

Ruby (ピクビー)

- Pictogramming を用いた Ruby 言語の学習環境の提案 -

伊藤 一成^{†1}

概要: 筆者は、人型ピクトグラムを用いたコンテンツ作成環境「Pictogramming (ピクトグラミング)」を提案している。Pictogramming は、プログラミング学習環境の側面も有し、様々な文脈で利活用されている。人型ピクトグラムを変形する“ピクトアニメーション”コマンドと人型ピクトグラムの体の部位の移動の軌跡を描画する“ピクトグラフィックス”の2種類のコマンドを併用することで、短時間でピクトグラムのデザイン指針に準じた多様な作品を作成できると同時に、プログラミングの諸概念が学習できるのが特徴である。ピクトグラミングはコマンドに独自の記法を採用しているが、この記法に基づくコード出力を支援する Ruby ライブラリと、Ruby 言語を学習できる Web アプリケーション「Picby (ピクビー)」を構築した。本稿では、実装方式と利用可能性を中心に論ずる。

キーワード: Pictogramming, Ruby, 人型ピクトグラム, プログラミング学習, 同調的学習

Ruby (Picby)

- Proposal of Learning Environment of Ruby Language using Pictogramming -

KAZUNARI ITO^{†1}

Abstract: We have been developing content creation environment using human pictogram, called “Pictogramming.” This application includes “Pictogram Animation Command”, which moves the parts of body and “Pictogram graphics Command”, which draws the movement history of parts of body. The combination of both type of commands enables user to create various works based on design guideline over a short period and can learn some basic concepts of programming languages. Pictogramming adopts its own original command notation, but this time we constructed a Ruby library that can output Pictogramming source code, and a web-based application that can learn Ruby language. We call this learning environment “Picby,” and discuss this availability and usefulness.

Keywords: Pictogramming, Ruby, human pictogram, programming learning, syntonc learning

1. はじめに

筆者は人型ピクトグラムを用いたコンテンツ作成環境「Pictogramming (ピクトグラミング)」を提案し Web で公開している。Pictogramming は、Pictogram (ピクトグラム) と Programming (プログラミング) を合わせた造語である。Pictogramming は、プログラミング学習環境の側面も有している。Pictogramming は、ピクトグラムを作るという目的指向で設計されているため、汎用プログラミング言語に比必定義されている命令も少なく、機能が限定されている。それが、学習コストをあまりかけなくても、ピクトグラムの作成を通じてプログラミングの諸概念 (逐次、並列、繰返し等) の理解が達成できるという利点にもなっている [1]。教育用途を意識しているため、プログラムは独自の記法を用いているが、一般的なプログ

ラミング言語の記法で記述できるように環境を拡張する意義は高いと考えられる。

そこで、日本発のプログラミング言語で、特に Web アプリケーション開発分野に広く用いられている Ruby 言語に着目した。Pictogramming の長所を継承した Ruby 言語の学習環境を構築する意義は高いと考えられる。本稿では、Pictogramming を応用した Ruby 言語の学習環境である Picby (ピクビー) を提案する。

以下 2 章で、開発の背景と動機、関連研究を説明したのち、3 章で Pictogramming の概要を示す。4 章で、今回実装した Picby について説明し、5 章でまとめる。

2. 背景、関連研究

Ruby はまつもとゆきひろ氏により開発されているスクリプト言語である。幅広い分野に関するライブラリが用意

されており、Web アプリケーション開発分野を中心に最も利用されているプログラミング言語の一つである。

Ruby は、プログラミング導入教育の文脈でも授業実践が多く報告されている[3][4][5]。一方で、長年プログラミングの教育において、興味関心の欠落、構文規則の理解の困難さ、つまずきや挫折感に伴う学習意欲の喪失等[6][7][8]が問題視されており、研究が進められている。Ruby 言語の場合でもこれらの懸念点は、十分に留意しなければならない。

岡本らは、プログラミングの概念理解に関して、出力の動作が物理的現象として見えるだけではなく、学習者がそれを認識していなければ学習上の効果が期待できないと述べ、視覚的に顕在化することの重要性を指摘している。そして、“視認性”、“判別性”、“予測可能性”、“独立性”の4つの状態から評価し有効性を示し、プログラミング教材作成の際の指針として提案している[9]。ここで岡本らが提示する、4つの側面とその具体的解決方法について表1に示す。

表1 視覚的顕在化と具体的解決方法
(文献[9]より抜粋要約)

Table 1 Visual manifestation in computer programming learning (from Ref. [9]).

