

# Goldberg Gator Engineering Explorers Summer Program

## Table of Contents

<b>PROGRAM MATERIALS</b>	<b>2</b>
<b>DAY 1: INTRODUCTIONS, PROGRAMMING BASICS, CODING APPLICATIONS</b>	<b>4</b>
OVERVIEW	4
MATERIALS NEEDED FOR DAY 1	4
ACTIVITY 1: INTRODUCTIONS, INTRODUCTION TO MICRO:BIT AND ICE BREAKER	4
<i>Icebreaker Activities</i>	4
<i>Establish Norms for Working together</i>	5
<i>Introduction to Micro:bit Hardware and Programming</i>	5
<i>How to Use a Micro:bit and MakeCode</i>	5
ACTIVITY 2: SMALL GROUP ACTIVITIES – PROGRAMMING BASICS AND COMPUTATIONAL THINKING SKILLS	6
1. <i>Process Mapping</i>	6
2. <i>Create a Micro:bit Name Badge</i>	8
LUNCH AND RECESS	11
3. <i>Understanding Logic</i>	11
4. <i>Troubleshooting and Debugging</i>	14
5. <i>Problem Solving</i>	16
<b>DAY 2: CODING APPLICATIONS, INTRO TO MICRO:BIT PET</b>	<b>17</b>
OVERVIEW	17
MATERIALS NEEDED FOR DAY 2	17
ACTIVITY 1: WELCOME	17
ACTIVITY 2: CODING APPLICATIONS	17
<i>Coding Application 1: Creating a Rock, Paper, Scissors Game</i>	17
<i>Coding Application 2: Collect Light Intensity Data and Create an Automatic Light</i>	20
LUNCH AND RECESS	26
ACTIVITY 3: DESIGN A MICRO:BIT PET – INTRO TO PET DESIGN CHALLENGE, FIND PARTNER, INTERVIEW	26
<b>DAY 3: DESIGN A MICRO:BIT PET - DEFINE, IDEATE, FEEDBACK, PROTOTYPE 1, FEEDBACK, PROTOTYPE 2, FEEDBACK, FINAL DESIGN</b>	<b>27</b>
OVERVIEW	27
MATERIALS NEEDED FOR DAY 3	27
ACTIVITY 1: WELCOME	28
ACTIVITY 2: DESIGN A MICRO:BIT PET – DEFINE, IDEATE, FEEDBACK, PROTOTYPE 1, FEEDBACK, PROTOTYPE 2, FEEDBACK, FINAL DESIGN	28
LUNCH AND RECESS	29
<b>DAY 4: DESIGN A MICRO:BIT PET, MICRO:BIT PET EXTENSION, AND INTRO TO DESIGN CHALLENGE</b>	<b>32</b>
OVERVIEW	32
MATERIALS NEEDED FOR DAY 4	32
ACTIVITY 1: WELCOME	32
ACTIVITY 2:	32
ACTIVITY 3: DESIGN A MICRO:BIT PET – EXTENSION	34
<i>Bluetooth Communication</i>	34
<i>Movement with Servo Motors</i>	36
<i>Reaction to Sensors - Accelerometer</i>	37
<i>Reaction to Sensors – Ultrasonic Detector</i>	38
LUNCH AND RECESS	40
ACTIVITY 4: DESIGN CHALLENGE – INTRODUCTION, MENTOR PRESENTATIONS, DESIGN TEAMS, ROLES	40

<b>Day 5: Design Challenge – Interview, Define, Ideate, Feedback, Prototype 1, Feedback</b>	<b>42</b>
Overview	42
Materials Needed for Day 5	42
Activity 1: Welcome and Icebreaker	42
Activity 2: Design Challenge - Interview, Define, Ideate, Feedback, Prototype 1, Feedback	42
Lunch and Recess	44
<b>Day 6: Design Challenge – Refine, Prepare to Present, Progress Check and Gallery Walk, Prototype 2</b>	<b>46</b>
Overview	46
Materials Needed for Day 6	46
Activity 1: Welcome and Icebreaker	47
Activity 2: Design Challenge – Refine, Prepare to Present, Progress Check/Gallery Walk, Prototype 2	47
Lunch and Recess	47
<b>Day 7: Design Challenge – Feedback, Refine, Final Design, Prepare to Present, Progress Check-in Jigsaw</b>	<b>49</b>
Overview	49
Materials Needed for Day 7	49
Activity 1: Welcome and Icebreaker	50
Activity 2: Design Challenge – Prototype 2 Feedback, Refine, Final Design, Create Plan, Jigsaw	50
Lunch and Recess	51
Lunch and Recess	52
<b>Day 8: Design Challenge – Final Day</b>	<b>54</b>
Overview	54
Materials Needed for Day 8	54
Activity 1: Welcome	54
Activity 2: Design Challenge – Final Product, Prepare Pitch, Pitch Presentations	55
Lunch and Recess	55
Activity 3: Extra Design Challenge	56
<b>Extra Camp Activities</b>	<b>57</b>
Any Time	57
Activity: Career Talks	57
After Programming Basics	57
Activity: Hot Potato Game	58
Activity: Dice Roll	60
Activity: Magic 8 Ball	61
Activity: Step Counter	62
After Micro:bit PET Extensions	63
Activity: Follow Black Line	63
After Technical Design Challenge	64
Activity: Turtle	64

## Program Materials

Technology	Craft Supplies	Activity Supplies
<ul style="list-style-type: none"> <li>Computers</li> <li>Micro:bits - 1 per student</li> <li>Servo motors</li> </ul>	<ul style="list-style-type: none"> <li>Paper</li> <li>Tape</li> <li>Markers</li> <li>Pencils</li> </ul>	<ul style="list-style-type: none"> <li>Sticky Notes</li> <li>Sticky Easel Pad (optional)</li> <li>Rulers</li> </ul>

<ul style="list-style-type: none"> <li>● Dupont Wires (male –female, male – male)</li> <li>● Alligator Clips</li> <li>● AAA Batteries</li> <li>● Micro:bit Stem Kits – sensors, LED lights, motors</li> </ul>	<ul style="list-style-type: none"> <li>● Scissors</li> <li>● Construction paper</li> <li>● Paper towel/toilet paper rolls</li> <li>● Cardboard/Cardstock</li> <li>● Other craft materials for design projects</li> </ul>	<ul style="list-style-type: none"> <li>● Flashlight/cell phone for light lab</li> <li>● Student Rewards</li> </ul>
---	--	--

## Day 1: Introductions, Programming Basics, Coding Applications

### Overview

Activity	Time, minutes
Summer Camp Team Introduction, Introduction to Micro:bit and Ice Breaker	70
Programming Basics (Small group Activities)	110
Lunch and Recess	45
Problem Solving Activity	15
Total	240
Extra Time	0

### Materials Needed for Day 1

Technology	Craft Supplies	Activity Supplies
<ul style="list-style-type: none"><li>• Computers</li><li>• Micro:bits - 1 per student</li><li>• AAA Batteries</li></ul>	<ul style="list-style-type: none"><li>• Markers</li><li>• Pencils</li><li>• Construction Paper</li></ul>	<ul style="list-style-type: none"><li>• Consent Forms</li><li>• Link to survey</li><li>• Name Tags</li><li>• Sticky Notes</li><li>• Sticky Easel Pad</li><li>• Stickers</li><li>• Rulers</li></ul>

### Activity 1: Introductions, Introduction to Micro:bit and Ice Breaker

**Estimated Time:** Day 1 Paperwork/Surveys/Sign into Teams – 15 minutes, Introduction, Team Introduction – 5 minutes, Ice breaker: Name Tag – 5 minutes, Programming Languages Activity - 15 minutes, IRB Consent – 10 mins, Establish Norms – 5 minutes, Intro to Micro:bit – 15 minutes. **Total time = 70 minutes**

#### Activity Goals

- Establish norms for working together and using the technology
- Introduce Facilitator Team
- Meet groups/partners

#### Icebreaker Activities

- Nametag Activity
  - Students will create a nametag using construction paper.
  - Show students how to fold a piece of paper into a trifold to create a nametag that can be placed by their computer.
  - Tell students to write their name on the paper and decorate it to represent themselves.
- CS Unplugged: Programming Languages Activity
  - Goal: Teaches students that computers work by following a list of instructions that they have to follow even if they do not make sense



- Demonstration Example: Have students draw a picture from the instructions you give them verbally. Page 3 of linked document
- Student Example: Choose a student to come to the front of the room and give them a simple drawing. That student has to verbally provide the instructions for the rest of the class to draw the image. Repeat with 1-2 more students.
- Discussion: Make the connections between how this exercise relates to computers and programming.
- [https://classic.csunplugged.org/documents/activities/programming-languages/unplugged-12-programming\\_languages.pdf](https://classic.csunplugged.org/documents/activities/programming-languages/unplugged-12-programming_languages.pdf)

### Establish Norms for Working together

Ask students what rules/norms they have for working with people – chart them down

Share our norms and add student norms

- Ask questions
- Be present
- Treat others with respect
- Share your thoughts
- Keep an open mind
- Do your part
- Treat technology and tools with care
- Listen with intent

### Introduction to Micro:bit Hardware and Programming

***Estimated Time: 15 minutes***

Goals:

- Establish that computing is broken into inputs, outputs, and processes
- How to use the Micro:bit – parts of the Micro:bit and saving code
- How to use MakeCode editor to program the Micro:bit
- How to program external attachments (sensors, LEDs, etc.)
- How to use Micro:bit to collect and analyze data

### How to Use a Micro:bit and MakeCode

- [Introduction to Micro:bit Presentation](#) – this is a PowerPoint Presentation to use with students
- These are reference material for the intro – TEACHER & LEADERS should review thoroughly
  - Micro:bit first steps: <https://microbit.org/get-started/first-steps/introduction/>
  - Parts of a Micro:bit: <https://microbit.org/get-started/user-guide/overview/>

## Activity 2: Small Group Activities – Programming Basics and Computational Thinking Skills

### 1. Process Mapping

**Estimated Time:** 25 minutes

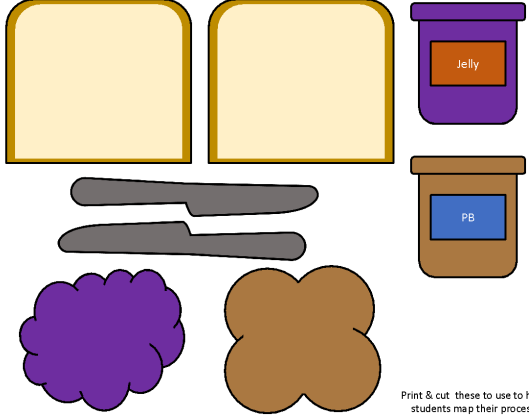
Materials: Print and cut out peanut butter jelly materials, draw morning routine process map

#### Activity Goals:

- Students learn the basics of process mapping and how it can connect to programming

#### Activity Procedure:

<p>1. Introduce the Process Map Key and the specifics of what each icon is used for</p> <ol style="list-style-type: none"> <li>Share morning routine process map – Small Group leader has SIMPLE morning routine process map drawn on anchor chart</li> </ol> <p><b>Facilitator Shares Slide or directs students to slide in online activity handout workbook OR Draws on Chart Paper</b></p>	<p style="text-align: right;"><b>Process Mapping</b></p> <p style="text-align: center;"><b>Morning Routine</b></p> <p>What do you notice about this process map?</p> <p>What do you see? What do you think it means?</p> <p>Are there any patterns?</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p style="text-align: center;"><b>Process Map Key</b></p> <ul style="list-style-type: none"> <li><span style="border: 1px solid green; border-radius: 50%; padding: 2px 10px;">Inputs</span> The start of your process</li> <li><span style="border: 1px solid red; border-radius: 50%; padding: 2px 10px;">Outputs</span> The end result of your process</li> <li><span style="border: 1px solid black; padding: 2px 10px;">Actions/Steps</span> The steps in the process: how to get from the start to the end</li> <li><span style="border: 1px solid blue; width: 20px; height: 20px; margin: 0 auto; transform: rotate(45deg);"></span> Questions or choices that can change the flow of the process</li> <li><span style="display: inline-block; width: 20px; height: 10px; background-color: black; margin-right: 5px;"></span> Show the direction of the steps in the process</li> </ul> </div>
<p>2. Have students in the small group draw a process map on how to make a peanut butter jelly sandwich by identifying the step together. Each student should draw their own map.</p> <ol style="list-style-type: none"> <li>Remind them of the icebreaker activity and how you had to be specific about the instructions and the actions.</li> <li>Throughout process mapping demonstrate the steps as they are written from the student process maps –</li> </ol>	<p style="text-align: right;"><b>Process Mapping</b></p> <p>Name: _____</p> <p>Process: A sequence of activities intended to produce particular results</p> <p>Map the process to make a PBJ sandwich</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p style="text-align: center;"><b>Process Map Key</b></p> <ul style="list-style-type: none"> <li><span style="border: 1px solid green; border-radius: 50%; padding: 2px 10px;">Inputs</span> The start of your process</li> <li><span style="border: 1px solid red; border-radius: 50%; padding: 2px 10px;">Outputs</span> The end result of your process</li> <li><span style="border: 1px solid black; padding: 2px 10px;">Actions/Steps</span> The steps in the process: how to get from the start to the end</li> <li><span style="border: 1px solid blue; width: 20px; height: 20px; margin: 0 auto; transform: rotate(45deg);"></span> Questions or choices that can change the flow of the process</li> <li><span style="display: inline-block; width: 20px; height: 10px; background-color: black; margin-right: 5px;"></span> Show the direction of the steps in the process</li> </ul> </div>

<p>they will see any missed steps or errors in their process</p> <p>c. Have students compare process maps and note differences between the steps.</p> <p><b>Printout*</b></p>	
<p>3. Students can use the cutouts to help guide their process map</p> <p><b>Printout*</b></p>	 <p>Print &amp; cut these to use to help students map their process</p>
<p>4. Have student groups share their process map with you and have a discussion on why sequences and clear instructions are important to programming</p>	
<p>5. Optional Extension: Introduce Students to draw.io software to create process maps. It is a very easy program to use and is really helpful for later projects. <a href="https://app.diagrams.net/">https://app.diagrams.net/</a></p>	

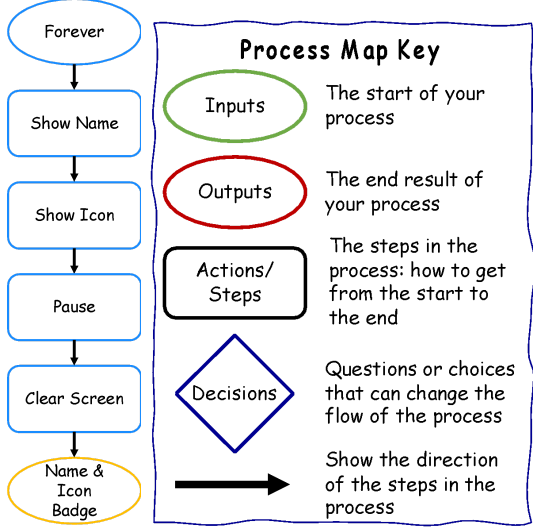
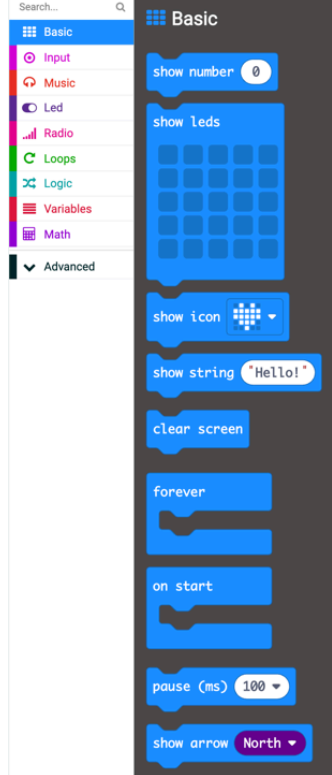
## 2. Create a Micro:bit Name Badge


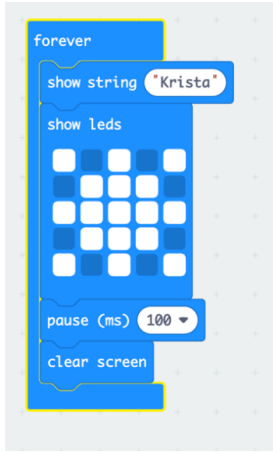
**Estimated Time:** 30 minutes

### Activity Goals:

- Students learn the basics of Micro:bit: show strings, icons, pause, forever loop
- Students extend programming to include inputs, loops, and basic sounds
- Students will create or analyze process maps for programming the name badge

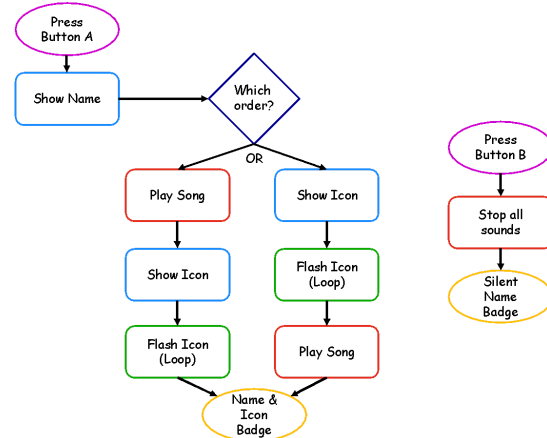
### Activity Procedure:

<p>1. Have students analyze the process map of the name badge code</p> <ol style="list-style-type: none"><li>Remind them of the process map key</li><li>Walk through inputs, outputs, and steps</li></ol> <p><b>Facilitator Shares Slide or directs students to slide in online activity handout workbook</b></p>	 <p>The diagram shows a flowchart on the left and a 'Process Map Key' on the right. The flowchart starts with an oval labeled 'Forever', followed by rectangular boxes for 'Show Name', 'Show Icon', 'Pause', and 'Clear Screen', ending with an oval labeled 'Name &amp; Icon Badge'. The 'Process Map Key' defines symbols: a green oval for 'Inputs' (start of process), a red oval for 'Outputs' (end result), a rectangle for 'Actions/Steps' (process steps), a diamond for 'Decisions' (choices that change flow), and an arrow for showing direction.</p>
<p>2. Then have students explore “Basic” code blocks on MakeCode to identify which code to use</p> <ol style="list-style-type: none"><li>Throughout this step ask students to share which code they believe is used for the name badge</li></ol>	 <p>A screenshot of the Micro:bit MakeCode interface showing the 'Basic' block palette on the left and a list of code blocks on the right. The blocks include 'show number', 'show leds', 'show icon', 'show string', 'clear screen', 'forever' loop, 'on start', 'pause (ms)', and 'show arrow'.</p>

<p>3. Students sequence the code blocks to develop the name badge using the process map</p>	
<p>4. Students input their name as the “string” and they choose an icon to use in their name badge</p> <p>a. <a href="https://microbit.org/projects/make-it-coder-it/name-badge/">https://microbit.org/projects/make-it-coder-it/name-badge/</a></p>	
<p>5. Show students how to test their code using the simulator in MakeCode to check for errors</p> <p>6. Then walk students through importing the code onto the Micro:bit</p>	
<p>7. After students complete the basic name badge, challenge them to add more to their code</p>	

8. Share the process map of the upgraded name badge – this includes inputs, loops, and sounds
- Walk through inputs, outputs, and steps

**Facilitator Shares Slide or directs students to slide in online activity handout workbook**



9. Have students explore the other types of blocks in MakeCode to identify which blocks to use
- Explain what loops are and how they function in code
    - Loops allow a portion of code to be repeated. The “Forever” block is a loop that has no end.

**Name & Icon Badge**

Explore the different types of code in MakeCode.

Which blocks of code would you use to...

Display your name?

>

Display the icon?

>

Play sound or music?

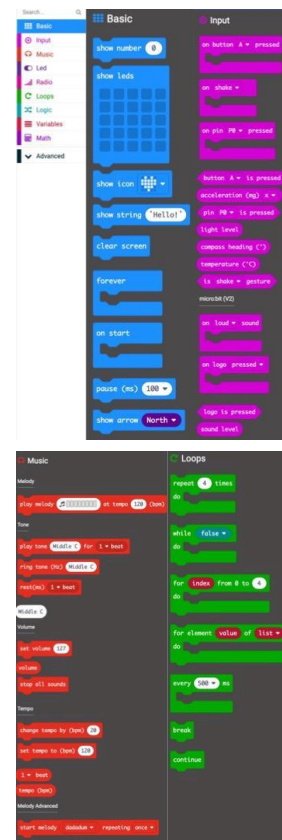
>

Repeat code?

>

Make the icon flash?

