

# 演示プログラミングを用いたプログラミング学習支援

中田 豊久<sup>1,a)</sup>

**概要:** 演示プログラミングとは、作成したいプログラムの動作をユーザがコンピュータに対して演じて見せて、そこからコンピュータが自動的にプログラムを作成することである。この演示プログラミングをプログラミング学習に利用することを本研究の目的とする。学習の題材は迷路ゲームとし、演示プログラミングによる生成されたプログラムが、学習者のプログラミング理解度を向上させるかを検討する。

## Programming Learning using Programming by Demonstration

TOYOHISA NAKADA<sup>1,a)</sup>

**Abstract:** Programming by demonstration is a method to create program by user's demonstration. The purpose of the study is to use the programming method in order to enhance learning of computer program. In this study a creation of searching goal on a maze is used for learning, and effect of the learning is discussed.

### 1. はじめに

#### 1.1 演示プログラミングとは

演示プログラミングとは、作成したいプログラムの動作をユーザがコンピュータに対して演じて見せて、そこからコンピュータが自動的にプログラムを作成することである。エディタや表計算ソフトウェアのマクロ機能と類似しているが、次の点で異なる。マクロは操作を記録して再生しているだけであり、マクロを作成した時とは異なる状況では機能しないことが多い。一方、演示プログラミングは、ユーザの意図を推定してプログラムを作成する。よって、マクロよりは汎用性の高い動作が期待できる。このような演示プログラミングは、プログラミングの知識を必要としないプログラミング方法として、Programming by Demonstration[1], Programming by Example[2][3] などの名称で研究が行われている。

#### 1.2 演示プログラミングをプログラミング学習に用いる

「この演示プログラミングをプログラミング学習に用い

ること」を本研究では目的とする。プログラミングの学習は近年、低年齢化および広範囲化している。Scratch [4], Stagecast Creator[5], Viscuit [6] のようなプログラミングを誰にでも気軽にできるようにする研究は、これらの環境の変化によってさらに重要性を増していると考えられる。しかしそれでもなお、プログラミング教育の現場では、プログラミング習得を挫折する人がいる。

本研究では、これまでのプログラミング学習方法で挫折してしまった、特にプログラミングに向いていない人に対する学習方法を検討する。学習方法の全体としては「ルールの抽出と適用」の練習に注目する。そしてその中のルールの抽出において、解答を見つけれない学習者に対するアドバイスの1つとして、この演示プログラミングを利用する。

### 2. プログラミングに向いていない人

#### 2.1 多くの人が最初のプログラミング授業で挫折する

Richardら [7] は、学力のレベルや専攻に関係なく、初めて受けるプログラミングの授業である程度の人数が挫折すると述べている。この挫折する人とは、ある現象を見てそこからルールを抽出する能力と、そのルールを別の状況でも正しく適用できる能力が欠けている人であるという。

<sup>1</sup> 新潟国際情報大学 情報システム学科  
Department of Information Systems, Niigata University of International and Information Studies, Japan

a) nakada@nuis.ac.jp

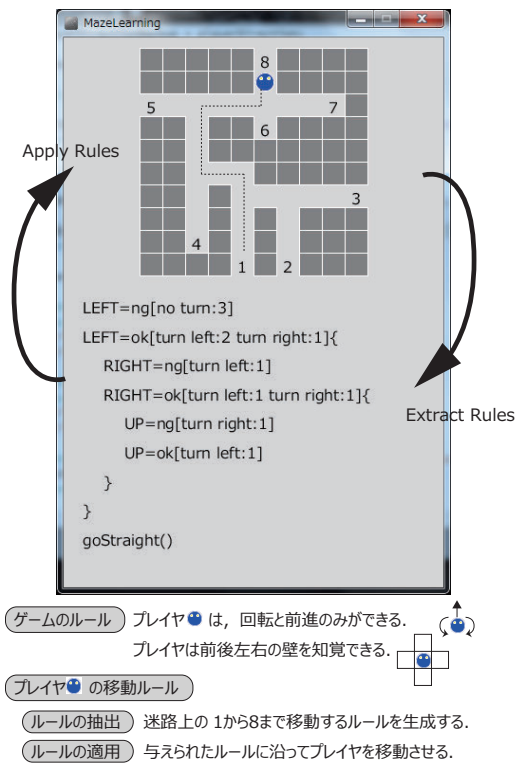
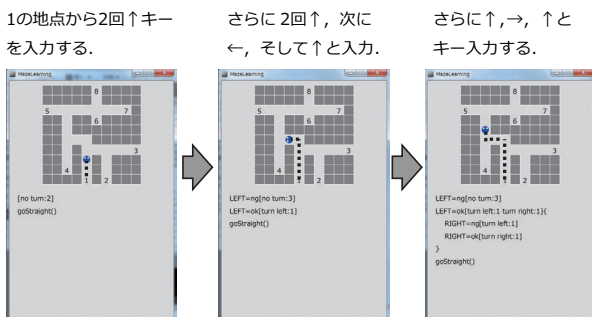