視覚的顕在化の4つの状態	具体的解決方法
視認性(大きさ、速さなどが視認可能な動作である)	表示サイズや動作速度を適切に変更する
判別性(周囲の視覚的要素を区別して認識可能である)	表示位置を分離するか、他の視覚的要素から際立たせる
予測可能性(動作および動作位置が予測できる)	事前に明示するか、既存の知識や経験をもとに容易に予測できるようにする
独立性(他の命令に基づく動作と区別分離できる)	複数の動作を区別が可能なかたちに分離する

西田らは、プログラミング入門教育に図形描画の課題を取り入れたコースウェアとそうでない場合を比較し、図形描画を伴う例題を扱う方が、繰返しのようなつまずきやすい学習内容でも、理解度や楽しさを下げることなく学習できていることを示している[10]。

3. Pictogramming

本章では人型ピクトグラムに着目したコンテンツ作成環境である Pictogramming について解説する。

3.1 ピクトグラム

ピクトグラムは元来、“視認性”、“判別性”を重視したグラフィックシンボル(図記号)で一種の図形である。共通化の重要性から ISO(国際標準化機構)を中心に規格の審議や策定が進められており、例えば、案内用図記号は ISO7001、安全用の図記号は ISO7010、装置用図記号は ISO7000 など、ピクトグラムは案内、安全、施設、機器等々、様々な用途で標準化されている。また ISO3864 では、禁止、注意、指示、安全の4項目に関するピクトグラムデザイン

に関し、文字や矢印等の図形を併記する際のガイドラインが策定されている。通常、世の中に広く普及しているピクトグラムは作成ガイドラインに則りデザインされており、また伝達すべき内容が人の行為や状態に関するピクトグラムが多い。そのため ISO3864 の付録には、人間の形状のピクトグラムに特化した作成ガイドラインが提示されている。以後本論文では、このピクトグラムを人型ピクトグラムと呼称する。有名な「非常口」のピクトグラムは、デザイン策定の段階で、実際に避難中の人々が如何に出口へ向かって走る人型ピクトグラムと自身とを同一視するかにデザインの労力が払われた。太田は、“走る人型を囲む空間が見る人を包む空間とつながって走る人は見る人の投影になる”と述べている[11]。Pictogramming では、人型ピクトグラムを人間の動作に模倣して動かす。その出力は自身の動作と連関していることになる。また社会問題や社会規範を表現するというピクトグラムの特性もある。よって、高い“予測可能性”も期待できる。

3.2 実装方式と画面説明

HTML5, CSS, JavaScript を用いて実装している。人型ピクトグラムの表示・操作部に限り Processing で実装し、Processing のソースコードを実行可能な JavaScript ライブラリである Processing.js を用いて実行している。

Web アプリケーションなので、ブラウザでアクセスするだけで利用できる。またブラウザのプラグインを一切使用しない純粋な Web アプリケーションである。PC へのネイティブアプリケーションやブラウザへのプラグインのインストールが教育機関によっては禁止されている場合があるため、このような実装方式とした。日本の教育機関での利用を重視しており IE(Internet Explorer)でも動作するように実装されている。また、PC だけでなく、タブレット端末やスマートフォン上でも動作する。図1に PC のブラウザでアクセスした場合のスクリーンショットを示す。

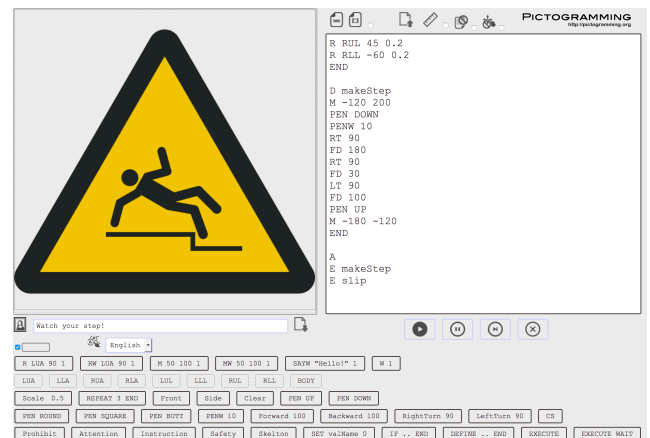


図1 Pictogramming のスクリーンショット
Fig.1 Screenshot of Pictogramming.

