

ハイパーピクトッチ：コンピューターショナルシンキングを育成するブロック型ピクトグラム制作環境の開発

伊藤一成^{1,a)}

概要： 計算論的思考 (Computational Thinking; CT) の育成を目指し、ブロック型ピクトグラム制作環境「ハイパーピクトッチ」を提案する。本アプリケーションは、人型ピクトグラムや物体を独立したオブジェクトとして定義し、並行実行・メッセージパッシング・条件分岐・手続き化により複数主体の相互作用や場面の複合性を表現できる点が特徴である。また、自由配色への対応により色彩学習への展開も可能である。本研究が対象とするピクトグラムは、形態の単純化と高コントラストによる高い視認性、言語に非依存の普遍性、規格化に基づく一貫性を有する。本アプリケーションでは、これらの特性に加えアニメーション生成機能により抽象的な動作や内的状態の表現を可能にした。これにより行為の流れや因果関係、主客関係の変化を時間軸上で明示できる。これらの設計は抽象化・分解・パターン認識・アルゴリズムといった CT の主要概念に即しており、制作とデバッグの反復を通じた同調的学習の促進が期待される。

キーワード： ピクトグラム、コンピューターショナルシンキング、情報デザイン、プログラミング、ブロック型

Hyper Pictoch: Development of a Block-Based Pictogram Authoring Environment to Foster Computational Thinking

KAZUNARI ITO^{1,a)}

Abstract: We propose "Hyper Pictoch," a block-based pictogram authoring environment designed to foster Computational Thinking (CT). This application defines human pictograms and objects as independent entities and features the ability to represent multi-agent interactions and complex scenes through concurrency, message passing, conditional branching, and procedures. Furthermore, its support for free color assignment enables applications in color learning. The pictograms targeted in this study possess high legibility due to simplified forms and high contrast, language-independent universality, and consistency based on standardization. In addition to these properties, this application enables the representation of abstract actions and internal states through animation generation functions. This makes the flow of actions, causal relationships, and changes in subject-object relationships explicit along the time axis. These design choices align with the core concepts of CT—abstraction, decomposition, pattern recognition, and algorithms—and are expected to facilitate syntonetic learning through iterative creation and debugging.

Keywords: pictogram, computational thinking, information design, programming, block-type

1. はじめに

コンピューターショナルシンキング (以下、CT) は、コンピュータ科学者だけでなく、すべての人にとって、「読み書きそろばん」と同等の基本的なスキルである[1]。CT は広義には、コンピュータ (またはその他の情報処理エージェント) が実行可能な形式で問題とその解決策を定式化するための、思考プロセスや戦略を指す[2]。Wing らによる

広く引用されている定義によれば、「CT とは、問題とその解決策を定式化し、その解決策を情報処理エージェントが効果的に遂行できる形式で表現することに関わる思考プロセスである」とされる[2]。CT は、必ずしもコンピュータそのものやプログラマになることではなく、コンピュータ科学者のように思考することを意味する。重要な点は、CT はコンピュータプログラミングや IT スキルと同義ではないということである。Wing が強調するように、コンピュータ科学者のように考えることには「コーディングではなく概念化」が含まれる。それは「コンピュータをプログラムできる以上のことを意味し、複数の抽象化レベルで思考す

¹ 青山学院大学

Aoyama Gakuin University, Kanagawa 252-5258, Japan

^{a)} kaz@si.aoyama.ac.jp

ることが求められる」のである[3]。言い換えれば、CTは特定の技術やプログラミング言語そのものではなく、問題解決に対する概念やアプローチに関するものである。

概念としてのCTは、Wingの当初の提唱以来、発展を続けている。Wingの2006年の論文はCTに対する現代的な関心を喚起したが[1]、文献においてCTの普遍的に合意された単一の定義は未だ存在しない。研究者や組織によってCTは様々に定義されており、それぞれが重視する点（コーディング、問題解決、あるいは認知的スキルなど）を反映している[4]。研究者や教育者が概念を精緻化するにつれ、CTの定義は継続的に進化している[5]。しかし、多くの定義に共通しているのは、CTとは問題を体系的に考案し、コンピュータやアルゴリズムのプロセスによって実行可能な形式で解決策を設計することを含む、という点である。

定義には相違があるものの、CTを構成する中核的な要素や手法については広範な合意が存在する。CTはしばしば、分解、パターン認識、抽象化、アルゴリズム的思考といういくつかの主要なスキルや柱に細分化される[6]。これら4つの柱は、多くのフレームワークやカリキュラムにおいて、CTの基礎的要素として頻繁に引用されている[7]。

