

## 小・中学生を対象にしたプログラミング学習を支援する アプリケーション開発プロジェクト

尾花拓海<sup>†1</sup> 鈴木龍成<sup>†1</sup> 吉村明人<sup>†1</sup> 白田莉菜<sup>†1</sup> 半澤魁士<sup>†1</sup>  
佐久間拓也<sup>†1</sup> 川合康央<sup>†1</sup> 池辺正典<sup>†1</sup>

### Application development project to support programming learning for elementary and junior high school students

TAKUMI OBANA<sup>†1</sup> RYUSEI SUZUKI<sup>†1</sup> AKIHITO YOSHIMURA<sup>†1</sup>  
RINA USUDA<sup>†1</sup> KAIJI HANZAWA<sup>†1</sup>  
TAKUYA SAKUMA<sup>†1</sup> YASUO KAWAI<sup>†1</sup> MASANORI IKEBE<sup>†1</sup>

**概要**：本研究システムは、現在において情報系人材が不足していることによってプログラミング教育が促進されている背景から、小・中学生を対象としたプログラミング学習を支援するためのアプリケーションである。本アプリケーションは、幅広いシステムの構築に応用が可能な基礎的な知識を得るために、プログラムを作成する前段階として必要となるアルゴリズム設計能力を養成することを目的とする。

**キーワード**：プログラミング教育，ビジュアルプログラミング，ゲーミフィケーション

#### 1. はじめに

現在、日本国内における情報関連の動向として、ビッグデータや AI, IoT 等の先端 IT における投資の増大やセキュリティ対策等へのニーズの拡大傾向がみられる。これにより、今後は情報分野において更なる人材の確保が必要であるが、現状は情報系人材が不足している点が課題となっている。経済産業省の推計[1]によると、2020 年で 36.9 万人、2030 年では 78.9 万人の情報系の人材不足が見込まれているが、若年人口は今後も減少していくことから、教育分野における課題としても情報系人材の不足が考慮されるべきである。この背景を受け、文部科学省は 2016 年 6 月に小学校におけるプログラミング教育を指導要領に組み入れることが示された[2]。これは、高度情報通信ネットワーク社会推進戦略本部が公開した 2016 年 5 月の世界最先端 IT 国家創造宣言の変更[3]でもプログラミング教育の推進に関する内容が含まれていることから、今後のプログラミング教育への取り組みは非常に重要であると考えられる。

これまでのプログラミング教育では、暗記型のプログラミング教育が多く実施されていた。これは、比較的容易に教育の実施が可能であり、プログラミング教育の人材育成等の課題を考えると効率の良い手法であった。しかし、暗記型のプログラミング教育は、教育を受ける対象である小・中学生がプログラミング的思考を学習する事が困難であるため、不適切な方式であると考えられる。こ

の問題の解決のために、プログラミング教育人材の育成促進の取り組み[4]やプログラミング教育を補助するツールの開発[5]が行われている。このため、本稿における研究でも、小・中学生のプログラミング教育に向けて、プログラミング的思考を養成可能なアプリケーションを開発することで、小・中学生を対象にしたプログラミング教育で必要とされる一連の活動を実現するために必要な組み合わせについて、どのような組み合わせや改善をする事で意図した動きになるのかを論理的に考える力を養成する。

#### 2. 本研究システムの概要

本研究システムは、小・中学生を対象としたプログラミング学習を支援するアプリケーションである。本アプリケーションは、プログラムを作成する前段階として必要となるアルゴリズム設計能力を養成することを目的とする。

##### 2.1 本研究システムで養成する能力

本研究システムでは、特定のシステム開発のための能力を養成するのではなく、幅広いシステムの構築に応用が可能な基礎的な知識を得るために、アルゴリズム的思考を養成することを目的とする。アルゴリズム的思考は、問題に対して、「何を」、「どのような順番で」、「何に対して行うのか」という手順を定式化した形式で表現する必要がある。アルゴリズム的思考を具体的に表現する手法としては、フローチャートやプログラム構造表記法 (PAD) がある。

アルゴリズムを学習するための方法としては、暗記型 (記憶型) 学習と思考型学習の 2 種類がある。最初に、暗記型学習とは、事前に複数のアルゴリズムの形を記憶する

<sup>†1</sup> 文教大学 Bunkyo University

ことで、システム設計時に記憶したアルゴリズムに適合する問題解決のパターンが存在する場合には、当該アルゴリズムを適用することを目標とした学習方法である。次に、思考型学習とは、処理を構成する部品を分解して、問題にあわせて再構築することができる能力を養成する学習である。前者の学習方法は、プログラミング初学者には比較的敷居の低い学習方法ではあるが、記憶したアルゴリズムのパターンの適用が困難である未知の問題への対応能力が低いままである。そして、後者の学習方法は、一定の基礎的なプログラミング能力の学習がなされているという訓練が前提となっているが、前述の問題を解消し、複数の解法のある問題についての対応能力も養成することができると考えられている。このため、暗記型学習から思考型学習への移行を補助することは重要であり、本研究システムは、この移行に必要となる能力を支援することを目的としている。

