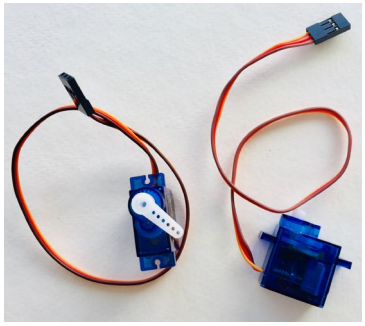
WEEK 7 DAY 1: INTRODUCTION TO SERVOS



INTRODUCTION

In this session, makers are introduced to programming servo motors with the Micro:bit.

The servo motor can be a neat addition to the options for coding the Micro:bit, in preparation for the Cyber Arcade projects they'll be working on in the following weeks.

ESSENTIAL QUESTIONS

- What is a servo motor and how can I control it with code?
- How do artists, engineers, and makers solve problems when they're working?

LEARNING OUTCOMES

1. Learn how to use code to program a servo motor.
2. Engage in project-based learning through problem-solving and troubleshooting by creating a game using a Micro:bit and code.

VOCABULARY

Servo motor: Motor that can be programmed with an electrical signal to move to a specific position

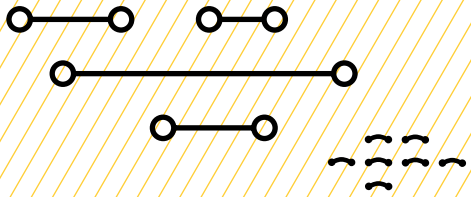
Servo mounting horn: Attaches to the servo to allow other objects to be connected to the servo (e.g., a wheel or an arm)

User: Person playing and interacting with the game

User interface (UI): Physical and digital design of how the user interacts with the game

User experience (UX): How natural and enjoyable the experience is

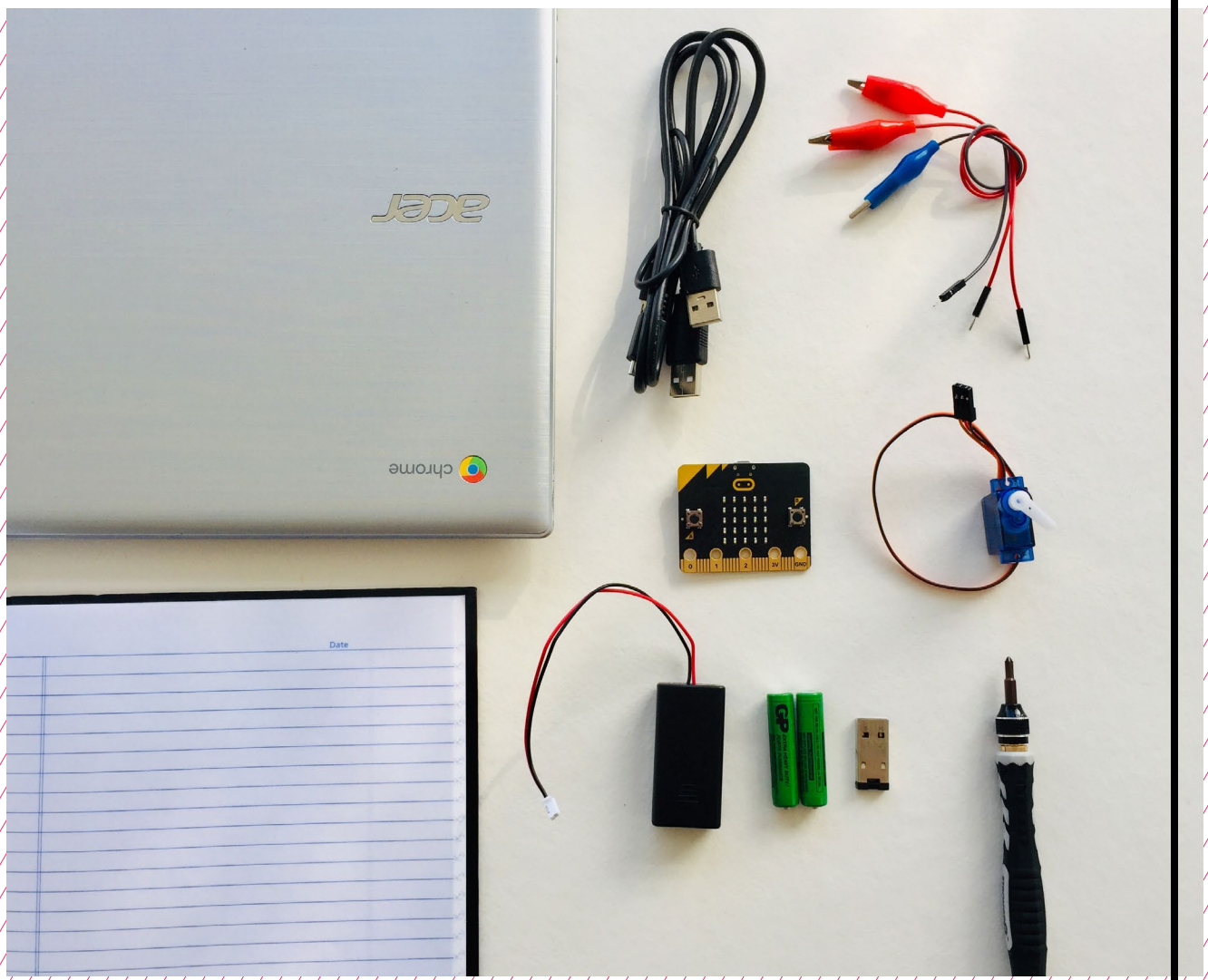
Troubleshooting: Using resources to solve issues as they arise



