

Best Practices to Reach all Learners In Our STEM Classrooms

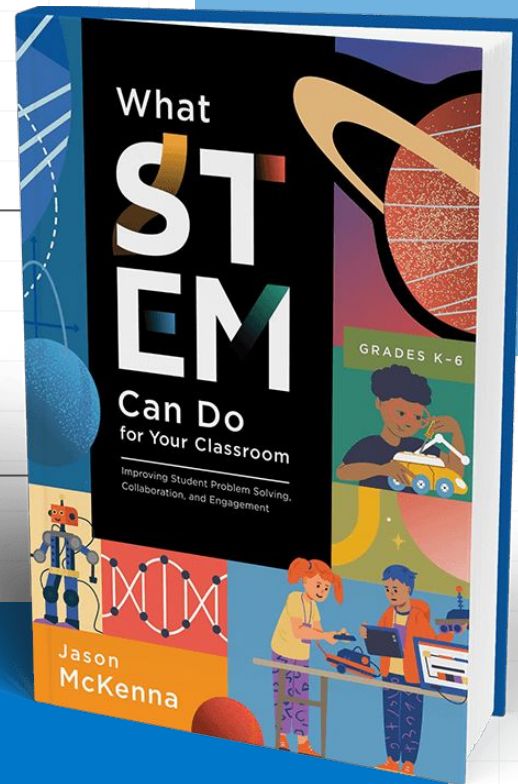
Jason McKenna

Jason McKenna

VP of Global Educational Strategy,
VEX Robotics, Top 100 Education Influencers

What STEM Can Do for Your Classroom:
*Improving Student Problem Solving,
Collaboration, and Engagement*

X@McKennaj72
jason@vex.com



Accessibility is the Law



Nondiscrimination on the Basis of Disability; Accessibility of Web Information and Services of State and Local Government Entities ([link](#))

- Implements a regulation based on Title II of Americans with Disabilities Act (ADA)
- Applies to state and local government entities including public schools
- Addresses web and mobile accessibility issues

Details

General. A public entity shall ensure that the following are readily accessible to and usable by individuals with disabilities:

- (1) Web content that a public entity provides or makes available, directly or through contractual, licensing, or other arrangements; and
- (2) Mobile apps that a public entity provides or makes available, directly or through contractual, licensing, or other arrangements.

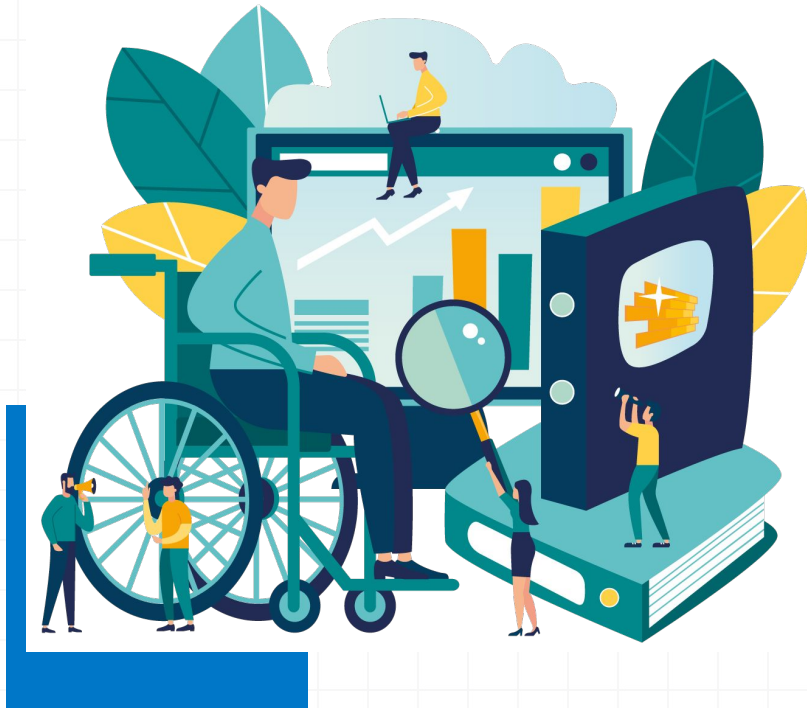
Level of compliance: WCAG 2.1 AA

Dates of compliance:

Governmental entities larger than 50,000: April 24, 2026

Governmental entities smaller than 50,000: April 27, 2027

Section 508



In 1998, Congress amended the Rehabilitation Act of 1973 to require federal agencies to make their electronic and information technology (EIT) accessible to people with disabilities.


The law 29 U.S.C § 794d applies to all federal agencies when they develop, procure, maintain, or use electronic and information technology.

Section 508

Under Section 508, agencies must give disabled employees and members of the public access to information comparable to the access available to others.

Educational software is included under Section 508 guidelines.

Several states have passed laws requiring 508 at the state level. California, Missouri, New York, Oklahoma, Washington.



**When we prioritize
accessibility, we create a
better learning experience
for every student.**

Examples

Voice-to-text messaging

Originally designed for individuals with disabilities, is now widely used by millions for convenience.



Teacher PD

Resources

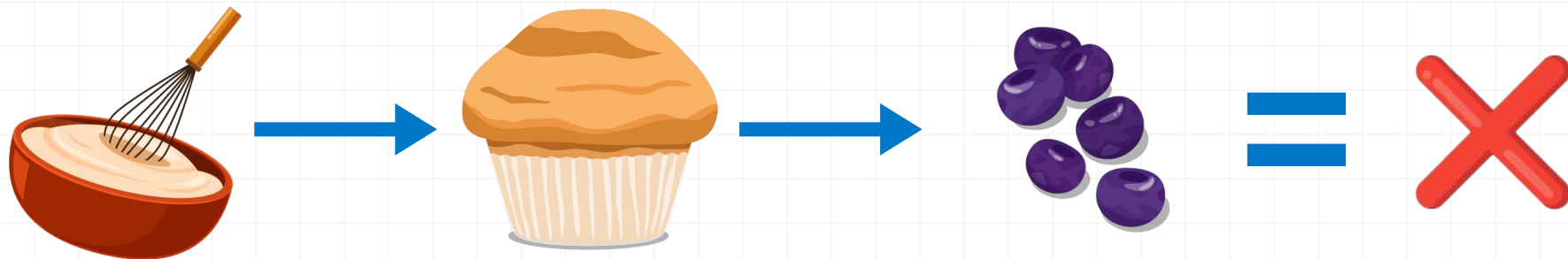
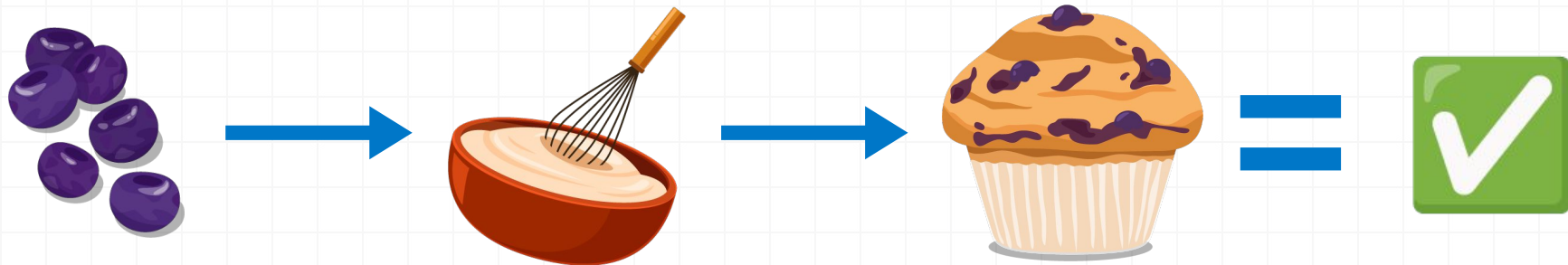
Curriculum

Instructional Leadership

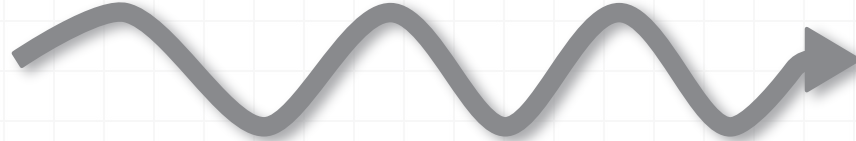
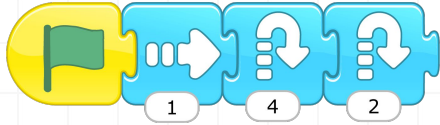
The Real Problem

An iceberg is centered in the image, divided horizontally by a line representing the water surface. The top of the iceberg, which is above the water, is light blue and has four labels: 'Teacher PD' at the top left, 'Resources' at the top right, 'Curriculum' on the left side, and 'Instructional Leadership' on the right side. The bottom of the iceberg, which is submerged in the water, is a darker blue and contains the text 'The Real Problem' in white. The background is a light blue grid pattern.

