

SorToio: 小型群ロボットを用いたアルゴリズム学習支援システムの提案

板垣 智也* 塚田 浩二*

概要. 近年、プログラミング教育の早期化にともなって、小学生でも理解しやすい工夫を行ったプログラミング教材などが開発されてきた。その中でも、手に取って扱えるタンジブルなオブジェクトを使ったプログラミング手法が多数提案されている。一方で、ソートや探索のアルゴリズムを学習するためのタンジブルなシステムは少ない。そこで本研究では、小型の群ロボットである toio を物理的に動かすことで、データ構造やアルゴリズムの学習を支援するシステム「SorToio」を提案する。各ロボットに値を持たせ、ロボット1台を定数、それらの並びを配列として扱うことで、配列内の要素の比較や交換をロボットの動作で表現することが可能になる。このシステムを用いて、データ構造やソートアルゴリズムなどを分かりやすく可視化する。本稿では、ソートの中でも単純なバブルソートと木構造を用いるヒープソートの2つを toio の動きで実装した。

1 はじめに

近年、日本では小学校での必修化をはじめとして、プログラミング教育の早期化が進んでいる。実務で広く利用されるテキストプログラミングは、初学者にとってハードルが高く学習しにくいいため、小学生でも理解しやすい工夫をしたプログラミング教材などが多数開発されている。例えば、理解しやすいプログラミング言語として Scratch[8] がある。Scratch はビジュアルプログラミング言語のひとつで、テキストではなくブロックを組み合わせることでプログラミングを行うことができ、入力もマウスのみで行うことが可能である。そのため、世界各地でプログラミング初学者を対象にした Scratch での情報教育が行われている。他にも、micro:bit[7] はセンサや LED が搭載された教育用マイコンボードで、センサや LED を内蔵しており、ビジュアルプログラミングからテキストプログラミングまでサポートされているため、段階的なプログラミング学習が可能である。

その中でも、手に取って扱えるタンジブルなオブジェクトを用いたプログラミング手法が提案されている。例えば、センサやアクチュエータ等の機能を持ったブロックをケーブルを接続してプログラミングを行う研究 [13] や、プログラミングの構成要素を VisualMarker として用意し、それを並べてスマートフォンで撮影することで、プログラミングを行う研究 [10] 等が提案されている。こうしたタンジブルなオブジェクトを用いたプログラミング手法では、複数人が同時に参加できたり、GUI よりも直感的な操作ができる可能性がある。

このように、初学者向けのプログラミング支援手法は多数提案されている一方で、ソートや探索等のアルゴリズムを学習する教材や研究事例はあまり多くない。アルゴリズム学習は、論理的思考力や問題解決能力の向上に効果があるとされており高校の学習指導要領 [12] でも、整列や探索などの基本アルゴリズムやその効率化を体験的に理解させるよう求めている。

そこで、本研究では、手に取って扱える小型の群ロボット (toio) を物理的に動かすことで、データ構造やアルゴリズムの学習を支援するシステム SorToio を提案する。各ロボットには任意の数値が付与されている。ユーザがロボットを配置して、アルゴリズムを選択すると、ロボットが順番に移動して、ソート等のアルゴリズムを物理的な動きとして観察することができる (図 1)。ロボット自体を入力装置としても扱うことで、入力/出力インタフェースを一体化して、初学者に分かりやすい設計を目指す。



図 1. システム利用例

Copyright is held by the author(s). This paper is non-refereed and non-archival. Hence it may later appear in any journals, conferences, symposia, etc.

* 公立はこだて未来大学

2 関連研究

2.1 小型ロボットを用いた情報提示

Sketched Reality[4]はARスケッチツールで描かれた仮想スケッチ（線、壁、振り子、バネなど）と実世界の小型ロボットのtoioが、同期して動作するシステムである。仮想世界と物理世界の双方向からインタラクションを行うことで、タンジブルな物理教育や機構学習などの可能性を示した。

Zooids[5]は円柱型車輪付き小型ロボット（Zooid）を多数利用することで、群ロボットを用いたインタラクション手法を提案した。例えば、グラフの点や線をZooidsでプロットすることで、時間経過によってデータが更新される動的なグラフを実世界上に再現した。

また、Zooidsを応用した研究[6]では、Zooidにユーザーデータを割り当てることで、ある教科の点数が高い生徒を特定のZooidに引き寄せたり、地図の上に置くと出身地の場所へ移動する機能など、タンジブルなインタラクション手法を実現した。

