

Pict_othon (ピクソン)

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

伊藤 一成

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。Pictogramming はコマンドに独自の記法を採用しているが、Python 言語を使って Pictogramming を利用できるように拡張した。実際に、(1)Pictogramming の記法によるコード出力を支援する Python ライブラリと(2) Python 言語で記述できる Web アプリケーション「Picthon (ピクソン)」を実装した。本稿では、その実装方式と利用可能性を中心に論ずる。

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

Pict_othon (Picthon)

- Proposal of Learning Environment of Python Language using Pictogramming -

KAZUNARI ITO

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 Python library that can output Pictogramming source code, and a web-based application that can be coded with Python language. We call this learning environment “Picthon,” and discuss this availability and usefulness.

Keywords: Pictogramming, Python, human pictogram, programming learning, syntonic 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べ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、Python 言語に着目し、Pictogramming の特徴を継承した Python 言語の学習環境 Picthon を提案する。

以下 2 章で、開発の背景と動機を説明したのち、3 章で Pictogramming の概要を示す。4 章で、今回実装した Picthon について説明し、5 章で有用性や利用可能性を中心に考察し、6 章でまとめる。

2. 背景

Python は動的型付け、マルチパラダイムなどの特徴を有するインタプリタ型のプログラミング言語である。機械学習、データサイエンス、行列演算など、科学技術関連を中心に幅広い分野に関するライブラリが用意されており、近年、最も利用されているプログラミング言語の一つである。

高等学校では 2022 年度からの次期学習指導要領において、情報の科学的理解を基軸とする「情報 I」が必修科目となる。2019 年 5 月に文部科学省から公開された情報 I 教員研修用資料[2]でも、プログラミング単元の解説に Python が取り入れられており、時流や政府戦略に沿ったテキスト型プログラミング言語重視の方針が伺える。大学においても、AI やデータサイエンス分野だけでなく、プロ

グラミング導入教育にいたる幅広い文脈で Python を用いた授業実践が多く報告されるようになってきた[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。

例えば、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、視覚的に頭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]。岡本らが提示する、4つの状態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)

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

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いる．通常，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．Pictogramming では，この人型ピクトグラムに注目している．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 を用いて実装されている。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活用を重視しており 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。

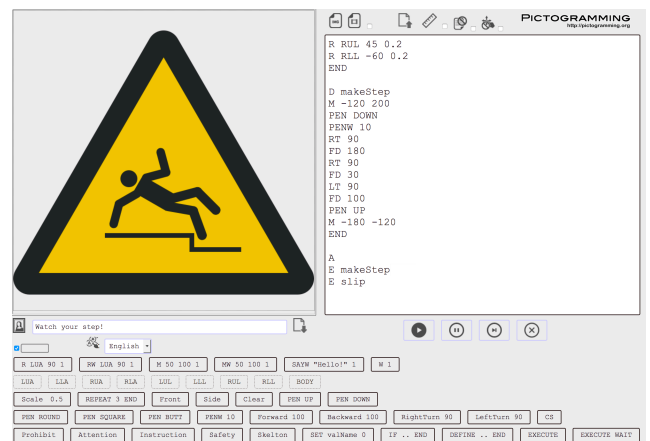


図 1 Pictogramming のスクリーンショット

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。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。

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(「ピクトアニメーション」という)と、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(「ピクトグラフィックス」という)がある。この両者を用いて、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を生成できる。

3.3 命令の様式とプログラム例

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

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 回繰り返す
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 2 に示す。

01	D DRAW_RECT :LEN	
02	PEN DOWN	
03	REPEAT 4	
04	FD :LEN	
05	RT 90	
06	END	
07	PEN UP	
08	END	
09		
10	SC 0.3	
11	E DRAW_RECT 100 //お立ち台を描く	
12	M 50 -200 //お立ち台に登る	
13	RW LUA -120 1 //左腕を斜めにあげる	
14	IF [rand(1,6) >= 4]	
15	REPEAT 3 //左右に手を振る	
16	RW LLA -60 0.3	
17	RW LLA 60 0.3	
18	END	
19	END	

図 2 Pictogramming のプログラム例

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 行目から 8 行目はお立ち台に見立てる正方形を描画するプロシージャ DRAW_RECT を定義している。

10 行目以降が処理の部分で、10 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。11 行目で、先に定義した DRAW_RECT を呼び出して正方形を描画している。12 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。13 行目の R(otate)W(ait)命令は、左上腕 L(eft)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)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命令である。逐次処理と並列処理を組み合わせることで、様々な動きを実現できる。14 行目から 19 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。

