WIP: Development and Pilot Results of ORBIT: a Visual Robot-Programming Tool Supporting Computational Thinking and Executive Functioning Skills in Special Education

Robert Hayes
Center for Engineering Education and Outreach
Tufts University
Medford, MA, USA
Robert.Hayes@tufts.edu

Elissa Milto
Center for Engineering Education and Outreach
Tufts University
Medford, MA, USA
Elissa.Milto@tufts.edu

Evelyn Goroza
Department of Mechanical Engineering
Tufts University
Medford, MA, USA
Evelyn.Goroza@tufts.edu

Jennifer Cross
Center for Engineering Education and Outreach
Tufts University
Medford, MA, USA
Jennifer.Cross@tufts.edu

2025 IEEE Frontiers in Education Conference (FIE) | 979-8-3315-0105-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/FIE63693.2025.11328697

***Abstract*—In the United States, 15% of K-12 students have some type of disability under the Individuals with Disabilities Education Act, and these students are provided fewer educational computing and robotics learning opportunities. This work-in-progress research paper describes the development and implementation of a custom visual robot-programming tool designed to support students in special education programs to practice computational thinking. This App was developed as part of an educational robotics program called Opportunities for Robotics, Building, and Inclusive Technology (ORBIT).**

We sought to answer (1) What types of resources can be provided by our program to enable students to meet their individualized learning goals while practicing computational thinking successfully? (2) What opportunities does participation in the program's activities provide for students to engage in computational thinking practices? Through co-design sessions with three special education teachers of multi-grade substantially separate middle-school autism classrooms, we explored the needs of their students to inform features of the ORBIT App. We have pilot-tested the App with students in two classrooms.

We present the salient features of our web-based, robot-programming tool and activities as a model of the resources that can enable the integration of robotics, computational thinking, and individualized learning goals in sub-separate autism classrooms. In addition, we present the preliminary results of our pilot test to illustrate how engagement with the co-designed resources created opportunities for practicing computational thinking while furthering individualized learning goals.

Our work-in-progress findings demonstrate the promise of the co-designed ORBIT programming tool. Our future work will include further iterations on the App design and associated materials with thematic analysis of the pilot data to reveal trends in student experiences.

***Index Terms*—component, formatting, style, styling, insert**

This material is based upon work supported by the National Science Foundation under Grant No. DRL-2318191.

I. BACKGROUND

Autistic students are underrepresented in STEM, both educationally and vocationally. We report on the in-progress educational robotics program, Opportunities for Robotics, Building, and Inclusive Technology (ORBIT), for autistic middle school students. Co-developed with educators, ORBIT is designed to integrate learning computational thinking (CT) practices with executive functioning (EF) skills. While the autism spectrum comprises individuals with unique characteristics and learning needs, EF skills are commonly prioritized in the learning goals of autistic students [1].

In school, autistic students do not get the same opportunities to engage in STEM as their neurotypical peers. This is due to a variety of factors, including special education services that take place outside of students' home classroom, which take time away from academics during the school day, and STEM curricula and tools providing limited support for equitable participation.

Our work on ORBIT integrates CT with EF to better address this gap in support and participation. ORBIT consists of a robotics coding application and a series of learning missions and analog EF tools to scaffold CT learning. Emphasis is on individualized learning and draws on the principles of Universal Design for Learning [2]. The ORBIT programming app, which will be referred to as "the App" throughout the paper, lowers barriers for entry while the supporting resources scaffold the needs of individual students.

[1]We use identity-first language (i.e., autistic student) rather than person-first language (student with autism) because that is generally preferred by individuals in the autism community [1].

II. Research Questions

The objective of the ORBIT project is to design, implement, and evaluate an educational robotics program, including activities and a custom visual robot-programming tool—the ORBIT App—which encourages students receiving special education services to practice computational thinking. Through a design-based research process, we sought to answer the following: (1) What types of resources can our program provide to enable students to meet their **individualized learning goals** while practicing computational thinking successfully? (2) What **opportunities** does participation in the program's activities provide for students to engage in **computational thinking practices**?

In this paper, we build on prior co-design research [3] to overview how the work-in-progress ORBIT App features are aligned to support autistic students' integrated CT & EF learning. We then present pilot data from initial trials with teachers, and with students, to examine ways in which the App supports students meeting individualized learning goals, and where we see evidence of opportunities for students to engage in computational thinking practices.