>



<p>10. Allow students to choose which sequence of code they would like to use</p>	<pre> graph TD     A([Press Button A]) --&gt; B[Show Name]     B --&gt; C{Which order?}     C --&gt; D[Play Song]     C --&gt; E[Show Icon]     D --&gt; F[Show Icon]     F --&gt; G[Flash Icon Loop]     E --&gt; H[Flash Icon Loop]     H --&gt; I[Play Song]     G --&gt; J([Name &amp; Icon Badge])     I --&gt; J     K([Press Button B]) --&gt; L[Stop all sounds]     L --&gt; M([Silent Name Badge])   </pre>
<p>11. Students program the more advanced name badge with loops to create flashing icons and sounds</p> <p>12. Check for errors using the simulator</p> <p>13. Load on to the Micro:bit.</p>	<pre> on button A pressed   show string Krista   start melody entertainer repeating once in background   repeat 4 times     do       show leds       pause ms 100       show leds   end repeat   clear screen  on button B pressed   stop all sounds   </pre>

## Lunch and Recess

**45 minutes**

### 3. Understanding Logic

**Estimated Time: 30 Minutes**

#### Activity Goals:

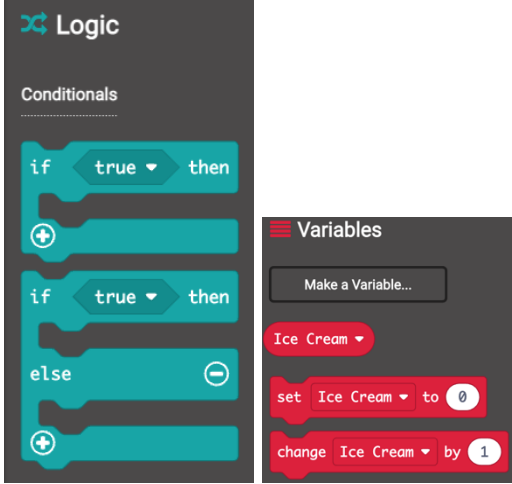
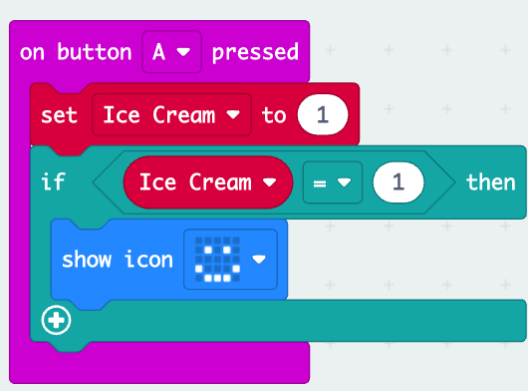
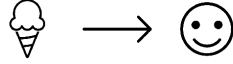
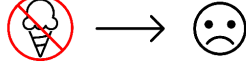
- Build understanding of logic from a programming and computational thinking lens
- Teach students the differences between If/Then statements, If/Else statements, and If/Else If/Else statement and how to use each variation.

#### Activity Procedure:

<p>1. Introduce the Idea of Logic and Variables</p> <ol style="list-style-type: none"><li>Logic is the way a computer can be programmed to make decisions</li><li>A Variable is a placeholder or symbol for a value</li></ol>	
<p>2. Have students analyze the process map of the IF/THEN Code</p> <ol style="list-style-type: none"><li>Remind them of the process map key</li><li>Walk through inputs, outputs, and steps</li><li>IF/THEN: Is for one action to happen in response to an input</li><li>Talk about Variables<ol style="list-style-type: none"><li>Ice Cream is the variable in this example because we are setting it equal to different numbers</li><li>It represents how many ice creams we have</li></ol></li></ol>	<p>Let's talk through examples of Logic. This is how computers are programmed to make decisions based on different inputs or conditions.</p> <p>We are going to talk about Ice Cream.</p> <p><b>Logic</b></p> <p><b>IF/THEN</b> One action to happen in response to an input</p> <p><b>If Ice Cream = 1</b></p> <p>5</p>

**Facilitator Shares Slide and/or directs students to slide in online activity handout workbook**

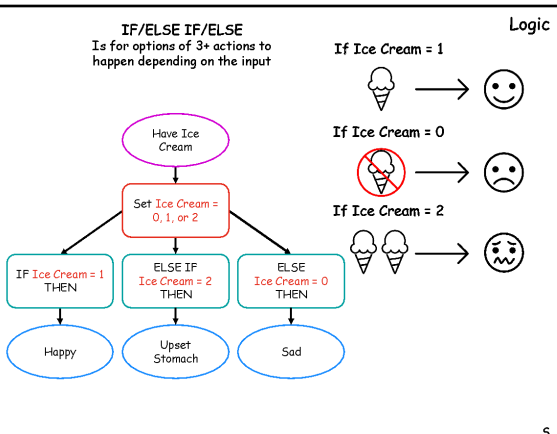


<p>3. Have students explore “Logic” and Variable code blocks on MakeCode to identify which code to use</p> <ol style="list-style-type: none"> <li>Throughout this step ask students to share which code they believe is used</li> <li>Name the Variable “Ice Cream” and set it equal to 1</li> <li>If needed, remind them that icons are in the “Basic” category</li> </ol>	
<p>4. Students sequence the code blocks to program an IF/THEN sequence</p>	
<p>5. Have students test their code using the simulator in MakeCode to check for any errors</p>	
<p>6. Students import the code onto the Micro:bit and see what happens.</p> <ol style="list-style-type: none"> <li>Does it match the process map?</li> <li>What happens when Ice Cream = 1?</li> </ol>	
<p>7. For IF/ELSE statements: repeat the process steps 2-6</p> <ol style="list-style-type: none"> <li>Change the variable to “Pick random 0 to 1” using the <b>Math</b> blocks <ol style="list-style-type: none"> <li>Introduce the math blocks to the students</li> </ol> </li> <li>Ask students why the program would need to have two different numbers the variable could be equal to</li> <li>Have student talk out how is else works.</li> <li>IF/ELSE: Is for two actions to happen depending on the input</li> </ol>	<p>Logic</p> <p>Let's talk through examples of Logic. This is how computers are programmed to make decisions based on different inputs or conditions.</p> <p>We are going to talk about Ice Cream.</p> <p><b>IF/ELSE</b> Is for two actions to happen depending on the input</p> <pre> graph TD     Start([Have Ice Cream]) --&gt; Set[Set Ice Cream to 1 or 0]     Set --&gt; If1{IF Ice Cream = 1 THEN}     Set --&gt; Else1{ELSE Ice Cream = 0 THEN}     If1 --&gt; Happy([Happy])     Else1 --&gt; Sad([Sad])   </pre> <p>If Ice Cream = 1</p>  <p>If Ice Cream = 0</p>  <p>s</p>

```

on button A pressed
  set Ice Cream to pick random 0 to 1
  if Ice Cream = 1 then
    show icon [happy face]
  else
    show icon [sad face]
  
```

8. For IF/ELSE IF/ELSE statements: repeat the process steps 2-6
- Change the variable to **"Pick random 0 to 2"** using the math blocks
  - Ask students why the program would need to have three different numbers the variable could be equal to
  - Ask students to explain how if, else if, and else work – have them talk it out
  - IF/ELSE IF/ELSE: Is for the option of 3+ actions to happen depending on the input



```

on button A pressed
  set Ice Cream to pick random 0 to 2
  if Ice Cream = 1 then
    show icon [happy face]
  else if Ice Cream = 2 then
    show icon [sad face]
  else
    show icon [neutral face]
  
```

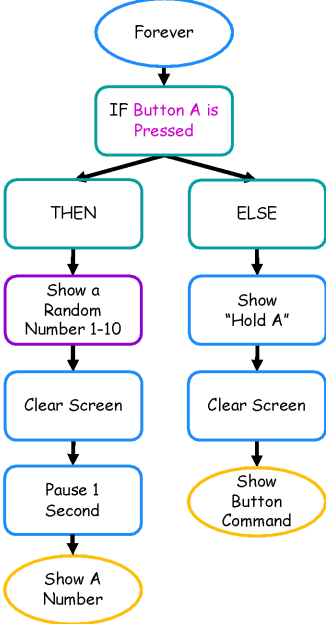
#### 4. Troubleshooting and Debugging

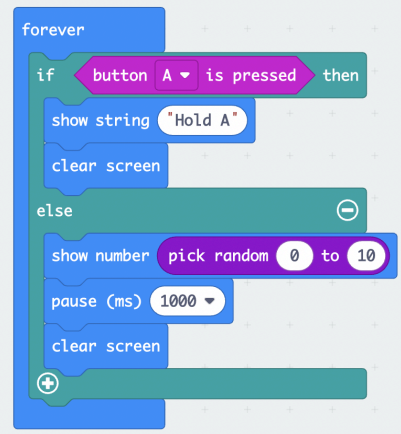
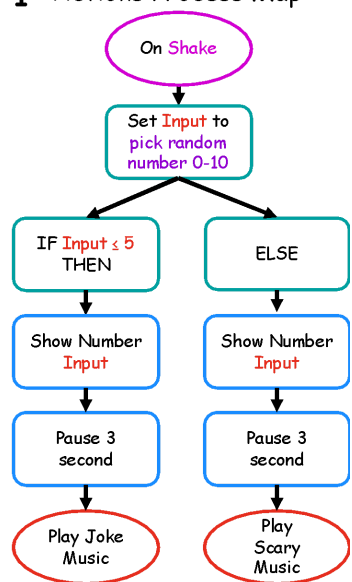
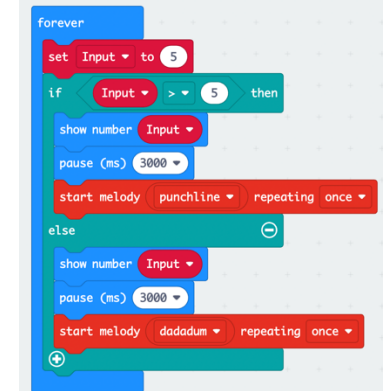
**Estimated Time:** 25 Minutes


##### Activity Goals:

- Introduce the concept of troubleshooting “bugs” – errors in the code
- Use computational thinking (decomposition) to deduce errors in code
- Students can identify errors in code that prevent the Micro:bit to perform its intended function
- Students create a process map and simple code which they will intentionally introduce a bug to
- Students will identify bugs in their partner’s code

##### Activity Procedure:

<p>1. Talk about what happens when there is an error in the code</p> <ol style="list-style-type: none"><li>Did the program provide correct instructions to the computer?</li><li>Errors in code are called “Bugs”</li></ol>	
<p>2. <b>Bug Challenge 1:</b> Students are provided with a process map of what the code should do (actions).</p> <ol style="list-style-type: none"><li>Identify the inputs, outputs, and steps of the process</li></ol> <p><b>Facilitator Shares Slide and/or directs students to slide in online activity handout workbook</b></p>	<p><b>1 Actions Process Map</b></p>  <pre>graph TD; Start([Forever]) --&gt; IF[IF Button A is Pressed]; IF --&gt; THEN[THEN]; IF --&gt; ELSE[ELSE]; THEN --&gt; ShowRN[Show a Random Number 1-10]; ShowRN --&gt; ClearScreen1[Clear Screen]; ClearScreen1 --&gt; Pause[Pause 1 Second]; Pause --&gt; ShowAN[Show A Number]; ELSE --&gt; ShowHA[Show "Hold A"]; ShowHA --&gt; ClearScreen2[Clear Screen]; ClearScreen2 --&gt; ShowBC[Show Button Command];</pre>

<p>3. Then, students assess the code in on the slide that contains a “bug”</p> <ol style="list-style-type: none"> <li>Have them talk through the sequence of the code that they see</li> <li>Have students copy the “bugged” code into Makecode and load it on a Micro:bit</li> </ol>	
<p>4. Have students identify the bug, provide reasoning on why they know it is a bug, and a solution to fix the code.</p>	
<p>5. <b>Bug Challenge 2:</b> Students are provided with a more <b>complex</b> process map of what the code should do (actions).</p> <ol style="list-style-type: none"> <li>Identify the inputs, outputs, and steps of the process</li> </ol> <p><b>Facilitator Shares Slide and/or directs students to slide in online activity handout workbook</b></p>	<p><b>1 Actions Process Map</b></p>  <pre> graph TD     Start([On Shake]) --&gt; SetInput[Set Input to pick random number 0-10]     SetInput --&gt; Decision{IF Input &lt;= 5 THEN}     Decision -- THEN --&gt; ShowNum1[Show Number Input]     ShowNum1 --&gt; Pause1[Pause 3 second]     Pause1 --&gt; PlayJoke([Play Joke Music])     Decision -- ELSE --&gt; ShowNum2[Show Number Input]     ShowNum2 --&gt; Pause2[Pause 3 second]     Pause2 --&gt; PlayScary([Play Scary Music])   </pre>
<p>6. Then, show students the code in Makecode with <b>multiple</b> “bug”</p> <ol style="list-style-type: none"> <li>Have them talk through the sequence of the code that they see</li> <li>Have students copy the “bugged” code into Makecode and load it on a Micro:bit</li> </ol>	

<p>7. Have students identify the bugs, provide reasoning on why they know it is a bug, and a solution to fix the code.</p>	
<p>8. <b>Bug Challenge 3:</b> Create a simple code to with a bug for your partner to find.</p> <ol style="list-style-type: none"> <li>Make a process map of the actions your code is supposed to do.</li> <li>Code your program in MakeCode</li> <li>Switch code with your partner</li> <li>Can you find the bug before you partner?</li> <li>Can you fix the code so it runs correctly?</li> </ol> <p><b>Facilitator Shares Slide and/or directs students to slide in online activity handout workbook</b></p>	<p style="text-align: right;">Troubleshooting and Debugging</p> <div style="display: flex; align-items: center;">  <div style="margin-left: 10px;"> <p><b>Make a bug. Find a bug. Fix the bug.</b></p> <p>Create a simple code to with a bug for your partner to find. Make a process map of the actions your code is supposed to do. Code your program in MakeCode then switch code with your partner. Can you find the bug before you partner? Can you fix the code so it runs correctly?</p> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <span>1    Actions Process Map</span> <span>2    Code</span> </div> </div> </div> <p style="text-align: right;">5</p>

## 5. Problem Solving

**Estimated Time:** 15 Minutes

### Activity Goals:

- Introduce students to problem solving starting with how to identify a problem, create a plan, and how to find a plausible solution to the problem.
- To help students understand that for some problems they need tools like a computer
- Establish connection between problem solving, computational thinking, and the real-world

### Activity Procedure:

<div>1. Introduce what it means to problem solve as an engineer or scientist in the real-world</div> <div><div>a. Connect it to de-bugging done in the previous activity</div><div>b. Computational Thinking: describe a problem, identify the important details needed to solve this problem, break the problem down into small, logical steps, use these steps to create a process (algorithm) that solves the problem, and then evaluate this process.</div></div>																															
<div>2. Students will brainstorm with themselves and their small group ideas of problems or situations that could be solved with the help of a computer.</div> <div><div>a. The camp has a design focus, so it is important to emphasize why problem solving skills are essential in the workforce</div></div>	<div><div>Name:</div><div>Problem Solving</div><table><tr><td>List some situations that you think could be solved with the help of computers.</td><td>Would finding a solution to this problem help the world?</td><td>Would finding a solution to this problem help your local community?</td><td>Would finding a solution to this problem help you or your family?</td><td>Do you think this problem is solvable?</td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table><div>Map a process to solve one of the problems you listed above</div></div>	List some situations that you think could be solved with the help of computers.	Would finding a solution to this problem help the world?	Would finding a solution to this problem help your local community?	Would finding a solution to this problem help you or your family?	Do you think this problem is solvable?																									
List some situations that you think could be solved with the help of computers.	Would finding a solution to this problem help the world?	Would finding a solution to this problem help your local community?	Would finding a solution to this problem help you or your family?	Do you think this problem is solvable?																											
<div>3. Students will then identify the types of problems they listed</div> <div><div>a. Global</div><div>b. Community</div><div>c. Personal/Family</div></div>																															
<div>4. Students will create a process map around a possible way to solve one of the problems they identified.</div> <div><div>a. Share their process with the small group and evaluate the process</div></div>																															

## Day 2: Coding Applications, Intro to Micro:bit Pet

### Overview

Activity	Time, minutes
Welcome	30
Cooding Applications	120
Lunch and Recess	45
Micro:bit Pet Intro	30
Total	225
Extra Time	15

### Materials Needed for Day 2

Technology	Craft Supplies	Activity Supplies
<ul style="list-style-type: none"><li>• Computers</li><li>• Micro:bits - 1 per student</li><li>• AAA batteries</li></ul>	<ul style="list-style-type: none"><li>• Paper</li><li>• Markers</li><li>• Pencils</li><li>• Paper towel/toilet paper rolls</li></ul>	<ul style="list-style-type: none"><li>• Sticky Notes</li><li>• Sticky Easel Pad (optional)</li><li>• Rulers</li><li>• Flashlights</li></ul>

### Activity 1: Welcome

**Estimated Time:** 30 Minutes

### Activity 2: Coding Applications

Coding Application 1: Creating a Rock, Paper, Scissors Game

<https://microbit.org/projects/make-it-code-it/rock-paper-scissors/>

**Estimated Time:** 60 Minutes

#### Activity Goals:

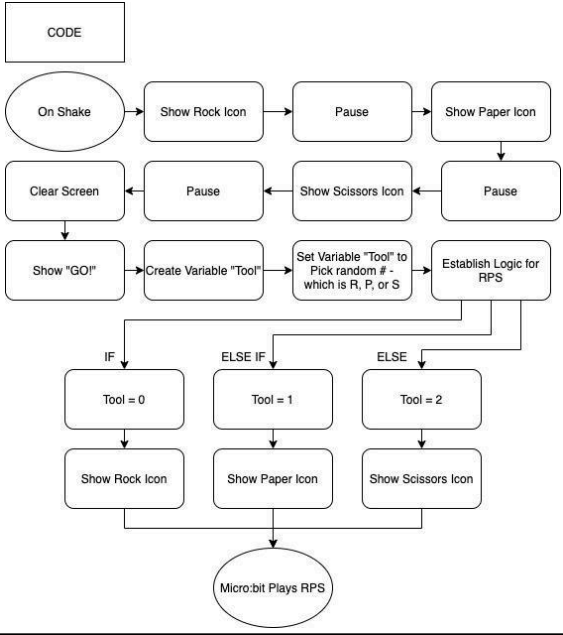
- Students will be able to program a Micro:bit to be able to play a game against it
- Students will apply code they previously learned such as Loops, Logic, Sounds, Variables

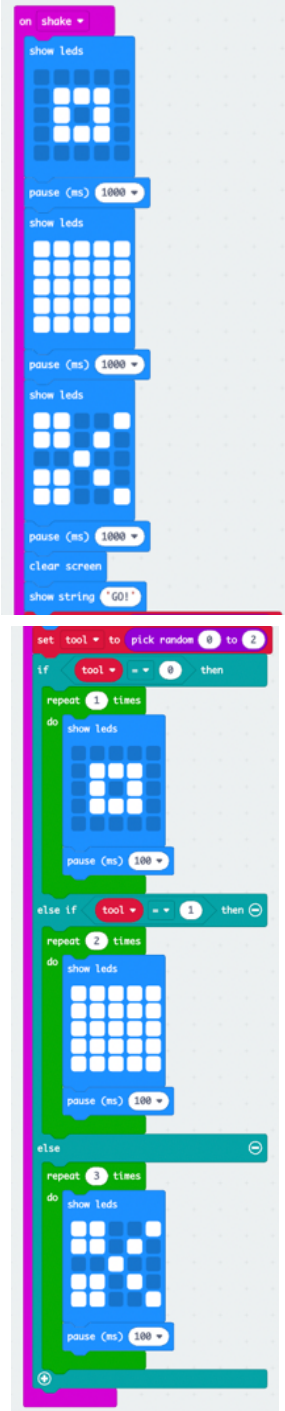
#### Activity Procedure:

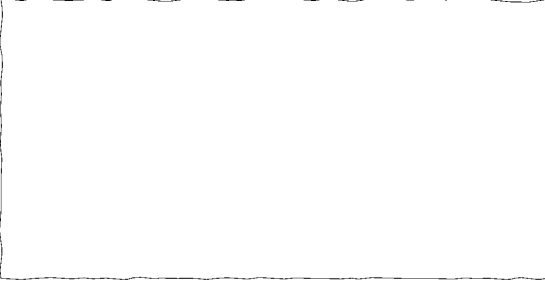
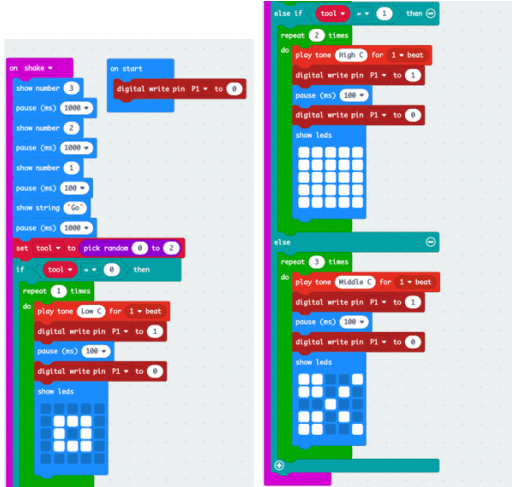
1. Have students play rock, paper, scissors against a partner for 2 rounds	
--	--