(1) 抽象化 (Abstraction) : 無関係な詳細を無視し、重要な情報に注目すること。本質的な特徴を捉え、不要な詳細を省くことで、あるクラスの問題に適用できる一般的な解決策やモデルを定式化することを意味する。これにはしばしば、問題の基礎となる一般的なルールや概念を定義することが含まれる。

(2) 分解 (Decomposition) : 複雑な問題を、より小さく扱いやすい部分や小問題に分割すること。これにより、大きな問題を理解しやすくし、部分ごとに解決することを容易にする。

(3) パターン認識 (Pattern recognition) : データや問題の中にある、繰り返し現れるパターン、傾向、類似性を特定すること。パターンを認識することで、既知の解決策を類似のケースに適用し、結果を予測したり問題を単純化したりすることができる。

(4) アルゴリズム (Algorithms) : 問題を解決するための段階的な指示や手順 (アルゴリズム) を開発すること。これには、コンピュータや人間が解決策を体系的に実行できるようなプロセスを設計することが含まれる。

しかしながら、初学者においてCTを効果的に育成する方法は依然として課題である。従来のプログラミング教育では、興味の欠如、構文規則の理解の難しさ、そして挫折による学習意欲の低下といった問題に直面することが多い。これらの問題に対処するため、本論文では教育的アプローチとしてピクトグラム制作に着目する。ピクトグラムをデザインする過程では、ユーザが概念を視覚的なシンボルへと単純化する際に、抽象化や分解といったCTの主要な構

成要素が自然に関与する。そこで我々は、ピクトグラムのデザインとプログラミングを統合したブロック型環境「ハイパーピクトッチ (Hyper Pictoch)」を提案する。本アプリケーションは、アニメーションやマルチエージェント間の相互作用を視覚的に制御可能にすることで、並行処理などの複雑な概念に対する障壁を下げ、楽しみながらCTを育成することを目指している。本論文の構成は以下の通りである。第2章では開発の背景と動機について述べる。第3章でアプリケーションの詳細について説明する。第4章でアプリケーションに関する考察を行い、第5章で本論文を総括する。

2. 動機と目的

ピクトグラムはコンテンツ表現の抽象度の高さから、それを見た人物が自分自身や本人に関わる人物事物など想起させる効果があると言われている。有名な「非常口」ピクトグラムのデザインの策定に関わった太田は”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[8]、実際に避難中の人が、如何に出口へ向かって走る人型ピクトグラムを見たときに、自身に投射し、同一視するかにデザインの労力が払われた。

この特性は情報教育の諸分野にも活用できるのではないかと考え、筆者が開発してきたのがピクトグラミング及び派生アプリケーション（以下これらを総称してピクトグラミングシリーズと呼称する）である。プログラムの記述に教育利用を念頭にしたテキストベースの独自記法を用いたピクトグラミング、Python言語で記述できるPicthon（ピクソン）、JavaScript言語で記述できるJavaScpict（ジャバスクピクト）、視覚的なブロックを使って記述可能なPictoch（ピクトッチ）などがある[9]。特にブロック型の環境であるピクトッチは、あらかじめ命令の一覧がブロックで用意されているので、事前学習なしでも扱いやすい。

図1は、ピクトグラム制作の場合におけるこれら主要概念のイメージを示す。

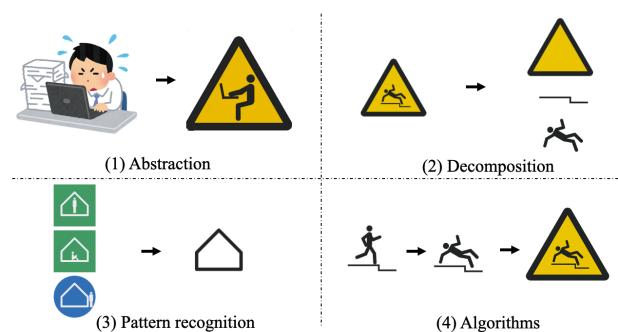


図1 CT概念とピクトグラム制作プロセスとの対応

Fig. 1. Correspondence between CT concepts and the pictogram creation process

(1) 抽象化 (Abstraction) : ピクトグラムは主にデザインやコミュニケーションの分野で研究されてきた。ピクトグラムの制作は、ある概念から高い抽象度の図記号・絵記号を作り出すことである。

