

論理的思考力を重視した初等教育におけるプログラミング教育

川手 くるみ*

尾崎 剛**

広瀬 啓雄**

Programming Education in Primary Education Emphasizing Logical Thinking

Kurumi KAWATE*

Takeshi OZAKI**

Hiroo HIROSE**

キーワード： 論理的思考力, 初等教育, プログラミング教育, Scratch, アルゴリズム

1. はじめに

現在, コンピュータやネットワークの普及によりコミュニケーションや情報のやり取りの形態が変化しつつある. 情報化社会の進展の中で, 情報教育もプログラミング教育や情報リテラシー演習など多様化してきた. 特にプログラミング教育は論理的思考力や情報コミュニケーション技術 (Information and Communications Technology : ICT) を活用する力の育成にも役立つと考えられている. また, 2020年4月から施行される小学校学習指導要領^①では, 「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つけるための学習活動」を実施することになっている.

しかし, 日本の初等教育においてプログラミング教育を実施するにあたり, ボトルネックとなる部分がある. 初等教育の段階ではプログラミングを行う際に重要な要素である英語をまだ十分に習得していないという点である. 通常プログラミングは英単語の命令や英字の変数を使って作成するので, プログラミングを習得するには一定の英語力が必要となる. しかし, 日本の初等教育の段階では, プログラミングを組むために

必要な英語力が十分に備わっていない. さらに, 初等教育では情報の授業が正規に組み込まれていないので, 教師のプログラミング教育に関する経験と知識が少ない傾向にある.

このため, 初等教育のプログラミング教育における本来の目的である, 論理的思考力を身につけるための授業デザインが確立されていない. また, 初等教育でのプログラミング教育の授業改善の方法論が確立されていないことが問題である.

先行研究ではそのボトルネックを解消するため, 文部科学省が開発した日本語でプログラミングができるビジュアルプログラミング言語「プログラミン」を使用した小学生対象のプログラミング授業の実施をしたものがある^②. しかし, これらのプログラミング教育に関する先行研究は, 小学生を対象にプログラミング教育において, プログラミングを簡略化したツールを使用して実施することは有効であること^③が判明しているものの, プログラミングで必要となるデータ構造やアルゴリズムなどの説明が不十分になってしまう点が問題点として挙げられる.

本研究の目的は, ビジュアルプログラミング言語の授業と, アルゴリズムを体験的に習得する授業を組み合わせたプログラミング教育の論理的思考力の向上を

* 諏訪東京理科大学大学院工学・マネジメント研究科 (Graduate School of Engineering and Management, Tokyo University of Science, Suwa)

** 諏訪東京理科大学経営情報学部 (Faculty of Business Administration and Information, Tokyo University of Science, Suwa)

検証する。また、課題作成の演習を通じて授業改善のデータを収集し、改善の糸口を見つける方法論を提案する。

2. 本研究で利用した教材

本章では、ビジュアルプログラミング言語である Scratch と論理的思考力を修得するもしくは評価するために採用した教材について述べる。

2.1 Scratch

Scratch⁴⁾は、米国 マサチューセッツ工科大学 (MIT) のメディアラボ (Media Lab) のライフロングキンダーガーテングループ (Lifelong Kindergarten Group) が開発したビジュアルプログラミング言語ツールである。今回使用した Scratch 2.0 はインターネット上で無償配布されているもので、Web 上でプログラミングの作成、実行が可能である。そのため、プログラムの保存に必要なアカウント作成は必要になるが、ソフトをインストールするなどといった手間はかからない。

日本語で出来たブロック型のコマンドを組み合わせることでプログラムを作成するため、小学生がプログラミングを勉強する際のボトルネックとなる英語も問題点は解決する。

Scratch では、動き・見た目・音・ペン・データ・イベント・制御・調べる・演算・その他の 10 種類のカテゴリに分類されたプログラム部品を組み合わせることによって、プログラミングする。本論文では、Scratch のカテゴリごとに授業を進め、それぞれのカテゴリの理解とプログラム部品を正しく使うスキルを評価し、プログラミング能力向上の評価や授業改善の材料として使う。

2.2 アルゴリズム

アルゴリズム⁵⁾は、プログラミングの基本となる論理的思考 (アルゴリズム) をゲーム感覚で習得するための課題型ゲームソフトで、電子情報技術産業協会 (JEITA) が開発した。インターネットに無償公開されているため、Flash に対応した Web ブラウザがあればすぐに利用することが出来る。プログラミング経験