<p>2. Introduce the activity: Your challenge is to design a rock, paper, scissors game using the Micro:bit. Provide the Program Requirements Below:</p> <ol style="list-style-type: none"> <li>Micro:bot needs a countdown to start playing the game – just like you say “Rock, Paper, Scissors, Shoot!”</li> <li>Micro:bit needs to randomly choose between Rock, Paper, or Scissors</li> <li>Micro:bit needs to display an icon for Rock, Paper, or Scissors</li> <li>Micro:bit needs an input to know when to start the game</li> </ol> <p><b>Facilitator Shares Slide and/or directs students to slide in online activity handout workbook</b></p>	<div style="border: 1px solid black; padding: 5px; margin: 10px;"> <p><b>Program Requirements</b></p> <ol style="list-style-type: none"> <li>Micro:bot needs a countdown to start playing the game – just like you say “Rock, Paper, Scissors, Shoot!”</li> <li>Micro:bit needs to randomly choose between Rock, Paper, or Scissors</li> <li>Micro:bit needs to display an icon for Rock, Paper, or Scissors</li> <li>Micro:bit needs an input to know when to start the game</li> </ol> </div>
<p>3. Have the students identify the inputs and outputs for the game</p> <ol style="list-style-type: none"> <li>What will start the game? What is the input?</li> <li>What is the end result of the game or the output?</li> </ol> <p>4. Have students think about the actions that have to happen to get from the start to the end result</p> <ol style="list-style-type: none"> <li>Which actions needs to happen for the game to run?</li> <li>Actions Process Map is developed</li> </ol> <p style="text-align: center;">*Example Process Map Shown*</p> <p><b>Printout*</b></p>	<div style="border: 1px solid black; padding: 5px; margin: 10px;"> <p><b>Process Mapping</b></p> <ol style="list-style-type: none"> <li>Review the process map for the <b>ACTIONS</b> needed for the Micro:bit to play Rock, Paper, and Scissors. <ul style="list-style-type: none"> <li>Recall the process map symbols</li> <li>What is the input? What is the output?</li> <li>Which actions do you think need to happen to get the end result?</li> </ul> </li> <li>Create a process map for the <b>CODE</b> you will need to make the actions happen <ul style="list-style-type: none"> <li>What kinds of code blocks will you need?</li> <li>What kind of input do you want use to start the game?</li> <li>What is the sequence the code needs to run in?</li> <li>How could you use Logic to make different actions happen based on different inputs?</li> </ul> </li> </ol> </div> <div style="text-align: center; margin-top: 20px;"> <p><b>Example Process Map</b></p> <p><b>Actions Process Map</b></p> <pre> graph LR     Start([Shake to Start]) --&gt; Countdown[Show Countdown to start game]     Countdown --&gt; PickRandom[Pick Random Rock, Paper, Scissors]     PickRandom --&gt; ShowIcons[Show Different Icons]     ShowIcons --&gt; PlayGame([Play Rock, Paper, Scissors]) </pre> </div>
<p>5. Guide students to take the Actions Process map they built and identify the code blocks they believe they will need. Students will create a CODE process map to match the actions they listed.</p> <ol style="list-style-type: none"> <li>What kinds of code blocks will you need?</li> <li>What kind of input do you want use to start the game?</li> <li>What is the sequence the code needs to run in?</li> </ol>	<p style="text-align: center;">Example Code Process Map</p>



<p>d. How could you use <b>Logic</b> to make different actions happen based on different inputs?</p>	 <pre> graph TD     Start([On Shake]) --&gt; ShowRock[Show Rock Icon]     ShowRock --&gt; Pause1[Pause]     Pause1 --&gt; ShowPaper[Show Paper Icon]     ShowPaper --&gt; Pause2[Pause]     Pause2 --&gt; ShowScissors[Show Scissors Icon]     ShowScissors --&gt; Pause3[Pause]     Pause3 --&gt; ClearScreen[Clear Screen]     ClearScreen --&gt; ShowGO[Show "GO!"]     ShowGO --&gt; CreateTool[Create Variable "Tool"]     CreateTool --&gt; SetTool[Set Variable "Tool" to Pick random # - which is R, P, or S]     SetTool --&gt; EstablishLogic[Establish Logic for RPS]     EstablishLogic --&gt; IF{IF}     IF --&gt; Tool0[Tool = 0]     IF --&gt; ELSEIF{ELSE IF}     ELSEIF --&gt; Tool1[Tool = 1]     ELSEIF --&gt; ELSE{ELSE}     ELSE --&gt; Tool2[Tool = 2]     Tool0 --&gt; ShowRockIcon[Show Rock Icon]     Tool1 --&gt; ShowPaperIcon[Show Paper Icon]     Tool2 --&gt; ShowScissorsIcon[Show Scissors Icon]     ShowRockIcon --&gt; End([Micro:bit Plays RPS])     ShowPaperIcon --&gt; End     ShowScissorsIcon --&gt; End   </pre>
<p>6. Students move to MakeCode editor to build out their code the process mapped</p> <ol style="list-style-type: none"> <li>Use the MakeCode editor to program the Micro:bit to play rock, paper, scissors</li> <li>Test your code periodically using the simulator or the Mirco:bit to identify any bugs or test your sequence.</li> </ol> <p>*Example of Code Shown*</p>	<p>Continuous Code</p>

	
<p>7. Check each group's programming when they are finished to make sure it follows their process maps and results in a complete game of Rock, Paper, Scissors.</p> <ol style="list-style-type: none"> <li>Students then load the program onto their Micro:bits</li> </ol>	

8. Each students goes 2 rounds of Rock, Paper, Scissors against their Micro:bit	
<p>9. <b>Rock, Paper, Scissors Challenge:</b> Challenge the students to add another element to the code of their game. Have them choose from these options:</p> <ol style="list-style-type: none"> <li>Different number of <b>flashes</b> of the Rock, Paper, or Scissor <b>icons</b></li> <li>Play <b>different tones</b> for Rock, Paper, or Scissors</li> <li>Display <b>different lights</b> or <b>flashes of lights</b> for Rock, Paper, or Scissors using an LED</li> </ol>	
<p>10. Have students mark up or add to their process maps to show where their changes will be added in both Actions and Code</p>	<p><b>Process Mapping</b></p> <p>1. Review the process map for the ACTIONS needed for the Micro:bit to play Rock, Paper, and Scissors</p> <ul style="list-style-type: none"> <li>Recall the process map symbols</li> <li>What is the input? What is the output?</li> <li>Which actions do you think need to happen to get the end result?</li> </ul> <p>2. Create a process map for the CODE you will need to make the actions happen</p> <ul style="list-style-type: none"> <li>What kinds of code blocks will you need?</li> <li>What kind of input do you want used to start the game?</li> <li>What is the sequence the code needs to run in?</li> <li>How could you use Logic to make different actions happen based on different inputs?</li> </ul> <p><b>Rock Paper Scissors</b></p>  <p style="text-align: right;">5</p>
<p>11. Students move to MakeCode editor to build out their code</p> <ol style="list-style-type: none"> <li>Use the MakeCode editor to map out code to play rock, paper, and scissors.</li> <li>Test your code periodically using the simulator or the Micro:bit to identify any bugs or test your sequence.</li> </ol> <p><i>*Example code shown with both LED and sounds/tones*</i></p> <p>LIGHTS NEED DIGITAL WRITE PIN</p>	<p><b>Continuous Code</b></p> 
12. Check each group's programming when they are finished to make sure it follows their process maps and results in a complete game of Rock, Paper, Scissors.	

13. Students can then load it onto their Micro:bit if they have not done so already	
14. Each students goes 2 rounds of Rock, Paper, Scissors against their Micro:bit	

## Coding Application 2: Collect Light Intensity Data and Create an Automatic Light

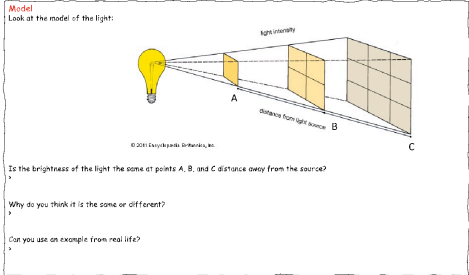
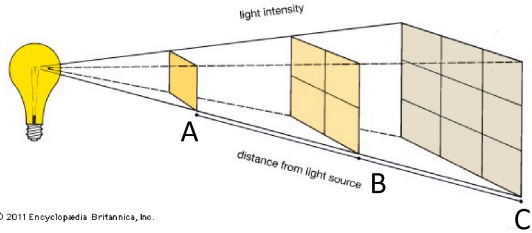
**Estimated Time:** 60 Minutes

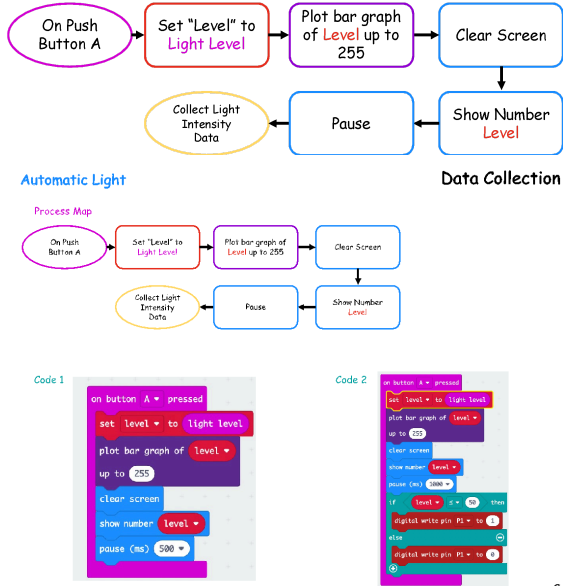

**Materials:** Flashlights for each group, batteries, it helps to use a toilet paper roll or paper towel roll to concentrate the light from flashlight to microbit, tape to map out distances (2 in, 4 in, 6 in, 8 in, 10 in), measuring tape to mark distance, GROUPS of 3

### Activity Goals:

- Students will program the Micro:bits to collect light intensity data using the internal photoresist in the Micro:bit and/or program an external photoresist sensor to collect light intensity values.
- Students will learn to collect, plot, and analyze data for trends using a spreadsheet software (google sheets, excel, etc.)
- Student will establish relationships between light intensity and distance using the data collected and the Inverse Square Law for light intensity.  $I = 1/d^2$   $I$  = light intensity  $d$  = distance from the light source

### Activity Procedure:

1. Introduce the challenge of creating a light that automatically turns on when the brightness reaches a certain level	
<p>2. Elicit student understanding of the concept by showing them the diagram. Have them think about and respond to the following prompts.</p> <ol style="list-style-type: none"> <li>Is the brightness of the light the same at points A, B, and C distance away from the source?</li> <li>Why do you think it is the same or different?</li> <li>Can you use an example from real life?</li> </ol> <p><b>Facilitator Shares Slide and/or directs students to slide in online activity handout workbook</b></p>	<p><b>Automatic Light</b></p> <p>Background Light intensity (lux) is determined by how much light (lumens) there is within a certain area (m<sup>2</sup>).</p> <p><b>Model</b> Look at the model of the light:</p>  <p>© 2011 Encyclopædia Britannica, Inc.</p> <p>Is the brightness of the light the same at points A, B, and C distance away from the source?</p> <p>Why do you think it is the same or different?</p> <p>Can you use an example from real life?</p> <p><b>Data Collection</b></p>  <p>© 2011 Encyclopædia Britannica, Inc.</p>

<p>3. Show students the photoresist sensor and explain how it works to measure light intensity</p> <p>a. When a photoresistor sensors is exposed to light, the resistance decreases so it become more conductive. We can use this change in resistance to measure the intensity of light.</p>	
<p>4. Have students think of the inputs, outputs, and possible steps (actions) that would need to take place in the program</p>	
<p>5. Show students the ACTIONS process map to collect and graph data from a light sensor</p> <p>a. Have them identify what the inputs, outputs, and steps are in the code provided</p>	 <p>The ACTIONS process map shows a sequence of steps: On Push Button A (input), Set "Level" to Light Level (action), Plot bar graph of Level up to 255 (action), Clear Screen (action), Show Number Level (output), Pause (action), and Collect Light Intensity Data (input). The map is divided into 'Automatic Light' and 'Data Collection' sections. Below the map are two code blocks: Code 1 and Code 2. Code 1 is a simplified version of the process map, and Code 2 is a more detailed version with additional steps like 'if level &lt; 10 then' and 'digitalWrite(pin, 1)'.</p>
<p>6. Show students the code to collect and graph data from a light sensor</p> <p>a. Have them identify and match the inputs, outputs, and steps from the process map</p> <p>b. What does the variable level equal?</p> <p>c. What number is going to display on the screen?</p>	 <p>The code block shows the following steps: on button A pressed, set level to light level, plot bar graph of level up to 255, clear screen, show number level, and pause (ms) 500.</p>

7. Have students experiment with data collection using the light sensor in the Micro:bit.

They will EACH take 5 measurements from different distances while aiming directly at the center of the Micro:bit.

Distances = 2 in, 4 in, 6 in, 8in, 10 in

Each student will cycle through each of the roles to support their groupmates: (1) Programmer, (2) Data Collector, (3) Data Recorder

- The Programmer builds the program to collect their light intensity data.
- The programmer will then collect their data with the support of their group members.
- Programmer stands at the first location and shines the flashlight at the LED screen of the Micro:bit where the sensors are. This is important for an accurate measurement.
- The Data Collector will Push A to collect light intensity when the Programmer is ready
- The Data Recorder will write down the light intensity reading from the Micro:bit in the data table
- Repeat steps C-E for the remaining distances away from the light source
- When finished the programmer will download their data from the Device Console in Micro:bit
- Students will rotate roles and Repeat steps A-G until all students have collected and exported their own light intensity data

**Facilitator Shares Slide and/or directs students to slide in online activity handout workbook**

**Printout\* Data Table**

**Automatic Light Data Collection**

**Instructions:** Experiment with collecting data using the light sensor in the Micro:bit.

Each of you will program your Micro:bit to measure light intensity. Each of you will take turns measuring light intensity for 5 different distances. Distances = 1ft, 2ft, 3ft, 4ft, 5ft

You will each cycle through the roles to support your groupmates: (1) Programmer, (2) Data Collector, (3) Data Recorder

- (1) The Programmer builds the program to collect their light intensity data using their Micro:bit. The programmer will collect their data with the support of their group members.  
Data Collection Steps:
  - (1) Programmer stands at the first location and shines the flashlight at the LED screen of the Micro:bit where the sensors are. This is important for an accurate measurement.
  - (2) The Data Collector will Push A to collect light intensity when the Programmer is ready
  - (3) The Data Recorder will write down the light intensity reading from the Micro:bit in the data table
  - (4) Repeat steps 1-3 for the remaining distances. When finished the programmer will download their data from the Device Console in Micro:bit
- (2) Students will rotate roles and Repeat steps 1-4 until all students have collected and exported their own light intensity data
- (3) After the data is collected and exported, each student will import the data into a spreadsheet program: Google Sheets or Excel. Google Sheets, Open New Sheet, File, Import, Select Data from downloads, import data
- (4) Then you will need to rename the Time readings to their respective distances
  - When exporting data from Micro:bit the data is saved as time and light intensity
    - The Times are affiliated with the different distances the students stood at
- (5) Rename the Columns to Distance (Column A) and Light Intensity (Column B)
- (6) Create a chart to display the data using a line graph
- (7) EACH student will explore the relationship between distance and light intensity from their data that they graphed

**Questions:**

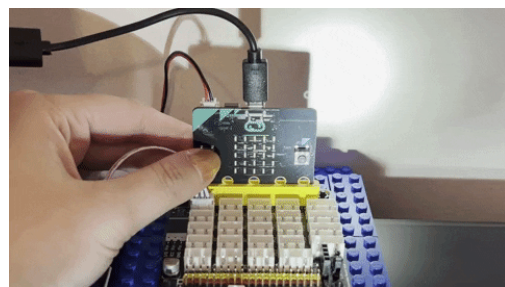
- (1) Does light intensity increase or decrease with distance?
- (2) What is the mathematical relationship between light intensity and distance from your plot?

5

**Programmer Builds the Program**



**Press A to Collect Data**



**Data Collector Records Values for Programmer**

Automatic Light

Data Collection

Programmer 1

Distance	Time	Light Intensity
12 inches		
24 inches		
36 inches		
48 inches		
60 inches		

Programmer 2

Distance	Time	Light Intensity
12 inches		
24 inches		
36 inches		
48 inches		
60 inches		

Programmer 3

Distance	Time	Light Intensity
12 inches		
24 inches		
36 inches		
48 inches		
60 inches		

S

Click “Show console Device” to view live data logging. Located under simulator

Show console Device

Use these icons to pause or download data files from “show console device”

Device

8. Each Student will import the data into a spreadsheet program like Google Sheets or Excel

- Google Sheets, Open New Sheet, File, Import, Select Data from downloads, import data

9. Rename the Time readings to their respective distances from the Micro:bit

- When exporting data from Micro:bit the data is saved as time and light intensity
- The times are affiliated with the different distances the students stood at

10. Rename the Columns to Distance (Column A) and Light Intensity (Column B)

11. Create a chart to display the data using a line graph

12. EACH student will explore the relationship between distance and light intensity from their data that they graphed

- Guide students to look at the relationship between the light intensity and the distance away

Raw Data

sep=	
time (source1)	Light Intesity
75.44	243
117.65	233
164.575	114
210.051	63
266.122	26

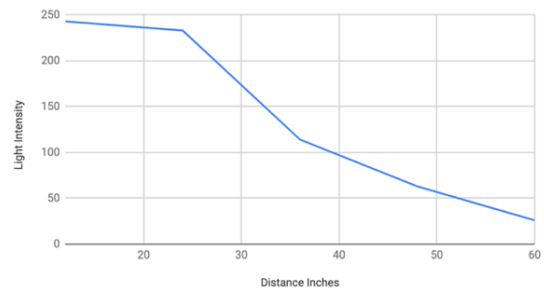
Change Time to related distance

sep=	
Distance Inches	Light Intensity
12	243
24	233
36	114
48	63
60	26

Graph Distance vs Light Intensity

- b. Does light intensity increase or decrease with distance?
- c. What is the mathematical relationship between light intensity and distance from your plot?

Light Intensity vs. Distance Inches



Automatic Light Data Collection

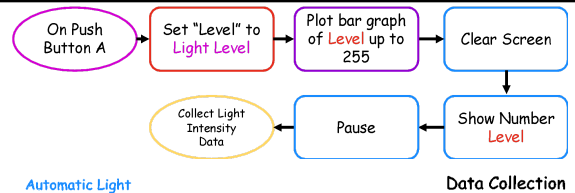
**Programmer**

- (1) Does light intensity increase or decrease with distance?
- >
- (2) What is the mathematical relationship between light intensity and distance from your plot?
- >

S

**13. NOW create an automatic light that turns on at a certain light intensity or distance from a light source**

14. Look at the original process map for the code. What actions would you add and where would you add them to create an automatic light?
15. What could the code and sequence look like for that?