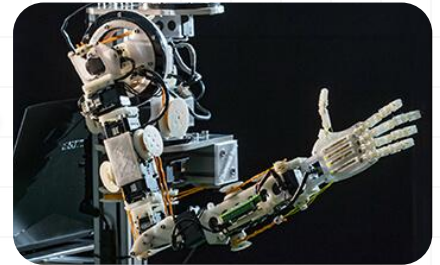
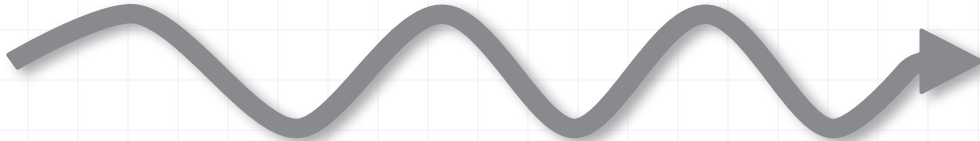
Blueberry Muffin



K

2nd5th8th12thCarnegie
Mellon
University

```
public void squirrel() {  
    Turtle turtle = new Turtle();  
    int dist = 0;  
    while (dist < 100) {  
        turtle.forward(dist);  
        turtle.right(90);  
        dist = dist + 10;  
    }  
    turtle.say("A Squirrel!");  
}
```



Special Education Teachers Evaluating the Accessibility of CS Educational Robotics

Andrew Bennett
University of Florida
bennettandrew@ufl.edu

Maya Israel
University of Florida
misaia@coe.ufl.edu

Joanne Barrett
University of Florida
jbarrett@coe.ufl.edu

Debra "Kelly" Thomas
Broward County, FL Public Schools
kelly.thomas@browardschools.com

Jason McKenna
VEX Robotics
jason_mckenna@innovationfirst.com

ABSTRACT

All students benefit when computer science (CS) materials are accessible, but it is critical for students with disabilities. In order to provide opportunities for all students to be successful, it is important for teachers to be able to evaluate the accessibility of their lessons and technology. One way to evaluate accessibility is the POUR framework. The POUR framework represents what can be Perceived through the senses, how users can Operate a material or technology, how it is Understandable to users, and the overall Robustness. POUR provides a promising way for K-12 CS teachers to evaluate accessibility for their learners. We describe how the POUR framework was used by a cohort of teachers to evaluate VEX 123 for their learners with disabilities. Findings from the teacher POUR analysis revealed that overall, the teachers noted that the VEX 123 provided the necessary range of entryways into coding through its three modalities: The touch coding on the robot itself, the code cards, and VECODE (the block-based coding environment). At the same time, the teachers indicated that some students with disabilities faced a number of motor and sensory difficulties. Overall, this study showcased a way for teachers to provide insight into the level of accessibility of CS education tools specific to their students' strengths and needs.

KEYWORDS

Teacher Education, Computer Science Education, Teachers' Identity, Teachers' Values

ACM Reference Format:

Andrew Bennett, Maya Israel, Joanne Barrett, Debra "Kelly" Thomas, and Jason McKenna. 2024. Special Education Teachers Evaluating the Accessibility of CS Educational Robotics. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 2 (SIGCSE 2024)*, March 20–23, 2024, Portland, OR, USA. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3606253.3635576>

1 INTRODUCTION

One way CS education can be supported is through the use of accessible materials. It has been suggested that accessibility can be

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner(s).
SIGCSE 2024, March 20–23, 2024, Portland, OR, USA
© 2024 Copyright held by the owner(s).
ACM ISBN 979-8-4007-0424-0/24/03.
<https://doi.org/10.1145/3606253.3635576>

SIGCSE 2024, March 20–23, 2024, Portland, OR, USA

Andrew Bennett, Maya Israel, Joanne Barrett, Debra "Kelly" Thomas, & Jason McKenna

poster presents the findings of the teachers' POUR analysis of the VEX 123.

2 TEACHER POUR ANALYSIS OF ROBOTICS

Data that was used as part of our understanding of how teachers used the POUR framework were: (1) Notes from meetings with the teachers as they described their experiences with the VEX123, and (2) materials that teachers created within their professional development course management system (CMS). These sources of

better maximized if accessibility is considered at the start of the design or evaluation process, not as an afterthought [8]. However, with many CS education tools, this is not always possible as these were not designed with accessibility in mind [7].

Teachers are essential to understanding how educational products have been used or not used in the classroom as they routinely interact with the technologies, use them with students, and evaluate its usage [1]. Hence, partnering with teachers to provide feedback on the accessibility of CS education tools and robotics can help to ensure that CS instruction is inclusive of all students.

One way for teachers to evaluate accessibility is through using the Perceivable, Operable, Understandable, and Robust (POUR) guidelines [4]. POUR was created to streamline accessibility guidelines for websites [5]. It now can be used to evaluate accessibility, including educational materials [9]. Yet, little research has applied the POUR principles to CS education.

We picked VEX 123 as a product to examine due to commitment from VEX to make their technologies more accessible. Additionally, educational robotics are important for increasing student learning, motivation, and sense of well-being [2]. VEX 123 is a hybrid educational robot aimed at pre-kindergarten through early elementary students; generally it is an entry point of CS education for the youngest learners [7]. VEX 123 is distinguished by its three ways to program: directly manipulating touch buttons on the device, tangible coding through a blue-tooth enabled code and code card system, and programming done on the VEXCODE online coding platform [10].



Figure 1: Three ways of coding with VEX 123

We created a professional development module for special education teachers on using the POUR framework to evaluate the CS education tools that they use with learners with disabilities. This module is part of a larger project funded by Google aimed at wide-scale professional development focused on computer science inclusion and accessibility. As part of the module, the teachers were asked to examine a technology they used in the classroom as well as taking part in a group discussion of VEX 123 and POUR. This

3 DISCUSSION

There is growing research suggesting that educational robotics can support learners' understanding of computational thinking in engaging ways [7]. However, many of these studies have not focused on the inclusion of learners with disabilities [6], [7]. This study demonstrates POUR as a potential framework to address accessibility.

Although there are ways we could improve the accessibility module presented to the teachers, this study added to ways teachers can evaluate educational technologies used to teach computer science.

One way for teachers to evaluate accessibility is through using the Perceivable, Operable, Understandable, and Robust (POUR) guidelines. POUR was created to streamline accessibility guidelines for websites. It now can be used to evaluate accessibility, including educational materials.



inaccessible

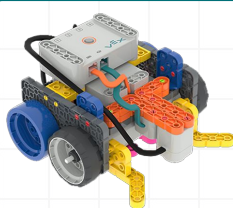
Solutions

The VEX Continuum



VEX 123
Coding Starts Early

Ages 4+



VEX GO
STEM Starts Early

Ages 8+



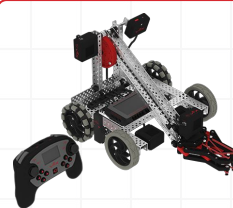
VEX IQ
Applied STEM
Learning

Ages 11+



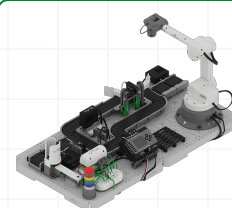
VEX EXP
Real World STEM
for Classrooms

Ages 14+



VEX V5
Real World STEM
for Competition

Ages 14+



VEX CTE
Workforce
Readiness

Ages 14+



VEX AIR
STEM Skills
Take Flight


Ages 14+

VEX CODE VR
Virtual Robot Coding

Ages 8+



The VEX Computer Science (CS) Continuum



VEX 123
Coding Starts Early

Ages 4+

The VEX 123 interface features a colorful block-based coding environment. On the left, there are several code blocks: 'say happy', 'if object', 'wait 1 second', 'go on green', and 'drive until crash'. In the center is a blue circular track with a white robot icon. On the right is a smartphone displaying a sequence of code blocks: 'when start 123', 'drive until object', 'play honk', 'turn random', and 'go to start'.



VEX AIM
Real World Coding

Ages 8+

The VEX AIM components include a purple lightbulb icon, a blue and orange ball, a white VEX controller, a black VEX motor, and a yellow and black soccer ball.



VEX CTE
Workforce
Readiness

Ages 14+

The VEX CTE features a white robotic arm mounted on a black base, with various sensors and components visible.

VEX CODE VR
Virtual Robot Coding

Ages 8+



Designing a Progression of Programming Environments to Support K-12 Learners as they Advance

Yuhan Lin
University of Maryland
United States
jimmylin@umd.edu

David Weintrop
University of Maryland
United States
weintrop@umd.edu

Jason McKenna
VEX Robotics
United States
jason@vex.com

Abstract: There are many ways that schools can support students in their computer science learning journeys as they move from grade to grade, but those pathways always include dramatic shifts in programming environments and languages. With the goal of addressing the current challenges in transitioning learners from one programming environment to the next, this paper presents the VEX continuum, which starts with a hands-on introduction to fundamental computer science concepts, such as sequencing, and progresses through stages, including block-based programming, to text-based programming and eventually to a professional integrated development environment. In this experience report, we present the VEX Continuum, a progression of programming environments designed to support learners as they move from introductory to advanced computer science education. In introducing the suite of VEX environments, we highlight how it is designed to scaffold learners as they move from environment to environment, increasing complexity and expressive power as they progress through their computer science education journey.

