SorToio: Learning Support System for Algorithm using a Small Swarm Robot

Tomoya Itagaki
g2125003@fun.ac.jp
Future University Hakodate
Hakaodate, Hokkaido, Japan

Koji Tsukada
tsuka@acm.org
Future University Hakodate
Hakaodate, Hokkaido, Japan

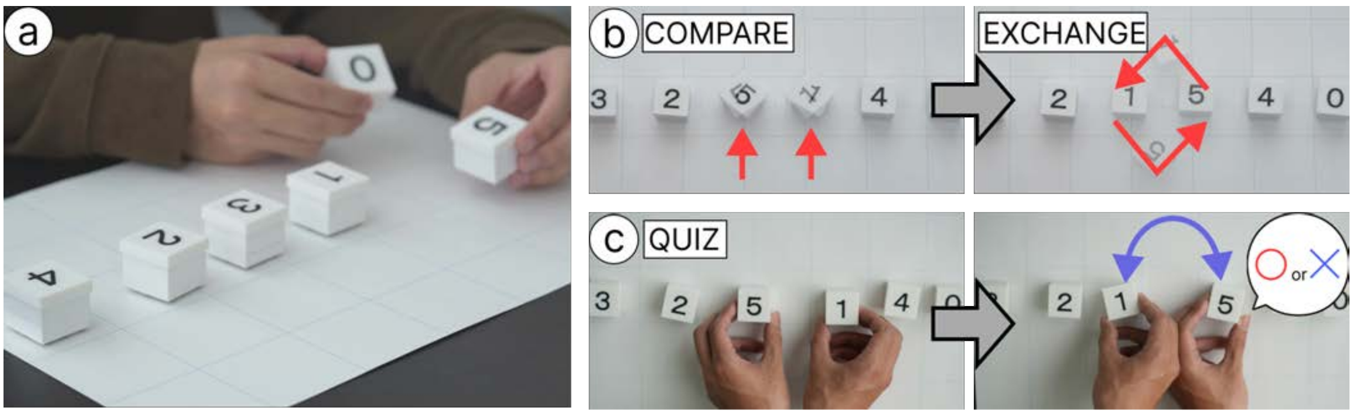

Figure 1: Overview of SorToio. (a) Appearance of a user to learn sort algorithm using SorToio. (b) Comparing/Exchanging elements. (c) The user can also manually exchange elements and obtain feedback.

ABSTRACT

Recently, various programming teaching materials have been developed in line with the early introduction of programming education. Among them, many use tangible objects. However, there are few tangible systems for learning sorting and search algorithms. Therefore, we propose "SorToio," a system that supports users to learn data structures and algorithms by physically moving small swarm robots. By assigning a number to each robot and treating the arrangement of robots as an array, the comparisons/exchanges of elements can be expressed by the movement of the robots. This allows users to observe the flow of algorithms through the physical movements of the robots. In addition, the user controls the movement state (e.g., play, pause, or reverse) by tapping or flipping the robot. Furthermore, the system includes a quiz mode that enables users to manually exchange robots while proceeding with the algorithm step-by-step. In this paper, we introduce two algorithms (bubble sort and heap sort) and two data structures (stack and queue) using SorToio.

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

CCS CONCEPTS

• **Human-centered computing → Human computer interaction (HCI)**.

KEYWORDS

Small Robots, Algorithms, Information Education

ACM Reference Format:
Tomoya Itagaki and Koji Tsukada. 2018. SorToio: Learning Support System for Algorithm using a Small Swarm Robot. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX).* ACM, New York, NY, USA, 3 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Therefore, in the field of human–computer interaction (HCI), programming methods using tangible objects that can be handled manually have been proposed [3][7]. These methods offer several advantages, including the abilities to explicitly demonstrate programming rules based on the shape of objects, learn by physically handling objects such as building blocks, and operate objects using both hands. However, there are few examples of support systems for sorting algorithms that use tangible objects. Algorithms are also an important component of information education, and there is a growing need to help children understand them through hands-on experience. There are algorithm learning methods that do not use electronic devices, such as CS Unplugged [1], but they face issues, such as the need to memorize rules and play with multiple people on certain topics. Therefore, we propose SorToio, a system that supports learning algorithms and data structures by physically moving

