

【特集 教育における ICT 化】

情報教育におけるプログラミング教育を支援する

情報基盤センター
荻野 哲男

1. はじめに

情報技術を活用した教育支援の歴史は古く、CAI に代表されるような計算機を利用した学習形態の研究は、学習モデルの構築や人工知能、認知科学など広い分野に大きな影響を与えた。一方、教師に代わって計算機が学習活動を進める形態は、その精度の問題のみならず、学習者のモチベーションなどの観点から限界があると認識されるようになった。そこで、最近においては、計算機を主体におくのではなく、教師と学習者をつなぐメディアとして計算機を活用する教育支援が主流になってきている。具体的には、計算機ネットワークを活用して、学習者がいつでも教師に質問したり、学習者同士で協調して作業したりすることを実現する LMS/CMS の研究が盛んである。

このような研究の方向性として、特定の学習内容に特化していくものと、学習活動を一般化することで幅広い学習内容を扱おうとするものが考えられる。後者の研究としては、Moodle のような LMS を対象としたものがあるが、ここでは、情報教育に特化した教育支援について述べる。

2. 情報教育におけるプログラミング教育

情報教育は、今日の情報技術の発展と、社会に与える影響の大きさから、その重要性が特に高まっている分野である。特に、2003 年に高等学校の課程として普通教科「情報」が新しく追加された影響は大きい[1]。2003 年の当初は、教科「情報」の科目として「情報 A」「情報 B」「情報 C」という構成であったが、2013 年から「社会と情報」「情報の科学」の 2 科目に見直されることとなった。

この教科「情報」における情報教育の目的は、以下の 3 つに整理されている。

- 情報活用の実践力
- 情報の科学的な理解
- 情報社会に参画する態度

この目標を達成するために、情報教育の中でプログラミング教育を行うことは、科学的な理解を進めるだけでなく、物事を論理的に考えたり、問題解決の能力を高めたりするという観点で、効果的であると思われる。一方で、問題解決の手段であるはずのプログラミングにおいて、プログラミングやそのプログラミング言語の文法や記法の学習が目的となり、情報技術者を育成するための狭義なプログラミング教育になりやすい問題点もある。

そのため、初等中等教育においては、プログラミング言語自身の学習をほとんど必要としない、タイルスク립ティング型のプログラミング環境を用いることが多い。特によく使われる環境を以下にあげる。

- Scratch
- Logo
- Lego Mindstorms
- Alice

2. 1. Scratch

Scratch は MIT メディアラボで開発されたプログラミング環境である[2]。1970 年から 1980 年にかけてアラン・ケイなどにより開発された Smalltalk 環境の一つである Squeak を用いて開発されており、Squeak eToys などと同じようなコンセプトを持つものである。条件分岐や繰り返し構造を表したブロックを視覚的に組み合わせてプログラミングするスタイルは、プログラミング言語の構文やその書き方を覚える必要がなく、プログラミング初学者に向いている。

