが完成する。その後さらにプログラムを徐々に加え、最終的にはシューティングゲームの完成を目指す。児童は、この段階的なプログラミングの体験の中で、意図する動きを実現させるためのプログラムの順序や組み合わせ方を考え、試行錯誤を繰り返しながら、効果的に「プログラミング的思考」を身に付けることができると考える。

Ⅲ 指導の実際

1 単元名「コンピュータでゲームを作ろう」

2 単元目標

プログラミングの体験を通して、コンピュータがプログラムによって動いていることを理解し、プログラミング的な見方や考え方を生活に生かそうとする態度を育てる。

3 評価規準

関心・意欲・態度	思考・判断・表現	知識・理解
コンピュータに関心を持ち、プログラミング的な考え方を自分の生活にも生かそうとする。	コンピュータに自分が意図する動きをさせるための記号の並べ方や組み合わせ方を考える。	コンピュータがプログラムによって動いていることを理解する。

4 学習指導計画

時	学習内容	体験するプログラミング	活用するツール ☆「プログラミング的思考」
1	迷路の問題をクリアしよう① 「順次」	課題をクリアするためのプログラムの組み合わせを考え、実行する。 	○「Hour of code」(古典的な迷路①～⑤) ☆「ながれをみとおす」「おきかえる」「くみあわせる」「かいぜんする」
2	迷路の問題をクリアしよう② 「繰り返し」	課題をクリアするためのプログラムの組み合わせを考え、実行する。 	○「Hour of code」(古典的な迷路⑥～⑬) ☆「ながれをみとおす」「おきかえる」「くみあわせる」「かいぜんする」
3	迷路の問題をクリアしよう③ 「分岐」	課題をクリアするためのプログラムの組み合わせを考え、実行する。 	○「Hour of code」(古典的な迷路⑭～⑳) ☆「ながれをみとおす」「おきかえる」「くみあわせる」「かいぜんする」
4	ネコを動き回るようにしよう	ネコのキャラクターに自分が意図する動きをプログラムする。 	○「Scratch」 ☆「うごきをあつめる」「くみあわせる」「おきかえる」「かいぜんする」
5	おいかけっこゲームを作ろう	キャラクターを追加し、マウスの矢印についてくるようにプログラムする。 	○「Scratch」 ☆「うごきをあつめる」「くみあわせる」「おきかえる」「かいぜんする」
6	おいかけっこゲームを作ろう	ネズミがネコに触れると動きが止まるようにプログラムする。 	○「Scratch」 ☆「うごきをあつめる」「くみあわせる」「おきかえる」「かいぜんする」
7	シューティングゲームを作ろう (玉のプログラムを考えよう①)	玉のイラストを描き、ネズミの手元についてくるようにプログラムする。 	○「Scratch」 ☆「うごきをあつめる」「くみあわせる」「おきかえる」「かいぜんする」
8	シューティングゲームを作ろう (玉のプログラムを考えよう②)	玉がネコに向かって移動するようにプログラムする。 	○「Scratch」 ☆「うごきをあつめる」「くみあわせる」「おきかえる」「かいぜんする」
9	シューティングゲームを作ろう (ネコのプログラムを考えよう)	ネコが玉に当たったら消えるようにプログラムする。 	○「Scratch」 ☆「うごきをあつめる」「くみあわせる」「おきかえる」「かいぜんする」
10 (本時)	シューティングゲームを作ろう (シューティングゲームを改造しよう)	ゲームをより面白くするための動きを考え、プログラムする。	○「Scratch」 ☆「ながれをみとおす」「うごきをあつめる」「くみあわせる」「おきかえる」「かいぜんする」「ひょうげんする」

5 本時の学習について

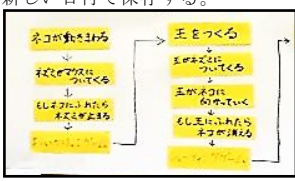
(1) 目標

「順次」「繰り返し」「分岐」の考え方を活用し、自分の意図する動きをプログラムする。

(2) 指導の工夫

これまでに少しずつ積み重ねながら作成してきたプログラムが繋がっていることを意識させ、ブロックを追加したり数値を変えたりすることで、動きを多様に変化できることに気づかせる。

