# プログラミング学習ゲーム「アルゴロジック」に関数を導入する拡張の検討

1215078　粕谷 彪人　【 プログラミング言語研究室 】

# A Study on Extending Game-based Programming Education Tool 'Algologic' with Functions

1215078　Ayato Kasutani　【 Programming Languages and Systems Lab. 】

## 1　はじめに

近年プログラミング教育への関心が高まっており，2020 年には初等教育でプログラミング教育が導入される．初等教育でのプログラミング教育の目的の一つに「プログラミング的思考を養う」がある [1]．しかし我々は，プログラミング的思考のみではなく Wing が提唱する Computational Thinking（以下，CT）[2] も学習するのが良いと考える．CT とは，大きな問題を細分化や一般化を行い問題を解決するための手順を明確にする全ての人にとっての基本的なスキルである．また，パパート [3] も構成要素をモジュール化し独立させる力を養うことの重要性を主張している．

プログラミング的思考を学習するために，ビジュアルプログラミング言語が開発されている．その一つにアルゴロジック [4] がある．アルゴロジックでは，ブロックでプログラムを作成し，プログラミングの制御構造である順次，ループ，分岐をゲーム感覚で体験することができる．しかし，アルゴロジックにはモジュールに関する機能は存在しない．そこで，アルゴロジックにモジュール機能を追加することを考えた．

アルゴロジックにモジュール機能を追加する場合，モジュールはブロックの列とし，一般のプログラム言語の関数のように扱えるようにするのが自然である．本研究では，ブロックの列を関数として扱う機能をアルゴロジックに導入する際に，どのような仕様にすれば良いかを検討する．特に，関数の引数を導入するかどうかや，引数として何を渡せるようにするか，関数の実行をどう表示するかを検討する．

## 2　アルゴロジック

アルゴロジックは，プログラムの基本をゲーム感覚で学習することができるプログラミング学習ゲームである．このゲームは，図 1 のように左側にゲームの課題（以下，ステージ）が出題される．ステージには，ロボットと旗が表示されており，壁に当たることなく全ての旗を回収するようにロボットを制御するプログラムを作成するとステージクリアとなる．プログラムは，図 1 の右側にあるブロックを使って作成することができる．ブロックはマウスで移動させ，図 1 の中央のスペースで

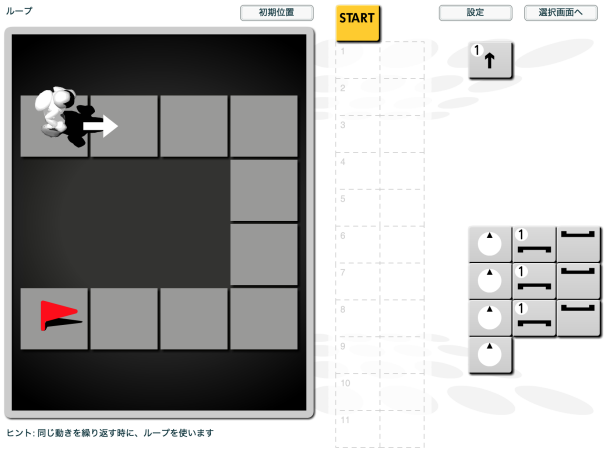


図 1　アルゴロジックの画面

プログラムを作成する．アルゴロジックでは各ステージで使用できるブロックが決まっており，学習者にループや分岐を使用しないといけない状況を作っている．

プログラムの実行では，ロボットの動作に対応してプログラムがハイライトされる．ハイライトすることで，ステージをクリアできなかった場合どのブロックが意図した動作をしていないかなど調べ，プログラムを改良できる．

## 3　プログラミング言語としての仕様

モジュール機能を追加したアルゴロジックで，関数に引数を渡せるようにするかどうかを検討した．引数を導入することで，関数を一般化しやすくなると考えた．しかし，二つ以上の引数を渡せるようにした場合，引数を二つ以上使うことができるステージを用意する必要がある．そうなると課題の難易度が上がり学習者の学習意欲を削ぐ可能性があるため，引数は一つだけ渡すことができるようにした．引数で渡すものは，空列を含むブロックの列にした．他にも，一つのブロックや数値のようなものを渡すことも考えられる．しかし，一つのブロックを渡すようにした場合，ループ開始のブロックを渡すことができ，実行時に構文エラーが発生する可能性がある．構文エラーは学習者が躓く要因になると考え，

構文エラーが発生しにくいプログラム列を渡すようにした．また，数値のようなものを導入すると，プログラムの実行中の状態を構成する要素が増える．そのため数値のようなものも導入しないことにした．また，プログラムの自由度を増やすために，使うブロックの数は制限しないことにした．

