

論文

Scratch を用いた実機ロボット制御を伴う プログラミング導入授業の実践

恐神 正博^{1,a)} 大熊 一正² 杉原 一臣¹

受付日 2016年2月9日, 再受付日 2016年5月10日,
採録日 2016年7月9日

概要: マサチューセッツ工科大学メディアラボで開発され、フリーソフトとして提供されているプログラミング環境 Scratch を利用した実機ロボット制御システムを試作し、それを教材として利用したプログラミング導入授業について報告する。授業では、視覚的にアルゴリズムを理解しながらプログラミングが行える Scratch を用いることで、アルゴリズム重視のプログラミングを行わせるだけでなく、授業で作成したプログラムによって、実機ロボットが動作する様子を見せる。このことにより、受講者に達成感を与え、プログラミングに関連する事項への興味・関心の向上を図った。さらに、この授業の効果を吟味するため、アンケート調査を実施した。このアンケート結果の解析から、実施した授業は、受講した生徒達のプログラミングに関連する事項への興味・関心を向上させることが分かった。

キーワード: プログラミング教育, スクラッチ, レゴマインドストーム, マイクロロボット

The Practice of Programming Education by Controlling Actual Robots Using Scratch

MASAHIRO OSOGAMI^{1,a)} KAZUMASA OHKUMA² KAZUTOMI SUGIHARA¹

Received: February 9, 2016, Revised: May 10, 2016,
Accepted: July 9, 2016

Abstract: A system based on the concept of controlling actual robots with Scratch developed by MIT Media Lab is proposed as a teaching material for computer programming. By using Scratch, it is possible to program visually and intuitively without learning specific computer languages, and also to learn algorithms that are more important for programming education. This new teaching material was used in introductory experience lessons of computer programming for high school students. A questionnaire was conducted in the lessons to investigate the change in awareness of programming. As the results of statistical analysis of the questionnaire showed, lessons in which the proposed educational tool were used improved participants' interest in matters related to programming.

Keywords: programming education, Scratch, LEGO mindstorms, micro robots

1. まえがき

今日の情報化社会を支える情報通信技術 (Information and Communication Technology : ICT) 基盤は、発展の一途をたどっており、我々が生活するうえで、必要不可欠な

ものとなっている。しかしながら、平成 25 年 7 月に内閣府から出された年次経済財政報告 [1] には、我が国の ICT 関連産業の GDP に占める割合は、平成 7 年から平成 22 年に向けて 6%程度から 11%弱へとおおむね倍増しているにもかかわらず、ICT 関連産業における雇用者数は 7%弱で推移し続け、慢性的に ICT 技術者の雇用不足が生じていると報告されている。このような情勢に配慮したためか、平成 21 年度 4 月より学習指導要領が刷新され、小学校では平成 23 年度から、中学校では平成 24 年度から、高等学

¹ 福井工業大学
Fukui University of Technology, Fukui 910-8505, Japan

² 岡山理科大学
Okayama University of Science, Okayama 700-0005, Japan

a) osogami@fukui-ut.ac.jp

校では平成 25 年度から年次進行で実施されている。この新学習指導要領では、小・中・高校のすべての課程において、「ICT の使い方」を中心とする学習から「ICT を活用した問題解決」に重きを置いた学習へと内容が拡充されるとともに、中学校の技術・家庭において「プログラムによる計測・制御」が必修化された。これを受け、プログラミング学習において、生徒の興味・関心を高めつつ学習効果を保証する教材や授業方法がさかんに研究されてきており [2], [3], [4], [5], プログラムの働きを即時に動きとして確認できる、実機ロボットを用いた初学者向けのプログラミング教育が急速な広がりを見せている。

我々も、実機ロボットを活用したプログラミング学習環境を提案してきたが、プログラミング初学者には、プログラミング言語によるロボット制御はハードルが高く、ロボットを制御したという達成感を持たせることが容易ではないことが分かってきた [6]。このため、特定のプログラミング言語の知識がなくても、直観的にプログラミングが行える Scratch に着目し、Scratch によるプログラミング授業 [7] および Scratch によるロボット制御教材を試作してきた [8]。

そこで、本論文ではこれまでに得てきた知見に基づき、プログラミング初学者を対象とした Scratch による実機ロボット制御を伴う授業を実施し、その授業の前後で行ったアンケート解析から、実施した授業が受講者のプログラミングに関連する事項への興味・関心をどのように変化させたかを報告する。

本論文は、2 章において試作した教材について説明を行い、3 章で試作教材を用いて行った授業内容の紹介およびアンケート調査の結果から、実施した授業の有効性を検証する。そして、4 章において本論を総括し、今後の課題について考察する。また、付録において実施した授業の発展内容に関する所見を記した。

2. プログラミング学習教材

本教材は、プログラミング環境としての Scratch、制御される実機ロボット 2 機種、そして、Scratch で作成したプログラム (スクリプト) を実機ロボットの制御プログラムに変換し、制御命令を発信するソフトウェアの 3 つを主要素として構成されている。以下に、試作した教材の特徴を紹介した後、教材を構成する主要素について説明する。

2.1 教材の特徴

本教材は、処理命令が書かれたブロックを組み合わせるプログラミングを行う Scratch [9] を用いることにより、プログラミング初学者にとって 1 つの難関となるプログラミング言語の文法修得の段階を排除し、作成した制御プログラムを実機ロボットの動きにより確認できる特徴を持つ。これにより、文法の習得に時間をかけ、それに翻弄されることなく、プログラミング学習において重要とされるアル



図 1 教材の構成図

Fig. 1 Configuration of teaching materials.