が全くない人でも楽しく「プログラミングをするための考え方」=「アルゴリズム」を知ることが出来る⁶⁾。

アルゴリズムは、ロボットの動き方を制御するコマンドブロックを組み立て、ロボットを旗のあるマスに移動させるゲームである。多数の難易度の異なる問題が用意されていて、「繰り返し」や「条件分岐」などの制御構造も利用できる。マウス操作だけで直感的に操作できるため、子供でも簡単に操作が可能である。

アルゴリズムは入門編～上級編までの問題を集めた「アルゴリズム (チャレンジ問題)」とアルゴリズムの基本動作を覚えるための「アルゴリズム Jr. (初心者問題)」の 2 種類で構成されている。難易度の異なる問題で分類された「アルゴリズム」、プログラミングの 3 つの制御構造である順次処理、繰り返し処理、分岐処理の考え方を学習する事を目的とした「アルゴリズム 2」、コマンドブロックでロボットを動かして、自由に絵を描くことが出来る「お絵かきアルゴリズム」の 3 つのバージョンがある。今回の実験では「アルゴリズム」の基本操作を説明した後、初心者用の問題が収録されている「アルゴリズム Jr.」を授業の取り入れた。ゲーム感覚で自宅でも学習できるため、授業のたびに自宅などで学習するように啓発した。

2.3 CAB

CAB(Computer Aptitude Battery)⁷⁾は日本 SHL 社が作成しているコンピュータ適正テストである。IT 業界やシステムエンジニア職の選考で使われる傾向が強い。本研究では、論理的思考力の向上を評価するツールとして、CAB を参考に小学生向け問題を作成した。

CAB のテスト項目は足し算、引き算、掛け算、割り算からなる演算を暗算で解く「暗算」、5 つの図形がどのような法則性で並んでいるかなどを問う「法則性」、命令によって、図形群が最終的にどのような結果になるかを問う「命令表」、始まりの図形と終わりの図形を見て図形の間にはさまれている暗号の変換法則を推測する「暗号」の 4 つに分類される。

複数の命令記号が示され、その命令の指示通り実行した場合にどのような結果になるかを問う命令表の問題に着目した。これは、プログラミングのアルゴリズムをトレースする能力に繋がり、また、論理的思考力を

評価するためにふさわしい分野である。本研究では、小学生用に命令記号と選択肢を少なくし、最終的な答えを導くための過程ごとに結果を問う形式とした。

3. 実験

3.1 目的

実験の目的は、論理的思考力の修得を重視してプログラミング教育の他にアルゴリズムを実践したクラスと、プログラミン教育のみ実践したクラスを比較し、授業内容の違いにより論理的思考力の向上に差異が生ずるかを検証することである。

3.2 方法

(1) 授業内容の比較

A小学校のパソコンクラブ児童4～6年生21名、B小学校のプログラミングクラブの児童4～6年生20名を被験者として実験を行った。それぞれの学年別人数は表1の通りで、学年別の人数比は、ほぼ同じ構成である。

表1 A小学校とB小学校被験者の学年別人数比

	4年	5年	6年
A小学校	7	7	7
B小学校	4	9	7

A小学校ではScratchを使用した45分授業を8回、その内1時限分をアルゴリズムの時間として割り当てた。各時限の学習内容を表2に示す。

2時限目のアルゴリズムの説明、演習は最初に「論理的思考（アルゴリズム）とは何か」について自動販売機で飲み物を買うときの人と自動販売機の動きに例えて説明し、普段の生活でも論理的思考が重要であるということを導入として話した。その後アルゴリズムの基本操作について説明した後、初心者用である「アルゴリズム Jr.」の問題を各自解いてもらった。また、自宅でアルゴリズムを起動する方法を説明し、ゲーム感覚で学習できるので自宅でも学習を続けるよう勧めた。これ以降の授業のたびに、「アルゴリズムを自宅で学習してみたか？」などの問いかけをして、自宅で自主的に学習するよう啓発した。

