

タンジブルなブロックを組み立てることにより AR 空間で実行可能な教育用プログラミングツール

服部 圭介 井上 馨太 志賀野 桐生 平井 辰典*

概要. 2020 年度から実施される新学習指導要綱において小学校でプログラミング教育が必修化されることから、教育に適したプログラミングツールが望まれている。初学者向けのプログラミングツールには、ビジュアルプログラミングや、ロボットプログラミングキットなどが存在する。しかし、ビジュアルプログラミングを教える際に使う PC を十分に用意できていない学校が存在したり、プログラミング教育用ロボットではプログラムの実行結果に物理的な制限が生じてしまう。本研究では教育用ロボットのように実物でのプログラミングを可能としながら、ビジュアルプログラミングのように実行結果に制限なく自由なプログラミングができるツールを AR を用いて実現する。具体的には、命令が書かれたタンジブルなブロックでプログラミングし、AR 技術でスマートフォンの画面を通して実行結果を現実にも重ねさせて見ることが出来るツールを提案する。

1 はじめに

2020 年度から小学校で実施される新学習指導要領 [7] において、プログラミング教育が必修化されることから教育に適したプログラミング教育ツールの開発が望まれている。児童に対して、一般的なテキストプログラミングを学習させるのはハードルが高い。そのため、視覚的なブロックを組み合わせることで簡単にプログラミングをすることができるビジュアルプログラミングが注目されている。ビジュアルプログラミングの多くは、キーボードによるタイプ入力をほとんど必要とせず、マウス操作だけで複雑なプログラムの作成が可能となっている。MIT のメディアラボで開発された、代表的なビジュアルプログラミングに Scratch [3] がある。Scratch は画面上の命令の描かれたブロックをユーザがマウスを使い組み合わせることで簡単に複雑なプログラムを作ることができる。Scratch は世界中のプログラミングのワークショップ等で児童に対して使用されるなど、プログラミング入門教材として適している。一方で、Scratch を始めとした多くのプログラミング言語/開発環境は PC を使わなければプログラミングができないということが課題だと我々は考えている。現在プログラミング教育に使われている多くのビジュアルプログラミングは通常一人一台の PC を操作してプログラミングを行う。総務省の 2019 年の通信利用動向調査 [5] によると、スマートフォンによるインターネットの利用割合が、PC での利用割合を上回った。全年齢の平均的な利用割合はスマートフォンが 59.5%、PC が 48.2% となっている。特に 20~29 歳ではスマートフォンでの利用が 88.7%、13~

19 歳では 76.6% と若年層では顕著に表れている。このデータから、若年層には特に、情報機器の利用割合は PC からスマートフォンへと変化しているため、プログラミングが PC だけに留まらずスマートフォンでもできるようにしなければ、スマートフォンしか使用できない子供と、PC とスマートフォンを使える子供でデジタルディバイドが生まれてしまう。

初学者がプログラミングを学ぶ上で、その言語を用いたプログラミングが直感的であるかどうかはとても大事な要素である。我々は PC をプログラミング初学者に使わせることが、直感性の低下をさせる要因の一つだと考えている。児童がプログラミングを学習する以前に教員は児童に PC 操作から説明しなくてはならないため、プログラミングの授業へ到達するまでに時間がかかってしまう。そのため PC を使用しないプログラミングツールを使うという選択肢も考えられる。

ビジュアルプログラミングは前提として PC を使い、画面の中でプログラミングをすることとなっている。児童は、画面の中の世界に対して命令を与えているという認識を持たないとプログラミングを学ぶことができない。これは児童がそれまでにどのような情報機器にどれだけ触れてきたのかによって理解度に差が生じることが懸念される。小学生全員に対する必修化であれば、そのような機器に生まれて一度も触れてこなかった児童も想定が必要である。

PC などの電子機器を使用するプログラミングツールとは対照的に、実際に手に取ることができる物体を使ってプログラミングを行うキットのことを物理プログラミングと呼ぶことにする。物理プログラミングは、命令に対応する物体を組み立てることで、ロボット等の対象物を操作することができる。実世界で作成したプログラムが実世界で直接対応さ

れ実行されるため、プログラミング初学者でも直感的にプログラミングが可能となっている。しかし、物理プログラミングの欠点として、物理的な物体を操作することによる物理的な制限がかかる点がある。例えば、児童はロボットを空に飛ばすような命令を思いついても実行させられない。