ゴリズムを視覚的に認識しつつ学習できる。つまり、プログラミング言語の文法やロボット固有の制御関数の説明を大きく削減でき、実際にプログラミングを行う時間および実機ロボットの動きを確認する時間を 60 分という限られた授業時間内でもある程度確保できる。この結果、受講者にプログラミングを行いロボットを制御したという達成感を与え、学習効果の向上が図れる。

また、試作教材は、出前授業等での利用に配慮し、2.2 節と 2.4 節に説明するソフトウェアおよび 2.3 節で紹介するロボットの制御に必要な補助プログラムを外部メモリにインストールした OS (Ubuntu 12.04) から利用できるようにした (図 1 参照)。この結果、既存のパソコン環境を変更することなく、外部メモリから OS を起動することによって試作した教材が利用できる。また、教材を構成するソフトウェア類がすべてフリーソフトであることも本教材の特徴といえる。

2.2 Scratch

Scratch は、Massachusetts Institute of Technology (MIT) の MIT メディアラボが、子供向けに開発したプログラミング環境であり、フリーソフトとして一般公開されている*1。

Scratch は、スプライトと呼ばれる画像を制御するプログラムを処理命令が書かれたブロックを組み合わせながらプログラミングを行う環境 (図 2 参照) であり、プログラミング初学者でも簡単に扱えるよう工夫されている*2。

すべてのプログラムは、基本構造と呼ばれる、順次処理、分岐処理、反復処理の組み合わせによって作成できるこ

*1 本教材は、旧バージョンではあるが、現在 (2016 年 5 月) でも、ダウンロード可能な「Scratch1.4」を採用している。旧バージョンを使用する理由としては、Scratch1.4 の次のバージョンから、作成したプログラムをテキスト形式で書き出す機能がなくなり、実機ロボットを制御するプログラムへの変換ができないことがあげられる。

*2 同様のプログラミング環境として、プログラミン [10]、Moon-Block [11] があるが、それらは Web アプリケーションであり、その使用にはインターネット環境が必要となる。しかしながら、出前授業ではインターネットを利用できない場合もあり、インターネット環境を使わなくても利用可能な Scratch を採用した。



図 2 Scratch の画面

Fig. 2 Scratch's application window.

表 1 Scratch における基本構造ブロックの例

Table 1 Sample blocks of elemental structure.

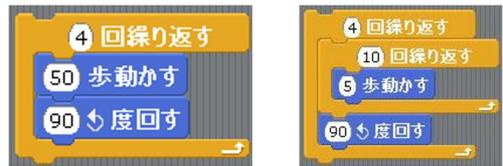
命令ブロック	処理
	順次処理
	分岐処理
	反復処理

とが知られている。しかしながら、通常のプログラム言語では、こういった基本構造を利用するために、まずは、使用するプログラミング言語の文法習得が必要となる。一方 Scratch は、図 2 に示す ① 領域のブロックパレットから、処理命令が書かれているブロックを ② 領域のスク립ティングエリアにドラックして組み合わせることにより、④ 領域のスプライト/ステージドックに置いてあるスプライトと呼ばれる画像 (図 2 の例ではネコ) の動きを制御するプログラミングが行える。この際、「○歩動かす」や「○度回す」または「○回繰り返す」といった処理内容が日本語表記されたブロック (○には数値等を入力) を利用するが、構文として成り立たないブロックは組み合わせられないため、構文を覚えなくても何かしら動きのあるプログラムを作ることができる。そして、作成したプログラムの実行結果は、③ 領域のアクションエリアですぐに確認できる。

さらに、表 1 に示すように、Scratch で使われる基本構造の命令ブロックは、その形や文字・記号から、その処理内容を直感的に理解しやすい形になっており、初学者にとっては非常に分かりやすいプログラミング環境であると考えられる [12], [13]。

たとえば、一辺が 50 歩の長さに相当する正方形をスプライトに描かせるプログラムは、『50 歩前進した後、90 度回転させる処理』を 4 回繰り返せばよいので、図 3(a) のようなブロックの組み合わせになる。

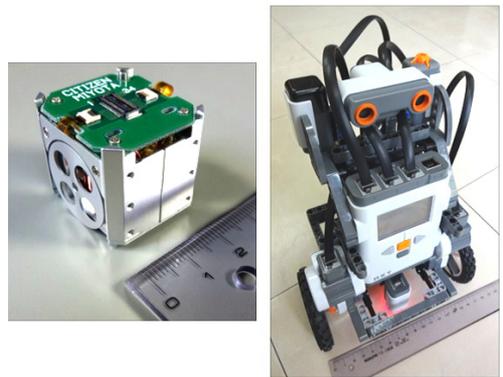
また、図 3(a) と同様の処理を行うプログラムとして、『<5 歩前進> を 10 回繰り返し、90 度回転させる処理』を 4 回繰り返させた、反復処理構文のネスト形となる図 3(b) に示す発展形プログラムも容易に説明できる。



(a) 正方形を描くプログラム (b) 2重の繰り返しを用いた (a) の発展形プログラム

図 3 Scratch で作成したプログラムの例

Fig. 3 A sample of the Scratch program.



