

物質プログラミング -物質によるプログラムの可視化と開発環境の制作-

八城 朋仁^{1,a)} 迎山 和司^{1,b)}

概要: コンピュータの普及に伴い、プログラミングは一部の専門職の行為だけではなく、広く一般に行われるようになってきた。そのため、使いやすいプログラミング言語が必要となってきた。Scratch というプログラミング言語は直感的に手でブロックを組み立てるようにプログラミングできるため初心者にも簡単に使いやすい。しかし、直接ブロックを手で組み立てるわけではない。そこで、Scratch のブロックを実体のあるブロックとして再現し、ブロックを組み合わせることでプログラミングができる環境の制作を行った。また、制作したプログラミング環境をイベントに出展し、体験者の観察を行った。

キーワード: Scratch, ビジュアルプログラミング, USB カメラ

Material Programming -a Visual Programming Development Environment With Material-

YASHIRO TOMOHITO^{1,a)} MUKAIYAMA KAZUSHI^{1,b)}

Abstract: Increasing popularization of computers, programming is used by not only professionals but also ordinal people. Therefore, we need more easy-to-use programming environment rapidly. Scratch is a good example because beginners can make a program easier such as building blocks on screen. However, Scratch will not be able to assemble the block directly by hand. So, we made real blocks of Scratch and also made a total programming development environment using with real blocks. Additionally, we observed people who use this environment in an exhibition.

Keywords: Scratch, Visual programming, USB camera

1. はじめに

コンピュータが一般に普及すると共に、より使いやすいプログラミング言語が求められるようになってきている。そこで、本研究ではプログラムの要素を実体のあるブロックとすることで、より使いやすいプログラミング環境の制作を目指す。

コンピュータは PC, タブレット端末, スマートフォンなどと挙げられるように、より身近な形で一般に普及してきている。その普及に伴い、プログラミングは一部の専門

職の行為だけではなく、広く一般に行われるようになってきた。その例として、大学の情報教育の授業の一環としてプログラミングを取り入れている事例がある [1]。そのため、専門知識の無い人々でも使いやすいプログラミング言語が必要となってきた。

使いやすいプログラミング言語として、MIT メディアラボが開発した低年齢向けプログラミング教育のための Scratch というものがある。Scratch のコンセプトのひとつとして、直感的に手でブロックを組み立てるようにプログラミングできるため、初心者にも簡単に使いやすいと言う点がある [2]。しかし、あくまでもマウスによる入力を主としており、ブロックを直接手で組み合わせるわけではない。そこで、本研究では Scratch の中でプログラムの要素を

¹ 公立はこだて未来大学大学院
Future University Hakodate

a) g2113032@fun.ac.jp

b) kazushi@fun.ac.jp

表している仮想のブロックを、実体のあるブロックとして再現することにより、より使いやすいプログラミング環境の制作を目指した。

2. 関連研究

ここでは、使いやすいプログラミング言語の研究として、上述した Scratch と、アルゴブロックについて述べる。

Scratch とは PC の画面上で絵を配置し、その絵に対してプログラミングを行うことでアニメーションなどを作成できるプログラミング言語である。プログラミングは、プログラムの要素を表した仮想的なブロックを組み合わせることで行っていくことができる [2]。

アルゴブロックとは、鈴木らによって開発された協同プログラミング学習の支援を目的とするブロック型プログラム言語である。このシステムでは、実体をもったブロックを手で接続することによってディスプレイ画面上の潜水艦の動きをコントロールするプログラムを作成することができる。鈴木らは、アルゴブロックによってプログラミングという本来個人的な作業が共同作業の場に引き出され、結果として学習者は、協同でプログラム言語を操作し、作業の過程を共有し、作業について語り合うことができると述べている [3]。

3. 物質プログラミング

3.1 概要

物質プログラミングは、実体のあるブロックを組み合わせることによってプログラミングを行うことができるシステムである (図 1)。このシステムでは、車型ロボット、または PC の画面上の車を操作するためのプログラムを構築することができる。実体のあるブロックは、Scratch の仮想的なブロックを参考に制作した。ブロックはそれぞれプログラムの要素を表している。以降、これらを命令ブロックと呼ぶ。PC の画面には車が描画されており、命令ブロックによって構築されたプログラムを実行すると PC はプログラムに従い、車を移動させる。PC はまた、Bluetooth 通信によって車型ロボットもプログラムに従って移動させる。プログラムの実行は、PC に付属しているポインティングデバイスを用いた実行ボタンによって行われる。