そこで、本稿ではタンジブルな物体を使った直感的なプログラミングと、制限のない自由な実行結果と、複数の児童が同時にプログラミングできる機能を合わせ持ったプログラミングツールを提案する。具体的には、物理プログラミングのように現実世界の物体を使ってプログラミングを行い、その結果をAR技術によって現実世界に重畳させる形で提示する。それにより、現実世界におけるプログラミングをベースに、拡張現実空間におけるCGアニメーションという形で自由なプログラミングを実現する。これにより、PCの画面の中に制限されていたビジュアルプログラミングと、物理法則やロボットの駆動域に制限されていた物理プログラミングの両者の制限を払拭したようなプログラミングツールの実現を目指す。

2 関連研究

プログラミング言語の種類は年々増加しており、プログラミングの学習に重きをおいた言語やツール、研究事例が多くある。学習に重きを置いた言語やツールの多くは、視覚的にプログラミングをするビジュアルプログラミングや、タンジブルにプログラミングをする物理プログラミング等をベースに作られている。

本研究では上記の二種類のプログラミングツールの利点を合わせ持つツールを開発するため、以下の要素を合わせ持ったツールの実現を目指す。

1. ビジュアルプログラミングを始めとしたPCを用いる従来のプログラミング言語のような実行結果の柔軟性（自由な実行結果）
2. 物理プログラミングにおけるプログラムと実行結果との明確な対応関係による直感性

上記の二つの要素を同時に実現するために、現実世界に直結していながらも、物理的な制約を受けずに現実世界を拡張可能なAR技術を用いることが有効であると考えられる。

八城らが提案した物質プログラミング[6]は、Scratchのソフトウェア上のブロックを実体のあるブロックとしてプログラムを制作できるツールである。それらの実態のブロックを組み合わせることでPC画面内の車、もしくは車型ロボットの動きをプログラミングする。このツールでは要素2のように、タンジブルにプログラミングすることによる物理プログラミングのような直感性はあるが、画面内の世界を理解しなければならない上に、要素1のような物理的

な制限を超えた実行結果をプログラミングすることができない。

坂本らは物理的な形状を持ったパーツを直接手で組み合わせ、それを画像認識によって読み込むことでプログラミングできるMR Code Weaver[4]を提案した。入力されたプログラムの実行結果は、三脚によってテーブルの周囲に固定されたプロジェクタによってテーブル上に投影される。全工程がテーブル上で完結しているため、パーツによるプログラミング操作と映像による実行結果を同時に確認可能としている。このツールは、物理プログラミングのように直感的にプログラミングをすることはでき、要素2に挙げた対応関係による直感性はある。しかし、プロジェクタを通して実行結果が投影されるため画面の中で行うプログラミングとの間に大きな違いがない。

Magenatらは画面内のブロックを操作することでプログラミングした、ロボットにタブレットPCをかざすことでプログラムの内容を視覚的にフィードバックするツールを研究している[2]。この研究は、プログラムの実行結果にAR技術を使うことで、プログラムのフィードバックをタブレット画面内に表示させ、ロボットの動きとプログラムの動きを同時に確認することができる。フィードバックを表示させることにより、プログラムの改善性を上げる研究となっている。このツールは、物理的なロボットを使うため、プログラムに制限が掛かってしまう他、画面内でプログラミングをするため要素1のような自由度はあるが、要素2のようなプログラミング行為自体に直感性はない。

近年多くのARインタフェースが開発されているが、それらのプロトタイプを作成するには、技術や知識が必要とされることが多い。Kellyらは、ARインタフェースの作成過程を視覚的に簡略化し、知識がなくても自由にプロトタイプを児童でも作ることができるツールのARcadia[1]を研究している。マーカの書かれたカードと、画面内のビジュアルプログラミングで使われるブロックを操作し、それぞれのマーカに対する命令を決め、ARインタフェースをプログラミングするツールとなっている。Kellyらは、このツールを用いて、児童らにARインタフェースの作成をさせるワークショップを開いている。このツールは、要素2とは違い画面内のブロックを組み合わせることで、ARインタフェースを作成するため、我々の提案ツールとは違い、プログラミングの行為自体には直感性が生まれない。

これまで挙げた既存研究は、タンジブルにプログラミングすることで、物理プログラミングにおける直感性を高めるツールや、ビジュアルプログラミングのように画面内でプログラミングをし、現実に行結果が見せることで直感性を高めるツールである。本研究は、タンジブルなブロックで物理プログラミ