MATERIALS LIST

EACH PAIR OF MAKERS NEEDS:

- Micro:bit microcontroller
- AAA batteries (2)
- External battery pack
- Alligator-to-pin wires (3)
- Servo motor with servo horns
- Laptop with internet connection
- USB to micro-USB cord
- USB flash drive
- Notebook
- Mini screwdriver



TEACHER PREP WORK

1. Prepare the projector and make sure the internet is working.
2. Preload the slideshow and videos to save time.
3. Prepare an example of a servo with code from Steps 2 and 3 (optional).
4. Attach servo mounting horns to servo motors using a small screwdriver.

FACILITATION TIPS

Tinkering with electronics: Tinkering with electrical connections on a servo can be a bit tricky and won't work if things aren't securely connected. If students get frustrated, encourage them to check the color coding of the wires throughout the process and to also use the Troubleshooting Tips. They can use painter's tape or other non-permanent ways of securing connections.

Materials management: It's up to you as the educator to decide what works best for your class. You can portion out maker materials into paper trays for each table, or have a dedicated area where makers can access materials freely as needed.

Collaboration: Let smaller issues work themselves out. Record specific positive examples that you can share with makers in the moment or at the end of the project. These examples provide models for all learners.

Frustration: When frustration levels aren't high, let learners figure it out or keep facilitation low touch by asking a question and walking away. When frustration levels are high, intervene more directly to help makers find some success.

Circulate among the makers and monitor for both collaboration and frustration.

ADDITIONAL RESOURCES

Behind the MakeCode Hardware: Servo Motors with Micro:bit

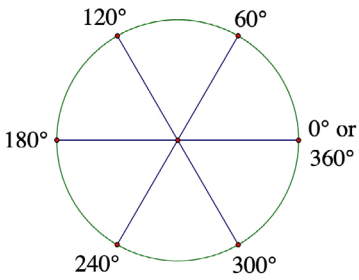
Driving a Servo with the Micro:bit

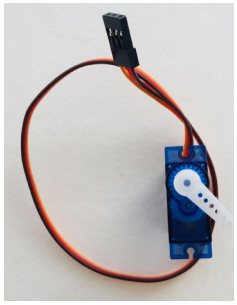
Connecting a Servo Motor to the Micro:bit

INTRODUCTION TO SERVOS

STEP 1 (5 MINUTES)

Introduce the servo motor.





Ask makers:

- What is a motor? *(A: Device that converts electrical energy into rotating mechanical energy.)*
- What kinds of devices have motors in them? (Students should brainstorm.)

Give each group of makers a **servo motor** with a **servo horn** attached. Makers will draw and list parts they notice in their notebook.

Note: Advise makers to handle gently and to not force the turning of the arm, as it could break.

Ask makers to:

- Study the servo parts closely.
- Describe what they see/notice.

The parts to emphasize are: the 3 colored wires, gears, plastic arms, and the fact that the arm turns.

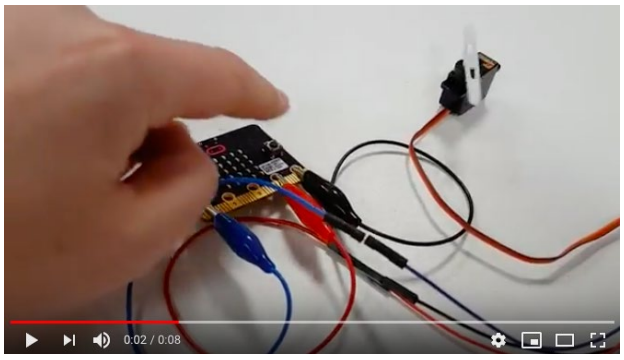
EXPLAIN

Today we'll connect and program movement using a special motor called a servo motor. A servo motor can be programmed to turn a specific number of degrees up to 180°, a full circle. We can use servos in addition to everything else you've done so far to create fun and interesting **user interface/user experience (UI/UX)** for interactive games or devices.

STEP 2 (20 MINUTES)

Program the servo motor.

Show makers this short video that demonstrates Micro:bit buttons A/B controlling a moving servo motor.