本研究の目的を達成し、思考的学習に至るための学習用問題として、本研究システムでは、不確定性を取り入れた問題を提示し、解法の検討を行うことで思考的学習を行う。不確定性とは、ランダム性とも呼ばれ、この要素を取り入れることで、単純な変数の変更や条件の変更で対応することが難しくなり、新たなアルゴリズムの設計が要求される。不確定性の代表的な例としては、トランプゲーム等のような人同士が競うものがあるが、本研究システムでもゲーム相手を想定したエージェントを配置し、設計したアルゴリズムを実行した際に、同条件でポイントを競うゲーム形式とすることで不確定性を実現した。本ゲームにおいて、操作するキャラクターの動きをあらかじめアルゴリズムとして設計して実行することで、プログラムの設計から動作確認までの手順と同等の学習効果を得ることとなる。

## 2.2 他の研究システムとの差異

他の研究システムでは、特定のシステム開発のための能力を養成する目的のものが多くあった。しかし、前述した通り本研究システムはアルゴリズム的思考を養成することを目的としているため、その点に本質的な差異が存在する。

## 2.3 本研究システムの特徴

本研究システムでは、プログラミング初学者が暗記型学習から思考型に移行する際に課題となるモチベーションの維持のために、ゲーミフィケーションやビジュアルプログラミングの概念を取り入れた。

### 2.3.1 ゲーミフィケーション

ゲーミフィケーションとは、ゲームの特性をゲーム以外の分野に活用する事で利用者の動的な参加を促す手法である。ゲーム的な要素である、ゴール及びルールの設定、即時的なフィードバック、自発的な参加の促進という事項を取り入れる事で、モチベーションの維持等の効果が期待できると考えられる。ゲーミフィケーションは教育分野での実践事例[6]もあることから、本研究システムでもゲーミフィケーションによる理解向上を目指した。具体的には、キ

ャラクタの動作の設計とプログラミングにおけるアルゴリズムの設計を関連付け、ゲームとしてプログラムの設計の能力養成を可能とし、対戦相手となるキャラクターとの競争から、競争による効果的な学習結果が得られるようにした。

### 2.3.2 ビジュアルプログラミング

プログラミング言語では、テキストデータとしてソースコードを作成し、これをコンパイルすることで実行可能な状態とする形式が一般的である。しかし、プログラミング初学者には、プログラミング言語の表記法等の理解とアルゴリズム的思考を同時に学習する必要性があり、後者のみを意識することが難しい。このため、アルゴリズムの検討のみに集中することができる仕組みとして、処理を示すパーツを組み合わせてプログラミングを構成するビジュアルプログラミング言語がある。本研究システムでもビジュアルプログラミング言語としてアルゴリズムを設計することを可能にして、ブロックタイ型のオブジェクトをドラッグ&ドロップしてプログラムを作成する。これによりユーザは直観的にプログラムを組み立てることが可能となる。

## 2.4 ゲームのルール

本研究システムでは、アルゴリズム的思考の能力を養成するために、ブロックタイ型のオブジェクトを組み立ててアルゴリズムを設計し、キャラクターを動作させるゲームである。本研究システムではアルゴリズム的思考を学習するために、難易度の異なる30のステージを設けており、1~20番目までのステージは、本システムの基本操作やルールの学習を行う。そして、21~30番目のステージでは、対戦相手となるキャラクターの動作を予測した上で、自身のキャラクターの動作を設計し、より多くのポイントの取得を目指すことで、アルゴリズム的思考を養う。対戦相手のキャラクターの動作パターンはステージ毎に6個のパターンがあり、その中からランダムに動作が決定されることで複数回の学習も可能にした。具体的なルールは以下の通りである。

- ゲーム画面は格子状に区切られた最大で縦横が6×6の画面であり、形状の異なる30のステージがある
- ゲーム画面上には、複数のチェック地点(1~3個)があり、チェック地点を表すオブジェクトが表示される。チェック地点は各ステージで固定である
- チェック地点にキャラクターが到達するとオブジェクトが消え、そのキャラクターにポイントが加算される
- ゲーム画面を確認した上で、プレイヤーは自身のキャラクターの動作を組み合わせて事前に行動を設計する
- キャラクタの動作の種類は、「前に進む」、「方向変換(左右)」、「ジャンプ」があり、「前に進む」および「ジャンプ」は動作に1秒を必要とする
- キャラクタの動作を制御するものとして、「動作の繰り返し」と「特定条件の場合のみ動作する」ことが可能である