(a) マイクロロボット (b) マインドストーム

図 4 本教材で用いられている実機ロボット

Fig. 4 The robots that were used.

2.3 実機ロボット

試作した教材は、Scratch から図 4 に示す実機ロボット 2 機種を制御することが可能である。

1 つは、RoboCup [14] Mixed Reality リーグで使用されていたロボットであり、シチズン時計株式会社が開発した一片約 3cm 立方の小型ロボット (以下、マイクロロボット) である (図 4a)。このマイクロロボットには、それぞれのバッテリーに接続されたモータが、左右の車輪に 1 台ずつ付いている。これらの車輪を制御することによって、前後進および回転だけでなく、曲線的な動きを可能にする。そして、そのような動きは、パソコンからの無線通信を利用して伝送される制御情報によって実現される [5], [6], [15]。2 機種目は、LEGO 社が開発し、ET ロボコン [16] でも使用される、レゴ® マインドストーム (以下、マインドストーム) によるロボット (図 4b) であり、これも無線通信で制御できる [17]。なお、マインドストームには専用の制御プログラム作成アプリケーションソフトウェアが用意されているが、本教材では、Scratch を利用することで、すべての実機ロボットの制御プログラム作成環境の統一を図っている*3。

*3 文献 [8] では、マイクロロボットだけを制御する教材であったが、今回の教材では、マインドストームを加えた 2 種類のロボットを選択的に制御できる機能を加えた。この際、制御ライブラリ (コード) 類が提供されている実機ロボットを本システムに追加する手順が確立できたため、比較的容易に選択可能な実機ロボットを追加でき、Scratch を共通の開発環境として、複数種類のロボットが制御可能となった。

2.4 プログラム変換ソフトウェア

マイクロロボット、および、マインドストームは、それぞれ固有のライブラリを用いて制御される。このため、通常、Scratch から制御することはできない。そこで、Scratch で作成したプログラムからそれぞれのロボット制御プログラムへと変換するソフトウェアを作成し、教材に組み込んだ。このソフトウェアは、GTK+^{*4}を利用して図 5 に示す GUI を作成し、その操作により、各ロボットに応じたライブラリを用いて、制御コードを作成する。

図 5 における ① 部から ⑤ 部で行う処理は、

- ① ラジオボタンから、2 機種ある実機ロボットのうち、どちらの制御プログラムへと変換するのかが選択
- ② コンボボックスから、マイクロロボットを動作させる場合に用いられる通信チップの ID 番号を指定
- ③ 枠内にある「開く」ボタンから、変換作業を行う Scratch で作成したプログラムのテキストファイルを選択し、プログラム変換を実施。また「開く」ボタンの左隣りにあるテキストエントリに、変換対象のファイル名を格納
- ④ 変換対象のファイル名をラベルとして表示
- ⑤ マイクロロボットを制御するときのみに利用するボタンであり、クリックすることで、マイクロロボットに制御信号を送信

である。

なお、③ の処理に伴うプログラム変換は、

1. Scratch で作成したプログラムのテキストファイルの読み込み
2. 読み込んだテキストデータから、「○歩動かす」、「○度回す」、「○回繰り返す」等、ロボットの制御に必要な命令部分を検索
3. 2 で検索された命令部以外の不要部を削除
4. 2 で検索された命令を各ロボットの相応する制御関数に変換
5. 変換先のプログラム言語に必要な header 等の挿入

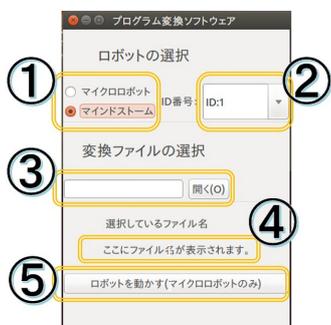


図 5 プログラム変換ソフトウェア

Fig. 5 The program conversion software.

^{*4} GTK+は、Linux をはじめとするマルチプラットフォーム上で GUI を作成するためのフリーの開発環境である。

といった手順で行われる [8].