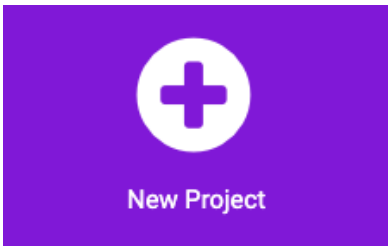
"Servo Attached to Micro:bit" on YouTube, uploaded by Teach with ICT, 6/25/2018

EXPLAIN

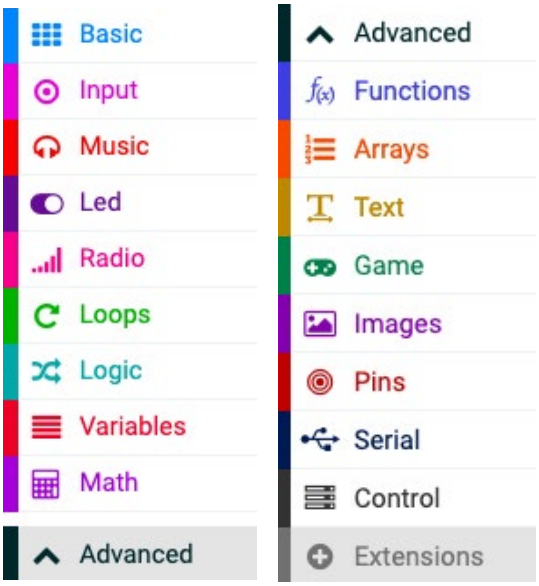
We can program a servo to turn a specific amount by connecting it to the Micro:bit board and using special servo coding blocks. First, let's work with the code in the simulator. Then, we'll upload it to the board and connect the servo to the board.

DEMONSTRATE AND HAVE MAKERS FOLLOW ALONG:

1. Go to the MakeCode editor URL (makecode.microbit.org) and open a new project.



2. In the blocks menu, click on **Advanced** and then scroll down to click on **Extensions**.



3. This will take you to a new page. Click on "servo A micro-servo library". This will add the additional blocks we need to program the servo.

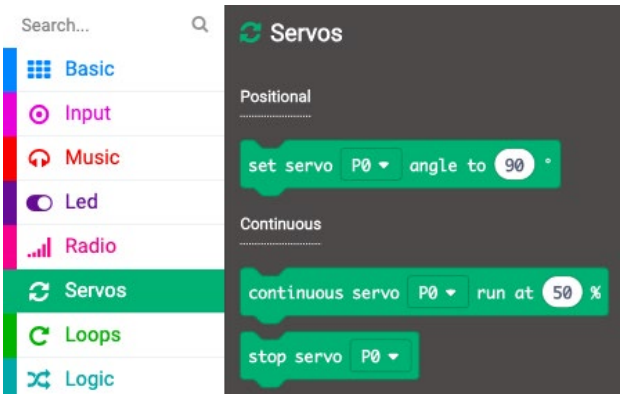
4. Click and drag over a **when button A is pressed** from the **Inputs** menu.

5. Drag over a **pause** block from the **Basic** menu and nest it in the first block.

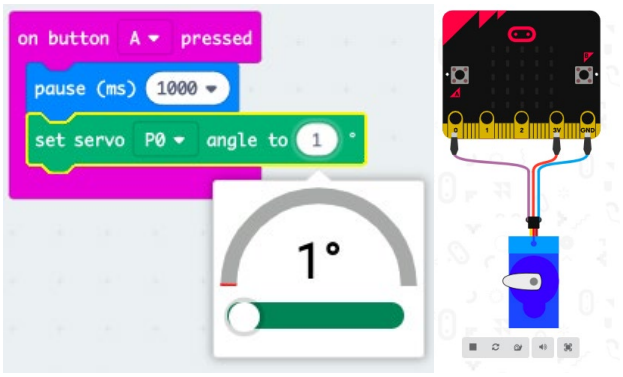
6. Change the pause time to be 1 sec (or 1000 ms).

7. Notice there's a new **Servos** block menu. Click and drag over a **set servo**

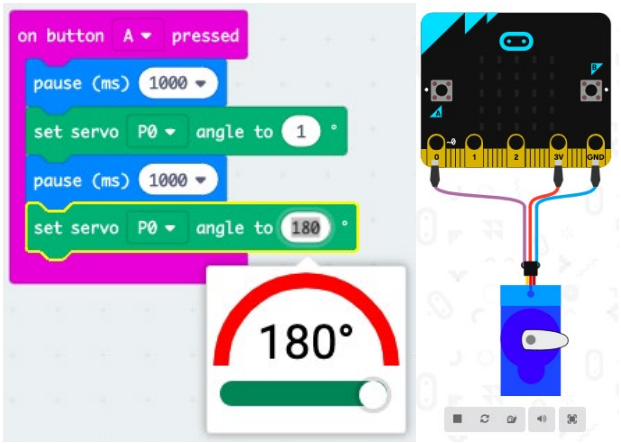
P0 angle to block under the **pause** block.



