

## 教育現場での実践に向けた プログラミング実行環境「ますめ」の試作

荻野 哲男<sup>†1</sup> 藤岡 健史<sup>†2</sup> 柳瀬 大輔<sup>†3</sup>

高等学校では、情報と情報技術の科学的理解とその活用を目的として、モデル化とシミュレーションを題材とした教育が行われている。そのプログラミング環境としては、初心者でも扱いやすい Squeak eToys 等のビジュアルベースのものが多く導入され、一定の教育効果が報告されている。しかしながら、次に C 言語や Excel VBA 等のテキストベースのプログラミング環境へと移行する際、学習者にとってこれらの環境間のギャップが大きく、有機的な学習につながりにくいことが明らかになってきている。そこで本研究では、このギャップを埋めるため、ブラウザベースのオープンソースプログラミング実行環境「ますめ」を新たに設計、試作した。本稿では、「ますめ」の設計と実装および学習のための利用モデルケースについて説明する。

### MASUME: Runtime Environment for Practices of Information Science Education in Classroom

TETSUO OGINO,<sup>†1</sup> TAKESHI FUJIOKA<sup>†2</sup>  
and DAISUKE YANASE<sup>†3</sup>

In high schools, 'modeling and simulation' plays a key role in information science education in order to understand information and information technology scientifically. Many kinds of visual-based programming environments, such as Squeak eToys, in which novice learners can also learn easily are introduced into information science education and it's reported that these programming environments have a certain amount of educational effect. When they shift to text-based programming environment, however, it is proved that the large gap between visual-based and text-based environment interferes with their smooth learning. In this study, we newly develop a browser-based and open-source programming environment called 'MASUME' in order to bury the large gap. In this paper, we explain its concept of design and implementation of 'MASUME' and some model cases of information science education with 'MASUME' in high schools.

### 1. はじめに

2013 年度から高等学校において、新しい学習指導要領<sup>1)</sup>の共通教科「情報」がスタートする。その科目の一つである「情報の科学」では、情報と情報技術の科学的理解とその活用を目的として、モデル化とシミュレーションについて学ぶこととなっている。新しい学習指導要領の解説には、「最終的にはコンピュータによる問題解決とモデル化やシミュレーションとの関係を理解させるとともに、問題解決においてモデル化とシミュレーションの考え方が活用できるようにさせる」とあり、初心者でも実際にコンピュータを用いてシミュレーションを行って理解を深めることが求められている。近年の ICT 分野の成長を鑑みても、高等学校においてこれまで以上にプログラミングを取り入れる動きが出てくると考えられる。初心者にも扱いやすいプログラミング学習環境としては、Squeak eToys や Scratch, ドリトル等が用いられている。これらのプログラミング環境は、ユーザインタフェースがビジュアルベースで直感的に分かりやすく、C 言語や Excel VBA 等のテキストベースに比べ、初心者がプログラミングの基礎的な概念を学習しやすい効果があると報告されている(2) 3) 4) 等)。

### 2. 問題点

プログラミング初心者に適したタイルスク립ティングなど、インターフェースがビジュアルベースであるプログラミング環境は、プログラミングの基本的な概念が容易に習得できる一方、アルゴリズムが少し複雑になったり規模が大きくなるとシミュレーションの実現が困難になる。例えば、タイルスク립ティングの場合、単純な分岐構文であっても画面のかなりの広さを使って表示されることが多く、多少複雑なプログラムでは、画面がタイルで占拠されてしまう。この結果、複雑なアルゴリズムを使うようなモデル化とシミュレーションの教材は少なく、単純な動きでとどまったり、グラフィックの見た目に注力してしまうことも少なくない。

<sup>†1</sup> 神戸大学情報基盤センター  
Information Science and Technology Center, Kobe University

<sup>†2</sup> 京都市立塔南高等学校  
Kyoto Municipal Tonan Senior High School

<sup>†3</sup> 京都市立堀川高等学校  
Kyoto Municipal Horikawa Senior High School

一方、C 言語や Java 言語などの、複雑なアルゴリズムや大規模なプログラミングも可能なテキストベースのプログラミング環境を利用すると、そのプログラミング言語自身の習得が最大の目的になってしまうという問題がある。具体的には、コンパイルエラーで躓いてしまったり、ポインタの理解ができずプログラムができなくなったりすることが多い。

従って、タイルベースのような初学者用の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、情報教育の中で行うモデル化やシミュレーションの実現に特化した新しいプログラミング実行環境が必要であると考えた。