(2) 分解 (Decomposition) : ピクトグラムは通常、複数のオブジェクトから構成される。オブジェクトをどう分離し、どれを記述すべきか、どれを独立して設計すべきかを検討する。このプロセスは分解に相当する。

(3) パターン認識 (Pattern Recognition) : 一般化・モジュール化ともいうことができる。複数のピクトグラムから共通の要素や共通の動作のステップを抽出して部品化する。この設計原則は、ピクトグラムの制作において推奨される。

(4) アルゴリズム (Algorithms) : ピクトグラムの構造や動きの連鎖を手続き的記述で表現する行為は、それはアルゴリズム的思考に基づいていると言える。

この指針に基づいて、従来のピクトタッチをベースに Computational Thinking の主要概念と取り入れ、機能拡張したアプリケーションを開発する。

また、ピクトグラムの理解度向上を目的として、アニメーション化に着目した研究も報告されている。例えば、抽象度の高い複雑な動詞や、思考・感覚といった内的事象を捉えるうえでは静止画表現が不十分であるという知見 [10] に基づき、大野らは動画ピクトグラムのデザイン指針を提案している [11]。また、コミュニケーション支援用絵記号をアニメーション化することでピクトグラムに対する理解度が向上し、適切な行動を取る被験者が増加したことを報告する研究もある [12]。一方で、以前のピクトタッチでは、オブジェクト指向プログラミングの概念は導入せず、同時にアニメーション可能な人型ピクトグラムは1体だけとなっており、アニメーションピクトグラムの制作に制限が生じていた。そこで、複数のピクトグラムや物体を独立に動作可能とする。このような表現力の拡張は、場面の複合性や主体間の関係性をよりの確に記述できる点で意義がある。

また以前のピクトタッチでは、ISO 3864-1 に準拠したピクトグラム作成を前提とする設計のため、色は規定色に自動的に設定されていた。これは短時間の実習でも標準仕様に準拠したピクトグラムを作成するという目的に沿ったものである。一方、物の識別やコミュニケーションには色が役立つこと [13] から、ピクトグラムの色彩化に関する研究も多い。例えば、AAC (Augmentative and Alternative Communication) と呼ばれるピクトグラムを用いたコミュニケーションにおいて、モノクロのピクトグラムよりもカラーピクトグラムの方が対象物を明確に表現できることを示した研究 [14] や、カラーピクトグラムによって着目すべき情報を引き立て、了解性・認識性・記憶性に優れるデザインが可能であることを示唆した研究 [15]、ピクトグラ

ムのカラーリングに用いる色が理解度に影響するため、カラーピクトグラムの設計には文化的背景や教育的背景への配慮が必要であることを示唆した研究 [16] がある。色彩や色相の学習へ応用することも考えられる。

3. ハイパーピクトタッチ

3.1 概要・画面構成

ハイパーピクトタッチは、ハイパーピクトグラミング [17] をベースに拡張された Web アプリケーションであり、ブラウザ上で動作する。図 2 にスクリーンショットを示す

画面は主に 4 つのコンポーネントで構成される。画面左上部は、プログラムの実行結果を表示する”ピクトグラム表示領域”、右上部はプログラムを入力する”プログラム入力領域”である。”プログラム入力領域”の左側にはブロックのカテゴリが列挙されている、カテゴリ名をクリックするとそのカテゴリに属するブロックの一覧が表示され、それらブロックをドラッグすることで、プログラム入力領域にブロック群を配置できる。左下部は、プログラムの実行を制御するコントロールボタンが配置されている。コントロールボタンは 5 つのボタンから構成されており、左から「再生」「高速再生」「一旦停止」「停止位置から再生」「プログラムの全削除」からなる。またその右側には、ファンクションボタン群が配置されている。左から、「ピクトグラム画像のダウンロード」、「ピクトグラム動画のダウンロード」、「プログラムのダウンロード」、「プログラムのアップロード」、「URL 形式によるプログラム情報のコピー」、「グリッド表示の切り替え」である。右下部は、オブジェクト操作ボタンが配置されている。左から、「構成素となる人型ピクトグラムの登録」、「構成素となる物体の登録」、「オブジェクトの名称の変更」、「オブジェクトの複製」、「オブジェクトの削除」、「選択されているオブジェクトのダウンロード」、「ファイルアップロードによるオブジェクト群の追加」である。オブジェクト単位でインポート、エクスポートが可能になるため、共通要素の部品化や高い再利用性がサポートされる。