8. Click on the number 90 within the block and change it to be the number 1. Then you'll notice a servo appear in the simulator.



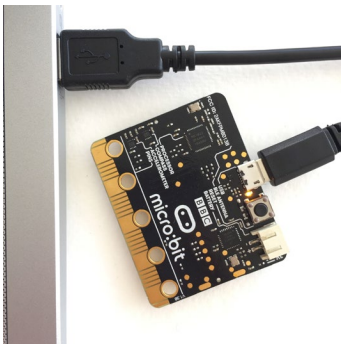
9. Repeat the **pause** and **set servo** blocks by duplicating or dragging over new blocks. This time, change the value in the **set servo** block to be 180. Click on **button A** in the simulator, and you'll notice that the servo will move to the opposite side.



10. Repeat Steps 8 and 9 two more times. You'll have a series of blocks that look like the blocks below. Again, click on **Button A** in the simulator and you should see the servo arm move back and forth twice.

11. Name your file with a unique name and save the file to the USB flash drive. Connect the Micro:bit with the USB cord.

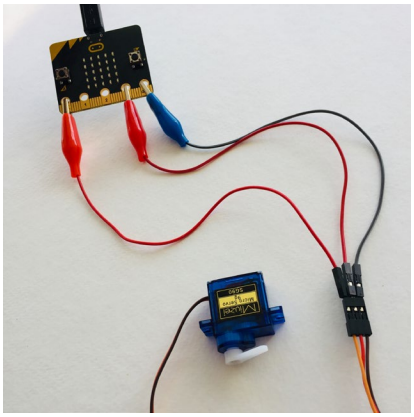
Click and drag the file on to the Micro:bit to upload.



Note: Leave the Micro:bit connected to the laptop for the next step.

STEP 3 20 MINUTES

Connect the servo motor.



Once makers have successfully programmed a servo in the simulator and uploaded the code to the Micro:bit, it's time to connect the physical servo to the board. The Micro:bit should still be connected to the laptop.

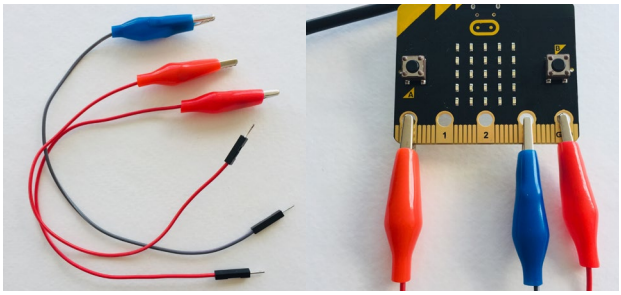
Give each group 3 alligator-to-pin wires, one servo, and a battery pack.

EXPLAIN

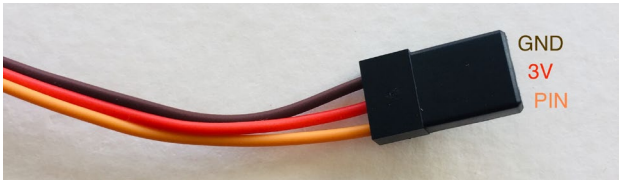
Now that we've tested our code and uploaded it to the Micro:bit, we'll connect the servo.

DEMONSTRATE AND HAVE MAKERS FOLLOW ALONG:

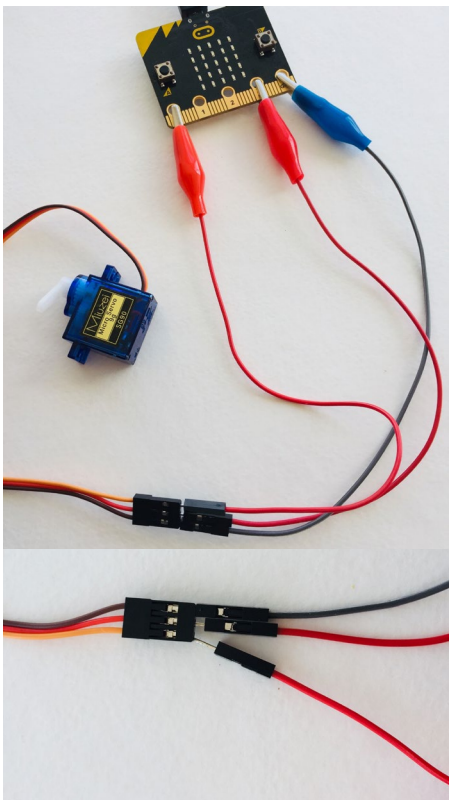
1. In order for the servo to get the code, we need to connect 3 of the alligator-to-pin wires to the GND, 3V, and PIN 0 terminals on the Micro:bit board using the alligator side of the wire.



