

ショートペーパー

フィジカルプログラミングを用いた 初心者向けプログラミング学習システムの開発

Dennis Stritzke^{1,a)} 前田 佑太² 岡本 浩行^{2,b)}

受付日 2015年2月12日, 再受付日 2015年5月8日/2015年7月17日,
採録日 2015年9月5日

概要: 近年, ヴィジュアルプログラミング言語などのプログラムを簡単に作成できるプログラミング言語が開発され, プログラム初心者にとってプログラミングは容易なものとなってきている. このような言語を用いてプログラミング学習を行うことで, 早い段階からプログラミングに興味を持つことができ, 将来における情報通信技術社会を支える人材育成に貢献できると考えられる. 本研究では, タンジブルユーザーインタフェースを利用して物理的なブロックによるプログラミングが可能な初心者向けのプログラミング学習システムを開発した. 開発したシステムは迷路を題材として, 迷路のゴールまでの手順についてのプログラムを考え, ブロックを用いてプログラミングするフィジカルプログラミングにより順次処理について学ぶことができる. また, タンジブルユーザーインタフェースを用いることで, 個人による学習だけでなく, グループによる学習が可能である. 作成するプログラムに条件を与えることでプログラミングに必要な論理的な思考力を身につけることができ, またグループ学習ではより高い学習効果を得ることができると考えられる. システムの評価をアンケートにより行い, 開発したシステムはシステムの目的である初心者が興味を持ってプログラミング学習可能なシステムとなっていることを確認できた.

キーワード: 初心者用プログラミング学習システム, タンジブルユーザーインタフェース, フィジカルプログラミング, グループ学習, 情報通信技術人材育成

Development of a Programming Learning System for Beginners by Using Physical Programming

DENNIS STRITZKE^{1,a)} YUTA MAEDA² HIROYUKI OKAMOTO^{2,b)}

Received: February 12, 2015, Revised: May 8, 2015/July 17, 2015,
Accepted: September 5, 2015

Abstract: We have developed a programming learning system for beginners by using physical programming. We have adopted a tangible user interface for using in a group learning. Users of the system can learn concepts of programming by thinking the program that moves a robot to goal, with interest and actively. A maze at any levels can be created. Physical programming is implemented by putting a block with a control marker. Programming learners using developed system can acquire high learning effect of logical thinking. We have confirmed that objects of the system have been realized, by questionnaire surveys. The system would contribute to human resource development for information and communications technology.

Keywords: programming learning system for beginners, tangible user interface, physical programming, group learning, human resource development for ICT

¹ オストファリア応用科学大学

Ostfalia University of Applied Sciences

² 国立高等専門学校機構阿南工業高等専門学校

National Institute of Technology, Anan College, Anan,
Tokushima 774-0017, Japan

a) dennis@stritzke.me

b) okamoto@anan-nct.ac.jp

1. はじめに

情報通信技術 (ICT) の急速な発展にともない, ICT 社会は今後さらに推進すると考えられる. ICT 社会において, ICT を活用した機器やサービスに触れる機会は増大することから, ICT に関する知識や技能を身につけることは非常

に重要であり、早い段階から ICT に関する知識や技能を学ぶことが必要とされている [1]。また ICT を使用するだけでなく、自ら創造し、実装できる ICT 人材を育成することも今後の ICT 社会の推進にとって非常に重要である [2]。

本研究は将来の ICT 人材を育成するためにグループ学習が可能な特徴を持つタンジブルユーザインタフェース (TUI) [3], [4] を用いてプログラミング初心者がプログラム作成の目的を明確に意識でき、プログラミングを簡単に行え、プログラミングに対して苦手意識を持たないような学習システムを開発することが目的である。開発したシステムはマークを記述した物理的なブロックを利用する TUI を用いて、ブロックを組み合わせることでプログラミングが簡単に行えるフィジカルプログラミングを用いる。またプログラミング初心者がプログラミングの目的を明確に意識できるよう迷路を題材として、迷路のゴールまでの手順をプログラミングすることによりプログラムの構造や考え方を学習でき、またプログラミングに必要な論理的思考力の育成ができるシステムとする。

2. 既存のプログラミング学習ツール

ICT 人材の育成を支援するためのツールはすでにさまざまなものが開発されている。その 1 つとしてプログラミング学習ツールがある。早い段階からプログラミングの基本に触れることができるプログラミング学習ツールとしてマサチューセッツ工科大学が開発した Scratch [5], NTT コミュニケーション科学基礎研究所が開発した Viscuit [6], Google 社が開発した Google Blockly [7], Microsoft 社が開発した Kodu [8] などがある。これらのツールは C 言語や Java 言語などのテキストプログラミングと異なり、ビジュアルプログラミング言語と呼ばれる [9]。テキストプログラミングには可読性が低いことや覚えることがたくさんあるなどの問題があり、初心者が学習するには難しいため [10], 苦手意識を持つことがある [11]。プログラミングなどの学習を行う場合、初心者にとって楽しく学べることは重要であり、学習する内容に苦手意識を感じている場合は学習を持続することは難しい [12]。そのため、初心者がプログラミングを簡単に行えて苦手意識を持たないことは学習を継続するために重要となる。ビジュアルプログラミング言語の特徴として簡単な操作により、さまざまなことをプログラミングで実現することが可能であり [13], プログラミングの基本的なことを楽しく学ぶことができる。これらツールは小学生を対象としたプログラミングの授業に用いられ、その教育効果などが報告されている [14]。