1. Introduction

The last decade has seen an increased emphasis on the integration of digital literacy and computational thinking skills into K-12 educational frameworks. As an important piece of this new educational emphasis, computer science education has a critical role to play in K-12 education to prepare learners to thrive in an increasingly technology world. As it is inevitable that they need to have some computer science knowledge and computational thinking to be successful in many fields (Yadav et al., 2011). Despite the importance of computer science education, current pedagogical approaches have struggled to provide a coherent, continuous learning pathway for learners as they progress from introductory tools in kindergarten through more powerful programming languages in high school. The result is a lack of continuity and at-times difficult transitions for learners. These transition points, and the difficulties learners face with them, are consequential as it is at the points that many learners choose to end their study of computer science, particularly learners from populations historically excluded from the field (Kölling et al., 2015; Lin & Weintrop, 2021; Weintrop et al., 2020).

Many educational tools and platforms, such as coding robots (Yu & Roque, 2018) and programming environments (Lin & Weintrop, 2021), have been developed to enrich the computer science education experience and make it more accessible and engaging (Malizia et al., 2017). These tools can facilitate the development of critical thinking, problem-solving, and algorithmic thinking skills, making abstract concepts more tangible for learners (Grover et al., 2017). However, the educational ecosystem still faces a significant challenge. Each tool and platform often exist as an isolated entity with its own set of design features, supported interactions, and capabilities that must be learned alongside the computing content. The transition from one tool or platform to the next is often not clearly defined or supported, which can disrupt the learning progression, overwhelm learners, and cause them to lose confidence or interest (Lin & Weintrop, 2021).

In response to these challenges, we iteratively developed the VEX continuum, a cohesive series of programming tools and environments designed to provide a seamless trajectory from introductory to advanced computer science courses. The VEX continuum addresses the need for a structured and comprehensive set of learning environments to support a K-12 learning pathway, where each stage evolves from the previous one and prepares learners for the next. This continuum incorporates a series of programming approaches and environments, beginning with Touch Button programming, and then continuing to Coder and Coder cards, then block-based programming, Switch mode, and eventually transition to text-based programming, each catering to a different learning level and computational skill set.

This paper presents a detailed overview of the VEX continuum, its development, and its contribution to the K-12 computer science education landscape. By providing a detailed reflection on the strengths and weaknesses of

“The VEX Continuum is structured to provide scaffolds across the progression from the lowest threshold entry points to the most advanced, highest ceiling...”

Example of Scaffolds



Turtle Creek Elementary STEAM Academy - VEX

@TurtleCreek_Vex

Follow



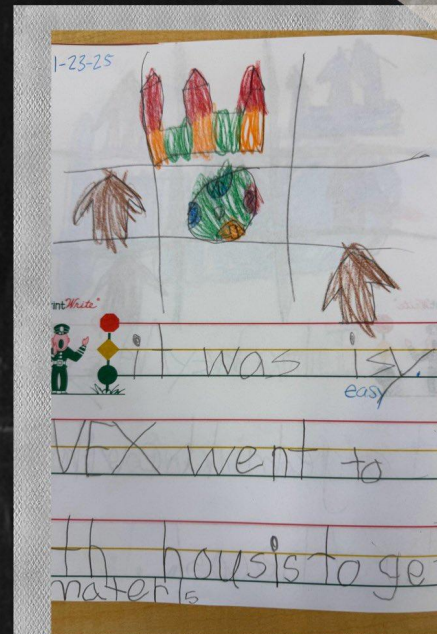
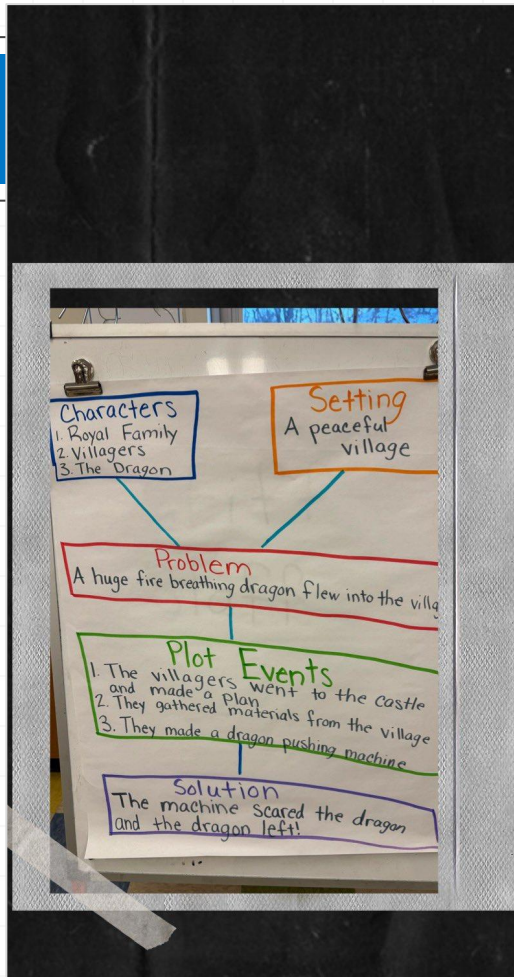
Kindergarteners in Mrs. Morse's class documented their VEX robotics activity in their VEX notebooks.

#VEX123

#CrossCurricular

Source:

https://x.com/turtlecreek_vex/status/1886557388469076108



Example of Scaffolds

Special Education Teachers Evaluating the Accessibility of CS Educational Robotics

Andrew Bennett
University of Florida
bennettandrew@@ufl.edu

Maya Israel
University of Florida
misrael@coe.ufl.edu

Joanne Barrett
University of Florida
jrbarrett@coe.ufl.edu

Debra "Kelly" Thomas
Broward County, FL Public Schools
kelly.thomas@browardschools.com

Jason McKenna
VEX Robotics
jason_mckenna@innovationfirst.com



Accessibility: VEXcode - Read Blocks

The screenshot displays the VEXcode IDE interface. At the top is a dark blue header bar with the AIM logo, a globe icon, and menu items: File, Tools, TUTORIALS, UNDO, REDO, SLOT, VEXcode Project, Not Saving, ROBOT, DOWNLOAD, RUN, STOP, SHARE, and FEEDBACK. Below the header is a light blue bar with a 'Code' tab and icons for settings, navigation, and help.

On the left is a vertical palette of category icons: Motion (selected), Emoji, Kicker, Sound, LED, Macro, AI Vision, Screen, and Controller. The 'Motion' category is expanded, showing a list of 'Motion - Actions' blocks:

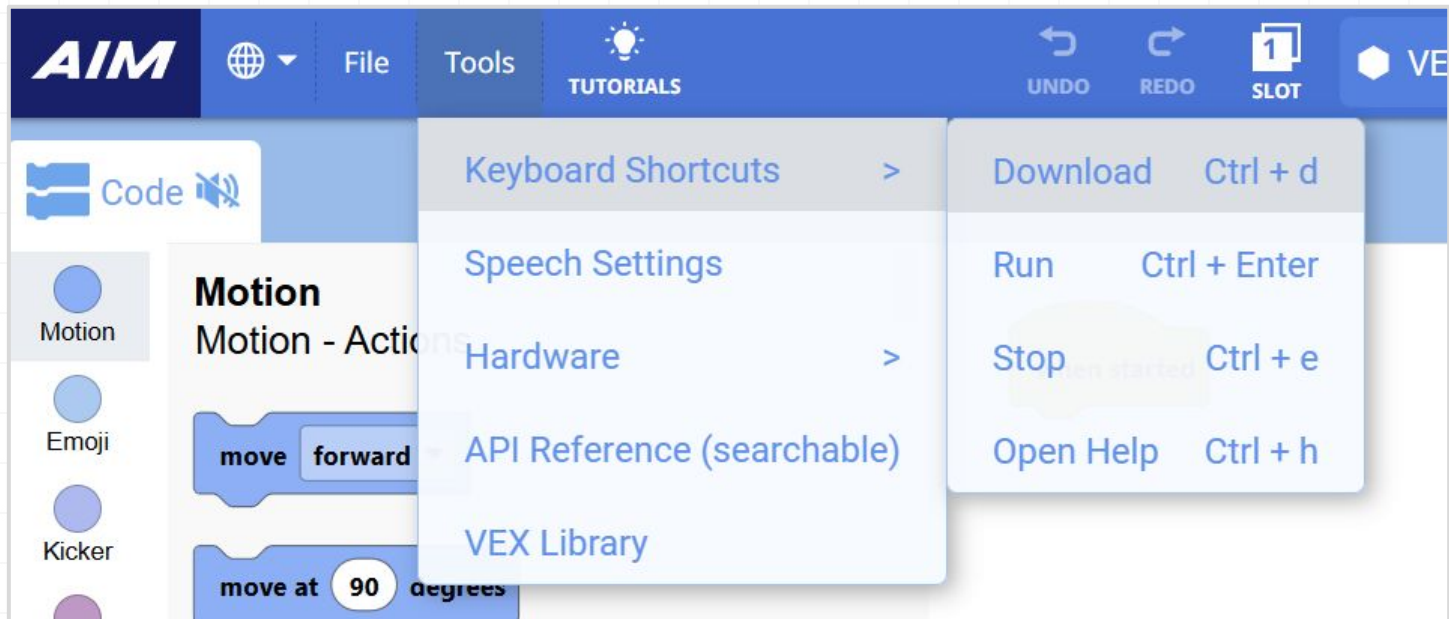
- move forward ▾
- move at 90 degrees
- move forward ▾ for 200 mm ▾ ▶
- move at 45 degrees for 200 mm ▾ ▶
- turn right ▾
- turn right ▾ for 90 degrees ▶
- turn to heading 270 degrees ▶

The main workspace on the right contains a sequence of three blocks:

- A yellow 'when started' block.
- A blue 'move forward ▾ for 200 mm ▾ ▶' block.
- A blue 'turn right ▾ for 90 degrees ▶' block.