ファンクションボタン群のさらに右には、「言語選択」のプルダウンがあり、それにより表示する言語を変更できる。現状では、英語、日本語、ひらがなをサポートしている。さらにその下に登録されているオブジェクトの一覧が表示される「オブジェクト管理領域」が配置されている。

URL にパラメタをいくつか指定できる。lang 属性を変更することによりラベルやブロックの表示言語を変更することができる。また code 属性にオブジェクトの情報を文字列化したデータを設定することで、URL でアクセスするだけで、指定されたコンテンツを初期状態で表示し、実行や編集の再開を行うことができる。

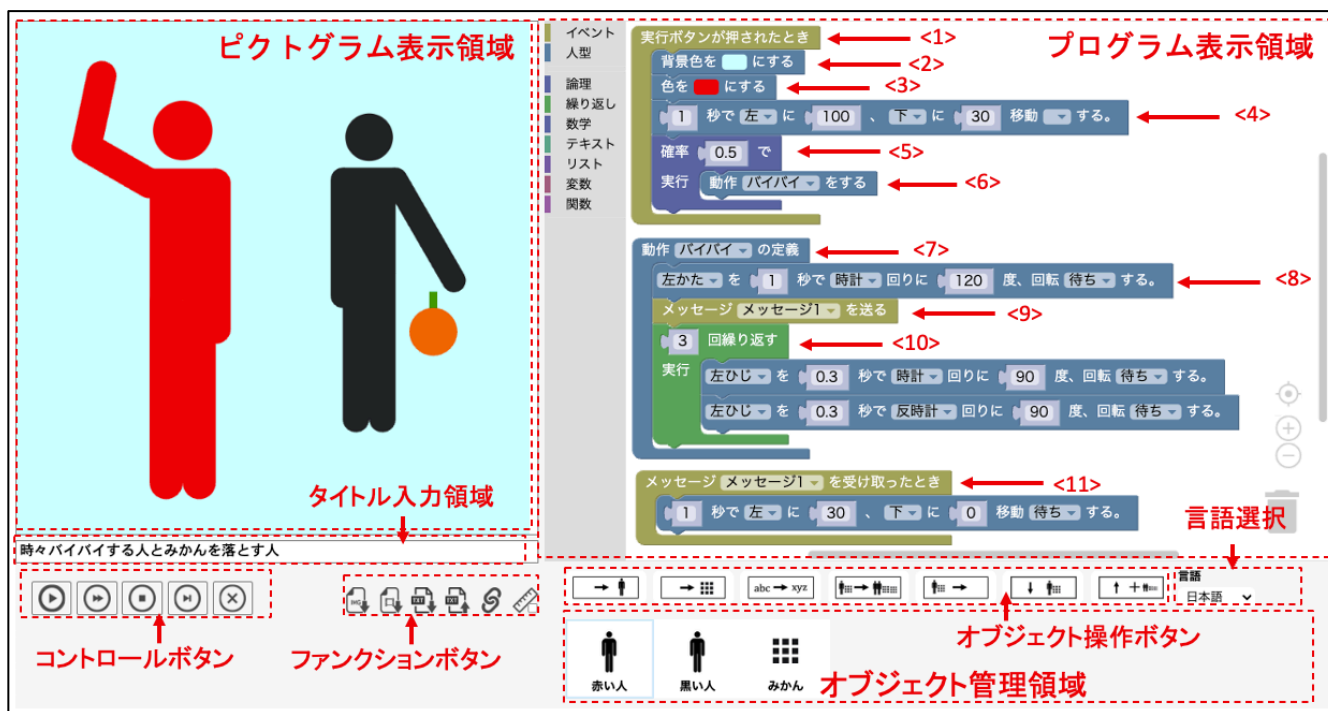


図2 ハイパーピクトタッチのスクリーンショット

Fig.2. Screenshot of Hyper Pictoch

3.2 オブジェクトの基本構成

ハイパーピクトタッチでは、ピクトグラムの構成素をオブジェクトと呼ぶ。オブジェクトは(1)人型ピクトグラム (2)物体の2種類から構成される。(1)人型ピクトグラムのオブジェクトには、1. 人型ピクトグラムの変形、移動などのアニメーションの命令、2. 禁止、注意等のピクトグラム固有のマーク（セーフティマーク）を描画する命令、背景色の設定の命令が用意される。(2)物体のオブジェクトには、1. 図形定義およびアニメーションの命令が用意される。