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[10]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、さらには低学齢の利用も想定し、日本語ひらがな表記をサポートしている。

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事で、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だし、これを同調的学習と呼んだ[11]。書籍[11]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3にPictogrammingで作成された作品例を示す。

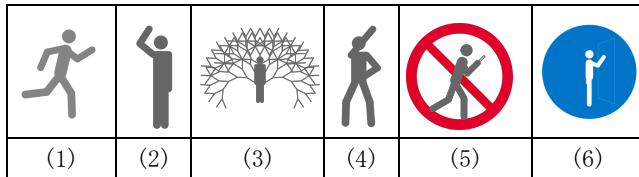


図3 Pictogramming の作品例

図3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3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3の「(5)歩きスマホ禁止」「(6)ドアはノックしましょう」など文化的活動に結びついている（文化同調）作品などが作成される。実際に数十分の比較的短時間の実習で同調的学習が認められる作品が作成されることが示唆されており[12], 同調的学習と相性が良いと考えられる。

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」[13], Laurel の「一人称的関与」「三人称的関与」[14]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[15]。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4に示す。

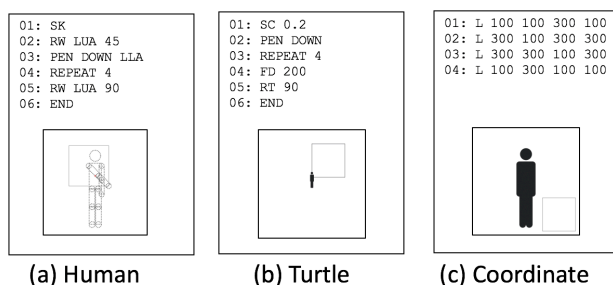


図4 Pictogramming による図形描画の例

図4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4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。本稿では“座標グラフィックス”という。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4の(b)のような人称が変化する視点の転換は起こりにくい。図4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4の(b)のような歩行の移動の履歴だけではなく, 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）して創作するモードも喚起する。

4. Picthon (ピクソン) の提案

4.1 実装方式

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。Python のコードと Pictogramming のコードを相互変換するトランスパイラを開発するのは, 使用可能な Python の機能を大幅に限定させることになる。そこで今回は, Pictogramming の操作命令に限定し, 対応するプログラム文字列を標準出力に出力する Python クラスを定義し, picthon ライブラリとして実装した。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していない。method_missing に相当する処理が Python では実現可能で, これを利用している。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 形式のプログラムを文字列で生成する。

4.2 利用想定

2通りの利用を想定する。概要図を図5に示す。

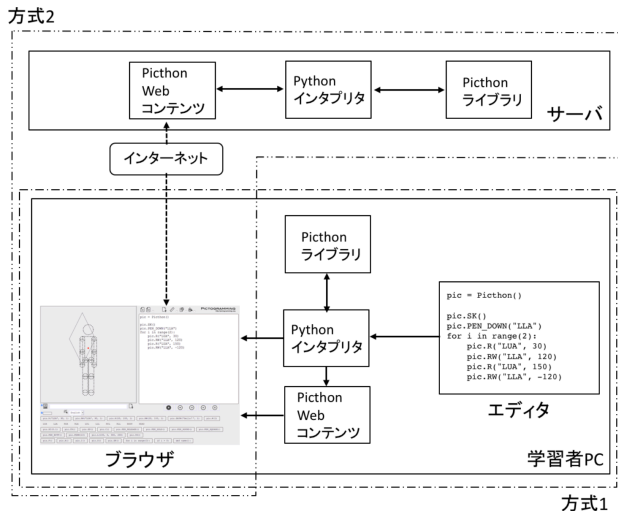


図5 利用方式の概要図

一つ目の方式は、図5の一点鎖線で示す領域（図5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Python形式のプログラムを作成し、同一PC上にインストールされているPythonインタプリタを使用して、プログラムを実行する。これは、Pythonインタプリタに標準で含まれるturtleライブラリ（タートルグラフィックスを描画するためのライブラリ）と同様の方式である。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。

二つ目の方式は、図5の二点鎖線で示す領域（図5中の方式2）のように、Pictogrammingとほぼ同一のユーザインタフェースを備えたWebアプリケーション上でプログラムを実行する。Webアプリケーションのスクリーンショットを図6に示す。