**Process Map**



**Code 1**

```

on button A = pressed
  set level to light level
  plot bar graph of level
  up to 255
  clear screen
  show number level
  pause ms 500

```

**Code 2**

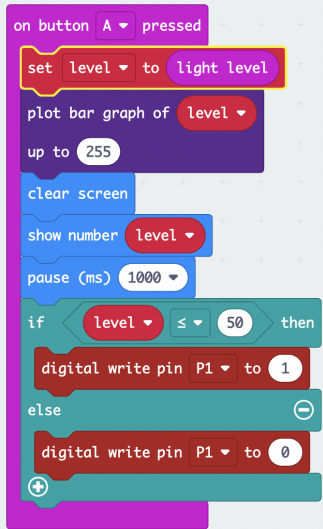
```

on button A = pressed
  set level to light level
  plot bar graph of level
  up to 255
  clear screen
  show number level
  pause ms 500
  if level < 10 then
    digital write pin P1 to 1
  else
    digital write pin P1 to 0
  end

```

S



<p>16. Show the students the updated code that includes an automatic light that turns on at a certain threshold.</p> <ol style="list-style-type: none"> <li>Ask the students where they see the changes in ACTIONS and in CODE</li> <li>What actions does the new code show when run?</li> <li>What tool is Pin P1?</li> <li>What do you think happens when P1 = 1? What happens when P1 = 0?</li> </ol>	 <pre> on button A pressed   set level to light level   plot bar graph of level   up to 255   clear screen   show number level   pause (ms) 1000   if level &lt;= 50 then     digital write pin P1 to 1   else     digital write pin P1 to 0 </pre>
<p>17. Using your data from the first experiment, estimate what distance you think the light will turn on at:</p> <ol style="list-style-type: none"> <li>What is your light intensity threshold?</li> <li>What distance do you think the light will turn on at?</li> </ol> <p>18. Have students repeat the experiment from above with the new code.</p> <ol style="list-style-type: none"> <li>Have them note at what distance and light intensities the LED light turns on at</li> <li>Make sure students download the data and graph the results accordingly</li> </ol>	

## Lunch and Recess

**45 minutes**

Activity 3: Design a Micro:bit Pet – Intro to Pet Design Challenge, Find Partner, Interview  
<https://mgraftin.edublogs.org/2020/06/21/designing-microbit-virtual-pets-monsters-so-many-possibilities/#.YWXHMBDMI56>

**Estimated Time:** Intro Design Pet Challenge – 5 minutes, Find Partner – 5 minutes, Empathize – 20 minutes. **Total Time = 30 minutes.**

### Activity Goals:

- Students will learn about and experience a design challenge and how the engineering design process is an essential component in completing design challenges
- Students will use the design thinking process: Empathize, Define, Ideate, Prototype, Test, Refine
- Students will learn to identify user needs and how to transform those needs into a design plan
- Students will use process mapping to map out the actions, purpose, and functions of the design
- Students will use and apply previous programming skills

### Activity Procedure:

<p>1. <b>Micro:bit Pet Design Introduction</b> (5 minutes)</p> <p>Introduce the goal of designing a Micro:bit pet for your partner. There will be an adoption contest at the end of the project.</p> <p><b>Facilitator directs students to slide in online activity handout workbook</b></p> <p><b>Printout* Ideate Page, Refinements Page</b></p>	
<p>2. <b>Find a Partner</b> (5 minutes)</p> <p>First students need to find a partner that they are going to design a Micro:bit pet for</p> <ol style="list-style-type: none"><li>Someone they did not work with yesterday</li><li>Someone who is on the other side of the room</li><li>With partner play 2 truths and a lie icebreaker and introduce their partner to the class</li></ol>	

<p><b>3. Empathize Interview Partner (20 minutes)</b></p> <p>Students will introduce themselves to their partners. They will proceed to Interview each other about what they would like in a Micro:bit pet. Then they will switch who gets interviewed and who is interviewing. (15 mins)</p> <ol style="list-style-type: none"> <li>What kind of pets does your partner want? (an Animal? Insect?)</li> <li>What are the attributes they would like in their pet? <ol style="list-style-type: none"> <li>Looks:</li> <li>Size:</li> <li>Emotions:</li> <li>Movements:</li> <li>Sounds:</li> </ol> </li> <li>How does the owner want to interact with their pet?</li> </ol>	<div> <div> Designer Name:  Pet Owner Name: </div> <div> Design a Micro:bit Pet  Empathize: Get to know your product users  Before you start designing a Micro:bit pet, you need to learn what its future owner would like in a robot pet. Interview your partner to learn about the kind of pet they would like to have designed for them.  What kind of pets does your partner want? (an Animal? Insect?)   What are the attributes they would like in their pet?   Looks:   Size:   Emotions:   Movements:   Sounds:   How does the owner want to interact with their pet? </div> <div> Design Plan - Micro:bit Pet </div> </div>
---	--

## Day 3: Design a Micro:bit Pet - Define, Ideate, Feedback, Prototype 1, Feedback, Prototype 2, Feedback, Final Design

### Overview

Activity	Time, minutes
Welcome	10
Design a Micro:bit Pet	170
Lunch and Recess	45
Total	225
Extra Time	15

### Materials Needed for Day 3

Technology	Craft Supplies	Activity Supplies
<ul style="list-style-type: none"> <li>Computers</li> <li>Micro:bits - 1 per student</li> <li>Servo motors</li> <li>Dupont Wires (male –female, male – male)</li> <li>Alligator Clips</li> <li>AAA Batteries</li> </ul>	<ul style="list-style-type: none"> <li>Paper</li> <li>Tape</li> <li>Markers</li> <li>Pencils</li> <li>Scissors</li> <li>Construction paper</li> </ul>	<ul style="list-style-type: none"> <li>Sticky Notes</li> <li>Sticky Easel Pad (optional)</li> <li>Rulers</li> <li>Student Rewards</li> </ul>

<ul style="list-style-type: none"> <li>● Micro:bit Stem Kits – sensors, LED lights, motors</li> </ul>	<ul style="list-style-type: none"> <li>● Paper towel/toilet paper rolls</li> <li>● Cardboard/Cardstock</li> <li>● Other craft materials for design projects</li> </ul>	
---	--	--

## Activity 1: Welcome

**Estimated Time:** 30 Minutes

Ice Breaker: What's my name – allocating memory

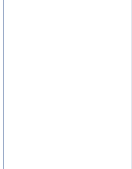

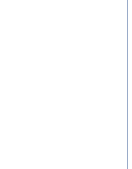



## Activity 2: Design a Micro:bit Pet – Define, Ideate, Feedback, Prototype 1, Feedback, Prototype 2, Feedback, Final Design

**Estimated Time:** Define – 10 minutes, Ideate/Brainstorm – 40 minutes, Ideate Feedback – 10 minutes, Pet Prototype 1 – 30 minutes, Pet Prototype 1 Feedback – 15 minutes, Pet Prototype 2 – 30 minutes, Pet Prototype 2 Feedback – 15 minutes, Pet Final Design – 20 minutes. **Total Time = 170 minutes**

<p><b>4. Define Design Criteria (20 minutes)</b></p> <p>After the interview, each student will define the design criteria: the wants and needs of the user.</p> <ol style="list-style-type: none"> <li>Create a problem statement of user's needs. Finish the sentence "Pet owner needs a pet that..."</li> <li>Which Micro:bit inputs and features do you need to use? (Buttons, LEDs, Accelerometer, etc.)</li> <li>What are the required tools and materials needed for the project?</li> <li>(Add-ons to Micro:bit: buttons, lights, paper, tape, drawing tools, etc.)</li> <li>Are there any constraints to the design based on resources?</li> <li>Are there any other details you need to ask the other team about for their pet?</li> </ol>	<div> <div> Designer Name: Pet Owner Name: </div> <div> <b>Design a Micro:bit Pet</b>  Define: the design criteria for the product   User Problem Statement: "Pet Owner Name" needs a pet that...   Which Micro:bit inputs and features do you need to use? (Buttons, LEDs, Accelerometer, etc.)   What are the required tools and materials needed for the project?  (Add-ons to Micro:bit: buttons, lights, paper, tape, drawing tools, etc.)   Are there any constraints to the design based on resources?   Are there any other details you need to ask the other team about for their pet? </div> <div> <b>Design Plan - Micro:bit Pet</b> </div> </div>
---	---

### 5. Ideate Micro:bit Pet Designs (40 minutes)

Students then move into the Ideation/Brainstorming stage where they will draw 4 versions of the Micro:bit pet and label the parts of the pet. They will also map out the actions the pet will do – emotions, icons, sounds etc.

Designer Name:		Design a Micro:bit Pet		Design Plan - Micro:bit Pet	
Ideate: Brainstorming the initial design					
Draw 4 versions of what the Micro:bit pet could look like based on the interview. Label the parts of the pet.					
					
					
Draw what each of the actions (emotions, movements, etc.) will look like. Note in the drawing which inputs would start each of the actions.					

### 6. Ideate Feedback (10 minutes)

Students will approach a group leader and another student for feedback on their design ideas.

- Students will approach another group leader to ask for feedback about their four design ideas. They will use the four-square framework as a guide:
  - What I like about the design?
  - What I do not like about the design?
  - Questions I have about the design.
  - I want the creator to know.
- After talking with the group leader, the students will brainstorm on how to refine their design.
- Students will approach their partner (the user) to ask for feedback about their four design ideas. They will use the four-square framework as a guide:
  - What I like about the design?
  - What I do not like about the design?
  - Questions I have about the design.
  - I want the creator to know.
- After talking to their partner (the user), the students will brainstorm on how to refine their design.

Designer Name:		Design a Micro:bit Pet		Design Plan - Micro:bit Pet	
Pet Owner Name:					
Group Leader Name:					
Test and Refine: Feedback from users to make refinements					
Group Leader Checkpoint: Get feedback from another group leader in the room. Make any design changes after the feedback.					
What I like about the design		Questions I have about the design			
What I do not like about the design		I want the creator to know...			
For Design Group - Identify Refinements					
What I agree with about the feedback:					
What I disagree with about the feedback:					
What is going to change about the design in the next version:					

Designer Name:		Design a Micro:bit Pet		Design Plan - Micro:bit Pet	
Pet Owner Name:					
Feedback Partner Name:					
Test and Refine: Feedback from users to make refinements					
Ideate User Feedback: Get feedback from your partner (the user). Make any design changes after the feedback.					
What I like about the design		Questions I have about the design			
What I do not like about the design		I want the creator to know...			
For Design Group - Identify Refinements					
What I agree with about the feedback:					
What I disagree with about the feedback:					
What is going to change about the design in the next version:					

## Lunch and Recess

45 minutes

### 7. **Prototype 1** (30 minutes)

The students will review the feedback from the group leader and their partner from the previous and use it to start building their first prototype for their Micro:bit pet.

- The students will start by creating a process map of the actions for their Micro:bit.
- The students will identify the codes they need for the actions and the sequence of codes.
- Once they have identified the codes and sequence, they will use MakeCode editor to program their Micro:bit.

**Design a Micro:bit Pet**      **Design Plan - Micro:bit Pet**  
 Prototype: Building and testing different designs

**PROTOTYPE 1**

**Instructions:**

- Grab a blank sheet of paper and label it: **PROTOTYPE 1** with your name at the top
- Create 2 process maps for the (1) **ACTIONS** of the Micro:bit pet and the (2) **CODE** you are going to use to create the pet. Label the process maps.
- Begin building a prototype of your Micro:bit pet based on the design ideas you came up with in the ideate stage.
- Add any additional resources you may have used to the **DEFINE** page of the design plan.
- Answer the questions on the **REFINEMENT** page to track the changes in your design.

**Example Paper**

Name	Prototype 1
Actions Process Map	
Code Process Map	

### 8. **Prototype 1 Feedback** (15 minutes)

Students will approach their partner (the user) to ask for feedback about their Micro:bit pet prototype -1. They will provide feedback simultaneously to each other. They will use the four-square framework as a guide:

- What I like about the design?
- What I do not like about the design?
- Questions I have about the design.
- I want the creator to know.
- After talking to their partner (the user), the students will brainstorm on how to refine their design.

Designer will reflect on the refinements they made from the ideate to the prototype stage. They will capture their thoughts on using the refinement chart.

**Designer Name:** \_\_\_\_\_ **Design a Micro:bit Pet**      **Design Plan - Micro:bit Pet**  
**Pet Owner Name:** \_\_\_\_\_  
**Feedback Partner Name:** \_\_\_\_\_

Test and Refine: Feedback from users to make refinements

**Prototype 1 User Feedback:** Get feedback from your partner (the user). Make any design changes after the feedback.

What I like about the design >	Questions I have about the design >
What I do not like about the design >	I want the creator to know... >

**For Design Group - Identify Refinements**

What I agree with about the feedback:  
 >

What I disagree with about the feedback:  
 >

What is going to change about the design in the next version:  
 >

**Designer Name:** \_\_\_\_\_ **Design a Micro:bit Pet**      **Design Plan - Micro:bit Pet**  
**Pet Owner Name:** \_\_\_\_\_

Refinement: Making and tracking changes to the design

Stage	What stayed the same from your previous design? How does it support the problem statement?	What changed from your previous design? How does it support the problem statement?
Ideate to Prototype 1		
Prototype 1 to Prototype 2		
Prototype 2 to Final Design		

## 9. **Prototype 2** (30 minutes)

The students will review the feedback from their partner and use it to start building their second prototype for their Micro:bit pet.

- The students will modify their process map of the actions for their Micro:bit based on the feedback they received.
- The students will modify the codes they need for the actions and the sequence of codes.
- Once they have identified the codes and sequence, they will use MakeCode editor to program their Micro:bit.

**Design a Micro:bit Pet**      **Design Plan - Micro:bit Pet**  
 Prototype: Building and testing different designs  
**PROTOTYPE 2**

**Instructions:**

- Grab a blank sheet of paper and label it: **PROTOTYPE 2** with your name at the top
- Create 2 process maps for the (1) **ACTIONS** of the Micro:bit pet and the (2) **CODE** you are going to use to create the pet. Label the process maps.
- Begin building a prototype of your Micro:bit pet based on the design ideas you came up with in the Ideate stage.
- Add any additional resources you may have used to the **DEFINE** page of the design plan.
- Answer the questions on the **REFINEMENT** page to track the changes in your design.

**Example Paper**

Name	Prototype 2
<b>Actions Process Map</b>   <b>Code Process Map</b>	

## 10. **Prototype 2 Feedback** (15 minutes)

Students will approach their partner (the user) to ask for feedback about their Micro:bit pet prototype -1. They will provide feedback simultaneously to each other. They will use the four-square framework as a guide:

- What I like about the design?
- What I do not like about the design?
- Questions I have about the design.
- I want the creator to know.
- After talking to their partner (the user), the students will brainstorm on how to refine their design.

Designer will reflect on the refinements they made from the ideate to the prototype stage. They will capture their thoughts on using the refinement chart.

**Design a Micro:bit Pet**      **Design Plan - Micro:bit Pet**  
 Designer Name: \_\_\_\_\_  
 Pet Owner Name: \_\_\_\_\_  
 Feedback Partner Name: \_\_\_\_\_  
 Test and Refine: Feedback from users to make refinements

**Prototype 2 User Feedback:** Get feedback from your partner (the user). Make any design changes after the feedback.

What I like about the design >	Questions I have about the design >
What I do not like about the design >	I want the creator to know... >

**For Design Group - Identify Refinements**  
 What I agree with about the feedback:  
 >  
 What I disagree with about the feedback:  
 >  
 What is going to change about the design in the next version:  
 >

**Design a Micro:bit Pet**      **Design Plan - Micro:bit Pet**  
 Designer Name: \_\_\_\_\_  
 Pet Owner Name: \_\_\_\_\_  
 Refinement: Making and tracking changes to the design

Stage	What stayed the same from your previous design? How does it support the problem statement?	What changed from your previous design? How does it support the problem statement?
Ideate to Prototype 1		
Prototype 1 to Prototype 2		
Prototype 2 to Final Design		

<p><b>11. Final Design (20 minutes)</b></p> <p>The students will review the feedback from their partner and use it to start building their final design for their Micro:bit pet.</p> <ol style="list-style-type: none"> <li>The students will modify their process map of the actions for their Micro:bit based on the feedback they received.</li> <li>The students will modify the codes they need for the actions and the sequence of codes.</li> <li>Once they have identified the codes and sequence, they will use MakeCode editor to program their Micro:bit.</li> </ol>	<div style="text-align: right;"> <b>Design a Micro:bit Pet</b>      <b>Design Plan - Micro:bit Pet</b> </div> <p style="text-align: center;">Prototype: Building and testing different designs <b>Final Design</b></p> <p><small>Instructions:</small></p> <ol style="list-style-type: none"> <li>Grab a blank sheet of paper and label it: <b>FINAL DESIGN</b> with your name at the top</li> <li>Create 2 process maps for the (1) <b>ACTIONS</b> of the Micro:bit pet and the (2) <b>CODE</b> you are going to use to create the pet. Label the process maps.</li> <li>Begin building a prototype of your Micro:bit pet based on the design ideas you came up with in the ideate stage.</li> <li>Add any additional resources you may have used to the <b>DEFINE</b> page of the design plan.</li> <li>Answer the questions on the <b>REFINEMENT</b> page to track the changes in your design.</li> </ol> <p><small>Example Paper</small></p> <div style="border: 1px solid black; padding: 10px; margin-top: 10px;"> <p style="text-align: center;">Name <span style="float: right;">Final Design</span></p>   <p>Actions Process Map</p>    <p>Code Process Map</p> </div>
---	---

## Day 4: Design a Micro:bit Pet, Micro:bit Pet Extension, and Intro to Design Challenge

### Overview

Activity	Time, minutes
Welcome and Ice Breaker	10
Micro:bit Pet Advertisement	55
Lunch and Recess	45
Micro:bit Pet Extension	35
Intro to Design Challenge	95
Total	240
Extra Time	0

### Materials Needed for Day 4

Technology	Craft Supplies	Activity Supplies
<ul style="list-style-type: none"> <li>Computers</li> <li>Micro:bits - 1 per student</li> <li>Servo motors</li> <li>Dupont Wires (male –female, male – male)</li> <li>Alligator Clips</li> <li>AAA Batteries</li> <li>Micro:bit Stem Kits – sensors, LED lights, motors</li> </ul>	<ul style="list-style-type: none"> <li>Paper</li> <li>Tape</li> <li>Markers</li> <li>Pencils</li> <li>Scissors</li> <li>Construction paper</li> <li>Paper towel/toilet paper rolls</li> <li>Cardboard/Cardstock</li> <li>Other craft materials for design projects</li> </ul>	<ul style="list-style-type: none"> <li>Sticky Notes</li> <li>Sticky Easel Pad (optional)</li> <li>Rulers</li> <li>Student Rewards</li> </ul>



## Activity 1: Welcome

**Estimated Time:** 10 Minutes

## Activity 2: Design a Micro:bit Pet - Create and Adoption Flyer

**Estimated Time:** 55 Minutes

<p><b>12. Presentation: Adoption Advertisement</b> (55 minutes)</p> <p>Students will create a flyer using Word or PowerPoint to get their pet adopted. They will use the four-point frame as a guide in creating the flyer. In the flyer, the students will include the following:</p> <ol style="list-style-type: none"> <li>Description of their pet and instructions on people can interact with the pet.</li> <li>Special attributes and features pf the pet they want to showcase.</li> <li>Picture of their pet.</li> <li>Personal story about their experience with their pet.</li> </ol>	<p style="text-align: center;"><b>Adoption Advertisement</b></p> <ul style="list-style-type: none"> <li>Create a flyer advertisement using Word or PowerPoint to showcase the pet you designed for adoption.</li> <li>The advertisement has to have the following information: <ul style="list-style-type: none"> <li><b>Description</b> <ul style="list-style-type: none"> <li>A written description of the Micro:bit pet to create vivid images for the reader. <ul style="list-style-type: none"> <li>Use facts, specific information that can be proven, in the description.</li> </ul> </li> <li>Describe how to interact with the pet <ul style="list-style-type: none"> <li>Which inputs to use</li> </ul> </li> </ul> </li> <li><b>Special Attributes and Features</b> <ul style="list-style-type: none"> <li>Provide an example of something you think people would want to know about your Micro:bit pet.</li> <li>Write an explanation to connect the design of your pet to the wants and needs of the users.</li> </ul> </li> <li><b>Picture</b> <ul style="list-style-type: none"> <li>Image of your Micro:bit Pet</li> </ul> </li> <li><b>Personal Story</b> <ul style="list-style-type: none"> <li>Write a personal anecdote, a short story, about an experience you have with the Micro:bit pet.</li> </ul> </li> </ul> <li><b>AFTER</b> you create the flyer for the Micro:bit pet you designed, <b>RECORD</b> a 3-minute video advertisement for the Micro:bit pet on Flipgrid.</li> <li>The recording should be a video version of the flyer. It should include you talking about: <ul style="list-style-type: none"> <li><b>Introduce the problem statement</b></li> <li><b>Introduce the Micro:bit pet</b> <ul style="list-style-type: none"> <li>Description of the pet</li> <li>How to interact with the pet</li> <li>Special attributes</li> </ul> </li> <li><b>Why you should adopt this pet</b> <ul style="list-style-type: none"> <li>A personal story/anecdote</li> </ul> </li> <li><b>How it met the owner's wants and needs</b></li> <li><b>A challenge you had to overcome when designing the pet from ideas to real-life</b></li> </ul> </li> </li></ul>				
<p><b>13.</b> Students will then present their advertisement to the class for the Micro:bit Pet they designed.</p> <ol style="list-style-type: none"> <li>The presentation should include you talking about:</li> <li><b>Introduce the problem statement</b></li> <li>Introduce the Micro:bit pet</li> <li>Description of the pet</li> <li>How to interact with the pet</li> <li>Special attributes</li> <li><b>Why you should adopt this pet</b></li> <li>A personal story/anecdote</li> <li><b>How it met the owner's wants and needs</b></li> <li><b>A challenge you had to overcome when designing the pet from ideas to real-life</b></li> </ol>	<p style="text-align: center;"><b>Adoption Advertisement</b></p> <p style="text-align: center;">Pet Name: _____</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; vertical-align: top; padding: 5px;"> <p style="text-align: center;"><b>Description</b></p> <ul style="list-style-type: none"> <li>Write a description of the Micro:bit pet to create vivid images for the reader.</li> <li>Use facts, specific information that can be proven, throughout the description.</li> <li>How to interact with the pet <ul style="list-style-type: none"> <li>Which inputs to use</li> </ul> </li> </ul> </td><td style="width: 50%; vertical-align: top; padding: 5px;"> <p style="text-align: center;">Picture</p> </td></tr> <tr> <td style="width: 50%; vertical-align: top; padding: 5px;"> <p style="text-align: center;"><b>Special Attributes and Features</b></p> <ul style="list-style-type: none"> <li>Provide an example of something you think people would want to know about your Micro:bit pet.</li> <li>Write an explanation to connect the design of your pet to the wants and needs of the users.</li> </ul> </td><td style="width: 50%; vertical-align: top; padding: 5px;"> <p style="text-align: center;"><b>Personal Story</b></p> <ul style="list-style-type: none"> <li>Write a personal anecdote, a short story, about an experience you have with the Micro:bit pet.</li> </ul> </td></tr> </table>	<p style="text-align: center;"><b>Description</b></p> <ul style="list-style-type: none"> <li>Write a description of the Micro:bit pet to create vivid images for the reader.</li> <li>Use facts, specific information that can be proven, throughout the description.</li> <li>How to interact with the pet <ul style="list-style-type: none"> <li>Which inputs to use</li> </ul> </li> </ul>	<p style="text-align: center;">Picture</p>	<p style="text-align: center;"><b>Special Attributes and Features</b></p> <ul style="list-style-type: none"> <li>Provide an example of something you think people would want to know about your Micro:bit pet.</li> <li>Write an explanation to connect the design of your pet to the wants and needs of the users.</li> </ul>	<p style="text-align: center;"><b>Personal Story</b></p> <ul style="list-style-type: none"> <li>Write a personal anecdote, a short story, about an experience you have with the Micro:bit pet.</li> </ul>
<p style="text-align: center;"><b>Description</b></p> <ul style="list-style-type: none"> <li>Write a description of the Micro:bit pet to create vivid images for the reader.</li> <li>Use facts, specific information that can be proven, throughout the description.</li> <li>How to interact with the pet <ul style="list-style-type: none"> <li>Which inputs to use</li> </ul> </li> </ul>	<p style="text-align: center;">Picture</p>				
<p style="text-align: center;"><b>Special Attributes and Features</b></p> <ul style="list-style-type: none"> <li>Provide an example of something you think people would want to know about your Micro:bit pet.</li> <li>Write an explanation to connect the design of your pet to the wants and needs of the users.</li> </ul>	<p style="text-align: center;"><b>Personal Story</b></p> <ul style="list-style-type: none"> <li>Write a personal anecdote, a short story, about an experience you have with the Micro:bit pet.</li> </ul>				

