

晴眼盲弱を区別しない直感的なプログラミングロボット教材

木室 義彦[†] 家永 貴史[†] 川下 峻平[†] 岩金 雄輔[†]

福岡工業大学 情報工学部[†]

1. 緒言

今日の情報化社会では、児童生徒もプログラミングを経験し、コンピュータでできることできないことを体験することが重要である[1]。しかし、初学者向けプログラミング教材は GUI を多用し、全盲弱視の児童生徒には利用不可能であった。文科省は小学校でプログラミングを必修化するが、盲学校に限らず、PC やスマホを簡単には利用できない学校現場では、晴眼盲弱を区別しないレベルで利用可能な簡単なプログラミング教材が必要と我々は考えている。

これまで我々は、視覚障害のある中高生を対象としたプログラミング教材を開発し、その有効性を確認してきた[2]。また、全国の盲学校へのアンケート調査により、盲学校小学部でプログラミング教育を行う場合もロボット教材が有用との示唆を得た[3]。これらに基づいて開発したプログラミングロボット教材の仕様と直感的な操作という特徴について報告する。

2. コンピュータの動作原理とプログラミング教育

我々の目的は、プログラミング教育ではなく、コンピュータの動作原理を教えることである。すなわち、コンピュータはプロセッサとメモリとで構成され、メモリに記憶されたプログラムをプロセッサが読み出し、実行する、というものである。これにより、コンピュータはプログラムされたとおりに動作するし、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、コンピュータもできない。児童生徒は、プログラミングを通してこれらを体験、理解する。

初等中等教育で用いられるプログラミング言語は、Scratch のようにマウスを用いるビジュアルプログラミングと呼ばれるものが多い。これらの謳う簡単さとして、以下の3つが挙げられている[4]。

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. 上2つの結果として文法エラーが生じにくい
- しかし、この特徴は、我々の目的には適さない。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、アルゴリズムのミスだけでなく、キー入力ミスや文法ミスもコンピュータ

の動作結果として体験できた方が良い。また、初学者の障壁と考えるとタイピングを除外するよりも、マウスよりも簡単にタイピングができれば良い。すなわち、我々の考える簡単さとは、次の3つである。

1. マウスと GUI に依存するのではなく、キーの数や命令語の長さを減ずることで、入力し易く、覚えやすく、すぐに「タッチタイプ可能」
2. 「直感的」とは、単に「見て分かる」ということではなく、「既に身につけている概念の延長として理解できる」
3. 文法エラーも含めて、プログラムの実行結果を「システム単体で確認」できる。他に PC などの外部機器を要しない

3. ロボット言語仕様と実装

3.1 移動ロボットの命令セット

ロボットへの命令は、移動とビーブ音に関する5つの基本命令、および、繰り返しと条件分岐を実現する制御命令である (Table 1)。

Table 1 Robot programming commands

Cmd	Param	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:inf)
NEXT	-	end of loop
IF	sid & ss	conditional branch block
ENDIF	-	end of IF block
ENDW	-	end of while block (with IF)

基本命令では、1 から 9 の 1 桁の数字で移動や回転の動作時間を、ビーブ音の場合は、音階を指定できる。[基本命令]+[0]の場合は、停止ではなく、実行時に 0~9 の乱数がセットされる。

制御命令の繰り返し処理では、FOR と NEXT に囲まれたブロックを FOR に続く数字 1 桁の回数だけ繰り返す。0 を指定すると無限回となる。

Fig.1(a)は、時計回りに四角形を描くプログラムの例である。また、条件分岐は、IF と ENDIF で囲まれたブロック構造とし、条件文は、IF+センサの識別子とセンサの状態 (ON/OFF) で表す (Fig.1(b))。コード中の“LT”は、左タッチセンサの

An Intuitive Programmable Robot Kit Requiring No Visual Information

Yoshihiko Kimuro[†], Takafumi Ienaga[†], Ryohei Kawashita[†], and Yusuke Iwagane[†]

[†]Fukuoka Institute of Technology

意である。この他、条件付き繰り返しの WHILE 文を用意している (Fig.1(c))。なお、ここでは、FOR や IF など、英文コマンドのように説明したが、日本語でも他言語でも自由に決めてよい。次節で述べるようにそれぞれボタン 1 個に対応しているだけだからである。