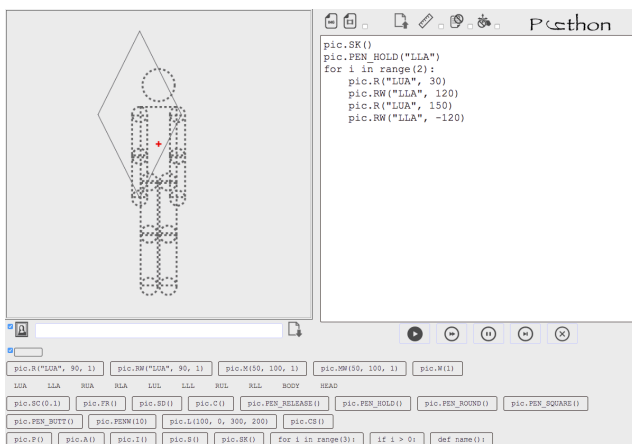


図6 Picthionのスクリーンショット

図6の上部右側のプログラム記述領域に入力されるのがPythonのプログラムであり、画面下部の命令入力支援ボタンで自動入力されるのがPythonのプログラム片である以外はPictogrammingと同様である。学習者のPCにPythonインタプリタやエディタのインストールが必要な

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。

4.3 命令の様式とプログラム例

Pythonでは、メソッドの引数はカンマ区切りで与えられ、引数列を丸括弧で囲う記法である。一方、Pictogrammingでは3.3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。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、Python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2のPictogrammingのコードと同等の動作をするPicthionライブラリを用いたPython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。

```

01 from picthion import Picthion
02 import random
03
04 def draw_rect(self, len):
05     self.pen_down()
06     for i in range(4):
07         self.fd(len)
08         self.rt(90)
09     self.pen_up()
10
11 Picthion.draw_rect = draw_rect
12
13 pic = Picthion()
14 pic.sc(0.3)
15 pic.draw_rect(100)
16 pic.m(50, -200)
17 pic.rw("lua", -120, 1)
18 if random.randrange(1, 6, 1) >= 4:
19     for i in range(3):
20         pic.rw("lla", -60, 0.3)

```

図7 Picthionライブラリを用いたPythonのプログラム例

4行目から9行目のメソッド定義、6行目、19行目のfor文、18行目のif文の定義文や制御文は、Pythonの文法に基づくものであり、このコードのままPythonで実行される。よって、ループ展開、手続き呼び出しの展開、条件判定の展開が行われるため、実際に生成されるPictogrammingのコードは図2とは異なる場合がある。

5. 実践・評価・考察

5.1 実践

被験者は、神戸大学附属中等教育学校に通う中学3年生4クラスの生徒計131名である。実践期間は、クラスによって多少の差異はあるが、全体では、2019年6月17日(月曜日)から2019年7月12日(金曜日)である。1回の授業時間は50分で週2コマ別日に実施される。一般的なPC実習室が配当されている。授業実践の概要を表3に示す。学校行事等の影響で、4クラスの2クラスは表3の第6回を除く計6回、2クラスは表3の通り計7回の授業を行った。今回の実践では、独自の開発環境を用いる方式1は実施不可能であったので、方式2のWebアプリケーション方式を使用した。第1回の冒頭では、ピクトグラムの歴史やピクトグラムの社会での利用例について説明し、ピクトグラムについての関心を喚起させた。第2回は、4クラス中2クラスでネットワーク障害により、インターネットに接続できなかったため、教員卓上のみでピクトグラミングをローカルアクセスし、生徒はコンピュータを使わずに、提示されたプログラムを自分自身の体を使って実行する人間プログラミングの方法で遂行した。そのため、第3回のはじめにコンピュータを用いて第2回の内容を復習する時間を全クラス設けた。

表3 実践の概要

回	内容	時間(分)
1	1.ピクトグラムに関する講義	30
	2. Picthon の操作方法	15
2	1. 逐次実行, 並列実行	45
3	1. 逐次実行, 並列実行 (復習)	10
	2. 繰り返し	35
4	1. 変数, リスト	45
5	1. ヒューマングラフィックス	45
6	1. タートルグラフィックス, 座標グラフィックス	45
7	1. アートとデザインの違いに関する講義	10
	2. 自由制作 (デザインの観点から)	35

