

系統的学習を考慮した小学校プログラミング教育用教材の試作

Prototype of Materials for Programming Education in Elementary Schools
by Considering Systematic Curriculum

太田 剛*

千葉県立袖ヶ浦高等学校/市川南高等学校, 教育テスト研究センター*

2020年度から始まる小学校のプログラミング教育は、発達段階に応じて各教科の中で実施することが示された。これに対して授業実践など始まっているが、単発の実践例は多いが、各学年を通じた系統的な実践は少ない。これに対して本稿では、授業でのプログラミングの位置づけを明確に定義するとともに、Scratchプログラミングの発達段階を考慮した教材プログラムを試作した。そして、今後の授業や教材の設計の指針となるような、学校現場で実現性の高い系統的なプログラミング教育のカリキュラム案を提案するものである。

キーワード：プログラミング教育、プログラミング的思考、コンピューショナル・シンキング

1. はじめに

新学習指導要領(以後、指導要領と略す)で2020年度より小学校からプログラミング教育を実施することが明記された。そして、文部科学省は新しい教科を立てることはせず、教科内でプログラミング教育を行うことを示し、その実施にあたり「プログラミング教育の実施を、子供たちの生活や教科等の学習と関連付けつつ、発達の段階に応じて位置付けていくこと」や「教科等における学習上の必要性や学習内容と関連付けながらプログラミング教育を行う」(文部科学省 2016)を示した。

その円滑な実施のため、官民で準備が進められていて、プログラミング教材や授業実践例が公開され始めているが、単発の授業や、高価なロボットを使用した教科内容とは関連が薄いものが多い。それらの中で、1)一般の学校でも実現可能、2)発達段階を考慮し系統的である、3)教科との関連付けを十分考慮している実践は、つくば市(つくば市総合教育研究所 2018)や京陽小学校(久野ら 2016)等限られたものである。

本稿では、授業でのプログラミングの活用方法の位置づけを明確に定義し、筆者の行ったScratchでの子供のプログラミング能力の使用状況の分析結果をもとに、発達段階を考慮した教材プログラムを試作した。そして、それらを元にした系統的なプログラミング教育のカリキュラム案を提案するものである。

2. 教材プログラム試作とその考慮点

以下の2点を考慮して教材プログラムの試作を行った。

2.1 プログラミングの活用方法の分類

プログラミング教育の授業実践等を体系的に分類する試みは始まったばかりである。兼宗・小林の分類の考え(井口 2017)を元に、表1に示す「プログラムで学ぶ」、「プログラミングを学ぶ」、「プログラミングで学ぶ」を定義し、さらに「プログラミングで学ぶ」を「プログラミングを活用する」と「教科の内容の本質をプログラミングする」に分けた。なお、前者は文部科学省の示す「学習上の必要性」に、後者が「学習内容と結びつけられた教材」に対応すると考える。例えば、「クイズを作る」は、クイズをつくる過程において、生徒は教科の内容を十分に把握・吟味して問題を作る必要があるだろう。但し、クイズのプログラム自体は汎用的なものであり、そこには個別の教科の内容の本質的な部分がプログラムとして表現されるわけでないと考ええる。

そして、「プログラミングで学ぶ」の内容を明らかにするため、「プログラムで学ぶ」や「プログラミングを学ぶ」を除く約40の授業実践を分析し、分類基準を作成した(表2参照)。分類は教科のどのような内容をプログラミングしたかに注目し、順次・分岐・

表1 プログラム/プログラミングの活用方法の分類

大区分	区分	概要
プログラムで学ぶ	従来からのドリル、シミュレーション、デジタル教材等の完成したプログラムを、児童が使用して学習する。	
プログラミングを学ぶ	総合的な時間等、児童がプログラムを作ること、コンピュータの特徴やプログラミング的思考を学習する	
プログラミングで学ぶ	プログラミングを活用する。(プログラミング活用)	「各教科の授業にプログラミングを取り入れる」ことでプログラミングを利用して教科理解や学習を深める。
	教科の内容の本質をプログラミングする。(本質のプログラミング)	教科学習の内容にプログラミング的思考やコンピューショナル・シンキングに対応した内容が含まれているため、プログラミングでそれを出すことで、教科理解や学習を深める。

表2 「プログラミングで学ぶ」プログラミングの分類と定義

区分	分類	定義	試作プログラム
プログラミングを活用する。(プログラミング活用)	理解支援	プログラミングすることを理解の支援する場とする。	・物語/物語基本[国語・社会] ・写真クイズ[各教科]
	協働・問題解決	協働・問題解決の場としてプログラミングを活用する。	・物語/物語基本[総合的な学習の時間] ・飛び出す地図[総合的な学習の時間・社会]
	表現	プログラミングの表現機能を活用する。	・物語/物語基本[各教科] ・写真アニメ[各教科]
教科の内容の本質をプログラミングする。(本質のプログラミング)	論理・アルゴリズム	プログラミングによって教科内容の持つ論理・アルゴリズムを表す。	・三角形/四角形/多角形[算数5年] ・速度01～03[算数6年]
	構造	プログラミングによって教科内容の構造を表す。	・起承転結[国語3/4年] ・AI英会話[外国語高学年]
	分類・分析	プログラミングによって教科内容の分類・分析方法を表す。	・水溶液分類[理科6年] ・雲(テキスト)/雲(写真)[理科5年]
	抽象化・パターン	プログラミングによって教科内容を抽象化して理解する。	・防犯ライト[理科6年] ・空気の粒00～05[理科4年]
	データ	プログラミングによってデータを扱うことで、教科内容を理解する。	・都道府県人口比較[社会5年] ・(都道府県統計データ)[社会]