3時限目、4時限目のScratchの実習でははじめに例題を示し、授業の主題となる要素について説明しな

がら例題の作品を作成し、その後枠内の数字を変更するなどの練習問題を提示する流れで授業を進めた。

5～7時限目のScratchの実習では1つのゲームを作成していく流れの中で変数や条件分岐などの説明をする形式で授業を進めた。

8時限目は実習で修得したカテゴリを使う基本仕様を指定し、その他の詳細な仕様は自由にアレンジしてよい課題作品の作成時間とした。

表2 A小学校の学習内容

時数(日付)	内容
1時限(6/6)	プログラミングとScratchの説明、事前テスト
2時限(7/11)	アルゴリズムの説明、演習
3時限(7/18)	Scratch実習①(基本的な動き)
4時限(8/29)	Scratch実習②(繰り返し)
5時限(9/5)	Scratch実習③(変数、条件分岐)
6時限(10/10)	Scratch実習④(変数、座標)
7時限(10/17)	Scratch実習⑤(座標、乱数)
8時限(10/31)	課題作品の作成、Scratchの評価
9時限(11/7)	クラブ活動のまとめと事後テスト

B小学校ではScratchのみを用いた45分授業を10回実施した。B小学校の各時限の学習内容を表3に示す。A小学校とは異なり、Scratchで重要な要素である座標や変数の説明、演習に時間をかけて授業を行った。

表3 B小学校の学習内容

時数(日付)	内容
1時限(5/30)	プログラミングとScratchの説明、事前テスト
2時限(6/13)	Scratch実習①(基本的な動き)
3時限(6/27)	Scratch実習②(繰り返し)
4時限(6/27)	Scratch実習③(繰り返し、座標)
5時限(7/18)	Scratch実習④(座標)
6時限(9/12)	Scratch実習⑤(繰り返し、変数)
7時限(10/3)	Scratch実習⑥(変数)
8時限(10/3)	Scratch実習⑦(変数)
9時限(10/17)	Scratch実習⑧(変数、演算)
10時限(10/31)	Scratch実習⑨(まとめ、事後テスト)

A小学校、B小学校ともScratchの授業の進め方は同じで、各時限とも最初に例題を提示し授業の主題となる要素について説明を行った。児童が例題のプログラムを組んで動きを確かめた後、例題に類似の仕様を

練習問題として出題し各自で解くといった流れで授業を進めた。授業は1名のメイン講師と1名のTA(Teaching Assistant)が担当した。

A小学校で8時限に行った課題作品の作成はB小学校では行っていない。

A小学校, B小学校共に授業の前後にCABの命令表の問題を参考にした事前テストと事後テストを実施し, 授業の内容により論理的思考力の向上に差は出るかを検証する。

(2) 論理的思考力の評価方法

論理的思考力の評価のためにCABを参考にした事前・事後テストを作成した。CABで出題される問題は暗算, 法則性, 命令表, 暗号の4つの項目に分かれている。このなかでプログラムのトレースやアルゴリズムの考案に近い思考の, 命令表問題に着目した。

CABの命令表問題のうち, 事前・事後テストの題材とした問題は, 10種類程度の命令記号が示され, その命令指示通りに実行した場合, どのような結果になるかを考えるという内容である。実際のCABの問題では5つの選択肢が示され, そのなかから解答を選ぶ形式となっている。しかし, 本実験では小学生を対象としてその思考力を評価するために, 問題を簡略化し, 最終的な回答を導き出す途中経過の思考も考慮する。そのため, 以下に示す図1の通り命令記号を4つに削減した。また, 選択肢も4つに削減した。

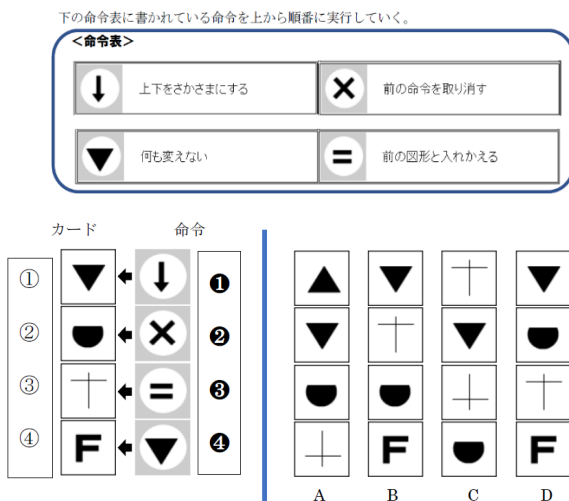


図1 CABを参考にした事前テスト

さらに, カードと命令にそれぞれ番号を振り, 問題を次の5題に分けて出題した。実際のCABでは問題

5のみが出題されるが, 本研究で作成した小学生の論理的思考力の評価テストは, 途中経過も考慮した表4の問題とした。

表4 論理的思考能力評価事前テスト

問題1	命令①が行われたときのカード①の絵はどんな絵になりますか。
問題2	命令②が行われたときのカード②の絵はどのようにになりますか。
問題3	命令③が行われたときのカード③の絵はどのようにになりますか。
問題4	命令④が行われたときのカード④の絵はどのようにになりますか。
問題5	命令①～④が上から順番に行われたとき, 正しいものはA～Dのうちのどれですか。