5.2 アンケート評価

4章で示した Picthon を用いた授業実践の内容の妥当性を評価する目的で各回の最後に無記名式のアンケートを行った。質問項目は表4に示す8項目である。

質問1～7は、いずれも6段階の回答選択肢を用意した。回答選択肢の内容と数量化は、「6.とてもそうである」、「5.かなりそうである」、「4.どちらかというそうである」、「3.どちらかというそうでない」、「2.あまりそうでない」、「1.全くそうでない」とした。質問番号5,6はピクトグラムの特徴と言われているユーモアさとシュールさについて調査した。質問8は自由記述形式である。質問1～7の回答結果を図8に示す。また、質問1～7の各質問の数量化した値の平均値を表5に示し、表6に質問8“今回の授業の感想

を自由に書いてください(自由記述)”の記述の抜粋を示す。

表4 アンケート項目

Q	内容
1	今回の授業内容はあなたにとって楽しかった
2	自分の作りたいと思う作品が設定できた
3	作品を作る過程において、自分の身体の動きを意識した
4	自分の作りたいと思った作品を作成できた
5	作品を作る過程で、ユーモアさを感じた
6	作品を作る過程で、シュールさを感じた
7	今回の授業内容はあなたにとって難しかった
8	今回の授業の感想を自由に書いてください (自由記述)

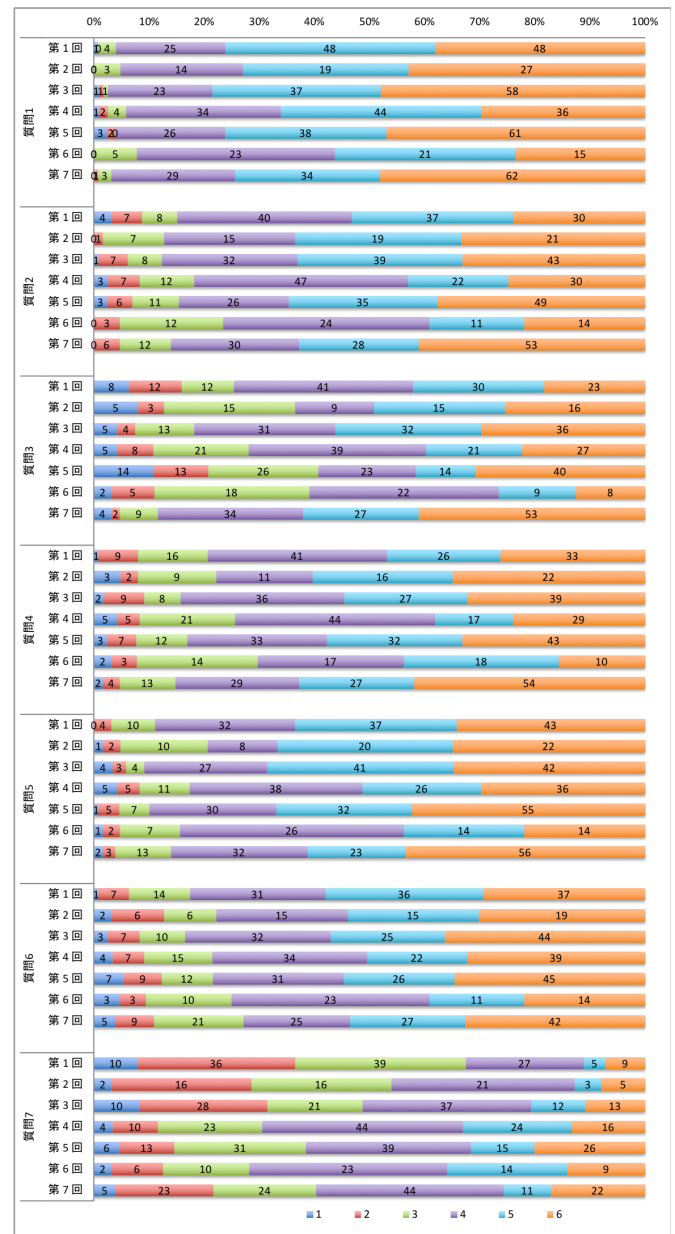


図8 アンケートの回答結果

表 5 質問ごとの平均値

回	質問 1	質問 2	質問 3	質問 4	質問 5	質問 6	質問 7
1	5.087	4.500	4.127	4.437	4.833	4.627	3.063
2	5.111	4.825	4.175	4.603	4.746	4.460	3.329
3	5.215	4.752	4.562	4.603	4.851	4.661	3.430
4	4.868	4.388	4.190	4.240	4.512	4.488	4.008
5	5.131	4.777	4.000	4.638	4.938	4.500	3.938
6	4.719	4.328	3.859	4.188	4.438	4.219	4.063
7	5.186	4.853	4.837	4.837	4.853	4.442	3.767

