

晴眼盲弱を区別しない利用性を備えた 10 キープログラミング教材の開発と実践

木室 義彦^{1,a)} 家永 貴史¹

受付日 2022年8月15日, 採録日 2023年1月19日

概要: 今日の情報社会では, コンピュータの動作原理を常識とすることが求められている. これを体験的に理解する方法の1つは, プログラミングを行うことである. 筆者らはこれまで, 視覚障害のある中高生を対象に PC と移動ロボットとを用いた教材を開発してきた. しかし, 小学校でプログラミングが必修化されるなか, 視覚障害のある小学生がこれを利用するには課題があった. これを解決するため, 筆者らは新たに, PC を使用せず 10 個の数字キーのみでプログラミングする手法を考案し, 晴眼盲弱の区別なく簡単な教材を開発した. この教材を用いた実験授業を行い, 視覚障害のある児童でも, 初めて教材に触れてから 15 分以内に, 晴眼児童では 5 分以内にプログラムの入力と実行ができるようになることを確認した.

キーワード: 10 キープログラミング, 視覚障害, ロボットプログラミング, 初等中等教育

Numeric Key Programming Material for both Visually Impaired and Sighted Students

YOSHIHIKO KIMURO^{1,a)} TAKAFUMI IENAGA¹

Received: August 15, 2022, Accepted: January 19, 2023

Abstract: In the informational society, it seems that elementary school students should learn programming. However, almost all programming tools for beginners employ graphical user interface, so visually impaired students cannot use such programming tools. To the problem, we have developed a new programming material only using a numeric keypad and a mobile toy robot. We then conducted experimental classes, which were focused on the ease of use the material by both visually impaired and sighted students. As a result, visually impaired students were able to obtain the programming skill within 15 minutes at maximum from their first touch of the robot. On the other hands, sighted students spent only 5 minutes to use the robot.

Keywords: numeric key programming, visually impaired, robot programming, primary and secondary education

1. はじめに

今日の情報社会では, コンピュータは児童生徒にも身近なものになっており, プログラミングを通してコンピュータでできること, できないことを理解することが重要となっている [1]. これは, 身体的な障害の有無には関係ない. しかし, 初学者向けプログラミング教材の多くは, 視覚情報を多用した GUI を用いており, 全盲弱視の児童生

徒には利用不可能なものがほとんどである. また, 学校現場への PC やタブレット等の情報端末の導入が進んでいるが, その維持管理や指導者の育成に課題も多い [2]. そのため筆者らは, 晴眼盲弱に関わらず, 簡単に利用できるプログラミング教材が必要だと考えている.

本稿では, 視覚障害のある児童生徒を対象としたプログラミング教育に関する先行研究と筆者らの研究との差異を明らかにしたうえで, 筆者らが設計・開発した新たなロボットプログラミング教材の詳細と実験授業により検証した教材の有効性について述べる.

¹ 福岡工業大学
Fukuoka Institute of Technology, Fukuoka 811-0295, Japan
^{a)} kimuro@fit.ac.jp

2. ロボットプログラミング教材

2.1 プログラミング教育とコンピュータの動作原理

筆者らの最終的な目的は、プログラミング言語の教育ではなく、コンピュータの動作原理を教えることにある。すなわち、コンピュータは、故障ではない限り、プログラムされたとおりに動作し、間違ったプログラムは間違ったとおりに動作する。人間がプログラムできないことは、コンピュータにもできない、というものである [3]。「小学校プログラミング教育の手引（第三版）」[4]でも、「コンピュータをより適切、効果的に活用していくためには、ブラックボックス化しているコンピュータの仕組みを知ることが重要であり、コンピュータは人が作成したプログラムで動いていること、コンピュータには得意なこと、できないことがあることを体験を通して気付かせる」べきとしている。

児童生徒はプログラミングを通して以上を学習することになるが、コンピュータがプログラムされたとおりに動作することを理解するには、正しく動くプログラムだけではなく間違ったプログラムも有用である。アルゴリズムのミスやキー入力ミス、文法ミスなど様々なミスの結果も、コンピュータの動作として体験できたほうが良いと筆者らは考えている。

2.2 関連研究

初等中等教育では、ビジュアル型プログラミング言語が用いられることが多く、特にマウスを用いてブロックを操作するブロックプログラミング言語がしばしば利用されている [4]。このブロックプログラミングの簡単さの要因として、Bauらは、(1) コマンドは暗記ではなく選択、(2) プログラムの構造をブロックで理解、(3) 文法エラーが生じにくい、を挙げている [5]。このような初学者向けプログラミング環境を視覚障害者も利用できるようにする研究がいくつか行われている。

Hadwen-Bennettらは、視覚障害者向けプログラミング環境の論文を調査し、スクリーンリーダーの併用やブロックプログラミングの改良の試みがあると述べている [6]。しかし、任意のブロックをスクリーンリーダーで読み上げることは難しく、タブレットPCといったタッチスクリーン環境がさらなる障壁となっているとMilneらは報告している [7]。タッチスクリーンやスクリーンリーダーを用いないものとしては、ブロックプログラミングのブロックを実体化したMicrosoftのCode Jumper [8]や本吉らのP-CUBE [9]、川崎らのブロックにQRコードを付したものの [10]などがある。しかし、これらは視覚障害者専用の教材であり、晴眼者が使うメリットはなく、普及に至っていない。

一方、プログラムの振る舞いを実体化するロボットは、

その動きを手で触って確認できることから、視覚障害者のプログラミング教材に適している。Oliveiraらは、ロボットを用いた視覚障害者向けプログラミングに関する論文のサーベイを行い、その有用性を確認している [11]。国内でも、触知ブロックを使用する市販ロボットの利用例がある [12]。

なお、晴眼者が使用するテキストベースのプログラミング環境を視覚障害者に対応させるものとしては、たとえば、長岡らのJAVAおよびC#の学習支援環境 [13]やCをベースにしたLudiらのLEGO Mindstorm用のJBrick [14]などがある。小林らは、スクリーンリーダーとArduinoボードを用いるJAVAプログラミングを試みている [15]。しかし、これらの多くは、PCの操作スキルのある高校や大学の学習者向けとなっている。

