

# 高校での「教えない」プログラミング教育の実践と評価

太田 剛<sup>†1</sup> 南雲 智<sup>†2</sup> 森本 得憲<sup>†2</sup> 加藤 浩<sup>†3</sup>

**概要:** 新学習指導要領では2022年度より高校の情報Iで、すべての生徒を対象にしたプログラミング教育を実施することが明記されたが、情報科の教師のプログラミング経験が少ないことや、学習開始時の生徒のプログラミング経験のばらつきが大きいことが問題として指摘されている。本実践では、これらの問題に対応するため、生徒が資料をもとに主体的に自由にプログラミングする、「教えない」プログラミング授業を行った。結果として、多くの生徒が主体的に学習してプログラミングを実施していた。そして、プログラミングの経験に関係なく、多くの生徒が分岐や繰り返しなどの要素を含むプログラムを作成していた。ただし、授業に対して半数以上の生徒が非常に肯定的であるが、他の生徒は普通・楽しくないと二極化し、全体として25%の生徒が簡易なプログラムの作成に留まるものだった。今後、すべての生徒がプログラミングを楽しみ、確実な学習をするためには、プログラミング経験の有無や生徒の特性等に対応し、作成の方向性の手がかりを与える、より細やかな教材・支援方法が必要であることも示唆された。

**キーワード:** プログラミング教育、アクティブラーニング、高校情報科

## The Trial of the Programming Education in High Schools by Lecture Less

GO OTA<sup>†1</sup> SATORU NAGUMO<sup>†2</sup> TOKUNORI MORIMOTO<sup>†2</sup> HIROSHI KATO<sup>†2</sup>

### 1. はじめに

政府の「日本再興戦略2016」[1]は、2020年度からの初等中等教育でのプログラミング教育の必修化を明記した。そして、新学習指導要領[2]において、高校情報科の現行科目「社会と情報」と「情報の科学」が「情報I」に統合されて、全高校生が2022年度よりプログラミングを学ぶこととなる。この実施にあたっては、現在の情報科の教師にプログラミングの高度な内容が教育できるか疑問視されている。さらに、2020年度より小学校からもプログラミング教育が始まることから、高校段階での生徒のプログラミング経験のばらつきが多くなることも予想されている。そのため、プログラミング能力が十分で無い情報科の教師が、プログラミング教育を実施しないということも危惧されている。

これに対して、著者らは、子供向けのプログラミング・ワークショップ[3][4][5]で現在行われている、指導者が教えることなく、参加した子供が自主的に自由なプログラミングを行う学習形式を、情報科の授業に取り入れることも一つの解決方法として考えている。

本稿は、この「教えない」プログラミング教育を高校情報科の授業で行った実践結果を報告するものである。2章では先行研究を、3章では、使用したカリキュラムと実施方法について説明する。そして、4章では生徒の授業に対する反応および作成プログラムの内容を分析した結果と、5章でその考察を示す。最後の6章では、今後の高校情報

科への適用するための課題等に関して述べる。

### 2. 先行研究

従来のプログラミング教育について、田中[6]は「文法の説明、簡単な例題の解、演習課題の提出」を繰り返して行う教育と示し、この方法では、プログラムの基本も全体の4割しか習得できていないことを示した。このような伝統的な教育方法に対して、学習者がプログラミングを主体的に学習する授業の試みが始まっている。

河村[7]は、教材を整備することにより、伝統的な学習内容を学生が自習する授業を行い、この授業方法について、学生が好意的な考えを持つことを示した。また、伝統的な講義型の授業の後に、個人の自由なプログラミングや、グループで実用的なプログラムを開発することも行われている[8][9]。さらに、吉田ら[10]は、フィジカル・コンピューティングを題材として、学生がグループ作業として主体的に開発を行う「教えない」授業を行い、学生から授業に対して概ね満足な反応を得られたが、どのように「手がかり」を教師が適切に与えるかという課題も示した。

そして、岡崎ら[11]は、子供のプログラム体験イベントでの学習方法として、a)講義型、b)協働型(テキストをペアで学習)、c)個別型(テキストを一人で自習)の比較を行い、講義型と協働型が個別型に比べて、プログラミング学習に対する動機づけが有意に上昇することを示した。ただし、阿部[4]が「授業化するワークショップ」について、「自由度を

†1 千葉県立袖ヶ浦高等学校/市川南高等学校  
Chiba Prefectural Sodegaura High School/ IchikawaMinami High School