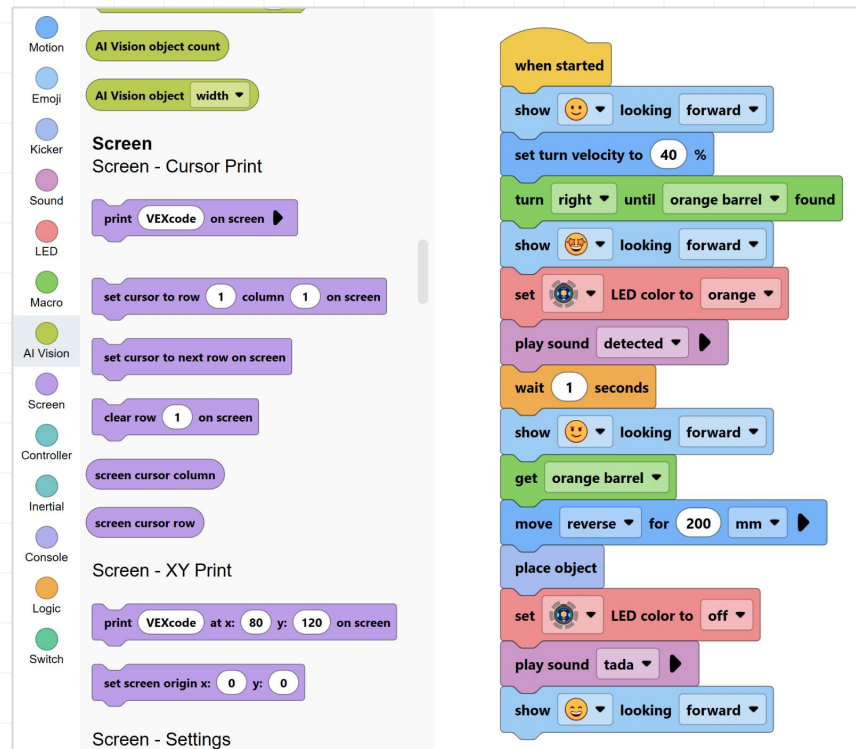
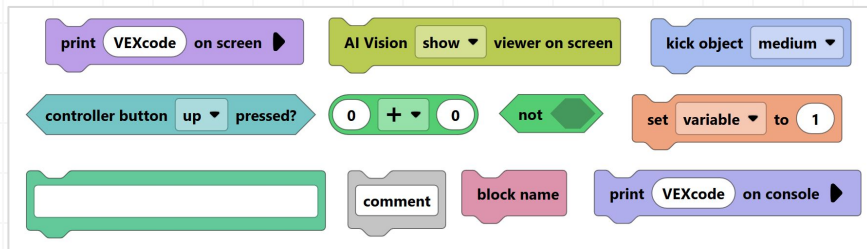
On the bottom right of the workspace, there are three circular icons: a magnifying glass with a plus sign, a magnifying glass with a minus sign, and an equals sign.

Accessibility: VEXcode - Keyboard Shortcuts

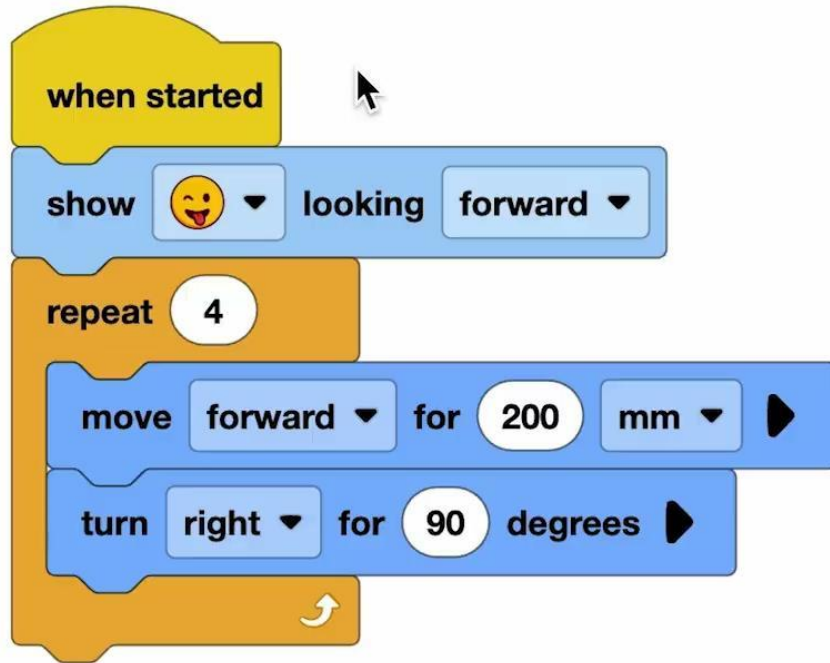


VEXcode AIM: High Contrast Blocks

High contrast blocks built into
VEXcode AIM



VEXcode Switch Blocks



Accessibility: VEX PD+

- Screen reader
- Navigate with keyboard

Chapter 1 Heading 2 Building the 6-Axis Arm

4 lessons



☒ Introduction

"Introduction to Chapter 1"



☐ Lesson 1: Introduction to Workcells

☐ Lesson 2: Building the CTF 6-Axis Arm

Chapter 2 Operating the Teach Pendant

3 lessons



☒ Introduction

"25% of Chapter 2 completed."



☐ Lesson 1: Accessing VEXcode EXP

☐ Lesson 2: Using the Teach Pendant

VR

4 Chapters | 5 Hours

This VEXcode VR training course will guide you through learning how to implement VEXcode VR in a variety of different subjects such as art, history, math, science and language arts. You will learn about different Playgrounds in VEXcode VR and different cross curricular activities.

Introduction

- Introduction Lesson: Welcome to VEX Masterclasses

Chapter 1

3 lessons

Incorporating VEXcode VR into Art Classrooms



Chapter 2

4 lessons

Incorporating VEXcode VR into Language Arts and Social Studies Classrooms



- ☐ Lesson 1 - Customizing Resources for Use with VEXcode VR Art Canvas +
- ☐ Lesson 2 - Ideas for Incorporating VEXcode VR into Language Arts Classrooms
- ☐ Lesson 3 - Ideas for Incorporating VEXcode VR into Social Studies Classrooms
- ☐ Lesson 4 - Cross-Curricular Connections with VEXcode VR - Week 2

Chapter 3

2 lessons

Incorporating VEXcode VR into Science Classrooms




Chapter 4

2 lessons

Incorporating VEXcode VR into Math Classrooms



Cross Curricular Connections with VEXcode VR Certification Exam

Certification Locked due to incomplete Chapters. Please complete all chapters before attempting this Exam again. 