2. Next, notice on the servo there are 3 colored wires: black, red, and orange. These wires are important to keep track of. The brown wire is the ground wire (GND) or the (–) terminal and the red wire is the 3V or (+) terminal. These help to deliver electricity and power to the servo. The orange wire is the signal wire, or the wire that actually delivers the code from the programmed pin on the Micro:bit to the servo.

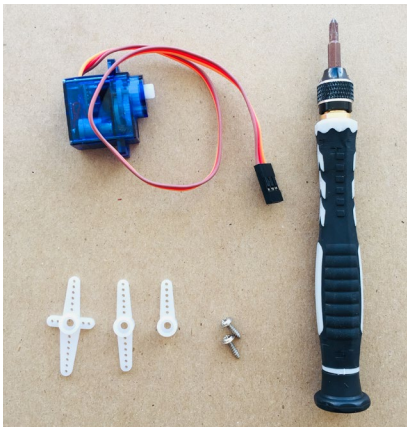


3. Connect your servo wires to the Micro:bit board following the table below. If you can match the colors of the servo motor, great! If not, the colors of the alligator-to-pin wires don't actually matter—it's just helpful to use the same colors so we can easily track where the wires are going.

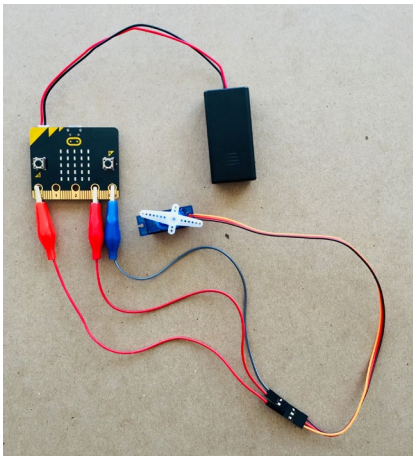


GND	Brown
3V	Red
PIN	Yellow

4. Once you think you have the connections right, try pressing Button A and check if the servo is moving or not. You'll hear a mechanical rotating sound if the servo is working. If your servo doesn't already have a plastic mounted "horn" piece, then use a mini screwdriver to attach one so that it's easier to see the movement.

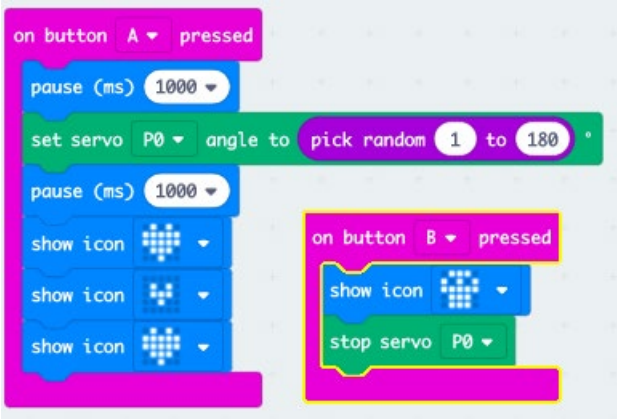


5. If the servo works as displayed on the computer simulator, you can disconnect the Micro:bit and connect it to a battery pack.
6. If the servo isn't moving, check your wiring to see if the connections are right (Steps 2 and 3). If the servo is still not moving from 0 to 180 degrees, then check the code on the screen.



STEP 4 10 MINUTES

Further explore programming the servo.



Once makers have successfully programmed and connected a servo, have them go back and tinker with the code (e.g.,. change inputs, add displays, move a different amount of degrees, etc.). Remind them that they'll need to plug their Micro:bit back into their computer.

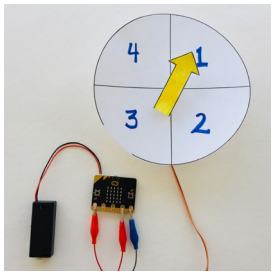
STEP 5 5 MINUTES

Clean up.

Makers will:

- Save their file with a unique name onto the USB flash drive.
- Safely disconnect the Micro:bit from the computer.
- Put materials away in their bins.
- Put away technology and make sure laptops are charging.
- Return tools and materials that can be used again to the right place.
- Clear tables of garbage and recycling.

WEEK 7 DAY 2: DESIGNING A GAME WITH A SERVO



INTRODUCTION

In this session, makers develop a game concept using a programmed servo and a spinner template, devising their own game mechanics.

ESSENTIAL QUESTIONS

- How can we design a game using a programmed servo?
- How do artists, engineers, and makers solve problems when they're working?

LEARNING OUTCOMES:

1. Continue to practice programming a servo motor.
2. Engage in project-based learning through problem-solving and troubleshooting by creating a game with a Micro:bit and a servo.