さらにプログラミングに対し興味を持たせるためにはツールとしてのおもしろさを提供する必要がある [15], プログラミングの目的を明確にして積極的に取り組める題材を提供する必要がある。Google Blockly は迷路やパズルなどを題材としており、プログラミングの目的を明確にする

ことで、途中で失敗してもそれぞれツールの目的に対して繰り返しプログラミングできることから、プログラミングの学習効果に影響があると考えられる [16]。

またプログラミングなど、学習を行う場合に個人で学ぶよりもグループで学ぶことで学習効果を向上させることができることが報告されている [17], [18]。Scratch などのプログラミング学習ツールは個人の学習を基本としており、グループ学習に用いる場合は入力デバイスの使用を交代で行うなどの工夫が必要となる。物理的なものを用いてコンピュータへの入出力を行う TUI は、ビジュアルプログラミング言語などで利用されているグラフィカルユーザインタフェースを利用するよりも理解しやすく [19], グループによる操作が可能となることからグループ学習を支援できるという特徴を持つ [3]。TUI をプログラミング学習ツールのインタフェースとして利用することにより、物理的なものによりプログラムを組み立てられるため初心者には使いやすい [20], グループ内で検討しながらプログラミングを行うことでお互いに教えあうことが可能となり、学習効果に影響を与えることができる [18]。TUI を利用したプログラミング学習ツールとして、木製のブロックを利用してロボットを制御するシステム [19], LED ブロックを利用したシステム [21], ビジュアルプログラミング言語である Scratch や Kodu をブロックや物理的なタイルで操作可能にしたシステム [20], [22] がある。

しかし、木製のブロックや LED ブロックなどの特別なデバイスを利用する場合は環境構築にコストを要するという問題がある。また Scratch や Kodu に TUI を利用する場合、さまざまなことがプログラミングできるため、初心者にはプログラミングの目的が明確でなく、最初は戸惑うことが考えられ、自分で目標を定めてプログラミングできるまでは何らかの工夫が必要である。そこで本研究ではこれらの問題を解決するシステムを提案する。

3. 提案するシステム

提案するシステムで使用するブロックはプリンタなどで印刷し、厚紙に貼り付けるだけで作成できるため、低いコストで環境の構築が可能である。またシステムでは迷路をプログラミングの課題として与えて目的を明確にすることで初心者が戸惑わずに取り組める。システムはオペレーティングシステム ubuntu 14.04 上で開発を行った。使用したプログラム言語はユーザインタフェース部分に Python2 を利用し、画像解析には C++ 言語を利用した。システムはプログラムを取り込む画面、取り込んだプログラムを実行する画面、迷路を作成する画面から構成される。

3.1 プログラミングの題材 (迷路) について

システムではブロックを組み合わせることでロボットをゴールまで移動させる手順についてプログラミングを行う。迷路

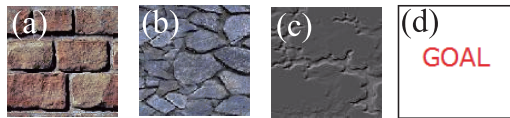


図 1 迷路に設置するタイル. (a) 移動禁止, (b) 通路, (c) 塀, (d) ゴール

Fig. 1 Tiles for creating maze (a) a forbidden area, (b) a path, (c) a wall, and (d) a goal.

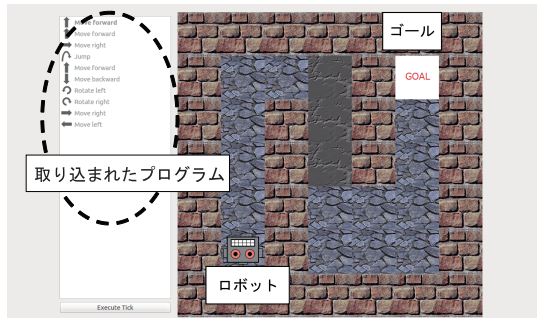


図 2 迷路の表示画面

Fig. 2 Screen of solving a maze.

は図 1 に示すタイルを複数組み合わせで作成する. 図 1(a) は移動禁止部分を構成するためのタイルであり, このタイル上は移動はできない. 図 1(b) は通路を構成するためのタイルであり, 通常の移動命令 (前, 後, 右, 左) により移動できる. 図 1(c) は塀を構成するためのタイルであり, ジャンプ命令により前に進む (塀の上に乗る) ことができる. 図 1(d) はゴールのタイルであり, 迷路のゴール地点を示し, 通常の移動命令 (前, 後, 右, 左) により移動できる. タイルの種類は自由に増やすことができ, タイルの特性として色, 図, タイル上の移動可否, また平面方向の移動だけでなく高さ方向の移動など特殊な移動が必要となるものなどの設定が可能である.