Accessibility: VEX API

- Navigate with keyboard
- Language options
- Python code copy
- Light / Dark modes
- Screen-reader Blocks



```
1 # Wait until note is finished to move
2 robot.sound.play_note("C6", 1000)
3 while robot.sound.is_active():
4     wait(50, MSEC)
5     robot.turn_to(180)
```

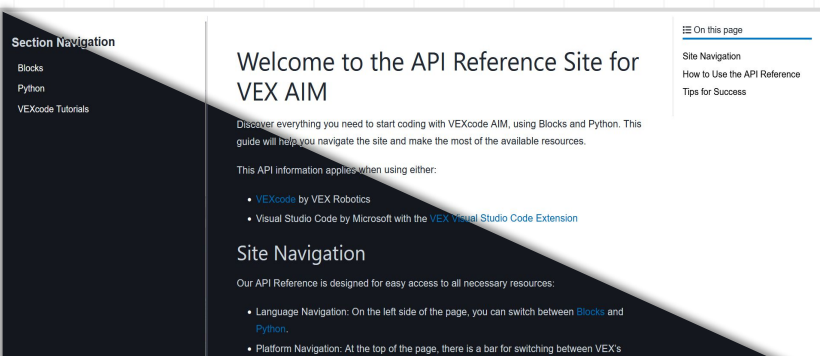
Copy

when started

Kick an object with full force.

kick object

hard ▼



Section Navigation

Blocks

Motion

Emoji

Kicker

Sound

LED

Macro

AI Vision

Screen

Controller

Inertial

Timer

Console

Logic

Switch

Python

VEXcode Tutorials

VEX API & VEXcode AIM

The screenshot displays the VEXcode AIM software interface. The top menu bar includes 'AIM', 'File', 'Tools', 'TUTORIALS', 'UNDO', 'REDO', 'SLOT', 'VEXcode Project', 'Not Saving', 'ROBOT', 'DOWNLOAD', 'RUN', 'STOP', 'SHARE', and 'FEEDBACK'. The left sidebar contains a 'Code' tab and a palette of blocks categorized by function: Motion, Emoji, Kicker, Sound, LED, Macro, AI Vision (selected), Screen, Controller, Inertial, Console, Logic, and Switch. The 'AI Vision' category is expanded, showing sub-sections: 'AI Vision - Actions' (with 'AI Vision show viewer on screen'), 'AI Vision - Settings' (with 'set AI Vision object item to 1'), 'AI Vision - Values' (with 'has sports ball ?', 'AI Vision object exists?', 'AI Vision object is sports ball ?', 'AI Vision object is AprilTag 1 ?', 'AI Vision object count', and 'AI Vision object width'), and 'Screen' (with 'Screen - Cursor Print' and 'print VEXcode on screen'). A central workspace shows a single 'AI Vision show viewer on screen' block. The right sidebar, titled 'Help', features a search bar and a list of 'Actions'. The 'AI Vision viewer' action is highlighted, with a description: 'The AI Vision viewer block enables or disables the live AI Vision feed on the robot's screen. When enabled, the screen displays real-time sensor data, preventing other images or text from appearing. To display other content, use this block to hide the feed.' Below the description is a table of parameters:

Parameters	Description
status	Controls the display of the data feed on the robot's screen: <ul style="list-style-type: none">show - Displays the data feed.hide - Removes the data feed from the screen.

An 'Example' section shows a 'when started' block followed by a block that says 'View the AI Vision sensor's feed for five seconds'.

Accessibility: Education Website - Descriptive Video

vex

Special Education Teachers Evaluating the Accessibility of CS Educational Robotics

Andrew Bennett
University of Florida
bennettandrew@ufl.edu

Maya Israel
University of Florida
misrael@coe.ufl.edu

Joanne Barrett
University of Florida
jrbarrett@coe.ufl.edu

Debra "Kelly" Thomas
Broward County, FL Public Schools
kelly.thomas@browardschools.com

Jason McKenna
VEX Robotics
jason_mckenna@innovationfirst.com

ABSTRACT

All students benefit when computer science (CS) materials are accessible, but it is critical for students with disabilities. In order to provide opportunities for all students to be successful, it is important for teachers to be able to evaluate the accessibility of their lessons and technology. One way to evaluate accessibility is the POUR framework. The POUR framework represents what can be Perceived through the senses, how users can Operate a material or technology, how it is Understandable to users, and the overall Robustness. POUR provides a promising way for K-12 CS teachers to evaluate accessibility for their learners. We describe how the POUR framework was used by a cohort of teachers to evaluate VEX 123 for their learners with disabilities. Findings from the teacher POUR analysis revealed that overall, the teachers noted that the VEX 123 provided the necessary range of entryways into coding through its three modalities: The touch coding on the robot itself, the coder cards, and VECODE (the block-based coding environment). At the same time, the teachers indicated that some students with disabilities faced a number of motor and sensory difficulties. Overall, this study showcased a way for teachers to provide insight into the level of accessibility of CS education tools specific to their students' strengths and needs.

KEYWORDS

Teacher Education, Computer Science Education, Teachers' Identity, Teachers' Values

ACM Reference Format:

Andrew Bennett, Maya Israel, Joanne Barrett, Debra "Kelly" Thomas, and Jason McKenna. 2024. Special Education Teachers Evaluating the Accessibility of CS Educational Robotics. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 2 (SIGCSE 2024)*, March 20–23, 2024, Portland, OR, USA. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3626253.3635576>

<https://research.vex.com>



better maximized if accessibility is considered at the start of the design or evaluation process, not as an afterthought [8]. However, with many CS education tools, this is not always possible as these were not designed with accessibility in mind [7].

Teachers are essential to understanding how educational products have been used or not used in the classroom as they routinely interact with the technologies, use them with students, and evaluate its usage [1]. Hence, partnering with teachers to provide feedback on the accessibility of CS education tools and robotics can help to ensure that CS instruction is inclusive of all students.

One way for teachers to evaluate accessibility is through using the Perceivable, Operable, Understandable, and Robust (POUR) guidelines [4]. POUR was created to streamline accessibility guidelines for websites [5]. It now can be used to evaluate accessibility, including educational materials [9]. Yet, little research has applied the POUR principles to CS education.

We picked VEX 123 as a product to examine due to commitment from VEX to make their technologies more accessible. Additionally, educational robotics are important for increasing student learning, motivation, and sense of well-being [2]. VEX 123 is a hybrid educational robot aimed at pre-kindergarten through early elementary students; generally it is an entry point of CS education for the youngest learners [7]. VEX 123 is distinguished by its three ways to program: directly manipulating touch buttons on the device, tangible coding through a blue-tooth enabled coder and code card system, and programming done on the VEXcode online coding platform [10].



One way for teachers to evaluate accessibility is through using the Perceivable, Operable, Understandable, and Robust (POUR) guidelines. POUR was created to streamline accessibility guidelines for websites. It now can be used to evaluate accessibility, including educational materials.

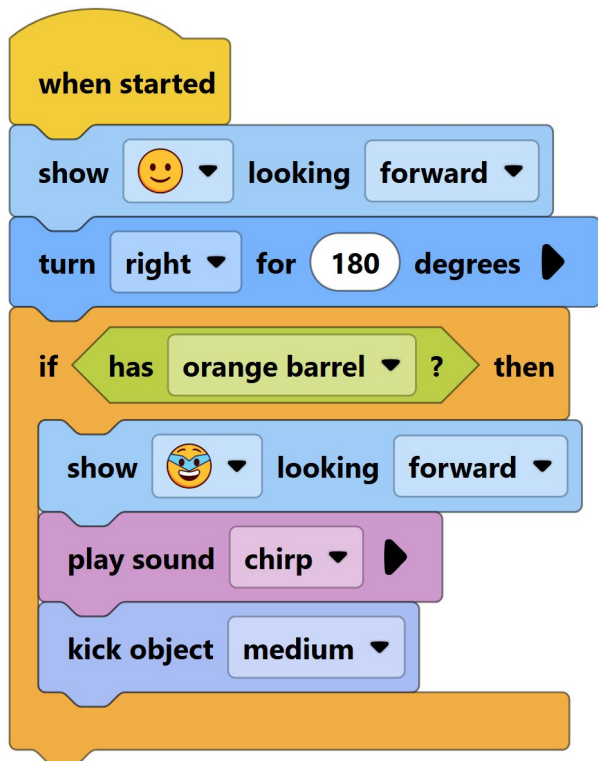
VEX AIM Driving with Controller



VEX AIM Touch Button Coding



VEXcode AIM Python Coding



<> Code



Motion



Emoji



Kicker



Sound



LED



Macro



AI Vision



Screen



Controller



Inertial



Console



Logic

Motion ▾

Actions

```
robot.move_at(0)
robot.move_for(200, 0)
robot.move_with_vectors(100, 100, 0)

robot.turn(RIGHT)
robot.turn_for(RIGHT, 90)
robot.turn_to(270)

robot.stop_all_movement()
```

Mutators

```
robot.set_move_velocity(100)
robot.set_turn_velocity(100)

robot.set_xy_position(0, 0)
```