本システムでは、命令ブロックの組み合わせが、画面上の車や車型ロボットを操作するプログラムとなる。命令ブロックの組み合わせは、USB 赤外線カメラによって PC に入力される画像から認識される。実行ボタンが押されると、PC は認識した命令ブロックの組み合わせから、プログラムを構築する。そして、構築されたプログラムが実行される。

3.2 仕様

本システムでは、USB 赤外線カメラ、赤外線ライト、PC、



図 1 物質プログラミング

Fig. 1 Material programming

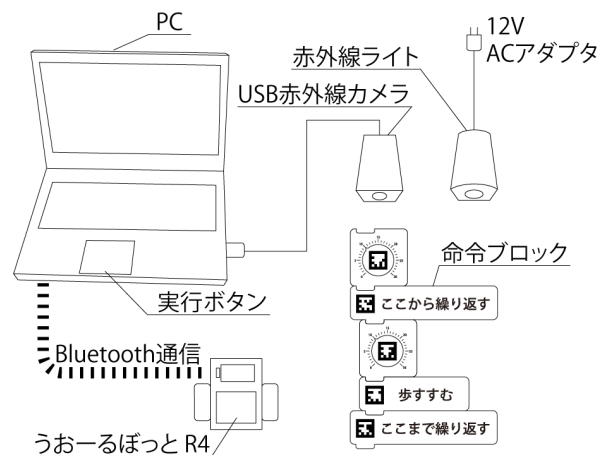


図 2 システム接続図

Fig. 2 System components

そして車型ロボットを使用する (図 2)。USB 赤外線カメラには、フォーカスと絞りが手で調整可能なレンズを取り付けている。システムの事前準備として、このレンズの調整が必要である。赤外線ライトは USB 赤外線カメラの照明として用いる。車型ロボットには、うおーるぼつと R4 を用いる [4]。うおーるぼつと R4 には Bluetooth USB ドングルが取り付けられており、PC との Bluetooth 通信によってデータの送受信が行われている。

3.3 命令ブロック

プログラムの構築に用いる命令ブロックを 4 色の不透明なカラーアクリル板を用いて制作した。命令ブロックには車の動きを操作するための「move」「turn (right)」「turn (left)」、操作を繰り返すための「repeat begin」「repeat end」、操作を条件分岐するための「if begin」「if end」「else」「more than」「less than」、数を表す「変数」の 11 種類がある (図 3)。命令ブロックの形や色そして色によるグループ分けの仕方は Scratch を参考にした。命令ブロックには凹

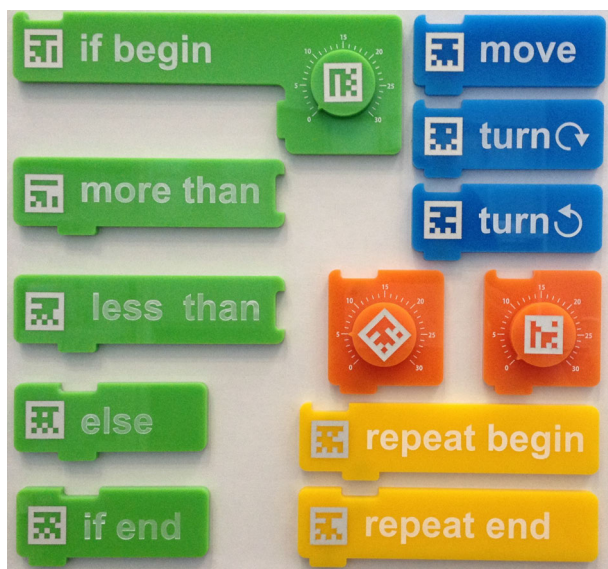


図 3 命令ブロック
Fig. 3 Command blocks

部、凸部があり、それぞれの裏側には磁石を付け、ブロック同士を近づけると凹部と凸部が数珠つなぎで連結する様になっている。また、命令ブロックの組み合わせを認識するために、それぞれの命令ブロックには固有のマークが貼り付けられている。加えて、「if begin」ブロック、「変数」ブロックは他のブロックと異なり、0から30までの目盛が記されたダイヤルが付いている。ダイヤルが示す値によって、命令ブロックを組み合わせた時のプログラムの挙動が異なる。例えば、「変数」ブロックと「move」ブロックを組み合わせた時、「変数」ブロックの示す値は車の移動距離となる。