図 2 にタイルを組み合わせで作成した迷路の一例を示す. 迷路は図 2 に示すスタート地点にあるロボット (図 2 左下) をゴール地点 (図 2 右上) まで移動させる経路を考えてプログラミングを行う. このシステムを利用することで, ゴールまでの経路を考える場合にそれぞれのタイルの特性を考慮し, プログラミングを行うため, 処理の順番を意識してブロックを並べることになり, プログラミングの順次処理を意識することになる. また使用するブロック数の制限や最短経路の探索を行うなど, 経路を考える場合の制限を加えることでプログラミング学習に必要な論理的思考力 [23] に対して影響を与えることが可能となると考えられる.

3.2 迷路の作成

迷路は自由に作成できる. また作成した迷路は迷路データとして共有できる. 図 3 に迷路作成の画面を示す. 配置するタイルを選択して, 迷路内に配置していくことで迷路

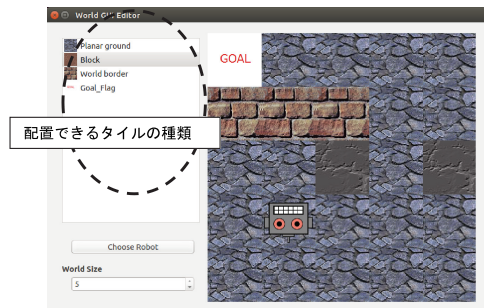


図 3 迷路作成時の画面

Fig. 3 Screen of creating a maze.



図 4 プログラム作成に用いるブロック

Fig. 4 The block using physical programming.

表 1 ブロック命令一覧

Table 1 List of blocks.

命令	動作
前	ロボットを前に移動
後	ロボットを後に移動
右	ロボットを右に移動
左	ロボットを左に移動
右に回し	ロボットの向きを右に 90 度回転
左に回し	ロボットの向きを左に 90 度回転
ジャンプ	ロボットの前方方向にジャンプ (ロボットの前方方向に高さの異なるタイルがある場合, 前に移動)

を作成する. スタート地点およびゴール地点は自由に設定が可能である. スタート地点にはロボット, ゴール地点にはゴールを示すタイルを配置する. 迷路の形状は正方形であり, サイズは変更可能である. 配置するタイルの場所, スタート地点, ゴール地点, 迷路のサイズなどを変化させることで迷路の難易度をユーザに合わせて自由に設定することが可能となっている.

3.3 プログラミング方法

開発したシステムはタイルにより作成された迷路についてロボットをゴールまで移動させる手順についてのプログラムを作成する. マウスなどのポインティングデバイスを使用せずにプログラミングできるフィジカルプログラミングを実現するため, 図 4 に示すマーカとそのマーカに対する動作を記述したブロックを作成した. 作成したブロックの種類を表 1 に示す. ブロックとしてロボットに対して移動, 向きの変更, 特殊な移動の命令を行うものを準備した. ロボットの移動は前, 後, 右, 左を作成した. これらの命

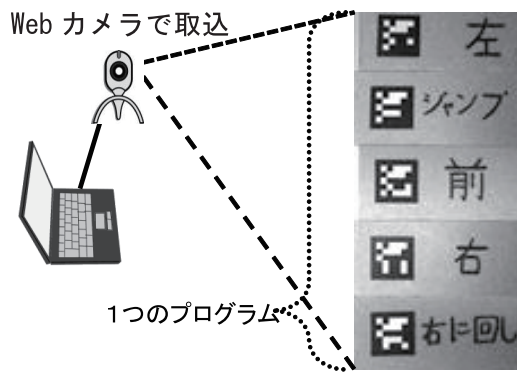


図5 開発したシステムを利用したプログラミング
Fig. 5 Programming of the system with blocks.

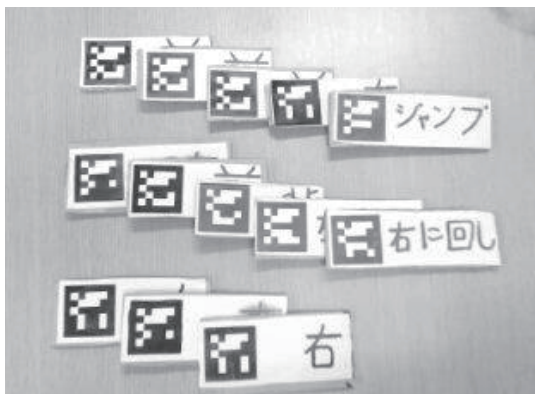


図6 複数のブロックの並べ方
Fig. 6 How to put many blocks.