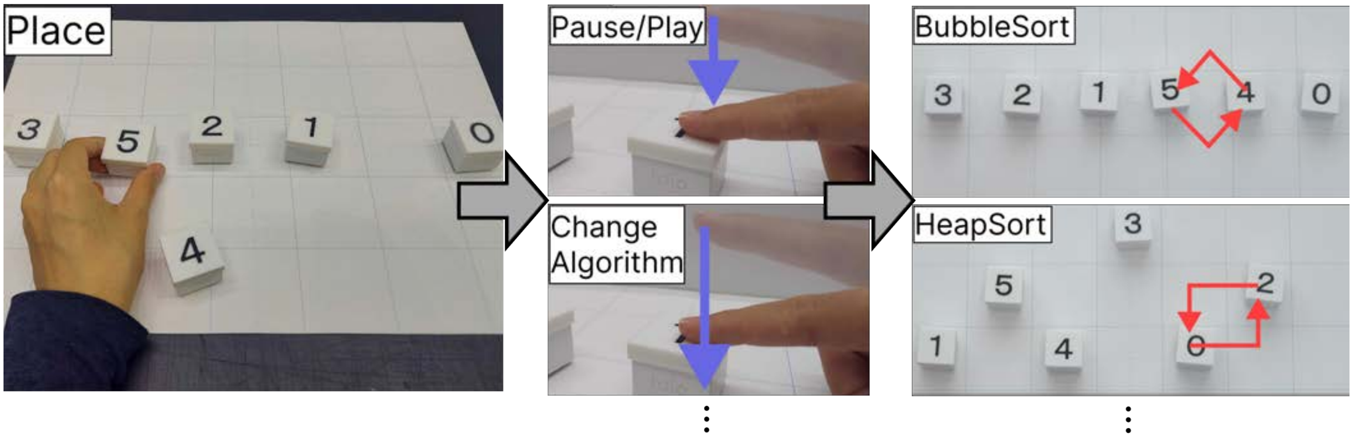

Figure 2: Process of using SorToio.

small swarm robots (toio by Sony). Various examples of learning support systems using small robots have been applied not only to programming education [4] but also to physics [6][5], art [8], and medical education [2]. SorToio is unique because of its focus on algorithms and data structures.

2 SORTOIO

In this study, we developed SorToio, an algorithm learning support system that uses small swarm robots. Each robot is assigned an arbitrary number, allowing it to be treated as an array. By expressing the comparison and exchange of elements in an array through robot movements, sorting algorithms and data structures can be visualized. The process of using SorToio is illustrated in Figure 2. Users can observe the flow of algorithms through physical movements and learn while interacting with the robots. The following explains SorToio's movements and interactive design.

Movement. The following movements are examples of characteristic features of sort algorithms:

- Comparison of array elements: swaying from side to side.
- Comparison result: rotating once.
- exchanging of array elements: exchanging positions.
- Start and end of sorting: taking a bow in order.

Interaction. This section explains the basic interactions provided by SorToio. Most operations can be performed using only toio:

- Play/stop algorithm: tapping the toio.
- Reverse algorithm: double-tapping the toio.
- Launch quiz mode: Lift the toio.

In quiz mode, the user must manually exchange the correct robots. If the operation is correct, the robots move around joyfully. If failed, the robots return to their original positions. Because the robot is lightweight and has low torque, users can easily interrupt movements.

2.1 HARDWARE

We used Sony's "toio" as our small robot. toio has two wheels and can be controlled from a PC. In addition, the included mat has a special pattern printed on it, which allows the reading sensor built into the bottom of toio to detect its absolute position in real time. Furthermore, it has built-in buttons and various sensors that allow unified input and output mechanisms. We attached parts printed with numbers using a 3D printer to the top of the toio. Each toio was controlled via BLE from a Python program on the host PC.

3 EXAMPLES OF ALGORITHMS

As examples of algorithms, we implemented "bubble sort," "heap sort," and "stack / queue." The algorithm can be changed by pushing down the robot itself. This action is detected by the push-button below the toio, which changes the color of the embedded LED.

