

プログラミング概念の学習教材「Icogram」の開発

Development on the “Icogram”, an environment for learning the concept of programming

山腰 貴大^{*1}, 蘆田 昇^{*1}
Takahiro YAMAKOSHI^{*1}, Noboru ASHIDA^{*1}

^{*1} 福井工業高等専門学校電子情報工学科

^{*1} Department of Electronics and Information Engineering, Fukui National Collage of Technology
Email: i45535ty@stu.fukui-nct.ac.jp

あらまし：小学校低学年の子供たちが楽しく、効率的にプログラミングの基本概念を学べることを目的に学習教材「Icogram」の開発した。「Icogram」の概要、構成及び子供たちの試用結果を報告する。

キーワード：早期プログラミング学習、プログラミング概念理解、カード型デバイス、アニメーション

1. 研究背景

最近、プログラミング講座などの形で子供たちがプログラミングを知り、学ぶことが多くなった。

子供向けのプログラミング教育で使われる学習教材として、(株)jig.jpの福野氏が開発した Progrun や MIT で開発された Scratch などが挙げられる。これらの環境は、様々な初期設定を行う必要がない、プログラムの文法が単純、グラフィック関係を簡単に扱える、といった特徴がある。

しかしながら、これらの環境にはタイピングをはじめとするコンピュータ操作のスキルを必要とする、機能が多すぎて混乱をもたらす、などといった問題点もある。また、短時間で詳細なところまで学習するのは教える側も教わる側も非常に負担が大きい。

そこで、コンピュータが命令した通りに動く、命令した順番に動く、といったプログラミングの基本概念を中心に学習できる教材の必要性を感じ、Icogram と名付けたシステムの開発を行った。

2. Icogram の概要

前述した様に、Icogram (以下、本システム) は本格的なプログラムを作成できることを目的とした学習教材ではなく、プログラミングの基本概念を効率よく学習できることを目指した教材である。従って、本システムはシンプルさと直感的な操作性を追求している。

本システムでは、物理的なカードを並べることでプログラミングを行う。一枚のカードには一つの命令がイラストで描かれている。また、作成したプログラムは画面上のキャラクターを動かすことを目的とする。

これによって、次のことが得られる。

- ・タイピングをする必要がない。
- ・ユーザのレベルに合わせてカードをそろえることが出来る。
- ・プログラミングの過程で文字を読む必要がほとんどなく、直感的に作業を行える。また、ローカライズも容易に行える。

この教材は、児童センターや学校などの公共の場所に設置し、大人の人とともに利用することを想定している。

3. システム構成

本システムは(1)コマンドカード、(2)プログラミングテーブル、(3)ソフトウェアから構成される。

3.1 コマンドカード

コマンドカードはダンボール製で、縦横 4.3cm 程度である。表面には命令がイラストで描かれており、裏面には、命令に対応するコードが描かれている。



図1 コマンドカードの外観

コードは 6×6 のビットデータで構成され、外側の 3 つのビットでカードが置かれた向きを表し、内部の 4×4 のビットで数値を表す。縦一列が 16 進 1 桁を表し、上から 8, 4, 2, 1 の数値を表す。

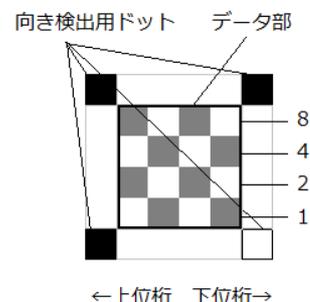


図2 各ビットの役割

3.2 プログラミングテーブル

プログラミングテーブルは高さ 38.5cm 程度であり、天面は透明なアクリル板が載っている。天面の大きさは、縦 30cm、横 45cm である。天面は十字型

のピースで区切られており、これらの間にカードを、イラスト面を上にして配置する。プログラミングテーブル底面に固定した web カメラでカードの裏面を撮影する。カードを配置する部分の奥には、読み取り精度を上げるために光源を備え付けている。



図3 プログラミングテーブルの外観

3.3 ソフトウェア

画像解析、データ化、アニメーションの再生などの処理を行う。GUIは、アニメーションを表示するメイン画面と、様々な設定を行う設定画面で構成されている。子供はメイン画面のみを操作し、各種設定は大人の人が行うことを想定している。ソフトウェアはJavaで開発した。

4. 処理の流れ

カードの画像を解析し、命令列のプログラムを実行するまでの流れは次の通りである。

- (1)カードを置いていないときの画像をあらかじめ取り込む。
- (2)カードを置き、その時の画像を取得する。
- (3)二つの画像を比較して、色の変化が大きい所を検出し、カードが置かれた部分とする。
- (4)検出した範囲のすべてで2値化を行う。あるビットとその周り数十ビットの明るさを比べて値を決定する方法をとっている。
- (5)二値化した画像を解析してコードの数値を得て、配列に格納する。
- (6)得られた配列データでキャラクターを制御する。
 - (1)の処理は設定画面の「画像取得開始」で行う。
 - (2)から(6)の処理はメイン画面の再生ボタンですべて行える。

5. 命令の種類とプログラムの文法

コマンドカードは大きく分けて3種類ある。

1. キャラクターを移動させるカード
(歩く, 走る, ジャンプする)
2. キャラクターの状態を変えるカード
(しゃがむ, 逆立ちする, 元の状態に戻る)
3. プログラムの制御を行うカード
(2回, 3回, 4回, 繰り返す)

状態変化系カードはプログラムが終了するか、ほかの状態変化系カードが現れるまでその状態を保ち続ける。

回数を表すカードと「繰り返す」カードで命令(複

数可)を挟むと、回数分だけその命令を繰り返す。最大2重ループまで対応している。

例えば、図4のカードは左から順番に、「しゃがむ」、「3回」、「ジャンプ」、「繰り返す」命令を表す。このように配置すると、「しゃがんだ後3回ジャンプを繰り返す」結果が得られる。カードの順番と文章で表した動作の順番が一致しているため、直感的にプログラムを作ることができる。



図4 プログラムの例

6. フィードバック

本システムを公共の場で展示し、様々な人に利用してもらい、感想を求めた。その結果、「キャラクターがかわいい」、「簡単にプログラミングを学べる」など、概ね好意的な評価が得られた。

また、「カードを置く向きで命令が変わるといい」という意見を元に、「歩く」、「走る」のカードを改良し、置く向きに応じて移動する向きを変えられるようにした。

プログラミングテーブルの天面に、カードを整理して置けるようにしたピースも、「カードを置くためのガイドがあったらいい」という意見から実現したものである。

7. 考察

7.1 問題点

開発、検証をしている中で、本システムには次のような問題があることが分かった。

- ・プログラミングする前に、デバイスのセッティングや画像処理のための準備が必要であり、手軽にプログラミングすることができない。
- ・コードを正常に読み取れないことが頻繁に発生する。天井にある蛍光灯の光が入り込むとき、著しく読み取り精度が下がる。

7.2 今後の展望

改善すべきこととして、システム面では実用に耐えるほどの読み取り精度にすること、光源がなくてもある程度読み取れるようにすること、などがある。

また、デバイスをほかの用途に流用できるように、コード解析やカメラ制御などの部分のライブラリ化も行いたい。

8. 参考文献

- (1) 酒井幸一: "Visual Basic&Visual C によるデジタル画像処理入門", CQ 出版社(2002)
- (2) 古谷立美: "Java Media Framework プログラミングガイド", <http://fry.no.coocan.jp/lecture/JAVA/JMFyy.pdf>