このようなプログラミング実行環境を実現することで、プログラミング言語の習得ではなく本来の目的である、問題解決においてモデル化とシミュレーションの考え方を活用することに時間を多く割り当てることが可能になる。

### 3. 設 計

前章であげた問題点を踏まえ、情報教育におけるモデル化とシミュレーションを実現する新しいプログラミング実行環境である「ますめ」を設計した。設計の方針として、初学者用のプログラミング環境の利点を分析し、その利点を踏まえた上でテキストベースの実践的なプログラミングが可能になる方法を検討した。この結果、設計された「ますめ」の特徴は下記の通りとなった。

- 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躓くことなく、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、生徒の躓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。

#### 3.1 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と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 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時と、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の考え方をプログラミングするのに適している。

#### 3.3 グラフィックオブジェクトの利用

ドリトル<sup>2)</sup>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。

#### 3.4 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。

## 4. 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1のとおりである。

本システムは、すべてブラウザ上で実装され、主に仮想マシンとユーザインターフェース部分から構成される。ユーザインターフェースでは、プログラミングの実現と結果の表示やグラフィックオブジェクトの表示を行い、仮想マシンでは、入力されたプログラムをコンパイルし、実行を制御する。

### 4.1 ユーザインターフェース

「ますめ」の実行画面を、図2に示す。この画面は(1)シートエリア(2)インスペクタエリア(3)キャンバスエリア(4)メニューエリアに分けられる。

**シートエリア** 表計算ソフトウェアと同じように、セルが格子状に配置され、セルが保持する値を表示している。すべてのセルは、値とプログラムの2つを保持しており、セルをクリックによって選択することで、そのセルのプログラムをインスペクタに表示させる

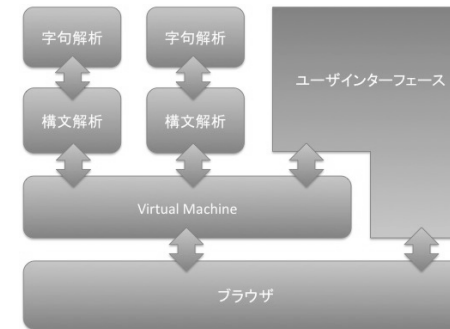


図1 システム構成

ことができる。

**インスペクタエリア** 選択されたセルのプログラムを編集するエリアである。編集中は、キーボードのキーが押される度に、現在のプログラムを仮想マシンでコンパイルし、コンパイルエラーが発生する場合は、セルを赤く表示するなどのフィードバックを行っている。これにより、単純なシンタックスエラーなどは早い段階で気が付き、初心者への支援が可能になる。

記述可能なプログラム言語は、下記で説明する仮想マシンのコンパイラでコンパイルできるものであれば、特に制限はない。本システムを実践する状況に応じてコンパイラを使い分け、使用する言語を変更することも想定している。

**キャンバスエリア** プログラムの中で生成されたグラフィックオブジェクトを表示するエリアである。HTML5のCanvasタグで実装されており、点や円・直線などの基本的図形を描画するコードをJavaScriptのオブジェクトでカプセル化したものが、グラフィックオブジェクトである。すべてのグラフィックオブジェクトは、表示座標のような基本属性を保持しており、その値をプログラムから変更することで、キャンバスエリア内でその位置に表示される。