図 1 ルールの抽出と適用を繰り返し行うことにより、プログラミングを学習する。

Fig. 1 Learning computer programming by repetitive extraction and application rules

### 1. プログラミング学習者が手動で迷路上のプレイヤーを動かす。



### 2. 逐次的にプレイヤーの移動ルールが自動生成される。

```

no turn;
go straight;
if(left side==closed){
  no turn;
}else{
  turn left;
}
go straight;
if(left side==closed){
  no turn;
}else{
  if(right side==closed){
    turn left;
  }else{
    turn right;
  }
}
go straight;

```

図 2 プログラミング学習者が手動で動かした迷路上のプレイヤーから、その移動ルールを逐次的に自動生成する。

Fig. 2 When a learner of programming moves player on the maze, the rule of the movement is automatically created one by one.

## 2.2 抽象化能力とプログラミング習得

Perrenet ら [8][9] は、プログラミングには抽象化能力が

重要であると主張している。アルゴリズムの理解をどの程度の抽象化で考えているかによって、プログラミング習得の度合いが計れるという。しかし抽象化能力とプログラミング習得には相関がないとする研究 [10][11] もあり、プログラミングにおける重要な要素は、まだ確実なものに分かっているわけではない。

### 2.3 迷路ゲームにおけるプログラミングに向いていない人の例

例えば図 1 のような迷路を探索するプログラムを作成することを考える。迷路上のプレイヤーは、前後左右に壁があるかどうか分かるようになっていとする。そしてスタート地点とゴールが与えられ、ゴールまでの移動ルールをプログラムとして記述することが求められる。例えば図 1 においてスタート地点を 1、ゴールを 8 としたときの移動ルールは、同図の下に書かれているプログラムとなる。プログラミングに向いていない人は、例えばスタート地点を 1 としゴールを 2 としたときのような簡単な例であっても、プログラムが作成できないことがある。

しかしプログラムを作れない場合であっても、ゲームコントローラによって迷路内のプレイヤーを一歩ずつ動かす方法ならばゴールに辿り着くことができる。つまりプレイヤーの動きはイメージできるわけであるが、そこからルールを導き出せないわけである。これは、Richard ら [7] のいう「ある現象を見てそこからルールを抽出する能力」が足りていないことを示している。

### 3. 「ルールの抽出と適用」を迷路ゲームによって練習する

「ルールの抽出と適用」を繰り返し練習することによって、Richard ら [7] のいうプログラミングに向いていない人に対してプログラミング教育を行うことを提案する。図 1 にその例を示す。この迷路ゲームは、丸で表されているプレイヤーを指定した場所から指定した場所まで移動させるゲームである。プレイヤーは、その場で回転することと前進することしかできないとする。またプレイヤーが知覚できる情報は、前後左右に壁があるかどうかだけである。このゲームによって次の練習を行う。

#### (1) ルールの抽出

- (a) 自動で動くプレイヤーからそのルールを特定する。
- (b) 指定した場所から指定した場所まで移動するルールを作る。

#### (2) ルールの適用

- (a) 与えられたルール通りに、ユーザがプレイヤーを移動させる。

## 4. 演示プログラミングによるルール生成支援

### 4.1 スタートからゴールまでの移動ルールを見つけられない時のアドバイス

プログラミング学習の初心者やプログラミングに向いていない人は、迷路ゲームにおいて指定したスタート地点からゴールまでの移動ルールを記述できないことが多い。しかしそのような場合であっても、例えばゲームコントローラによって迷路上のプレイヤーを一步步進めてゴールまで到達させることはできる。