Getters

```
robot.get_x_position()
robot.get_y_position()

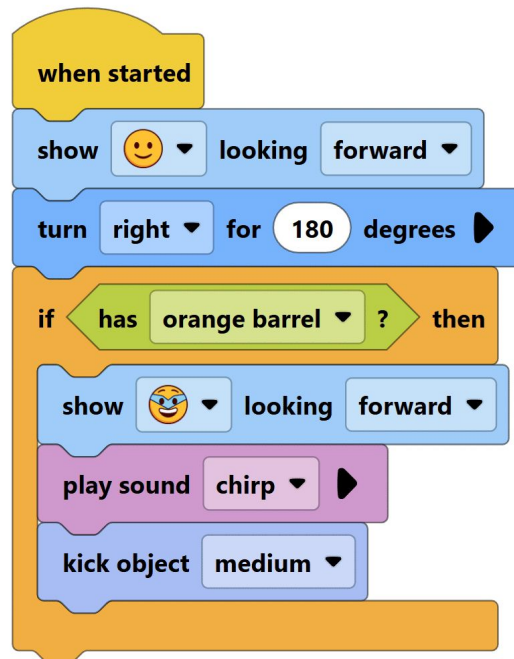
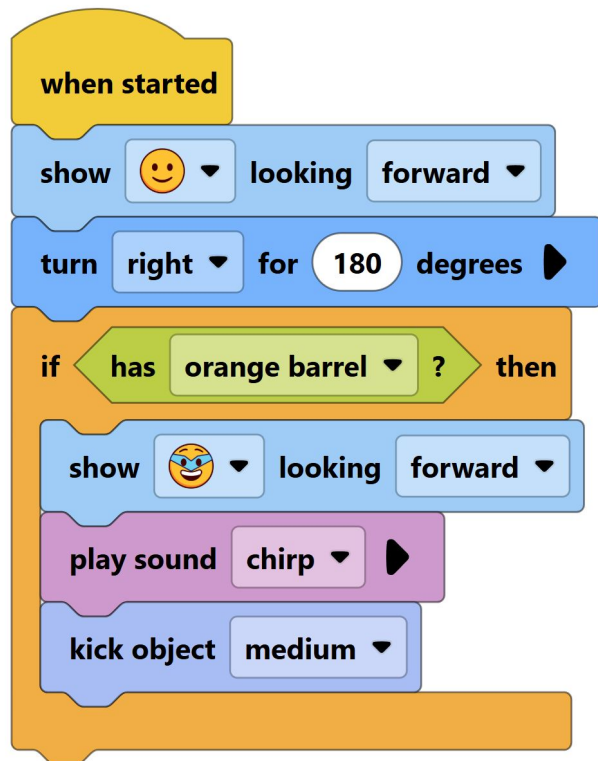
robot.is_move_active()
robot.is_turn_active()
robot.is_stopped()
```

Emoji ▾

Actions

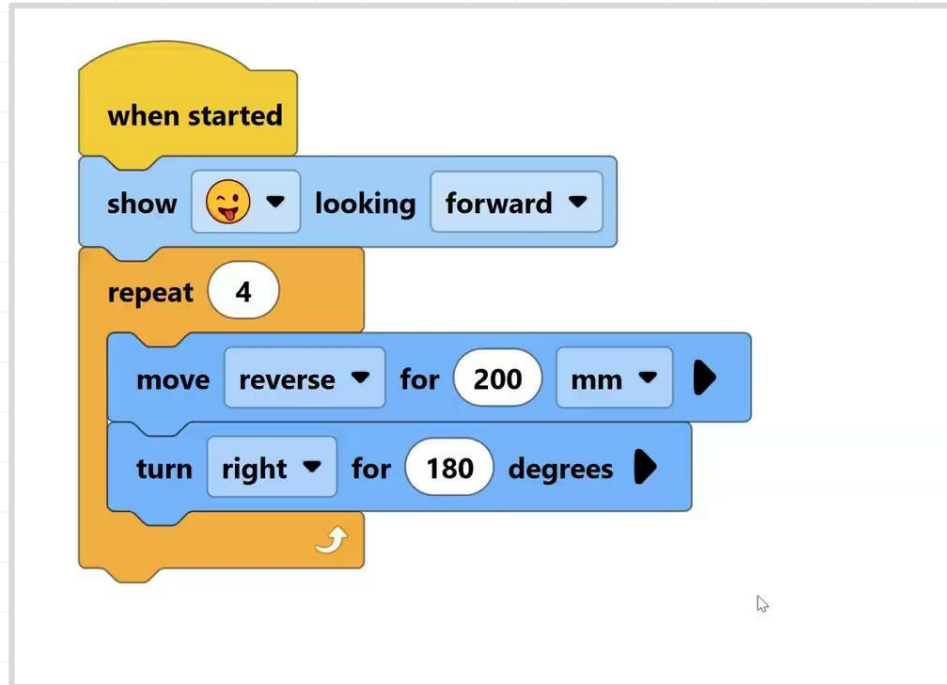
```
1 #region VEXcode Generated Robot Configuration...
102
103 robot.screen.show_emoji(HAPPY, LOOK_FORWARD)
104 robot.turn_for(RIGHT, 180)
105 if robot.has_orange_barrel():
106     robot.screen.show_emoji(STRONG, LOOK_FORWARD)
107     robot.sound.play(CHIRP)
108     while robot.sound.is_active():
109         wait(50, MSEC)
110     robot.kicker.kick(MEDIUM)
111
```

VEXcode AIM Blocks / Switch Blocks



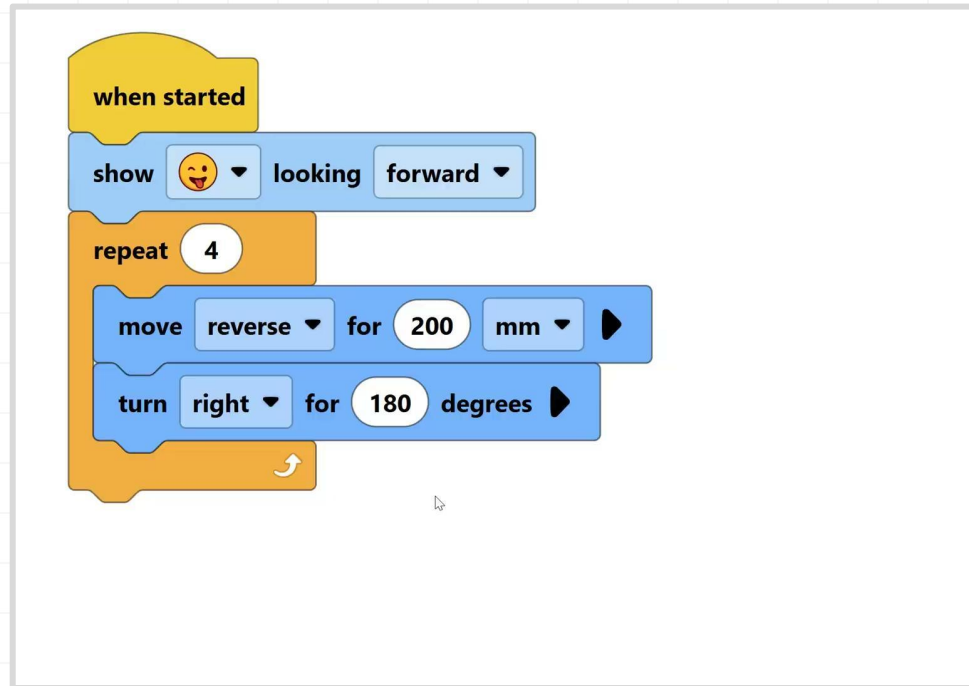
Convert Only

Learners initially compose their program using conventional blocks, including adjusting parameters, before converting the block to Switch mode.



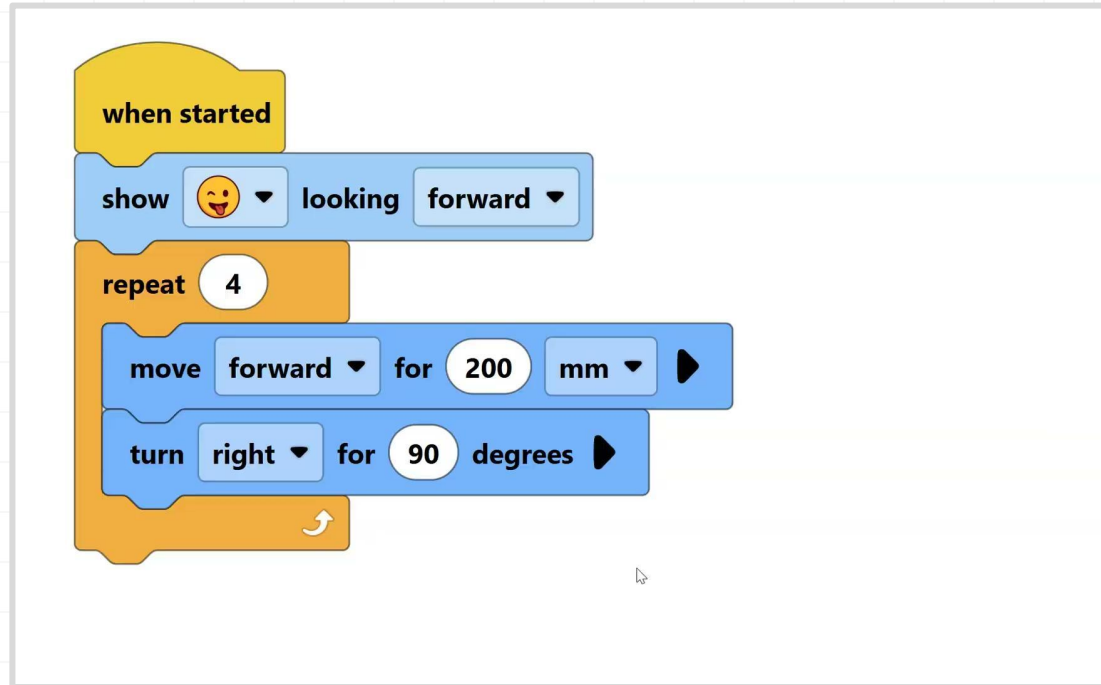
Convert and Modify Parameter

Learners compose with conventional blocks, convert to Switch mode, and then exclusively modify the parameter in Switch mode.