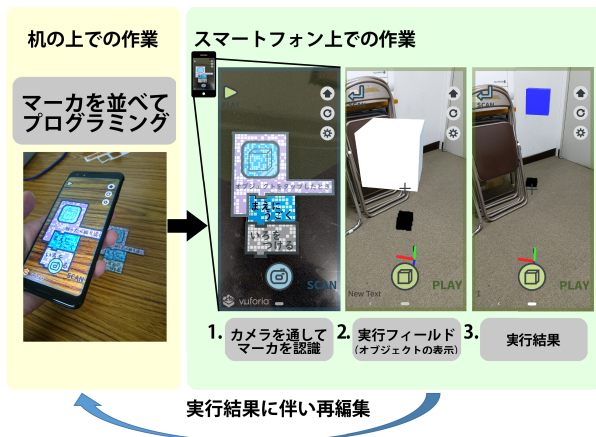


図 1. 本システムを使ったプログラミングの流れ

ングのように直感的にプログラミングすることができ、AR を用いて物理的な制限なく現実と重畳しながら実行できるもので、ビジュアルプログラミングと物理プログラミングの利点を保持しつつ、双方の欠点を補えるものとなっている。

3 提案システム

本ツールは、机の上でブロックを並べることで物理的にプログラミングした結果が、スマートフォンやタブレットの画面を通じて現実世界に重畳されるという形で実行される。本システムにおけるプログラミングから実行までの流れを図1に示す。プログラミングには「マーカブロック」というタンジブルに組み合わせることができる木製のブロックを利用する。マーカブロックは、Scratch におけるブロックの概念に AR マーカの機能を付加したものであり、一つひとつに命令が記述されている。AR マーカの機能によりマーカの種類を判別している。ユーザがマーカブロックを取捨選択して並び替える作業が本システムにおけるプログラミングの作業にあたる。Scratch に倣って本システムのマーカブロックにも凹凸をつけることで不適切な順番で命令を並べることができないようなデザインにした。マーカには「まえにうごく」などの動作に関する命令や、「もし～～にふれたら」、「○回くりかえす」などの if 文や for 文に相当する制御に関する命令が記載されている。マーカブロックを組み合わせることによって作成したプログラムは、現実世界に重畳させる CG オブジェクトそれぞれに対してプログラミングされる。この CG オブジェクトは、Scratch におけるスプライト（キャラクタ）にあたるオブジェクトであり、現在の実装では、立方体オブジェクト、車のオブジェクト、ドローンのオブジェクト、アヒルのオブジェクトの四種類から選択できる。ユーザが机の上でマーカブロックを並べ終えてプログラムを完成

させたら、スマートフォンのカメラを通してプログラムを認識する。すべてのマーカブロックが認識されている状態で「ロード」ボタンを押すことでプログラムが読み込まれる。この作業が一般的なプログラミング言語におけるコンパイルの作業にあたる。

一般的なプログラミング言語では実行画面やコンソール等を通じてプログラムの実行結果を確認するが、本システムでは現実世界すべてが実行画面に対応する。そこで、本システムにおいて、実行結果を重ねられる現実世界のことを「実行フィールド」と呼ぶ。実行結果を表示するにあたり、まず、実行フィールドにおける座標の中心となる基準点を設定する。この作業は、ユーザが現実世界における床をスマートフォンでタップすることによって行う。次に、実行フィールド内のオブジェクトを出現させたい場所にスマートフォンを移動させて、オブジェクト出現ボタンをタップすることでフィールド内の任意の位置にオブジェクトが出現する。このオブジェクトは拡張現実空間における CG オブジェクトであるため、ユーザが移動して別の角度から見ても、初めに出現させた位置に留まり続ける。

以下に提案システムの詳細を述べる。

3.1 マーカブロックの種類

本システムにおけるプログラミングはマーカブロックを組み合わせることで行われる。マーカブロックは図1及び図2及び図3の一部に示されているような凹凸のあるカラフルなブロックである。これらは、Scratch を参考にして制作したものであり、Scratch と同様に命令のジャンルによって色分けされている。命令のジャンルとして、「イベント」、「制御」、「動き」、「見た目」の4種類のマーカブロック合計28種類が実装されている。以下に、現時点で制作、実装済みのマーカブロックについて簡単に述べる。また、実装済みのマーカブロックの一覧を表1に示す。

まず、「イベント」マーカブロックは、プログラムを実行させるきっかけを与えるための命令ジャンルで、紫色で実装している。これらは、実行フィールド上に配置したプログラミング済みのオブジェクトに対して、命令を実行させるトリガーとなる命令である。