III. Theoretical Framework

Executive functioning (EF) refers to skills that are used to plan and execute goals. EF is described as a set of cognitive processes essential for goal-directed behavior, which include "inhibition, working memory, cognitive flexibility, planning, and problem-solving" [4], [5]. Working memory, set shifting, and inhibitory control are the components we focus on as these are skills required for computing and robotics activities to stay focused, keep track of information, and move from task to task.

Computational Thinking (CT) is a set of critical thinking and problem-solving practices used to formulate problems in ways that they can be solved with algorithms and computing [6]. While there is no generalized definition of CT that is accepted across all disciplines, there is consensus that ubiquitous computing makes CT skills valuable for everyone [7]. We use the PRADA CT framework, including pattern recognition, abstraction, decomposition, and algorithms, when integrating CT with EF as PRADA emphasizes a memorable set of components that are most applicable to interdisciplinary contexts [6].

IV. App Design

The positive learning impacts associated with educational robotics programs, including critical thinking, problem solving, interest, and engagement with STEM, have encouraged a proliferation of robotics hardware and software tools supporting education [8]–[14]. Our team's objective in designing the ORBIT App was to develop new educational robotics tools for computational thinking activities specifically designed for autistic students, through co-design with three[2] partner special education classroom teachers who are autism specialists. We wanted a customizable interface for programming LEGO® robotics kits that addressed CT goals while allowing students to work on goals that are commonly included on their Individual Education Plans. The App design and objectives were highly focused on the needs of our co-design partners. We worked together for the past eighteen months on the development of the App across three co-design workshops, focused on sharing teaching experiences and technology preferences, developing a low-fidelity paper prototype, and testing high-fidelity prototypes Figma [3], respectively.

[2]Two teachers participated since the outset of the project in all workshops. The third joined in 2025 and participated in both an App testing workshop and piloting the App with students.

A. ORBIT App Overview

The App prototype is implemented in JavaScript on a React.js framework and uses Bluetooth Low Energy to communicate with the LEGO® Spike™ Prime Hub and is available open-source at https://orbit-code.web.app/ [15]. The App interface design has two primary modalities, the guided Mission mode (Fig. 1) and the open Sandbox mode (Fig. 2). In both modes, the interface is divided into three primary areas: the Steps Menu, the Coding Track, and the Command Panel.

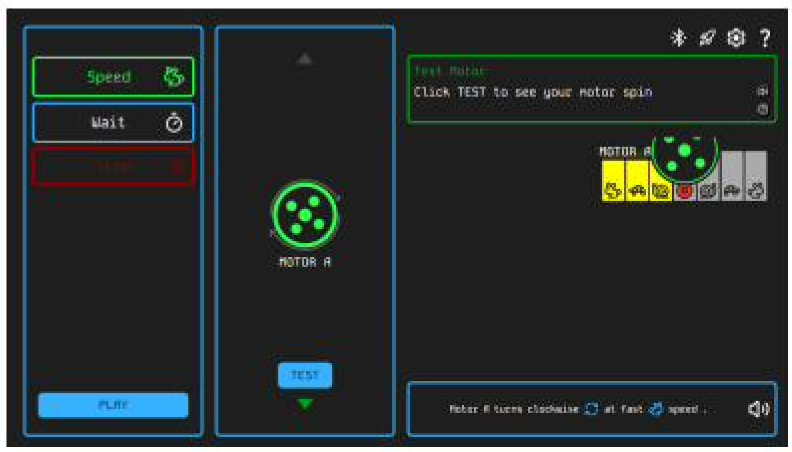

Fig. 1. The ORBIT App in Mission mode with the *Steps Menu*, *Coding Track*, and *Command Panel* shown (from left to right) in a neon theme. The Mission Mode provides structured scenarios with specific sub-tasks specified in the top-right of the app. The highly scaffolded scenario enables interface simplification, granular feedback, and hints to be provided during completion.