**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。

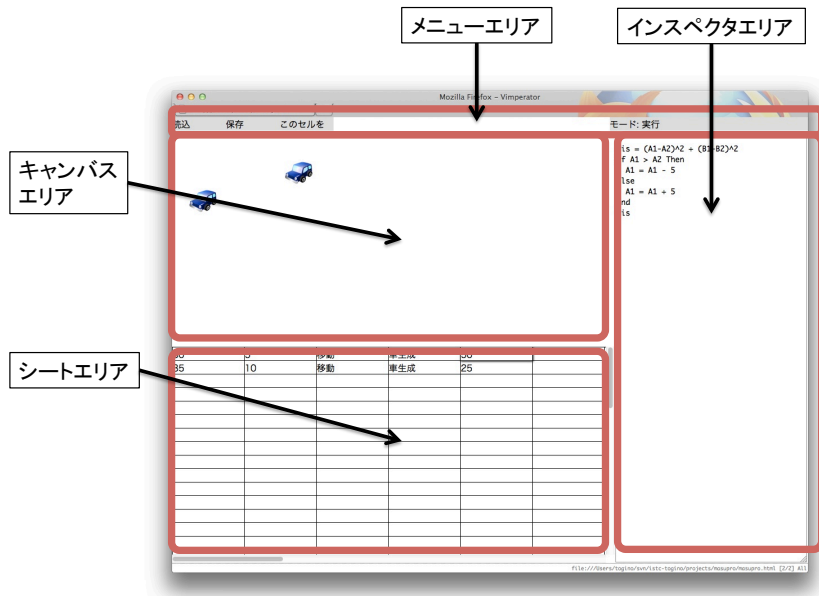


図2 「ますめ」の実行画面

## 4.2 仮想マシン

ブラウザで動作する仮想マシンは、JavaScript で記述されており、下記の機能を持っている。

- セルが保持しているプログラムをコンパイルする
- セルの参照関係を保持し、プログラムの連鎖実行を制御する
- メモリおよびセルやヒープなどのオブジェクトを管理する

### 4.2.1 コンパイル

仮想マシンには、コンパイラが含まれており、構文を BASIC 言語に似せた文法を定義し、yacc を用いて生成した。定義した文法は、以下の通りであり、逐次実行・条件分岐・反復実行などの基本的な制御構文と変数の値参照と代入、式演算、関数呼び出しなどを実装している。

```
eos      : CRLF | ':' ;
start    : stmts ;
```

```
stmts    : stmt
          | stmts eos stmt ;

stmt     :
          | let_stmt
          | if_stmt
          | for_stmt
          | while_stmt
          | dowhile_stmt
          | doloop_stmt
          | expr ;

while_stmt : WHILE expr eos stmts WEND ;
dowhile_stmt: DO WHILE expr eos stmts LOOP ;
doloop_stmt : DO eos stmts LOOP WHILE expr ;
for_stmt    : FOR ID '=' expr TO expr opt_step eos stmts NEXT ;
opt_step    :
          | STEP expr ;

if_stmt     : IF expr THEN CRLF stmts opt_else END ;
opt_else    :
          | ELSE stmts ;

expr       : expr '+' expr
          | expr '-' expr
          | expr '*' expr
          | expr '/' expr
          | expr '^' expr
          | expr OP_MOD expr
          | expr OP_EQ expr
          | expr OP_NE expr
          | expr '<' expr
          | expr '>' expr
          | expr OP_LE expr
          | expr OP_GE expr
```

```

| '(' expr ')'
| NUMBER
| STRING
| ID
| ID_PAR opt_args ')'
| expr '.' ID_PAR opt_args ')' ;
let_stmt : ID '=' expr
| expr '.' ID '=' expr ;
opt_args :
| args ;
args : expr
| args ',' expr ;

```

文法を簡略化するために、字句解析の時点で、空白やタブ文字などを取り除き、大文字小文字を区別しない予約語 (If や While など) の認識を行っている。また、空白を取り除くため、関数呼び出しの ID と '(' の間での空白文字の有無が分からない。そのため、字句解析において ID\_PAR を定義し、空白文字を含まない関数名の識別子と括弧の並びを認識している。

#### 4.2.2 実行

コンパイラによって構文解析されたプログラムは、仮想マシンで実行可能な形式に変換される。図3は、モンテカルロ法を用いて円周率を計算するサンプルプログラムと、それをコンパイルした例である。