「動き」マーカブロックは、オブジェクトの動きに関する命令を表すもので、青色で実装しており、18種類のマーカブロックを実装している。命令は主に座標や回転角度に関するものである。

「動き」マーカブロックの一部には数値を入力できるように数字の書かれた円形の歯車のようなマーカを取り付けることもできる。数値入力の対象であるマーカブロックは入力された数値に応じて動き方が変化する。各マーカブロックへの入力方法としては、数字が書かれているマーカを回転させ、マーカの角度によって10段階(0-9)に数値を入力すること

表 1. マーカブロック一覧

種類	マーカブロック	数値入力
イベント	オブジェクトを タップしたとき	×
	触ったら繰り返す	×
動き	まえにすすむ	○
	うしろにすすむ	○
	うえにすすむ	○
	したにすすむ	○
	みぎにすすむ	○
	ひだりにすすむ	○
	X,Y,Z ほうこうに みぎかいてん	○
	X,Y,Z ほうこうに ひだりかいてん	○
	重力	×
	おおきさをかえる	○
	カメラにむける	×
	音を出す	○
	プラスにする	×
	マイナスにする	×
見た目	いろをつける	○
	きえる	×
制御	地面にふれたら	×
	もしカメラにふれたら	×
	もしはこにふれたら	×
	もしクラップしたら	×
	○回くりかえす	○
	ここまで	×

ができる。その他の数値入力機能として、上記数値マーカと同様の形で「中心と自分との距離」と書かれたマーカを使うことで数値の入力ができる。このマーカで数値入力をした場合、実行フィールド内における座標の原点とスマートフォンまでの距離が数値として入力される。

「見た目」マーカブロックは、オブジェクトの見た目に関する命令を表すものであり、「いろをつける」、「きえる」の2種類がある。「いろをつける」はオブジェクトに色をつける命令で、「きえる」はオブジェクトを消去する命令である。

「制御」マーカブロックは、プログラミングにおいて重要な、様々な制御を行うための命令を表すものであり、if 文、for 文に対応するようなマーカブロックが用意されている。if 文に対応する制御マーカブロックとして、全ての制御マーカブロックと「}」に対応する「ここまで」を実装した。「地面にふれたら」は、オブジェクトが地面に触れた際に「地面にふれたら」から「ここまで」の中の命令が実行され

るというもので、「カメラにふれたら」は同様にオブジェクトがカメラに触れた際に実行されるというものである。「はこにふれたら」は現実世界に事前に用意した立方体の箱を実行フィールド内で認識することで配置された AR オブジェクトの箱に触れた際に実行されるものである。「クラップしたら」はユーザが使用するスマートフォンに付属しているマイクの近くで拍手をした際に命令が実行されるというものである。以上が if 文に対応するマーカブロックであり、for 文に対応する制御マーカブロックとしては「○回くりかえす」がある。

3.2 AR を用いたことによる特徴的なプログラム

現状のシステムは、まだプログラミング可能な命令の種類が限られているが、3.1 節で説明したマーカブロックを組み合わせることで基本的な動作に関するプログラムが作成できる。本節では、AR 技術を利用していることにより実現可能な、現実と CG オブジェクトとのインタラクションを可能とするプログラムの例を紹介する。現状のマーカブロックを用いて制作した本システムの特徴的なプログラムとして、AR 技術を利用したラジコンのプログラムを図2に示す。このプログラムでは、車のオブジェクトが地面に落下した後、時計周りに曲がりながら地面を走行する。プログラムが実行されている間にユーザがスマートフォンのマイクに向けて拍手すると、その瞬間「マイナスにする」というマーカブロックが作用し、回転方向が逆転する。その後すぐに数値がプラスになり、時計回りに戻る。

他にも、オブジェクトがユーザ（カメラ）の方に向かってくる鬼ごっこのプログラムや、図3のような実物の箱と AR のオブジェクトを組み合わせたホッケーゲームのようなプログラムも実装可能である。以上のように、本システムはプログラミングでありながらも現実と密接に繋がった体験を提供できる。これにより、ユーザがプログラミングをより身近なものとして感じられるようなシステムとすることを目指している。