2.2 タンジブルインターフェースによる情報教育

Plugramming[13]はセンサやアクチュエータ等の機能を持ったブロック同士をプラグケーブルでつなぐことで、プログラムを作成することができる。センサのブロックには、センサの値によって条件分岐をさせる機能があり、周辺の環境によって条件が変わるプログラムを作成可能になる。実験を行っており、目的であったプログラムの実行順序の学習ができており、グループでの協調作業を促すことにも効果的であったと報告されている。

また、トイオ・プレイグラウンド[9]は、タンジブルなカードを組み合わせて、小型ロボットtoioの動作を制御することができる。各カードにはIDが付与されており、toio下部のイメージセンサで読み取る。「こっちを向く」「はやくまっすぐ」といったカードの上をtoioが通ることで、その挙動が動的に変化する。また、ループ等のカードもあり、ごく基礎的なプログラミング制御も体験できる。

2.3 アルゴリズムの可視化

コンピュータを使わずにアルゴリズムを可視化する例として、Computer Science Unplugged(CSアンプラグド)[1]がある。ソートアルゴリズムや探索アルゴリズム、オートマトンを、数字の書かれたカードやピンポン玉などを用いてゲームをする感覚で学ぶ教育法である。多言語に翻訳されており、日本でもコンピュータを使った学習よりもCSアンプラグドの方が分かりやすかったという実験結果が報告されている[11]。本研究では、小型ロボットを用いてアルゴリズムを可視化することで、初学者が理解しやすいように工夫する

アルゴリズムの可視化をコンピュータ上で行った例として、VisuAlgo[3]がある。VisuAlgoはWebベースで作成された、アルゴリズムの可視化ツールで、多くのデータ構造とアルゴリズムが視覚化されている。配列やノードの値は、ユーザが任意に設定できるようになっている。本研究では、アルゴリズムの可視化をロボットで行うことで、配列の変更をキーボードやマウスの入力を必要とせず、ロボットの位置変更だけでスムーズに行える。

タンジブルインターフェースとARを利用してアルゴリズムを学習させる研究にARCat[2]がある。モバイルデバイスを用いたAR上に迷路と猫を表示し、ユーザがタンジブルなカードを使って、猫をゴールに導くゲームを実装している。このゲームで遊ぶことにより、深さ優先探索(DFS)の概念を直感的に学ぶことができる。猫は、前方>左>右>後方の順で探索することが決められており、ユーザがカードを使って、「障害物のあるマスには進まない」、「すでに通った道ならステップバックする」などを追加で定義していく。本研究では、AR上ではなく現実世界のロボットを使うことで、ユーザがロボットを手で動かしながら考えることができる。

3 提案

本研究では、手に取って扱える小型の群ロボット(toio)を物理的に動かすことで、データ構造やアルゴリズムの学習を支援するシステム「SorToio」を提案する。まず、各ロボットに値を持たせることで、ロボット1台を定数、その並びを配列として扱えるようにする。次に、配列内の要素の比較や交換をロボットの動作で表現することで、データ構造やソートアルゴリズム等を分かりやすく可視化する。ユーザがロボットを配置して、アルゴリズムを選択すると、ロボットが順番に移動して、ソート等のアルゴリズムを物理的な動きとして観察することができる。さらに、ロボット自体を入力装置としても扱うことで、入力/出力インターフェースを一体化して、初学者に分かりやすい設計を目指す。

3.1 システム構成

本システムは、小型自走ロボットのtoio、toio開発用A4プレイマット、制御用PC、補助的な情報提示用のディスプレイを中心に構成される。

本システムの利用の流れを説明する。事前準備として、各toioの個体ID(Shortened Local Name)と任意の定数を、PC上のソフトウェアで対応付ける。標準設定としては、 $1 \sim n$ ($n = \text{toioの台数}$)を割り当てる。現段階では、各toioの識別が容易なように、定数をラベルとして貼り付けている。

ユーザはプレイマット上の任意の位置にtoioを配置する。現在のシステムでは一次元の配列を想定しているため、各toioの左右方向の座標に基づいて

配列の順序が決定される。

ディスプレイ上には、アルゴリズムの説明が表示されており、toio 自体をボタンとして押し込むことで、アルゴリズムを切り替えることができる。たとえば、toio を長押しすることで、選択されたアルゴリズムにしたがって、toio が動作を開始する。