3.4 命令ブロックの認識方法

命令ブロックの認識はUSB赤外線カメラとPC、そして命令ブロックに貼り付けられたマークによって行われる。PCはUSB赤外線カメラから入力された画像に対し、画像処理を行いマークを検出する。そして、PCは検出されたマークから命令ブロックの組み合わせの情報を抽出する。

マークの検出は、光学的に行われるため、周囲の環境光の影響を受けやすい。そこで、マークの検出率を安定させるために、USB赤外線カメラの照明として赤外線ライトを用いる。また、マークの素材に再帰性反射材を用いる。再帰性反射材には入射光を光源の方向に戻す特性がある。そのため、周囲の環境光からの影響を軽減し、赤外線ライトの効果を高めることができる。

3.5 プログラムの構築と実行

命令ブロックの組み合わせの情報は、PC内でプログラムの構築を行う際に使用される。プログラムの構築、実行の手順をフローチャートとして示す(図4)。

プログラムの構築は、命令ブロックの情報のソート、「変数」ブロックから値への変換、操作を繰り返す部分の展開の順番で行われる。まず、命令ブロックの情報は、USB赤外線カメラで撮影した際、より上部に写っていた順番にソートされる。次に、命令ブロックの組み合わせの中に「変数」ブロックがあった場合、「変数」ブロックのマークの角度と、その他の命令ブロックのマークの角度から、その「変数」ブロックのダイヤルが示す値を計算し、その値を保持する。最後に「if begin」ブロック、「if end」ブロックを検索し、繰り返すことによって省略されている命令を展開する。展開は、「if begin」ブロックの情報の前にある「変数」ブロックの値を繰り返す回数と見なして行う。「if begin」ブロックと「if end」ブロックの情報は展開が終わったときに命令ブロックの情報の並びから削除する。展開は「if end」ブロックの情報が無くなるまで行う。

プログラムの実行は、ソートされた命令ブロックの情報に従って行われる。実行される命令は、それぞれの命令ブロックの情報の前に並んでいる「変数」ブロックの示す値によって度合が変化する。「move」ブロックならば、車の移動距離が変化する。「turn(right)」ブロック、「turn(left)」ブロックならば、車の向きの回転角度が変化する。「if begin」ブロックは自身のダイヤルの値と前に並んでいる「変数」ブロックの示す値を比較し、条件分岐を行う。比較は「変数」ブロックの示す値(比較演算子)「if begin」ブロックの示す値、という形で行われる。比較演算子は、「if begin」ブロックの直後に「more than」ブロック、または「less than」ブロックを組み合わせることで指定される。プログラムの実行が行われると、実行された命令に合わせて車の座標と角度を再計算し、再描画する。この時、Bluetooth通信によってうおーるぽっとR4の制御も行われる。

うおーるぽっとR4とのBluetooth通信は、「move」ブロック、「turn(right)」ブロック、「turn(left)」ブロックの3種類の命令が実行された時のみ行う。PCからうおーるぽっとR4へ送信される情報は、命令ブロックの種類と、速度である。速度は「変数」ブロックが示す値を変換して求められる。うおーるぽっとR4は受信した命令ブロックの種類からタイヤの回転方向を決定し、速度をタイヤの回転速度として動作する。

4. 展示

物質プログラミングを展示し、体験者から物質プログラミングの評価を得た。展示は、2013年5月19日に行われたScratch Day 2013 in Tokyoと2013年11月4日から2013年11月5日にかけて行われたMaker Faire Tokyo 2013の2つのイベントで行なった。

4.1 展示内容

物質プログラミングの展示内容は、2つのイベントで異

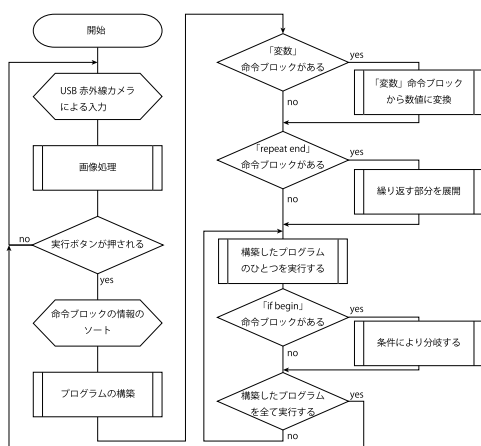


図 4 フローチャート

Fig. 4 Flowchart

なるものだった。

Scratch Day 2013 in Tokyo では通常の USB カメラ, PC, 命令ブロックを使用し, USB 赤外線カメラ, 赤外線ライト, そして, うおーるぼっと R4 は使用しなかった。また, 「if begin」ブロック, 「if end」ブロック, 「else」ブロック, 「more than」ブロック, 「less than」ブロックは無かった。