## 4　プログラムの実行の表示

実行中のブロックをハイライトすることでプログラムの実行を表示する．関数呼び出しの場合は，関数を呼び出している様子を表す必要がある．そこで，実行画面にコールスタックを表示し，関数呼び出しが実行された場合，コールスタックの中に呼び出した関数の定義を表示し，ハイライトするようにした．

引数の実行の表示に関しては，仮引数を実行した時に呼び出した側の関数の定義中にある実引数のブロックをハイライトする方法（戻り表示），プログラムが呼び出された際に仮引数を実引数で置き換える方法（置換表示），仮引数の実行時に実引数をコールスタックに追加し表示する方法（スタック表示）の 3 種類を実装した．図 2 は同じプログラムを実行した時の各表示方法での表示である．この図 2 のプログラムは，関数呼び出しを行い，引数として前進ブロックを 2 個渡す．呼び出された関数は，前進ブロックの後，仮引数を実行するプログラムである．図 2 では，呼び出された関数であるクローバーが引数を実行し，その実引数の 1 ブロック目を実行中の様子を示している．

戻り表示は，仮引数の代わりに実引数をハイライトする表示方法である．コールスタック中の仮引数のブロックが実行された場合，呼び出し元のコールスタックにハイライトが移り，関数呼び出しの際に指定した実引数をハイライトする．そうすることで，仮引数と実引数の対応が分かる．

置換表示は，仮引数を実引数に置き換える表示方法である．関数を呼び出し，コールスタックに呼び出された関数の宣言を追加する際に仮引数を実引数に置き換えて表示している．そうすることで，最新のコールスタックのみを見るだけで実行中のブロックが分かる．

スタック表示は，仮引数の実行時に実引数をコールスタックに追加し表示する表示方法である．仮引数が実行された際，コールスタックに実引数を追加し表示する．表示された実引数にハイライトを移すことで実行中のブロックを分かりやすくしている．そうすることで，最新のコールスタックを見るだけで実行中のブロックが分かりやすく，また仮引数と実引数の対応も分かる．

## 5　比較

本教材の開発に携わった大学院生の被験者二人に，それぞれの引数の実行の表示方法で 3 種類のステージを解いてもらい，どの表示方法が一番分かりやすかったか回答してもらった．その結果，二人ともスタック表現が

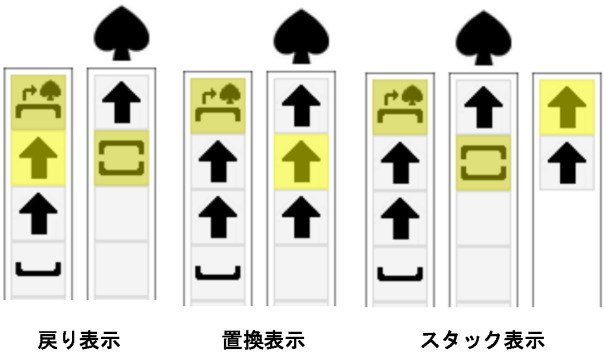


図 2　引数の実行の表示

一番わかりやすいと答えた．置換表現は現在実行中のブロックが関数で宣言されているものなのか，実引数を置き換えたものなのか分かりにくいというコメントがあった．また，複数の引数を使いたかったというコメントもあった．

## 6　おわりに

本研究では，アルゴロジックに関数を追加する際の仕様について検討した．検討した結果，引数は各関数一つだけ宣言でき，プログラムの列を受け取るようにした．この仕様でプログラムを実行した際の表示方法について，引数の表示方法が異なる 3 種類を実装した．この 3 種類の表示方法を用いて試行錯誤しながらステージを解いてもらいどの表示方法が一番わかりやすいか調査した．その結果，仮引数を実行時に実引数をコールスタックに追加する表示方法が一番わかりやすいことがわかった．また，複数の引数を使いたかったというコメントもあり，今後の課題として引数の数を再検討する必要がある．

## 参考文献

[1] 文部科学省，小学校プログラミング教育の手引き（第二版）

[2] Wing, J. M.,「Computational thinking」，Commun，ACM，Vol.49，No.3（2006），pp.33–35

[3] シーモア・パパート（1982）,「マインドストーム 子供，コンピューター，そして強力なアイデア」，奥村貴世子訳，未来社

[4] 大山祐,「アルゴリズム体験ゲーム「アルゴロジック」」，情報処理，Vol.53，No.3，pp.316–320，Mar.2012