†2 千葉県立袖ヶ浦高等学校  
Chiba Prefectural Sodegaura High School

†3 放送大学  
The Open University of Japan

制限したテンプレートを用意して、予定調和的な成功体験で終わる」と批判しているように、もともと講義型は参加者に短時間の活動の中で満足感を与えるように設計されたものと見ることができるとも思えない。

### 3. 実践方法

生徒間のプログラミング経験の差が予想されることから、授業開始時から資料をもとにした自習を行う授業とした。そして、楽しんで自由にプログラミングすることを目指し、特に課題は設定せずに、最後に発表会で友達に見せることだけを学習の目的として指示した。このように本実践は、吉田ら[10]と類似した授業方法であるが、高校の IT 環境の状況を考慮し、個人作業でパソコンだけを使用してプログラム開発を行うものとした。

具体的には、初心者でも一人で学習できるように、子供のプログラミング教育で広く使用されているビジュアル言語である Scratch[12]を使用言語とした。授業の進め方は、筆者も含め多くの人が運営している、子供向けのプログラミングクラブである CoderDojo[13]で主に採用されている学習方法を参考にした(使用教材については、付録参照)。

本実践の授業内容と実施方法について以下に示す。

#### 3.1 授業内容

単元計画を表 1 に示す。授業は「フェーズ 1: Scratch の入門と自由な作品の開発」と、「フェーズ 2: プログラムの基本構造とフローチャートを意識した自主的なプログラミング」から構成される。なお、本稿執筆時は、フェーズ 1) の授業が終了した段階であり、この段階のみを本稿の対象とした。

フェーズ 1 では、筆者が作成し、多くの CoderDojo でも使用している入門書(はじまりの書)とサンプルプログラム集(Scratch 写経用教材)[14]を教材として使用し、生徒はこれらをブラウザで表示しながら学習を進めた。授業は基本的に生徒自身のプログラミング作業が中心であるが、

表 1 「教えない」プログラミングの単元計画

時数	生徒の活動	教師の活動・教材
フェーズ 1: Scratch の入門と自由な作品の開発		
1 限	ScratchID の作成 Scratch の入門教材の実施(必要に応じて) プログラムの開発	Scratch の ID 作成と基本操作の説明 入門教材とサンプルプログラムの提示
2 限	プログラムの開発:	グラフィックの作成方法
3 限	・他のプログラムの実行	著作権上の注意点
4 限	・友達との相談 ・自分のプログラム開発	他のプログラムの活用方法)
5 限	・教師・友達への質問	デバッグ方法
6 限	発表会(デモ形式)	
フェーズ 2: プログラムの基本構造とフローチャートを意識した自主的なプログラミング		
1 限 ～ 4 限	チェックリストによるプログラムの基本構造とフローチャートの自習	チェックリストの使い方の説明 生徒への個別の説明

1 時間目には、Scratch の基本操作、2 時間目以降はグラフィックの作成方法、デバッグの方法等について、授業開始時に 10 分程度の講義を行った。6 時間目の発表会では、数名ずつ出席で、他の生徒に対して自分の作ったプログラムをデモンストレーションし、他の生徒にプログラムを操作してもらう形式で行った。

さらに、生徒同士の助け合いや情報交換で学習を促進するため、「分からなかったら、出来ている人のやっていることを見る」、「分からなかったら、出来ている人に相談する」、「出来て、余裕のある人は、わからなかった人を助ける」、「分かる・分からないに関係なく、他の人がどんなことをしているか見る、話す」を実習時間のルールとした。そのため、「他の友達との話は自由です」、「席を立てて、うろついていいです」、「自分で凄いいことやったと思ったら、声出して友達を呼んで見てもらっていいです」を生徒に許可した。

#### 3.2 対象者と実施方法

表 2 に示す 2 高校で普通科 3 年生 (以後、普通科と略す)80 名(男子 58 名, 女子 22 名), 情報科 2 年生 (以後情報科と略す)41 名(男子 24 名, 女子 17 名)に対して、2018 年 4～5 月にかけてフェーズ 1 の授業を各 6 時間行った。

普通科は、実施科目を選択科目として選択し、1 年生時に「社会と情報」を履修済みであるが、その中でプログラミングの授業は行われていない。そして、情報科は 1 年生時に Scratch の 10 時間の授業を受けているが、他のプログラミングの授業は受けていない。

#### 3.3 評価・分析方法

本実践が生徒にどのような効果を与えたか検証するため、事後質問で、主体的・自律的な学習と協働学習、およびプログラムの難しさ等に関して調査した。そして、プログラミング経験や学習者の特性が学習に影響するか判断するため、Scratch プログラミングの経験、関心・興味事項等を事前質問として利用した(本来、この事前質問は、年度初めに授業での生徒のグループ分け等の参考情報として取得したものである)。