2.3 設計方針と特徴

以上のように、多くの教材は、晴眼初学者向けの教材に機能や機材を追加するものであり、晴眼盲弱を区別せず、視覚障害児と晴眼児とがともに学べるという観点から教材の設計を行ったものは、ほとんどない。これに関し筆者らは、前掲のBauのプログラミングの3つの簡単さに注目した。彼らの示した特徴は、できるだけ間違えさせない、間違ったプログラムは実行させないというものである。間違ったプログラムも実行させ、コンピュータの動作原理を教えようという筆者らの目的には適さない。すなわち、筆者らの考える教材の簡単さとは、次の3つである。

- (1) マウスとGUIに依存せず、入力しやすく、覚えやすく、タッチタイプ可能な命令セット
- (2) 見て分かる、ではなく、身体性の概念の延長として理解できる直感的な教材
- (3) 文法エラーや入力ミスも含めて、プログラムの実行結果をシステム単体で確認できるPC不要の教材

さらに、教材が果たすべき追加要件として、プログラミングの概念を学齢や小中高の学校種に応じて段階的に体験できることも重要である [16]。しかし、様々なプログラミング教育実践により、晴眼、盲弱双方でその課題も指摘されている [17], [18]。

以上の検討により、次節で述べるように、プログラミングに必要なキーを数字キー10個のみと制限を付けた。プログラミングの対象は、児童生徒の興味を引きやすい移動ロボットとする。プログラムの実行はインタプリタ型とし、文法エラーがあっても実行する。すなわち、最も低レベルのCPUとして動作する単体のロボット教材である。

3. ロボット言語仕様と実装

筆者らは、これまでに移動ロボットとPCのテンキーを用いた視覚障害のある中高生向けのプログラミング教材を開発してきた [19]。PCの使用スキルが必要なこの教材に

対し、今回、小学校低学年からの利用を想定し、PC を用いず、ロボット単体でプログラミング可能にするために、テンキーを移動ロボットに搭載し、プログラム入力も行えるインタプリタの開発を行った。また、この移動ロボットの命令セットに新たに乱数および WHILE 文を追加した。小学校ではサイコロを既知としてその展開図を学習し、乱数自体は学ばないが、中学では無作為抽出、高校では、確率・統計で学ぶ [20]。高校情報では、計算機シミュレーションでも扱われる [21]。すなわち、乱数の機能は、サイコロ以外のランダムな動きを体験するきっかけになる。一方、条件付き繰り返しの WHILE 文は、FOR 文、IF 文の次のステップとして高校で学習 [22] することから、小中高のプログラミング学習の連続性として追加した。これにより、変数の概念を除く多くのプログラミングの概念を学齢に応じて段階的に体験できるようになる。図 1 は、開発したロボット教材と全盲児童による使用の様子である。

3.1 移動ロボットの命令セット

独立 2 輪駆動の移動ロボットの命令セットは、前進後退、左右回転の各移動とピープ音に関する 5 つの基本命令と繰り返し、条件分岐、条件付き繰り返しの 3 つの制御命令である (表 1)。

基本命令は、命令コードとオペランドの 2 語から構成される。オペランドは 1 桁の数字をとることができ、移動についてはその動作時間、回転については、回転角度 (4 の

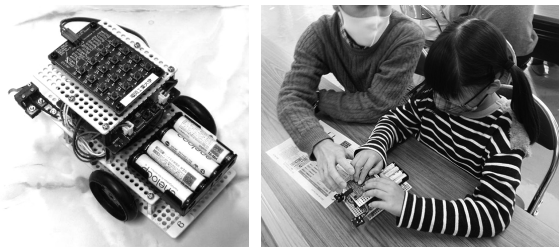


図 1 開発したプログラミング教材

Fig. 1 Our developed Robot.

表 1 移動ロボットの命令セット

Table 1 Robot programming command set.

Command	Operand	Description
FW	d	Move forward (0:random)
BK	d	Move backward (0:random)
LR	d	CCW rotation (0:random)
RR	d	CW rotation (0:random)
BP	d	Musical scale (0:random)
FOR	d	Loop block (d times; 0:∞)
NEXT	-	End of loop or while block
IF	sid & ss	Conditional branch
ENDIF	-	End of IF block
ELSE	-	Else clause

ときに 90 度回転)、ピープ音についてはド (C4)~レ (D5) の音階となっている。ただし、オペランドが 0 の場合は、停止や無音ではなく、実行時に 0~9 の乱数がセットされる。

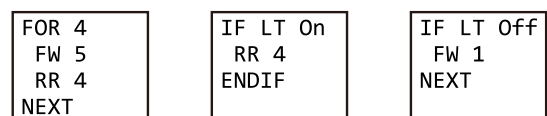
制御命令は、ブロック構造で記述する。繰り返し処理は、FOR と NEXT に囲まれたブロックを FOR に続く 1 桁の数字の回数だけ繰り返す。ただし、0 が指定された場合は無限ループとなる。条件分岐は、IF と ENDIF で囲まれたブロックを条件節が真のときに実行する。条件節は、センサの識別子とセンサの状態 (On/Off) で表す。条件付き繰り返しでは、命令コードの WHILE と ENDWHILE がなく、IF と NEXT に囲まれたブロックが WHILE 文として実行される。これは、次節で説明するように 10 個のキーに命令セットを収めるためである。図 2(a) は、前進 (FW) と右回転 (RR) を 4 回繰り返し、四角形を描く FOR 文、図 2(b) は、タッチセンサを用いた IF 文の例である。コード中の LT は、左タッチセンサを意味している。図 2(c) は、WHILE 文の例である。

なお、以上の説明では、FOR や IF など、英文コマンドのように記述したが、後述するように、各命令はそれぞれボタン 1 個に対応しているだけである。そのため、命令コードを覚える際には日本語でも他言語でも学習者が自由に決めてよい。