画面は3つの部分から構成されている。図1において上部左側は、プログラムの実行結果を表示するピクトグラム表示パネル、上部右側はプログラムを入力するプログラム記述領域、下部にはプログラムの入力を支援するボタン群が配置されている。

ピクトグラム表示パネルには、ISO3864で定義されている正面方向あるいは側面方向の人型ピクトグラムのいずれかが表示される。正面方向と側面方向の人型ピクトグラムの形状を図2に示す。いずれも体と頭を合わせた部分が1つと、上腕、前腕、上腿、下腿が左右それぞれ1つの計9種の部品から構成される。いずれの方向の人型ピクトグラムも各部位のサイズ比はISO3864で定義されているものを忠実に再現している。

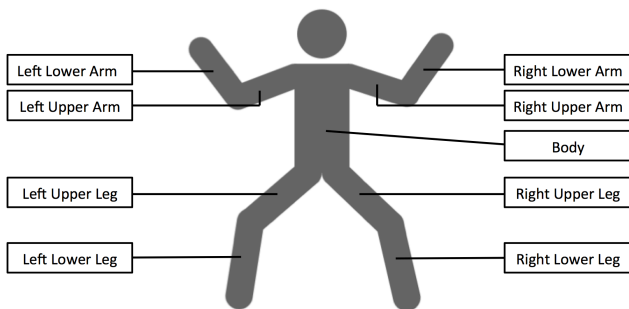
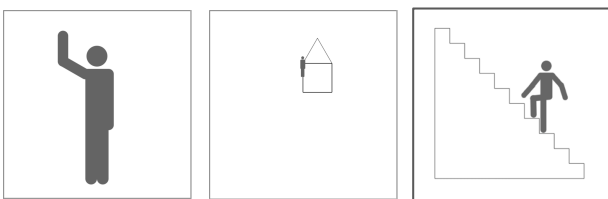


図2 人型ピクトグラムの部位の名称 (英語表記)

Fig.2 Shape and name of human pictogram.

人型ピクトグラムの変形及び動きの変化を表現できるアニメーションの機能を実装している。これを総称して「ピクトアニメーション」という。また画面上に表示される人型ピクトグラムの体の部位の移動の軌跡を図として表示する機能も有し、これを総称して「ピクトグラフィックス」という。

図3に「ピクトアニメーション」(図3の左)と「ピクトグラフィックス」(図3の真ん中)の対比を示す。両方を実装することで、図3の右に示すような、人型ピクトグラムの姿勢・動作と描画図形に関する両方の知識と経験に基づいた作品が生成可能となる。



ピクトアニメーション ピクトグラフィックス 両者の併用

図3 ピクトアニメーションとピクトグラフィックス

Fig.3 Pictogram animation and pictogram graphics.

3.3 プログラム例と実行方法

プログラムは、「プログラムコード記述領域」に入力する。命令列は、一般的なプログラミング言語と同様、上から順番に実行される。命令は先頭に「命令名」を記述し、続けて、引数列を空白で区切る以下の方式とする。

命令名 引数 1 引数 2 ...