3.2 表現対象のアルゴリズム

本論文では、バブルソートとヒープソートの2つのアルゴリズムを、toio の動きで表現する。ここでは、各アルゴリズムについて紹介する。バブルソートはソートの中でも単純なアルゴリズムであり、大学などの講義でも最初に学ぶことが多いため、初学者向けのアルゴリズム学習に適すと考えた。

ヒープソートは、二分ヒープと呼ばれる木構造を用いたアルゴリズムである。配列を木構造として扱いながらソートを行うため、一般的に初学者には理解が難しい。本システムでは、toio を二次元的に動かして配列と木構造を表現できるため、理解しやすくなると考えた。

4 実装

4.1 ハードウェア

前述したように、本システムでは、小型自走ロボットの toio を利用する。約 30g と軽量であるため片手で容易に動かすことができ、ボタンや加速度センサを内蔵しているため入力手法としても活用できる。また、専用のプレイマットと組み合わせることで、絶対位置を取得できる。各 toio は、BLE (Bluetooth Low Energy) を介して、ホスト PC (OS: Windows

10 Home, CPU:Corei5-1135G7 2.40GHz, メモリ: 8GB) 上の Python プログラムから制御する。ディスプレイへの表示や補助操作の GUI は、TkInter を用いて実装した。なお、toio の同時制御台数は明確には規定されておらず、ホスト PC の性能に影響される。著者らの環境で安定して制御可能な台数は 7 台であったため、本論文の実装では最大 7 台の toio を利用する。

4.2 アルゴリズムの実装

4.2.1 実装例 1: バブルソート

バブルソートは、配列内の隣り合う値を端から順に比較・交換する操作を繰り返すことで整列する方法である (図 2)。本研究では、配列の長さ n は toio の台数となり、前述の実装上の理由から最大 7 とする。さらに、プレイマットに配置した状態で一番左側にある toio を配列の 0 要素目、1 番右側にある toio を配列の $n-1$ 番目の要素として扱い、ソートは昇順で行う。

ユーザが toio をマット上に配置して、GUI 上で「bubble-sort」を選択すると、バブルソートの可視化が開始される。まず、全ての toio を Y 軸の中央に等間隔で一列に並び、その時点で配列の要素が確定する (図 3)。その後バブルソートのアルゴリズムに沿って、要素の交換があれば toio の位置の交換を行う。また、隣り合う値を比較する際、交換の有無にかかわらず、比較する toio を前進させて、比較対象の要素を分かりやすく表示している (図 4)。