3.2 プログラムの入力方式

初学者によるプログラム学習の障壁の 1 つとして命令の暗記と入力がある。ビジュアルプログラミング言語は、これをメニューの一覧表で解消している。一方筆者らは、晴眼盲弱を区別せずプログラムを入力できるようにするために、0 から 9 の数字キーのみのキーボードを用いることとした。すなわちプログラムは、数字列であり、人間には理解困難とされる機械語といえる。この障壁に対し、本システムでは、命令の配置を人の身体性に一致させることで命令を記憶しやすくする。

テンキーボードの数字配列は、一般に電話のボタン配列と PC のキーボード配列の 2 種類があるが、どちらも、中央に「5」キーが配置されている。ここで筆者らはキーの数字ではなく、キーの空間的な配置に着目する。すなわち、ロボットの前進と後退、左右回転の命令を「5」キーを中心として上下左右に配置する (図 3 右)。これによ



(a) FOR stmt (b) IF stmt (c) WHILE stmt

図 2 サンプルプログラム

Fig. 2 Sample programs.

り、電話ボタン配置かテンキー配置かによらず、共通した入力が可能となる。なお、数を入力する場合は、数字の表記どおりである。

制御命令やセンサ選択については、空いているキーに配置する。筆者らのこれまでの実験授業の経験から、制御命令の学習では、FOR文を先に教えたほうが児童生徒の理解は早かった。少ない命令数で長い複雑な動きを実現でき、児童生徒の興味が持続するためである。このことから、FORとNEXTは、キーボード上段左右に配置することとした。その後に学ぶIFとENDIFは、下段左右とした。また、センサ選択キーの配置は、実際のロボットのセンサ配置（左右や上下等）に一致させることで、視覚的に、また、視覚障害のある学習者にはセンサを触って理解できるようにしている（図4）。実行/停止とリセットは、「0」キーの左右に配置している。

以上の工夫により、前節の図2のサンプルプログラムは、実際には、図5のように入力されることになる。

3.3 インタプリタと状態遷移

10個の数字キーに命令コードとオペランドの2種類を対応させたことから、同じキーでも入力の順序で意味が異なってくる（図6）。これをキー入力に応じて状態を遷移

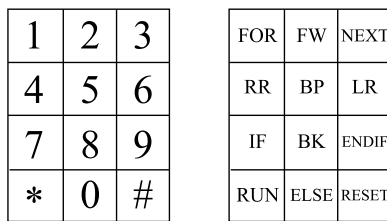


図3 電話ボタン配置のキーパッドとコマンド配置

Fig. 3 Layout of a telephone type keypad and commands.

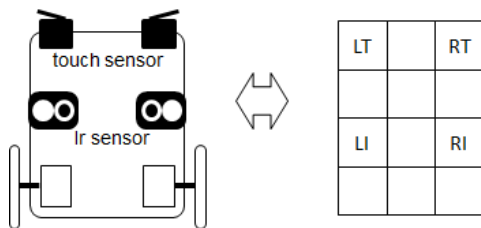


図4 センサ配置とキーパッド

Fig. 4 Layout of sensors and command keys.

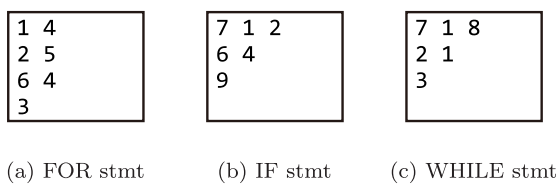


図5 サンプルプログラム2

Fig. 5 Sample programs.2.

させることで、基本命令や制御命令、オペランドの入力を可能にしている（図7）。

インタプリタが起動すると、基本命令と制御命令を入力できる状態1となる。基本命令もしくはFORを入力した場合は、数字入力の状態2に遷移し、入力が終わると状態1に戻る。状態1においてIFが入力されると、状態3でセンサを選択、状態4でセンサのOn/Offを指定、その後、状態1に戻る。状態3と状態4の入力はマージされて1つのオペランドとしてメモリには格納される。このほか、すべての状態で、プログラムの実行と停止を「*」キーで、プログラムのクリアを「#」キーで行うことができる。

3.4 実装

ロボット教材のベースは、市販ロボット玩具（KOROBO MR-9172 EK-JAPAN）である。2つのモータ、2つのタッチセンサと光センサ、ブザー等を搭載している。これにArduino UNOボードとシールド基板（SU-1201 EK-JAPAN）、および、16個のタクトスイッチを搭載した4x4キーパッド（アイロジック）を設置する（図8）。キーパッドは、抵抗分圧式の回路となっており、Arduino

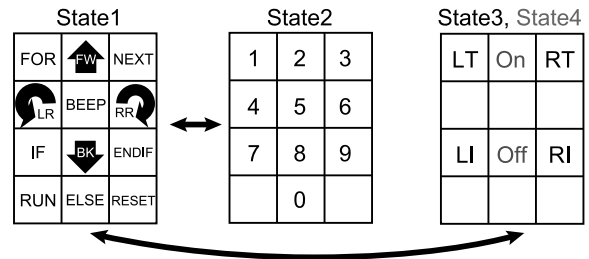


図6 各状態のキー配置

Fig. 6 A layout of keys in each state.

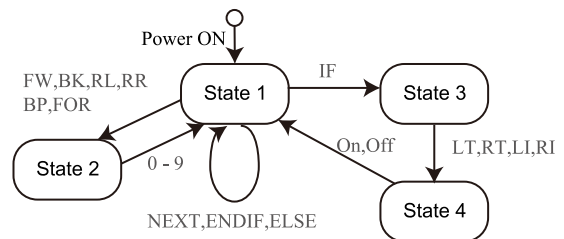


図7 キーパッドの状態遷移

Fig. 7 State transition of the keypad.

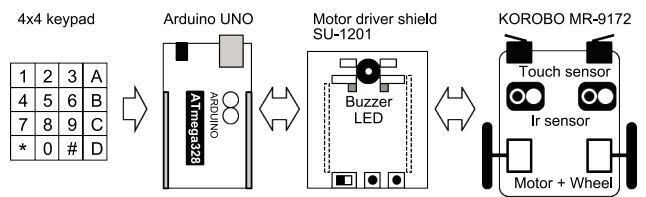


図8 ハードウェア

Fig. 8 Robot hardware.

