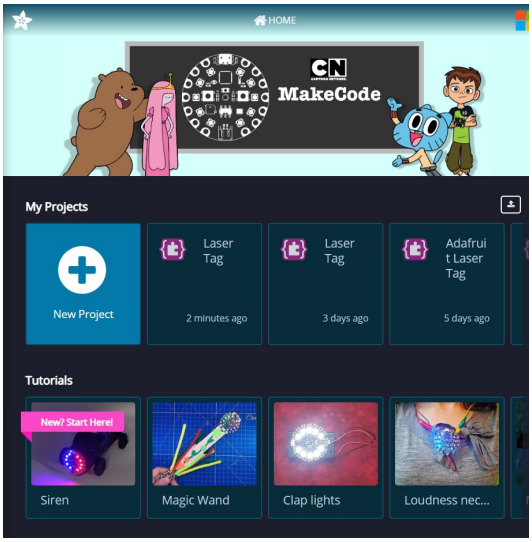
Day 2: Laser Tag - Programming

Step 1: Start a New Project

1.1. Go to https://makecode.adafruit.com/
1.2. Start a new project
1.3. Name your project



Step 2: Set Initial Loops

2.1. Click on “LOOPS”
2.2. Select “on start”. A “forever” loop is already in your workspace.

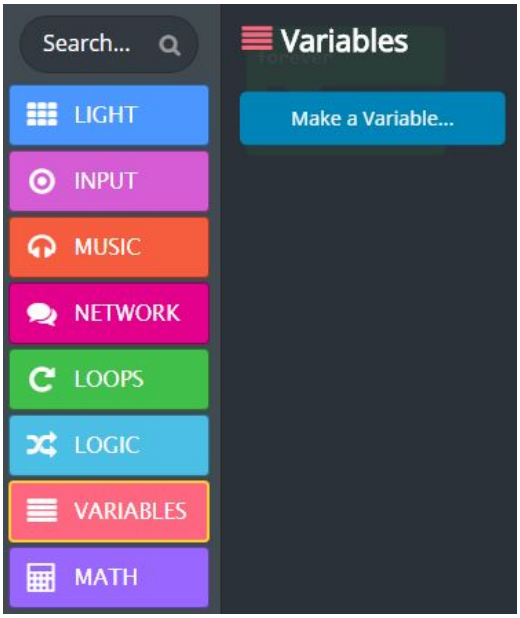
We constantly want to update what the neopixels look like to display a player’s points, so we need a “forever” loop that will constantly run.

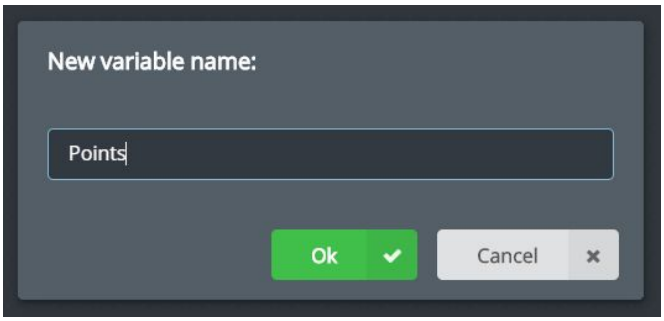
Similar to the “Dice” game we will use “On Start” to set initial variables for the game.

Step 3: Create Variables

- Variables serve as containers that store values we can use and change, later
- For laser tag, we need to keep track of a player's points. We will do that by creating a new variable and naming it "**Points"**.

3.1. Click on the button that says, "VARIABLES"
3.2. Click on "Make a Variable..."
3.3. Type the name of the variable as "Points" and click "OK"





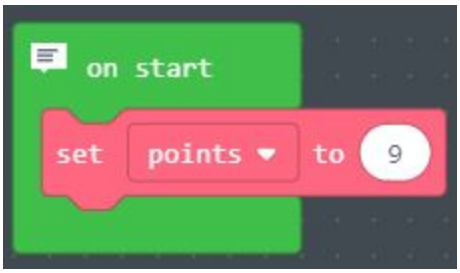
- We will be tagging another friend based on their distance from the CPX. To store this distance, we make a second variable **"Distance".**

3.4. Create a second variable and name it "Distance". We'll be keeping track of if we've been tagged based on other people's distance to your CPX.