問題1～4の回答方式は, 答えとなる絵を実際に描く, 問題5はAからDのどれか1つを記号で選択する形式とした。

事前テストは表4の通りだが, 事後テストも事前テストと同様にCABで出題される命令表の問題を参考にして, 事前テスト同様はじめに示される命令記号を4つに削減し, 問題5の解答の選択肢も1つ減らし4つとした。事前テスト, 事後テストとも満点は5点で, 難易度はほぼ同程度である。

(3) 課題作成から理解度とスキル度の評価

A小学校8時限目の課題作成においては, まず教員が課題作品の基本仕様を説明する。次に, 児童は作品を完成させるために使用するプログラム部品が属するScratchのカテゴリを答案用紙に記入する。さらに, 自分が作成しようとしている作品の独自仕様を自由記述方式で記入する。自由記述にあたっては, 教員およびTAが机間巡視しできるだけ詳細に記述するように促した。その後, 自由に作品を作らせた。

できあがった作品と事前に記入した仕様書との差異を見ることで, プログラミングのスキルが評価できる。また, 使用するであろうと児童が予測したカテゴリと実際に使うカテゴリの差異を分析することにより, それぞれのカテゴリの使い方や意味などの理解度が評価できる。

課題作成で指定した基本仕様は, パドルを使ったピンポンゲームの作成で, ①Scratchのプログラムスタートボタンをクリック直後にボールが動き出す機能, ②ボールがパドルに当たらず落ちてしまったときの機

能、③ボールがパドルにあたったとき得点を追加する機能について、機能を実装するために必要なプログラム部品のカテゴリを回答させる。また、それぞれの機能を実行するときのイメージ(仕様)をできるだけ詳細に自由記述させた。教員は、実際に使用するカテゴリと児童が作成前に必要と考えたカテゴリを比較することにより、児童の各カテゴリに関する理解度を把握できる。本論文では、作ろうとしている機能からそれを実現するために必要なプログラム部品のカテゴリを選択する力をカテゴリ理解度と呼ぶ。

また、児童の作品に使われているプログラム部品をカテゴリ別に分類し、正しく使用されていれば、そのカテゴリのプログラム部品を使いこなすスキルがあることから、この正解率をカテゴリスキル度と呼ぶ。

ビジュアルプログラミング言語では、完成形をイメージせずにトライアル・アンド・エラーを繰り返しながら、プログラム部品を組み合わせていくことにより、本人が意図しない機能が備わってしまうことがある。特に小学生の場合、アルゴリズムを意識せずプログラムのパーツを組み合わせ、それぞれのパラメータを適当に変更することにより、思わぬ機能が付加されてしまうことがよく起こる。初等教育のプログラミング教育で論理的思考力を修得させるには、事前に作りたいモノをイメージし、その目標に向かって試行錯誤する方が論理的思考力の向上が望める。こういった理由から、プログラムを作成する前に仕様を記述し、その仕様を満たすようにプログラミングするように指導した。

3.3 結果と考察

(1) 論理的思考力について

A小学校とB小学校それぞれの、論理的思考力を評価する事前テストと事後テストの平均値と標準偏差、A小学校とB小学校の事前テストと事後テストの結果より回帰成就値⁹⁾を求めた、A小学校とB小学校それぞれの回帰成就値の平均値と標準偏差を表5に示す。

A小学校の事前テストと事後テストの平均値、B小学校の事前テストと事後テストの平均値、A小学校とB小学校の回帰成就値の平均値についてt検定をした結果、A小学校の事前テストと事後テストの平均値は有意水準1%未満で有意、A小学校とのB小学校の回帰成就値の平均値は有意水準5%未満で有意となった。

これより、A小学校の授業構成のほうがB小学校より論理的思考力が向上していると考えられる。つまり、ビジュアルプログラミング言語をつかって例題と演習の繰り返しによるプログラミング教育より、アルゴリズムなどアルゴリズムを考える教材を取り入れ、プログラムを作る前に作品の仕様を明確にして課題に取り組みせるプログラミング教育のほうが、論理的思考力を向上させることができる。