のアナログ端子 1 ピンに接続することで、16 個のキーを区別して入力することができる。「5」キーのキートップには、通常のキーボードや電話のボタンと同様、凸部を設け、視覚障害のある児童生徒でもキーの配置が分かるようにしている。なお、キーパッド右端の「A」～「D」の 4 つのキーは、プログラミングには使用しておらず、押しても無視される。

前節で設計したロボット言語のインタプリタは、Arduino 上にファームウェアとして実装している (図 9)。電源オンと同時にインタプリタが起動し、キーパッドからプログラム列を入力し、実行することができる。キー押下時

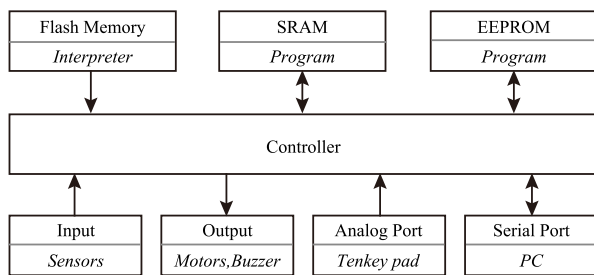


図 9 ブロックダイアグラム

Fig. 9 Block diagram.

表 2 シラバス

Table 2 Syllabus.

移動ロボットで学ぶプログラミング	
達成目標	移動ロボットのプログラミングを通じてコンピュータの動作原理を理解する。
学習時間	45 分 × 2 コマ
対象者	プログラミング初学者 (晴眼盲弱の区別なし)
学習の流れ	
1	機材の確認と導入 <ul style="list-style-type: none"> 移動ロボットの構造の確認 テンキーパッドのボタン配置、「5」キーの凸部の確認 プログラムのリセット、実行と動作確認 (「2」+「5」)
2	逐次処理 <ul style="list-style-type: none"> 基本命令：前進後退、左右回転、ピープ音命令の発見と確認 命令とオペランドとの関係 演習：U ターン、四角形周回、ロボットダンス等 乱数 (サイコロ、ルーレット)
3	繰り返し処理 <ul style="list-style-type: none"> 繰り返し処理のプログラミング方法 無限回の繰り返し 演習：繰り返し処理を用いたプログラム
4	条件判断 <ul style="list-style-type: none"> センサの説明と確認 演習：サンプルプログラムでセンサ動作を確認 条件判断の理解 演習：センサに応じてロボットの回転方向を制御
5	応用 <ul style="list-style-type: none"> 演習：盲導犬ロボットまたはライトレースロボットの作成 条件付き繰り返し
6	まとめ <ul style="list-style-type: none"> 学習内容のまとめ

は、操作音が鳴る。入力されたプログラム列は、自動的にマイコンの EEPROM にも保持され、電源オフ後も再実行可能となっている。これにより、暴走して電源オフでなければ止められない間違ったプログラムも再現できる。現在のファームウェアでは、入力可能なプログラムは 256 ステップであり、FOR、IF および WHILE 文の入れ子は、それぞれ 10 段まで可能である。なお、表示用のモニタは搭載しておらず、また、プログラムコードの部分的な修整や削除、読み上げといった編集支援機能も備えていない。このため、学習者は頭の中でプログラムを考え、動きを予想し、コードの追加と実行、クリアといったデバッグを繰り返す。他者には自分の言葉やロボットの動きでプログラムを説明することになる。プログラムが長くなると紙に書くなどの工夫が始まる [23]。

3.5 シラバス

本教材を用いた授業のシラバスの例を表 2 に示す。このシラバスは、文献 [19] において視覚障害のある中学生以上を対象として設計したものをベースとしている。学校授業の 45 分 × 2 コマで実施可能なものであるが、小学生では、学齢や授業時間に応じて、逐次処理から繰り返し処理や条件分岐、応用のどこまで行うかを選べば良い。次章で説明する視覚特別支援学校での実験授業では、このシラバスの「逐次処理」までを実施した。

4. 実証実験

初等教育における移動ロボットを用いたプログラミング教材の有用性については、晴眼および視覚障害者それぞれについて、数多くの研究者から報告されている。今回筆者らは、視覚障害のある中高生向けのロボットプログラミング教材をベースとして、晴眼盲弱を区別せず、小学校低学年からプログラミング体験可能を目標に、新たに筆者らが定義した「3 つの簡単さ」と「学齢に応じた学び」の 2 つの観点で開発してきた。このため、評価の 1 つは、「3 つの簡単さ」により、視覚障害のある児童も晴眼児童もプログラミング体験が可能になるかである。また、「学齢に応じた学び」として、視覚障害のある児童と中高生徒とのプログラミング体験の違い、また、晴眼盲弱を問わず、小中高の各学齢での教材の有用性を評価しなければならない。ここでは、晴眼盲弱を区別せず、様々な学齢で利用できるかという観点から、筆者らのプログラミング教材が視覚障害のある小学生に利用可能か、また、晴眼盲弱の違いで、どの程度の時間で利用可能になるかの検証を行うこととした。具体的には、(1) 視覚特別支援学校の小学生を対象にした実験授業、(2) 不特定多数が参加する技術イベントにおけるプログラミング実験において、逐次処理のプログラムを入力、実行できるようになるまでの時間を調査することとした。

なお、「3つの簡単さ」と今回の機能追加を含む「学齢に応じた学び」との関係やその晴眼盲弱の違いという観点については、実験途上にあるため、4.3節において評価の方針や考察を述べる。