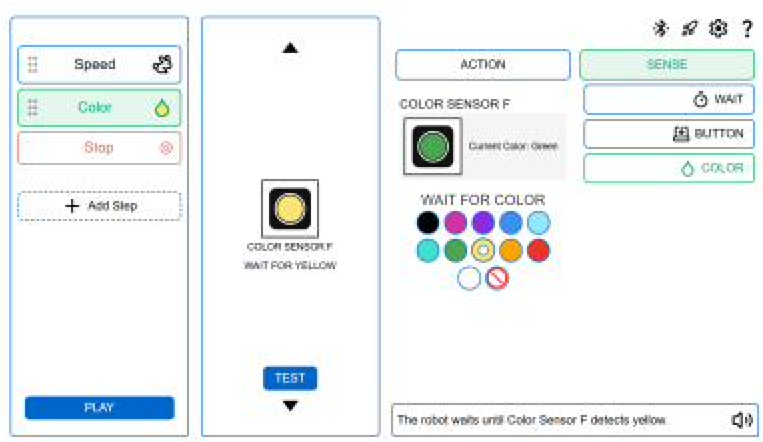

Fig. 2. The ORBIT App in Sandbox mode with the *Steps Menu*, *Coding Track*, and *Command Panel* shown (from left to right) in a high-contrast white theme.

In the **Steps Menu**, students are shown an overview of robot code as a sequence of steps and can run this full sequence.

In the **Coding Track**, students are shown a detailed view of the instruction associated with the currently selected step. The **Command Panel** is used to add or modify instructions in the currently selected code step. The Command Panel is presented as a choice between Actions or Sense, which leads to a selection of particular output or sensor types. Each step is set as a single output or sensor type, and a control dashboard for the chosen component type is shown. Below, we highlight five core features of the present app design.

1) Aligning to Prior Student Knowledge & Preparing for Future Success: While consistency with expectations and prior knowledge are core tenets in both interface design and instructional design, this was raised as a critical need during our co-design sessions [16], [17]. In particular, maintaining consistency with known interfaces and representations is supportive of student working memory. For example, during early co-design, we introduced a novel algorithm representation that presented a "code-as-pathway" analogy that was illustrated as going forward into the screen, using a vanishing point perspective. However, this was raised as an immediate concern due to misalignment with life skill goals like completing an English checklist from top to bottom. By maintaining consistency with prior experience and across tools (e.g. a coding Planning Board [3]), the App design has the potential to serve as an opportunity to be a bridge between coding, computational thinking, and executive functioning.

2) Customization for a Diverse Classroom: From the first co-design session with teachers, where we asked them to share their favorite educational software tools, customization controls emerged as a cross-cutting theme. Desired customization included levels of support for emergent readers, task complexity, reward structures, and visuals. With this in mind, the App is designed to support high levels of individualization, both for the teacher in structuring activities and for the student to tailor the tool to their desires. These individualization options currently include visual settings, e.g., color themes for a variety of contrast levels; audio settings, e.g., sound effects and text-to-speech voices; and instructional settings, e.g., the number of code steps and reading level.

3) Cultivating Sustained Engagement: Across our co-design work teachers named that successful technologies facilitate student learning by supporting sustained engagement. To support *inhibition control*, the App features an immersive, game-like experience, providing rewards in the form of amusing, playful, or surprising interactions, which helps teachers ensure that their students can complete their robot activities while also being fun for the students. The ORBIT App prototype provides this support in the Mission mode through the immersive theme and consistent animation and sound rewards as each step of the mission is completed. When an entire Mission is concluded, animated confetti fills the screen, and a bouncing rocket ship directs students to the next Mission. The Sandbox mode provides less overt engagement support through a few simple sound effects and animations; all of these can be toggled off for students who, for instance, may experience audio-visual sensory overload.

4) Supporting Guided Decision-Making: In response to teachers' input about EF and how students learn new tasks and engage with new applications, we designed the App to scaffold guided, step-by-step decision-making. Our goal was to support working memory and inhibitory control while reducing the cognitive load students feel while doing CT. One such decision-making scaffold involves collapsing all possible options for a given function. For example, sensor options only show once the sense option has been selected. By reducing the clutter, there are fewer options for students to balance in their working memory. This also helps with *inhibition control* since it guides students to suitable actions.