そのほかには、変数定義、繰り返し、条件分岐や関数定義、メッセージパッシングなどのブロック群が用意されている。

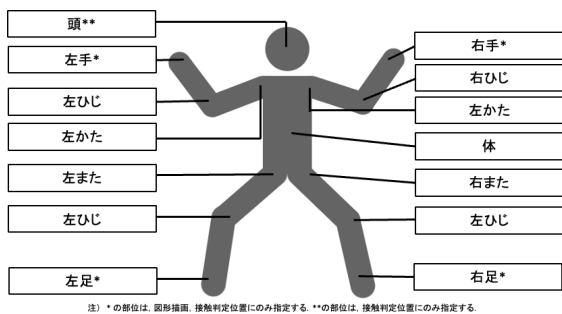
図2では、2体の人型ピクトグラム(ラベル名：赤い人、黒い人)と物体1つ(ラベル名：みかん)が表示されている。オブジェクト管理領域にあるオブジェクト群配置された矩形領域のいずれかをクリックすると、オブジェクトに対応するプログラムがプログラム表示領域に表示される。図2のプログラム表示領域にはそのうちの人型ピクトグラム1体(ラベル名：赤い人)に対応するプログラムが表示されている。このプログラムを例に説明する。

図2の<1>で示された囲いブロックは、実行ボタンが押されたときに実行されるブロック群を指定する。<2>は背景色を、<3>のブロックは人型ピクトグラム自身の色を設定するブロックである。ハイパーピクトタッチでは、人型ピクトグラムや図形、背景に対して自由な配色を許容する仕様となっている。<4>は人型ピクトグラムを並行移動するブロックである。<5>は、このブロックに囲まれた内側の

命令を一定の確率で実行するブロックである。分岐命令に相当する。標準的な条件分岐ブロックも使用できるが、条件式を定義しブロックで作るのは容易ではない。そのため、分岐処理を理解してもらうために、このようなブロックを用意している。

ハイパーピクトタッチでは、共通のパターンを抽出し、部品として再利用化することに価値を見出している。<6>は定義された動作を実行するためのブロックで、「バイバイ」と定義された動作を実際に実行する。動作の定義はいずれかの人型ピクトグラムのオブジェクトで定義しておけば、全ての人型ピクトグラムのオブジェクトでその定義を利用できるようになっている。つまり再利用可能な形式になっている。<7>はその一連の命令の塊を動作として定義するための囲みブロックで、ここでは「バイバイ」と命名し、バイバイの動作をするための一連のブロックを定義している。

<8>は、人型ピクトグラムの身体部位に対する回転の命令である。関節点、回転方向・角度(反時計周りが正)、回転に要する秒数を設定するブロックである。人型ピクトグラムは、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つの部品から構成され、それらが連結されている(図3参照)。この人型ピクトグラムに対する、体の部位を中心とする回転操作と座標系上の平行移動の2種類の命令があり、さらに逐次実行と並行実行を組み合わせることで、人型ピクトグラムの多様な動作を可能としている。



注) *の部位は、図形描画、接触判定位置にのみ指定する。**の部位は、接触判定位置にのみ指定する。

図3 人型ピクトグラムの部位指定

Fig.3. Labels for the parts of the human pictogram

次のブロックを同時に実行するか、実行が終了するまで次のブロックを実行しないか、ブロック中にある「待ち」の有無で選択できる。図2のプログラムでは、<4>と<5>のブロックは並行実行され、<8>のブロックは、<6>のブロックに対応する動作が終了したのち実行される。

<9>のブロックは、メッセージ送信ブロックである。指定したメッセージ名で自分自身を含む全てのオブジェクトに対してメッセージを送出する。<10>のブロックは、繰り返しブロックである。このブロックに囲まれた内側の命令を指定回数繰り返す。<11>はメッセージを受け取ったときに実行されるブロック群を定義するためのブロックである。これにより複数のオブジェクト間で同期をとることができる。

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のガイドラインが ISO 3864 で策定されている。ハイパーピクトタッチもこれに準拠し、禁止、注意、指示、安全(3通り)のピクトグラムを作るために単一のブロックでこれら6種のセーフティサインを設定できる。一覧を図4の右に示す。コミュニケーション用途で用いるピクトグラムのデザイン指針を定めた JIS T 0103 および ISO 19026 では、黒背景に白字のピクトグラムが推奨されている。そのため、ハイパーピクトタッチでは反転(図4の左下)も指定できる。



