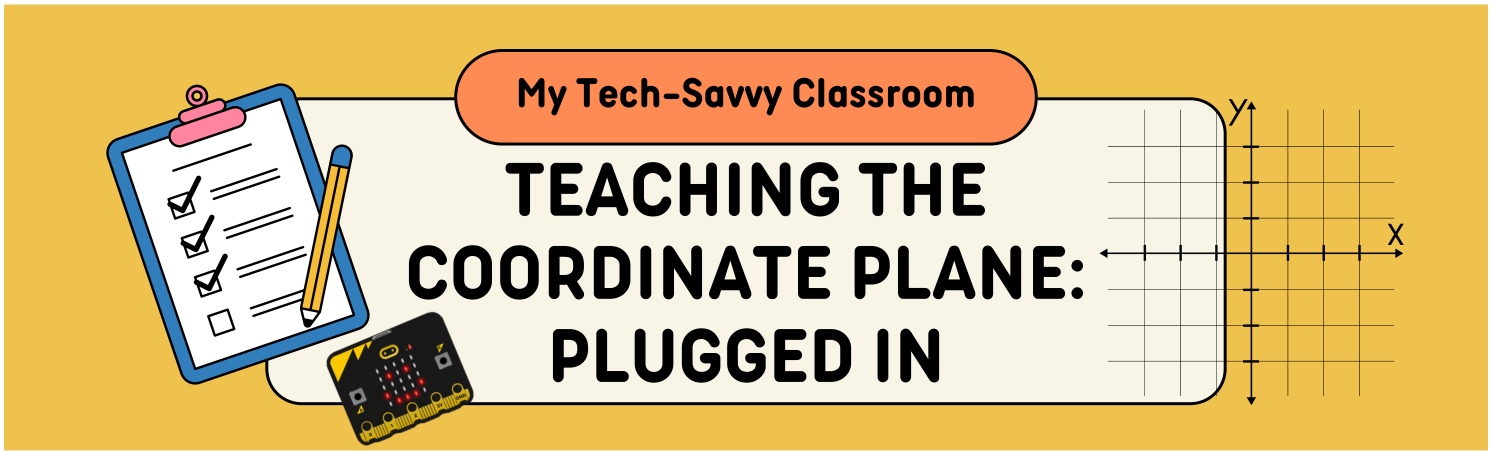


My Tech-Savvy Classroom

TEACHING THE COORDINATE PLANE: PLUGGED IN

Learning Objectives:

- Students will create and use variables to represent X and Y coordinates
- Students will write conditional statements to control the sprite's movement on the LED grid
- Students will use various inputs (tilt, button presses) to change the direction of the sprite
- Students will solve challenges by modifying their code to prevent the sprite from moving off the edge of the grid

Materials needed:

- Micro:Bit devices (or online Micro:Bit simulator on makecode.microbit.org)
- Laptops or tablets to access MakeCode
- Anchor chart or notebook with basic Micro:Bit inputs listed (optional)

Extensions and Modifications:

- If students complete the task early, introduce challenge cards.
  - These can include tasks like creating obstacles on the grid for the sprite to avoid, adding a scoring system, or building more complex boundaries for the sprite's movement.

Facilitator Notes:

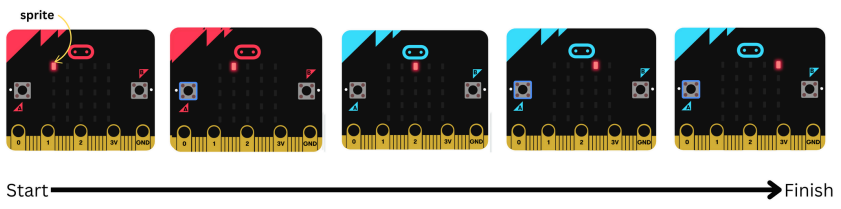
- Break down the steps clearly for students, especially when introducing new coding blocks or concepts
- Use visuals like an anchor chart or live coding demonstration to reinforce key points
- Monitor student progress by circulating and offering individual support where needed. Some students may need more time to grasp variables and conditionals
- Encourage problem-solving by giving students time to try coding on their own and providing hints instead of solutions when they encounter challenges