学習成果として、生徒の成果物であるプログラムについては、筆者が開発した Scratch プログラムの自動診断システム[15]を用いて、プログラムの大きさと含まれるプロ

表 2 実践対象の生徒数

実施科目	クラス数	学年	人数	小計	
情報の科学	普通科	2	3 年生	49	80
	情報科	1	2 年生	41	
アルゴリズムとプログラム	普通科	2	3 年生	31	41
	情報科	1	2 年生	41	
				合計	121

表 3 プログラム構成要素の評価基準

分類 / レベル	1	2	3	4
分岐	if	if-else	論理演算子	if(-else)の入れ子
ループ	無限ループ	ループ回数指定	終了条件付きループ	ループの入れ子
モジュール	複数のスプライト*	カスタムブロック**	引数のあるカスタムブロック	クローン***
モデル/モジュール共有	1 スクリプト*で複数のカスタムブロックの利用	異なるスクリプトでカスタムブロックの利用	異なるスプライトで同一カスタムブロック利用	カスタムブロックの再帰的呼び出し
データ利用	変数利用	スプライトで変数共有	リスト型変数利用	クラウド変数利用
発動・トリガー	特定キーによる起動	マウス操作による起動	背景変化による起動	タイマー等による起動
連携・同期	背景変化を介した複数スクリプトでの連携	同スプライトでのメッセージの利用	他のスプライトへのメッセージでの連携	メッセージ後のWaitでの連携
ユーザインタフェース	文字入力の使用	キーの検出	マウスクリックの検出	マウスの座標位置の使用

\* 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。

注) 文献[15]より引用

グラム構成要素について分析を行った。このシステムは、表 3 に示す評価基準がプログラムに含まれているか検出することが可能である。

#### 4. 結果

事前事後質問の未記入事項等の確認後、112 名を対象に分析を行った。

##### 4.1 事後質問の分析方針

事後質問は、5 件法 (5. そう思う～ 3. 普通～ 1. そう思わない) で集計した。そして普通科と情報科で学習の効果が異なると予想されるため、この両者間で分析を行った(表 4 参照)。各項目とも、肯定的な結果であり、普通科と情報科の差異について、t 検定の結果、「自分でプログラミングした内容は、時間がたっても覚えていると思う。」(以後、表 4 の略称(保持)等で示す)のみ有意差が見られた。

なお(楽しさ)については、分布状態から正規性が確認できなかったため検定は行わなかったが、図 1 に普通科と情報科別の(楽しさ)の分布示すように、非常に肯定的な見方と、やや否定的な見方に、二極化しているようである。そのため、以後の分析は、(楽しさ)で生徒をグループ分けして分析を行った(表 6 参照)。なお、(楽しさ)の中央値は 5 であるため、便宜的に 3 以下を下位群、4 以上を上位群とした。まず、(楽しさ)について Scratch プログラミングの経験の有無が影響することが考えられるため、この経験と両群の関係を直接確率検定した結果、有意差は見られなかった(表 5 参照)。

##### 4.2 主体的・自律的な学習と協働学習に注目した分析

(主体的)(考える)(集中)(協働)の項目について、下位群でも肯定的であり、上位群については、非常に肯定的な意見であった(表 6 参照)。多くの生徒が従来の授業に比べて主体的・自律的に学習したと感じたようである。

##### 4.3 授業形態や課題に注目した分析

「教えない」プログラミング教育を実施したが、(要説明)が全体で 3.46 というように、やや講義形式の説明を生徒が求めているようである(表 4 参照)。さらに、(要時間)については、上位群が下位群に比べて有意に高かったように、プログラミングにより多くの時間が必要という生徒もいた。

そして、プログラミング自体の難しさについては、(困難:構想)の方が(困難:Scratch)より高く、プログラミングより、どんなプログラムを作るかを考えることが難しかったようである。なお、(困難:構想)と(困難:Scratch)を比較した場合、t 検定の結果、上位群のみ(困難:構想)が有意に高かった ( $t(124) = 4.35, p < 0.01$ )。

##### 4.4 プログラム内容に関する分析

学習成果の指標としてプログラムの大きさと、含まれるプログラム構成要素の分析を行った。大きさは、Scratch プログラムで内のブロック(テキスト型のプログラミング言語のコマンドや命令に対応)数を使用した。構成要素については、表 3 に示す各要素を含む場合に 1 点としてプログラム構成要素数を計算した(最大 32 点)。