(3) 本時の展開 (10/10)

	学習内容	指導の留意点 (☆プログラミング的思考)
導入 (10分)	○前時に作成したファイルを開き、新しい日付で保存する。 ○これまで学習してきたことの繋がを確認する。  ○前時までに作成したプログラムを確認する。 ○めあてを確認する。 めあて：ゲームを面白くするプログラムを考え、紹介しよう。	◇前時に作成したファイルと区別するため、確実に新しい名前 で保存させるようにする。 ◇これまでの学習内容を正しい順序でフローチャートに表しな がら、学習の繋がりを意識させる。(☆ながれをみとおす) ◇前時までに作成したプログラムのお手本を提示し、動きを確 認させる。  お手本のプログラム
展開 (25分)	○どうすればゲームが面白くなるか考える。 ○考えた動きをさせるためのプログラムを作成する。 間違いを探している様子  ○作成したゲームをペアで遊びながら、プログラムを紹介し合う。 ペアの作品で遊ぶ様子 	◇児童からアイディアを出させる。 (☆「うごきをあつめる」) ◇ブロックを追加したり、数を変えたりすると、動きに変化が 出ることを想起させる。 (☆「くみあわせる」「おきかえる」「かいぜんする」) ◇話し手には、面白くしたところを中心に紹介させる。聞き手 には、お互いの工夫している点に着目しながら聞くようにさせ る。(☆「ひょうげんする」) ◇ゲームをやってみて、面白いと思ったところや工夫されてい ると思ったところを伝えさせる。
まとめ (10分)	まとめ：プログラムをつけ足したり、数を変えたりすると、ゲームを面白くすることができる。 ○振り返りシートに感想等を記入する。	◇これまでの体験を通してわかったことや、これから学んでみ たいことを中心に記入させる。

6 仮説の検証

研究仮説に基づき、段階的なプログラミングの体験が、児童の「プログラミング的思考」の育成に有効であったか、アンケート結果と回答の理由及び作成したプログラムの変容から検証する。

(1) プログラミングの学習に対する意識

実践後のアンケートで「これからもプログラミングを学びたいですか」という質問に対し、「とてもあてはまる」「あてはまる」と答えた児童の割合は、実践前の71%から100%へと上昇した(図5)。実践前は、「あてはまらない」「まったくあてはまらない」と答えていた児童も、全員が今後もプログラミングを学びたいと思うようになったことが分かる。

実践前のアンケートにおいて、「あてはまらない」「まったくあてはまらない」と答えていた児童4名(1名は無記入)の理由(図6)を見ると、どの児童もプログラミングに対する意識が肯定的なものへと変容していることが分かる。また、全ての学習を終えての感想(図7)からは、どの児童も楽しく取り組むことができたということがうかがえる。

A児は実践前、「パソコンを操作することが好きで、得意である」が、プログラミングを学びたいかという質問に対しては、「あてはまらない」と答えていた児童である。今後もプログラミングを学びたい理由の記述を見ると、コーディングによるプログラムをすると予想し、難しそうであるという理由から学習に対する意欲