---

Dot Mover Game: Educator Guide:

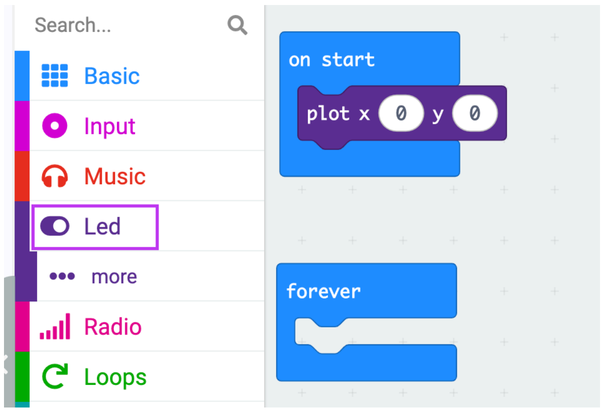
Part 1: Creating a Sprite and Variables
Begin by explaining the challenge: moving a sprite across the Micro:Bit's 5x5 LED grid without letting it fall off the edge. Introduce the "on start" and "forever" blocks in MakeCode, highlighting their functions. "On start" executes code once when the program starts, while "forever" executes code continuously throughout the program.

As you click the A-Button, the dot moves right



1. Creating a sprite:

Guide students to create their sprite using the "plot x y" block. Explain that the sprite's position is controlled by X and Y coordinates. Ask, "What happens when I change the X value? The Y value?" This will demonstrate how changing X moves the sprite left or right and changing Y moves it up or down.

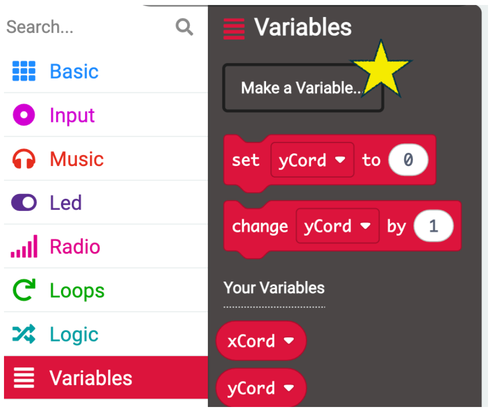


```
on start
    plot x 0 y 0

forever
```

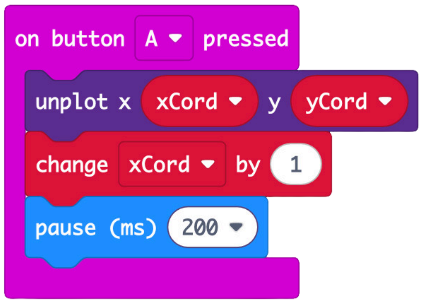
2. Introducing variables:

Discuss what variables are and why they are used in coding. For example, in the equation "x + 5 = 7," x is the variable, representing a value that can change. In coding, variables store and retrieve data. For this activity, students will create variables to store the X and Y coordinates of their sprite. Walk them through creating variables in MakeCode, such as "xCord" and "yCord."

Created for Edutopia by Sabrina Tirachen (Of *My Tech-Savvy Classroom*)



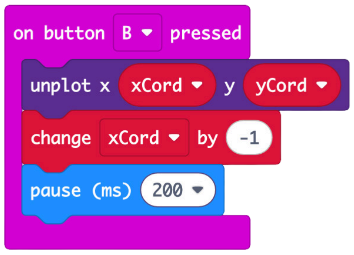
3. Moving the sprite right (modeling):
Explain the use of conditional statements, which direct the program to execute an action based on a condition being true or false. Guide students to write code that moves the sprite to the right when button A is pressed. Use the block "on button A pressed" and unplot the sprite's current coordinates. Then, change the value of "xCord" by 1 to move the sprite right. Discuss why it's necessary to unplot the sprite first, since the "forever" loop constantly re-plots the sprite unless it's removed first. The code to move the sprite right should look like this:



```
on button A pressed
    unplot x xCord y yCord
    change xCord by 1
    pause (ms) 200
```