そこで学習者が手動で迷路上のプレイヤーを一步步動かすごとに、そこまでの移動ルールを自動的に生成し学習者に提示することを提案する。

### 4.2 演示プログラミングによるプログラムの自動生成例

図2は、ユーザがキーボード操作でプレイヤーを一つずつ移動し、そこまでのプレイヤーの行動を自動的にルール化して表示している例である。まず最初に1の地点からスタートし、前進する。このときのプレイヤーの移動ルールは、プレイヤー周りの環境に関係なく常に前進するとなっている。次に最初の突き当たりに着いて左に向きを変える。ここでは、左側が壁が無く通行可能であれば左に行く、というルールが自動的に生成されている。そして再度突き当たりに着いたときに、右に曲がっている。このときには、プレイヤーの左側が通行可能でも、右に行けるならば右に行く、というルールに変わっている。

### 4.3 演示プログラミングの期待される学習効果

ユーザは、自らプレイヤーを動かして一つずつプレイヤーの移動ルールを確認することができる。迷路の分岐点に着いたときに、これまでに作成されたルールと、これから作成したいルールを照らし合わせ、ルールを統合していくことを学ぶことができる。

これは、問題を分割して考えていることになる。スタート地点からゴールまでの移動ルールを見つけられない学習者は、スタート地点からゴールまでを一気に考えていると思われる。そして問題が大きすぎるため、理解できずに解答を見つけられない。そこでキーボード操作で一步步迷路上のプレイヤーを移動させて、そのつど自動生成されるプログラムを1つずつ理解していくことにより、最終的にスタート地点からゴールまでの問題を解けるようになると考える。

## 5. おわりに

本研究では、プログラミングに向いていない人に対する学習方法として、ルールの抽出と適用を繰り返し行うことが良いと考え、そしてその中のルール抽出を支援する機能

として、ユーザが実際に動作させた行動からその行動を自動的にプログラミングする演示プログラミングについて提案した。

今後は、この提案を実際にプログラミング学習の現場で実施することと、このルールの抽出と適用の出来具合によってプログラミングの素養があるのかどうかを判定することに対しても考えていきたい。

## 参考文献

- [1] Cypher, Allen: Watch What I Do: Programming by Demonstration, Daniel C. Halbert, MIT Press, ISBN 0-262-03213-9, (1993).
- [2] Halbert, Dan: Programming by Example, U.C. Berkeley (PhD diss.). Retrieved 2012-07-28, (1984).
- [3] Aditya Menon, Omer Tamuz, Sumit Gulwani, Butler Lampson, Adam Kalai: A machine learning framework for programming by example, The International Conference on Machine Learning (ICML), (2013).
- [4] John Maloney, Mitchel Resnick, Natalie Rusk, Brian Silverman, and Evelyn Eastmond: The Scratch Programming Language and Environment. Trans. Comput. Educ. 10, 4, Article 16 (November 2010), 15 pages. (2010).
- [5] Jill Denner, Linda Werner, and Eloy Ortiz: Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts?. Comput. Educ. 58, 1 (January 2012), 240-249, (2012).
- [6] Y. Harada: Computer Programming Education Using the Visual Programming Language Viscuit, NTT Technical Review, Vol. 8, No. 11, 2010.
- [7] Richard Bornat, Saeed Dehnadi, and Simon: Mental models, consistency and programming aptitude, In Proceedings of the tenth conference on Australasian computing education - Volume 78 (ACE '08), Simon Hamilton and Margaret Hamilton (Eds.), Vol. 78. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 53-61, 2008.
- [8] Perrenet, J., Groote, J. F., and Kaasenbrood, E., Exploring students' understanding of the concept of algorithm: levels of abstraction, 10th Conference on Innovation and Technology in Computer Science Education, Lisbon, Portugal, pp.64-68, (2005).
- [9] Perrenet, J. and Kaasenbrood, E., Levels of abstraction in students' understanding of the concept of algorithm: the qualitative perspective, 11th Conference on Innovation and Technology in Computer Science Education, pp. 270-274, (2006).
- [10] Bennedsen, Jens and Caspersen, Michael E.: Abstraction ability as an indicator of success for learning object-oriented programming?, SIGCSE Bull. vol.38, no.2, pp.39-43, (2006).
- [11] Bennedsen, Jens and Caspersen, Michael E.: Abstraction ability as an indicator of success for learning computing science?, Proceedings of the Fourth international Workshop on Computing Education Research, ICER '08, pp. 15-26, (2008).