反復を基本概念としたプログラミング的思考だけでなく、英国等のコンピューショナル・シンキング等の考え方も取り入れた分類になっている。

そして、表2に示すように、他の実践のプログラムなどを参考に、指導要領の中身を吟味し、この分類に対応した試作プログラムを開発した(プログラムはScratchコミュニティの「夏休み自由研究01 教科書プログラミング」スタジオで公開済である)。

2.2 発達段階(プログラミングでの使用状況)

英国の教科コンピューティングのプログラミングの学習目標を参考にして、筆者はプログラミング評価基準を作成した。そして、Scratchコミュニティ内の、小4～6の子供28名の628本のプログラムを分析し、各学年でどの評価基準内容を使用しているか明らかにした(表3参照)(太田ら 2017)。これらの子供は一般の小学生とは乖離しているとも考えられるが、各学年でどのようなプログラミングが可能であるかの一つの目安であると考え。例えば、多くの小4でもif、回数指定のループ、変数の利用が可能である

と考える。この使用状況をもとに試作プログラムの難易度を考慮した(表4参照)。このような難易度から見ると、「四角形/三角形/多角形」は、簡単なものであり入門用としても適していて、かつ、「本質のプログラミング」にも対応したものであると判断できる。

幾つかの授業実践では、プログラム自体が高度なため、児童は単に教師の指示のままプログラミングしていることが見受けられることもあるが、試作プログラムは、使用状況の分析結果に対しても簡易なものとして作成している。これは多くの児童がプログラミング自体を困難なものと感じないように、創造的にプログラムを作ることとを考慮している。例えば、図1に示す「防犯ライト」も簡易なプログラムであるが、児童にとって、初めはセンサーの距離判断に学習が集中できるようにし、さらにプログラムを作成し理解した後は、センサーの条件を変える、光にあたるスプライトの形状を変える、点灯している時間を調整する、音を出すなど、いろいろ試行錯誤しながら工夫してプログラムを拡張していくことを想定している。

表3 プログラミングの評価基準と使用率

	レベル	小4	小5	小6
分岐	C1:if	100%	100%	100%
	C2:if-else	88%	89%	100%
	C3:論理演算子	63%	78%	91%
	C4:if(-else)の入れ子	63%	89%	82%
ループ	L1:無限ループ	100%	100%	100%
	L2:ループ回数指定	100%	100%	100%
	L3:終了条件付きループ	100%	67%	100%
	L4:ループの入れ子	88%	78%	100%
モジュール	M1:複数のスプライト*	100%	100%	100%
	M2:カスタムブロック**	50%	44%	82%
	M3:引数のあるカスタムブロック	13%	22%	55%
	M4:クローン***	50%	56%	100%
データ利用	D1:変数利用	100%	100%	100%
	D2:スプライトで変数共有	88%	89%	100%
	D3:リスト型変数利用	63%	56%	64%
	D4:クラウド変数利用	50%	33%	100%
発動・トリガー	T1:特定キーによる起動	100%	89%	82%
	T2:マウス操作による起動	88%	89%	91%
	T3:背景変化による起動	50%	22%	45%
	T4:タイマー等による起動	0%	11%	27%
連携・同期	S1:背景変化を介した複数スクリプトでの連携	50%	0%	27%
	S2:同スプライトでのメッセージの利用	100%	89%	100%
	S3:他のスプライトへのメッセージでの連携	100%	89%	100%
	S4:メッセージ後のWaitでの連携	50%	33%	36%

* Scratchではスプライトが一つのオブジェクトであり、その中にプロシジャールとして複数のスクリプトを持つ。

** 一般のプログラミング言語のサブルーチンに対応。

*** 自身のスプライトの複製を作る

3 系統的なカリキュラム案

試作プログラムを組み合わせた系統的なカリキュラム案を表5に示す。これらは、児童のプログラミング能力について学年ごとに積み上げが必要かどうかで二種類に分類される。案1-3は比較的簡易なプログラムを使用し、児童は前の学年で十分にプログラミングを習得していなくても、新たに学習できるものになっている。但し、この場合でも、出来る児童は発展的により複雑なプログラミングを行うことを想定している。これに対して案4/5は、ある程度前の授業でプログラミング能力を獲得していることを前提にしている、より高度なプログラミング教育を実施することを目的としている。