プログラム構成要素数に関して、上位群と下位群で差が無いようであり、これに関連すると考えられる Scratch プログラムの経験の有無の要因を追加して分析を行った(図 2 参照)。上位・下位群と経験の有無の二要因の分散分析の結果、経験の有無のみ有意差が見られた( $p < 0.01$ )。そしてプログラムのブロック数に関して同様に分析を行った(図 3 参照)。これは、構成要素数とは反対に、上位・下位群のみ有意差が見られた( $p < 0.05$ )。これは、プログラムの技術的難易度から見ると、経験がある生徒が難しいプログラムを作成しているが、(楽しい)と感じた生徒で中には経験が少なくとも大きなプログラムを作ったと考えられる。

さらに、図 4 に上位群と下位群のブロック数の違いを、作成した生徒数の割合の分布で示す。本実践で使用している入門書内のサンプルプログラムでも 30 ブロック程度の

表 4 普通科と情報科での事後質問結果

略称	事後質問	普通科 (N=73)		情報科 (N=39)		p	全体 (N=112)	
		M	SD	M	SD		M	SD
(楽しさ)	従来の授業に比べて、楽しく授業ができた。	3.96	1.31	3.68	1.33	n.a	3.86	1.32
(主体的)	従来の授業に比べて、自分で学習したと思う。	4.10	1.02	4.13	0.87		4.11	0.97
(考える)	従来の授業に比べて、自分で考えたと思う。	4.46	0.85	4.10	1.02		4.33	0.93
(集中)	従来の授業に比べて、集中して学習したと思う。	4.17	0.96	4.10	0.89		4.14	0.93
(協働)	授業中、プログラミングについて友達と相談した。	4.24	1.10	4.33	1.10		4.27	1.10
(保持)	自分でプログラミングした内容は、時間がたっても覚えていると思う。	3.92	1.04	3.45	1.18	< 0.05	3.75	1.11
(困難:構想)	どんなプログラムにするか考えるのが難しかった。	4.42	0.92	4.13	1.17		4.31	1.03
(困難:Scratch)	Scratch 自体が難しかった。	3.68	1.18	3.78	1.13		3.71	1.16
(要説明)	プログラミングについて講義形式でより説明があった方がよかった。	3.57	1.01	3.25	1.11		3.46	1.06
(要時間)	プログラミングの時間がもっと欲しかった。	4.13	0.94	4.00	1.14		4.08	1.02

補足: (楽しさ)については、分布状態から正規性が確認できなかったため検定は行わなかった(図 1 参照)

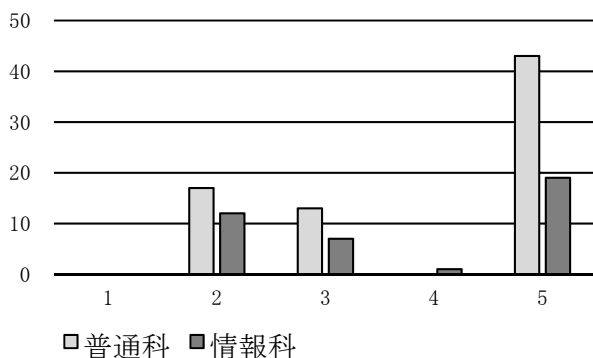


図 1 (楽しさ)の事後質問結果の分布(人数)

表 5 Scratch プログラムの経験

	未経験	経験あり		
		少しだけ	普通	得意
上位群	26	21	13	3
下位群	25	15	8	1

表 6 上位群と下位群での事後質問結果

略称	事後質問	上位群 (N=63)		下位群 (N=49)		p
		M	SD	M	SD	
(楽しさ)	従来の授業に比べて、楽しく授業ができた。	4.98	0.12	2.41	0.49	< 0.01
(主体的)	従来の授業に比べて、自分で学習したと思う。	4.59	0.68	3.49	0.93	< 0.01
(考える)	従来の授業に比べて、自分で考えたと思う。	4.79	0.57	3.73	0.96	< 0.01
(集中)	従来の授業に比べて、集中して学習したと思う。	4.62	0.70	3.53	0.84	< 0.01
(協働)	授業中、プログラミングについて友達と相談した。	4.71	0.65	3.69	1.28	< 0.01
(保持)	自分でプログラミングした内容は、時間がたっても覚えていると思う。	4.25	0.89	3.10	1.03	< 0.01
(困難:構想)	どんなプログラムにするか考えるのが難しかった。	4.63	0.62	3.90	1.27	< 0.01
(困難:Scratch)	Scratch 自体が難しかった。	3.90	1.16	3.47	1.11	< 0.05
(要説明)	プログラミングについて講義形式でより説明があった方がよかった。	3.60	1.16	3.27	0.88	
(要時間)	プログラミングの時間がもっと欲しかった。	4.49	0.73	3.55	1.09	< 0.01