3.1 BUBBLE SORT

Bubble sort is a method of sorting by repeatedly comparing and exchanging adjacent values in an array from one end to the other. Bubble sort is a relatively simple sorting algorithm that is often taught first in classes; therefore, we thought it would be suitable for teaching algorithms to beginners. When the user taps the robots in any order, all robots line up in a row, and the visualization of bubble sort begins(Figure 2).

3.2 HEAP SORT

Heap sort is a fast sorting method that treats an array as a binary tree. It compares the parent and child elements and repeats exchanges such that the smaller value becomes the parent element. This sorting method treats the array as a tree structure, which is generally difficult for beginners to understand. Using SorToio, both array and tree structures can be smoothly represented using robot movements. When the user taps the robots in any order, all robots move to a configuration that mimics a tree structure. Subsequently, visualization of heap sort begins(Figure 2).

3.3 STACK/QUEUE

Stack and Queue are data structures, with LIFO (Last-In-First-Out) and FIFO (First-In-First-Out). We implemented stack and queue, which are simple data structures suitable for beginners. In this system, the mat is divided into upper, middle, and lower sections.

When the user taps the robots in any order, all robots line up and move into the upper space (waiting area). When the user taps a robot in the waiting area, the element on the far left moves to the middle space (data structure area). When the user taps a robot in the data structure area, the robots move one-by-one to the lower space (used area) according to the stack/queue structure.

4 FUTURE WORK

In future research, we plan to develop additional algorithms, improve usability, and create learning support materials. We will then conduct user evaluation experiments using these materials to confirm their effectiveness.

REFERENCES

[1] [n. d.]. *CS Unplugged.* Retrieved July 11, 2025 from https://www.csunplugged.org/en/

[2] Mehrad Faridan, Marcus Friedel, and Ryo Suzuki. 2022. UltraBots: Large-Area Mid-Air Haptics for VR with Robotically Actuated Ultrasound Transducers. In *Adjunct Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) *(UIST '22 Adjunct)*. Association for Computing Machinery, New York, NY, USA, Article 101, 3 pages. https://doi.org/10.1145/3526114.3561350

[3] Michael S. Horn and Robert J. K. Jacob. 2007. Designing tangible programming languages for classroom use. In *Proceedings of the 1st International Conference on Tangible and Embedded Interaction* (Baton Rouge, Louisiana) *(TEI '07)*. Association for Computing Machinery, New York, NY, USA, 159–162. https://doi.org/10.1145/1226969.1227003

[4] Sony Interactive Entertainment Inc. [n. d.]. *GoGo Robot Programming The Secret of Logibo* . Retrieved July 11, 2025 from https://toio.io/titles/gogorobot.html

[5] Hiroki Kaimoto, Kyzyl Monteiro, Mehrad Faridan, Jiatong Li, Samin Farajian, Yasuaki Kakehi, Ken Nakagaki, and Ryo Suzuki. 2022. 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* (Bend, OR, USA) *(UIST '22)*. Association for Computing Machinery, New York, NY, USA, Article 1, 12 pages. https://doi.org/10.1145/3526113.3545626

[6] Jiatong Li, Ryo Suzuki, and Ken Nakagaki. 2023. Physica: Interactive Tangible Physics Simulation based on Tabletop Mobile Robots Towards Explorable Physics Education. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference* (Pittsburgh, PA, USA) *(DIS '23)*. Association for Computing Machinery, New York, NY, USA, 1485–1499. https://doi.org/10.1145/3563657.3596037

[7] Amanda Sullivan, Mollie Elkin, and Marina Umaschi Bers. 2015. KIBO robot demo: engaging young children in programming and engineering. In *Proceedings of the 14th International Conference on Interaction Design and Children* (Boston, Massachusetts) *(IDC '15)*. Association for Computing Machinery, New York, NY, USA, 418–421. https://doi.org/10.1145/2771839.2771868

[8] Yue Zhu, Zhiyuan Zhou, Jinlin Miao, Haipeng Mi, and Yijie Guo. 2024. TangibleNegotiation: Probing Design Opportunities for Integration of Generative AI and Swarm Robotics for Imagination Cultivation in Child Art Education. In *Companion of the 2024 on ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Melbourne VIC, Australia) *(UbiComp '24)*. Association for Computing Machinery, New York, NY, USA, 66–70. https://doi.org/10.1145/3675094.3677586