3. 体験学習とアンケート解析

本章では、2 章で紹介した教材を用いて高校生を対象に行った授業内容の紹介と、そこで行ったアンケートの解析結果について述べる。

3.1 受講者の構成

授業は、福井県内にある 3 つの高校（普通科および専門教育を主とする学科）に通う生徒を対象に、それぞれ別の日に実施された。受講者は、3 校で 59 人であり、その構成は、表 2 に示したとおりである。ただし、内 3 名はアンケートの回答に不備があったため、解析対象から除外している。

なお、受講者らは中学校において旧学習指導要領による教育を受けているため、この授業以前に必修科目としてプログラミングの授業は受けていない。また、実習中の机間巡視からも、プログラミング経験があると言っていた生徒は、(全体で) 1, 2 名程度いたが、それらの生徒を含めても、プログラミングに熟達している様子の生徒は見受けられなかった。

3.2 授業内容

授業は、Scratch を利用可能なパソコンが、受講者全員分設置されている教室で、2.1 節に示した教材を 3 セット準備して行った。しかしながら、実機ロボットを 3 台独立に動かせるスペースが教室で確保できなかったことと、プログラム変換から実機ロボット制御までの処理操作に熟達している教員が 1, 2 名(実施授業によって異なる)であったため、実機ロボットの制御に際しては、あらかじめ分けておいた受講生のグループごとに、作成した Scratch のプログラムの中から、グループの代表となるプログラムを 1 つずつ選ばせ、USB メモリを用いて、教材の使えるパソコンにコピーした後、2.4 節で紹介した変換ソフトウェア(図 5)により、プログラム変換および実機ロボットの制御を行った。そして受講者がグループごとに実機ロボットの動きを確認する形式をとった^{*5}。

表 2 受講者の構成

Table 2 Breakdown of participants.

全体 56 人 (男 41 人, 女 15 人)					
学 校 別	A 高校	21 人	学 別	1 年生	45 人
	B 高校	24 人		2 年生	11 人
	C 高校	11 人		3 年生	0 人

^{*5} プログラミングに手間取り、自分たちの作ったプログラムによる実機ロボットの動きを確認できないグループもあったが、その場合も、他のグループのプログラムによる実機ロボットの動きを見るように指示し、受講者全員が 1 度は実機ロボットが動いている様子を見られるように配慮した。

表 3 授業計画
Table 3 Class design.

段階	受講者の学習活動	教師の指導・支援
授業前		● Scratch を起動する
導入 15分	<p>本時の目標：Scratch を用いて作成したプログラムでロボットを制御する</p> <ul style="list-style-type: none"> ● 授業前アンケートに回答する ● Scratch の基本的な操作方法を習得する <p>例題：スプライト画像を前進させるプログラムを作成する</p> <ul style="list-style-type: none"> ● 順次処理 “○歩動かす” ブロックと反復処理 “○回繰り返す” ブロックを組み合わせるプログラムを作成する 	<ul style="list-style-type: none"> ● アンケート項目の説明をする ● Scratch によるプログラミング方法を説明する ● Scratch が異常な動作を示したときには積極的に支援する ● 順次処理 “○歩動かす” ブロックの○の値を変える方法と、順次処理ブロックを積み重ねる方法について考察させる ● 順次処理 “○歩動かす” ブロックの積み重ねによる動きを反復処理ブロックを利用して再現し、反復処理ブロックの使い方を紹介する*6
展開 15分	<p>問題 1：スプライト画像に正方形の軌道を描かせるプログラムを作成する</p> <ul style="list-style-type: none"> ● 順次処理 “○歩動かす” ブロックと “○度回す” ブロックを組み合わせるプログラムを作成する ● 反復処理 “○回繰り返す” のブロックを利用してプログラムの簡素化を図る <p>問題 2：スプライト画像に正三角形の軌道を描かせるプログラムを作成する</p> <ul style="list-style-type: none"> ● 順次処理と反復処理ブロックを組み合わせるプログラムを作成する <p>問題 3：スプライト画像に五芒星の軌道を描かせるプログラムを作成する</p> <ul style="list-style-type: none"> ● 五芒星における 1 つの角の角度を算出する ● 順次処理と反復処理ブロックを組み合わせるプログラムを作成する 	<ul style="list-style-type: none"> ● 順次処理ブロックだけで正方形を描くようにプログラミングするには、同じブロックの組み合わせを 4 回利用することに気づかせる ● 正三角形の 1 つの内角は 60 度であるが、スプライトが正三角形を描くように動くにはスプライトを 120 度回転させる必要があることに気づかせる ● スプライトが五芒星を描くように動かす場合には、スプライトを 144 度回転させる必要があることに気づかせる
発展 20分	<ul style="list-style-type: none"> ● 順次処理と反復処理ブロックを組み合わせるプログラムを自由に作成する ● 作成したプログラムによって、ロボットが制御されることを確認する ● スプライトの動きとロボットの動きでは、ずれが生じることを確認する 	<ul style="list-style-type: none"> ● プログラム変換ソフトウェアを用いて、ロボットの制御プログラムに変換する ● ロボットに命令を送信して制御する ● Scratch 上で、たとえば正方形を描いた場合、スプライトは正確にスタート地点まで戻ってくるが、実際のロボットの場合、戻ってくる地点にずれが生じることや、同じ種類のロボットであったとしても、動きに個体差が生じることを気づかせる
まとめ 10分	<ul style="list-style-type: none"> ● Scratch を終了する ● 授業後アンケートに回答する 	<ul style="list-style-type: none"> ● アンケートの回答を行っているかを確認する ● 参加者全員のアンケートを回収する