大きさであるため、40ブロック以下のプログラムは非常に簡単なプログラムと考えることができる。これから、下位群で約3割の生徒が、簡単なプログラムの作成に留まっていたようである。ただし、下位群でも3割以上の生徒が100ブロック以上の大きなプログラムを作成していた。さらに、上位群でも40ブロック以下のプログラムを作成した生徒がいて、全体としては25%の生徒が簡単なプログラムを作成していた。

表 7 に個々のプログラム構成要素を使用している生徒の割合を示す。Scratch プログラムの未経験者でも、分岐を76.0%、繰り返しを94.0%の生徒が使用していた。ただし、入門書内のサンプルプログラムでも、これらの分岐や繰り返しを含んでいる。そして、「変数の利用」と「終了条件付キループ」に経験者と未経験者の間に大きな差が見られた。

#### 4.5 生徒の特性に関して

まず、デジタル・デバイドの要因としてジェンダーがあ

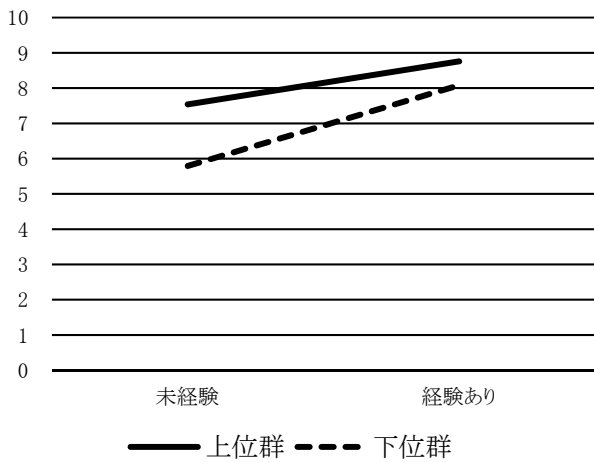


図 2 プログラム構成要素数の平均

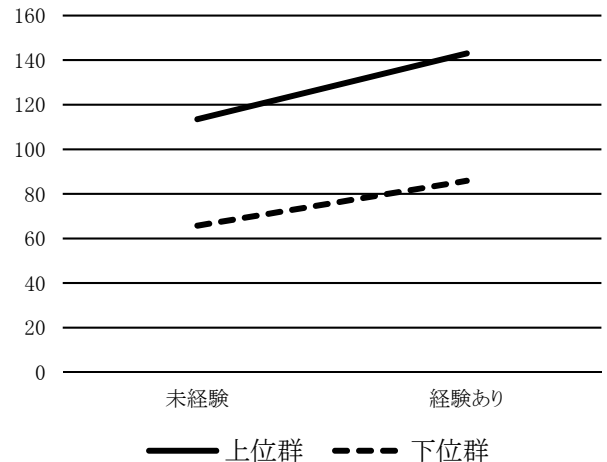


図 3 プログラムのブロック数の平均

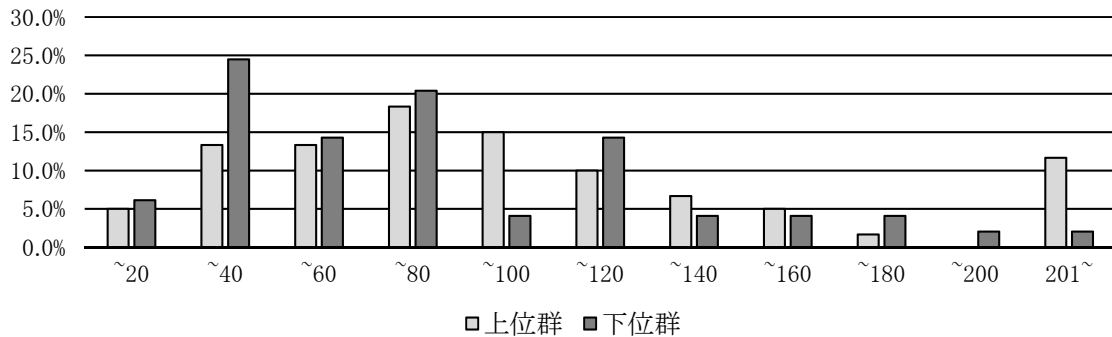


図 4 プログラムのブロック数の分布(上位群・下位群ごとの人数の割合)