令はロボットの向きを変えずに命令された方向にロボットを移動させる。向きの変更は右に回し、左に回しを作成した。これらの命令はロボットの向きを命令された回転方向に90度回転する。ただし、移動はしない。特殊な動作の命令として、ジャンプを作成した。ジャンプは前方向に高さが異なるタイルがあった場合、前に移動する。前方向のタイルの高さが同じ場合は移動しない。

プログラミングはこれらのブロックを並べることで作成する。図5に開発したシステムを用いてプログラミングを行う場合の概略図を示す。本システムは取り込んだブロックの位置からブロックの並べられた順番を把握し、ブロックの上から順に処理が実行される。この処理はプログラムの順次処理と同じ考えである。取り込み可能なブロック数はWebカメラで撮影できる範囲に限られるため、図5に示すようにブロックを垂直に配置した場合、取り込むことのできるブロック数は10枚未満となり、ブロックが10枚以上必要となるプログラム作成はできない。そこで取り込むことのできるブロック数を増やすため、Webカメラの撮影領域全体を使用し、図6に示すように重ねてブロックを並べた場合でも取り込めるようにした。この場合、垂直にブロックを並べたときと同じように配置したブロックの上から順に処理を実行するプログラムが生成される。最大で30枚程度のブロックを取り込めることを確認できている。

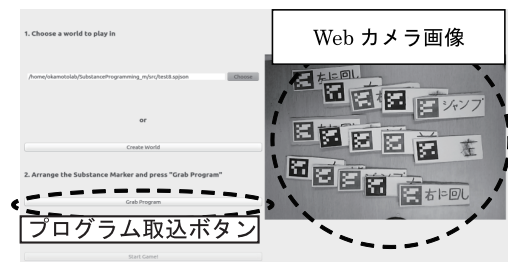


図7 プログラム取り込み画面
Fig. 7 Screen of grabbing program.

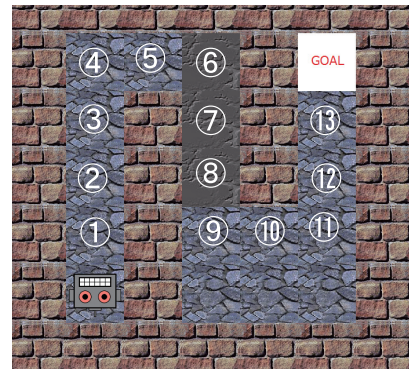


図8 迷路の例
Fig. 8 Maze of sample.

図7にシステムのプログラム取り込み画面を示す。プログラムを取り込む画面にWebカメラからの画像を表示することで、取り込み可能なブロックを確認しながら、プログラミングが可能である。ブロックの配置が終わるとプログラム取込ボタンを押下し、Webカメラの画像をシステムに取り込む。取り込まれた画像はマーカの解析が行われ、マーカに割り当てられた命令に従ってプログラムが作成される。

たとえば図8に示す迷路についてスタート地点のロボットをゴール地点まで移動させるプログラムを表2に示す。ロボットの置かれているスタート地点から1~4の「前」の命令で④まで移動する。次に5の「右」の命令で⑤に移動する。次は塀があるため、ジャンプをする必要があるが、ジャンプは前向きにしかできないため6の「右に回し」でロボットを右に90度回転させる。7の「ジャンプ」の命令で⑥に移動させる。次の塀についても同様に8の「右に回し」でロボットを右に90度回転させ、9, 10のジャンプの命令で⑧まで移動させる。11の「前」の命令で⑨に移動させる。12の「左に回し」の命令でロボットを左に90度回転させ（ロボットは右方向を向く）、13, 14の「前」の命令で⑩に移動させる。15の「左に回し」の命令でロボットは左に90度回転し（ロボットは前方向を向く）、16~18の「前」の命令でゴールに到着する。

3.4 課題の展開

図8に示す例はスタートからゴールまでに考えられる経

表 2 プログラム例 (図 8 の迷路に対する)

Table 2 Example of the program for reaching goal in Fig. 8.

順番	命令	ロボットの位置	ロボットの向き
1	前	①	前
2	前	②	前
3	前	③	前
4	前	④	前
5	右	⑤	前
6	右に回し	⑤	右
7	ジャンプ	⑥	右
8	右に回し	⑥	後
9	ジャンプ	⑦	後
10	ジャンプ	⑧	後
11	前	⑨	後
12	左に回し	⑨	右
13	前	⑩	右
14	前	⑪	右
15	左に回し	⑪	前
16	前	⑫	前
17	前	⑬	前
18	前	Goal	前

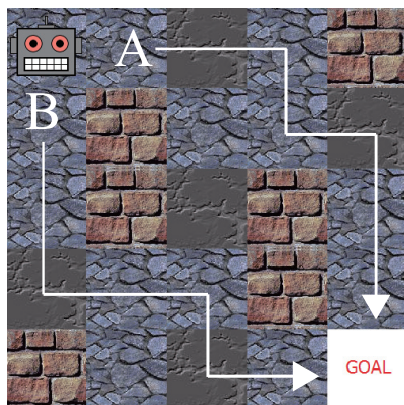


図 9 複数の経路が考えられる迷路の例

Fig. 9 Maze that has plural paths from start point to goal point.