VOCABULARY

Servo motor: Motor that can be programmed with an electrical signal to move to a specific position

Servo mounting horn: Attaches to the servo to allow other objects to be connected to the servo (e.g., a wheel or an arm)

Game spinner: Device that spins and determines what comes next in a game

User: Person playing and interacting with the game

User interface (UI): Physical and digital design of how the user interacts with the game

User experience (UX): How natural and enjoyable the experience is

Game mechanics: Rules and rewards that make up game play and create a fun and engaging experience

Troubleshooting: Using resources to solve issues as they arise

MATERIALS LIST

EACH PAIR OF MAKERS NEEDS:

- Micro:bit microcontroller
- Servo motor
- Laptop with internet connection
- USB to micro-USB cord
- USB flash drive
- Notebook
- Alligator-to-pin wires (3)
- Mini screwdriver

ALL MAKERS NEED ACCESS TO:

- Markers
- Colored pencils
- Scissors
- Cardboard scissors
- Masking tape
- Thin cardboard (cereal boxes or tag board)
- Cardstock
- Aluminum foil
- Pipe cleaners, pom-poms, popsicle sticks, bottle caps/recycling (optional)

Items can be portioned out per table or set up in an area where students can access them freely.

TEACHER PREP WORK

1. Ensure the internet connection is working, and connect your laptop to a projector or screen.
2. Preload videos and slideshow to save time.
3. Print the spinner game templates and shapes on white cardstock.
4. Set up an equitable system for student access of materials (either as one materials table or distributed evenly across tables).
5. Print the Troubleshooting Tips at the end of the lesson and post in the classroom.

FACILITATION TIPS

Servos and code: Servos can be tricky to work with. There could be an issue with the code or the wire connections to the board or servo (see Troubleshooting Tips). If you're able to test, troubleshoot, and work out issues in an example, it'll save time when you're facilitating makers encountering the same issues during class.

Materials management: It's up to you as the educator to decide what works best for your class. You can portion out maker materials into paper trays for each table, or have a dedicated area where makers can access materials freely as needed.

Collaboration: Let smaller issues work themselves out. Record specific positive examples that you can share with makers in the moment or at the end of the project. These examples provide models for all learners.

Frustration: When frustration levels aren't high, let learners figure it out or keep facilitation low touch by asking a question and walking away. When frustration levels are high, intervene more directly to help makers find some success.

Circulate among the makers and monitor for both collaboration and frustration.

ADDITIONAL RESOURCES

Behind the MakeCode Hardware: Servo Motors with Micro:bit

Driving a Servo with the Micro:bit

Connecting a Servo Motor to the Micro:bit

DESIGNING A GAME WITH A SERVO

STEP 1 (3 MINUTES)

Review programming a servo.

Ask makers:

- How did we connect the **servo motor**? *(A: By connecting it to the Micro:bit with 3 wires to GND, 3V, and Pin.)*
- Which color of wire on the servo is which? *(A: Red=3V or +, Brown=GND, Orange=Pin)*

STEP 2 (15 MINUTES)

Program a servo game spinner.

"Custom Game Spinner and Spin Wheel" on YouTube, uploaded by BoardGamesMaker, 4/6/2017

Ask makers, "Have you seen any digital or board games that use a game spinner?"

EXPLAIN

A **spinner** is a tool used to randomize choices, just like a dice. In the following example, an online spinner is used to randomize choices of what to have for dinner.

DEMONSTRATE

1. On the projector, or by sharing the link, show makers **this online spinner**. Spin the wheel several times, noticing the results.
2. Next, click on the "Modify Wheel" button below the wheel and delete some of the items (e.g., seafood and diner), and ask makers to add a category of food they like.

EXPLAIN

A game spinner can also be a physical object in a game. Show this **video** and ask makers, "How does the spinner act as a user interface to help create the **game mechanics**, or rules of the game?"

EXPLAIN

Last session, we programmed a servo to turn from one direction to another. Today we'll program a servo motor to turn at a random number of degrees, and then we'll design a simple game.

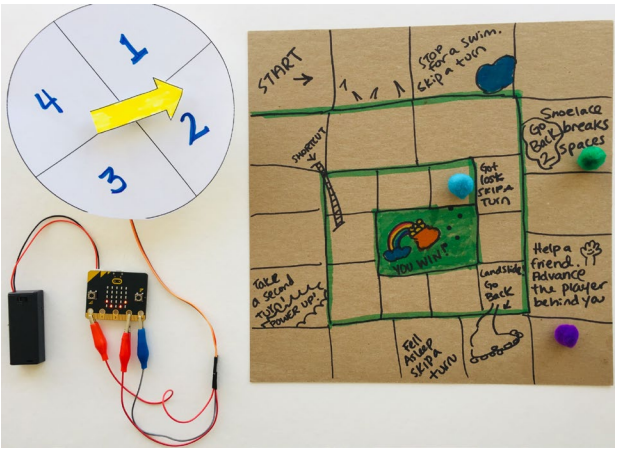
Similar to when we programmed a Micro:bit "dice" using **Math** blocks, we'll program a servo to be a game spinner. You and your partner will decide the game mechanics of what happens next.