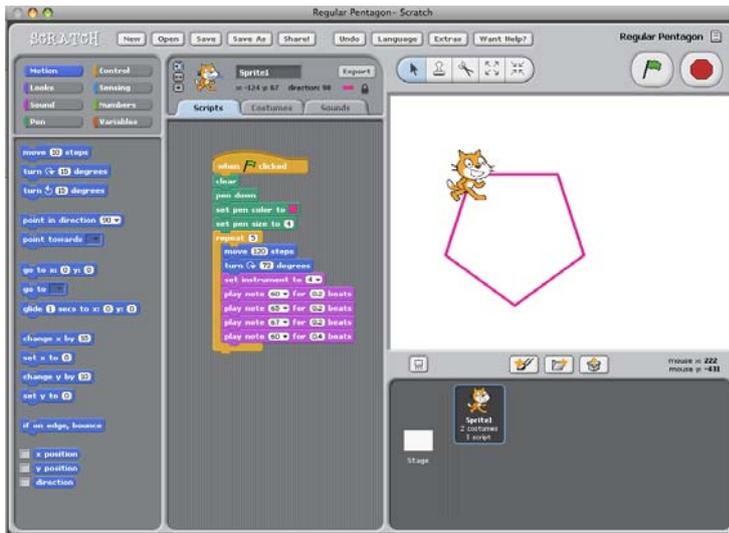


図 1 Scratch の実行画面

特に、学習者が描いた絵を動かしたり回転させたりすることで、簡単にアニメーションを実現させたり、キーボードやマウスからの入力に反応するだけでなく、音や光などのセンサー取り込むことで、実世界のオブジェクトと対話するプログラミングが可能になり、小学校におけるプログラミング授業としての利用が多い[3]。

3. 教育用プログラミング実行環境「ますめ」

教育用プログラミング実行環境「ますめ」は、プログラミング初心者にも扱いやすいグラフィカルなプログラミング環境と、テキストベースの実践的なプログラミング環境の間を埋める、Web ベースの新しいプログラミングとその実行環境である[4]。「ますめ」の最大の特徴は、表計算アプリケーションのセルごとにそれぞれテキストベースのプログラムが記述できることである。「ますめ」の実行画面を図2に示す。

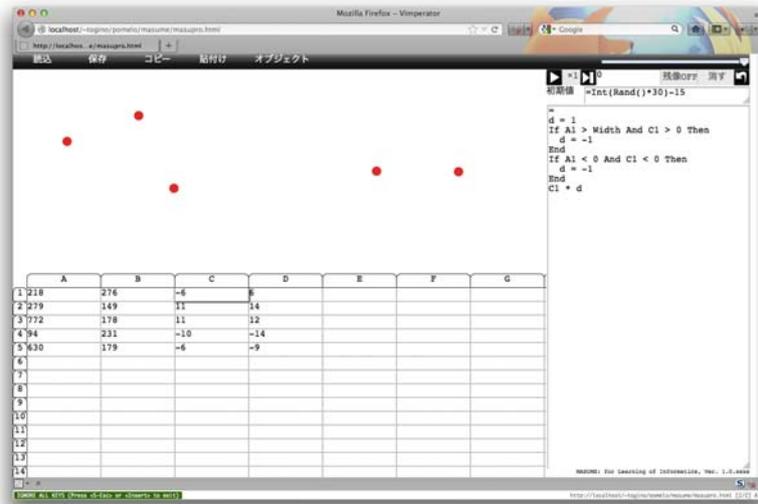


図 2 「ますめ」の実行画面

3. 1. 「ますめ」の設計

情報教育におけるプログラミング教育として、プログラミング初心者を対象とした場合、Scratch のようなブロック型のグラフィカルなプログラミング環境が用いられる。このような環境は、プログラミングの基本的な構造である条件分岐や反復構造などを理解するにはとても効果的であるが、一方で、絵を動かしてアニメーションを作ったり、ロボットを操縦したりして単なる遊びで終わってしまうことも少なくない。そこで次のステップとして、情報活用の実践力を目指し、学習者が設定した課題を、計算機を活用することで自ら解決させてみたり、問題のモデル化やシミュレーションを行わせてみたりすると、グラフィカルなプログラミング環境では実施が困難になる。他方、C 言語や Java 言語などのテキストベースの実践的なプログラミング環境を用いようとした場合、言語自身の学習が目的となってしまう、初等中等教育では、こちらも実施が困難である。

「ますめ」は、高等学校で Microsoft Excel と VBA を活用している事例[5]を参考とし、ブロック型のプログラミング環境から比較的高度なテキスト記述型のプログラミング環境へステップアップするための新しい環境を設計したものである。この設計として、表計算アプリケーションを用いてシミュレーションなどの課題を行う場合、頻繁に値を変更するパラメータや、シミュレーションの結果などの値は、セルに入力することが多い点に着目した。セルを使う理由は、値の変更が容易で、また、その変化の結果がリアルタイムに反映されるからと考えられる。ところが、VBA などのプログラミングからの観点でみると、セルはプログラミングの変数として利用していることになる。つまり、セルが値をリアルタイムに可視化する入出力のインターフェースと、プログラムにおける変数の役割を兼ねている。そこでインターフェースとプログラムの行き来を出来るだけスムーズに行えるよう、セルにプログラミングするスタイルを設計してみることにした。