表 7 プログラム構成要素を使用した生徒の割合

	if	if-else	論理演算子	if(-else)の入れ子	無限ループ	ループ回数指定	終了条件付きループ	ループの入れ子	変数利用	リスト型変数利用
未経験	76.0%	20.0%	16.0%	18.0%	94.0%	52.0%	22.0%	34.0%	38.0%	4.0%
経験あり	91.9%	12.9%	16.1%	8.1%	95.2%	62.9%	46.8%	35.5%	71.0%	4.8%
合計	84.8%	16.1%	16.1%	12.5%	94.6%	58.0%	35.7%	34.8%	56.3%	4.5%

表 8 上位群下位群の性別

	上位群	下位群	合計
女子	16	19	35
男子	47	30	77
合計	63	49	112

表 9 クラブ活動の有無

	所属無し	体育会系	文化系
上位群	12	43	8
下位群	23	14	12

表 10 生徒の興味・関心

	上位群			下位群			P
	関心無し	普通	好き	関心無し	普通	好き	
絵やイラストを描くこと	11	31	21	19	13	17	< 0.05
物をつくること(工作, 手芸 など)	2	27	34	16	19	14	< 0.01
楽器を演奏すること	16	28	19	18	19	12	
機械などをいじること	9	27	27	8	27	14	
文章を書くこと	29	27	7	23	21	5	
本を読むこと	11	15	37	14	12	23	
アニメやコミックを見ること	3	14	46	4	17	28	
SNS への投稿(LINE 除く)	15	29	19	13	22	14	

るため、男女の違いを分析した。上位群・下位群の男女数の違いに関して、直接確率検定した結果で有意差は見られなかった( $p>0.05$ )(表 8 参照)。さらに、プログラム構成要素数とブロック数についても男女差を t 検定したが有意差は見られなかった。

主体的な学習の影響する学習者の特性として、自分がある状況において必要な行動をうまく遂行できるかという自己効力感がある[16]。この自己効力感と学校での部活が関係あると考えられるため[17]、事前質問のクラブに所属しているかどうかに関して上位群・下位群で分析した。表 9 に示すように上位群の方が所属している生徒が多いようであり、直接確率検定した結果で有意差が見られた( $p < 0.05$ )

そして、プログラミングをもの作りととらえ、事前質問の興味のある事項については表 10 に示す。「絵やイラストを描くこと」と「物をつくること(工作, 手芸 など)」の項目に関して、上位群が下位群に比べて「普通」又は「好き」が多いようであり、直接確率検定した結果で有意差が見られた。ただし、「楽器を演奏すること」、「機械などをいじること」、「文章を書くこと」には有意差が見られなかった。

## 5. 考察

### 5.1 授業の楽しさに関して

半数以上の生徒が従来の授業より非常に楽しいと感じていることに対して、残りの生徒は普通、又楽しくないという二極化した結果となった。これについて、楽しくないと答えた情報科の生徒の中で数名は、ブロック数の大きな Scratch プログラムを作り、筆者らの観察では授業も楽しんでいたようである。一方、プログラミングがよくわからない、又はどんなプログラムを作ってもいいかわからず戸惑っていた生徒はブロック数も少なく、楽しくないと回答しているようである。このように、プログラミングが出来ても、出来なくても楽しくないと感じた生徒がいて、今後、その両者の違いを明らかにするとともに、その対応を考慮することで、「教えない」プログラミング教育はより良いものになると考える。

特に、Scratch プログラミング自体より、どのようなプログラムを作るかに難しさを感じている生徒も多い。間辺ら[8]も、同様に生徒が何をつくっていいかわからないことを指摘している。プログラミングとは別の場面であるが、谷口ら[18]は、ダンス学習の場面でも、「嫌い」な理由として「創作」を示したように、従来の授業で、このような創造的な活動が少なく、生徒自身が自由に作品を作ること自体に不慣れなことも考えられる。

### 5.2 「教えない」ことに関して

(要説明)に関して上位群および下位群ともに、ある程度講義としての説明は必要であると考えられる。そして、(主体的)(考える)(集中)の項目について上位群は非常に肯定的

な反応を示し、下位群でも従来の授業に比べて肯定的である。このように「教えない」プログラミング教育は、アクティブラーニングの手法としても有効であると考えられる。さらに、(協働)についても有効と考えられる。ただし、情報科は単一クラスであったため、生徒間で活発な情報交換や教え合いが行われていたが、普通科は選択科目だったため、複数クラスが集まっていたので、生徒間の協働活動は同じクラスの範囲で限定的だったように思われる。