First, we'll program the servo within the software, then you'll remix that code and change things before uploading to your board and servo.

STEP 3

25 MINUTES

Design a game with a servo motor.

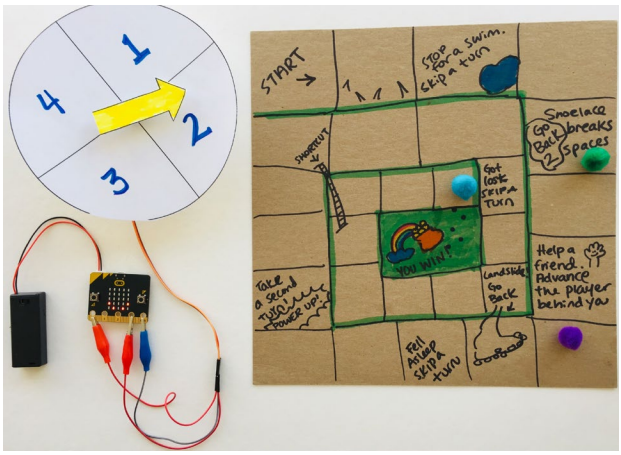


Makers now remix the code to create a simple game using a **spinner game template**.

EXPLAIN

Once you've coded the servo in the simulator to move a random number of degrees, you and your partner can remix and change the code to create a simple game.

You can use any of the printed circle templates and fill them in with your own rules of game mechanics. Remember that game mechanics are the rules and rewards that make up game play and create a fun and engaging experience.



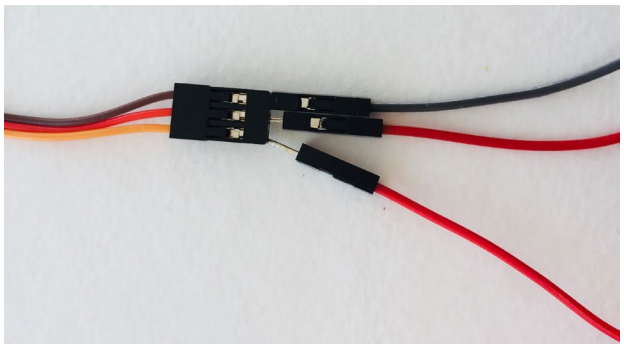
In this example, the servo is programmed to move to a random degree between 1 and 180 degrees. The number it turns to indicates the number of spaces to move on the board.

You can use cardboard and the maker materials to modify your servo horn as an arrow or other shape.

We'll share our game ideas in the last 10 minutes of the session.

Note: Once makers have come up with an idea for their servo spinner game, they'll likely need support in saving, downloading, uploading, and connecting the servo correctly.

Refer back to the previous session when we connected the servo wires to the board according to color.



GND	Brown
3V	Red
PIN	Yellow

STEP 4

10 MINUTES

Share and reflect.

Have makers explain and play each other's games. This can be done gallery walk style or presentation style.

STEP 5

7 MINUTES

Clean up.

Makers will:

- Save their file with a unique name on to the USB flash drive.
- Safely disconnect the Micro:bit from the computer.
- Put materials away in their bins.
- Put technology away and make sure laptops are charging.
- Return tools and materials that can be used again to the right place.
- Clear tables of garbage and recycling.

TROUBLESHOOTING TIPS

The servo isn't moving or responding.

Alligator-to-pin wires

- Check to see that the wires are properly connected to the servo (GND to brown, 3V to red, and Pin to orange).
- Make sure alligator clips are secure on the correct pins and are touching the metallic parts. Try extending from the board with a paper fastener.
- Try switching out the alligator clips for new ones.

Power

- The servo might not be getting enough power from the battery pack. Try powering the Micro:bit using the USB to the laptop. If it works, replace the batteries or the battery pack and try again.
- Try pressing the reset button on the Micro:bit.

The servo is making some sound but not rotating.

Bugs in the code

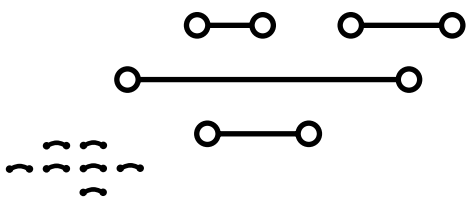
- Read through the code.
- Check to make sure there's a short pause block in the code before and after every turn. The servo cannot move without a pause before and after moving.
- The servo gears could be burnt out or broken. Try replacing the servo.

The servo isn't turning the amount we thought it should.

Servo type

- Try connecting a different servo and try again.
- Servo gears could be uncalibrated or broken. Replace the servo and try again.