通常、表計算アプリケーションのセルは、値を保持するだけでなく、数式などを入力することで、その計算結果をセルの値とすることが出来る。「ますめ」では、この方式を拡張し、数式だけでなく一般のプログラムも記述できるようにし、そのプログラムが最後に評価した値をセルの値とすることにした。これにより、セルをプログラムで使用される変数とみなすことで、変数の値が常に画面に表示されることとなり、プログラムの実行によってどのように変化したのかが可視化され、プログラミング初心者の理解を助けることが出来る。

一方、「プログラムによって変数の値を変化させる」という従来のプログラミングの基本的な考え方と比較し「変

数の値はどうか変化するかをプログラムする」という考え方をする必要がある。具体的には、「カウンターが 10 を超えたら、フラグを True にする」というプログラムを、「フラグが True なのは、カウンターが 10 を超えたときである」というプログラムで記述する必要があるということである。このような、変数を主体に考えるプログラミングスタイルは、オブジェクトがルールに従って自律的に変化するプログラムやシミュレーションに適していると考えている。

3. 2. 「ますめ」の特徴

「ますめ」では、基本的な設計であげられる、セルへのプログラミングを含む下記のような特徴があげられる。

- プログラミング変数の視覚化
- 個々のセルにプログラミングが可能
- グラフィックオブジェクトの利用
- ブラウザベースのオープンで柔軟な実行環境

これらの特徴により、Squeak eToys などの環境でプログラミングの基本を学んだ生徒が、C 言語や Java 言語などの実践的なプログラミングとのギャップにつまずくことなく、モデル化とシミュレーションを用いて計算機を活用した問題解決を経験させることが容易になる。

さらに、この環境では、生徒のつまずき等の学習プロセスを分析するため、生徒の操作履歴を詳細に記録できるようにする。

3. 2. 1. プログラミング変数の視覚化

Squeak eToys のようなプログラミング環境が初学者にむいている理由のひとつに、変数の視覚化が考えられる。プログラミングは、命令を逐次実行させるだけでなく、変数の値によって実行する命令を分岐したり反復させたりする。また、そのように制御させるために、変数へ値を代入する必要がある。初学者向けのプログラミング環境では、変数自身がタイルなどの視覚的なパーツとして表現されることが多く、そこに変数の値が表示され、そのタイルを選択することで値の参照や代入できるという特徴がある。

C 言語や Java 言語などの変数が視覚化されていないテキストベースのプログラミングでは、プログラミングする人の頭の中で、変数名を用いてその変数をイメージし、変数の値を管理する必要がある。初学者には、この頭の中での変数のイメージが難しく、慣れた人でも、デバッグトレースを行って、実際の変数の値を出力させながら確認することも多い。そこで、「ますめ」を用いたプログラミングでは、表計算ソフトウェアでいうセルを変数とみなし、変数の視覚化を行う。変数は格子状に画面に配置され、変数の値が常に表示される。

3. 2. 2. 個々のセルにプログラミングが可能

セルが格子状に配置された表計算ソフトウェアである Microsoft Excel では、例えば、セルに =A1+B1 と入力すると、数式とみなして A1 セルの値と B1 セルの値を加算した結果にそのセルの値が変更される。また、その数式で参照されるセルの値が変更されると自動的に再計算が行われる。この機能はシミュレーションを行う際にとっても有効である。

「ますめ」では、より複雑なプログラミングを実現するために、セル1つ1つにプログラミングができるようにしている。すべてのセルは値とプログラムの2つを保持し、プログラムの中で参照している他のセルの値が変更された

ときと、明示的に指示した場合にプログラムが実行される。

シミュレーションにおいて、パラメータとなる変数の値のそれぞれに、その値が変化する条件をプログラミングでき、多くのプログラムが並列的に実行されるという環境は、エージェントベースシミュレーションの考え方をプログラミングするのに適している。