これらは案であり、表4の他のプログラムや教師が作ったプログラムを使う場合も、表3や表4に示した



図1 「防犯ライト」の内容

プログラミングの発達段階を参考にすれば、独自の系統的なカリキュラムを作ることができるだろう。

4 おわりに

本稿では、Scratchのプログラミング獲得段階に参考に教材プログラミングを試作するとともに、学校現場で実現性の高い系統的なプログラミング教育のカリキュラム案を提案した。本稿で示した、プログラミングの活用方法の分類や、プログラムの難易度を評価基準に基づいて考慮することが、今後のプログラミング教育の一助になることを期待する。

参考文献

- 文部科学省 (2016) 幼稚園，小学校，中学校，高等学校及び特別支援学校の学習指導要領等の改善及び必要な方策等について（答申），http://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/059/siryo/attach/1364914.htm.
- つくば市総合教育研究所 (2018) つくば市プログラミング学習の手引き第2版，http://www.sukuba.ed.jp/~ict/?page_id=515.
- 久野靖，阿部和広，日下部和哉，池田葉乃，山崎 翔，上野美智恵，西下義之，守田由紀子 (2016) 《小特集》学校まるごとわくわくプログラミング—品川区立京陽小学校の事例—，情報処理2016年12月号別刷，情報処理学会。
- 井口裕右 (2017) 小学校のプログラミング教育，目指すべき本質はどこにあるのか，<https://japan.cnet.com/article/35110186/>.
- 太田 剛，森本容介，加藤 浩 (2017) プログラミング能力の発達段階と要因に関する定量的分析，情報教育シンポジウム論文集2017(13)，pp. 85-92.

表4 各試作プログラミング内で使用している評価基準

評価基準	分岐				ループ			モジュール				データ利用			発動トリガー			連携・同期			
	C1	C2	C3	C4	L1	L2	L3	M1	M2	M3	M4	D1	D2	D3	T1	T2	T3	S1	S2	S3	S4
物語								*							*						
物語基本								*							*				*	*	
起承転結	*				*	*		*									*	*	*	*	*
飛び出す地図								*								*					
写真アニメ					*																
写真クイズ	*	*																			
四角形					*	*															
三角形					*	*															
多角形					*	*						*									
速度 01					*			*													
速度 02					*			*				*			*						
速度 03					*			*				*			*						
防犯ライト	*	*			*																
自動ブレーキ	*				*																
水溶液分類	*	*		*																	
雲(テキスト)	*	*	*	*								*									
雲(写真)	*				*							*	*			*			*	*	
都道府県人口					*							*		*					*	*	
空気の粒 00	*				*																
空気の粒 01	*				*	*		*			*										
空気の粒 02	*				*	*		*			*				*						
空気の粒 04	*				*	*		*			*	*			*						
空気の粒 05	*				*	*		*			*	*									
AI 英会話	*	*			*	*		*	*	*		*	*	*					*	*	
入門ネコ歩き					*																
入門ネコ逃げ	*				*			*													

表5 系統的カリキュラム案

案	分類*	タイプ	使用するプログラム例	補足
1	A	新指導要領の例示	4年:(入門[総合的な学習の時間]) 物語/物語基本[総合的な学習の時間] 5年: 三角形/四角形/多角形[算数5年] 6年: 自動ブレーキ or 防犯ライト[理科6年]	・各プログラムとも入門的なものであり、積み上げの必要は低い。 ・児童がプログラムを発展させる時間をとることが望ましいだろう。
2	B	低学年から各学年で実施	1～2年: アンブラグド又はViscuit等 3年: 物語/物語基本[総合的な学習の時間] 4年: 物語/物語基本[国語・社会] 5年: 三角形/四角形/多角形[算数5年] 6年: 自動ブレーキ or 防犯ライト[理科6年]	・低学年では簡易なものを使用。 ・3～4年で物語を作成しScratchを十分に学習できた場合は、5～6年でやや難しいプログラムの使用可能。
3	B	総合的な学習の時間中心	4年:(入門[総合的な学習の時間]) 物語/物語基本[総合的な学習の時間] 5年: 飛び出す地図[総合的な学習の時間・社会] 6年: 物語/物語基本[総合的な学習の時間]	・協働学習を中心に、学校紹介、防災マップなどの作品を作る。
4	B	5/6年 算数・理科集中	5年: 三角形/四角形/多角形[算数5年] 5年: 写真クイズ[各教科](理科) 6年: 速度01～03[算数6年] 6年: 自動ブレーキ or 防犯ライト[理科6年] 6年: 水溶液分類[理科6年]	・5/6年に集中して実施して、分岐等を活用する、やや高度なプログラミングができるようにする。
5	B	リスト使用を含む高度なプログラミング	4年:(入門[総合的な学習の時間]) 物語/物語基本[総合的な学習の時間] 5年: 三角形/四角形/多角形[算数5年] 5年: 空気の粒00～05[理科4年] 6年: 水溶液分類[理科6年] 6年: 都道府県人口比較[社会5年]	・リストの使い方を学習するプログラミングを6年に実施することが望ましい。 ・可能であれば、空気の粒は4年に実施する。

* 文部科学省の「小学校段階のプログラミングに関する学習活動の分類(例)」