路は 1 通りだけで使用ブロック数なども制限していないため、ロボットをゴール地点まで移動させるプログラムを考えることでプログラムの順次処理について学習できる。さらに、スタートからゴールまでの経路が複数あり、条件に従って最適な経路を選択する場合、筋道を立てて論理的に考える能力が必要であり、論理的な思考力をも身につけさせることが期待できる。図 9 に示す例ではスタートからゴールまでに考えられる経路は複数あり、たとえば A と B の経路が考えられる。この 2 つの経路でスタートからゴールまでに移動するタイル数は両方ともに 8 個であり、同じである。しかし、ブロック数を考慮した場合、経路 A では最低でも 9 ブロック必要 (左→左に回し→ジャンプ→前→右→ジャンプ→右→右→右) となり、経路 B では最低 8 ブロック (前→前→ジャンプ→左→左→ジャンプ→左→左)

でゴールまで移動できる。制限として最も少ないブロック数でゴールできるプログラムを考えるという条件を与えられた場合は経路 B を選択する必要がある。そのほかに手持ちのブロックを、たとえばそれぞれ 2 枚ずつなどに制限することで作成するプログラムは変化し、選択する経路も変化する。図 9 に示す迷路の縦と横のタイル数は 5×5 であるが、さらに大きなサイズの迷路とすることで選択できる経路の数は大幅に増加する。考えられる経路が多数となる場合、1 人よりもグループで考えることで、1 人では気付くことができなかった経路を教え合うことができ、学習効果に影響を与えることが可能となる。このようにプログラム作成に条件を与え、複数のプログラムの中から条件を満たすプログラムを考えることで、プログラミングに必要な論理的な思考力を身につけられると考えられる。さらに複数のプログラムを考える場合、グループ学習によりお互いに教え合うことで、より高い学習効果を得られると考えられる。

4. システムの評価

本研究はグループ学習が可能でプログラミング初心者がプログラム作成の目的を明確に意識でき、プログラミングが簡単に行えて苦手意識を持たない学習システムを開発することを目的としてシステムを開発した。開発したシステムについて目的がどの程度実現できているか確認するため、被験者にシステムを使用してもらい、評価を行った。目的の 1 つであるグループ学習が可能なシステムについては TUI を利用したことから達成できていると考えており、評価では被験者は単独でシステムを使用した。被験者は高等専門学校の 4~5 年生および専攻科 1 年生の学生 22 名とした。被験者には次に示す流れでアンケートに答えてもらった。被験者の 22 名全員はプログラミングを学んだことがあり、プログラム構造の順次処理について理解している。

- (1) システムの使用方法について説明 (5 分程度)
- (2) システムの使用 (60 分程度)

目的：図 8 に示す迷路においてゴールまでロボットを移動するプログラムを作成する (ブロックによるプログラム作成)

- (3) アンケート回答

4.1 取り組み姿勢について

開発したシステムは迷路を題材とすることでプログラミング初心者が目的を明確に意識して、プログラミング学習できるシステムを目的としている。課題の迷路に対してゴールまでロボットを移動させるプログラミングに積極的に取り組めたか確認するため、質問 1 に対して回答してもらった。

「質問 1：システムを用いてプログラミングの学習に積極的に取り組むことができましたか？」

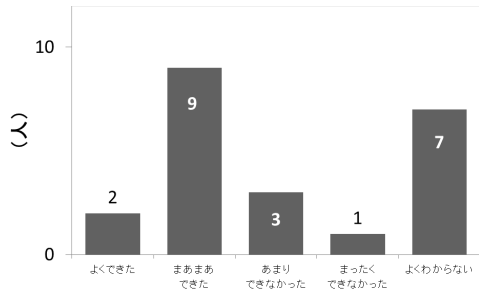


図 10 質問 1：システムを用いてプログラミングの学習に積極的に取り組むことができましたか？に対する回答

Fig. 10 Result of Question 1.

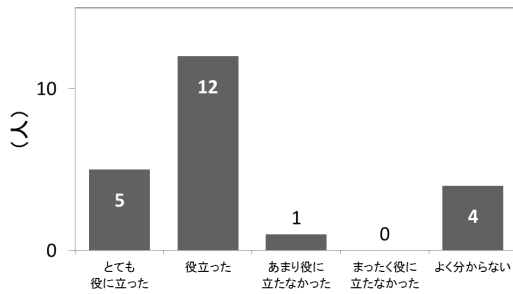


図 11 質問 2：システムはプログラミングの順次処理についての理解に役に立ちましたか？に対する回答

Fig. 11 Result of Question 2.