### 5.3 プログラミング成果物に関して

未経験者でも上位群の中にはプログラミングの知識は少ない場合でも、プログラミングを積極的にやる生徒いたが、一方、全体としては、26%の生徒が 40 ブロック以内という小さなプログラムを作成していた。後者の生徒について、プログラミングがよくわからないから自由課題のプログラムが作れないのか、もしくは創造性の欠如からどんなプログラムを作ってもいいかわからず戸惑っていたのか今後明らかにする必要がある。そのため、授業中にプログラムのサイズやプログラム構成要素を細かくモニタリングすれば、プログラミングで足踏みしている生徒を見つけ出し、その原因も分かるかもしれない。

また、変数の利用に関しては、経験の有無で違いがあったが、変数はゲームを作っていた生徒等が得点を入れる時に使うなど、作成するプログラムの種類によって使用の有無が分かれているようである。

### 5.4 学習者の特性に関して

一般的にメイキングの活動である「絵やイラストを描くこと」や「物をつくること(工作, 手芸 など)」が好きな生徒が上位群に多く、さらにクラブに所属している生徒も上位群に多い。

これらのことから、何らかの学習者の特性が自主的なプログラミング学習に向いていることを示唆している。本実践では前述したように事前質問は特にプログラミングに特化して設計されたものではなく、分析では自己効力感を推測する質問として、クラブ活動の有無を使用した。今後、例えば自己効力感や創造性を調べる質問項目の利用等、プログラミングに関して学習者特性を、より明らかにし、事前にそれを調べることで、個々の学習者に適したプログラミング学習方法が提供できると考えられる。

## 6. まとめ

本稿では、「教えない」プログラミング教育を、半数以上の生徒が非常に楽しい、2 割以上の生徒は楽しくないと評価する二極化する傾向を示した。ただし、どんな生徒であっても、従来の授業に比べて主体的に学習する方法であり、分岐や繰り返し等のプログラミングの基本的な要素を使うようになることも示した。

このため、プログラミング経験が不十分な情報科の教師

に対して、「教えない」でプログラミング授業を実施することを有効な授業方法として提案したい。ただし、情報科の教師がプログラミングの知識が不要と言っているわけではなく、生徒に対する適切な働きかけ等行うためには十分な知識や経験が必要である。むしろ、教師にとっても「教えない」プログラミングの授業時間が、生徒といっしょに自らプログラミングを学習する場を提供する機会だと考える。

そのため、初めての授業で教師が、生徒の質問に答えるなどの支援が十分に出来なくても、徐々に行えるようになるだろう。そして、2020年度から小学校でプログラミング教育が始まることから、高校レベルでプログラムに詳しい生徒もいるはずで、教師が支援出来ないときに、それらの生徒が他の生徒の手助けをすることを促進すれば、授業が良いものになるであろう。

そして、本実践のようなプログラミングを「楽しくない」と感じる生徒がいることと、全体として簡単なプログラムの作成に留まる生徒がいることが問題であり、要因の明確化とその対応が必要であることも明らかになった。そのため、次のような研究・開発が早急に必要であると考えられる。

- ・プログラミング能力の高い生徒が、本実践のような場面を「楽しくない」と感じる要因の明確化。
- ・自由なプログラムを作成するための必要最低限なプログラミングの知識と、その確実な習得の支援。
- ・創造性の観点から、自由なプログラミング作成の方向性を与える手がかりの支援方法の確立。
- ・プログラミングに関する学習者の特性の特定と、その把握方法の確立。そしてその特性に基づく指導方法の開発。
- ・学習者の状態を把握するため、プログラムサイズ等のプログラム内容の細やかなモニタリング方法の確立。

今後、多くの授業実践や教材研究等を通じて、これらの課題を解決することにより、より多くの生徒が主体的にプログラミングの授業を楽しみ、プログラミング能力を獲得することができるようにしていきたい。

## 謝辞

本研究は JSPS 科研費 17K18665 の助成を受けたものです。

## 参考文献

- [1] “日本再興戦略 2016”. 2016, <https://www.kantei.go.jp/jp/singi/keizaisaisei/kettei.html#saikou2016>, (参照 2018-06-03).
- [2] “高等学校学習指導要領”. 2018, [http://www.mext.go.jp/component/a\\_menu/education/micro\\_detail/\\_icsFiles/fieldfile/2018/04/24/1384661\\_6\\_1.pdf](http://www.mext.go.jp/component/a_menu/education/micro_detail/_icsFiles/fieldfile/2018/04/24/1384661_6_1.pdf), (参照 2018-06-03).
- [3] 原田 康徳, 渡辺 勇士. ビスケットプログラミングワークショップ -なぜワークショップなのか-. 情報処理. 2017, Vol.58(10), p. 891-893.
- [4] 阿部 和広. 参加者の主体性に基づく, 変化を前提とした Scratch ワークショップの実践. 情報処理. 2017, Vol.58(10), p. 894-897.