Step 4: Set Variables

- Setting variables is what we call giving an initial value to a variable
- For our game, we want to show the player's points on the 10 neopixels on the circuit playground. We set the variable "Points" with a value of 10, because there 10 neopixels on the circuit playground.

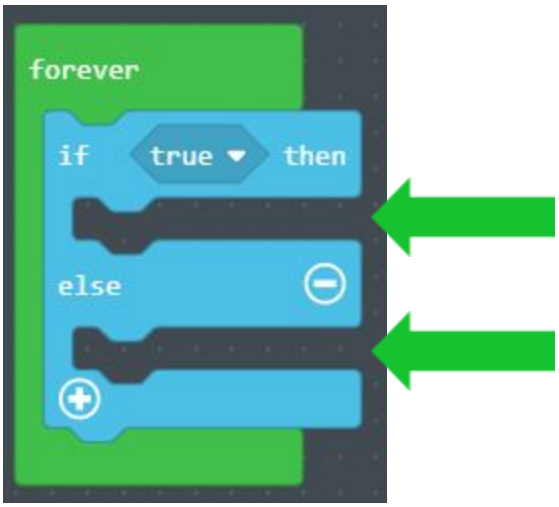
4.1. Click on VARIABLES and use the block that says, "set"
4.2. Change the number "0" to "10"



Step 5: Use Neopixels to show a Player's Points

To know when the game is over and a player is out of points, we need to constantly check if the "points" variable is greater than 0. Therefore, we add if/else logic in our forever loop.

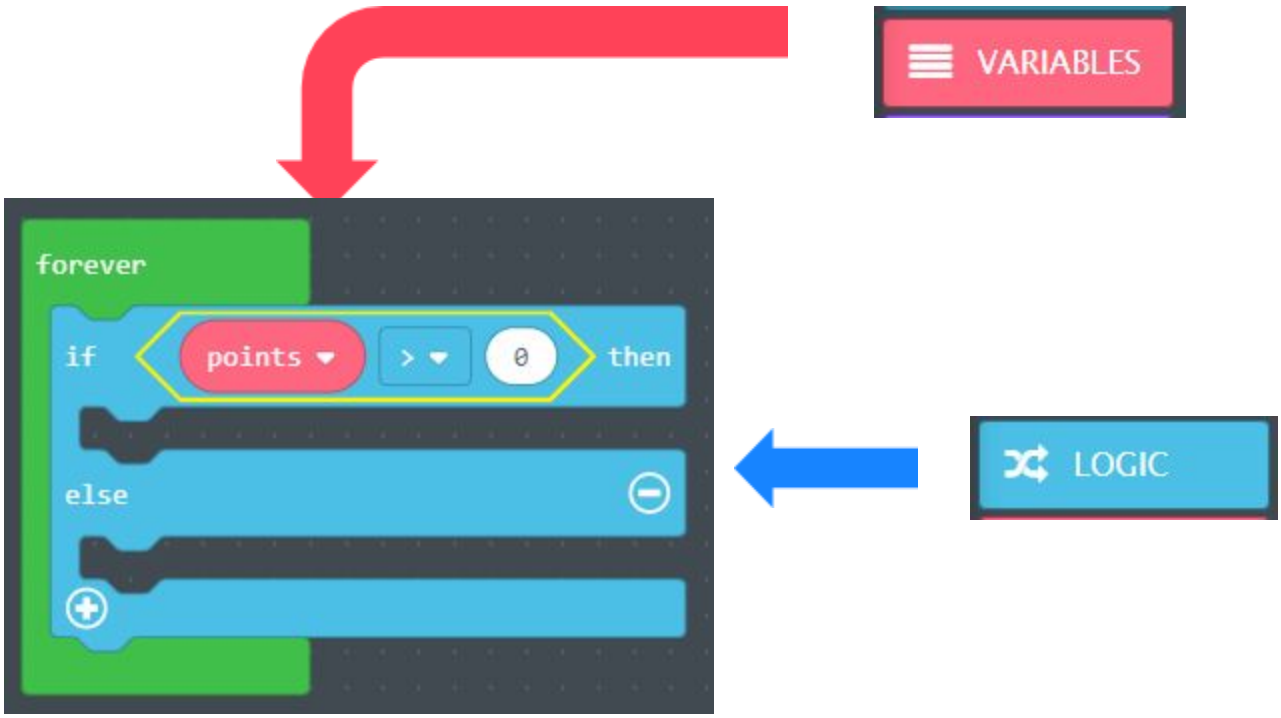
5.1. Click on "LOGIC" and select the second blue block that says, "if...else"