Maker Faire Tokyo 2013 では, 3.2 章で示した内容で展示を行なった。しかし, 当日の展示会場では電波を使う展示が多く, 通信状態が悪くなかったため, うおーるぼっと R4 と PC 間の通信が出来なかった。そのため, 体験者には PC の画面上の車の操作のみを体験させた。

4.2 観察結果

命令ブロックによって画面上の車の動きが変わることを確認した体験者から「面白い」、「楽しい」という意見を得た。体験者が自ら「変数」ブロックが示す数を変化させたり, 命令ブロックを取り替えてプログラムを再度実行する様子が見られた。特に, 年齢の低い体験者ほど, この傾向が強く見られた (図 5)。また, 年齢の低い体験者の中にはシステムの説明を聞かず, さらにプログラムとして実行せずとも, 何度も命令ブロックを組み合わせてたり, 重ねたり, 裏返したりする人もいた。

5. まとめ

本研究では, プログラムの要素を実体のあるブロックとすることで, より使いやすいプログラミング環境の制作を行った。また, 制作した環境をイベントで展示した。展示では, 実体のブロックを用いることで, プログラミングを行う上で必要であるプログラムの動作を確認しながらプログラムの構築を行う試行錯誤を引き出した。また, システムの説明を聞かず, さらにプログラムとして実行せずとも, 何度も命令ブロックを組み合わせてたりする体験者がいたことから, 命令ブロック自体が体験者にとって興味を惹

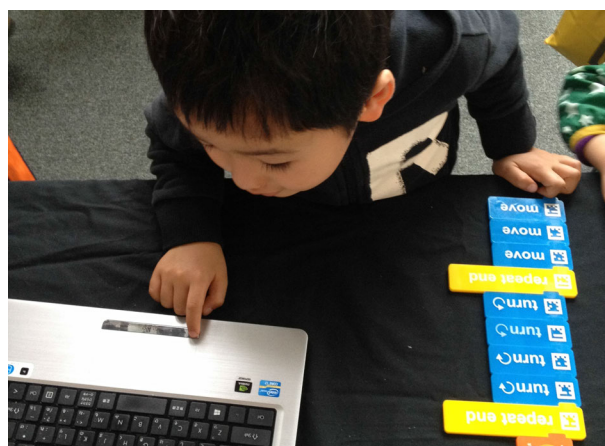


図 5 体験者の様子

Fig. 5 People who use this environment

きつける対象だったと考えられる。これは, 命令ブロックの形や色が要因として考えられる。その他にも, 実体のブロックに対して可能な身体的動作が多かったことも考えられる。

6. 今後の課題

システム上の課題として, 命令ブロックの認識率が低いことが挙げられる。カメラは周辺の環境からの影響を受けやすいため, 今後はカメラを用いない命令ブロックの認識方法を考える必要がある。

また, アクリルを使った実体のブロックは体験者の行動を誘発しやすいことが分かった。特に年齢の低い体験者の興味を惹きつけやすい。これらの理由としてはブロックの色, 形, そして可能な身体的動作の多さなど様々な要因が考えられるため, 実験によって明らかにする必要がある。

今後は, 命令ブロックの認識方法の再検討と, システムの評価実験の計画と実施を行っていく。また, 本システムの有用性を確かめるために日常的な場面への応用を考察していきたい。

参考文献

- [1] 大西建輔：プログラミング, 何をどう教えているか 連載開始にあたって, 情報処理, Vol. 51, No.10, p. 1341, 2010.
- [2] RESNICK M, et al: Programming for All, Communications of the ACM, Vol. 52, No. 11, pp. 60-67, 2009.
- [3] 加藤浩, 有元典文: 認知的道具のデザイン, 金子書房, 東京, pp. 72-73.
- [4] 勝 純一: うおーるぼっとサポートページ, 入手先 <<http://jksoft.cocolog-nifty.com/blog/wllbot.html>> (2013.11.20).