4.1 視覚障害児童の場合：視覚特別支援学校における実験授業

4.1.1 目的

視覚障害のある小学生に対して、開発した移動ロボット教材を用いた実験授業を行い、どの程度の時間でプログラムが作れるようになるか検証する。

4.1.2 方法

2018年3月6日、福岡視覚特別支援学校から小学3年生の理科の授業としてプログラミング教室の実施依頼を受け、他学年からの参加希望もあり、小学生6名（3年生2名（男女各1）、5年生4名（男女各2）；全盲2名（小3女子、小5女子）、弱視4名）に対し、45分の授業を行った。授業では、一人1台のロボット教材のほかに、点字もしくは拡大文字によるA4用紙1枚のテンキーパッドのコマンド配置の説明資料を配布した。点字資料は、弱視2名を含む4名が利用していた。講師2名のうちの一人が授業を進行し、もう一人が児童の補助に付いた。

授業内容は、下学年の小学3年生が参加していることから、表2のシラバス1コマ目の逐次処理までとし、1) 移動ロボットの構造を触れて確認、2) プログラムの入力・実行方法の説明、3) 基本命令、4) 逐次処理、5) 自由課題を行うこととした。学校側の許可を得て、この授業の様子をビデオカメラで撮影し、時間経過と受講者の行動を観察し、評価を行った。四角形の広い机の3辺に児童2名ずつが着席し、残る1辺側にビデオカメラを設置し、児童の手元が撮影できるようにした（図10）。

4.1.3 結果と考察

授業時間経過と児童たちの様子を表3に示す。

視覚障害のある児童でも、初めて教材に触れてから、おおよそ15分程度で、自分自身で考えた逐次処理のプログラムを実行可能になっていたことがビデオで確認できた。そのうちの6分は、ロボットの構造やキーパッドの確認、電源ボタンの確認に要していた。視覚情報があれば、一目で分かる部分である。しかし、視覚情報がなくとも、短時間にプログラミングの実習が行えるようになったことは、

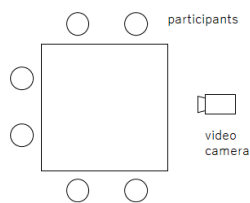


図10 カメラ配置

Fig. 10 Video camera layout.

大きな特徴といえる。

また、授業終了後、参加児童の担任教諭により個別にアンケート調査を行ってもらった。参加者の感想は、以下のものがあつた。「ちょっと難しかったけど楽しかった」、「動かす番号が分かって、動かして面白かった」、「初めてロボットプログラミングができて良かった」、「ロボットの動かし方の仕組みが分かってよかった」、「音で『かえるの合唱』が作れてよかった」、「45分が短かった」などがあつた。自由回答からも、短時間にプログラミングを習得し、プログラミングへの興味関心を引き出すことができていたことが確認できた。

この実験授業では、学齢の異なる小学生を対象とし、また、講師がすべてを教えるのではなく、児童自らが基本命令や逐次処理を見つけだし、理解することを目的とし、達成することができた。一方で、今回の1コマ45分の授業では、これまでの中高生を対象とした実験授業[19]とは異なり、複数回のUターンや四角形周回のプログラムを作る過程で繰り返しの概念自体の気付きには至らず、小学生から制御命令の有無を聞いてくることはなかつた。このことから、児童に対し、繰り返し処理や条件判断の必要性、有用性をどのように気付かせるかという問いかけの仕方が課題になると思われた。たとえば、少ないコードで長時間の面白いダンスプログラムを作ろうといった動機付け

表3 経過観察

Table 3 Experimental class process.

時間	詳細
0'00"	実験授業開始のあいさつ
0'55"	ロボットを触り、車輪や電池、触覚などを確認し始める
2'05"	キーパッドの確認を始める。ボタンの数や「5」ボタンの凸を各自で確認
3'50"	全員がキーパッドの確認終了
4'45"	電源 ON（電源スイッチの確認に手間取る）
5'00"	ボタン押下で鳴動すること、リセット、実行ボタンの説明
7'00"	「2」「5」と順に入力し、実行。前進動作を確認。驚きの歓声。各自オペランドを変更し、動作時間との関係を確認
8'45"	プログラムリセットの説明
9'35"	「5」の下のボタンは？と問いかけ、後退命令を各自確認
10'30"	左右の回転命令を各自で発見
13'30"	最初に小5全盲女子が逐次処理を習得
15'15"	全員が複数の命令を組み合わせた逐次処理を達成
15'20"	各自が自由にプログラミング
19'00"	移動ロボットの基本命令、逐次処理について説明
19'55"	課題挑戦：往復移動、Uターン等
33'50"	ここまでのまとめ（同じプログラムは同じ動き、間違ったプログラムは違う動き）
35'20"	ロボットダンス（1名が自分で考えたプログラム列を読み上げ、全員で入力し、実行）
37'05"	3名成功。間違えました、止まっちゃいましたの発言
40'20"	数名の試行後、小3全盲女子のロボットダンスで全員成功
42'25"	ここまでのまとめ
42'35"	「5」キーの動作を問いかけ、ピーブ命令（音階）を確認
44'55"	女子児童3名と小5男子の4名が作曲に熱中
46'30"	終了

や複数台のロボットダンスで衝突状態を引き起こすなどが考えられる。

また、今回、小学3年生と5年生が同時に授業を受けており、学齢の差の影響も大きいという印象をもった。たとえば、「直角」の概念は小学2年で、角度の単位は小学4年で学習する [20]。逐次処理で実施するUターンを180度回転と角度で説明しても、理解できない児童もいることが説明の際に確認された。算数や数学で学ぶ概念や用語、学習指導要領との関係など、学齢にあわせたシラバスおよび必要な学習時間の検討が課題であることが分かった。

なお、2020年12月11日にも視覚障害のある小1から小6まで6名(全盲3名、弱視3名、男児3名、女児3名)を対象に同じ教材を用いた実験授業を行い、小学1年生でも同様に短時間でプログラミングできることを確認している。

4.2 晴眼児童生徒の場合：不特定多数が参加するイベント

4.2.1 目的

不特定多数の参加者に開発したロボット教材を試用してもらい、アンケート調査を通じ、本プログラミング教材による習得時間や教材の有用性を検証する。授業のような統制された環境ではないが、来歴の異なる児童生徒がそれぞれのペースで教材を使うという状況での結果が得られると考えたためである。

4.2.2 方法

2018年8月4日と5日に行われたモノづくりの展示会(Maker Faire Tokyo 2018)の来場者に本教材のプログラミングを体験してもらった後、アンケート調査を行い、教材の感想のほか、プログラミング習得(逐次処理まで)に要した時間を主観的に答えてもらった。

1回のプログラミングの参加者は4人以下であり、1人1台のロボットを使用し、紙資料としてA4用紙1枚のキーパッドのコマンド配置の説明書を用意した(図11)。



図11 ロボットコマンド説明書

Fig. 11 Leaflet of robot command list.