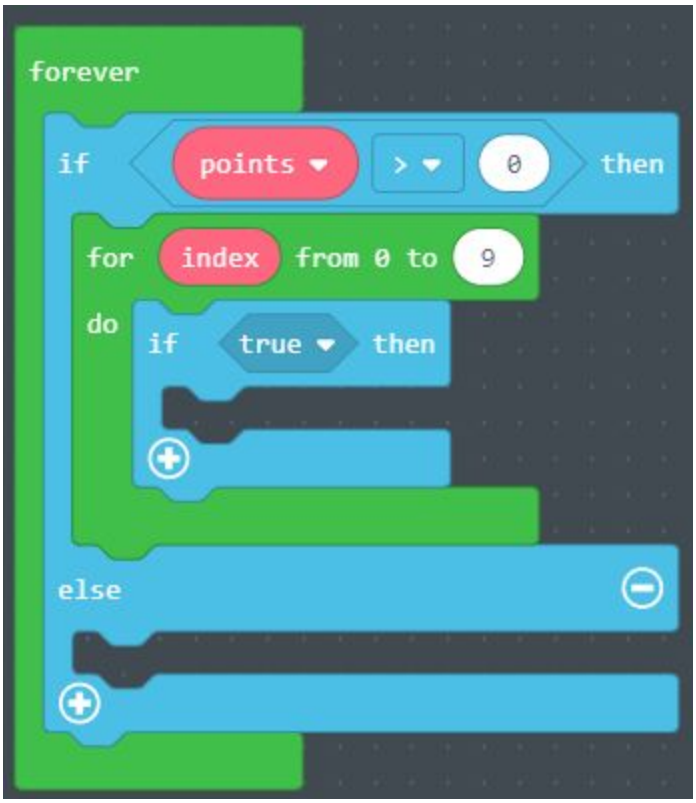
In this logic section we will have code to display a player's points through neopixels while they still have points left.

In this logic section we will have code to display when a player has no more points and has lost the game.

5.2. Add the variable “points” to the “if” block



5.3. Start a “for...do” loop to light up the 10 pixels



We need another loop to be able to light all of the neopixels at once.

We only want to show a neopixel for each of a player’s points. So what should we be checking inside this inner if statement to make that happen?

Step 6: Add Neopixels for Points and Endgame

- Use some creative light colors and sounds to indicate the player's current points (inside the inner "if…") and the end of game (inside the outer "else…").

Step 7: Setting up Input

We will use the **button A** as input to indicate someone is tagged

7.1. Click on "INPUT" and select the button A

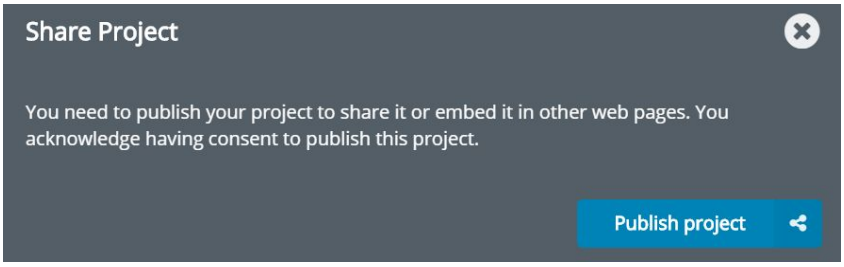
7.2. Click on "VARIABLES" and select "Change" and place it inside the "on button A click" block.

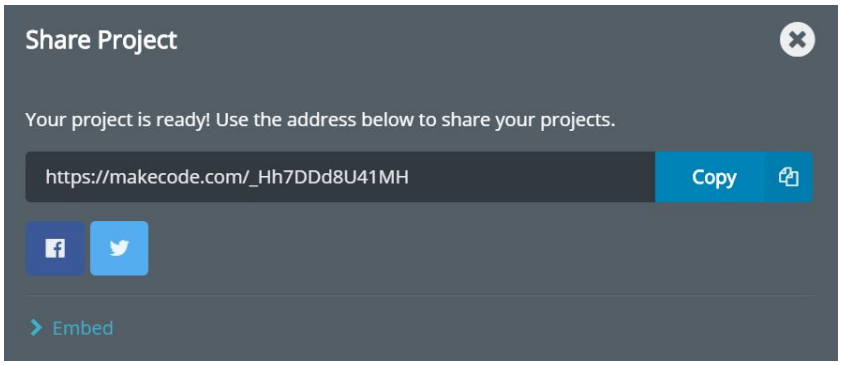
Test Your Code!