4. Moving the sprite left (student challenge):

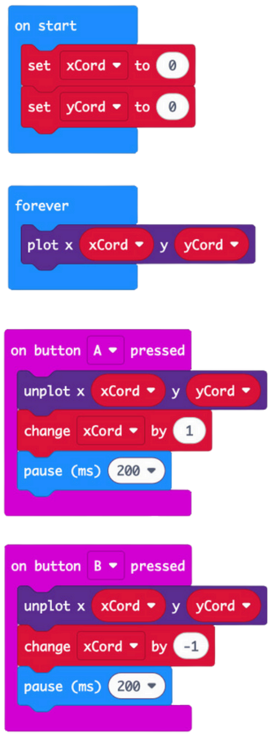
As a check for understanding, have students program "on button B pressed" to move the sprite to the left. The code to move the sprite left should look like this:



```
on button B pressed
    unplot x xCord y yCord
    change xCord by -1
    pause (ms) 200
```

Created for Edutopia by Sabrina Tirachen (Of *My Tech-Savvy Classroom*)

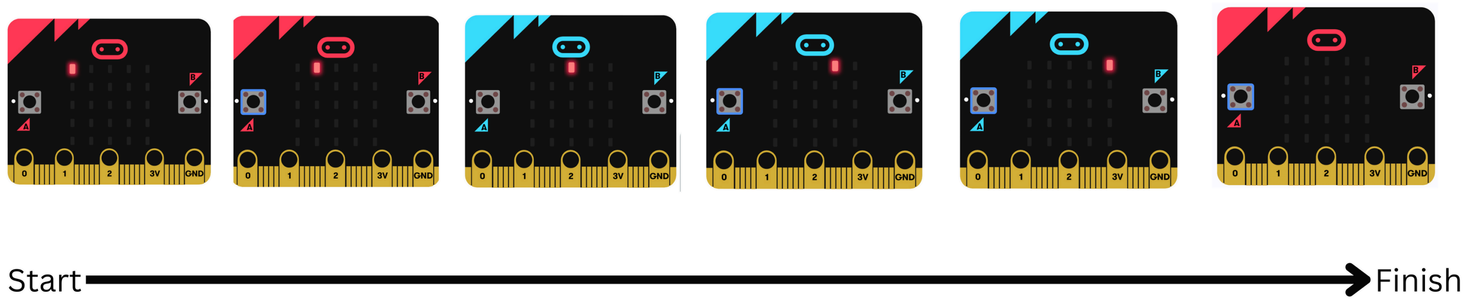
The completed code should look like this:



```
on start
    set xCord to 0
    set yCord to 0

forever
    plot x xCord y yCord

on button A pressed
    unplot x xCord y yCord
    change xCord by 1
    pause (ms) 200

on button B pressed
    unplot x xCord y yCord
    change xCord by -1
    pause (ms) 200
```

Created for Edutopia by Sabrina Tirachen (Of *My Tech-Savvy Classroom*)

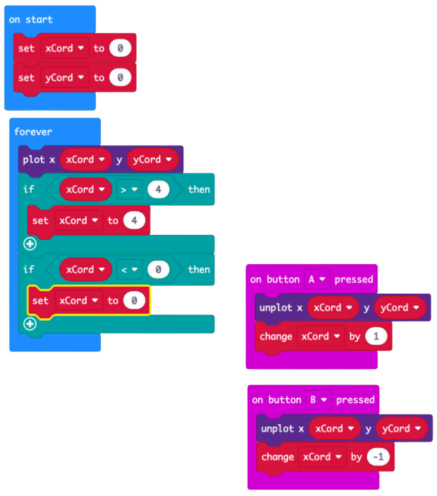
After 5 clicks, the dot falls off the grid! And you'll need an invisible fence to keep it on the screen



You'll need an invisible fence to keep it on the screen

Part 2: Creating an Invisible Fence. Ask students what happens when the sprite moves all the way to the edge of the grid. It falls off! Explain the concept of creating an "invisible fence" using conditional statements to prevent the sprite from leaving the grid. Walk students through setting a conditional rule: "If the X coordinate is greater than 4, set xCord to 4." This ensures the sprite can't move beyond the right edge of the grid.