5) Scaffold Abstraction: Assisting users in making connections between the software and a real-world system through visual analogies and feedback is another common trait in user interface design and especially robotics [11], [16]. Teacher feedback has been crucial in helping to guide the types of information representations, visual analogies, and feedback is helpful for their classes. The App presents information in multimodal and multifaceted ways since different students may find certain formats more helpful in making that connection. This is akin to the CT concept of *Abstraction*, where, for example, the student must make a connection between an abstract representation of the robot motor and their desired motion of the real-world motor. In this context, abstraction is also directly related to EF *Cognitive Flexibility*. We provide stepping stones for cognitive flexibility and abstraction by showing actual animations of a motor-like diagram moving on the screen, a text-to-speech reader that says what the robot will be doing, and descriptive text paired with icons. Students are able to refer to those representations that they find most helpful [2]. These representations not only serve as abstraction stepping stones but might also help to introduce additional representations, such as written text for emergent readers.

V. METHODS

Data for this study come from two primary sources: audio/video of a workshop with three K-12 autism specialist teachers piloting and providing feedback on the App; and memos and audio/video from in-classroom implementation of the App with autistic middle school students.

The workshop spanned three hours and included time for solving open-ended tasks, exploring the App, and providing feedback. For feedback, teachers were first asked for general impressions, then more specifically about what they felt was working and barriers they anticipate students encountering. Two of the participating teachers have been involved with the ORBIT project for close to two years, while the third teacher was new to the project and to robotics education more broadly.

In-classroom lessons were taught by undergraduate outreach learning fellows who have been teaching robotics to these students once a week for the school year. At the time classroom data was collected, students had done some LEGO® programming using the SPIKE Prime icon blocks. The App's programming tasks were structured around Missions, as described in the App Design section, to support students learning the new user interface. In addition to audio and video data of consented students, a researcher observed the class and memoed at five-minute intervals, recording student interactions with the App with the explicit goal of addressing the research questions (pertaining to individualized learning goals and computational thinking).

A. Analytic methods

In this analysis we share preliminary evidence of students integrated executive functioning and computational thinking learning.

Our first research question is "What types of resources can be provided by our program to enable students to meet their individualized learning goals while practicing CT successfully?" In workshop data, we seek evidence for individualized learning goals in teachers' expressions of App features which support objectives they have for students outside of ORBIT. This could include concrete skills such as learning new vocabulary, as well as broader life skills such as increased independence in accomplishing tasks. In classroom data, we seek evidence for individualized learning goals in moments where students use skills or techniques learned in their classroom outside of engineering lessons. Since our work is grounded in a conception of students' executive functioning skills, we pay special attention to stated objectives and observed moments related to EF skills, such as improving or scaffolding working memory, inhibitory control, and task switching.

Our second research question is "What opportunities does participation in the program's activities provide for students to engage in computational thinking practices?" In classroom data and workshop data, we seek evidence of CT using the PRADA framework, identifying moments when students illustrate the PRADA skills (Table I) in their work on the App, and when teachers name PRADA skills in their feedback on the App.

TABLE I
PRADA Components for CT, adapted from [6]

Pattern Recognition	Observing and identifying patterns, trends, and regularities in data, processes, or problems
Abstraction	Identifying the general principles and properties that are important and relevant to the problem
Decomposition	Breaking down data, processes, and problems into meaningful smaller, manageable parts
Algorithms	Developing step by step instructions for solving a problem

VI. Preliminary Findings

A. Supporting Individualized Learning Goals

We identified two broad themes for how the ORBIT App can help students to achieve individualized learning goals: the first related to how the App can leverage familiar knowledge and integrate into existing IEP structures to support executive functioning; the second involved an orientation towards supporting students' independence by aligning with what they might expect to see in "real world" settings.

1) Integrating with familiar knowledge and structures to support executive functioning skills: In the workshop teachers identified multiple ways that the App's structure aligns with EF scaffolds familiar to students. For example, they named that the consistent ordering of coding sequences from top to bottom, both in the App and in the companion Planning Board, aligns with tasks familiar to students such as reading and completing checklists. Likewise, teachers appreciated the implementation of accessibility features in the App, such as coupled text with icons, an audio text function (with multiple voice speeds and modulation), and multiple color themes with dyslexia-friendly fonts. They suggested a number of potential improvements that could leverage other familiar structures, such as using PECs icons [18] which are used broadly in special education settings, and providing a visual impairment color scheme with specific color contrasts.