表 6 “今回の授業の感想を自由に書いてください”抜粋

回	内容
1	回答 1. 自分の体の動きを意識して作れたのでとても楽しかったです。 回答 2. 普段見かける絵記号（ピクトグラム）を作ることができることを知りました。楽しかったです。
2	回答 1. 仕組みがわかると、この前の授業よりもより面白かった。 回答 2. いろいろな動きを文字で表すことができ、人間にはできないような動きもアニメーションで見れて面白かった。
3	回答 1. 繰り返すために何個も命令を書くのではなくて、繰り返しをさせる命令の表記があって便利だなと感じました。 回答 2. ピクトグラムに自分がなりきって、走っている姿を繰り返しを使いながら学ぶことができた。またいろいろなものをつくって見たいなと感じた。
4	回答 1. angle などを使うと書き直さなくて済むので便利だなと思いました。 回答 2. 変数はわかって、使えるけれど、リストになると書かなければいけない命令も増えて、少し難しく感じました。
5	回答 1. 角度や時間を少し変えるだけで、全く違う図形が出来たのが面白かった。 回答 2. ほかの人が作ったものを見て、どうやってそういうものを作ったのが気になりました。
6	回答 1. ピックが小さくなるのがかわいかった。ピックの動かし方をちゃんと教えてほしい。 回答 2. 記号がたくさんありすぎて、いちいちテキストを見てやるのに時間がかかった。
7	回答 1. 学校やマンション内、街中などでいろいろなピクトグラムを見るけれど、実際に作ってみると一目見て伝えるものを作るのは工夫が必要なんだなとわかった。 回答 2. ピクト君と別れるのが名残惜しかった。 回答 3. ほかの人の作品がすごくてとても感動した。いいことを学べたと思うので家でも時間があれば挑戦したいです。

質問 1 “今回の授業内容はあなたにとって楽しかった”について、第 4 回、第 6 回以外の全ての回で、「6. とてもそうである」、「5. かなりそうである」、「4. どちらかというとそうである」の回答が 95%以上を占めた。第 4 回、第 6 回でも 90%以上を占めた。楽しみながら受講できることは学習において非常に重要な要素であり、その条件は満たしていると言える。

短時間での演習の場合、テーマを簡単に設定することが非常に重要な要素である。質問 2 “自分の作りたいと思う作品が設定できた”の回答結果は、さまざまなタイプの同調的学習に対する高い親和性によるものと考えられる。第

4 回、第 6 回についてやや平均値が低い、いずれもプログラミングの概念や座標系などの要素が学習内容に含まれていたため、それを含んだ自由作品を作成するという条件が制約となった可能性が示唆される。

質問 3 “作品を作る過程において、自分の身体の動きを意識した”では、第 5,6 回で他の回に比べ、平均がやや低いが、グラフィック描画の回は、描画される図形に注目が行くため、自分の身体の動きへの意識が薄れたのではないかと、推測する。今後一層の分析が必要である。

質問 4 “自分の作りたいと思った作品を作成できた”について、授業での運用を考えると、限られた時間の中でも重要な要因の一つである。第 4 回、第 6 回についてやや平均値が低い、質問 2 と同様に、いずれもプログラミングの概念や座標系などの要素が学習内容に含まれていたためである可能性が示唆される。

質問 5 “作品を作る過程で、ユーモアを感じた”すべての回において、「6. とてもそうである」、「5. かなりそうである」、「4. どちらかというとそうである」の回答が 80%以上を占めており、平均の値は高い。ただし第 4 回、第 6 回については、「4. どちらかというとそうである」以下の回答が他の回に比べてやや多い。ユーモアは、学習者間のコミュニケーションを促進させる一要素であるという先行研究がいくつかある[16]。

シニールとは表現や発想が非日常的・超現実的であるさまをいう。誤った命令を入力すると通常の間ではありえないポージングとなり、それがシニールを生み出していると考え、質問 6 “作品を作る過程で、シニールを感じた”の回答から、どの回も 7 割以上の生徒がシニールを感じていることがわかる。

質問 7 の回答を表 11 に示す。第 1,2 回、やや「6. とてもそうである」、「5. かなりそうである」の人数が少ないものの、どの回も、評価は 1 から 6 まで適度に分散しており、難易度に関しても無理のない設定であったと考えられる。