代表的な命令を表2に示す。

表2 代表的な命令一覧

Table 2 Major command list

命令の様式	処理
R arg1 arg2 [arg3 [arg4]]	arg4秒後にarg1で指定される体の部位を反時計回りにarg2度だけarg3秒かけて支点を中心に等速回転する。arg4が省略された時は、arg4に0が、arg3、arg4の両方が省略された時はいずれも0が入力されているものとして取り扱う。
RW arg1 arg2 [arg3]	arg1で指定される体の部位を反時計回りにarg2度だけarg3秒かけて支点を中心に等速回転する。回転が終了するまで次の命令は実行されない。
M arg1 arg2 [arg3 [arg4]]	arg4秒後にarg3秒かけてx軸正方向にarg1ピクセル、y軸正方向にarg2ピクセルだけ全体を等速直線移動する。arg4が省略された時は、arg4に0が、arg3、arg4の両方が省略された時はいずれも0が入力されているものとして取り扱う。
MW arg1 arg2 arg3	arg3秒かけてx軸正方向にarg1ピクセル、y軸正方向にarg2ピクセルだけ全体を等速直線移動する。直線移動が終了するまで次の命令は実行されない。
SC arg1	人型ピクトグラムの拡大率をarg1にする。(標準は1)
FD arg1	人型ピクトグラムを進行方向に距離arg1だけ進める。(初期の方向は画面上向き)
BK arg1	人型ピクトグラムを進行方向と逆向きに距離arg1だけ進める。
RT arg1	人型ピクトグラムの進行方向を時計回り方向に角度arg1だけ回転する。
LT arg1	人型ピクトグラムの進行方向を反時計回り方向に角度arg1だけ回転する。
PEN arg1 [arg2]	arg1がUPの場合、ペンを上げる。DOWNの場合、ペンを下げる。arg1がUPまたはDOWNの時は、ペンの上げ下げをする体の部位の名称をR,RW命令と同様の表記でarg2に指定できる。arg1がFILLの時は、R,RW命令と同様の表記で指定されたarg2の体の部位の座標を含む閉じた線画の内部を塗りつぶすことができる。arg2が省略された場合はBODYが記述されているものとみなされる。
L arg1 arg2 arg3 arg4	座標(arg1, arg2)から座標(arg3, arg4)まで線分を描く。
REPEAT arg1	対応するENDまでの命令をarg1回繰返す
SK	人型ピクトグラムを縁取りの透明表示に変更する。この状態で、再度命令すると元に戻る。

ピクトアニメーションは、回転、移動命令(R,RW,M,MW命令)が主であり、これらの命令を使い分けることで、逐次実行と並列実行を組み合わせ、多様な動作を実現可能とする。ピクトグラフィックスは、教育利用も重視し、タートルグラフィックスに相当する命令(FD,BK命令等)、座標点を指定して描画する命令(L命令)が実装されている。加えて描画の有無や線種などを指定する命令(PEN命令)には、体の部位を指定でき、手や肘にペンを持たせて、その部位のピクトアニメーションの履歴を描画する形式も実装している。

ピクトアニメーション命令、ピクトグラフィックス命令のいずれも、命令の名称や引数は既存の知識や経験をもとに容易に予測できるように配慮している。さらに、人型ピクトグラムの動作をプログラムの命令を細粒度で対応させることで、高い“独立性”をPictogrammingでは実現している。

表2に示す以外にも、変数定義や繰返し、条件分岐、関数定義等の制御命令が用意されている。プログラム例を図4に示す。このプログラムは、お立ち台の上で人型ピクトグラムが50%の確率で手を振るプログラムである。

1行目は、人型ピクトグラムの表示倍率を0.3倍に変更している。2行目から7行目はお立ち台に見立てる正方形を描画している。9行目のM(ove)命令は、x軸正方向に50ピクセル、y軸負方向に200ピクセル移動する。人型ピクトグラムをお立ち台の上に移動させている。

10行目のR(otate)W(ait)命令は、左上腕L(left)U(pper)A(rm)を反時計回りに-120度、つまり時計回りに120度、1秒間かけて回転する。R(otate)命令と異なり、R(otate)W(ait)命令は、命令の実行が終了するまで、次の命令の実行を開始しない。11行目から16行目では、50%の確率で左右に3回手を振る動作を記述している。

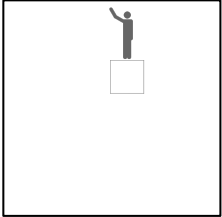
01	SC 0.3	
02	PEN DOWN	
03	REPEAT 4	
04	FD 100	
05	RT 90	
06	END	
07	PEN UP	
08		
09	M 50 -200	
10	RW LUA -120 1	
11	IF [rand(1,6)] >= 4]	
12	REPEAT 3	
13	RW LLA -60 0.3	
14	RW LLA 60 0.3	
15	END	
16	END	

図4 Pictogrammingのプログラム例
Fig.4 Example of Pictogramming's program.