プログラミングの実施内容は、福岡視覚特別支援学校のときと同様に、シラバス(表2)の逐次処理までを行い、その後は希望に応じて、繰り返しや条件分岐、障害物回避やライトレースなどを体験してもらった。最後にアンケート調査票に記入してもらった。低学齢児の場合は、保護者が記入した。

回答者は、老若男女127名であり、全員が晴眼者であった。内訳を図12に示す。なお、横軸のPSは未就学児、ES、JH、SHはそれぞれ、小中高、UCは大学生を示している。また、児童生徒の多くは、モノづくりに興味のある大人についてきた子供であった。

4.2.3 結果と考察

図13に学齢・年代ごとのプログラミングの習得時間の集計結果を示す。この図では、学齢を未就学と小学校低学年(1,2年:ES-L)、中学年(3,4年:ES-M)、高学年(5,6年:ES-H)、および、中学、高校と区分している。サンプルサイズが小さかった60代は、50代以上にまとめている。調査の結果、未就学児を除くほぼすべての年代が、初めて教材に触れてから平均5分以内に逐次処理までのプログラミングができるようになったと回答した(図13×印

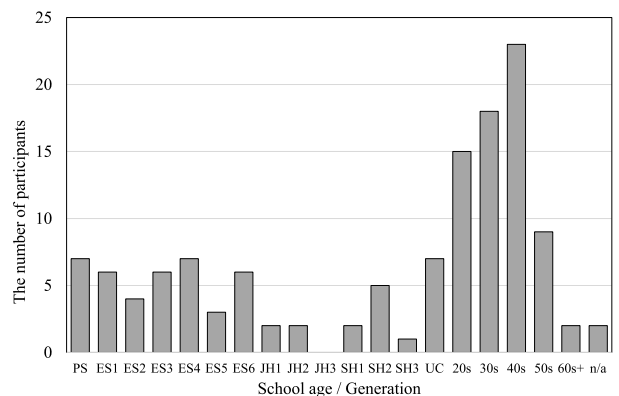


図12 回答者の年齢構成

Fig. 12 School age or generation of participants.

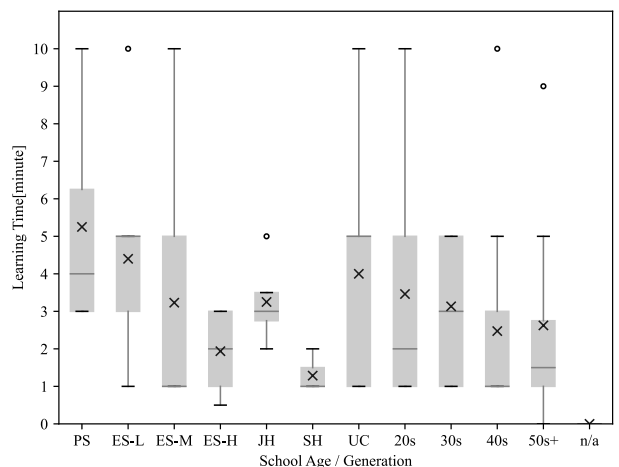


図13 学齢・世代ごとの平均習得時間

Fig. 13 The mean learning time by school age or generation.

部)。なお、この箱ひげ図では、第1および第3四分位数は、それぞれ25パーセンタイルおよび75パーセンタイルとして求めており、四分位範囲の1.5倍を超えた値を外れ値として○で、平均値を×として表現している。外れ値も平均値の計算に入っている。小学校高学年や高校生で平均習得時間が1分程度と非常に短時間になっているが、これは、電源ONと「前進5」の確認の後、講師の説明を聞かず、コマンド説明書のみで理解にたどり着いた参加者がおり、その影響と考えている。以上の実験結果から、小学校低学年であってもPCやマウス、メニューの操作を覚えることなく、十分短い時間でプログラミングを学習できるといえる。一方で、大学生以上の回答では習得時間が短くならず、小学校高学年よりも長い場合もあった(図13中の最大値や外れ値)。このような回答者は、短時間に逐次処理を習得後、繰り返しや条件分岐を試み、ロボットの使用時間も長くなっていた。今回のアンケートでは、「プログラミングができる」という点について、たとえばシラバス(表2)のどの段階まで達したかなど回答意図についての標準化を行っていなかった。そのため「プログラミングができる」という認識が、「逐次処理のプログラムでロボットを動かせる」ではなく、「制御命令を含むプログラミングの概念自体を理解する」のように異なっている可能性がある。

4.3 総合考察

4.3.1 3つの簡単さと小学校低学年からの試行錯誤

まず、2つの実践における観察結果やアンケートから、晴眼盲弱を区別せず、小学校低学年からプログラミングを体験できることを確認した。プログラミング時の児童の様子を観察する限り、以前のPCとの接続が必要な筆者らの教材[19]よりも、短いサイクルで試行錯誤を繰り返していた。PCのスキル不要でプログラミングそのものに注力できており、筆者らが定義した「3つの簡単さ」により、体験を通して学ぶという目的には適していたと考えられる。

一方、今回の実験授業を通して、プログラミングの理解度や教材の評価方法について、今後の課題があることが分かった。今回の実践では、アンケートの結果や具体的なプログラミング課題を実現できるかをもとに理解度を判断した。たとえば、ロボットダンスの作成と説明や独力によるUターン等の課題の解決である。しかし、4.2.3項で述べたようにアンケートにおける到達水準の標準化という点や量的な検証については十分にはできていない。今回はプログラミング習得に要する時間のみを検証したが、実際に学習者がどれくらいプログラムの内容を考え、どのくらい試行錯誤したか、その量的な検証には至っていない。プログラムの総入力行数や実行回数のログ機能を追加するなどして、プログラミングに関する学習者の理解度とプログラミ