表 3 に実施した授業の授業計画を示す。この授業は、高校生を対象として実施されたが、プログラミング初心者という意味で、中学校の新学習指導要領における「プログラムによる計測・制御」の「情報処理の手順を考え、簡単なプログラムが作成できること」[18] を目標とした授業の一担として実施された。実際に行った授業でも、すべてのクラスで、ほぼ表 3 の計画どおり進行できた。図 6 は授業の様子である。

また、授業においては、実機ロボットの動きが、Scratch のアクションエリアで確認した動きを完全には再現せず、

*6 ここでは、スプライトの移動に図 3(b) に示すような 2 重の反復処理を使用している。これは、図 3(a) に示すような処理を実行した場合、Scratch の “○歩動かす” ブロックに対応するスプライトの動きが速すぎ、移動が瞬間的に終了してしまうため、○歩を分割することで、動きをコマ送りのように見せる効果を狙ったものである。この際、2 重の反復処理を用いることに関して、受講生からの質問や戸惑いは見られていない。



図 6 授業の様子
Fig. 6 Class appearance.

多少のずれが生じたり、使用する個体によって多少動きが異なったりすることもあわせて示し、Scratch 上での動きと実機の動きの違いについても言及した。

なお、今回の授業では、受講者がプログラミングを通じて、それに関連する事項への興味・関心が変化するかどうかを知りたかったこともあり、1 回 (60 分) の授業に、多くの教授内容を含ませることにより生じる受講者の混乱を

表 4 授業前後で異なる設問群とそれらの回答結果

Table 4 Contents and results of the questionnaire before and after classes.

授業前設問	回答結果 (選択肢: 結果)			
	1 国語: 12 (21%)	2 数学: 22 (39%)	3 理科: 19 (34%)	4 社会: 14 (25%)
前 0) あなたの好きな教科をすべて選択してください	5 英語: 7 (13%)	6 体育: 30 (54%)	7 技術家庭科: 11 (20%)	8 芸術: 16 (29%)
前 A) コンピュータの操作が好きですか	4: 13 (23%)	3: 33 (59%)	2: 10 (18%)	1: 0 (0%)
前 I) 高校で習った情報の授業は理解できていますか	4: 7 (13%)	3: 39 (70%)	2: 9 (16%)	1: 1 (2%)
授業後設問	回答結果 (選択肢: 結果)			
後 A) 今回の授業内容を理解できましたか	4: 25 (45%)	3: 30 (54%)	2: 1 (2%)	1: 0 (0%)
後 I) この授業は楽しかったですか	4: 45 (80%)	3: 11 (20%)	2: 0 (0%)	1: 0 (0%)

表 5 授業前後で同じ設問群とそれらの回答結果の評価値の差の分布

Table 5 Distribution of evaluation value differences before and after classes.

設問 (授業の前後で共通)	授業後から授業前の評価値の差の分布						
	3	2	1	0	-1	-2	-3
前 B, 後 B) コンピュータに関連する知識の勉強は大切だと思いますか	0	0	6	45	5	0	0
前 C, 後 C) 将来「コンピュータに関連する職業」につきたいですか	0	2	10	40	4	0	0
前 D, 後 D) アプリケーションソフトウェアに興味がありますか	0	1	13	40	2	0	0
前 E, 後 E) ハードウェアに興味がありますか	0	1	13	42	0	0	0
前 F, 後 F) 情報系の授業では, 実際にコンピュータに触れる方が好きですか	1	0	11	41	3	0	0
前 G, 後 G) 日常生活でコンピュータに関する知識は役に立つと思いますか	0	0	4	48	3	1	0
前 H, 後 H) 自分にとって将来コンピュータの知識は役に立つと思いますか	0	0	5	48	3	0	0

避けるため, 順次と反復処理だけを利用した内容としている. しかしながら, プログラミング学習において, 分岐処理は必須である. そこで, 分岐処理を含めた授業内容を実施する場合, 表 3 の展開部分における問題 3 に関連する内容を分岐処理の内容に置き換える案が考えられる. これについては付録 (A.1) で紹介する.

3.3 アンケート内容と解析結果

アンケートは, 授業の前後に実施し, 授業前後で異なる設問群 (表 4 の 0, A, I) と, 授業前後で同じ設問群 (表 5 の B~H) によって構成され, 設問 0 以外は, 選択肢 (4: そう思う, 3: どちらかといえばそう思う, 2: どちらかといえばそう思わない, 1: そう思わない) からの選択形式とし, 授業前後にそれぞれ回答させた. ここで, 良い傾向の回答に対する選択肢に対しては大きい数値, 悪い傾向の選択肢に対しては小さい数値の選択肢番号を対応させ, 評価値として利用した.