表 6 は抜粋であるが、質問 8 の回答から以下のような点が明らかになった。

- 第 1 回の回答 1, 第 3 回目の回答 2 に例示するように、同調的学習に関する記述が多く見られた。
- 第 5 回の回答 2 や第 7 回の回答 3 にあるように、視認性の良さから周囲の生徒の実行の様子が目に入ることに起因する社会的比較に関する記述も見られた。
- 第 4 回の回答 2 にあるように、変数やリストの回は、記述が難しかったという回答が複数見られた。また第 6 回について、タートルグラフィックスは、第 5 回までのように人の体の部位の動作を記述するのではなく、人の移動の履歴で描画するため、認知視点が異なっていたこと、これまでと異なる命令体系であった。また、座標グラフィックスは、座標系の概念が必要である。そのため、短い実習時間

の中では十分な理解に至らなかった生徒が少なからずいたことが推測される。これらの理由が、第4回、第6回の質問1~7の平均値の値に影響したものと考えられる。

d) 全体的に、「楽しかった」「面白かった」という質問1を肯定する記述が大半を占めた。第7回目の回答2に見られるように、人型ピクトグラムを擬人化した感想も多く見られた。さらに第7回目の回答3に見られるように、授業時間外にも利用を希望する生徒が複数存在し、創作意欲を喚起しているのが窺いしれる。

6. おわりに

本稿では、人型ピクトグラムを用いた Python 言語の学習環境「Picthon」について解説した。Picthon は、Pictogramming と同様に Web 公開する予定である。

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[17]。2022年度より高等学校の必修教科目となる「情報I」の「情報デザイン」「コンピュータとプログラミング」領域との関連性が高いのはいうまでもない。また中学校でも、新学習指導要領の「技術・家庭科」で、計測・制御のプログラミングに加え、インタラクティブなコンテンツを生成するプログラミングも追加される。人型ピクトグラムは、四肢二関節のロボットとみなすことができ、ロボット制御のシミュレータソフトウェアとしての活用が想定できる。またピクトグラムというインタラクティブコンテンツ生成の視点のプログラミング環境としても当然活用できる。

Pictogramming は、小学校から高等学校に至るまで様々な切り口での活用されている。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を主題にあるゆえに、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考えられる。Picthon も Pictogramming のこれらの特性を継承しているのはいうまでもない。

しかし、Python は教育向けに特化したプログラミング言語に比べ、文法や記法が複雑であるため、2章で述べた、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の問題をはらんでいる。適用する学齢や対象に応じて十分に留意しなければならず、今後は、授業利用や評価実験を通じて、本環境の有効性を検証し、改善すべき点を分析、評価する。

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 Picthon ライブラリ、Picthon Web アプリケーションの機能も追加されることになる。また Python 以外のプログラミング言語を対象にした学習環境についても、同様の実装方式で開発していく予定である。

謝辞

授業の実践及び評価の機会を提供いただいた神戸大学附属中等教育学校 米田貴先生に感謝いたします。

参考文献

- [1] 伊藤一成. ビクトグラミング - 人型ビ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] Simon, Raina Mason, Tom Crick, James H. Davenport, and Ellen Murphy. 2018. Language Choice in Introductory Programming Courses at Australasian and UK Universities. In Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18), pp. 852-857.
- [4] Jessica Q. Dawson, Meghan Allen, Alice Campbell, and Anasazi Valair. 2018. Designing an Introductory Programming Course to Improve Non-Majors' Experiences. In Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18), 2018, pp. 26-31.
- [5] David G. Sullivan. 2013. A data-centric introduction to computer science for non-majors. In Proceeding of the 44th ACM technical symposium on Computer science education (SIGCSE '13). ACM, New York, NY, USA, pp.71-76.
- [6] 岡本雅子. ペタ語義: はじめてのプログラミングとつまずき. 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)*, 2005, pp.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, pp.35-45.
- [10] 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, pp.78-90.
- [11] Papert, S. *Mindstorms: Children, computers, and powerful ideas*. Basic Books Inc, 1980.
- [12] 伊藤一成: ビ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] Norman, D. A. *Things that make us smart: Defending human attributes in the age of the machine*. Basic Books, 1993.
- [14] Laurel, B. *Interface as mimesis. User centered system design: New perspectives on human-computer interaction*, 1986, p. 67-85.
- [15] 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.
- [16] J. W. Neuliep. An examination of the content of high school teachers' humor in the classroom and the development of an inductively derived taxonomy of classroom humor. *Communication Education*, 1991, vol. 40, no. 4, pp. 343-355
- [17] 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和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, pp.48-61.