- ゲームを開始すると、自身のキャラクタと対戦相手のキャラクタが同時に動き、全てのチェック地点のオブジェクトが無くなったならゲームが終了する
- 自身のキャラクタと対戦相手のキャラクタの初期の配置位置は各ステージで固定であり、ゲームが開始されると同じ速度で移動する
- ゲーム終了時点でポイントの多い方のキャラクタの勝利とする

後半で出現する対戦相手のキャラクタはステージ毎に 6 種類の行動パターンからランダムで行動が決定されるために、これが不確定な要素となり、同じステージでも異なる動作が必要となる。また、対戦相手のキャラクタの動作を予測するヒントとして、対戦相手のキャラクタがチェック地点のうちの 1 つを何番目に通過するかの情報が示されるため、この情報から対戦相手のキャラクタの動作を予測し、勝利するための動作の組み合わせを設計する必要がある。

### 3. 本研究システムの具体的な機能

本研究アプリケーションはビジュアルプログラミング言語によりプログラミング初学者でもアルゴリズムを設計するための学習を行うことができる。

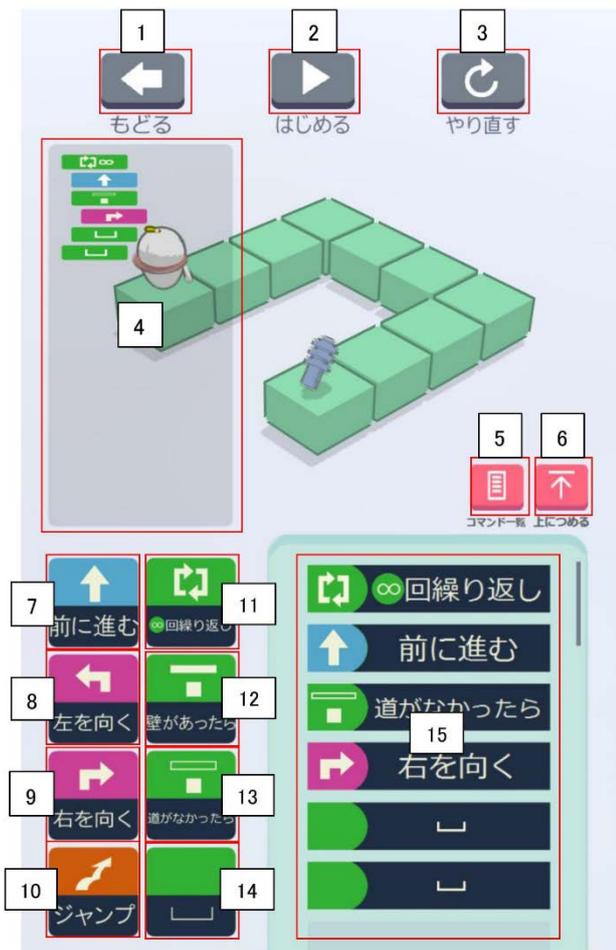


図 1 システムの画面

Figure 1 System Screen Description

#### 3.1 動作環境

本研究システムは小・中学校におけるプログラミング教育を想定しているために、これらの学校で比較的導入が進んでいるタブレット端末で動作することを目的とした。このため、本研究システムは Android 端末での動作が可能とするように開発を行った。動作環境を Android 端末とすることで、スマートフォン等での動作も可能であることから、比較的容易に自宅等での学習環境を構築することができ、学習効率の向上に繋がると期待することができる。

#### 3.2 アプリケーションの画面

本アプリケーションの画面は、縦に 3 つに分割された領域から構成される。最上部の領域にはボタンが 3 つ配置され、「ステージの選択画面に戻る」、「設計したプログラムを実行する」、「設計からやり直す」の操作が可能である。また、中央用の領域はゲーム画面であり、前述のルールに従い、自身のキャラクタおよび対戦相手となるキャラクタが動作する領域であり、実行時には現在動作しているパネルを確認することができる。また、最下部にはプログラムを設計するための動作パネルの一覧と、現在設計中のパネルの並びが表示される。プレイヤーはこのパネル一覧からドラッグ&ドロップによりプログラムの設計を行う。本研究システムの画面例を図 1 に示す。図 1 の通り、システムの画面を構成する要素は 15 個あり、それぞれの箇所の操作方法および表示される情報の意味は次の表 1 の通りである。