- To make sure our circuit playground shows the correct number of points on the neopixels, use the emulator to see which light colors turn come on.
- Press button A to see if the pixel changes color when you are “tagged”.

Save & Share Your Code

- Publish your code by clicking on “Share” on top of the page.
- Click “Publish project” and then “Copy” to copy the link to your code.
- Open **http://tiny.cc/hcde2**
- Paste the link to your code on the form, add your name, and submit the form



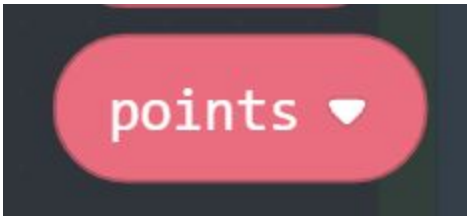


Day 3: Laser Tag with Distance Sensor

Recap

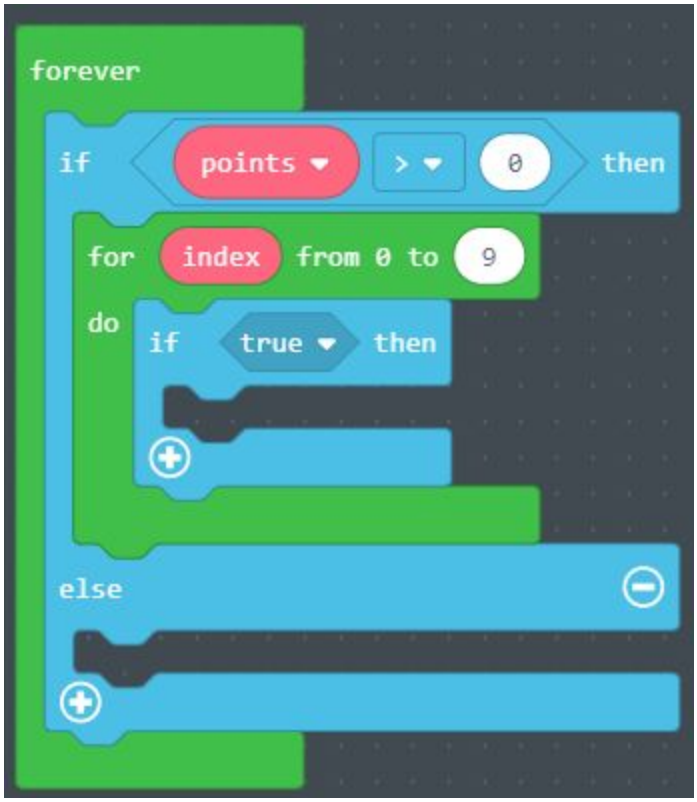
Variables

Keeping track of if we've been tagged based on other people's distance to your CPE