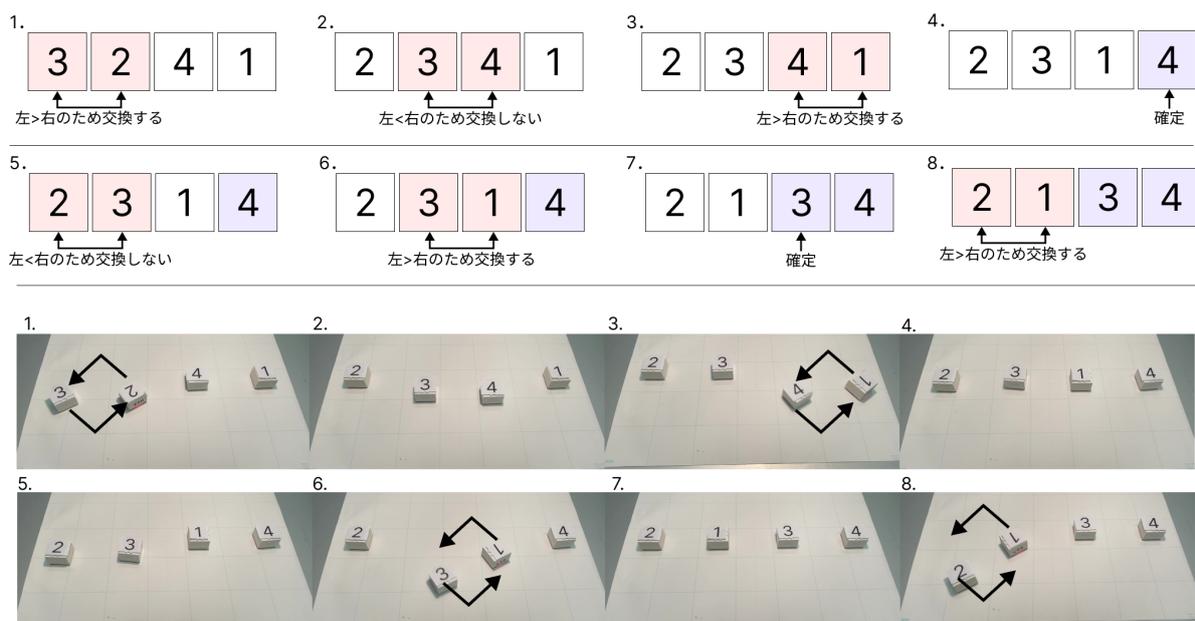


図 2. バブルソートの図解と実行の様子

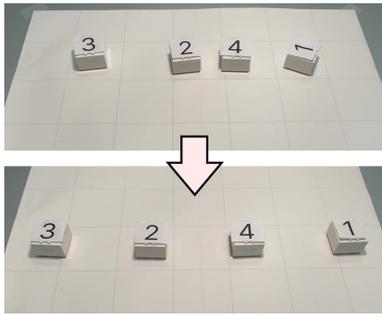


図 3. バブルソートのスタンバイ

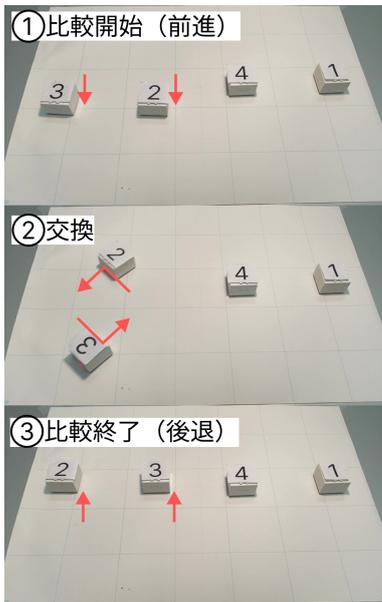


図 4. バブルソートの交換イメージ

4.3 実装例 2：ヒープソート

ヒープソートは、配列を木構造として扱い、二分ヒープという木の作成と根の移動を繰り返すことでソートを行うアルゴリズムである (図 5)。本実装では昇順にソートするため、親要素が子要素以下の値をとる、最小ヒープ (以下、ヒープ) を用いる。

ヒープソートのアルゴリズムを説明する。あるノードとその配下のノードを全てヒープを満たす、関数 `heapify` を定義する。`heapify` は、引数のノードとその子ノードを比較する。引数のノードが最小のノードでなかった場合、最小のノードと交換を行い、交換先で再び `heapify` を適用する。木全体をヒープにするには、葉以外のノードを根ノードに向けて順に `heapify` を行うことで構築できる。このヒープの構築を行う関数を `build_heap` とする。

まず、`build_heap` を行いヒープを構築する (図 5 の 1)。その後、根ノードをソート済み配列に移動させ、末尾のノード (葉の右端のノード) を根に移動させる (図 5 の 2, 3)。出来た木の根ノードに

`heapify` をすると、再び木がヒープとなる (図 5 の 4)。「根ノードと末尾ノードの交換」、「末尾ノードをソート済み配列に移動」、「根ノードに `heapify` を適用」をノードが無くなるまで繰り返すことができる。ノードが無くなるまで繰り返すと、ソート済み配列にてソートが完了している。

`toio` をマット上に配置して、GUI 上で「heap-sort」を選択すると、ヒープソートの可視化が開始される。まず、全ての `toio` がマット下部に一直列に並び、木構造の配置になるよう段階的に移動する (図 6)。前述したアルゴリズムで、ノードの交換が行われれば、`toio` の位置を移動させる。要素の比較は「`toio` を左右に回す」動作で、比較の結果最少だった要素は「`toio` を一回転させる」ことで示すなど、動作が分かりやすいように工夫している。ソート済み配列はマット下部に並ぶことで表現している。つまり、アルゴリズムが進むにつれ、`toio` がマット下部に昇順に並んでいく様子が確認できる。

5 議論

現時点では、ユーザが介入可能な操作は、配列の初期値の操作やアルゴリズムの選択に限定されており、タンジブルインタフェースの特徴を十分にいかしきれていない。アルゴリズムの実行途中で物理的に介入して順序を変更したり、途中でアルゴリズムを変更するような柔軟なインタラクションを実装していきたい。

また、現在の実装では、`toio` の物理的な移動距離に応じて実行時間が長くなってしまいうため、アルゴリズムの効率 (計算量) を体感することが難しい。よって、計算量を重視した表現モードも実装していきたい。具体的には、`toio` の移動速度や移動距離などを変化させて、ソート完了までの実行時間を計算量に併せて調整することで実現できると考える。