#### 3.3 動作パネルについて

動作パネルは全部で 8 種類あり、これらを組み合わせることで、フローチャートを書くようにプログラムを設計することが可能となる。動作パネルの種類は、プログラミング構造の基本となる「順次処理」、「分岐処理」、「繰り返し処理」のを学ぶことができる。最初に、順次処理を構成するための動作パネルとしては、「前に進む」、「ジャンプ」、「右を向く」、「左を向く」の 4 種類がある。順次処理ではこれらの動作パネルを単純に並べることによって、処理を組み立てる。

次に、分岐処理は特定の条件下においてのみ動作を行うという振る舞いを作成するものである。一般的なプログラミング言語においては、IF 文や SWITCH 文を用いた処理制御がこれにあたり、本アプリケーションでは具体的な動作パネルとして「壁があったなら」、「壁がなかったなら」の 2 種類の条件判定を行う動作パネルがある。

最後に、繰り返し処理は一定の条件が満たされている場合に同じ処理を行うものであるが、本アプリケーションにおいては回数指定で処理を繰り返すという制御を行う「指定回数繰り返し」という動作パネルがある。繰り返しと分岐処理の動作パネルを組み合わせることで、複雑な行動も

簡単に書くことができるようになり、効率的なプログラム設計を意識して動作パネルを組み立てるということが可能となる。

表 1 画面の説明  
Table 1 画面の説明

番号	名称	説明
1	戻る	ステージ選択画面に戻る
2	再生	入力されたコマンドを上から順番に実行してキャラクターが動作する
3	リセット	入力されたコマンドを全て消す
4	入力コマンド表示エリア	入力されたコマンドが表示される。「繰り返しコマンド」や「分岐コマンド」の範囲は「インデント」で表現
5	コマンド表示切り替え	「入力コマンド表示エリア」の表示、非表示を切り替える
6	コマンド上詰め切り替え	「コマンド入力エリア」のコマンドの自動的に上に詰めるかを設定する
7	前に進む	キャラクターが1歩前進する
8	左を向く	キャラクターが90度左を向く動作パネル
9	右を向く	キャラクターが90度右を向く動作パネル
10	ジャンプ	キャラクターがジャンプしながら1歩前進する動作パネル
11	繰り返し	「カッコ閉じコマンド」までに入力されているコマンドを繰り返す
12	壁があったら(分岐)	キャラクターの1歩前に壁がある場合の動作を設定する
13	道がなかったら(分岐)	キャラクターの1歩前に道がない場合の動作を設定する
14	カッコ閉じ	「繰り返しコマンド」や「分岐コマンド」の終了地点を配置する
15	コマンド入力エリア	各コマンドの配置エリア、動作パネルの配置可能個数は20個まで

#### 4. おわりに

本研究システムの利用により、アルゴリズム的思考を訓練するための方法を提供した。また、プログラミング初学者がアルゴリズム設計能力の養成に移行する際に課題となるモチベーションの維持をビジュアルプログラミング環境やゲーミフィケーションの要素を取り入れることで解消することが可能であると考えられる。

本研究システムの今後の課題としては、対戦相手となるキャラクターの動作パターンを複数の固定された動作からのランダム抽出ではなく、毎回自動的に難易度に応じて動的な設定とすることで複数回の学習でも効果的とする点等が挙げられる。

#### 参考文献

- [1]経済産業省：IT人材の最新動向と将来推計に関する調査結果について、  
[http://www.meti.go.jp/policy/it\\_policy/jinzai/27FY\\_report.html](http://www.meti.go.jp/policy/it_policy/jinzai/27FY_report.html)(2017年5月22日参照)。
- [2]文部科学省：小学校段階におけるプログラミング教育の在り方について（議論の取りまとめ）、  
[http://www.mext.go.jp/b\\_menu/shingi/chousa/shotou/122/attach/1372525.htm](http://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/attach/1372525.htm)(2017年5月22日参照)。
- [3]高度情報通信ネットワーク社会推進戦略本部：世界最先端IT国家創造宣言、  
<http://www.kantei.go.jp/jp/singi/it2/kettei/pdf/20160520/siryou1.pdf>(2017年5月22日参照)。
- [4]吉田研一、伊藤寿晃、山脇智志、大森康正：小学生を対象としたプログラミング教育指導者育成方法とその評価、情報処理学会研究報告、Vol.2017-CE-138, No.10, pp.1-4 (2017)。
- [5]光永法明、井芹威晴、吉田図夢：タブレット端末で動作する、マイコン用ビジュアルプログラミング環境 aiBlocks の開発、情報処理学会論文誌、Vol.3, No.1, pp.53-63 (2017)。
- [6]鈴木聡：大学生の統計リテラシー獲得を目指す模擬コンペディション振り返りによる授業実践、情報処理学会研究報告、Vol.2017-CE-139, No.21, pp.1-7 (2017)。