Keeps tracks of how many points you have

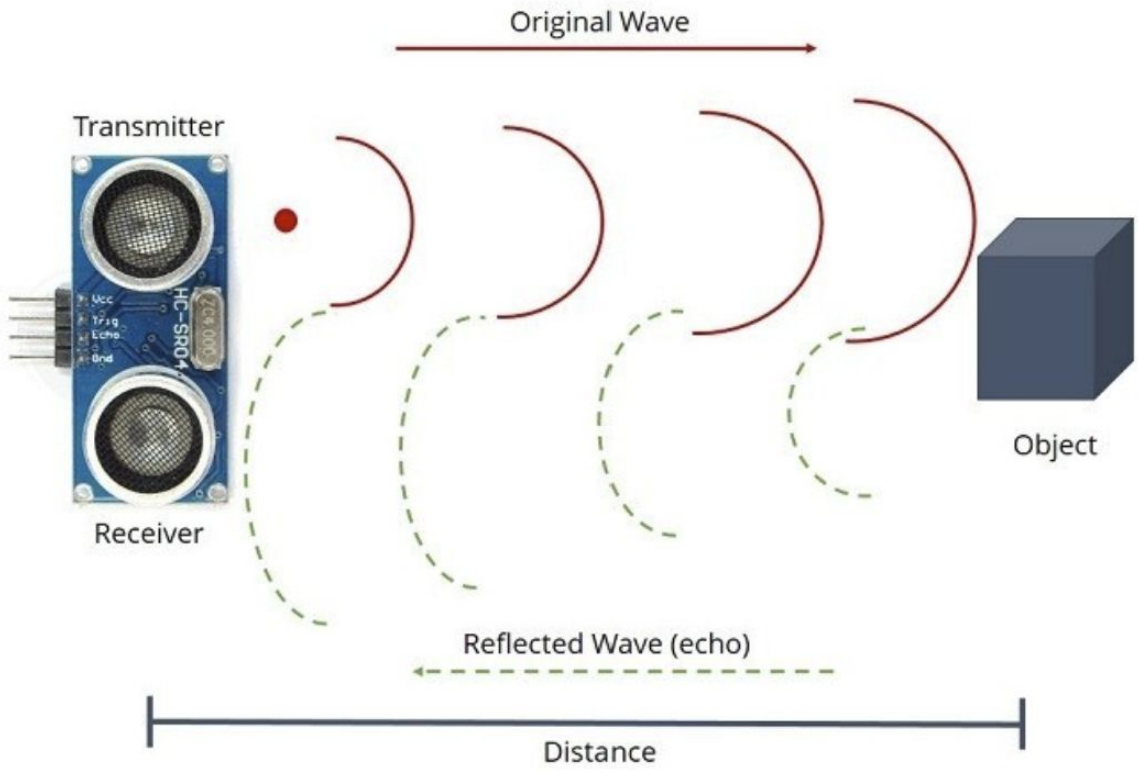
Neopixels with loops



Shows a neopixel for each of a player's points.

Step 8: Connecting the Distance Sensor

The distance sensor works by sending out continuous signals in front of it. If and when those signals get blocked, like by a hand moving in front of it, the distance sensor sends the information the CPX indicating there is an object in front of it.



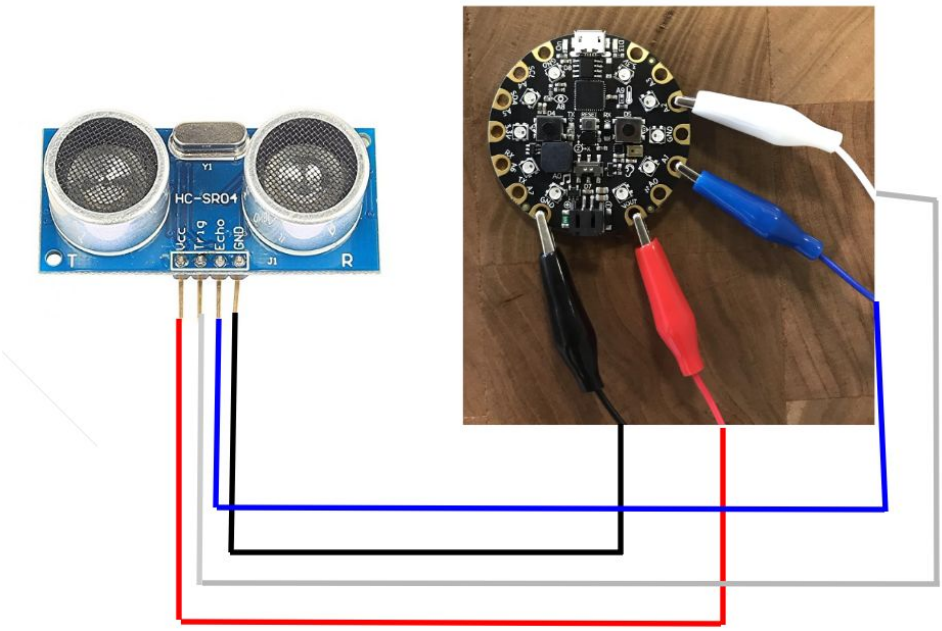
8.1. Gather your CPX, wires, and distance sensor

8.2. With a wire connect the “Gnd” pin on the distance sensor to the “GND” pin on the CPX with an alligator clip

8.3. Connect the “Trig” pin on the sensor to the “A2” pin on the CPX. This is the pin that sends a signal: a high-frequency sound.