図4 セーフティサインの例

Fig.4. An example of safety sign

物体のオブジェクトは基本図形の集合で描く。単純図形の作画は、ピクトグラム作成において不可欠な工程である。また、プログラミングの入門講座で、図形描画がテーマの

実践も多い[18]。図1で表示されている物体(物体名:みかん)に対応するプログラムを図5に示す。



図5 物体のオブジェクトのプログラム例

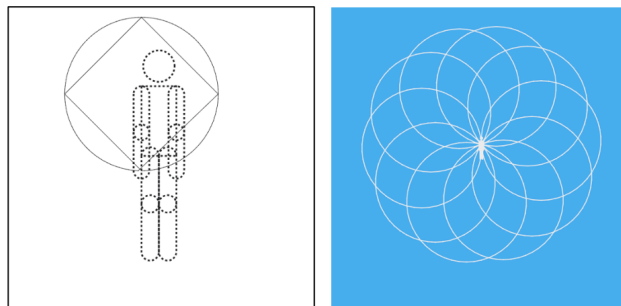
Fig.5. Example of object's program

図5の<1>のブロックは物体の定義をするためのブロックである。描画する図形群をこのコ型ブロックの内部に配置する。<2>、<4>のブロックは描画色を指定するブロックである。<3>、<5>のブロックはそれぞれ、線分と楕円を描く命令である。<6>はメッセージを受け取ったときに実行されるブロック群を定義するためのブロックである。

3.3 図形描画機能

単純図形の作画は、ピクトグラム作成において不可欠な工程である。図5で示したように座標指定した基本図形の組み合わせで物体の描画ができることは示した。ハイパーピクトタッチでは、従来のピクトタッチと同様に人型ピクトグラムのオブジェクトの動作の履歴に基づく、2種類の描画方式がある。図6の「(a)身体動作」は、体の部位を指定して、その部位の動作の履歴を線画で表現する。この例では、左手の動きの履歴で円と正方形を描画している。

図6の「(b)泳ぎ」は、画面一面を水面と見なし、鉛直上方向から見たときの人型ピクトグラム自体の移動(泳ぎ)の履歴で描画する方式で、主に教育用プログラミング言語で用いられるタートルグラフィックスに時間の概念を組み入れ拡張したものである。いずれも、図2のプログラムで説明した逐次実行と並行実行の概念を組み合わせることで、少ないブロック数で様々な図形が描画できる工夫がされている。



(a) 身体動作

(b) 泳ぎ

図6 図形描画の例

Fig.6 An example of diagram drawing

図7は図6の左側の図形を描画するプログラムである。<1>のブロックは人型ピクトグラムを透明にするブロックで、線画が人型ピクトグラムに隠れて見えなくなるのを防いでいる。<2>は人型ピクトグラムの体の大きさの倍率を変更するブロックである。<3>は所持するペンの太さを変更するブロックである。<4>のブロックは、人型ピクトグラムの左手の部分にペンを持つ処理をする。<5>のブロックは左腕を1回転(360°)するので、円が描かれる。<6>は90°左腕を回転する命令である。命令の時間が0より大の場合、その軌跡自体が描かれるが、時間が0の場合、動作の始点と終点を結ぶ直線が描かれる仕様になっている。そのため<6>のブロックが4回繰り返されることにより正四角形が描かれる。この仕様により、曲線と直線を組み合わせた図形が少ないブロックで作成できる。



図7 身体動作による描画プログラム例

Fig.7. Example of program by human motion

図8は図6の右側の図形を描画するプログラムである。<1>は10秒間直進する命令のブロックである。<2>は同時に1秒かけて360°左回りに回転するブロックである。つまり1秒かけて円が描画される。<3>は円が描画されると同時に0秒で進行方向を36°左回りに回転する。これを10回繰り返すので、円が10個方向をずらして描画される。

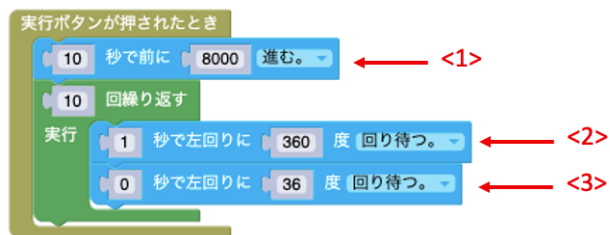


図8 泳ぎによる描画プログラム例

Fig.8. Example of program by swimming.