さらに、初学者に対するインタラクティブな教材としての機能の充実を行いたい。例えば、バブルソートの学習ならば、ソートされていない `toio` 群を提示し、学習者は比較すべき `toio` 2 台をアルゴリズム順に選択していく。システムでは、選択された `toio` の比較・入れ替え処理とともに、学習者の選択がアルゴリズムに沿っているかの判定を行い、フィードバックを行う。他にも、自動で行われるソートを一時停止したり、一手ずつ進める・戻す機能などを実装し、学習効果等を検証していきたい。

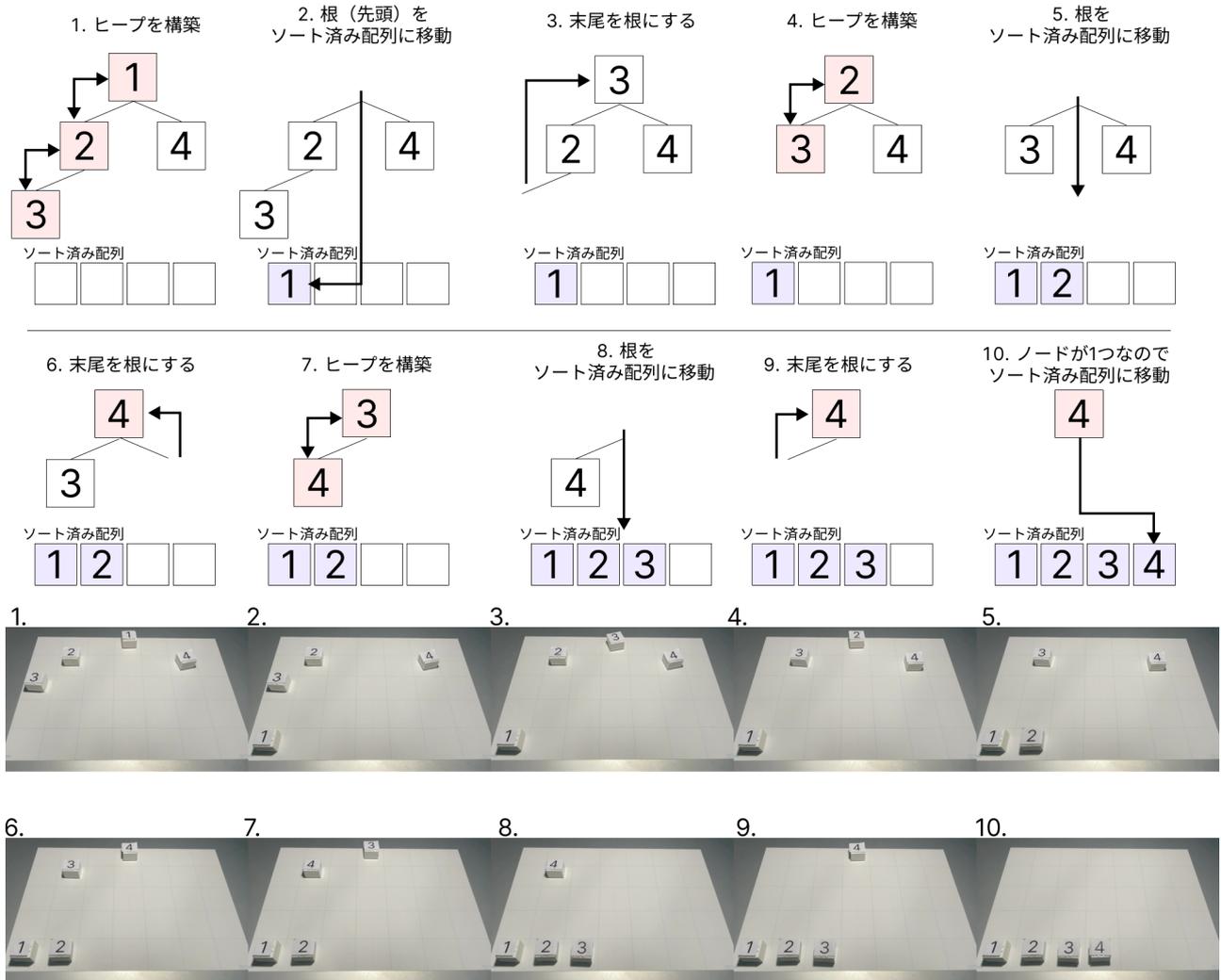


図 5. ヒープソートの図解と実行の様子

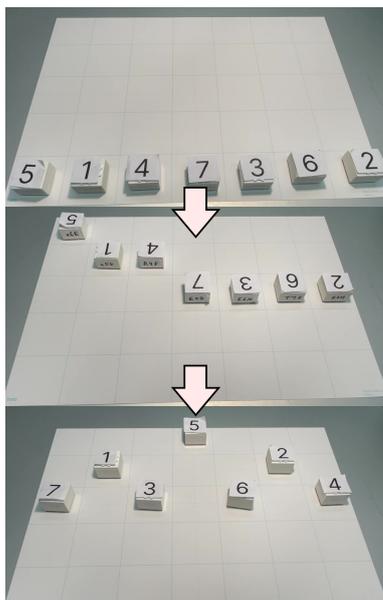


図 6. 木構造を作成する様子

6 まとめ

本研究では、小型の群ロボットである toio に値を持たせて物理的に動かすことで、データ構造やアルゴリズムの学習を支援するシステム「SorToio」を提案し、2つの実装例を提示した。今後は、システムの動作中の介入等を含めた、多様なインタラクションを提示することで、より体験的にアルゴリズムを学習できる仕組みを構築したい。さらに、多くのデータ構造、アルゴリズムを可視化し、計算量比較などの追加機能の実装などを行い、より良いシステムの開発と学習効果等の検証を行いたい。

参考文献

- [1] T. Bell, J. Alexander, I. Freeman, and M. Grimley. Computer science unplugged: school students doing real computing without computers. In *New Zealand Journal of Applied Computing and Information Technology*, Vol. 13, pp. 20–29,

- 2009.
- [2] X. Deng, D. Wang, Q. Jin, and F. Sun. AR-Cat: A Tangible Programming Tool for DFS Algorithm Teaching. In *Proceedings of the 18th ACM International Conference on Interaction Design and Children, IDC '19*, p. 533–537, New York, NY, USA, 2019. Association for Computing Machinery.
- [3] S. HALIM. VisuAlgo – Visualising Data Structures and Algorithms Through Animation. In *Olympiads in Informatics*, Vol. 9, p. 243–245, 2015.
- [4] H. Kaimoto, K. Monteiro, M. Faridan, J. Li, S. Farajian, Y. Kakehi, K. Nakagaki, and R. Suzuki. Sketched Reality: Sketching Bi-Directional Interactions Between Virtual and Physical Worlds with AR and Actuated Tangible UI. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology, UIST '22*. ACM, Oct. 2022.