実行は画面上に配置されている実行ボタンのほか、プログラム入力中でも改行コードが入力された時点で、文法エラーが発生しなければ、実行される。

また、教育利用を重視しているため、既存の教育向けプログラミング言語[14]にあるように、命令コードは、英語の略記だけでなく、英語の単語表記、日本語表記、さらには低年齢の利用も想定し、日本語ひらがな表記をサポートしている。変数名も日本語をサポートし、数値、算術・比較演算子やスペースが全角文字で表記されていても対応する半角文字が入力されていると想定し動作する。例えば、13行目の「RW LUA -120 1」は「回転待ち 左肩 -120 0 1」でも同じく動作する。

3.4 同調的学習

LOGOを開発したPapertは、子どもが自分自身の体を使ってタートルになったふりをする中で、LOGOの命令を実行することができるという特徴に大きな重要性を見出し、これを同調的学習と呼んだ[13]。書籍[13]では、自分の身体に対する感覚や知識と強く結びついている(身体同調)こと、意図や目的、欲求、好き嫌いを持った人間としての自意識と一貫している(自我同調)こと、文化にしっかりと肯定的に根を張った活動に結びついている(文化同調)ことが示されている。図5にPictogrammingで作成された作品例を示す。

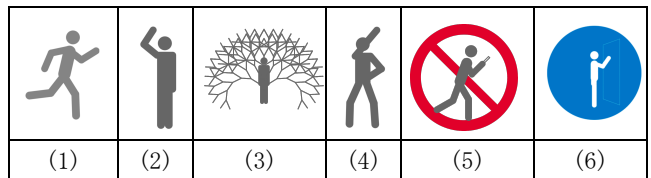


図5 Pictogrammingの作品例

Fig.5 An example of works using Pictogramming.

図5の「(1)走る」、「(2)手をあげる」のような自分の身体に対する感覚や知識と強く結びついている(身体同調)作品、図5の「(3)キューピット」「(4)人気アニメキャラクターのポーズ」のような意図や目的、欲求、好き嫌いを持った人間としての自意識と一貫している(自我同調)作品、図5の「(5)歩きスマホ禁止」「(6)ドアはノックしましょう」など文化にしっかりと肯定的に根を張った活動に結びついている(文化同調)作品などが作成される。

3.5 認知視点の多重化

岡本は、Normanの「体験的認知」「内省的認知」[14]、Laurelの「一人称的関与」「三人称的関与」[15]の2種類の認知的関与を示し、認知主体自身が思考し内省する空間と、認知する対象としての外部世界に対応し、認知主体として

のユーザがこれら二重の認知モードを使い分けられるような状態の確立こそが、システム-ユーザ間コミュニケーションを円滑にする条件の一つであると述べている[16]。岡本は、本人が対象と同一化されていると、通常は視点の二重性を意識されず、一方、傍観者として他人同士のインタラクションやコミュニケーションを観察する場面で、その場から離れてあくまでも他人事として関与する限りにおいてもそうした視点の二重性は成立しないことを示している。

認知主体が共感によって二重の視点を持ち、二重の認知空間を横断することが、対象空間としての仮想世界への没入を可能にしており、同時にその共感によって、他者と自己同一化を行うことで、対象空間における当事者性を増大させインタラクションへの心理的関与を生み出すとしている。

Pictogramming では、認知視点の多重化を重視しており、「ピクトグラフィックス」は、認知視点の異なる 3 種類の描画方式を実装している。四角形を例に 3 種類の描画方式による描画を図 6 に示す。

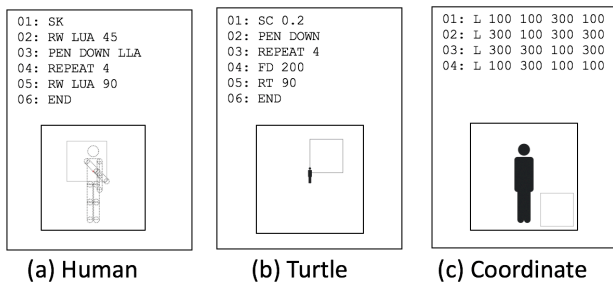


図 6 Pictogramming による図形描画の例
Fig.6 An example of figure drawing using Pictogramming.