「ますめ」では、スタック型マシンを採用しており、下記にあげる命令を処理する。

**CALL** *num id num* 回スタックからポップし、それらを引数として *id* という名前の関数を呼び出し、その結果を、スタックにプッシュする。

**PCALL** *num id num* 回スタックからポップし、それらを引数として、さらにスタックからポップしたオブジェクトに対し、*id* という名前の関数を検索して呼び出す。その結果をスタックにプッシュする。

**LOAD** *id id* をヒープより検索し、見つかったオブジェクトの値をスタックにプッシュする。

**PUSH** *val val* をスタックにプッシュする。

**STR** *id id* をヒープより検索し、スタックからポップした値を、見つかったオブジェクト

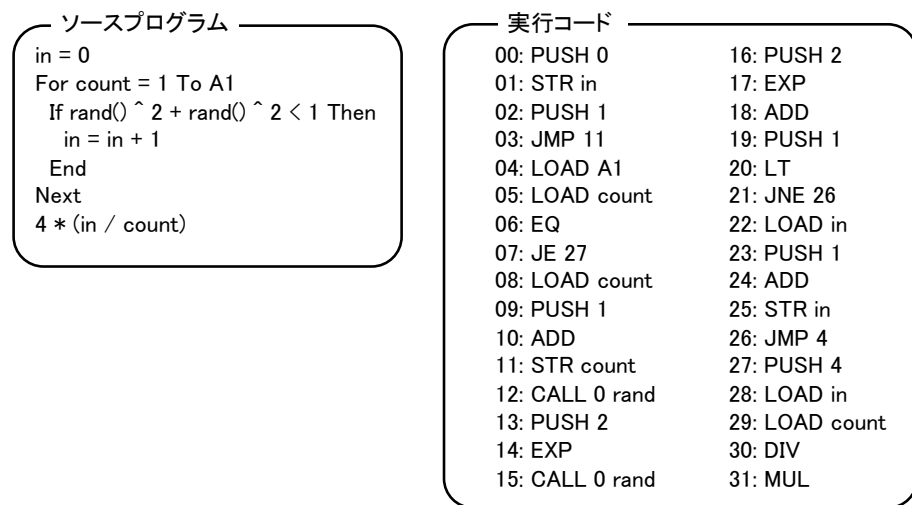


図3 コンパイルの例

の値として代入する。

**PSTR** *id* スタックからポップした値を、さらにポップしたオブジェクトに対し、*id* の名前のプロパティとして代入する。

**ADD, SUB, MUL, DIV, EXP** スタックから2回ポップした値を *y* および *x* として、それぞれ、 $x + y$ ,  $x - y$ ,  $x * y$ ,  $x / y$ ,  $x^y$  をスタックにプッシュする。

**EQ, LT, GT, LE, GE** スタックから2回ポップした値を *y* および *x* として、それぞれ、 $x == y$ ,  $x < y$ ,  $x > y$ ,  $x <= y$ ,  $x >= y$  の結果を真または偽でスタックにプッシュする。

**JE** *line* スタックからポップした値が真であれば、*line* に飛ぶ。

**JNE** *line* スタックからポップした値が偽であれば、*line* に飛ぶ。

**JMP** *line line* に飛ぶ。

「ますめ」の特徴として、セルのそれぞれがプログラムと値を保持している点あげられる。セルの値は、そのセルのプログラムを実行した結果である。また、プログラムは他のセルの値を参照することができ、セルの値が変更されるとそのセルを参照しているプログラムが実行される。この連鎖実行の結果、複数のセルのプログラムにおいて循環参照が存在する

場合、無限に実行が続いてしまう。

このため、仮想マシンでは循環参照による実行を検出し、その時点で実行を中断したり、ステップ実行のように利用者の指示に従って継続実行したりするモードを持っている。

## 5. おわりに

本研究では、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新しいプログラミング実行環境である「ますめ」の設計と試作を行った。この環境を利用することで、高等学校の教科「情報」で求められる、問題解決においてモデル化やシミュレーションを活用するという目的を実現することが可能になると考えている。

今後の課題としては、実際の教育現場で用いるための具体的な教材の開発が挙げられる。「ますめ」が最も得意とする内容は、複数の対象がそれぞれのモデルに従って状態を変化する場合に、パラメータを変えながらシミュレーションを行うものである。このような形で扱える対象はたくさん存在するが、教科「情報」の授業として、生徒が興味をもち主体的に問題解決を行えるようにするには、数多くの実践が必要であると考えられる。今後、高校の情報科教員と共同で継続的に研究を行っていく。

## 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](http://www.mext.go.jp/a_menu/shotou/new-cs/youryou/kou/kou.pdf).
- 2) 兼宗 進, 御手洗理英, 中谷多哉子, 福井眞吾, 久野 靖：学校教育用オブジェクト指向言語「ドリトル」の設計と実装, 情報処理学会論文誌. プログラミング, Vol.42, No.11, pp.78-90 (オンライン), 入手先(<http://ci.nii.ac.jp/naid/110002726068/>) (2001-11-15).
- 3) 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 (オンライン), 入手先(<http://ci.nii.ac.jp/naid/110002983384/>) (2005-03-20).
- 4) Fujioka, T., Takada, H. and Kita, H.: What Does Squeak Provide Students with?— A Comparative Study of Squeak eToy and Excel VBA as Tools for Problem-Solving Learning in High School—, *Creating, Connecting and Collaborating through Computing, International Conference on*, Vol.0, pp.42-49 (online), DOI:<http://doi.ieeecomputersociety.org/10.1109/C5.2006.43> (2006).