表 4 における前 A から後 I の設問に対する回答結果欄において, それぞれの選択肢番号の右側にある数値は, その選択肢を選択した回答者数とその割合を指している. また, 表 5 における設問右の各数値は, 授業後に選択した選択肢番号の数値から, 授業前に選択した選択肢番号の数値を引いた値が, それぞれ, 3, 2, 1, 0, -1, -2, -3 となった受講者の数を指している.

3.3.1 受講者の傾向

表 4 に示した設問 (前 0) 「あなたの好きな教科をすべて選択してください」の回答結果では, 数学と理科を選択し

た生徒の割合がそれぞれ 39%と 34%であり, 体育を除く他の教科よりも高い. このことから, この授業に参加した生徒は, 文系の教科より理系の教科を好んでいることが分かる. これは, この授業の参加者が, 基本的に自らの興味に基づいて受講しており, 理系に分類される教科「情報」に興味を持つ生徒たちの参加が多かったためと考えられる. さらに, 表 4 の設問 (後 A, 後 I) から, 受講者らは, この授業を楽しみつつも授業内容を理解したと感じていたことが分かる.

3.3.2 授業前後でのプログラミングに関する事項への興味・関心の変化

表 5 の分布からは, 全体的に授業後の評価値が向上しているように見える. 表 6 に示した, 授業前後における評価値の差の平均においても, 設問 G 以外は正の値をとっており, このことから, 実施した授業が生徒たちのプログラミングに関連する事項への興味・関心を向上させたことが窺える.

そこで, この変化が偶発的なものではないことを確認するため, 帰無仮説 H_0 : 「授業前後の評価値の差 (授業後の評価値 - 授業前の評価値) の平均は 0 である」とし, 対立仮説 H_1 : 「授業前後の評価値の差 (授業後の評価値 - 授業前の評価値) の平均は 0 よりも大きい」とした, 一対の標本による平均値の検定を行い, その結果として t 値も示した. ここで, t 値の左上にある「*」は, 有意水準 5% で有意のもの, 「**」は, 有意水準 1% で有意のものをそれぞれ表している. その結果, 全 7 項目中 4 項目 (設問 C~F) について, 帰無仮説が棄却され「授業前後の評価値の差 (授業

表 6 授業前後における評価値の差の平均と標準偏差および有意差検定結果

Table 6 Averages of evaluation value difference, standard deviation and results of standard deviation and results of the significance test.

	設問 B	設問 C	設問 D	設問 E	設問 F	設問 G	設問 H
授業前後の評価値の差の平均値	0.018	0.179	0.232	0.268	0.196	-0.018	0.036
標準偏差	0.443	0.601	0.534	0.481	0.610	0.443	0.376
授業前後の評価値の差の平均値の有意差検定結果 (t 値)	0.299	*2.204	**3.223	**4.126	*2.388	-0.299	0.704

*: $P < 0.05$, **: $P < 0.01$

表 7 授業前後における各設問間の相関係数

Table 7 Correlation coefficient of the questionnaire before and after classes.

a. 授業前

	前 A	前 B	前 C	前 D	前 E	前 F	前 G	前 H
前 B	0.283							
前 C	0.303	0.392						
前 D	0.421	0.352	0.464					
前 E	0.378	0.182	0.512	0.614				
前 F	0.621	0.438	0.092	0.244	0.102			
前 G	0.413	0.439	0.498	0.385	0.405	0.260		
前 H	0.385	0.623	0.459	0.320	0.391	0.312	0.853	
前 I	0.293	0.290	-0.005	0.149	-0.028	0.408	0.011	0.130

b. 授業後

	後 A	後 B	後 C	後 D	後 E	後 F	後 G	後 H
後 B	0.552							
後 C	0.296	0.329						
後 D	0.382	0.456	0.376					
後 E	0.242	0.417	0.295	0.595				
後 F	0.395	0.590	0.280	0.370	0.339			
後 G	0.306	0.557	0.346	0.329	0.296	0.739		
後 H	0.414	0.653	0.303	0.392	0.406	0.685	0.831	
後 I	0.485	0.515	0.109	0.221	0.231	0.414	0.309	0.487

後の評価値 - 授業前の評価値) の平均は 0 よりも大きい」とする対立仮説が採択されたことが分かる。

つまり、実施した体験授業は、受講者に対してプログラミングに関連する事項へのイメージを向上させたと考えられる。

3.3.3 各設問間の相関

表 7 に、アンケートの各設問に関する相関係数を示す。ここで、a は授業前、b は授業後の各設問に関する相関を示している。なお、相関係数が 0.7 以上 (強い相関がある) の項目には網掛を入れている。

表 7a, b の結果から、強い相関が見られた設問の組み合わせは、「前 G と前 H」、「後 F と後 G」、「後 G と後 H」の 3 通りであった。設問内容は表 5 で示したとおり、

- 前 G・後 G：日常生活でコンピュータに関する知識は役に立つと思いますか
- 前 H・後 H：自分にとって将来コンピュータの知識は役に立つと思いますか
- 前 F・後 F：情報系の授業では、実際にコンピュータに触れる方が好きですか