3.3 想定される教材としての利用方法

本システムは、プログラミング学習を支援することを目的として開発しているため、プログラミング学習を支援する教材としてどのような利用方法を想定しているかについて記述する。本システムは、児童全員に Android OS のスマートフォンやタブレットを配布し、紙に印刷したマーカブロックを用意することができれば一人一人がプログラミング体験をすることができるため、低コストで導入できる。また、一人一台の端末が用意できなくても、また、一人一台の端末が用意できなくても、数人に一台の Android 端末で実行結果を確認させるという利用方法も考えられる。これは、プログラミングの作業と

タンジブルなブロックを組み立てることにより AR 空間で実行可能な教育用プログラミングツール

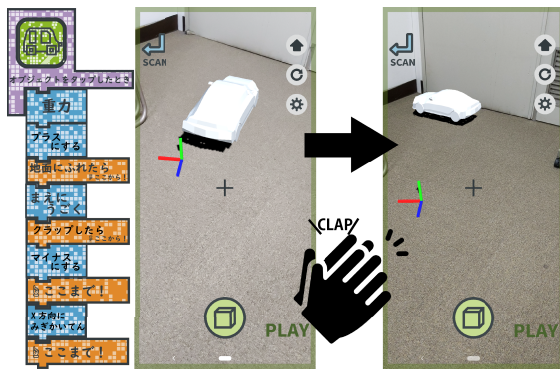


図 2. AR 技術を用いた現実とのインタラクションを可能とするラジコンのプログラム

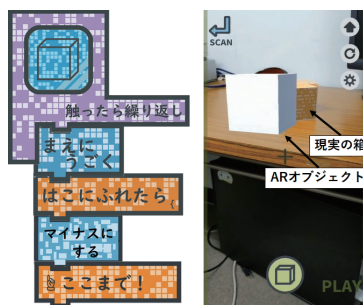


図 3. AR 技術を用いた現実とのインタラクションを可能とするホッケーゲームのプログラム

実行結果の確認が独立した工程となっているからこそ可能な方法である。実体のあるマーカーブロックを使うため、PC を使ったプログラミングと違い、ペアプログラミング等の複数人の協力によるプログラミング学習も比較的やりやすいものと考えている。

すでに多くの小学校で導入され始めている PC を利用した授業の問題点として、児童が PC を勝手に操作してしまうことを教員一人では完全に制御しきれないことが挙げられる。それに対して、本システムを用いてマーカーブロックを並べる作業は、教員一人でも児童の作業を全体的に観察できるため、児童の様子を見ながら授業を進めることができる。

4 ユーザスタディ

ユーザスタディを通じて本システムの評価を行った。評価は、5 名のプログラミングスクールの教員に本ツールを使用してもらった後に、「教育機関において複数の児童に対して本ツールを使った際のメリットとデメリット」についてのインタビューに回答してもらうという形式で行った。

4.1 実験条件

被験者は著者が所属しているプログラミングスクールの教員で、児童に対してビジュアルプログラ

ミングを教えた経験がある 5 名に評価してもらった。被験者の教員らが児童に教育している言語は Scratch であり、全員ビジュアルプログラミングの知識を持っており、また、児童にプログラミングを教える際の授業の様子を知っている。5 名の被験者全員に本ツールの操作説明と簡単なプログラムの例を見せた後、数分間自由に本ツールを体験してもらった。その後、本ツールのメリットとデメリットについて口頭で回答するという形で評価してもらった。

4.2 評価結果

評価の結果、複数の児童が同時に本ツールを使った際のメリットとして、「このツールであれば隣の席の児童同士と一緒にプログラミングが可能である」という意見が挙げられた。また、他に挙げられたメリットとしては「机の上にプログラムコードがあることで教員が児童のプログラムをすぐにチェックでき、プログラムの修正をさせることが容易にできる」という意見も挙げられた。

そしてデメリットの一つとして挙げられた評価としては「スマートフォンのカメラを使うことによる児童に対してのプライバシー保護についての機能が無い点」という意見が得られた。児童が所有しているスマートフォンを授業において使用することは、プライバシーのリスクから難しいのではないかという意見が得られた。

4.3 考察

評価の一つとして、実際のブロックを使ってプログラミングしていることから複数人でのプログラミングが可能であるという意見が得られた。このことから、本システムを使えば、学校で機材が人数分用意できなくとも全員にプログラミング教育が可能になるのではないかと考えられる。また、教員が複数の児童のプログラムを見る際にも容易に児童の進行度を確認することができると考えられる。一方で、教育現場でのプライバシーの配慮に関しても意識しながら今後の開発を進める必要がある。