<p>14. Students will listen to student presentations.</p> <ol style="list-style-type: none"> <li>At the end students will have a “pet parade” and put their pets at the front of the class.</li> <li>Students will each get 3 sticky notes to vote for their 3 favorite pets.</li> </ol> <p>If available, Rewards will be given for the top 3 pets.</p>	

### Activity 3: Design a Micro:bit Pet – Extension

**Estimated Time:** Introduction – 10 minutes, Choose and do extension - 25 mins. **Total time = 35 minutes.**

#### Activity Goals:

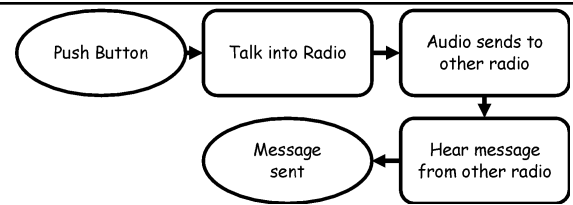
- Allow students more time to work with each other before going into design teams
- Provide students the opportunity to explore more advanced programming in Micro:bit: Bluetooth communication, movement with servo motors, and reactions to sensors

#### Activity Procedure:

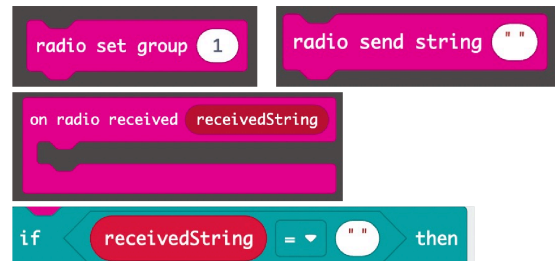
<p><b>Extension Activity Introduction</b> (10 minutes)</p> <p>Welcome students to the project and share with them that we will be learning 4 new tools for Micro:bit. The goal for this mini project is to explore new programs and bring different skills to their future design teams.</p> <p>The new programs are:</p> <ol style="list-style-type: none"> <li>Bluetooth communication</li> <li>Movement with servo motors</li> <li>Reactions to sensors (2 types)</li> </ol> <p>Students identify which out of the three makes most sense with their Micro:bit pet design. They will then learn how to program the new element. (20 mins)</p>	
<p><b>Bluetooth Communication</b></p> <ul style="list-style-type: none"> <li>• Have students brainstorm about the inputs, outputs, and steps a walkie talkie uses.</li> </ul>	Process Map

- This should look like: Push button to talk, sends audio, hear message
- Guide students to think about how the Micro:bits could talk to each other through signals like the walkie talkies
  - Create a process map of the actions for the program
- Show students the essential code blocks under “Radio” and let them talk through what each block may do
- Have students set up the basic code sequence in MakeCode and have them use it on the simulator and between each other.
- **\*Make sure each pair of students is on a different radio number\***
  - Have them talk through what expected to see and what they saw between the two Micro:bits
- Invite the students to expand on that code with a partner to have their pets send and receive different emotions when different inputs are triggered.
  - Need to have matching radio numbers
  - Need to send “strings” as inputs for different actions
  - Process map your actions and codes
  - Add loops and logic to support your actions

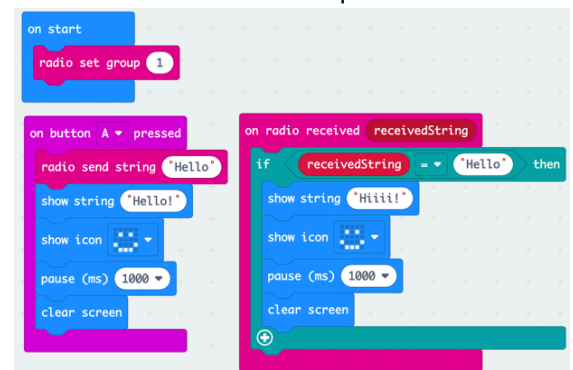
\*Example code on next page

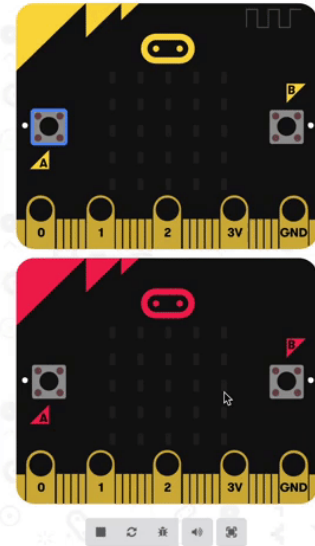


### Code Essentials:

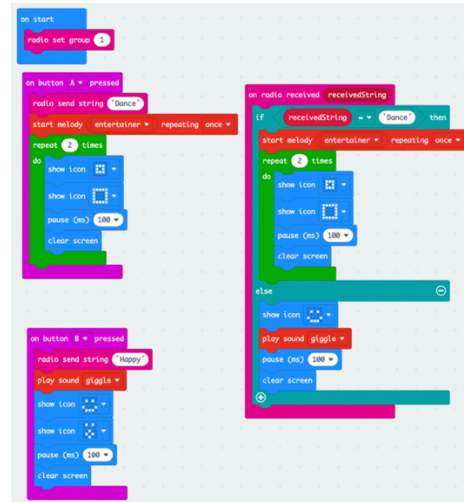


### Basic Radio Sequence



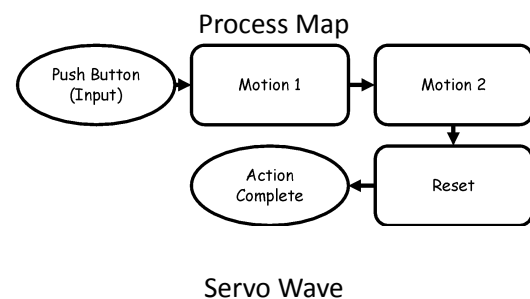


Example Communication Sequence

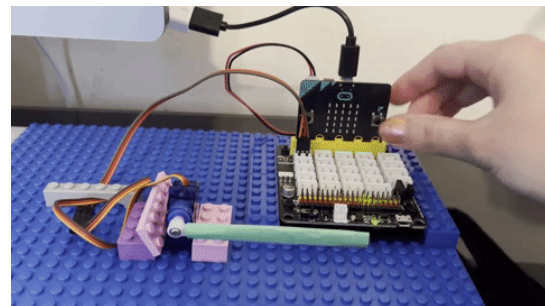
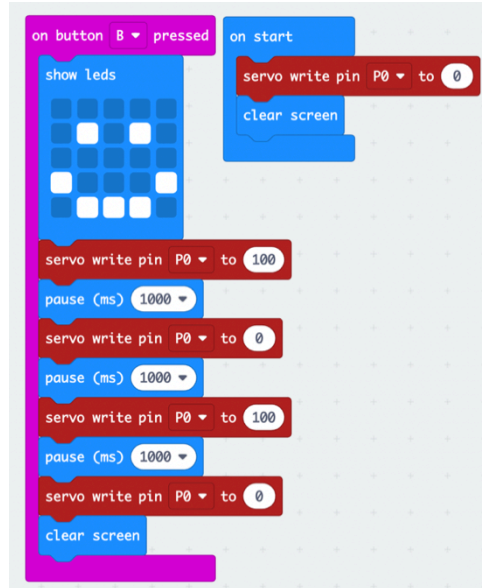


### Movement with Servo Motors

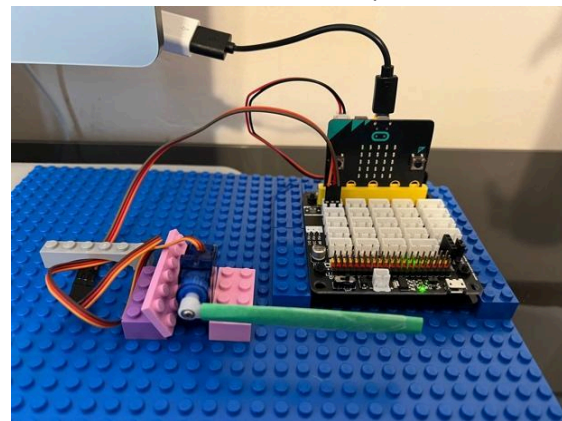
- Have students brainstorm about the inputs, outputs, and steps would be to program movement
  - This should look like: input, motion 1, motion 2, reset, output
- Show students the essential “Servo Write” code blocks under “Pins” and let them talk through what each block may do



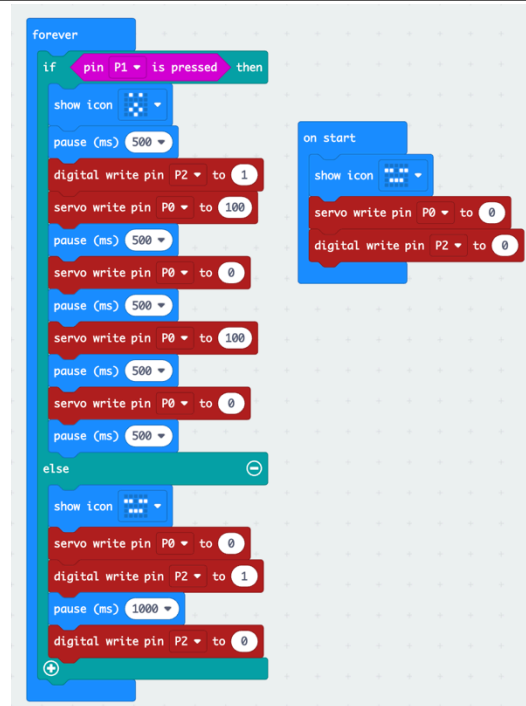
- Before we have only used 0 and 1 to control a device on a pin
- What could the numbers in “Servo Write to ###” mean?
  - Location to move to
- Have students set up the basic code sequence in MakeCode and have them try it out on the Micro:bit
  - Ensure the pin the connect the Servo to is the correct pin
  - Show students how to connect the servo to the board using the male to female cables. Explain that they need to match the connection to the connection on the board – ie. ground to ground
- Then invite the students to expand on that code to have their pets do different movements with different triggers
- Code could look like something below
  - When a button on P1 is pressed, the show surprised face, wave and blink the LED



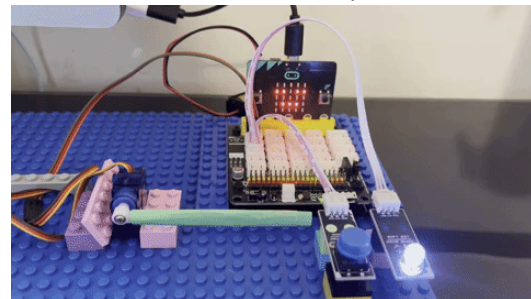
Connection example



Example Advanced Code



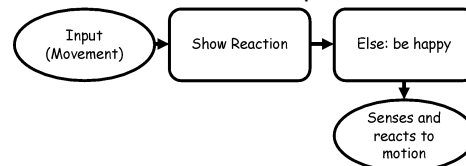
Advanced Example



### Reaction to Sensors - Accelerometer

- Have students think about what they do when they hear a loud noise or how they feel on a drop when they ride a rollercoaster
- Micro:bits can react to the world around them like we can react to things
  - What are some types of sensors that we can use with Micro:bit?
  - What senses do we as humans have?
- Students analyze the process map about using a sensor.
  - How is this like or different than the codes they have worked with today?
  - What is the input? What is the output?

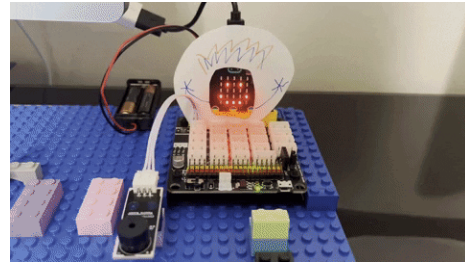
### Process Map



### Example Basic Code for Accelerometer Sensor



- This sensor is built into the Micro:bit and detects when the Micro:bit experiences different types of motion.
- Have students replicate the example of Basic code in their MakeCode editors.
- Have them break down the actions and reactions they see in the Micro:bit.
- Invite students to expand on the code to make a unique reaction to the Micro:bit pet and the accelerometer.
  - So many options on using the accelerometer: can set acceleration threshold, can use shake, or direction of movement, can use direction Micro:bit is facing
  - Have students add in sounds and other features to the motion. They can use more logic to create different reactions

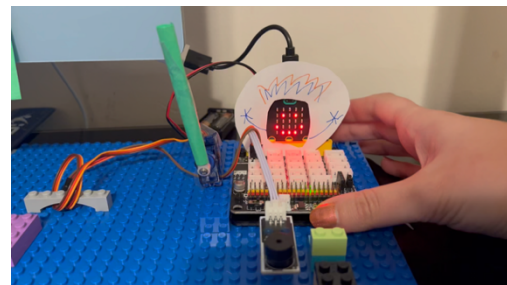


Example of Advanced Sensor Reactions

```

forever
  if acceleration (mg) >= 100 then
    show icon [sad face]
    play tone Low A for 1 beat
    play tone Low F for 1 beat
    servo write pin P1 to 180
    pause (ms) 200
    servo write pin P1 to 0
    pause (ms) 200
    servo write pin P1 to 180
    pause (ms) 200
    servo write pin P1 to 0
    pause (ms) 200
  else
    show icon [happy face]

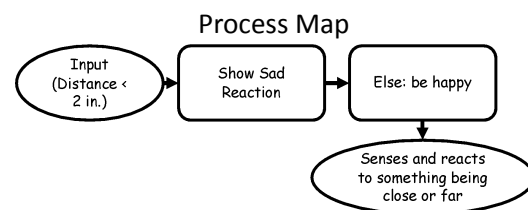
```



### Reaction to Sensors – Ultrasonic Detector

#### ONLY IF YOU HAVE THE STEM KIT BOARD

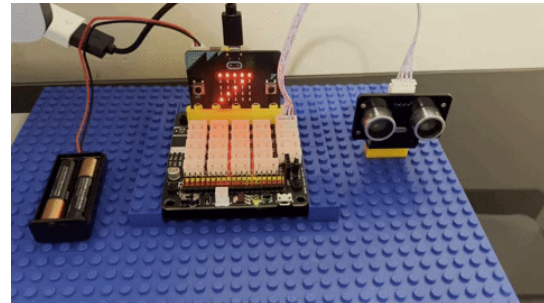
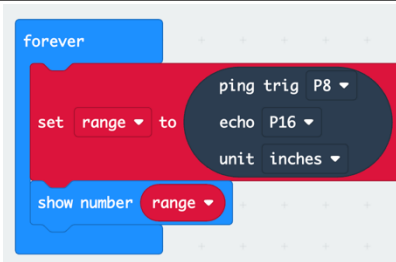
- Have students think about what they do when they hear a loud noise or how they feel on a drop when they ride a rollercoaster
- Micro:bits can react to the world around them like we can react to things
  - What are some types of sensors that we can use with Micro:bit?
  - What senses do we as humans have?
- Students analyze the process map about using a sensor.
  - How is this like or different than the codes they have worked with today?



Basic Ultrasonic Detector Code

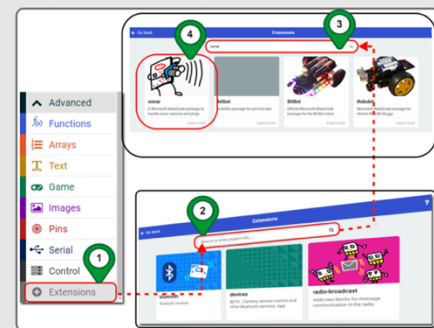


- What is the input? What is the output?
- This sensor is external to the Micro:bit and detects distance from the sensor.
- Have students replicate the example of Basic code in their MakeCode editors.
  - They will need to add the “Sonar” extension from Micro:bit. Click extensions and search sonar. Then choose the first option.
- Have them break down the actions and reactions they see in the Micro:bit.
- Invite students to expand on the code to make a unique reaction to the Micro:bit pet and the ultrasonic detector.
  - So many options on using the ultrasonic detector: can set distance threshold to trigger different emotions, sounds, or movements.
  - Have students add in sounds and other features to the motion. They can use more logic to create different reactions



### How to add Sonar Extension

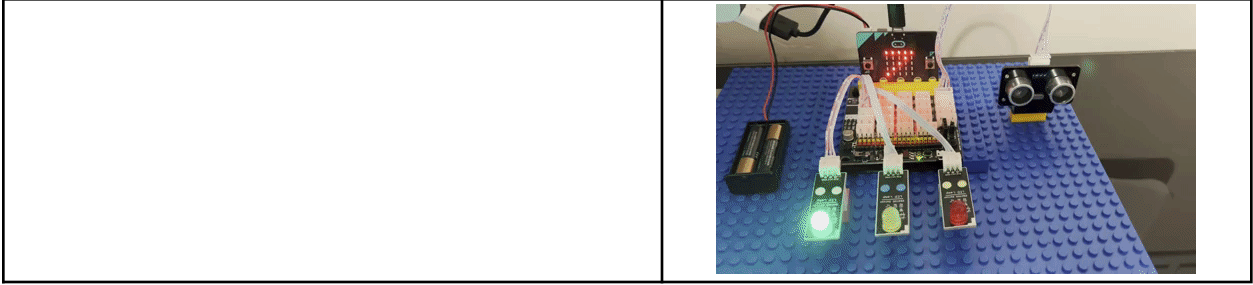
Step 4. In the Extensions page, enter project “Sonar” in the Search textbox and click on the Search button. The page will show you all the matching extensions based on your search string. Now, click on sonar to add it into your project as following figure.