One detailed conversation emerged over the use of vocabulary, particularly around the "Wait" command that is common to many coding environments, including LEGO® SPIKE™ and Scratch™. The teachers felt that the word wait would mean to "stop what you're doing" to their students. Saying "wait" aloud illuminated this discrepancy: the meaning of words may feel different when *heard* than when *read*. Since we are programming the App with an audio text function, any vocabulary we use in the App must be examined both visually and audibly for clarity of meaning. This may mean replacing a command like "Wait" with something like "Hold".

In the first classroom lesson, the observing researcher noted two students' use of "earning" timers. While working, the two students each had a one-minute timer, and for each minute they worked, they would earn a sticker on a sheet; when they filled up the sheet, they would have a pre-set amount of time doing another preferred task, such as drawing or using a tablet. These timers are meant to scaffold students' inhibitory control and planning by providing structured incentive to stay on task. The researcher observed that over the course of the coding lesson, one of the students became less attentive to his earning timer, indicating that the directed feedback from the App was supporting his engagement and reducing the need for the EF scaffold. It also became clear that a similar structure could be built into a later version of the App to enable teachers to leverage and modulate existing scaffolds for their students.

2) Consistency with what students may need to use in the real world – independence motivated: Teachers' design suggestions were also motivated by reflecting "real world" skills. In the workshop, this emerged as a justification for not overscaffolding the App experience when added challenges could support concrete skill development. One discussion centered on the *complexity* of coding logic, particularly where abstracting and decomposing instructions can feel more conceptually challenging than having an integrated command that does everything. For instance, when running a motor, setting all the relevant parameters (speed, direction, duration) in one command can be easier to understand than setting them across multiple commands. However, this overspecification does not reflect the modularity often observed in programming (e.g.

what if we wanted a motor to stop with a sensor rather than a duration of time?). Teachers wanted to reduce or scaffold complexity as much as possible, but through discussion a paradigm emerged whereby complexity could be justified when it aligned with what students would see in a more advanced environment.

B. Supporting Computational Thinking with the ORBIT App

We observed two clear design considerations related to PRADA skills—one regarding Decomposition, and the other Abstraction & Algorithms.

In solving open-ended, learning prompts using the ORBIT App, one teacher expressed a perceived trade-off between two important life skills—following directions (as one might do with instructions in a LEGO kit), and breaking down problems for oneself. Both of these skills are valuable for their students' independence beyond schooling, and they felt a strong need to support Decomposition as a life skill, which they saw happening with the implementation of robotics in this App. In the classroom we observed students Decomposing story prompts into programs through the use of the Planning Board. This tool supported students to generate step-by-step instructions that they could translate into the App and upload to their robots.

The aforementioned discussion on coding complexity highlighted the value of Abstracting commands and learning Algorithms to sequence instructions. Teachers explored how multiple motors could be set in the App to wait for different start/stop triggers, and expressed how this abstraction and algorithmic thinking could extend to other life-skills for their students. In the classroom, we observed students' demonstrating Abstraction skills through differentiating between Action and Sense commands, and developing Algorithmic thinking through constructing short, repeatable command sequences to accomplish desired robot actions.

VII. IMPLICATIONS AND FUTURE DIRECTIONS

Across our co-design we encountered balancing a "low floor" (making an app easier to access) with a "high ceiling" (making an app have flexible and advanced functionality), prompting explicit consideration of what felt the most important to prioritize for students. This reflects a common tension special education teachers face between scaffolding executive functioning to create space for learning outcomes, and supporting students' growth in executive functioning skills to improve independence. ORBIT App design strategies must dynamically balance broad accessibility with individualized supports.

We continue to iterate App changes and refinements, based on feedback from teachers and the analysis of classroom data of students using the program. The next iteration of the ORBIT App will be based on student data from ongoing pilot testing in the classroom during the school's Extended School Year program, along with learning progressions developed through a final workshop with partner teachers. Planned improvements include further reducing the number of options to improve guided decision-making by reducing visual "clutter", and refining Missions to scaffold the initial student experience in the ORBIT App.

Alongside the ORBIT App we continue to refine the Analog Learning Tools, such as the planning board, and implement them with students in the school's Extended Year Program. The aim of these tools is to support students' executive functioning while enabling students to meet their individual learning goals.