また、本ツールに対するコメントとして、児童が作ったプログラムに対して評価をすることができるよう機能の追加を提案された。現状、児童のプログラムを評価は、教員がそれぞれの判断で決めることになっている。そのため、教員ごとに評価が異なってしまうことがある。ツール内での評価を可能とすることで、児童のプログラムの評価は統一され、評価のばらつきが少なくなると考えられる。

教育機関において使用されるようなツールとして開発する際には、児童にとっても、教員や学校にとっても使用しやすいツールとして開発する必要がある。

5 まとめ

本稿では、ビジュアルプログラミングと物理プログラミングのメリットを享受しつつ AR 技術を用いることによりそのデメリットを補うツールの提案をした。そして、教育機関における教材としての可能性について検討した。本システムは、物理プログラミングのようにタンジブルにプログラミングを行い、その結果を AR 技術によって現実世界に重畳させる。それにより、直感的なプログラミングを可能としながら、拡張現実空間における CG アニメーションという形で物理法則に縛られない自由なプログラミングを実現するものである。タンジブルなブロックを使いつつ、PC を使わずにプログラミングが可能なことから、複数人でのプログラミングが容易である。それに加え、教員が児童の作ったプログラムのチェックを容易にできるのではないかという意見をもらった。

本研究の今後の課題として、児童が本ツールを用いてプログラミングをすることで、実際に学びやすくなっているかを検証することが挙げられる。しかし、現状のマーカブロックの種類が少なく自由なプログラムが組めないため、有効な評価が期待できない。そのため、今後、マーカブロックの種類を増やし、自由なプログラムを組めるような環境を整える予定である。それに加えて、ユーザのプログラミン

グの直感性を上げるために、AR 技術を用いることに関連したユニークな命令やプログラミング体験のデザインを今後も引き続き行っていきたい。

参考文献

- [1] A. Kelly, R. B. Shapiro, J. de Halleux, and T. Ball. ARcadia: A Rapid Prototyping Platform for Real-time Tangible Interfaces. (4):D314:1–D314:4, 2018.
- [2] S. Magnenat, M. Ben-Ari, S. Klinger, and R. W. Sumner. Enhancing Robot Programming with Visual Feedback and Augmented Reality. pp. 153–158, 2015.
- [3] M. Resnick. Scratch: Programming for All. 2009.
- [4] 坂本恋, 大島登志一. MR Code Weaver : 投影型ミクストリアリティによるタンジブルなプログラミング学習ツール. インタラクシオン 2018 論文集, 情報処理学会 2018, 2018.
- [5] 総務省. 報道資料「平成 30 年通信利用動向調査の結果」.
- [6] 八城朋仁, 迎山和司. 物質プログラミング -物質によるプログラムの可視化と開発環境の制作-. 情報処理学会インタラクシオン, 2014.
- [7] 文部科学省. 小学校プログラミング教育の手引 (第二版) .

未来ビジョン

将来的には、ネットワークを使って各デバイス間の通信を行うことでユーザが作ったプログラムを拡張現実空間内で共有できるようにする予定である。複数の児童がプログラミングをする場面において、児童が席を立ち、他の児童の画面を見に行くという場面が見られる。そういった場合に、プログラムが AR 空間内で共有されていれば席を移動せずとも確認が可能になる。我々は、他人が作ったプログラムと相互に作用するような機能を追加し、より魅力的なプログラミングツールとしたい。そして、他人が作ったプログラムに対してだけ作用することにとどまらず、現実の物体に対しても作用できるようなツールの実現を目指したい。それが実現できれば、生活において便利なものを作成できるツールとなることが期待できる。自身にとって便利な物を作るというモチベーションがあれば、初学者が自発的にプログラ

ミングを学ぶ姿勢を身につける助けとなると考えられる。

また本ツールの技術的な記述をすると、特定のマーカブロックに数値を入力することはできるが、現在は「10 段階の数値入力」、「中心と自分との距離」以外に数値入力の方法は無いため、今後はユーザの周辺の気温やユーザがアプリケーションを開いている時間などの一般的なプログラミング言語でいう変数を入力できるようにしていく。

そして、本ツールのプログラミング教育への有効性を測るための将来的な実証実験では、実際に児童を対象として、本システムを体験するワークショップを開く予定である。ワークショップを通して本システムのプログラミング教育における効果を検証していきたいと考えている。実際に児童を対象としたワークショップを行うことで、本ツールを児童が使う際に必要な機能を考察することが可能である。