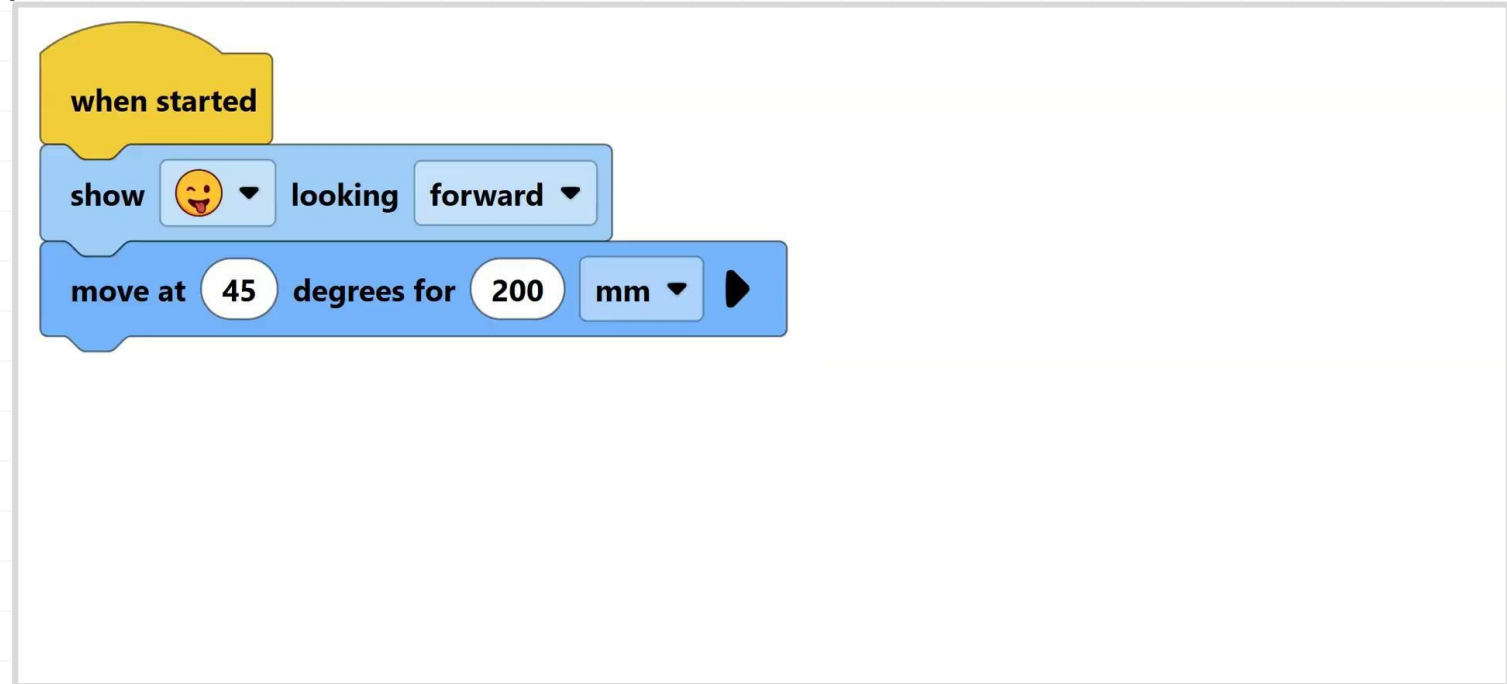
Convert then Reference

Learners compose with blocks, convert to Switch mode and then manually input what they saw in the converted Switch mode block.



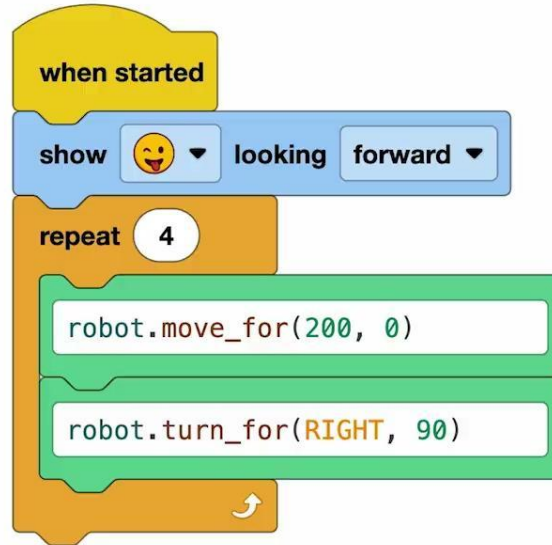
Type, Convert, then Reference

Learners drag a Switch mode block to the canvas, begin typing, then drag a regular block, convert it, and use the converted block as a reference to complete their initially half-typed command.



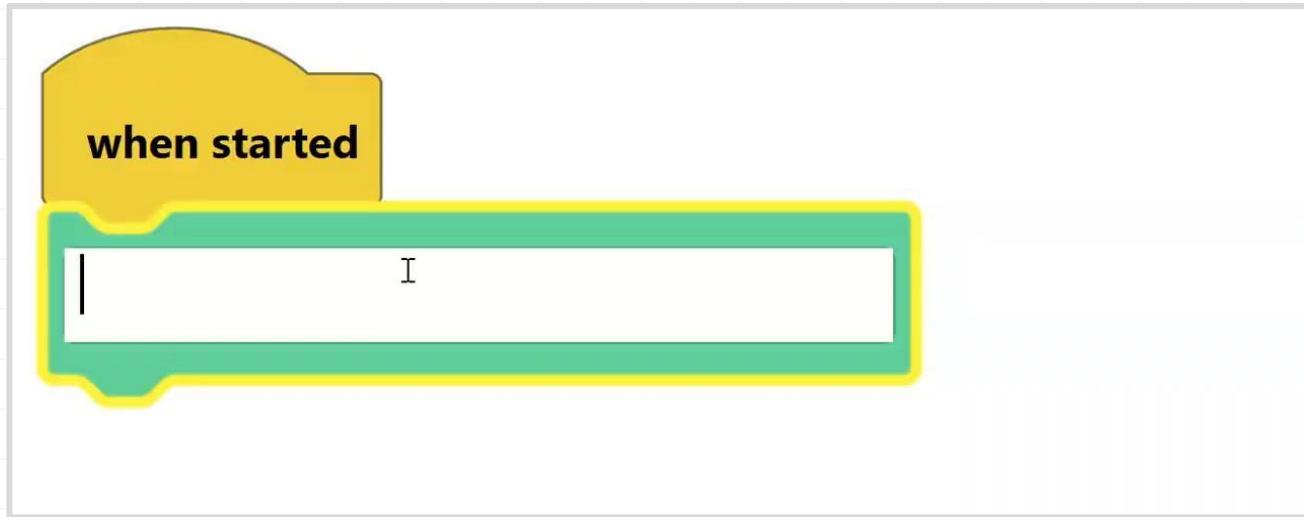
Modify in Switch mode

This involves learners making more alterations in Switch mode than just modifying the parameters.



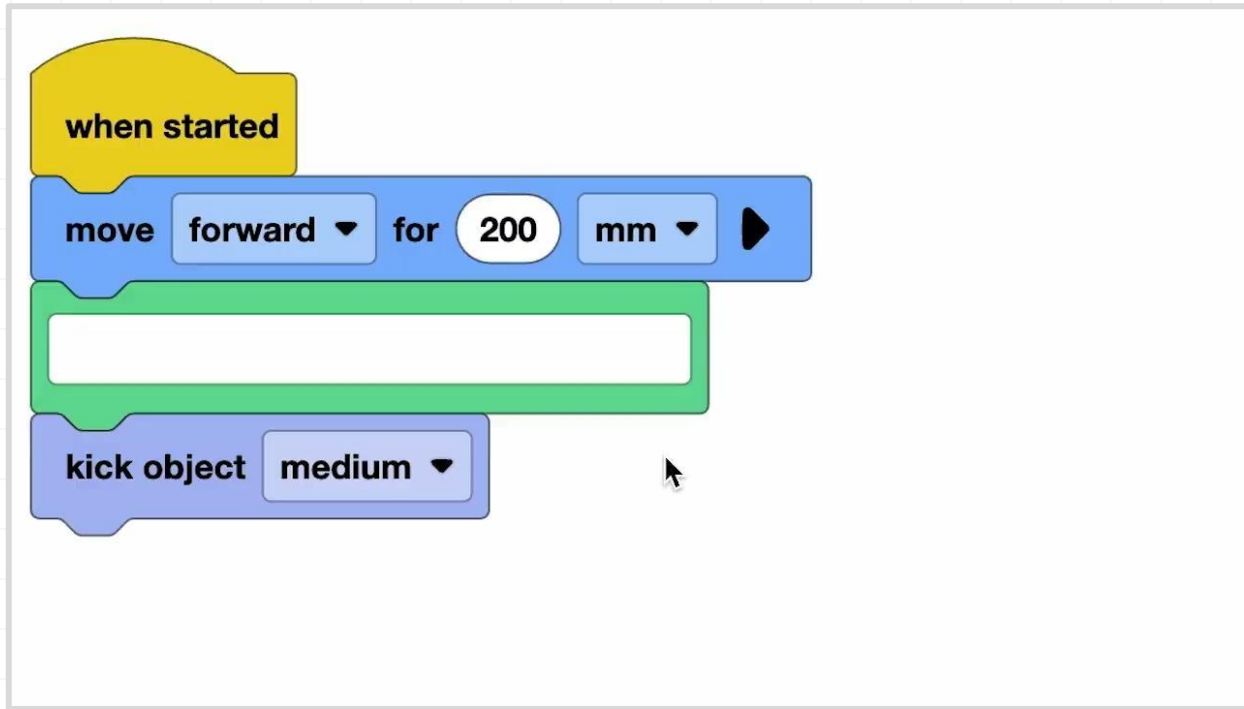
Auto-complete with Error Correction

Learners use the auto-complete feature of Switch mode to fix errors, such as correcting lowercase to uppercase typing, regardless of where these Switch mode blocks came from.