である。ここで、設問 G と設問 H の相関係数は、授業前

が **0.853**、授業後が **0.831** であり、授業前後の双方で強い相関が表れている。このことから、受講した生徒は、「コンピュータの知識は、日常生活と将来の双方で役に立つ (立たない)」という考えを持っていたことがいえる。

一方、設問 F と設問 G の相関係数は、授業前には **0.260** とまったく相関が見られなかったにもかかわらず、授業後では **0.739** と強い相関が表れている。このことから、本授業を受けたことにより、「情報系の授業でコンピュータに触れることを好む (好まない) 生徒は、日常生活においてコンピュータに関する知識が役に立つと思う (思わない)」という思考を持つようになったと考えられる。

4. まとめ

今回開発した教材で用いた Scratch は、処理命令が書かれたブロックをマウス操作で組み合わせるだけでプログラミングが可能であり、ブロックもその制御処理が直感的に理解できる形をしている。この結果、ロボット制御のアルゴリズムが視覚化され、プログラミング初学者にとって 1 つの難関となるプログラミング言語の文法習得の段階を排除できる。したがって、プログラミング学習において重要

とされるアルゴリズムを視覚的に認識しつつ学習でき、さらに、作成したプログラムによって、実機ロボットが制御される様子を確認させることで、受講者の学習効果の向上を図った。

本論文では、試作教材を用いたプログラミング体験授業を実施し、そこで得たアンケートの結果から授業の評価を行った。その結果、実施した授業により、受講者のプログラミングに関連する事項への興味・関心を向上させることが可能であることが確かめられた。また、今回の解析結果は、Scratch だけを利用した授業のアンケート結果の解析 [7] と比べても、授業後の評価値が全体的に向上しており、特に、好きな科目として、文系科目を選択した生徒たちの評価がより向上していることが分かってきた [19]。よって、Scratch だけでなく、実機ロボットも利用することで、教育効果はより高められたと思われる。

一方で、用意した実機ロボットのすべてを利用するだけのスペース確保、およびプログラム変換ソフトウェアを操作する人材不足等の課題が残った。しかしながら、実機ロボットを動かすスペースに関しては、ロボットの一步に相当する動きを小さくし、動作範囲を縮小することによって対応可能であると思われる。また、プログラム変換ソフトウェアの操作に関しては、ひとまず、ティーチングアシスタントの増員で対応可能であろうが、最終的には、受講者でも利用可能な操作インタフェースへの改善も必要であると考えている。

最後に、実施した授業の発展として、マインドストームのセンサ機能を用いる内容への拡充があげられる。この案については付録 (A.2) に示す内容で検討する予定である。

謝辞 本研究において多大なるご協力をいただいた、福井工業大学の魚崎勝司氏、山西輝也氏、鹿間敏弘氏、および、四折直紀氏に感謝いたします。また、本研究の一部は JSPS 科研費 24501221, 16K01138 の助成を受けたものであり、記して謝意を表します。

参考文献

- [1] 内閣府：平成 25 年度 年次経済財政報告 (オンライン), 入手先 (<http://www5.cao.go.jp/j-j/wp/wp-je13/index.html>) (参照 2016-05).
- [2] 兼宗 進, 中谷多哉子, 御手洗理英, 福井真吾, 久野 靖: 初中等教育におけるプログラミングの実践と評価, 情報処理学会論文誌, Vol.44, No.SIG13, pp.58-71 (2003).
- [3] 千田和範, 野口孝文, 梶原秀一, 荒井 誠, 稲守 栄: ロボットを用いた初心者のためのプログラミング教育, 信学技法, ET-2006-25, pp.21-24 (2006).
- [4] 内山雅文, 市橋浩典, 高橋友一: 情報教育におけるロボット教材の比較, 第 71 回情報処理学会全国大会論文集, 2Q-9, pp.2-99-2-100 (2009).
- [5] 河原林友美, 吉田廉浩, 三田村健司: マイクロロボットを用いた教育用教材, 第 27 回ファジィシステムシンポジウム論文集, P3-03, pp.993-996 (2011).
- [6] 山西輝也, 杉原一臣, 大熊一正: マイクロロボットを用いたプログラミング導入教育の試み, 福井工業大学研究