3. 2. 3. グラフィックオブジェクトの利用

ドリトルのようなプログラミング環境では、プログラムの実行結果が視覚的に得られるため、何が起こったのか理解しやすいという特徴がある。具体的には、「かめ太！100 歩く」というプログラムを実行した場合、内部的には「かめ太」の座標を示す値が 100 加算される訳であるが、学習者には、「かめ太」の画面上での表示位置が変化するという動きで理解することとなる。

このように、プログラミング変数が扱う値を数字で表示するだけでなく、グラフィックオブジェクトの位置座標や大きさ・色などに関係付けて表示できる機能が必要である。

3. 2. 4. ブラウザベースのオープンで柔軟な実行環境

高等学校のような計算機環境において、教師が環境を整えるためには、計算機に対する豊富な知識と経験を必要とする。このことが情報教育の中で、モデル化とシミュレーションを生徒に体験させることを困難にしている一つの理由でもある。そのため、以下の点をふまえて「ますめ」の設計を行った。

- 使用する計算機の OS などに影響を受けないこと
- インストールなどの作業が不要であること
- 学校だけでなく、自宅などでも利用できること

計算機の OS としては、Windows をはじめ Mac OS X や Linux などさまざまなものが存在している。多くは Windows であるが、Windows 7 や Windows XP などそのバージョンによって影響を受けることも少なくない。このような環境の影響を最小限に抑えるため、HTML/CSS/JavaScript を用いたブラウザベースの実行環境を採用した。ブラウザは多くの計算機に導入されており、ブラウザの種類によって違いが生じることもあるものの、規格が統一されていく方向にあるため、最も適したプラットフォームであると判断した。

また、ブラウザベースであっても、Flash や Silverlight のようなフレームワークを導入することで、よりリッチな GUI を提供することが可能になるが、逆にフレームワークの導入や管理などの追加の作業が必要になるため、このようなフレームワークを採用しなかった。

ブラウザベースの実行環境とし、その作業をサーバ側に保存できるようにすることで、授業で作成したプログラムを自宅の計算機で開いて続きの作業が行えるようになる。また、サーバ・クライアント方式にすることで、このような自宅での作業に対する操作履歴も記録できるので、分析の対象を広げることができる。

3. 3. 「ますめ」の実装

試作したプログラミング実行環境「ますめ」のシステム構成は、図3のとおりである。

メニューエリア プログラミング作業を中断するための保存や、再開するための読込などのシステムへの操作や、セルを選択した際にそのセルのプログラムを実行するためのメニューボタンなどが配置される。このようなボタンは、選択されたものや状況に応じて自動的に表示される。

4. 今後

「ますめ」自身は開発段階にあり、動作の安定度はまだまだであるが、高等学校の授業での利用を計画しており、生徒がどのような課題に取り組むのかなど検討を進めている。授業での実践を通して、情報教育におけるプログラミング環境についてどのような支援が可能なのか明らかにする予定である。

参考文献

- [1] 文部科学省. 高等学校学習指導要領. http://www.mext.go.jp/a_menu/shotou/new-cs/youryou/kou/kou.pdf.
- [2] Sctrach Team Lifelong Kindergarten Group MIT Media Lab. Scratch -imagine.program.share-. <http://scratch.mit.edu/>.
- [3] 森秀樹, 杉澤学, 張海, 前迫孝憲. Scratch を用いた小学校プログラミング授業の実践 : 小学生を対象としたプログラミング教育の再考 (教育実践研究論文). 日本教育工学会論文誌, Vol. 34, No. 4, pp. 387-394, mar 2011.
- [4] 荻野哲男, 藤岡健史. 教育用プログラミング実行環境「ますめ」における学習者操作の記録・再生機能の実装. 第 28 回全国大会講演論文集, pp. 579-580, 2012-09-15.
- [5] 藤岡健史, 高田秀志, 岩井原瑞穂. 高等学校における squeak を用いた課題解決型情報教育の実践と評価. 日本教育工学会論文誌, Vol. 28, pp. 141-144, 2005-03-20.