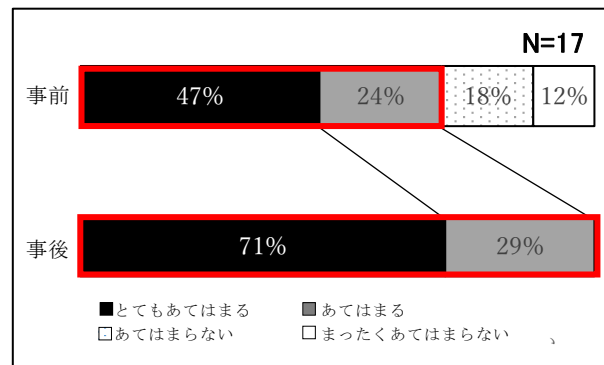


図5 プログラミングを学びたいですか

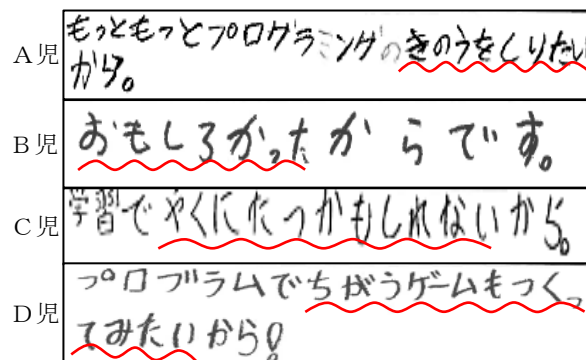


図6 今後もプログラミングを学びたい理由

は低かった。しかし、「ビジュアル」タイプでブロックを組み合わせることによるプログラミングであることを理解すると、自分なりの順序でブロックを組み合わせても同じ動きをさせられるということに喜びを見出していた。また、数値を変えることで動きを遅くしたり速くしたりできることを誰よりも早く気づいていた。今後はさらにプログラミングの機能を知りたいという意欲も出てきている。

B児とE児は、「パソコンを操作することが好きではなく、得意ではない」と答えていた。二人は、自分で考えたり、考えたことを記述することに苦手意識を持っている児童である。実践の序盤では、コンピュータの操作やブロックの組合せ方に困難を示し、教師を呼んで確認することが多かったが、後半からプログラムのお手本を提示することによって、それを見ながら自力で取り組む姿が見られるようになった。一つ一つのブロックを丁寧に組み合わせながら、意図する動きをさせられたことを喜ぶ姿も見られるようになった。アンケートの記述からは、言葉は少ないが、自分なりに楽しみながらプログラミングの学習に取り組むことができたのではないかと考えられる。

C児とD児は、「パソコンを操作することは好きであるが、得意ではない」と答え、プログラミングを学びたいかという質問には、「まったくあてはまらない」と答えていた児童である。C児は感想にも見られるように、初めの頃は楽しくないと感じていたようであった。しかし図9のプログラムの変容に見られるように、授業を重ねていくごとに使うブロックの数が増え、それに伴って徐々に高度な動きを実現させることができた。また、数値を変えることで動く速さを変えられることにも気づくことができた。実践後には、プログラミングの考え方が他の学習にも役立てることができるかもしれないと、活用しようとする姿勢がうかがえた。C児と同様に他の児童にも、ブロックを増やしたり数値を変えたりすることで実行できる動きが多様になっていくことに、驚いたり喜んだりしている姿が見られるようになった。