従来のピクトタッチでもこの2種類の描画手法がサポートしていたが、ハイパーピクトタッチでは、複数の人型ピクトグラムを登場させて同時並行してこの方式で描画できるため、従来に比べて多様な図形がより描画しやすくなっている。

以上より従来版ピクトタッチとハイパーピクトタッチの機能比較を表1にまとめる。

表1 従来版ピクトタッチとハイパーピクトタッチの機能比較

比較項目	ピクトタッチ	ハイパーピクトタッチ
操作対象	同時に動作可能な人型は1体のみ	複数の人型や物体を独立して操作可能
オブジェクト定義	単一人型ピクトグラムに限定	人型および「物体」を独立した実体として定義
プログラムの実行制御	単一主体の逐次動作・並行動作	加えてメッセージパッシングによる相互作用
色の設定	規格 (ISO 3864-1) に準拠した自動設定	自由な配色が可能 (色彩学習への対応)
図形描画の拡張性	1体による描画	複数主体による同時並行での図形描画

4. 考察

本章では、ハイパーピクトタッチの教育的意義について、コンピュータショナル・シンキング (CT) 教育の観点から議論する。特に、身体的抽象化、表現手段としてのコーディング、および制約を通じた認知的発達の3点に着目する。

4.1 身体的抽象化と同調学習

従来のプログラミング教育では、抽象的なコードと具体的な実行結果との間のギャップを埋めることに苦労することが多い。Hyper Pictoch は、操作の主対象として「人体」を利用することでこの課題に対処している。この設計は、Papert が提唱した「身体的同調性 (Body Syntoncity)」の概念を活用したものである[19]。彼は、学習者の身体感覚と共鳴するとき、学習は最も効果的になると論じている。Truchetet と Lund もまた、CT の概念を物理的現実根付かせるためには、このような身体的な相互作用が不可欠であると主張している[20]。本アプリケーションにおいて、ユーザが「手を振る」動作をプログラムする場合、自身の関節の動き (例：肘の回転) をメンタルシミュレーションし、それをコードに翻訳しなければならない。このプロセスにより、学習者は対象との身体的な同一視を通じて「分解 (Decomposition)」や「順序 (Sequencing)」の概念を内面化することができ、抽象的なアルゴリズムの理解に伴う認知的負荷を効果的に軽減できる。

4.2 計算論的表現と視覚リテラシ

Kafai と Burke は、プログラミングを単なる技術的スキルとしてではなく、表現とコミュニケーションのための新たな「リテラシー」として捉えるべきだと強調している[21]。これは、コンピューティングをアイデア表現の素材とする diSessa の「コンピュータショナル・リテラシー」の見解とも共鳴するものである[22]。Hyper Pictoch は、コードを「情報デザイン」のための道具として扱うことで、この視点に整合している。汎用的な環境とは異なり、本システムは「ピクトグラム」、すなわち意味を効率的に伝えるための記号の作成に焦点を当てている。これには、記号論的な意味での「抽象化 (Abstraction)」の訓練が必要となる。つ

まり学習者はメッセージを普遍的に伝えるために、どの詳細を省略し、どの特徴を誇張するかを決定しなければならない。プログラミングとビジュアルデザインを統合することで、Hyper Pictoch はコードを読み書きする能力と、視覚情報を読み書きする能力という「二重のリテラシー」を育成する。

4.3 制約を通じた認知的発達

創造性はしばしば無制限の自由と結び付けられるが、構築主義的アプローチにおいては、支持的な枠組みの中でアーティファクト（人工物）の構築に取り組む際に有意義な学習が生じると示唆されている[23]。Román-González らもまた、CT は制約下での問題解決能力といった認知能力と密接に関連していると指摘している[24]。Hyper Pictoch は、ピクトグラムデザインに基づいた具体的な制約を課している。これらの制約は創造性を妨げるものではなく、むしろ制限された範囲内で表現の目標を達成するために、学習者に対して「アルゴリズム的思考」に取り組むことを強制するものである。例えば、標準的な幾何学図形のみを用いて複雑なアニメーションを作成するには、創造的な工夫と効率的なアルゴリズム設計が必要となる。本環境は、制約そのものが高度な計算戦略の発達を促すような、足場かけされた体験を提供する。

5. おわりに

本論文では、アニメーションピクトグラム制作や人型ピクトグラムの特徴の活かした図形描画を通じてコンピュータシミュレーション・シンキングを育成する、新しいブロックベースのピクトグラム制作環境「ハイパーピクトタッチ」を提案した。今後の課題として、長期的な実証研究を行い、これらのスキルの転移可能性を評価する予定である。

参考文献

[1] J. M. Wing, "Computational thinking," *Communications of the ACM*, vol. 49, no. 3, pp. 33–35, 2006.

[2] J. M. Wing, "Computational thinking: What and why?," *The Link*, vol. 6, no. 1, pp. 20–23, 2011.