質問 1 に対する回答を図 10 に示す。システムを用いてプログラミング学習を積極的に取り組むことができたとの回答は 50% (11 名) であり、積極的に取り組むことができなかったとの回答は 18% (4 名) であった。また、よく分からないとの回答が 32% (7 名) あったが、これはプログラミング学習とシステムとの関連について詳しい説明が事前にできていなかったことが原因であると考えられる。

4.2 プログラム構造の理解について

開発したシステムはプログラミング初心者がプログラムについて学習するためのシステムであり、システムを使用することでプログラミングを学ぶことができるようになっている。実際に使用することでプログラムの順次処理の理解に役に立つことを確認するために質問 2 に対して回答してもらった。

「質問 2：システムはプログラミングの順次処理についての理解に役に立ちましたか？」

質問 2 に対する回答を図 11 に示す。プログラミングの順次処理についての理解に役立ったとの回答は 77% (17 名) であり、役に立たなかったとの回答は 5% (1 名) であった。この結果より、システムはプログラミングの順次処理の理解に役立つと考えられる。ゴールまでロボットを移動するためのプログラミングは 1 つのブロックを配置するたびにロボットの位置 (図 2 参照) を想定しながら次のブロックを決定する。これはプログラム構造の順次処理と同じ考え方であり、システム利用により順次処理の理解に

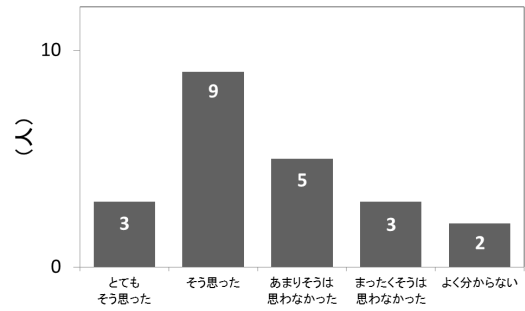


図 12 質問 3：システムを用いることで、プログラミングは簡単であると思えましたか？に対する回答

Fig. 12 Result of Question 3.

役に立つと考えられる。

4.3 プログラミングへの意識

プログラミング初心者に対するプログラミング学習では初心者が簡単にプログラミングでき、苦手意識を持たずにプログラミングに興味を持ってもらうことが重要である [24]。開発したシステムに対してユーザが感じるプログラミング学習の意識を確認するため、質問 3 に対して回答してもらった。

「質問 3：システムを用いることで、プログラミングは簡単であると思えましたか？」

質問 3 に対する回答を図 12 に示す。システムを用いることでプログラミングは簡単であると思ったとの回答は 55% (12 名) であり、被験者はシステムを用いることでプログラミングは簡単に学べると感じている。プログラミングを簡単と思わなかった被験者は 36% (8 名) であった。

5. システム改善の考察

システム評価時に改善点の要望として複数の指摘があった項目を示す。

- (1) 命令ブロックに記述されているマークの認識率の向上
- (2) 命令ブロックの色分け
- (3) 複数回に分けてのプログラム取り込み
- (4) ロボットを移動したときのアクション
- (5) 命令ブロックの読込数の増加

評価を行った環境では照明などの影響でマーカ認識率が非常に悪くなる場合があり、配置したブロックを正確に認識できないことがあった。マーカ認識に関する問題が発生するとプログラムが思いどおりに生成できなくなることから、大きなストレスを感じながらの学習となる可能性があり、改善が必要である。またシステムを使用する場所の照明などについても十分注意する必要があることが明らかになった。その他の改善要望としてはブロックの読み取り枚数に関する内容が多く、読み取ることのできるブロック数を増やす必要があると考えられる。一度に読み取ることのできる枚数は Web カメラにより制限されるため、複数回

に分けて読み取れるような改善が必要であると考えられる。またデザインやロボット動作に関する改善要望があり、初心者が積極的に取り組めるようなデザインや動作を取り入れることで、見た目においても楽しみながら学習できるシステムとすることについて考える必要がある。

グループ学習に対しては TUI を利用していることで可能であると考えられるが、実際にグループによる評価を実施し、グループ学習による学習効果やグループ学習を実施する場合のシステムの改善点を明らかにする必要がある。グループ学習の効果については次に示す手順で確認する。プログラム作成において条件を与えて、最初は被験者（プログラミング未経験者を含める）に単独で回答してもらう。次にプログラム作成に同じ条件を与えて、グループにより他の人が気付かない経路や移動方法などをお互いに教え合いながら回答してもらう。この後アンケートなどにより、1人の場合とグループの場合を比較して論理的な思考力や学習意欲などへの影響について確認することで開発したシステムを利用したグループ学習の効果を明らかにしたい。

また、開発したシステムはプログラム構造の条件分岐や繰り返し処理などに対応できていない。プログラミングの学習には条件分岐、繰り返し処理やその他のプログラミングの要素についての理解が必要である。今後はロボット周囲のタイルのチェックや条件分岐などのブロックを追加することでタイルの条件により進行方向を変化可能としたり、繰り返し処理のブロックを追加して、迷路のゴールまでの手順を考えるプログラミングではなく、進めなくなった場合の処理などを考えたプログラミングを行えるよう改善する。