The LED on the Micro:bit isn't flashing when we click Upload.	**Bad cable or port** • If the Micro:bit isn't showing up in the computer menu, try a different cable. • Try a different USB port on the laptop.
Our code isn't uploading correctly to the board. The board feels hotter than usual.	**Burnt board** • Try pressing the reset button on the board. • Try uploading to a new Micro:bit board. • Check to see if there's conductive material touching the board. It could be causing a short circuit.
The board isn't turning on when connected to the battery pack.	**Battery** • Test the batteries to see if they're charged. • Check to see if the batteries are flipped.
The LED on the Micro:bit isn't flashing when we click Upload.	**Bad cable or port** • If the Micro:bit isn't showing up in the computer menu, try a different cable. • Try a different USB port on the laptop.
The connections won't stay in place.	**Alligator clips** • Use painters tape to hold the alligator clips in place. • Try using foil or paper fasteners to extend the metal parts of the board (Pin, GND, 3V).



TROUBLESHOOTING TIPS

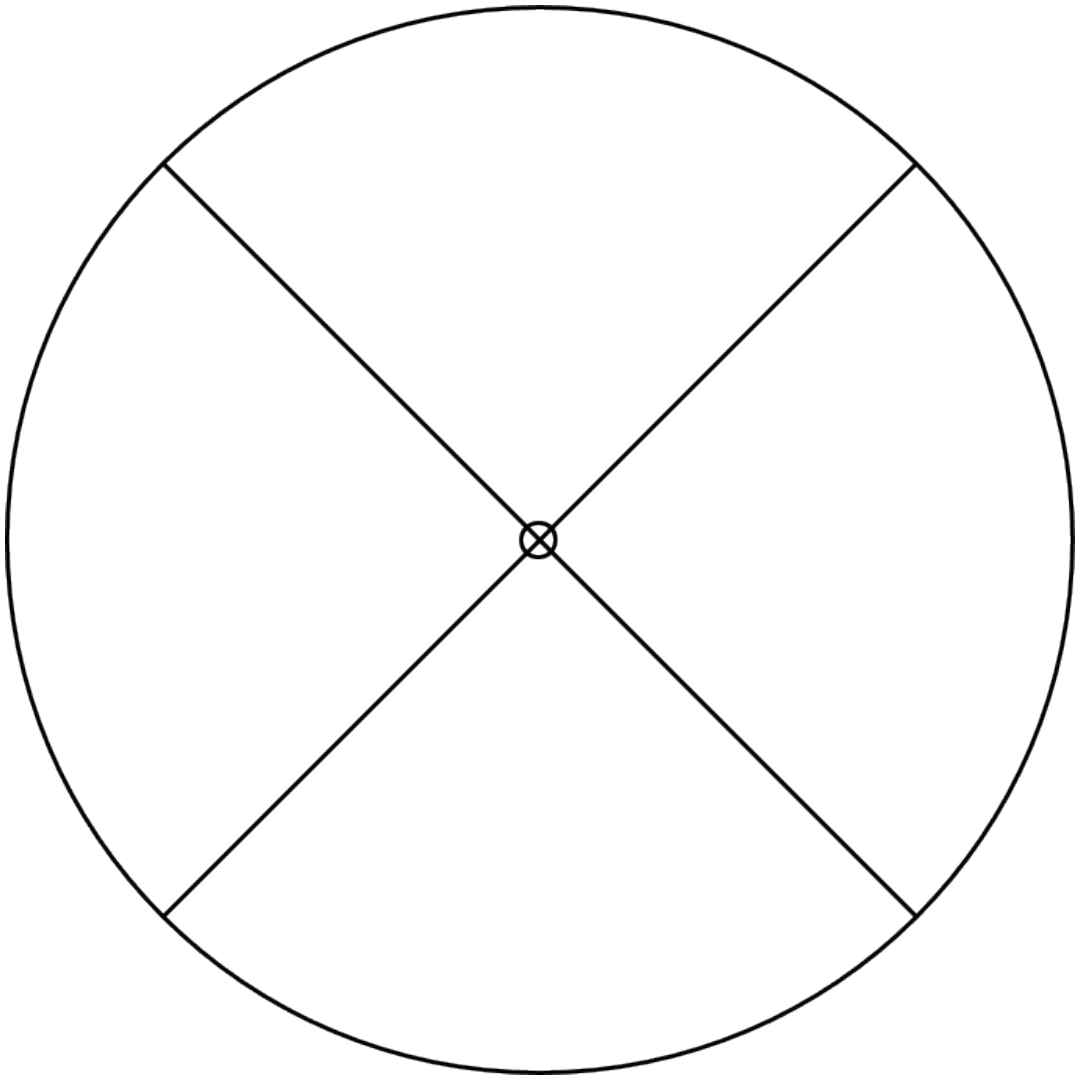
Print and use the empty rows to fill in with other problems and solutions that can be shared.

SERVO SPINNER GAME

Use the spinner template below and cut out any of the shapes on the next page to use in designing your game.

Title of Game: _____

How to Play (game mechanics):



SERVO SPINNER GAME