ングの体験量との関係を調査するなどが考えられる。

4.3.2 3つの簡単さと状態遷移

キー入力を数字10個に制限したことから、キー入力の状態遷移が必要となり、学習者がすぐに理解できるかが疑問視される。まず、基本命令での状態遷移1, 2(図7)は、命令と数値を1つずつ交互に入力するだけであり、学習者にとっては容易であったことは、ボタン操作に係るアンケート調査から確認できている。また、今回の小学生を対象とした実践では、FOR文やIF文を扱わなかったが、FOR文の繰り返し回数は、基本命令と同じ1桁であり、これまでの中高生の実験では容易に習得することができていた[19]。

状態の遷移数が3つになるIF文については、学習者にとっては各状態の認知が難しそうに思われる。しかし、「もし」、「○○」が、「△△」だったら』という文が理解できていれば、この文のとおり状態が遷移するため、中高生を対象としたこれまでの実験授業では、シラバスどおりに短時間で習得できていた[19]。「もし〜ならば」という仮定の概念が小学校のどの段階で習得できるかが鍵となる。

4.3.3 3つの簡単さによる想定外の動作と学び

本教材の特徴の1つは、モニタやエディタのような編集機能はないものの、PCのスキルが不要であり、短時間でプログラミングの試行錯誤や失敗を繰り返すことができるというものである。ここでの失敗体験が、プログラミング学習に有効に作用するかどうかは、今後検証が必要であるが今回の2つの実践でもその1つと考えられる事象が確認された。具体的には、リセットボタンの押し忘れという失敗によって、意図せず命令が重ねられ、前に入力したコードと連続して実行という逐次処理に気付く児童がしばしばみられたというものである。すなわち、操作ミスをきっかけに、想定外の動作を観察したことで、「逐次処理」というプログラミングの概念を獲得とするきっかけとなったと考えられる。

学習者の想定外の動作を引き起こす仕組みは、本教材の言語仕様上の制限にもある。たとえば、基本命令のオペランドは数値1桁に制限されている。学習者が「前進10」と入力しても、想定した10ステップ前進はしない。学習者が乱数の機能を知る前に「前進0」を実行すると、動かないと予想したロボットが動作し、また、各ロボットで違う動作になる。「FOR 0」は、0回の繰り返しと予想していると、ロボットは止まらなくなる。FOR文による10回以上の繰り返しは、数値の制限により、必然的に入れ子になる。これらは、コンピュータ上での数値の制限や乱数の概念、無限ループ、入れ子の概念を知るきっかけであり、これまでの実践でも児童生徒の驚きとして観察されてきた[19], [24]。このような想定外の動作は、児童生徒の他のプログラミングの学習履歴も影響すると思われることか

ら、今後検証したいと考えている。

最後に、プログラムがすべて数字列として置き換えられることから、写経 [25] が容易という特徴がある。中高で学ぶとしている制御命令を用いる複雑なプログラムも、入力だけなら数字キーを押すだけであり、小学1年生でも障害物回避やライントレースを体験することができる。IF文を学ぶ前に、「もし～ならば」という仮定の概念に気付かせるきっかけにすることができるのではないかと考えている。

5. おわりに

視覚障害のある児童でも利用可能な簡単な10キープログラミングロボット教材を開発した。この教材を用いて、視覚障害児および晴眼児に対して実験授業を行い、その有用性を確認した。すなわち、実証実験では、視覚障害のある小学校低学年生であっても、15分という短時間で直感的にキー操作を習得し、プログラミングと逐次処理まで理解できることが確認できた。

また、ビジュアルプログラミング言語は、初学者向けの教育用言語とされるが、PCやタブレットを必要とし、また各国語にローカル化されなければ利用しにくい。プログラミング教育における母語の有用性が指摘されているが [26]、筆者らの教材は、10個のキーのみを使用し、母語をそのまま用いることができる点でも初学者にとって利用しやすい教材といえる。

もちろん、10キープログラミングですべてのプログラミング学習ができるわけではない。高機能な機械語やアセンブラ言語は、初学者向きではない。3.4節でも述べたように、この教材のプログラム編集機能は弱く、コードの追加と全削除しかない。このため長大なプログラムの作成には不向きであり、晴眼の学習者も暗記できない場合は、紙と鉛筆でコードを記述していた。本教材は、短いプログラムによるコンピュータの動作体験を主としており、短時間の試行錯誤を体験できる一方、学齢が上がり、長大なプログラムが必要となった場合は、文献 [19] と同様、コード編集のためにPCを接続することになる。今回の10キープログラミングでどの程度までのプログラミング学習ができるかを確認していく必要がある。

今回の実践で視覚障害のある児童生徒の場合、ロボット教材の構造の把握に晴眼児童生徒よりも時間を要すること、学齢に応じて習得している概念や学習用語に差があることなど、シラバスの作成にあたっては、児童の特性に応じた時間配分や構成が必要であることが確認された。今後は、小学生の学齢や特性に応じた教育内容を検討し、学齢に応じた段階的な学びをどう構築し評価していくか、引き続き実践を通して調査していく予定である。

なお、実験写真の撮影と使用については、事前に学校および保護者の同意を得ている。

謝辞 本研究を実施するにあたり、参加していただいた福岡県立福岡視覚特別支援学校の児童と先生方に感謝の意を表します。ことに、故沖本教諭には、多大な支援とご協力をいただき、感謝の念が絶えません。