- [5] M. Le Goc, L. H. Kim, A. Parsaei, J.-D. Fekete, P. Dragicevic, and S. Follmer. Zooids: Building Blocks for Swarm User Interfaces. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology, UIST '16*, p. 97–109, New York, NY, USA, 2016. Association for Computing Machinery.
- [6] M. Le Goc, C. Perin, S. Follmer, J.-D. Fekete, and P. Dragicevic. Dynamic Composite Data Physicalization Using Wheeled Micro-Robots. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):737–747, 2019.
- [7] Micro:bit Educational Foundation. micro:bit. <https://microbit.org/>. (参照 2024-08-31) .
- [8] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Miller, E. Rosenbaum, J. Silver, B. Silverman, and Y. Kafai. Scratch: programming for all. *Commun. ACM*, 52(11):60–67, nov 2009.
- [9] Sony Interactive EntertainmentInc. トイオ・プレイグラウンド — タイトル - TOIO. (参照 2024-08-31) .
- [10] 服部 圭介, 井上 馨太, 志賀野 桐生, 平井 辰典. 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. 日本ソフトウェア科学会 WISS 2019 予稿集, 2019.
- [11] 兼宗進, 正田良, 紅林秀治, 鎌田敏之, 井戸坂幸男, 保福やよい, 久野靖. コンピュータを使わない情報科学教育-Computer Science Unplugged の翻訳と実践-. 情報処理学会シンポジウム論文集, 6:5–10, 2007.
- [12] 文部科学省. 情報編 高等学校学習指導要領 (平成 30 年告示) 解説. <https://www.mext.go.jp/content/000166115.pdf>, 2018. (参照 2024-08-31) .
- [13] 八城 朋仁, 迎山 和司, 原田 泰. Plugramming : 協調作業のためのタンジブルなプログラミングツール. 日本デザイン学会研究発表大会概要集, 64:76, 2017.