Type with Auto-complete

Learners drag a Switch mode block to the canvas and use the auto-complete feature of Switch mode to complete their Python commands.

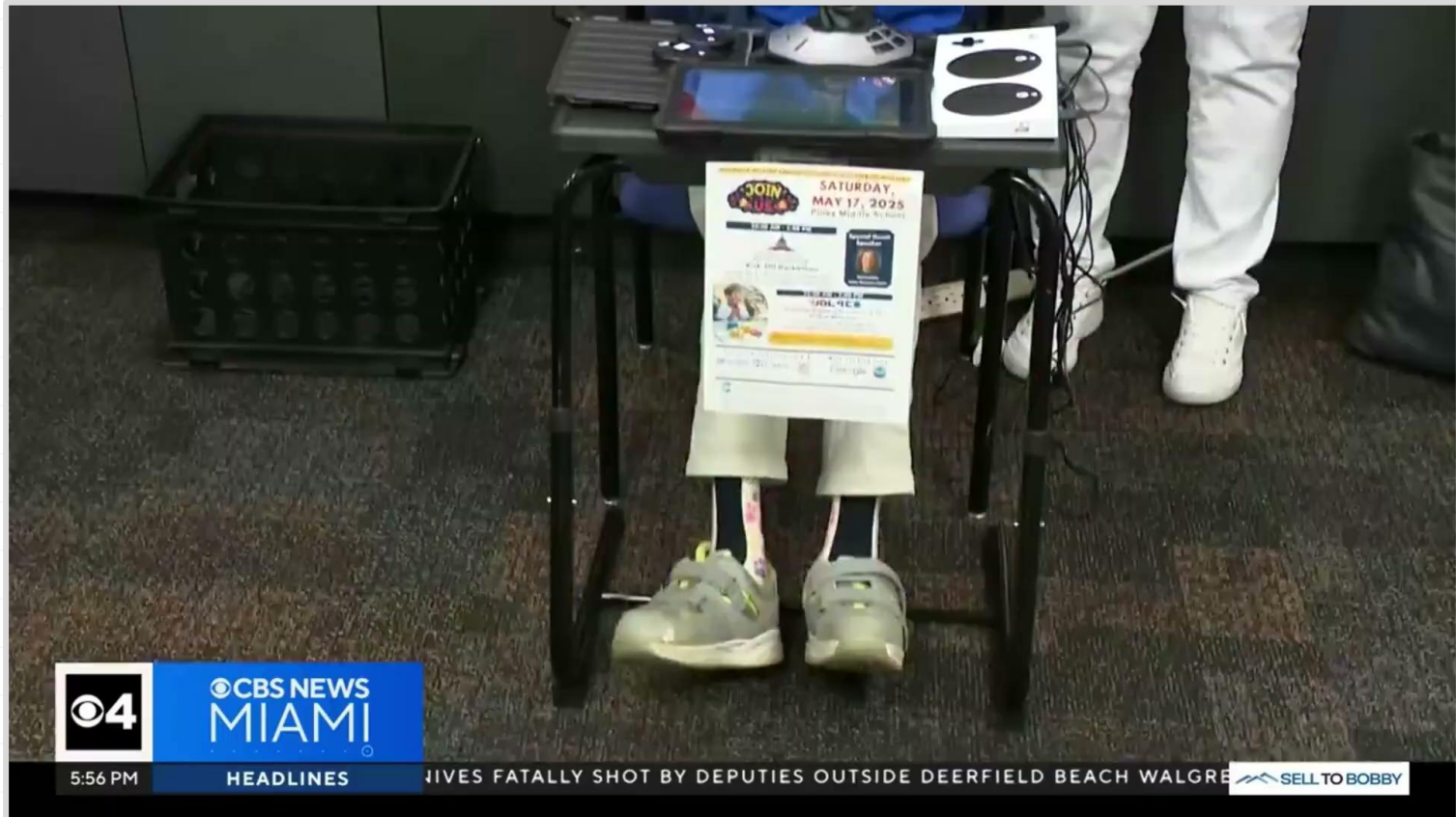


Typing without Auto-complete

Learners drag a Switch mode block to the canvas and manually type the command, deliberately not using the auto-complete feature.



Bridge App



CBS NEWS
MIAMI

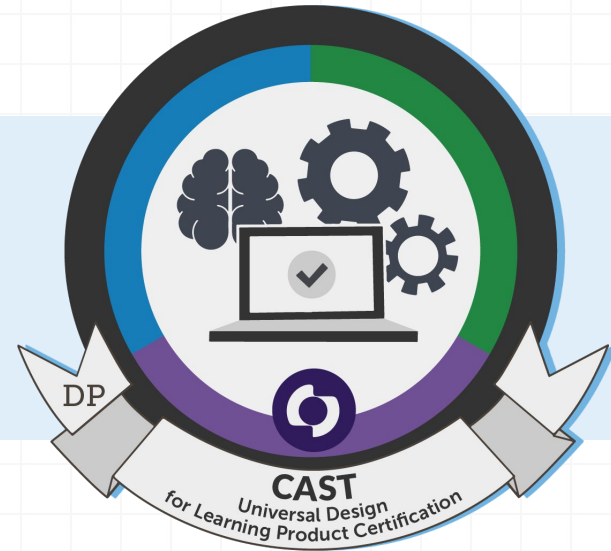
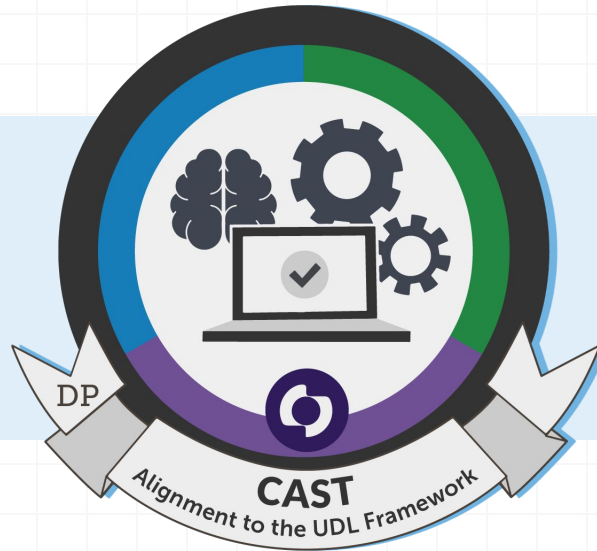
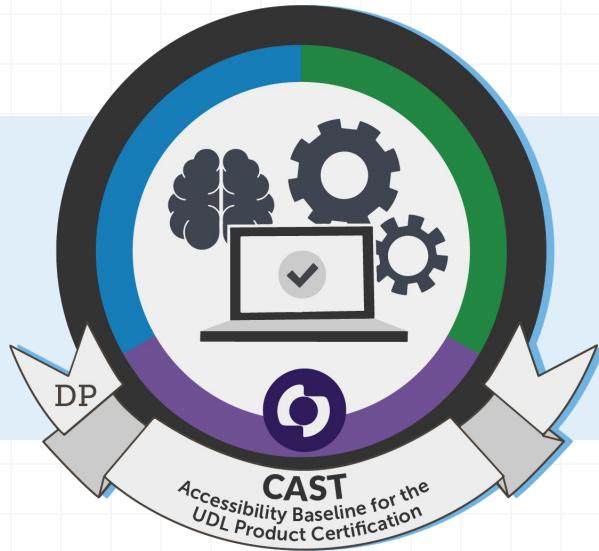
5:56 PM

HEADLINES

WIVES FATALLY SHOT BY DEPUTIES OUTSIDE DEERFIELD BEACH WALGREEN

SELL TO BOBBY

Certified by CAST





Summary:

- Begin with accessibility in mind
- Work with researchers and experts
- Attack the “real” problem

Questions?

*Chat with me and a community
of amazing educators in the
PD+ Community!*