(a) loop (b) conditional branch (c) while loop
Fig.1 Sample programs

3.2 プログラム入力方式

初学者の場合、PC やマウスの操作さえも不慣れなことが多い。そのため、プログラムの入力方式も 10 個の数字キーのみの最小限かつ身体的に直感的なものとする事とした。

数字キー配置としては、PC のテンキー、または、電話のダイヤルボタンを用いる。ロボットの動作命令のうち、前進や後退、左右回転は [5] キーを中心とした上下左右に配置する (Fig.2)。テンキー配置とダイヤル配置は数字の並びが昇順か降順かの違いがあるが、数字そのものではなく、キーの空間的な配置に着目することで、直感的な操作を実現する。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。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。実行/停止とリセットは、[0] キーの左右に配置している。

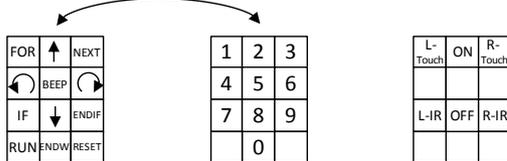


Fig.2 Commands layout and its state transition

3.3 実装

ロボット教材のベースは、市販ロボット玩具 (MR-9172 EK-JAPAN) である。2 つのモータ、2 つの光センサとタッチセンサ、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 (Fig.3)。

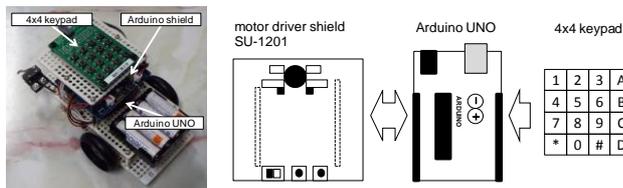


Fig.3 Mobile Robot (KOROBO) + Arduino + 4x4 keypad

前節で設計したロボット言語のインタプリタは、Arduino 上に実装している。キーパッドから入力されたプログラム列は、マイコンの EEPROM にも保持され、電源オフ後も再実行可能である。

4. 実証実験

開発したロボット教材の操作性に関し、アンケート調査を行った。Maker Faire Tokyo 2018 (2018/8/4-5) の来場者にプログラミングをしてもらい、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。回答者は、老若男女 127 名であった。調査の結果、小学生を含むほぼすべての年代で、平均 5 分以内にプログラミングできるようになったと回答した (Fig.4)。図中、大学 2 年生で突出した値になっているのは、逐次処理だけでなく、繰り返しと条件分岐までの習得時間を回答した人がいたためであった。なお、全盲および弱視の小学生と中学生 6 名に対し行った福岡盲学校での実験授業では 15 分以内にプログラミングが習得できている。

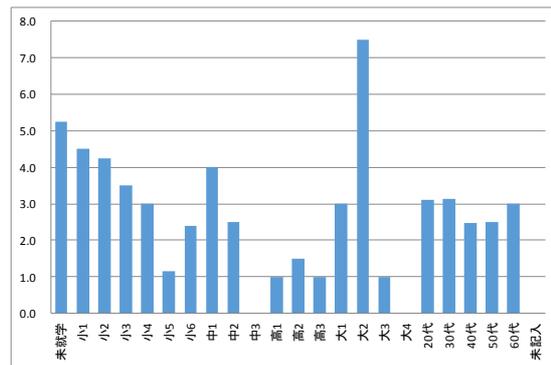


Fig.4 Subjective learning time for programming

5. 結言

視覚障害のある児童でも利用可能なロボットプログラミング教材を開発し、直感的な操作性に関するアンケート調査を行った。この実験により、小学生であっても、数字さえ理解できれば、直感的に短時間でプログラミングを習得できることが分かった。現在、福岡および長崎の盲学校に本教材を提供し、授業の中で有効性を検証中である。なお、本研究は、文科省科研費 (17K00992) によるものである。

参考文献

- [1] 安浦, "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] 木室他, "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
- [3] 木室他, "視覚障害をもつ児童生徒のための携帯電話を介したロボットプログラミング教育の可能性", 電気通信普及財団 研究調査報告書 No.29, 2014.
- [4] D. Bau et.al, "Learnable Programming: Blocks and Beyond", Communications of the ACM, pp. 72-80, 2017.