### Example of extended ultrasonic code







## Lunch and Recess

**45 minutes**

### Activity 4: Design Challenge – Introduction, Mentor Presentations, Design Teams, Roles

**Estimated Time:** Introduction to Design Challenge – 10 minutes, Mentor Presentations – 30 minutes, Create design teams - 10 minutes, Assign roles and norms – 10 minutes, Start Kanban Boards – 25 minutes. **Total Time = 95 minutes**

#### Activity Goals:

- Student will learn about and experience a design challenge and how the engineering design process is an essential component in completing design challenges.
- Students will start the Empathize, Define and Ideate stages of the engineering design process while learning how to use interview to gather information from users (as part of empathize), teamwork, project management (Kanban board).
- Student will learn about and experience a design challenge and how the engineering design process is an essential component in completing design challenges.
- Students will go through the cycle of engineering design process with the guidance of a mentor.
- Students will learn from mentors experiences and stories.
- Students will experience a rapid prototyping and testing challenge

#### Activity Procedure:

##### 1. Design Challenge Introduction (10 minutes)

You will be in teams of 2-3 and your task is to design a product for specific users. This is a multiday challenge where your team will go through the Engineering Design process stages. There will be daily whole-group check-in presentations and a final presentation at the end of the week.

- a. Introduce the Design Theme(s) for the camp
- b. Industry Mentors will provide design challenges for students

##### 2. Challenge Presentations

(30 minutes)

Each mentor or teacher will talk about their experiences with computer science projects, engineering design and project management. The goal is to inspire and motivate the students. This is when the technical design challenges will be described to the students.

##### 3. Student Team Creation (10 minutes)

First students need to form design teams of four

- a. Teams could be formed randomly by asking students to count and group them according to their number.
- b. Student groups choose which design challenge they want to work on

#### 4. Team roles and norms (10 minutes)

Once the teams are formed, students will introduce themselves to their team. They will then identify essential roles for successful teamwork and come up with an agreed list of norms. **Note students will have more than 1 role since there are groups of 2–3.** Suggested roles are:

- a. Lead Designer
  - i. Interview and interface with the user, inform the others of users input, lead design ideation, provide prototype feedback
- b. Lead Engineer
  - i. Work with the designer to make something that will actually work, identify and process map the actions and technology needed for the design
- c. Lead Programmer
  - i. Create the code to program the actions of the technology, process map the code, work with Engineer to make something that actually works
- d. Lead Communicator
  - i. Run Kanban Board, check on tasks, make sure the team is documenting and communicating, lead progress presentations and design pitch

Design Team Member Names:

#### Team Roles and Norms

As a team, discuss what roles are important for the success of your design task and assign the roles to the members.  
List the norms that you want your team to follow during the whole design process.

Role	Team Member
<b>Lead Designer</b> Interview the user, inform the engineers and programmers to users, interface back with the user, lead design ideation, provide feedback for prototypes	
<b>Lead Engineer</b> Work with the designer to make something that will actually work, identify the actions and technology needed for the design, process map the actions	
<b>Lead Programmer</b> Create the code to use the technology and program any actions, process map the code	
<b>Lead Communicator</b> Interview the user, Run Kanban Board, check on tasks, make sure everyone is documenting and communicating, lead on the design pitch and progress presentations	
<b>Whole Group</b> Construct prototypes	

#### Team Norms:

1. Write Norms for working together
2. Remember our group Norms

#### Whole Group: Construct Prototypes

#### 5. Kanban boards (10 minutes)

Now that the students have presented their four design ideas and received feedback from the mentor and other team, they can start planning for the next steps.

- a. The teams will identify tasks for each team member, write these tasks on sticky notes and post them under the To-do column on their team's Kanban board, which will be provided by the facilitators.
- b. The facilitators will also present the whole group's Kanban board and remind

#### Design Group Kanban Board:

- Team members decide which tasks are assigned to who and create sticky notes for each task and place them under To Do column.
- At the end of each day, each member moves the sticky notes to the corresponding column.

Name	To Do	In Progress	Testing & Refining	Complete
Johnny	<div>Task 1</div> <div>Task 2</div> <div>Task 3</div>			
Sally	<div>Task 4</div> <div>Task 5</div> <div>Task 6</div>			

<p>each team to update both boards at the end of each day.</p> <p>C.</p>	<p><b>Whole Group Kanban board.</b> At the end of each day, students will move sticky notes containing specific tasks to the corresponding stage of the design process.</p> <p>Sample sticky notes to provide to students:</p> <ul style="list-style-type: none"> <li><b>Interview:</b> Interview: Meet with Mr. Smith, Interview: Meet with Mrs. Jones, Interview: Meet with Mr. Brown, Interview: Meet with Mrs. White, Interview: Meet with Mr. Green, Interview: Meet with Mrs. Black.</li> <li><b>Define:</b> Define: Create a user story, Define: Create a user story, Define: Create a user story, Define: Create a user story, Define: Create a user story, Define: Create a user story.</li> <li><b>Ideate:</b> Ideate: Brainstorm ideas, Ideate: Brainstorm ideas, Ideate: Brainstorm ideas, Ideate: Brainstorm ideas, Ideate: Brainstorm ideas, Ideate: Brainstorm ideas.</li> <li><b>Prototype:</b> Prototype: Build a prototype, Prototype: Build a prototype, Prototype: Build a prototype, Prototype: Build a prototype, Prototype: Build a prototype, Prototype: Build a prototype.</li> <li><b>Feedback:</b> Feedback: Get feedback from Mr. Smith, Feedback: Get feedback from Mrs. Jones, Feedback: Get feedback from Mr. Brown, Feedback: Get feedback from Mrs. White, Feedback: Get feedback from Mr. Green, Feedback: Get feedback from Mrs. Black.</li> <li><b>Deploy:</b> Deploy: Launch the product, Deploy: Launch the product, Deploy: Launch the product, Deploy: Launch the product, Deploy: Launch the product, Deploy: Launch the product.</li> </ul>
--	---

## Day 5: Design Challenge – Interview, Define, Ideate, Feedback, Prototype 1, Feedback

### Overview

Activity	Time, minutes
Welcome and birthday icebreaker	15
Design Challenge (Interview, Define, Ideate, Feedback, Prototype 1, Feedback)	160
Total (including +45 min for lunch)	220
Extra Time	20

### Materials Needed for Day 5

Technology	Craft Supplies	Activity Supplies
<ul style="list-style-type: none"> <li>Computers</li> <li>Micro:bits - 1 per student</li> <li>Servo motors</li> <li>Dupont Wires (male –female, male – male)</li> <li>Alligator Clips</li> <li>AAA Batteries</li> <li>Micro:bit Stem Kits – sensors, LED lights, motors</li> </ul>	<ul style="list-style-type: none"> <li>Paper</li> <li>Tape</li> <li>Markers</li> <li>Pencils</li> <li>Scissors</li> <li>Construction paper</li> <li>Paper towel/toilet paper rolls</li> <li>Cardboard/Cardstock</li> <li>Other craft materials for design projects</li> </ul>	<ul style="list-style-type: none"> <li>Sticky Notes</li> <li>Sticky Easel Pad (optional)</li> <li>Rulers</li> <li>Student Rewards</li> </ul>

### Activity 1: Welcome and Icebreaker

**Estimated Time:** 15 minutes

Ice Breaker: Have the students play two truths and a lie

## Activity 2: Design Challenge - Interview, Define, Ideate, Feedback, Prototype 1, Feedback

**Estimated Time:** Interview (Empathize) – 30 minutes, Define design criteria and needs – 15 minutes, Ideate – 30 minutes, Ideate feedback – 15 minutes, Prototype 1 – 60 minutes, Prototype 1 Feedback – 30 minutes, Refine – 10 minutes, Prepare Prototype 1 Progress Presentation – 25 minutes, Wrap-up and Kanban Boards – 10 minutes. **Total Time = 160 minutes.**

### 6. Interview mentor (Empathize) (30 minutes)

Students will then proceed to interview a mentor to get to know more information about the users of the design and about the design needs.

- Who are the intended users of your design?
- What are the needs of the users?
- What is the purpose of the design?
- How will your design help the intended users (if applicable)?
- What are the attributes of your design?
  - Looks
  - Size
  - Emotions (if any)
  - Movements (if any)
  - Sounds (if any)

How can the users interact with their design? (if possible)

**\*\* Note if mentor is not available, students will use the information from the design challenge slides to get this information. Internet research can also be used at this stage.**

Design Team Member Names: \_\_\_\_\_

**Empathize: Get to know your product users**  
Interview your mentor to get information about your design plan.

Get to know the intended users	Choose a scenario you are working for a company focusing on designing the future.
• Who are the intended users of your design?  • What are the needs of the users?  Purpose of the design • What is the purpose of your design?  • How will your design help the intended users (if applicable)?	<b>Design Attributes</b> What are the attributes of your design? • Looks:  • Size:  • Emotions (if any):  • Movements (if any):  • Sounds (if any):  • How can the users interact with the design (if possible)?

### 7. Define design criteria and needs (15 minutes)

After the interview, the design team will define the design criteria: the wants and needs of the users.

- User Problem Statement: \*User\* needs a way to.....
- Which Micro:bit inputs and features do you need to use? (Buttons, LEDs, Accelerometer, etc.)
- What are the required tools and materials needed for the project?

Design Team Member Names: \_\_\_\_\_

**Define: Set design criteria for the product**

User Problem Statement: \*User\* needs a way to....

Which Micro:bit inputs and features do you need to use? (Buttons, LEDs, Accelerometer, etc.)

What are the required tools and materials needed for the project? (Add-ons to Micro:bit: Servo motors, external buttons, sensors, lights, paper, tape, etc.)

Are there any constraints to the design based on resources?

Are there any other details you need to ask the mentors about the design materials and constraints?

<div><div><div>i. (Add-ons to Micro:bit: buttons, lights, paper, tape, drawing tools, etc.)</div><div>i. Are there any constraints to the design based on resources?</div><div>Are there any other details you need to ask the mentor about the design?</div></div></div>	
<div><div>8. <b>Ideate</b> (30 minutes)</div><div>Student teams then move into the Ideation/Brainstorming stage where they will draw 4 versions of the design and label the parts of the each design. They will also map out the actions the design will do (if applicable) – icons, sounds, movements, sensors, etc. They will also brainstorm few names for their design.</div></div>	<div><div>Design Team Member Names:</div><div><b>Ideate: Brainstorming the initial design</b></div><div>Draw 4 versions of what your design could look like based on the interviews. Label the parts of the design.</div><div><div><div></div><div></div><div></div><div></div></div><div>Draw what each of the actions (emotions, movements, etc.) will look like. Note in the drawing which inputs would start each of the actions.</div></div><div><div>Brainstorm a few names for your product:</div></div></div>

## Lunch and Recess

**45 minutes**

<p><b>9. Ideate Feedback (15 minutes)</b></p> <p>Students will approach their mentor for feedback on their design ideas. The mentor will use the four-square framework as a guide:</p> <ol style="list-style-type: none"> <li>What I like about the design?</li> <li>What I do not like about the design?</li> <li>Questions I have about the design.</li> <li>I want the creator to know.</li> </ol> <p>After talking with the mentor, the students will brainstorm on how to refine their design.</p> <p><b>** Note if a mentor is not available, the students will get feedback from the teachers or other adults supporting the camp.</b></p> <p><b>10. Students will another design team to ask for feedback about their four design ideas. They will use the four-square framework as a guide:</b></p> <ol style="list-style-type: none"> <li>What we like about the design?</li> <li>What do we not like about the design?</li> <li>Questions we have about the design.</li> </ol>	<p><b>Test and Refine: Feedback to make refinements</b></p> <p><b>Ideate Mentor Feedback:</b> Get feedback from your mentor or the user. Identify and make any design changes after the feedback.</p> <table border="1" data-bbox="883 1297 1390 1503"> <tr> <td>What I like about the design &gt;</td><td>Questions I have about the design &gt;</td></tr> <tr> <td>What I do not like about the design &gt;</td><td>I want the creator to know... &gt;</td></tr> </table> <p><b>For Design Group - Identify Refinements</b></p> <p>What I agree with about the feedback: &gt;</p> <p>What I disagree with about the feedback: &gt;</p> <p>What is going to change about the design in the next version: &gt;</p> <p>D3A2.5</p>	What I like about the design >	Questions I have about the design >	What I do not like about the design >	I want the creator to know... >
What I like about the design >	Questions I have about the design >				
What I do not like about the design >	I want the creator to know... >				

<p>d. We want the creator to know.</p>	<p style="text-align: center;"><b>Test and Refine: Feedback to make refinements</b></p> <p><b>Ideate Peer Feedback:</b> Get feedback from another design group. Identify and make any design changes after the feedback.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; padding: 5px;">         &gt; What I like about the design       </td><td style="width: 50%; padding: 5px;">         &gt; Questions I have about the design       </td></tr> <tr> <td style="width: 50%; padding: 5px;">         &gt; What I do not like about the design       </td><td style="width: 50%; padding: 5px;">         &gt; I want the creator to know...       </td></tr> </table> <div style="border: 1px dashed black; padding: 5px; margin-top: 10px;"> <p style="text-align: center;"><b>For Design Group - Identify Refinements</b></p> <p>What I agree with about the feedback:</p> <p>&gt;</p> <p>What I disagree with about the feedback:</p> <p>&gt;</p> <p>What is going to change about the design in the next version:</p> <p>&gt;</p> </div> <p style="text-align: right; font-size: small;">D3A2.5</p>	> What I like about the design	> Questions I have about the design	> What I do not like about the design	> I want the creator to know...		
> What I like about the design	> Questions I have about the design						
> What I do not like about the design	> I want the creator to know...						
<p><b>11. Build Prototype 1 (40 minutes)</b></p> <p>The students will continue to brainstorm following the feedback they received from their mentor and another design team during the previous day. The students will use the feedback received to start building the first prototype for their design.</p> <ol style="list-style-type: none"> <li>a. The students will start by making changes to the process map of the actions for their design.</li> <li>b. The students will identify the changes needed for their codes and sequence of codes.</li> <li>c. Once they have identified the changes to the codes and sequence, they will use MakeCode editor to program their Micro:bit.</li> </ol> <p><b>12. Build prototype 1 CONTINUED (20 minutes)</b></p> <p>The students will continue to brainstorm following the feedback they received from their mentor and another design team during the previous day. The students will use the feedback received to start building the first prototype for their design.</p> <ol style="list-style-type: none"> <li>d. The students will start by making changes to the process map of the actions for their design.</li> <li>e. The students will identify the changes needed for their codes and sequence of codes.</li> </ol>	<p style="text-align: center;"><b>Prototype: Build and test different designs</b></p> <p style="text-align: center;"><b>PROTOTYPE 1</b></p> <p><b>Design Team Member Names:</b></p> <p><b>Instructions:</b></p> <ol style="list-style-type: none"> <li>1. Grab a blank sheet of paper and label it: <b>PROTOTYPE 1</b> with your name at the top.</li> <li>2. Create 2 process maps for the (1) <b>ACTIONS</b> of the Micro:bit and the (2) <b>CODE</b> you are going to use to create the design. Label the process maps.</li> <li>3. Begin building a prototype of your design based on the ideas you came up with in the ideate stage.</li> <li>4. Add any additional resources you may have used to the <b>DEFINE</b> page of the design plan.</li> <li>5. Answer the questions on the <b>REFINEMENT</b> page to track the changes in your design.</li> </ol> <p><b>Example Paper</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; padding: 5px;">Names</td><td style="width: 50%; padding: 5px;">Prototype 1</td></tr> <tr> <td colspan="2" style="height: 100px; padding: 5px; vertical-align: top;"> <b>Actions Process Map</b> </td></tr> <tr> <td colspan="2" style="height: 100px; padding: 5px; vertical-align: top;"> <b>Code Process Map</b> </td></tr> </table> <p style="text-align: right; font-size: small;">D3A2.5</p>	Names	Prototype 1	<b>Actions Process Map</b>		<b>Code Process Map</b>	
Names	Prototype 1						
<b>Actions Process Map</b>							
<b>Code Process Map</b>							

<p>f. Once they have identified the changes to the codes and sequence, they will use MakeCode editor to program their Micro:bit.</p>									
<p><b>13. Prototype 1 Feedback (30 minutes)</b></p> <p>Students will approach their mentor and another design team for feedback on their first prototype. The mentor will use the four square framework as a guide:</p> <ol style="list-style-type: none"> <li>What I like about the design?</li> <li>What I do not like about the design?</li> <li>Questions I have about the design.</li> <li>I want the creator to know.</li> </ol> <p><b>** Note if a mentor is not available, the students will get feedback from the teachers or other adults supporting the camp.</b></p> <p><b>14. Students will approach another design team to ask for feedback about their first prototype using the four square framework as a guide:</b></p> <ol style="list-style-type: none"> <li>What we like about the design?</li> <li>What we do not like about the design?</li> <li>Questions we have about the design.</li> <li>We want the creator to know.</li> </ol>	<p><b>Test and Refine: Feedback to make refinements</b></p> <p><b>Prototype 1 Mentor Feedback:</b> Get feedback from your mentor or the user. Identify and make any design changes after the feedback.</p> <table border="1"> <tr> <td>What I like about the design &gt;</td><td>Questions I have about the design &gt;</td></tr> <tr> <td>What I do not like about the design &gt;</td><td>I want the creator to know... &gt;</td></tr> </table> <p><b>For Design Group - Identify Refinements</b></p> <p>What I agree with about the feedback: &gt;</p> <p>What I disagree with about the feedback: &gt;</p> <p>What is going to change about the design in the next version: &gt;</p> <p>D3A2.5</p> <p><b>Test and Refine: Feedback to make refinements</b></p> <p><b>Prototype 1 Peer Feedback:</b> Get feedback from another design group. Identify and make any design changes after the feedback.</p> <table border="1"> <tr> <td>What I like about the design &gt;</td><td>Questions I have about the design &gt;</td></tr> <tr> <td>What I do not like about the design &gt;</td><td>I want the creator to know... &gt;</td></tr> </table> <p><b>For Design Group - Identify Refinements</b></p> <p>What I agree with about the feedback: &gt;</p> <p>What I disagree with about the feedback: &gt;</p> <p>What is going to change about the design in the next version: &gt;</p> <p>D3A2.5</p>	What I like about the design >	Questions I have about the design >	What I do not like about the design >	I want the creator to know... >	What I like about the design >	Questions I have about the design >	What I do not like about the design >	I want the creator to know... >
What I like about the design >	Questions I have about the design >								
What I do not like about the design >	I want the creator to know... >								
What I like about the design >	Questions I have about the design >								
What I do not like about the design >	I want the creator to know... >								

## Day 6: Design Challenge – Refine, Prepare to Present, Progress Check and Gallery Walk, Prototype 2

### Overview

Activity	Time, minutes
Welcome and Ice Breaker	10
Design Challenge ( <i>Refine, Prepare to Present, Progress Check-in and Gallery Walk, Prototype 2</i> )	165
Total (including +45 min for lunch)	220
Extra Time	20



## Materials Needed for Day 6

Technology	Craft Supplies	Activity Supplies
<ul style="list-style-type: none"> <li>• Computers</li> <li>• Micro:bits - 1 per student</li> <li>• Servo motors</li> <li>• Dupont Wires (male –female, male – male)</li> <li>• Alligator Clips</li> <li>• AAA Batteries</li> <li>• Micro:bit Stem Kits – sensors, LED lights, motors</li> </ul>	<ul style="list-style-type: none"> <li>• Paper</li> <li>• Tape</li> <li>• Markers</li> <li>• Pencils</li> <li>• Scissors</li> <li>• Construction paper</li> <li>• Paper towel/toilet paper rolls</li> <li>• Cardboard/Cardstock</li> <li>• Other craft materials for design projects</li> </ul>	<ul style="list-style-type: none"> <li>• Sticky Notes</li> <li>• Sticky Easel Pad (optional)</li> <li>• Rulers</li> <li>• Student Rewards</li> </ul>

## Activity 1: Welcome and Icebreaker

**Estimated Time:** Welcome – 5 minutes, Icebreaker – 5 minutes. **Total Time = 10 minutes**

### Welcome

**Icebreaker Activity:** Students choose a superpower but cannot use words to say what it is

- Students are in random groups of 5
- Each one has to share their superpower

### Activity goals:

- Students have to use system thinking to portray an idea without using words

## Activity 2: Design Challenge – Refine, Prepare to Present, Progress Check/Gallery Walk, Prototype 2

**Estimated Time:** Refine – 20, Prepare to Present – 30, Progress Check and Gallery Walk – 60 minutes , Prototype 2 – 55 minutes. **Total Time = 165 minutes**

**15. Refine (20 minutes)**

The student teams will continue to discuss the feedback from their mentor and another design team. They will brainstorm on how to incorporate the feedback and proceed to changing the necessary actions in their action process maps and then changing the codes.

- a. Then the students will approach their mentors again to show their actions process map and codes. The students will not incorporate the changes yet, but only show the changes to the actions process map. This is to prepare them for the following day's activities.

**Refinement: Making and tracking changes to the design**

Stage	What stayed the same from your previous design? How does it support the problem statement?	What changed from your previous design? How does it support the problem statement?
Ideate to Prototype 1		
Prototype 1 to Prototype 2		
Prototype 2 to Final Design		

D3/A2.5

**Lunch and Recess****45 minutes****16. Prepare to present prototype 1 (30 minutes)**

Students will create Google slides to present their first design prototype. They will use the following points as a guide for their presentation: (30 minutes)

- a. Who are the intended users.
- b. The purpose of the design.
- c. Features of your designs. (demonstrate what the prototype can do)
- d. The functions of your design.
- e. What are your next steps?