(2) 児童の「プログラミング的思考」の変容

児童が作成したプログラムを「プログラミング的思考」の定義（平易な言葉）に照らし合わせて、どのような力が使われていたか、その変容を考察していく。

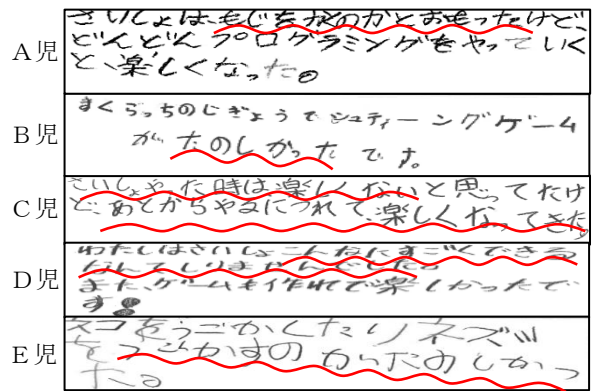


図7 プログラミング学習の感想①

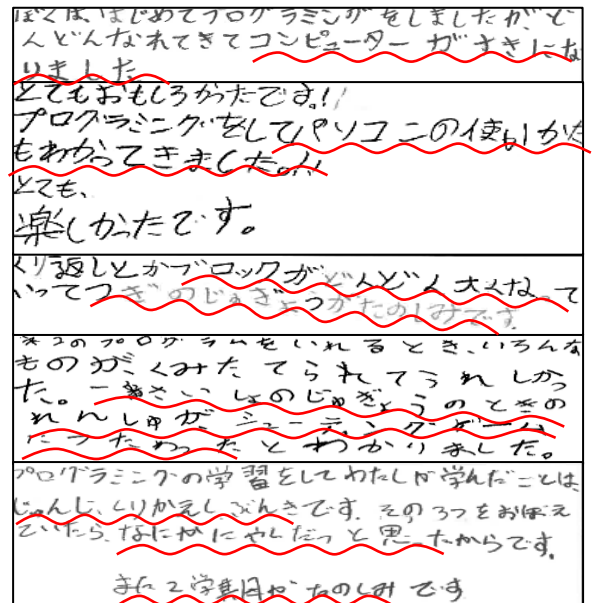


図8 プログラミング学習の感想②

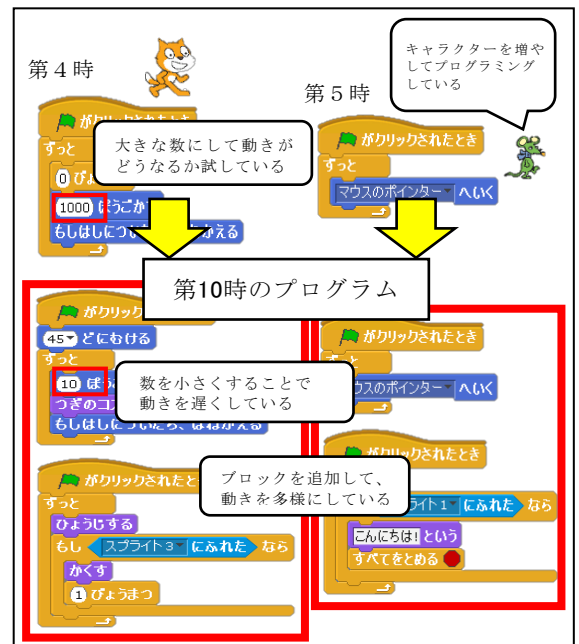


図9 C児が作成したプログラムの変容

これが第9時になるまでに、自分がイメージする動きにするために試行錯誤しながらプログラムを作成する姿が見られるようになった。例えば、図11に見られるように、「45どにむける」が「100どにむける」へと変化しているのは、動き回る傾きをもっと緩やかにしたいという児童の意図が反映されている。ここでは、「45ど」から「100ど」に変化するまでに、児童は50や80など他の数値を入れたらどうなるか試行錯誤する姿も見られた。また、横やたての動きをさせる際には、「横にうごかす」や「たてにうごかす」というブロックがないため、「横=x ぎひょう」「たて=y ぎひょう」というコンピュータに伝わるような言葉に置き換えることが必要であることにも気づいていった。これらの児童の作品の変化から、体験を重ねていくうちに、「うごきをあつめる」「くみあわせる」力に加えて「ながれをみとおす」「かいぜんする」「おきかえる」「かんがえる」力を使ってプログラムするようになっていっていると考えられる。

N=17

実践後のアンケートの結果から、「順序を考えながら問題を解いている」と答えた児童の割合は、82%から88%へと微増（1名増）した（図10）。そのうち、「とてもあてはまる」と答えた児童の割合は29%から53%へと上昇（4名増）していることが分かる。また、「順序を考えながら書いたり作ったりしている」と答えた児童の割合は、どちらも83%と変化はないが、「とても59%へと上昇（6名増）している（図11）。

これらの変容から、問題を解いたり何かを書いたり作ったりする際に、順序を考えようとする意識がより高まっていることがうかがえる。プログラミングの体験において、順序よくブロックを組み合わせれば自分の意図する動きをコンピュータにさせることができるという経験をしたことが、手順を考えようとする意識の高まりに繋がっていると考えられる。