図 6 の(b)は、人型ピクトグラム自体の移動の履歴で描画する方式で、タートルグラフィックスと同等である。タートルグラフィックスでは、人型ピクトグラムをシステムのエージェントとみなし、学習者自身が人型ピクトグラムとして同一視する視点（一人称視点）と、俯瞰的に観察や指揮をとることを想定した視点（三人称視点）が相互作用することを期待している。図 6 の(c)は、デカルト座標系で線分を描画する L 命令を 4 回用いて、座標(100, 100), (300, 300)を対頂点とする正方形を描画する例である。これはプログラミング言語で図形描画する際の一般的な方式である。アフィン変換などの処理により、三人称視点内での視線や視点の変換は行われるものの、図 6 の(b)のような人称視点の転換は起こりにくい。図 6 の(a)は、体の部位を指定してその部位の動きの履歴で描画する方式で、ヒューマングラフィックスと呼称する。この例では、左手の動きで正方形を描画している。図 6 の(b)のような歩行の移動の履歴だけではなく、ヒューマングラフィックスでは、多様で、より日常的な人間の動作、行為に関連させて図形描画することにより、共感を喚起させ、

人型ピクトグラムや他の参加者、他の参加者が創作する人型ピクトグラムへの自己同一化を促す。一方で、人型ピクトグラムという、人を抽象化、一般化した表現を使用することで、人型ピクトグラムを客観視（三人称視点）して創作することも喚起する。

4. Picby (ピクビー) の提案

本章では、提案する Ruby で記述可能なピクトグラムコンテンツ作成環境 Picby (ピクビー) について説明する。

4.1 概要

Pictogramming はピクトグラム生成に特化しているため、定義されている命令の種類は多くない。Ruby と Pictogramming のトランスパイラを開発するのは、使用可能な Ruby の機能を大幅に限定させることになる。そこで今回は、Pictogramming の操作命令のみをプログラム文字列を標準出力に出力する Ruby プログラムを学習者が実装する方式を採用する。

4.2 実装方式

Picby クラスを定義する Picby ライブラリを実装した。Picby ライブラリを用いて、図 4 の Pictogramming のプログラム例を Ruby で、プログラムコード記述領域に記述した例を図 5 に示す。実際には、ライブラリファイルでは、RW,M 等の Pictogramming で定義される諸命令をメソッド定義せず、method_missing を用いる。method_missing とは、呼び出されるメソッドが未定義の場合でも、エラーを発生させるのではなく、特定のメソッドに、呼び出されたメソッド名と引数列の両方を与えて実行する仕組みである。これを利用して、Pictogramming 形式のプログラムの文字列を生成する。

Ruby では、メソッドの引数はカンマ区切りで与えられ、一方、Pictogramming では 3.2 節で紹介したように、命令や引数の間を空白文字で区切っている。

Pictogramming の命令の引数には、体の部位や図形描画のペンのタイプを表すラベル文字列がいくつか定義されている。それらをメソッド名の一部に組み入れて表記を単純にできる。そこで、メソッド名中にアンダーバー("_")が入力されている場合も、空白文字に変換する。例えば、Pictogramming で PEN DOWN という記述は、Ruby では、pen "down" だけではなく、pen_down のようにも記述できる。つまり、method1_method2 arg1, arg2, arg3 は method1 method2 arg1 arg2 arg3 に変換され標準出力される。Pictogramming で記述する操作命令は、この変換ルールで全て記述可能である。

図 4 の Pictogramming のコードと同一の動作をする Picby ライブラリを用いた Ruby 形式のプログラムを図 7 に示す。実際には、このコードの前に、Picby ライブラリの読み込み、各種モジュールの読み込み、pic インスタンス

の生成の処理が追加される。図 7 の 3 行目, 12 行目の繰返し文, 11 行目の if 文の定義文や制御文は, Ruby の文法に基づくものであり, このコードのまま Ruby で実行される。

よって, ループ展開, 手続き呼び出しの展開, 条件判定の展開などが行われるため, 実際に生成される Pictogramming のコードは図 4 とは異なる。

```

01 pic.sc 0.3
02 pic.pen_down
03 4.times do
04   pic.fd 100
05   pic.rt 90
06 end
07 pic.pen up
08
09 pic.m 50, 200
10 pic.rw "lua", -120, 1
11 if rand(6) + 1 >= 4 then
12   3.times do
13     pic.rw "lla", -60, 0.3
14     pic.rw "lla", 60, 0.3
15   end
16 end

```

図 7 Picby ライブラリを用いた Ruby のプログラム例
Fig.7 Example of Ruby program using Picby library.