Design Team Member Names:

**Progress Check In: Present Prototype 1**

Present your prototype to the whole group. Use the following to guide your presentation.

1. Present your User Problem Statement and users
2. Show and demonstrate your prototype
3. When presenting, show and explain the following:
  - The actions process map showing your prototype's intended function.
  - The program (codes and sequence) on your MakeCode editor.
  - How your designs changed from ideate to prototype 1
  - What are your next steps?
4. Move the sticky notes on the Kanban board to show where you are on your Design Challenge and what needs to be done to complete it.

**17. Update Kanban and Wrap-up (10 minutes)**

Students will then update their team's Kanban board regarding what tasks are completed and in progress. They will also update the Kanban board for the whole group to update which design stage they are at.

**Activity Procedure:**

1. **Progress check and gallery walk** (60 minutes) In pairs, members of each design teams will take turns in presenting their design and walking around to check the designs of other teams. Each pair must visit all the design groups and provide feedback.  
During the first 30 minutes, two members of each design team will stay to present, while the other two members will walk around to give feedback on the other designs. The pairs switch after 30 minutes. (60 minutes)
2. When students approach a Design team, the design team will talk about the following:
  - a. Who are the intended users.
  - b. The purpose of the design.
  - c. Features of your designs. (demonstrate what the prototype can do)
  - d. The functions of your design.
  - e. What are your next steps?
  - f. Students will leave sticky notes about the following:
    - i. What I like about your design.
    - ii. A question I have about the design

**\*Each design group has a T-chart with “what I like about the design” and “what I have questions about.” Students will have sticky notes to give feedback on\***

What I like about the design	What I have questions about

3. **Build prototype 2** (55 minutes)
  - a. The students will continue to brainstorm following the feedback they received from their mentor, another design team and from the Gallery walk. The students use the feedback to start building the second prototype for their design.
  - b. The students will start by making changes to the process map of the actions for their design.
  - c. The students will identify the changes needed for their codes and sequence of codes.
  - d. Once they have identified the changes to the codes and sequence, they will use

Design Team Member Names: \_\_\_\_\_ **Prototype: Build and test different designs**  
**PROTOTYPE 2**

Instructions:

1. Grab a blank sheet of paper and label it: **PROTOTYPE 2** with your name at the top
2. Create 2 process maps for the (1) **ACTIONS** of the Microbit and the (2) **CODE** you are going to use to create the design. Label the process maps
3. Begin building a prototype of your design based on the ideas you came up with in the ideate stage.
4. Add any additional resources you may have used to the **DEFINE** page of the design plan.
5. Answer the questions on the **REFINEMENT** page to track the changes in your design

**Example Paper**

Names	Prototype 2
Actions Process Map	
Code Process Map	

D3A2.5

MakeCode editor to make changes to their program.	
---	--

## Day 7: Design Challenge – Feedback, Refine, Final Design, Prepare to Present, Progress Check-in Jigsaw

### Overview

Activity	Time, minutes
Welcome and Ice Breaker	10
Design Challenge (Feedback, Refine, Final Design, Prepare to Present, Progress Check-in Jigsaw, Update Kanban board)	180
Total (including +45 min for lunch)	235
Extra Time	5

### Materials Needed for Day 7

Technology	Craft Supplies	Activity Supplies
<ul style="list-style-type: none"> <li>Computers</li> <li>Micro:bits - 1 per student</li> <li>Servo motors</li> <li>Dupont Wires (male –female, male – male)</li> <li>Alligator Clips</li> <li>AAA Batteries</li> <li>Micro:bit Stem Kits – sensors, LED lights, motors</li> </ul>	<ul style="list-style-type: none"> <li>Paper</li> <li>Tape</li> <li>Markers</li> <li>Pencils</li> <li>Scissors</li> <li>Construction paper</li> <li>Paper towel/toilet paper rolls</li> <li>Cardboard/Cardstock</li> <li>Other craft materials for design projects</li> </ul>	<ul style="list-style-type: none"> <li>Sticky Notes</li> <li>Sticky Easel Pad (optional)</li> <li>Rulers</li> <li>Student Rewards</li> </ul>

### Activity 1: Welcome and Icebreaker

**Estimated Time:** Welcome – 5 minutes, Icebreaker – 5 minutes. **Total Time = 10 minutes**

#### Welcome

**Icebreaker Activity:** Students choose a superpower but cannot use words to say what it is

- Students are in random groups of 5
- Each one has to share their superpower

#### Activity goals:

- Students have to use system thinking to portray an idea without using words

## Activity 2: Design Challenge – Prototype 2 Feedback, Refine, Final Design, Create Plan, Jigsaw

**Estimated Time:** Prototype 2 Feedback – 30 minutes, Refine design plan – 30 minutes, Start Final Design – 60 minutes, Prepare Progress Presentation – 30 minutes, Final Design Plan Jigsaw – 30 minutes. **Total time = 180 minutes**

### Activity Goals:

- Student will learn about and experience a design challenge and how the engineering design process is an essential component in completing design challenges.
- Students will go through the cycle of engineering design process with the guidance of a mentor.
- Student will learn public speaking skills.

#### 4. **Prototype 2 feedback (30 minutes)**

Students will approach their mentor and a peer design team for feedback (30 minutes)  
The mentor will use the four square framework as a guide:

- What I like about the design?
- What I do not like about the design?
- Questions I have about the design.
- I want the creator to know.

**\*\* Note if a mentor is not available, the students will get feedback from the teachers or other adults supporting the camp.**

Students will approach another design team to ask for feedback about their second prototype using the four square framework:

- What we like about the design?
- What we do not like about the design?
- Questions we have about the design.
- We want the creator to know.

**Test and Refine: Feedback to make refinements**  
**Prototype 2 Mentor Feedback:** Get feedback from your mentor or the user. Identify and make any design changes after the feedback.

What I like about the design >	Questions I have about the design >
What I do not like about the design >	I want the creator to know... >

**For Design Group - Identify Refinements**

What I agree with about the feedback:  
>

What I disagree with about the feedback:  
>

What is going to change about the design in the next version:  
>

D3A2 5

**Test and Refine: Feedback to make refinements**  
**Prototype 2 Peer Feedback:** Get feedback from another design group. Identify and make any design changes after the feedback.

What I like about the design >	Questions I have about the design >
What I do not like about the design >	I want the creator to know... >

**For Design Group - Identify Refinements**

What I agree with about the feedback:  
>

What I disagree with about the feedback:  
>

What is going to change about the design in the next version:  
>

D3A2 5

## Lunch and Recess

### 5. Refine (30 minutes)

- The student teams will continue to discuss the feedback from their mentor and another design team. They will brainstorm on how to incorporate the feedback and proceed to changing the necessary actions in their action process maps and then changing the codes.
- Then the students will approach their mentors again to show their actions process map and codes. The students will not incorporate the changes yet, but only show the changes to the actions process map. This is to prepare them for the final design.

**\*\* Note if a mentor is not available, the students will get feedback from the teachers or other adults supporting the camp.**

#### Refinement: Making and tracking changes to the design

Stage	What stayed the same from your previous design? How does it support the problem statement?	What changed from your previous design? How does it support the problem statement?
Ideate to Prototype 1		
Prototype 1 to Prototype 2		
Prototype 2 to Final Design		

D3/42 S

### 6. Start final design (60 minutes)

- The students will continue to brainstorm following the feedback they received from their mentor and another design team. The students will use the feedback received to start building their final design. They do not need to finish building their final design at this stage. (60 mins)
- The students will start by making changes to the process map of the actions for their design.
- The students will identify the changes needed for their codes and sequence of codes.
- Once they have identified the changes to the codes and sequence, they will use MakeCode editor to program their Micro:bit.

Design Team Member Names:

#### Prototype: Build and test different designs Final Design

##### Instructions:

- Grab a blank sheet of paper and label it: Final Design with your name at the top
- Create 2 process maps for the (1) ACTIONS of the Micro:bit and the (2) CODE you are going to use to create the design. Label the process maps.
- Begin building a prototype of your design based on the ideas you came up with in the ideate stage.
- Add any additional resources you may have used to the DEFINE page of the design plan.
- Answer the questions on the REFINEMENT page to track the changes in your design

##### Example Paper

Names	Final Design
Actions Process Map	
Code Process Map	

D3A2 S

## Lunch and Recess

**45 minutes**

<div>7. Create final design plan presentation (30 minutes)</div> <div><div>a. Students will create a Google slide to present their plan for their final design. They will use the following points as a guide for their presentation:</div><div>8. Present one Google Slide that shows the following:</div><div><div>a. Present your User Problem Statement and users</div><div>b. Show and demonstrate your prototype</div><div>c. When presenting, show and explain the following:</div><div><div>i. The actions process map showing your prototype’s intended function.</div><div>ii. The program (codes and sequence) on your MakeCode editor.</div><div>iii. How your designs changed from ideate to prototype 1</div><div>iv. What are your next steps?</div></div></div><div>9. Move the sticky notes on the Kanban board to show where you are on your Design Challenge and what needs to be done to complete it.</div></div>	<div>Design Team Member Names: _____</div> <div>Progress Check In: Present Final Design Plan</div> <div>Present your plan for your final design to the whole group. Use the following to guide your presentation.</div> <div><div>1. Present your User Problem Statement and users</div><div>2. Show and demonstrate your prototype</div><div>3. When presenting, show and explain the following:</div><div><div>• The actions process map showing your prototype's intended function.</div><div>• The program (codes and sequence) on your MakeCode editor.</div><div>• How your designs changed from Prototype 1 to prototype 2 to Final Design</div><div>• What are your next steps?</div></div><div>4. Move the sticky notes on the Kanban board to show where you are on your Design Challenge and what needs to be done to complete it.</div></div>												
<div>10. Progress Check In: Jigsaw design plan (30 minutes)</div> <div><div>a. JIGSAW Presentations</div><div>b. Each member of the design groups will be in a new team. They will present their team’s design to the small group for feedback (25 minutes)</div><div><div>i. What I like</div><div>ii. What I have questions about</div></div><div>c. Design groups will regroup with their feedback to make any small changes to their designs for the final design. (25 minutes)</div></div>	<div>Progress Check In: Present Final Design Plan</div> <div>Jigsaw - Each member of your design group will break off into a new group. You will represent your design group in your new small group. Each design group representative will present their design update and receive feedback.</div> <div><table><tr><td>What I like about the design</td><td>What I do not like about the design</td><td>Questions I have about the design</td></tr><tr><td>What I like about the design</td><td>What I do not like about the design</td><td>Questions I have about the design</td></tr><tr><td>What I like about the design</td><td>What I do not like about the design</td><td>Questions I have about the design</td></tr><tr><td>What I like about the design</td><td>What I do not like about the design</td><td>Questions I have about the design</td></tr></table></div>	What I like about the design	What I do not like about the design	Questions I have about the design	What I like about the design	What I do not like about the design	Questions I have about the design	What I like about the design	What I do not like about the design	Questions I have about the design	What I like about the design	What I do not like about the design	Questions I have about the design
What I like about the design	What I do not like about the design	Questions I have about the design											
What I like about the design	What I do not like about the design	Questions I have about the design											
What I like about the design	What I do not like about the design	Questions I have about the design											
What I like about the design	What I do not like about the design	Questions I have about the design											
<div>11. Update Kanban and wrap-up (10 minutes)</div> <div><div>a. Students will then update the Kanban boards for their team to</div></div>													

<p>update what tasks are completed and in progress. They will also update the Kanban board for the whole group to update which design stage they are at.</p>	
--	--



## Day 8: Design Challenge – Final Day

### Overview

Activity	Time, minutes
Welcome and Ice breaker	20
Design Challenge Continuation (Final Design, Prepare Pitch, and Pitch Presentation)	175
Extra Design Challenge (only use if needed)	0-125
Total (including +45 min for lunch and excluding extra design challenge)	240
Extra Time	0

### Materials Needed for Day 8

Technology	Craft Supplies	Activity Supplies
<ul style="list-style-type: none"><li>• Computers</li><li>• Micro:bits - 1 per student</li><li>• Servo motors</li><li>• Dupont Wires (male –female, male – male, female- female)</li><li>• Alligator Clips</li><li>• AAA Batteries</li><li>• Micro:bit Stem Kits – sensors, LED lights, motors</li></ul>	<ul style="list-style-type: none"><li>• Paper</li><li>• Tape</li><li>• Markers</li><li>• Pencils</li><li>• Scissors</li><li>• Construction paper</li><li>• Paper towel/toilet paper rolls</li><li>• Cardboard/Cardstock</li><li>• Other craft materials for design projects</li></ul>	<ul style="list-style-type: none"><li>• Sticky Notes</li><li>• Sticky Easel Pad (optional)</li><li>• Student Rewards</li></ul>

### Activity 1: Welcome

**Estimated Time:** 20 minutes

**Welcome – 5 minutes**

**Icebreaker Activity – Sorting Networks - 15 minutes**

#### Activity Goals:

- Get students moving and working with different groups
- Get students thinking computationally
- Allow students to experience sorting networks and figure out how they work

#### Activity

- Group students randomly into groups of 6, give each group either random number or letter
- Students follow the sorting algorithm drawn on the floor. Pairs will step into the first box, introduce themselves, and then decided which number or letter comes before or after the other.

If it is BEFORE the students follows the arrow on the left to the next box. If it is AFTER the student follows the number of the right to the next box. Students repeat introductions, comparisons, and movements until they reach the end.

- Have students observe what occurred while they made their way through the network. What did the comparisons and movements facilitate? \*They should be sorted from lowest to highest going left to right.
- <https://www.csunplugged.org/en/topics/sorting-networks/unit-plan/reinforcing-numeracy-through-a-sorting-network/>

## Activity 2: Design Challenge – Final Product, Prepare Pitch, Pitch Presentations

**Estimated Time:** Complete final design - 55 minutes, Create Pitch Presentation – 60 minutes, Product Pitch Presentations and Feedback – 60 minutes. **Total time = 180 minutes**

### Activity Goals:

- Student will learn about and experience a design challenge and how the engineering design process is an essential component in completing design challenges.
- Students will go through the cycle of engineering design process with the guidance of a mentor.
- Student will learn public speaking skills.

### Activity Procedure:

1. <b>Final Design Preparation</b> (55 minutes) Design Teams will complete their final design. They will have to make sure that all the tasks in the Kanban boards are completed.	
--	--

## Lunch and Recess

**45 minutes**

2. <b>Create Pitch Presentation</b> (60 minutes) Students will create a 5-minute pitch presentation to showcase their final design. They will use the following points as a guide for their presentation. Each student will be assigned one of the talking points below: <ul style="list-style-type: none"> <li>a. Introduce the problem statement, Introduce the Product: Name, the users, the purpose (Communicator)</li> <li>b. How does your product meet the user's needs? Demonstrate what your final product can do. (Designer)</li> </ul>	<p>Design Team Member Names: _____ Design Name: _____</p> <p>Present Final Design</p> <p>Present your final design to the whole group. Use the following to guide your presentation.</p> <ul style="list-style-type: none"> <li>• Create Pitch Presentation with your design team</li> <li>• Create a 5-minute pitch presentation to showcase their final design. They will use the following points as a guide for their presentation. Each student will be assigned one of the talking points below: <ul style="list-style-type: none"> <li>• Introduce the problem statement, Introduce the Product: Name, the users, the purpose (Communicator)</li> <li>• How does your product meet the user's needs? Demonstrate what your final product can do. (Designer)</li> <li>• Explain how the designs changed from the ideate, prototype stages 1 and 2, and the final design? (Engineer)</li> <li>• What was the biggest challenge or barrier your group had to overcome while moving from ideas on paper to a final product? (Programmer)</li> <li>• What could be improved about the product? (Communicator)</li> </ul> </li> <li>• We will then hold a "pitch competition" where each design team will present their design products and receive votes.</li> </ul> <p style="text-align: right;">D3A2.5</p>
--	---

<p>c. Explain how the designs changed from the ideate, prototype stages 1 and 2, and the final design? (Engineer)</p> <p>d. What was the biggest challenge or barrier your group had to overcome while moving from ideas on paper to a final product? (Programmer)</p> <p>e. What could be improved about the product? (Communicator)</p>																			
<p>3. <b>Project Pitch and Feedback</b> (60 Minutes) – 5 mins, 3 mins, 2 mins set up (10 mins per group X 6 groups)</p> <p>Students will create a 5-minute pitch presentation to showcase their final design. They will use the following points as a guide for their presentation. Each student will be assigned one of the talking points below:</p> <p>a. Introduce the problem statement, Introduce the Product: Name, the users, the purpose (Communicator)</p> <p>b. How does your product meet the user's needs? Demonstrate what your final product can do. (Designer)</p> <p>c. Explain how the designs changed from the ideate, prototype stages 1 and 2, and the final design? (Engineer)</p> <p>d. What was the biggest challenge or barrier your group had to overcome while moving from ideas on paper to a final product? (Programmer)</p> <p>e. What could be improved about the product? (Communicator)</p>	<p style="text-align: center;">Final Design Presentation Rubric Chart</p> <table border="1"> <thead> <tr> <th colspan="2">Product Name:</th> </tr> <tr> <th>Category</th> <th>Rating 1-5 (1 lowest, 5 highest)</th> </tr> </thead> <tbody> <tr> <td>The final product has a clear purpose.</td> <td></td> </tr> <tr> <td>The final product meets the users needs.</td> <td></td> </tr> <tr> <td>There was a clear progression in product design from each of the design stages. (Ideate, Prototype 1, Prototype 2, Final Design)</td> <td></td> </tr> <tr> <td>The design team was able to identify and explain how they overcame a barrier in the design process.</td> <td></td> </tr> <tr> <td>The design team had a firm understanding of where future improvements could be made.</td> <td></td> </tr> <tr> <td>The pitch presentation was organized and easy to understand.</td> <td></td> </tr> <tr> <td><b>Total Points</b></td> <td>_____/30 points</td> </tr> </tbody> </table>	Product Name:		Category	Rating 1-5 (1 lowest, 5 highest)	The final product has a clear purpose.		The final product meets the users needs.		There was a clear progression in product design from each of the design stages. (Ideate, Prototype 1, Prototype 2, Final Design)		The design team was able to identify and explain how they overcame a barrier in the design process.		The design team had a firm understanding of where future improvements could be made.		The pitch presentation was organized and easy to understand.		<b>Total Points</b>	_____/30 points
Product Name:																			
Category	Rating 1-5 (1 lowest, 5 highest)																		
The final product has a clear purpose.																			
The final product meets the users needs.																			
There was a clear progression in product design from each of the design stages. (Ideate, Prototype 1, Prototype 2, Final Design)																			
The design team was able to identify and explain how they overcame a barrier in the design process.																			
The design team had a firm understanding of where future improvements could be made.																			
The pitch presentation was organized and easy to understand.																			
<b>Total Points</b>	_____/30 points																		

### Activity 3: Extra Design Challenge

**Estimated Time:** 0 - 125 minutes (Can be adjusted due to remaining time).

#### Activity Goals:

- Students apply what they have learned this week to create a product of their own design using the microbit and extensions of choice.

**Activity Procedure:**

<b>1. Project Explanation and Creation</b> (75 minutes) <ul style="list-style-type: none"><li>a. Present Challenge to students. Students are to create their own product of choice using the microbit and extensions. Some possible choices are games and toys. Students may work independently or in groups of two.</li><li>b. Encourage students to create a process map of their product first.</li><li>c. Give students remaining time to create their product.</li><li>d. Encourage students to give and receive feedback to peers during this time.</li></ul>	
<b>2. Share Products</b> (20 minutes) <ul style="list-style-type: none"><li>a. Students will informally share their final products with the class.</li><li>b. Students can also vote on the best product.</li></ul>	
<b>3. Wrap Up</b> 30 minutes <ul style="list-style-type: none"><li>a. Clean up all Materials</li><li>b. Complete Final Surveys</li></ul>	

## Extra Camp Activities

This section is composed of extra activities that you can do on certain activity days or on any day during the camp if there is extra time during the day.