REFERENCES

[1] A. Taboas, K. Doepke, and C. Zimmerman, "Preferences for identity-first versus person-first language in a US sample of autism stakeholders," *Autism*, vol. 27, no. 2, pp. 565–570, feb 2023. [Online]. Available: https://journals.sagepub.com/doi/10.1177/13623613221130845

[2] E. Hartmann, "Universal Design for Learning (UDL) and Learners with Severe Support Needs," *International Journal of Whole Schooling*, vol. 11, no. 1, pp. 54–67, 2015.

[3] R. Hayes, E. Milto, and J. Cross, "Considerations from co-design with special education teachers: Integrating computational thinking with executive functioning skills for autistic middle school students," in *The American Society for Engineering Education Annual Conference*, Montreal, Ca, 2025.

[4] A. Diamond, "Executive functions," *Annual Review of Psychology*, vol. 64, pp. 135–168, 2013.

[5] A. Miyake, N. P. Friedman, M. J. Emerson, A. H. Witzki, A. Howerter, and T. D. Wager, "The unity and diversity of executive functions and their contributions to complex "frontal lobe" tasks: a latent variable analysis," *Cogn. Psychol.*, vol. 41, no. 1, pp. 49–100, Aug. 2000.

[6] Y. Dong, V. Catete, R. Jocius, N. Lytle, T. Barnes, J. Albert, D. Joshi, R. Robinson, and A. Andrews, "Prada: A practical model for integrating computational thinking in k-12 education," in *Proceedings of the 50th ACM technical symposium on computer science education*, 2019, pp. 906–912.

[7] V. J. Shute, C. Sun, and J. Asbell-Clarke, "Demystifying computational thinking," *Educational research review*, vol. 22, pp. 142–158, 2017.

[8] S. Anwar, N. A. Bascou, M. Menekse, and A. Kardgar, "A systematic review of studies on educational robotics," *Journal of Pre-College Engineering Education Research (J-PEER)*, vol. 9, no. 2, p. 2, 2019.

[9] F. Mondada, M. Bonani, F. Riedo, M. Briod, L. Pereyre, P. Rétornaz, and S. Magnenat, "Bringing robotics to formal education: The thymio open-source hardware robot," *IEEE Robotics & Automation Magazine*, vol. 24, no. 1, pp. 77–85, 2017.

[10] E. Hamner, L. Zito, J. L. Cross, M. Tasota, P. Dille, S. Fulton, M. Johnson, I. Nourbakhsh, and J. Schapiro, "Development and results from user testing of a novel robotics kit supporting systems engineering for elementary-aged students," in *2017 IEEE Frontiers in Education Conference (FIE)*. IEEE, 2017, pp. 1–9.

[11] J. Cross, C. Bartley, E. Hamner, and I. Nourbakhsh, "A visual robot-programming environment for multidisciplinary education," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 445–452.

[12] I. M. Souza, W. L. Andrade, L. M. Sampaio, and A. L. S. O. Araujo, "A systematic review on the use of lego® robotics in education," in *2018 IEEE frontiers in education conference (FIE)*. IEEE, 2018, pp. 1–9.

[13] B. Erwin, M. Cyr, and C. Rogers, "Lego engineer and robolab: Teaching engineering with labview from kindergarten to graduate school," *International Journal of Engineering Education*, vol. 16, no. 3, pp. 181–192, 2000.

[14] Z. Zhang, S. Willner-Giwerc, J. Sinapov, J. Cross, and C. Rogers, "An interactive robot platform for introducing reinforcement learning to k-12 students," in *Robotics in Education: RiE 2021 12*. Springer, 2022, pp. 288–301.

[15] LEGO® Education. (2024) Spike™ prime protocol. Date retrieved April 2025. [Online]. Available: https://lego.github.io/spike-prime-docs/

[16] R. Molich and J. Nielsen, "Improving a human-computer dialogue," *Communications of the ACM*, vol. 33, no. 3, pp. 338–348, 1990.

[17] G. P. Wiggins and J. McTighe, *Understanding by design*. Ascd, 2005.

[18] A. Bondy and L. Frost, "The picture exchange communication system," *Behav. Modif.*, vol. 25, no. 5, pp. 725–744, Oct. 2001.