- [5] 石原 淳也. 「自律」と「協調」で広がるオープンなプログラミングワークショップ CoderDojo. 情報処理. 2017, Vol.58(10), p. 907-909.
- [6] 田中 宏和. アクティブラーニングによるプログラミング教育法の提案～基幹系業務システム開発演習の実践事例. 日本情報経営学会誌. 2015, Vol.36(2), p. 63-73.
- [7] 河村 一樹. 講義レスによるプログラミング実習教育の試み. 研究報告コンピュータと教育(CE). 2015, Vol.21, p. 1-7.
- [8] 間辺広樹, 長島和平, 並木美太郎, 長 慎也, 兼宗進. 自宅で行うオリジナル作品制作の学習効果と問題点 ～オンラインプログラミング学習環境を用いて～. 情報教育シンポジウム (SSS2017), 2017, p. 101-109.
- [9] 高岡 詠子, 山内 崇裕, 滑川 敬章. 高等学校における実用的プログラミングの教育実践. 情報処理学会論文誌教育とコンピュータ(TCE). 2016, 2(2), p. 37-52.
- [10] 吉田 葵, 伊藤 一成, 阿部 和広. ものづくり体験を通じたプログラミング授業の設計と評価. 研究報告コンピュータと教育(CE). 2016, 134(12), p. 1-10.
- [11] 岡崎善弘, 大角茂之, 倉住友恵, 三島知剛, 阿部和広. プログラムの体験形式がプログラミング学習の動機づけに与える効果. 日本教育工学会論文誌. 2017, 41(2), p. 169-175.
- [12] “CoderDojo Japan - 子どものためのプログラミング道場”. 2012, <https://coderdojo.jp/>, (参照 2018-06-03)
- [13] Scratch. <https://scratch.mit.edu/>, (参照 2018-06-03).
- [14] “CoderDojo Japan Kata”. <https://coderdojo.jp/kata>, (参照 2018-06-03)
- [15] 太田剛, 加藤浩, 森本容介. コンピュータショナル・シンキング概念に基づくプログラム自動評価機能を持つ Scratch 用学習支援システムの開発. 教育システム情報学会誌. 2018, Vol.35, no.2, p. 204-214.
- [16] 山田政寛. ラーニング・アナリティクス研究の現状と今後の課題. 日本教育工学会. 2017, vol. 41, no 3, p. 189-197.
- [17] 横井 彩奈. 部活動が与える自己効力感への影響--達成場面と人間関係に着目して. ベネッセ総合教育研究所報. 2011, vol 60, p. 122-132.
- [18] 谷口 陽子, 松木 伸子. ダンス学習指導における導入方法について. 大手前女子大学論集. 1992, vol 26, p. 35-43.

## 付録 使用教材の概要

本実践で生徒が使用した教材を次に示す。なお、下記のサイトで公開している。

<http://multiassist.sakura.ne.jp/0Presen/index.html>

	概要
授業用スライド	<ul style="list-style-type: none"> <li>・課題と授業スケジュール</li> <li>・Scratch の実習参考資料</li> <li>・Scratch 経験者・初心者別実習方法</li> <li>・パスワードについて</li> <li>・背景やスプライトを描こう</li> <li>・プログラムの公開とスタジオへの登録</li> <li>・公開のための著作権</li> <li>・Scratch の小技(他のプログラムの利用方法)</li> <li>・デバッグ方法</li> </ul>
はじまりの書: 90分でScratch 入門	<ul style="list-style-type: none"> <li>・Web 版スクラッチの起動.</li> <li>・スクラッチのアカウントを作る</li> <li>・スクラッチを使ってみよう:(ネコ歩き)</li> <li>・修行 A(ネコ逃げ)</li> <li>・修行 B(ミュージック&amp;ダンス)</li> <li>・修行 C(多角形)</li> </ul>
写経資料(サンプルプログラム)	<ul style="list-style-type: none"> <li>・忍者練習 001 ボールよけ</li> <li>・忍者練習 002 シューティング</li> <li>・忍者練習 003 ネコたたき</li> <li>・忍者練習 004 エアーホッケー(対戦型)</li> <li>・忍者練習 005 かいじゅうを倒せ</li> <li>・忍者練習 006 歩行者信号機</li> <li>・忍者練習 007 雪の空</li> <li>・忍者練習 008 昔話</li> </ul>