なお、本研究の教材開発に係る部分は、文科省科研費(17K00992)によるものである。

参考文献

- [1] 安浦寛人：情報技術を社会常識にするためには、情報処理, Vol.40, No.1, pp.47–49 (1999).
- [2] 長谷川春生：小学校におけるプログラミング教育の方向と課題, 知能と情報, Vol.30, No.3, pp.137–147 (2018).
- [3] 甲斐康司, 木室義彦, 坂口良文, 安浦寛人：情報社会に生きる小中学生のための計算機の動作原理の教育, 情報処理, Vol.43, No.4, pp.1121–1131 (2002).
- [4] 文部科学省：小学校プログラミング教育の手引(第三版)(2020).
- [5] Bau D., Gray J., Kelleher C., Sheldon J., and Turbak F.: Learnable Programming: Blocks and Beyond, *Communications of the ACM*, Vol.60, No.6, pp.72–80 (2017). DOI: 10.1145/3015455
- [6] Hadwen-Bennett A., Sentance S., and Morrison C.: Making Programming Accessible to Learners with Visual Impairments: A Literature Review, *Intl. Journal of Computer Science Education in Schools*, Vol.2, No.2, pp.3–13 (2018). DOI: 10.21585/ijcses.v2i2.25
- [7] Milne L., and Ladner R.: Blocks4All: Overcoming Accessibility Barriers to Blocks Programming for Children with Visual Impairments, *Proc. of the 2018 CHI Conf. on Human Factors in Computing Systems*, pp.1–10 (2018). DOI: 10.1145/3173574.3173643
- [8] Morrison C., Villar N., Hadwen-Bennett A., Regan T., Cletheroe D., Thieme A., and Sentance S.: Physical Programming for Blind and Low Vision Children at Scale, *HUMAN COMPUTER INTERACTION*, pp.1–35 (2019). DOI: 10.1080/07370024.2019.1621175
- [9] 本吉達郎, 掛橋 駿, 小柳健一, 増田寛之, 大島 徹, 川上浩司：P-CUBE：視覚障害者のプログラミング教育支援用ブロック型ツール, 日本ロボット学会誌, Vol.33, No.3, pp.172–180 (2015).
- [10] 川崎 空, 菅原 研, 松本章代：視覚支援学校における低学年向けプログラミング教育環境の開発, 情報処理学会東北支部研究報告, Vol.2020 (2021).
- [11] Oliveira J., Campos M., Amory A., and Manssour I.: Teaching Robot Programming Activities for Visually Impaired Students: A Systematic Review, *Intl. Conf. on Universal Access in Human-Computer Interaction*, pp.155–167 (2017). DOI: 10.1007/978-3-319-58700-4_14
- [12] 大阪府立大阪北視覚支援学校：視覚障がいのある生徒へのプログラミング授業～キューベットを用いた情報活用能力の育成～, 第44回実践研究助成 特別支援学校, パナソニック教育財団 (2018).
- [13] 長岡英司・宮城愛美：重度視覚障害者用プログラミング環境の開発とその活用, 視覚リハビリテーション研究, Vol.1, No.1, pp.36–40 (2011).
- [14] Ludi S.: Robotics Programming Tools for Blind Students, *Journal on Technology and Persons with Disabilities* (2014). URL: <http://hdl.handle.net/10211.3/121968>
- [15] 小林 真, 滝沢穂高, 大矢晃久：視覚障害の児童生徒用プログラミング教育教材の検討, 信学技報, WIT2019-49, Vol.119, No.478, pp.57–62 (2020).
- [16] 兼宗 進：小中高で必修化されるプログラミング教育と

- それに向けた研究の紹介, 通信ソサイエティマガジン, Vol.13, No.2, pp.92–99 (2019).
- [17] 日本産業技術教育学会編: 小・中・高等学校でのプログラミング教育実践 一問題解決を目的とした論理的思考力の育成, 九州大学出版会 (2018).
- [18] Mountapmbeme A., and Ludi S.: How Teachers of the Visually Impaired Compensate with the Absence of Accessible Block-Based Languages, *ASSETS'21* (2021). DOI: .10.1145/3441852.3471221
- [19] 家永貴史, 江頭尚弥, 寺岡章人, 木室義彦, 山口明宏, 沖本誠司: 移動ロボットとテンキーパッドを利用する視覚障害のある児童生徒のためのプログラミング教材, 信学論 D, Vol.J98-D, No.1, pp.52–60 (2015).
- [20] 大日本図書: 令和3年版 数学の世界 領域・単元別学習内容系統表 (2021).
- [21] 文部科学省: 高等学校学習指導要領 (平成30年告示) 解説 情報編 (2018).
- [22] 保福やよい: なぜプログラミングは難しいのか? 繰り返しの理解構造とCの教科書分析からのアプローチ, 情報処理, Vol.54, No.3, pp.252–255 (2013).
- [23] 木室義彦, 牟田口幸紘, 酒井洋一, 家永貴史: 10 キープログラミング教材による地域 ICT クラブの実践, 情報処理学会 情報教育シンポジウム SSS2019 (2019).
- [24] 木室義彦, 寺岡章人, 家永貴史, 八木博子, 沖本誠司: 視覚障害のある中高生のためのロボットを用いたプログラミング教育, 信学論 D, Vol.J95-D, No.4, pp.940–947 (2012).
- [25] 岡本雅子: ペタ語義: 写経プログラミングをめぐる終わりそうもない論争, 情報処理, Vol.59, No.1, p.81 (2018).
- [26] 大岩 元, 中鉢欣秀: 母語によるプログラミング教育, 情報教育シンポジウム 2015 論文集, Vol.2015, pp.193–198 (2015).



木室 義彦 (正会員)

1986年九州大学大学院修士課程修了。同年同大学助手。1997年同講師。1998年(財)九州システム情報技術研究所研究員。2006年同第3研究室長。2008年(財)九州先端科学技術研究所 生活支援情報技術研究室長。2010年より福岡工業大学情報工学部教授。現在に至る。ロボットシステムおよびマンマシンインタフェースの研究に従事。情報処理学会, 計測自動制御学会, 日本ロボット学会の各会員。博士(工学)。



家永 貴史

2004年九州大学大学院システム情報科学府博士後期課程修了。同年,(財)九州システム情報技術研究所(現九州先端研)研究員。2010年福岡工業大学助教。2015年同大准教授。現在に至る。ICTとRTを活用する支援技術の研究に従事。電子情報通信学会, RSJ, VRSJ, ITE, IEEEの各会員。博士(工学)。