[3] V. J. Shute, C. Sun, and J. M. Asbell-Clarke, "Demystifying computational thinking," *Educational Research Review*, vol. 22, pp. 142–158, 2017.

[4] S. Grover and R. Pea, "Computational Thinking: A Competency Whose Time Has Come," in *Computer Science Education: Perspectives on Teaching and Learning in School*, 2nd ed. London: Bloomsbury, 2018, pp. 20–38.

[5] R. A. Yuana, S. Sajidan, Wiranto, and M. Nizam, "Strategies for integrating computational thinking and scientific approaches in programming education: a systematic literature review," *Discover Education*, vol. 4, art. 371, 2025.

[6] L. L. Ung, J. Labadin, and F. S. Mohamad, "Computational thinking for teachers: Development of a localised e-learning system," *Computers & Education*, vol. 167, art. 104185, 2021.

[7] Q. Li, "Computational thinking and teacher education: An expert interview study," *Human Behavior and Emerging Technologies*, vol. 2, no. 3, pp. 200–210, 2020.

[8] 太田幸夫. ユニバーサル・コミュニケーションデザインの認識と実践 <https://forum8.co.jp/topic/universal107.htm> (参照 2025-11-02)

[9] 伊藤一成: 複数のプログラミング言語で記述可能なピクトグラムコンテンツ作成環境の提案と実装, *情報処理学会論文誌 TCE*, Vol.7, No.3, pp. 1-11 (2021)

[10] 大野森太郎, 原田利宣, 宗森純: “動詞”の情報量分析に基づくピクトグラムデザイン支援システム, *デザイン学研究*, 58(2), pp.55-64 (2011)

[11] 大野森太郎, 原田利宣, 宗森純: 動画表現を用いたピクトグラムにおけるデザイン指針の提案, *デザイン学研究*, 60(1), pp. 95-102 (2013)

[12] Fujisawa, K., Inoue, T., Yamana, Y., Hayashi, H.: The effect of animation on learning action symbols by individuals with intellectual disabilities, *Augmentative and Alternative Communication*, 27(1), pp.53-60 (2011)

[13] Schloss, K. B., Lessard, L., Walmsley, C. S., Foley, K.: Color inference in visual communication: the meaning of colors in recycling, *Cognitive research: principles and implications*, 3, pp.1-17 (2018)

[14] 稲田勤, 野々篤志, 本田梨佐, 吉村知佐子, 石川裕治: シンボルコミュニケーションにおける受信者側のイメージに関する研究 モノクロシンボルとカラーシンボルのイメージ測定, *高知リハビリテーション学院紀要*, (9), pp.49-53 (2008)

[15] 下江優太, 小河幸次, 濱本和彦, 野須潔: ピクトグラムの了解性, 認識性, 記憶性の色彩デザイン依存性の検討, In *IEICE Conferences Archives*, The Institute of Electronics, Information and Communication Engineers (2008)

[16] Hiranchiracheep, S., Yamazaki, A. K., Foypikul, W.: A preliminary surveying of the meaning of colored pictogram instructions for emergency settings in manufacturing. *Procedia computer science*, (96), pp.1528-1534 (2016)

[17] 木下倭, 伊藤一成: ハイパーピクトグラミング - ピクトグラムの作成を題材にした Web ページ作成技法の学習環境 -, *情報処理学会 情報教育シンポジウム SSS2025*, pp.1-8 (2025)

[18] 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*, vol. 3, no. 1, pp.26-35 (2017)

[19] S. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*. New York, NY, USA: Basic Books, 1980.

[20] F. Truchetet and H. H. Lund, "Embodied programming and computational thinking," in *Proceedings of the 10th International Conference on Education and New Learning Technologies (EDULEARN18)*, Palma, Spain, 2018, pp. 9362–9371.

[21] Y. B. Kafai and Q. Burke, "Computer programming as a literacy: A historical review," *Educational Researcher*, vol. 42, no. 9, pp. 603–633, 2013.

[22] A. A. diSessa, *Changing Minds: Computers, Learning, and Literacy*. Cambridge, MA, USA: MIT Press, 2000.

[23] S. Y. Lye and J. H. L. Koh, "Review on teaching and learning of computational thinking through programming: What is next for K-12?," *Computers in Human Behavior*, vol. 41, pp. 51–61, 2014.

[24] M. Román-González, J. C. Pérez-González, and C. Jiménez-Fernández, "Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test," *Computers in Human Behavior*, vol. 72, pp. 678–691, 2017.