6. まとめ

本研究はグループでの学習が可能でプログラミング初心者がプログラミングに興味を持って積極的に学習できるシステムを目的として、TUI を利用し、迷路を題材としたプログラミング学習システムを開発した。初心者が簡単にプログラミングを行えるようにするため、ブロックを並べることでプログラミングできるシステムとした。本システムで利用するブロックの作成はマーカを印刷し、厚紙に貼り付けることで実現できるため、TUI を利用するための環境構築を低いコストにより実現可能である。また、開発したシステムについて評価を行い、次の項目について実現できていることを確認した。

- (1) 初心者が積極的にシステムを使用できる。
- (2) プログラミング学習に効果を有する。
- (3) 簡単にプログラミングが行え、プログラミングに苦手意識を持たせない。

本システムはプログラミング初心者がプログラミング学習の初期段階で用いることで、プログラミングに興味を持って積極的に学習できると考えられる。また、使用するブロック数や手持ちのブロックなどに条件を与えることで

複数のプログラムの中から条件を満たすことのできるプログラムを検討することにより、プログラミングに必要な論理的な思考力を身につけることが可能である。さらにはグループによる学習ではお互い教え合うことが可能であり、1人で学ぶよりも高い学習効果を得られると考えられる。たとえば本システムは小学校への出前授業などでプログラミングのグループ学習などを実施するとき用いることで、プログラミング学習だけでなくプログラミングに必要な論理的な思考力を身につけることも可能である。このシステムを使用してプログラミングに興味を持ったユーザは、次のステップとして Scratch などのシステムを用いて自由にプログラミングを行い、プログラミングの学習を継続していくことで、自ら創造し、実装できる ICT 人材の育成に貢献ができることを期待する。

謝辞 本研究は e-とくしま推進財団調査・研究事業の助成を受けたものです。また、アンケートによる調査を快く引く受けてくれた阿南工業高等専門学校の学生たちに心より感謝します。

参考文献

- [1] 総務省情報流通行政局情報通信利用促進課：教育情報化の推進，総務省（オンライン），入手先
(http://www.soumu.go.jp/main_sosiki/joho.tsusin/kyouiku.joho-ka/keihatsusozai_dl.html)（参照 2015-01-23）。
- [2] 総務省情報流通行政局情報通信利用促進課：ICT 人材の育成，総務省（オンライン），入手先
(http://www.soumu.go.jp/main_sosiki/joho.tsusin/joho.jinzai/index.html)（参照 2015-01-23）。
- [3] Zuckerman, O., Arida, S. and Resnick, M.: Extending Tangible Interfaces for Education: Digital Montessori-inspired Manipulatives, *Proc. SIGCHI Conference on Human Factors in Computing Systems*, pp.859-868 (2005).
- [4] Sapounidis, T., Demetriadis, S. and Stamelos, I.: Evaluating children performance with graphical and tangible robot programming tools, *Pres. Ubiquit. Comput.*, Vol.19, pp.225-237 (2015).
- [5] Maloney, J., Resnick, M., Rush, N., Silverman, B. and Eastmond, E.: The Scratch Programming Language and Environment, *ACM Trans. Compu. Educ.*, Vol.10, No.4, p.16 (2010).
- [6] 原田康徳：子供向けビジュアル言語 Viscuit とそのインターフェース，情報処理学会研究報告，HI，ヒューマンインタフェース研究会報告，Vol.2005, No.114, pp.41-48 (2005).
- [7] Google: Blockly Games, Google Inc. (online), available from (<https://blockly-games.appspot.com/>) (accessed 2015-02-04).
- [8] MacLaurin, M.: The design of kodu: A tiny visual programming language for children on the Xbox360, *Proc. 38th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pp.241-246 (2011).
- [9] 兼宗 進，阿部和広，原田康徳：プログラミングが好きになる言語環境，情報処理，Vol.50, No.10, pp.986-995 (2009).
- [10] 吉永卓矢，脇田 建：ビジュアル・プログラミング言語