表5 小学校の事前・事後テストの比較

		事前テスト	事後テスト
A小学校 n=21	平均値	2.952	4.048**
	標準偏差	1.532	1.071
	p	0.005	
		事前テスト	事後テスト
B小学校 n=20	平均値	2.850	3.250
	標準偏差	1.089	1.482
	p	0.168	
		A小学校 n=21	B小学校 n=20
回帰 成就値	平均値	0.377*	-0.396
	標準偏差	0.948	1.571
	p	0.028	

(2) 授業改善について

A小学校8時限目の課題作成の結果、教員が指定した3つの基本機能を実現するために、児童がプログラム作成前に必要と答えたカテゴリの正解率をカテゴリ理解度、児童の作品に正しく使用されているプログラム部品カテゴリの正解率をカテゴリスキル度として表6に示す。Scratchのカテゴリは10個あるが、ペンとその他のカテゴリは課題で使用していない。また、授業中騒がしくなるため音のカテゴリは除いた。

これより、動き・制御に関してはカテゴリ理解度、カテゴリスキル度とも高く教授方法に問題が無いこと、イベント・調べるに関してカテゴリ理解度は低いがカテゴリスキル度は高いので演習として問題ないが説明をもう少し工夫する必要があること、見た目・データ・調べる・演算に関してはカテゴリ理解度、カテゴリスキル度とも低いので説明や演習の方法、時間配分など工夫する必要があることがわかる。

児童の作品を見ると、動き・制御を正しく使用した作品は多く見られたものの、約半数の児童が変数を扱うデータや演算カテゴリに手を付けることが出来なかった。これは変数や演算が理解できていなかった事に

起因し、このカテゴリを扱う授業に関して何らかの授業改善が必要である。

表6 カテゴリ理解度とカテゴリスキル度

カテゴリ	カテゴリ理解度	カテゴリスキル度
動き	69%	95%
見た目	55%	33%
データ	36%	57%
イベント	48%	100%
制御	69%	95%
調べる	0%	57%
演算	7%	43%

4. まとめ

初等教育で2020年より論理的思考力の向上を目的の一つに掲げたプログラミング教育が導入される。本論文の目的は、論理的思考力を重視したプログラミング教育について提案し、授業で得られるデータから授業改善の糸口を見つけることであった。

2つの小学校のプログラミング教育を比較することにより、プログラミングの体験を通して論理的思考力を修得するためには、ビジュアルプログラミング言語であるScratchだけではなく、アルゴリズムなど論理的思考を学ぶことが出来る教材を組み合わせた授業を行うことが有効であることがわかった。また、児童が課題作成に取り組む前に課題の仕様を明確にすることも論理的思考力の向上に繋がる。

授業改善については、課題作成前に機能を実現するために必要なカテゴリの調査を行うことでカテゴリ理解度が、児童が作成した作品のカテゴリを正しく使われているか分析することによりカテゴリスキル度が把握できる。これより、授業の説明や演習の内容で改善すべきカテゴリが洗い出せ、授業改善の糸口となる。

今後の課題として、文科省は初等中等教育の情報教育の学習活動を「プログラミングを体験することが、

探求的な学習の過程に適切に位置くようにすること」^①としている。本研究で我々が実施した授業では、課題作成中に疑問や問題が見つかったとき、教員に積極的に質問はするが、自ら調べだけで問題解決する授業に達していない。プログラミング教育を通じて、児童が自ら調べ主体的に学習する姿勢を修得できるような授業デザインについて考える必要がある。

参考文献

- (1) 文部科学省:“小学校学習指導要領”,pp.8,pp163 (2017)
- (2) 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,Vol.36,pp.9-12 (2012)
- (3) 伊藤一成:“Scratchを用いた授業実践報告”,情報処理,Vol.52, No.1, pp.111-113 (2011)
- (4) 阿部和広「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)
- (5) JEITA: “アルゴリズム”, <http://home.jeita.or.jp/is/highschool/algo/index.html> (2018年1月29日参照)
- (6) 大山裕:“アルゴリズム体験ゲーム「アルゴリズム」”,情報処理, Vol.53, No.3, pp.316-320, (2012)
- (7) CAB・GAB 試験対策集会所, www.hatikai.info/index.html (2018年1月25日参照)
- (8) 川手くるみ,大久保利亮,尾崎剛,広瀬啓雄,“初等教育におけるプログラミング学習効果の客観的評価方法の提案”,日本教育工学会第33回全国大会講演論文集,pp.255-256. (2017)
- (9) 山本洋雄,“e-Learningでの学習成績・学習時間・投資対効果などの効果測定”,教育システム情報学会誌, Vol.19, No.1, pp.46-53 (2002)