4.3 利用方式

2 方式を実装した。利用方式の概要図を図 8 に示す。

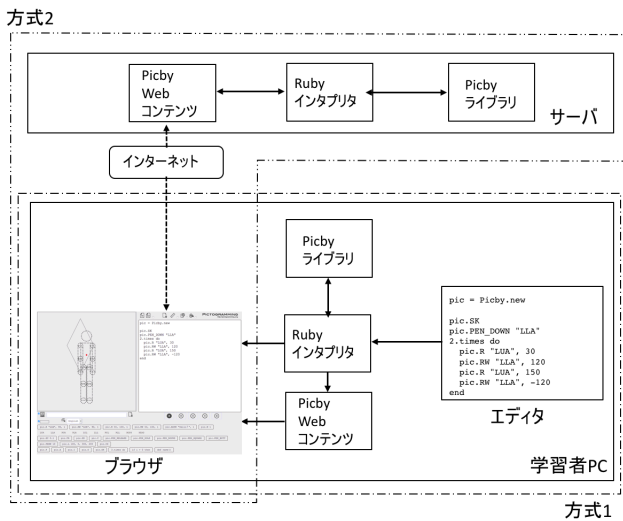


図 8 利用方式の概要図
Fig.8 Outline of Usage

一つ目の方式は, 図 8 の一点鎖線で示す領域 (図 8 中の方式 1) のように, 学習者が自由な統合開発環境やエディタで Ruby 形式のプログラムを作成し, ローカルの PC 上にインストールされている Ruby インタプリタを使用して, プログラムを実行する。これは, Ruby インタプリタに標準で内包されている turtle ライブラリと同様の方式である。自由度も高いため, 様々な応用アプリケーションや応用システムの開発が見込める。また, Pictogramming では,

アプリケーション上に配置されたボタンを押下することで実行される様々な処理は, Picby ライブラリで定義されており, ライブラリのメソッドを呼び出して実行する。

二つ目の方式は, 図 8 の二点鎖線で示す領域 (図 8 中の方式 2) のように, Pictogramming とほぼ同一のユーザーインターフェースを備えた Web アプリケーションをインターネット上に提供する。図 9 にスクリーンショットを示す。

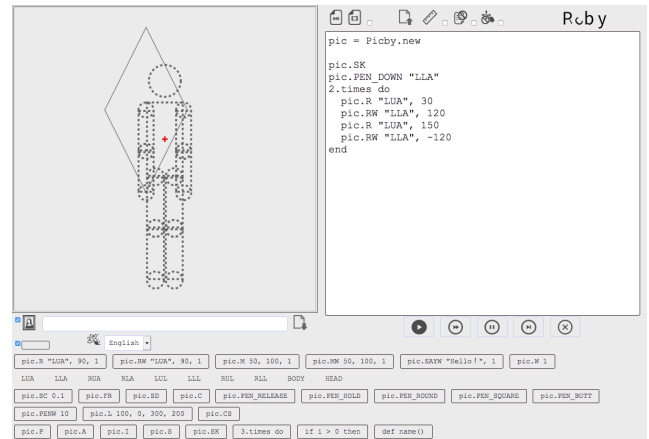


図 9 Picby のスクリーンショット
Fig.9 Screenshot of Picby.