Check for understanding: Students will now create a similar rule to prevent the sprite from falling off the left side of the grid (i.e., if xCord < 0, set xCord to 0). The completed code will look like this:



```
on start
    set xCord to 0
    set yCord to 0

forever
    plot x xCord y yCord
    if xCord > 4 then
        set xCord to 4
    if xCord < 0 then
        set xCord to 0

on button A pressed
    unplot x xCord y yCord
    change xCord by 1

on button B pressed
    unplot x xCord y yCord
    change xCord by -1
```

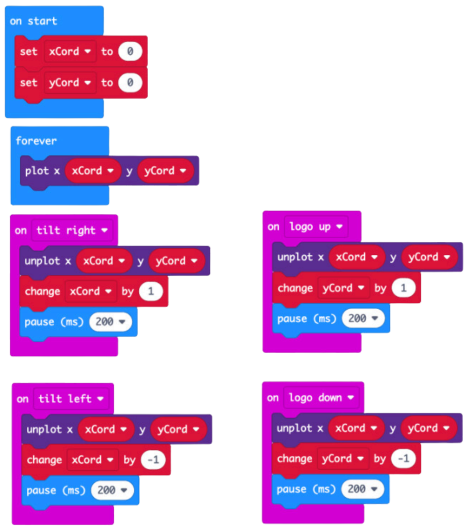
Created for Edutopia by Sabrina Tirachen (Of *My Tech-Savvy Classroom*)

Part 3: Using Tilt Inputs to Move the Sprite
1. Introducing new inputs

Start by reviewing the inputs students have already used, like button A and button B. Then, ask them to identify other inputs available on the Micro:Bit, such as tilt or shake. Explain that they will now use the tilt input, which utilizes the accelerometer in the Micro:Bit. Ask if they've played phone games where tilting the phone controls movement; this is the same mechanism.

2. Moving the sprite with tilt:

Guide students to write code that moves the sprite right using the "on tilt right" block. Like before, they will unplot the sprite's current position, change the "xCord" value by 1, and add a short pause (200 ms) to see the movement. Model how to use "on tilt down" to move the sprite downward by adjusting the "yCord" value by -1. Remind students to unplot the sprite and re-plot it after updating the coordinates. The completed code will look like this:



```
on start
    set xCord to 0
    set yCord to 0

forever
    plot x xCord y yCord

on tilt right
    unplot x xCord y yCord
    change xCord by 1
    pause (ms) 200

on logo up
    unplot x xCord y yCord
    change yCord by 1
    pause (ms) 200

on tilt left
    unplot x xCord y yCord
    change xCord by -1
    pause (ms) 200

on logo down
    unplot x xCord y yCord
    change yCord by -1
    pause (ms) 200
```

This combined 5x5 Battleship game and sprite movement activity is an engaging and practical way for students to become comfortable with key concepts in both math and coding. By using X and Y coordinates in a familiar and enjoyable game format, students not only reinforce their understanding of the coordinate plane but also develop essential problem-solving and collaboration skills. As they move into coding with MakeCode, students are introduced to foundational programming concepts such as variables, inputs, and conditional statements. By manipulating a sprite's movement on a 5x5 LED grid, students gain hands-on experience with the logic and structure of coding. This approach effectively integrates math and coding, making learning both accessible and enjoyable while building critical thinking skills.:

Created for Edutopia by Sabrina Tirachen (Of *My Tech-Savvy Classroom*)