- とその実行・開発環境の提案, 情報処理学会第 73 回全国大会講演論文集, pp.363-364 (2011).
- [11] 河崎雅人: 短期大学における情報教育カリキュラムの開発 (2)—学習によるコンピュータイメージの変容, 日本教育情報学会年会論文集, Vol.8, pp.21-24 (1992).
- [12] 崎山 充, 渡辺和夫, 藤岡直矢, 皆月昭則: プログラミング教育における導入期の苦手意識の変化に関する一考察, 情報科学技術フォーラム講演論文集, pp.519-520 (2009).
- [13] 網代 孝, 土田賢者: ビジュアルプログラム言語 A-BITS, 電子情報通信学会技術研究報告, SS, ソフトウェアサイエンス, Vol.103, No.318, pp.15-20 (2003).
- [14] 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, 日本教育工学会論文誌, Vol.34, No.4, pp.387-394 (2011).
- [15] 蛭田雄一, 桑原 悟, 柴原一友, 但馬康宏, 小谷善行: すご@ぶろ: 双六をモチーフとしたビジュアルプログラミング言語, 情報処理学会研究報告, CE, コンピュータと教育研究会報告, Vol.2008, No.13, pp.41-48 (2008).
- [16] 知見邦彦, 樋山淳雄, 宮寺庸造: 失敗知識を利用したプログラミング学習環境の構築, 電子情報通信学会論文誌 D-I, Vol.J88-D-I, No.1, pp.66-75 (2005).
- [17] 山本利一, 田賀秀子, 新屋智絵, 小林靖英: 共同学習を取り入れたプログラミング学習の課題の提案—カーリングゲームを取り入れたプログラミング指導, 日本教育情報学会誌, Vol.22, No.3, pp.11-18 (2006).
- [18] 生田目康子: ピア・レビューをとまなうグループ学習の評価—斉型プログラミング授業への適用, 情報処理学会論文誌, Vol.45, No.9, pp.2226-2235 (2004).
- [19] Horn, M.S., Solovey, E.T., Crouser, R.J. and Jacob, R.J.: Comparing the Use of Tangible and Graphical Programming Languages for Informal Science Education, *CHI '09, Proc. SIGCHI Conference on Human Factors in Computing Systems*, pp.975-984 (2009).
- [20] 八城朋仁, 迎山和司: 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, 情報処理学会インタラクシオン 2014, pp.647-650, 情報処理学会 (2014).
- [21] 川崎基輝, Nguyen, N.T.H., 秋田純一: 2色での入力・表示可能なマトリクス LED ブロックを用いたタンジブルなプログラミング教材, 情報処理学会インタラクシオン 2014, pp.206-209, 情報処理学会 (2014).
- [22] Touretzky, D.S.: Teaching Kodu with physical manipulatives, *ACM Inroads*, Vol.5, No.4, pp.44-51 (2014).
- [23] 中園長新: アルゴリズム・論理的思考学習の実践と成果分析—単元前後の生徒アンケート結果を元に, CIEC 研究会論文誌, Vol.3, pp.4-11 (2012).
- [24] 中尾茂子, 安達一寿: プログラミング学習による学生のイメージ変化と学習効果の分析, 日本教育情報学会学会誌, Vol.11, No.1, pp.3-11 (1995).

推薦文

現在, 初等中等段階を中心に初心者向けのプログラミング学習が重要性を増している. そこで使用されるプログラミング言語・環境についても, テキスト型のものに加え, ブロックなどを組み合わせるスタイルのものなどが注目されるようになってきている. さらにそれを実世界に広げたものが TUI (タンジブルユーザインタフェース) を持つプログラミング環境であり, 物理的なブロック等を組み合わせてプログラムを作る形をとる. これは, 複数人から成るグループが共同作業でプログラムについて考える形の学習に有効に活用されるものと期待される. 本研究では, 厚紙

に模様や部品を印刷したものをユーザが配置・構成し, それを Web カメラで撮影・認識することで動作する形で実現した TUI のプログラミング環境を提案している. また, プログラムが扱う題材として迷路を取り上げている. 本論文の提案は迷路の使用やプログラムの構成について明確な制約を設けるという方向の新たな試みとなっている. 研究としてはまだ途上であり評価も主観的なものに留まっているが, このような多様な提案が積み重なってプログラミング学習環境のさまざまな可能性が試されて行く際の一步という意味では重要な貢献となっているものと考えている.

(論文誌「教育とコンピュータ」アドバイザー 久野 靖)



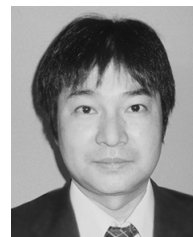
Dennis Stritzke

1993 年生. 2015 年 Ostfalia University of Applied Sciences 卒業. 現在, Goethe University Frankfurt 在学中. 2011 年 LINEAS Informationstechnik GmbH 入社. 2015 年株式会社 NTT データ入社. コンピュータサイエンス, ソフトウェア工学およびプログラミング学習支援システムの研究に従事.



前田 佑太

1994 年生. 2014 年阿南工業高等専門学校制御情報工学科卒業. 現在, 同高等専門学校専攻科電気・制御システム工学専攻在学中. プログラミング学習支援システムの研究に従事.



岡本 浩行 (正会員)

1969 年生. 1994 年徳島大学大学院博士前期課程修了. 2003 年同博士後期課程単位取得退学. 博士(工学). 1994 年株式会社明電舎入社. 2000 年株式会社ジャストシステム入社. 2004 年阿南工業高等専門学校助手. 2006 年同高等専門学校助教授. プログラミング学習支援システムの研究に従事. 応用物理学会, 電子情報通信学会, 日本教育工学会, 日本バーチャルリアリティ学会各会員.