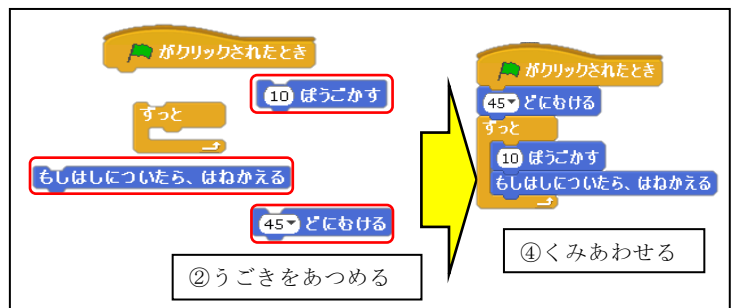


図10 第4時に作成したプログラム

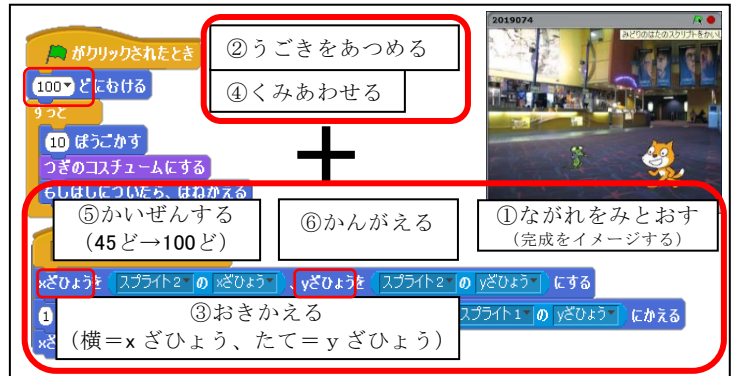


図11 第9時に作成したプログラム

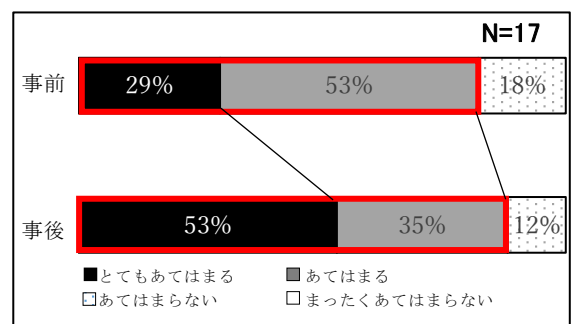


図12 順序を考えながら問題を解いている

(4) 「間違えた際の対処の仕方」に関する意識

児童は、プログラミングの体験において、自分が意図する動きをコンピュータに実行させるために、組み合わせ方や順序を考えながらプログラムを作成していった。初めは、思い通りに動かないことに困惑したり、苛立ちを感じていた児童もいた。彼らのほとんどは、教師や友達に答えを聞いたり、闇雲にブロックを組み合わせるというようにして改善を図っていた。しかし、授業の回数を重ねるごとにその姿は変化していった。意図した動きが実現させられなかったときには、順序よくブロックが組み合わされているか、足りないブロックはないか等、どこに問題があるのかを探しながら、改善する姿が見られるようになった。

実践後のアンケートでは、間違えたり、失敗したとき、「どこを直せばいいか探している」(図13)と答えた児童の割合が82%から94%へと上昇(2名増)し、「成功するまでチャレンジできる」(図14)と答えた児童の割合は76%から94%に上昇(3名増)していることも分かった。「ビジュアル」タイプのプログラミング教材は、一つ一つの動きのプログラムを画面上に残すことができるため、間違えた際の問題点を探しやすいという利点があった。また、間違えても何度でも試しながらやり直せるということも良さであった。そのため、児童らは自分が意図した動きとコンピュータの動きが違った場合には、すぐに諦めるのではなく、どこに問題があるのかを探しながら、成功するまでチャレンジしようとするようになったと考える。