プログラム記述領域に記述するプログラムが Ruby 形式となり, 画面下部の命令入力支援ボタンで自動入力される命令が Ruby 形式である以外は Pictogramming と同様である。学習者の PC に Ruby インタプリタやエディタのインストールが必要ないため, 学習環境の整備が困難である教育機関でも利用可能である。

5. おわりに

本稿では, 人型ピクトグラムを用いた Ruby 言語の学習環境「Picby」について解説した。今後は, 授業利用や評価実験を通じて, 本提案の有効性を示していく。

久野らは初等中等段階を通じた情報教育の必要性を論じ, カリキュラム体系を提案している[15]。Pictogramming は, 小学校から高等学校に至るまで様々な切り口で活用されている。その理由としてやはり人型ピクトグラムという抽象度の高い人間を模した表現が主題にあるゆえに, 人の日常的活動や社会的活動にリンクしたテーマが教科や教育段階の枠を超えて設定しやすいことがあげられる。Picby も言語の特徴を鑑みると中学, 高等学校を中心に活用が期待できる。

Pictogramming の Web アプリケーションは継続して機能追加を進める。これにより Picby ライブラリ, Picby Web アプリケーションの機能も自動的に追加される。また, この方式を用いて Ruby 言語以外のプログラミング言語を対象にした学習環境への応用が容易に可能であり, 現在 Picby に加え, Python 言語版も開発中である[19]。今後,

参考文献

- [1] 伊藤一成. ピクトグラミング - 人型ピクトグラムを用いたプログラミング学習環境 - 情報処理学会論文誌 教育とコンピュータ, 2018, vol. 4, no. 2, pp.47-61.
- [2] 高等学校情報科「情報 I」教員研修用教材 (本編)
http://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1416756.htm
- [3] Wolfgang Müller and Ulrich Kortenkamp: Learning Programming with Ruby, Proceedings of the IFIP Workshop “New developments in ICT and Informatics education”, 2010
- [4] Stacey Watson and Heather Richter Lipford.: Motivating Students Beyond Course Requirements with a Serious Game. In Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19), 2019, pp. 211-217.
- [5] 森畑明昌. プログラミング入門をどうするか : 6. 東京大学における全学プログラミング. 教育情報処理学会誌, 2016, vol.57, no.4, pp. 362-365
- [6] 岡本雅子. ペタ語義 : はじめてのプログラミングとつまづき. 情報処理学会誌, 2015, vol.56, no.6, pp. 580-583
- [7] 長谷川聡, 山住富也. プログラミング教育と学習者のイメージ形成(その 2), 名古屋文理短期大学紀要, 1997, vol.23, pp.9-14.
- [8] Essi Lahtinen, Kirsti Ala-Mutka, and Hannu-Matti Järvinen. 2005. A study of the difficulties of novice programmers. In Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education (ITiCSE '05). ACM, New York, NY, USA, 14-18.
- [9] 岡本雅子, 村上雅之, 吉川直人, 喜多一. 「視覚的顕在化」に着目したプログラミング学習教材の開発と評価. 日本教育工学会論文誌, 2013, vol. 37, no. 1, p.35-45.
- [10] 西田知博, 原田章, 中西通雄, 松浦敏雄. プログラミング入門教育における図形描画先行型コースウェアが学習に与える影響. 情報処理学会論文誌 教育とコンピュータ, 2017, vol. 3, no. 1, p.26-35.
- [11] <http://www.forum8.jp/topic/pictogram119.htm>
- [12] 伊藤一成: ピクトグラミングを用いたデザイン教育とプログラミング教育の融合, 日本デザイン学会 第 66 回研究発表大会, 2019
- [13] Papert, S. Mindstorms: Children, computers, and powerful ideas. Basic Books Inc, 1980.
- [14] Norman, D. A. Things that make us smart: Defending human attributes in the age of the machine. Basic Books, 1993.
- [15] Laurel, B. Interface as mimesis. User centered system design: New perspectives on human-computer interaction, 1986, p. 67-85.
- [16] Okamoto, M. Nakano, I. Y. and Nishida, T. Toward enhancing User Involvement via Empathy Channel in human-computer interface design. Intelligent Media Technology for Communicative Intelligence. Springer Berlin Heidelberg, 2005, p. 111-121.
- [17] 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, 2001, vol. 42, no. 12, p.78-90.
- [18] 久野靖, 和田勉, 中山泰一. 初等中等段階を通じた情報教育の必要性とカリキュラム体系の提案. 情報処理学会論文誌 教育とコンピュータ, 2015, vol. 1, no. 3, p.48-61.
- [19] 伊藤一成. Pichthon (ピクソン) Pictogramming を用いた Python 言語の学習環境の提案 -, 情報処理学会 情報教育