- 紀要第 41 号, pp.486-496 (2011).
- [7] 大熊一正, 恐神正博, 籠谷隆弘, 四折直紀, 杉原一臣, 山西輝也: 高校生を対象とした Scratch プログラミング体験授業の実施とその展開, 福井工業大学研究紀要第 43 号, pp.426-437 (2013).
- [8] 大熊一正, 恐神正博, 籠谷隆弘, 四折直紀, 杉原一臣, 山西輝也: マイクロロボット制御アルゴリズムの見える化に向けて—組込みシステムの教材として, 福井工業大学研究紀要第 42 号, pp.610-619 (2012).
- [9] MIT Media Lab: Scratch 公式サイト (オンライン), 入手先 (<http://scratch.mit.edu/>) (参照 2016-05).
- [10] 文部科学省: プログラミン (オンライン), 入手先 (<http://www.mext.go.jp/programin/>) (参照 2016-05).
- [11] MOONBlock: ムーンブロック (オンライン), 入手先 (<http://moonblock.jp/#>) (参照 2016-05).
- [12] Malan, D.J. and Leitner, H.H.: Scratch for budding computer scientists, *Proc. ACM SIGCSE Bull.*, pp.223-227 (2007).
- [13] Fesakis, G. and Serafeim, K.: Influence of the familiarization with scratch on future teachers' opinions and attitudes about programming and ICT in education, *Proc. ACM SIGCSE Bull.*, pp.258-262 (2009).
- [14] ロボカップ日本委員会: ロボカップサッカー (オンライン), 入手先 (<http://www.robocup.or.jp/>) (参照 2016-05).
- [15] 佐原弘太郎, 堀澤了介, 伊藤暢浩: ロボカップ MR リーグ用ロボットへの無線通信システムの実装, ファジィシステムシンポジウム講演論文集 27, pp.989-992 (2011).
- [16] 組込みシステム技術教会: ET ロボコン 2015 (オンライン), 入手先 (<http://www.etrobo.jp/>) (参照 2016-05).
- [17] 鹿間敏弘: ET ロボコンのロボットを利用したプログラミング教育用教材の開発, 福井工業大学研究紀要第 43 号, pp.9-20 (2013).
- [18] 文科省: 現行学習指導要領・生きる力, 第 2 章 各教科 第 8 節 技術・家庭 (オンライン), 入手先 (http://www.mext.go.jp/a_menu/shotou/new-cs/youryou/chu/gika.htm) (参照 2016-05).
- [19] 大熊一正, 恐神正博, 杉原一臣: 現在準備中.

付 録

A.1 分岐処理を含む授業計画案

今回実施した授業に、分岐処理を含める場合、表 3 の展開部分における、問題 3 の五芒星を描かせる問題に関連した内容を表 A.1 のように置き換える案が考えられる。

この場合、ここまでの授業内容で作成する、正方形と正三角形を描くプログラムをそのまま 1 つのブロック (プログラム) の塊として利用する。そして、それらのプログラムを選択的に実行する方法を考えさせる。つまり、分岐処理のブロックを利用し、正方形と正三角形の動きに対応させた (変数の) 値に応じて、正方形を描くか、正三角形を描くかに処理が分岐するプログラムを作成させる。

具体的には、たとえば図 A.1 で示すようなプログラムとなるが、ここでは、変数や制御構造におけるネストについては、詳しく説明はせず、あくまで、場合 (図 A.1 の場合は、a の値) によって、処理が分岐する (選択できる) ことを実感させることを目的とする。

表 A.1 表 3 における五芒星を描かせる問題部分の置き換え内容
Table A.1 The content to replace in the exercise to draw five-pointed-star in Table 3.

展開部分	内 容
変更前	五芒星の軌道を描かせるプログラムの作成
変更後	あらかじめ設定しておく変数の値によって、スプライトが正方形の軌道を描く場合と、正三角形の軌道を描く場合に分かれるプログラムの作成



図 A.1 分岐処理のプログラム例

Fig. A.1 A program of branch processing.



図 A.2 ラインを踏んだら戻ってくるプログラム例

Fig. A.2 A program to reverse direction if the line is stepped on.

A.2 センサ機能を用いた発展的授業内容

本文で紹介した授業内容の発展として、授業時間を長くできたり、授業回数を複数に渡って実施できたりする場合には、マインドストームのセンサ機能を利用した内容への展開を考えている。この場合、Scratchのアクションエリア両端に黒いラインを配置し(図 A.2 a)、マインドストームに相当するスプライトがラインに触れたら180度回転して戻ってくる、というプログラム(図 A.2 b)を作成させる。そして、マインドストームをラインに向かって走らせ、ラインの間を行き来することを確認させれば、センサによる分岐処理を体験させることが可能となる。

なお、本文で扱ったマイクロロボットはセンサ機能を持っていないため、ラインを踏んだかどうかの判定は、定点カメラを用いた画像処理により行う方法 [15] が考えられるが、定点カメラの設置は、持ち運び面でのデメリットがあるため、センサを利用する授業の際には、マインドス

トームを利用することとする。



恐神 正博

1965年生。1988年福井大学卒業。同年日本IBM(株)入社。1995年福井工業大学大学院修士課程修了。2016年福井工業大学教授。博士(工学)。プログラムの自動生成、教育工学の研究に従事。



大熊 一正 (正会員)

1973年生。1996年香川大学卒業。2001年神戸大学大学院博士課程修了。博士(理学)。2003年福井工業大学講師。2008年同准教授。2014年岡山理科大学准教授。素粒子物理学、教育工学の研究に従事。



杉原 一臣

1975年生。2000年大阪府立大学大学院博士前期課程修了, 2003年大阪大学大学院博士後期課程修了, 博士(工学)。同年福井工業大学講師, その後, 同准教授を経て, 2014年同教授。現在, オペレーションズリサーチや, データマイニング(ラフ集合)の研究に従事。