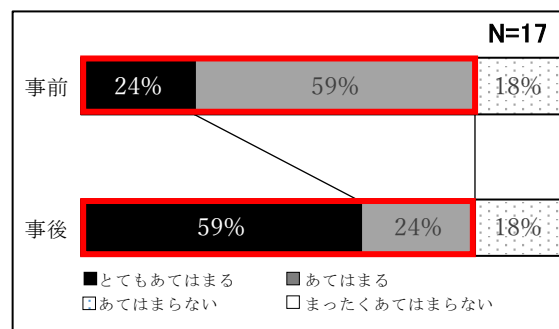


図13 順序を考えながら書いたり作ったりしている

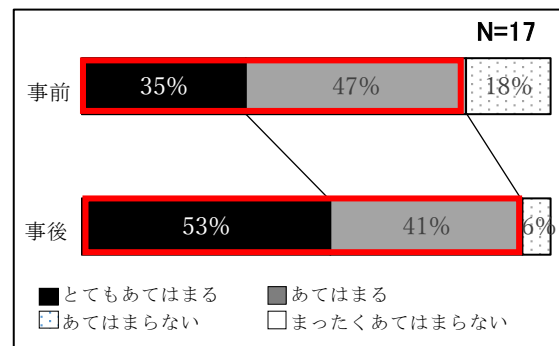


図14 間違えたり、失敗したとき、どこを直せばいいか探している

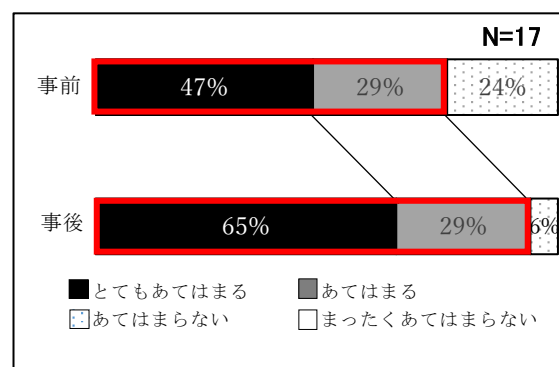


図15 間違えたり、失敗したとき、成功するまでチャレンジできる

IV 成果と課題

1 成果

- (1) ブロックを組み合わせることで意図する動きを実現させる体験を通して、問題解決の際に動きの手順や組み合わせ方を考えるなど、「プログラミング的思考」を高めることができた。
- (2) プログラムを段階的に積み重ねながら作品を作っていたことで、児童のプログラミング学習に対する意欲を高めることができた。
- (3) 「ビジュアル」タイプの教材を用いて思考を視覚化したことによって、改善点を見つけたり、試行錯誤しながら成功するまでチャレンジする姿勢が見られるようになった。

2 課題

- (1) 「プログラミング的思考」を各教科等の学習や日常生活における課題解決への活用を図る学習の計画が必要である。
- (2) プログラミングを実施する学年、教科等との関連を踏まえた授業づくり及び教材開発が必要である。

〈参考文献〉

- 小林祐紀・兼宗進・中川一史編著 2019 小学校プログラミング教育の研修ガイドブック 翔泳社
- 関和之 2019 学校では教えてくれない大切なこと 25 プログラミングって何？IT 社会のしくみ 旺文社
- 小林祐紀・他編著 2018 これで大丈夫！小学校プログラミングの授業 翔泳社
- 堀田龍也編著 2018 新学習指導要領時代の間違えないプログラミング教育 小学館
- 松村太郎・他著 2018 プログラミング教育が変える子供の未来 翔泳社
- 文部科学省 2018 『小学校学習指導要領（平成29年3月告示）』
- 文部科学省 2018 『小学校学習指導要領解説 総則編』
- 文部科学省 2018 『小学校学習指導要領解説 総合的な学習の時間編』
- 落合ヒロカズ 2017 マンガでマスター プログラミング教室 ポプラ社
- 黒田晴夫・堀田龍也 2017 プログラミング教育 導入の前に知っておきたい思考のアイデア 小学館
- 利根川裕太・佐藤智 2017 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
- 松田孝・他著 2017 小学校の「プログラミング授業」実況中継 技術評論社

〈参考 WEB サイト〉

Houe of Code

<https://hourofcode.com/jp/>（最終閲覧：2019年8月）

東京都教職員研修センター 2019 「児童の情報活用能力の育成-小学校段階におけるプログラミング教育の推進を通して-」『平成30年度 東京都教職員研修センター紀要 第18号』（最終閲覧：2019年8月）

<http://www.kyoiku-kensyu.metro.tokyo.jp/09seika/reports/bulletin/h30.html>

文部科学省 2018 『小学校プログラミング教育の手引（第二版）』（最終閲覧：2019年8月）

http://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1403162.htm

文部科学省 2016 『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）』（最終閲覧：2019年8月）

http://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/attach/1372525.htm

未来の学びコンソーシアム 『小学校を中心としたプログラミング教育ポータル』（最終閲覧：2019年8月）

<https://miraino-manabi.jp/>