### Any Time

Activity: Career Talks

**Estimated Time:** 30 minutes

<p>1. Students will create a process map for their future including career goals, higher education goals, and the path they wish to take to reach their goals.</p>	<p>Name: _____</p> <p><b>Process Mapping</b></p> <p>Process: A sequence of activities intended to produce particular results</p> <p><b>Steps to Building a Process Map</b></p> <ol style="list-style-type: none"> <li>1. Identify the process</li> <li>2. Determine the boundaries: inputs (start) and outputs (end)</li> <li>3. Start with the big picture steps</li> <li>4. Add in the details: Actions, Decisions and Questions, Relationships to map flow</li> <li>5. Complete the ENTIRE map before refining</li> <li>6. Assess the map for improvement areas: Ask questions and make notes to locate possible problem areas</li> <li>7. Get feedback from peers or users</li> </ol> <p><b>Process Map Key</b></p> <ul style="list-style-type: none"> <li>Inputs: The start of your process</li> <li>Outputs: The end result of your process</li> <li>Actions/Steps: The steps in the process: how to get from the start to the end</li> <li>Decisions: Questions or choices that can change the flow of the process</li> <li>Flow: Show the direction of the steps in the process</li> </ul> <p>Make a process map for your career</p> <p>D1A2.1.5</p>
--	--

## After Programming Basics

The minimum requirement is to complete the programming basics to be able to complete the following activities. These activities can be added in extra time throughout the camp.

### Activity: Hot Potato Game

**Estimated Time:** 60 minutes

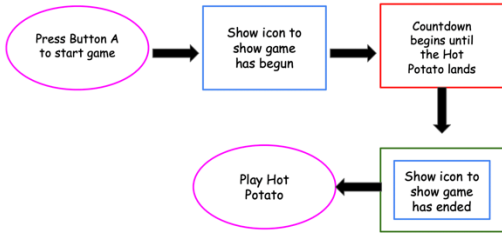
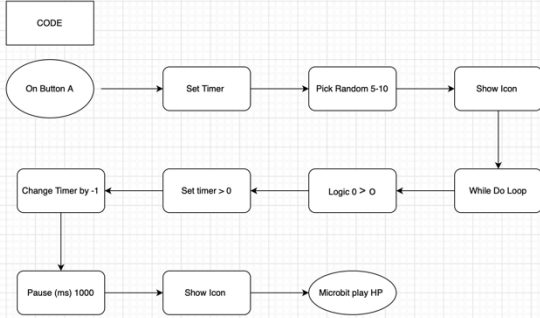
#### Activity Goals:

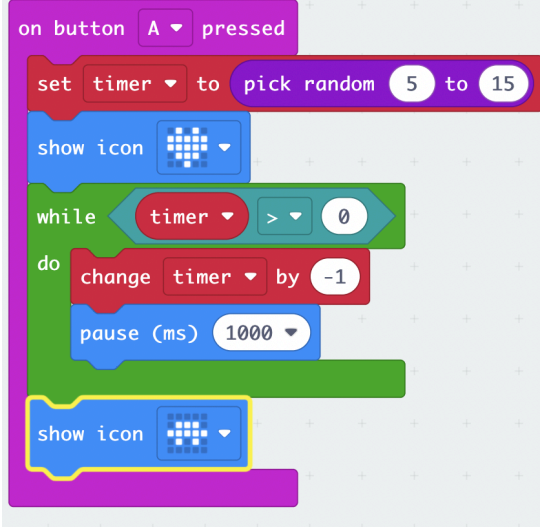
- Describe how the time taken grows with the size of input
- Explore the probability of finding a particular value in a random set.
- Identify how loops determine code

#### Activity Procedure:

- Have students play Hot Potato and familiarize themselves with the game
- Have students go through process mapping the Hot Potato Game
- Once students have created a map, allow them to code the game until they are satisfied and have hit all the requirements
- Have students play the game to recognize the different countdown possibilities

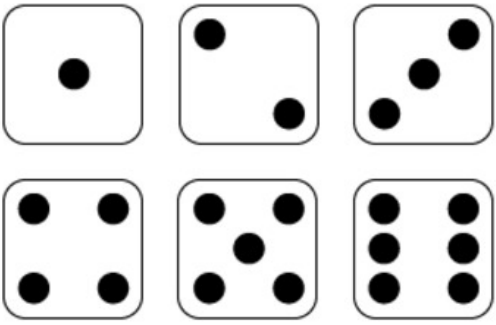
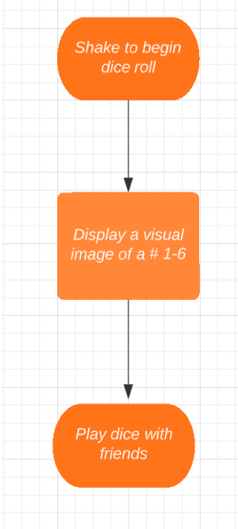
<p>1. Have students play hot potato in groups of 4</p>	
<p>2. Introduce the activity: Your challenge is to design a Hot Potato Game using the Micro:bit. Provide the Program Requirements Below:</p>	<p><b>Program Requirements</b></p> <ol style="list-style-type: none"> <li>1. Microbit needs a starting button to start the game</li> <li>2. Microbit needs a countdown to create the passing motion between students</li> <li>3. Microbit needs to display an icon for the starting and ending of the game</li> <li>4. Microbit needs a loop and a timer</li> </ol>

<ol style="list-style-type: none"> <li>a. Micro:bot needs a starting button to know when to start the game (the “potato becomes hot”)</li> <li>b. Micro:bit needs to countdown to represent the number of seconds left before someone is caught holding the potato</li> <li>c. Micro:bit needs to display an icon to show the game has started and ended</li> <li>d. Micro:bit needs a loop and a timer variable</li> </ol> <p><b>Facilitator Shares Slide and/or directs students to slide in online activity handout workbook</b></p>	
<ol style="list-style-type: none"> <li>2. Have the students identify the inputs and outputs for the game               <ol style="list-style-type: none"> <li>a. What will start the game? What is the input?</li> <li>b. What is the end result of the game or the output?</li> </ol> </li> <li>3. Have students think about the actions that have to happen to get from the start to the end result               <ol style="list-style-type: none"> <li>a. Which actions needs to happen for the game to run?</li> <li>b. Actions Process Map is developed</li> </ol> <p style="text-align: center;">*Example Process Map Shown*</p> </li> </ol>	<p><b>Process Mapping</b></p> <ol style="list-style-type: none"> <li>1. Review the process map for the <b>ACTIONS</b> needs for the Micro: bit to play Hot Potato               <ul style="list-style-type: none"> <li>o Recall the process map symbols</li> <li>o What is the input? What is the output?</li> <li>o Which actions do you think need to happen to get the end result?</li> </ul> </li> <li>2. Create a process map for the <b>CODE</b> you will need to make the actions happen               <ul style="list-style-type: none"> <li>• What kind of code blocks will you need?</li> <li>• What kind of input do you want to use to start the game</li> <li>• What is the sequence the code needs to run in?</li> <li>• How could you use <b>Logic</b> to make different actions happen based on different inputs?</li> </ul> </li> </ol> <p style="text-align: center;"><b>Example Process Map</b></p> <p style="text-align: center;"><b>Actions Processing Map</b></p>  <pre> graph LR     A([Press Button A to start game]) --&gt; B[Show icon to show game has begun]     B --&gt; C[Countdown begins until the Hot Potato lands]     C --&gt; D[Show icon to show game has ended]     D --&gt; E([Play Hot Potato])   </pre>
<ol style="list-style-type: none"> <li>4. Guide students to take the Actions Process map they built and identify the code blocks they believe they will need. Students will create a CODE process map to match the actions they listed.               <ol style="list-style-type: none"> <li>a. What kinds of code blocks will you need?</li> <li>b. What kind of input do you want use to start the game?</li> <li>c. What is the sequence the code needs to run in?</li> </ol> </li> </ol>	<p style="text-align: center;"><b>Example Code Process Map</b></p>  <pre> graph LR     A([On Button A]) --&gt; B[Set Timer]     B --&gt; C[Pick Random 5-10]     C --&gt; D[Show Icon]     D --&gt; E[While Do Loop]     E --&gt; F[Logic 0 &gt; 0]     F --&gt; G[Set timer &gt; 0]     G --&gt; H[Change Timer by -1]     H --&gt; I[Pause ms 1000]     I --&gt; J[Show Icon]     J --&gt; K([Microbit play HP])   </pre>
<ol style="list-style-type: none"> <li>5. Students move to MakeCode editor to build out their code the process mapped</li> </ol>	<p style="text-align: center;"><b>Continuous Code</b></p>

<p>a. Use the MakeCode editor to program the Micro:bit to play rock, paper, scissors</p> <p>b. Test your code periodically using the simulator or the Micro:bit to identify any bugs or test your sequence.</p> <p> *Example of Code Shown*</p>	
<p>6. Check each group's programming when they are finished to make sure it follows their process maps and results in a complete game of Hot Potato</p> <p>a. Students then load the program onto their Micro:bits</p>	
<p>7. Each students goes 2 rounds of Hot Potato against their Micro:bit</p>	

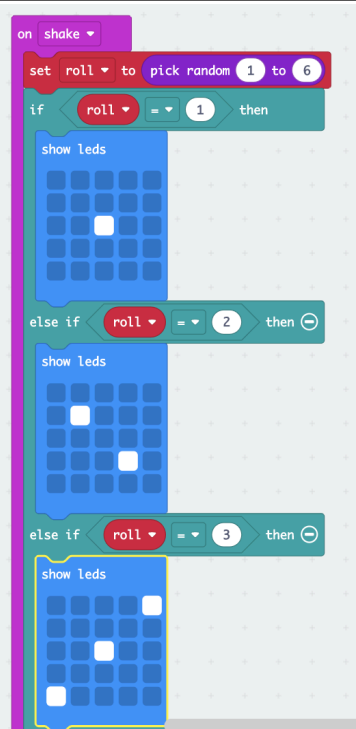
## Activity: Dice Roll

**Estimated Time:** 20 minutes

<ol style="list-style-type: none"><li>1. Students will code their Micro:bit to mimic that of a standard die with 6 numbers<ol style="list-style-type: none"><li>a. Students will use code to pick a random number and then assign led's to show the number</li></ol></li></ol>	 <p>Diagram showing six dice faces arranged in a 2x3 grid. The top row shows faces with 1, 2, and 3 dots. The bottom row shows faces with 4, 5, and 6 dots.</p>
<ol style="list-style-type: none"><li>2. Students will follow the process map given to code the Micro:bit.<ol style="list-style-type: none"><li>b. Students will utilize the actions process map to create a code process map</li></ol></li></ol>	 <pre>graph TD; A([Shake to begin dice roll]) --&gt; B[Display a visual image of a # 1-6]; B --&gt; C([Play dice with friends]);</pre> <p>Diagram showing a process map with three steps: 1. Shake to begin dice roll, 2. Display a visual image of a # 1-6, 3. Play dice with friends.</p>



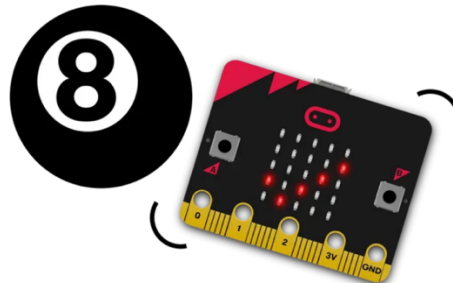
3. When coded correctly, the students will shake the Micro:bit and LEDs will appear that look like a face of a die
  - c. Students will utilize the “on shake” function to initiate the start of the roll.
  - d. Students will utilize variable tab, math tab, and logic tab in their code.
  - e. Have students add sounds for each number or create new symbols to represent numbers
4. Have students play against each other and create games. Example games below.
  - a. Highest roll wins
  - b. First to get equal numbers
  - c. Try to see how long it takes for all six numbers to appear on die (Micro:bit)

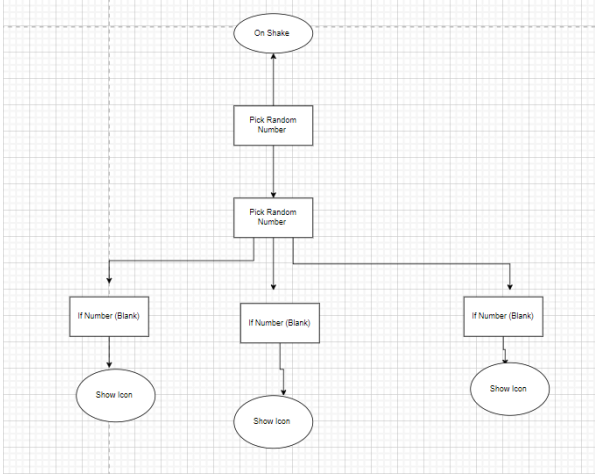
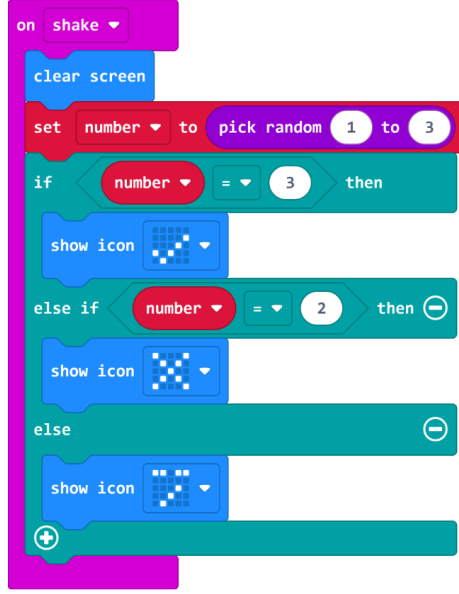


#### Activity: Magic 8 Ball

**Estimated Time:** 20 minutes


1. Students will code their Micro:bit to mimic that of a magic 8 ball

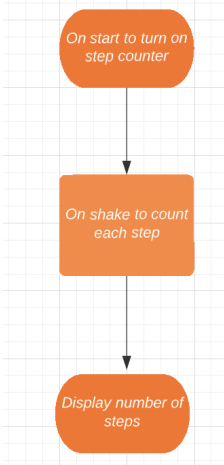
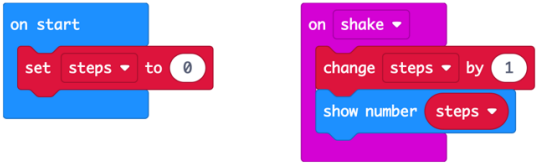



<p>2. Students will follow the process map given to code the Micro:bit.</p> <p>f. Students will utilize the actions process map to create a code process map</p>	
<p>3. Then coded correctly, every time the Micro:bit is shaken, the students will see an icon/image pop up on the screen</p> <p>g. Students will utilize the “on shake” function to initiate the game</p> <p>h. Students will utilize a “pick random” to create game</p> <p>i. Students will use the logic blocks to create different scenarios</p> <p>j. Students will then display the variable</p> <p>5. Utilize the battery pack to have students walk around and test out their Micro:bits</p>	

### Activity: Step Counter

**Estimated Time:** 20 minutes

<p>1. Students will code their Micro:bit to mimic that of a step counter</p> <p>k. Students will use the Micro:bit accelerometer to count how many times the Micro:bit has been shaken</p>	
--	--

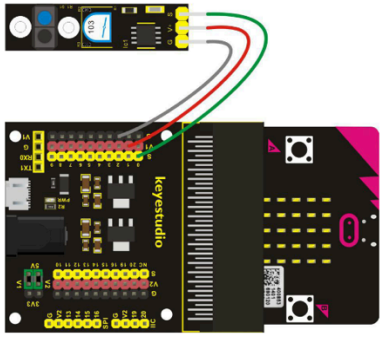
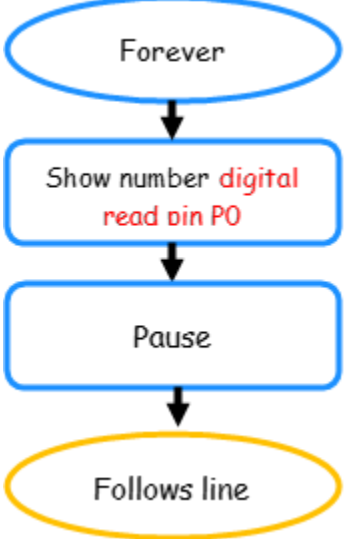
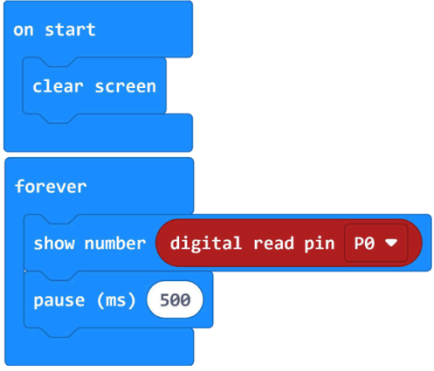
<p>2. Students will follow the process map given to code the Micro:bit.</p> <ol style="list-style-type: none"> <li>Students will utilize the actions process map to create a code process map</li> </ol>	
<p>3. When coded correctly, every time the Micro:bit accelerometer input senses a shake, the program increases the variable by 1, and shows the new number on the LED display output.</p> <ol style="list-style-type: none"> <li>Students will utilize the “on start” function to initiate the variable to 0</li> <li>Students will utilize “on shake” function to count each step</li> <li>Students will use the “change variable” function to increase the variable by 1 each time</li> <li>Students will then display the variable</li> </ol> <p>4. Utilize the battery pack to have students walk around and test out their Micro:bits</p> <ol style="list-style-type: none"> <li>See if students can fashion the Micro:bit to their shoe detect the steps without having to manually shake it</li> </ol>	 

### After Micro:bit Pet Extensions

The completion of the Micro:bit pet extensions is required to complete this activity. Students need to have experience wiring external sensors to the Micro:bit.

### Activity: Follow Black Line

**Estimated Time:** 20 minutes

<ol style="list-style-type: none"> <li>Students will wire the magnetic sensor to their Micro:bit. <ol style="list-style-type: none"> <li>Wiring example shown.</li> <li>The sensor will identify if there is a black line under it. This is done by detecting if there are infrared rays are emitted at a high or low level.</li> </ol> </li> </ol>	
<ol style="list-style-type: none"> <li>Students will follow process map given to code the Micro:bit.</li> </ol>	 <pre> graph TD     A([Forever]) --&gt; B[Show number digital read pin P0]     B --&gt; C[Pause]     C --&gt; D([Follows line]) </pre>
<ol style="list-style-type: none"> <li>When coded correctly, the sensor when detecting a black line will read the number 1, if reading any number besides 1, there is no line detected. <ol style="list-style-type: none"> <li>This is due to when a line is detected, the infrared rays are not emitted or have a low intensity. The sensor's signal will output a high level showing the number 1.</li> </ol> </li> <li>Have students test the code using different thicknesses of black lines.</li> </ol>	 <pre> on start   clear screen   forever     show number digital read pin P0     pause (ms) 500 </pre>

### After Technical Design Challenge

This activity is a great way to get students into programming python. Python is a language that could be considered the next step after block programming.

## Activity: Turtle

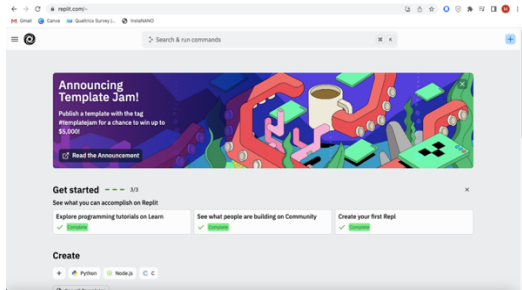
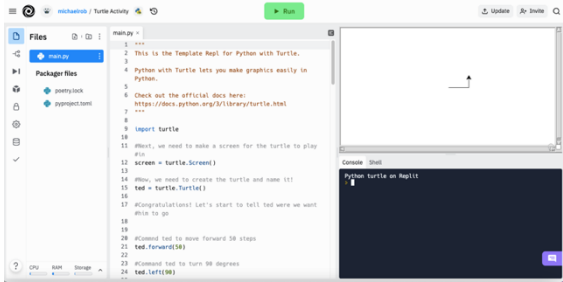
**Estimated Time: 60 minutes**

### Activity Goals:

- Develop a basic understanding of how to use the Turtle package in Python
- Learn how to customize and create a simple software application like a software engineer

### Activity Procedure:

- Have students work through the included activity posted on Repl.it
- Have students get creative with the code they learn in the activity
- Have students identify what the different functions presented in the activity do

1. Have students go to the the Repl.it site and find the activity named “Turtle Activity”	<p><a href="https://repl.it">repl.it</a></p> <p>Show the first 42 seconds of this video if able <a href="https://www.youtube.com/watch?v=pGpDFmVa4Uo">https://www.youtube.com/watch?v=pGpDFmVa4Uo</a></p>
2. Introduce the activity: You are to follow the activity and eventually create a drawing using the code learned	
3. After letting the students work through the activity, have the students identify the functions in the application	
4. Have students think about the actions that have to happen to get from the start to the end result	

The activity is entirely done in a script of code they can run on Repl.it, all the instructions are in there.

Basically, I have created a template in which the students will copy and paste lines of code in the appropriate order.

A big thing the students need to know how to do is to uncomment lines of code.

Students can either copy and paste whole commented code lines with the “#” and then delete the “#” when they want to use the command

Students can also only copy lines of code without including the “#”