8.4. Connect the “Echo” pin on the sensor to the “A1” pin on the CPX. This is the pin that receives the reflected sound.

8.5. Connect the “VCC” pin on the sensor to the “VOUT” pin on the CPX

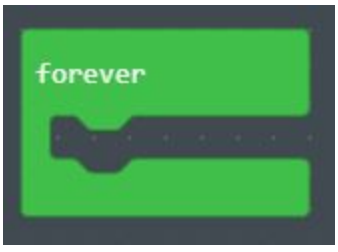


Go to: **makecode.adafruit.com/**

Step 9: Look for Input from the Distance Sensor

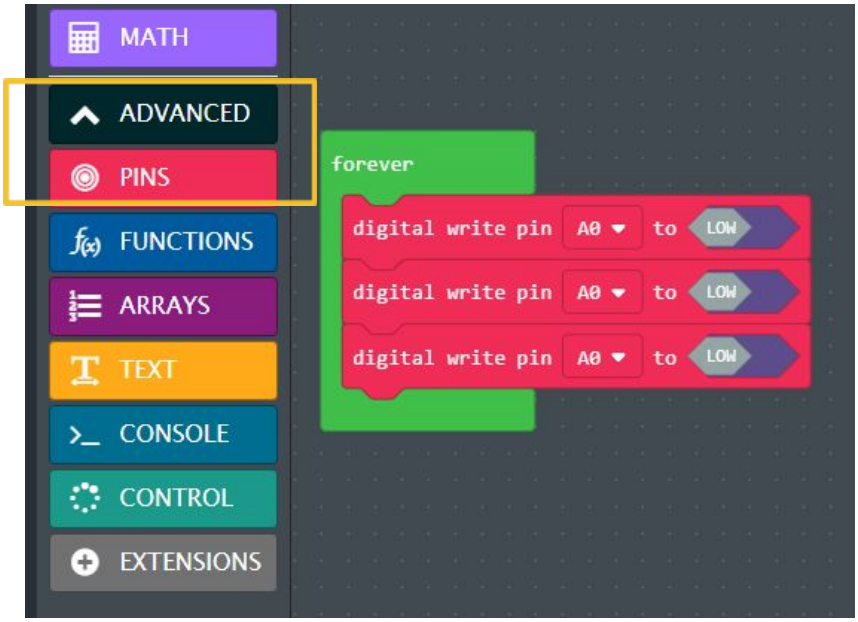
- We want to constantly get information from the distance sensor to check whether someone or something is close enough to a player to be “tagged” and lose points

  9.1. Select a new “forever” block from the green “LOOP” block.



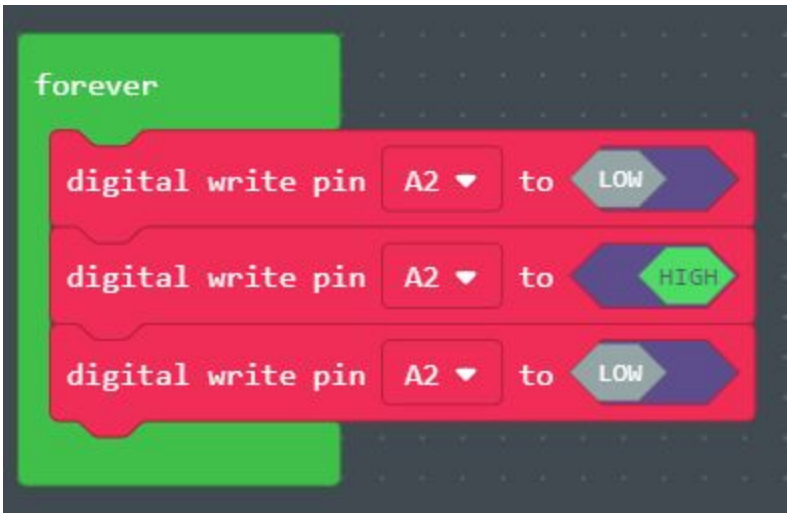
- In the new forever block, we need to continuously get information from the distance sensor which is connected to the CPX at pin A2. For this, we use a block called “digital write”.

9.2. Click on “Advanced” → “Pins” → select “digital write pin” block 3 times and place them inside the “forever” block.



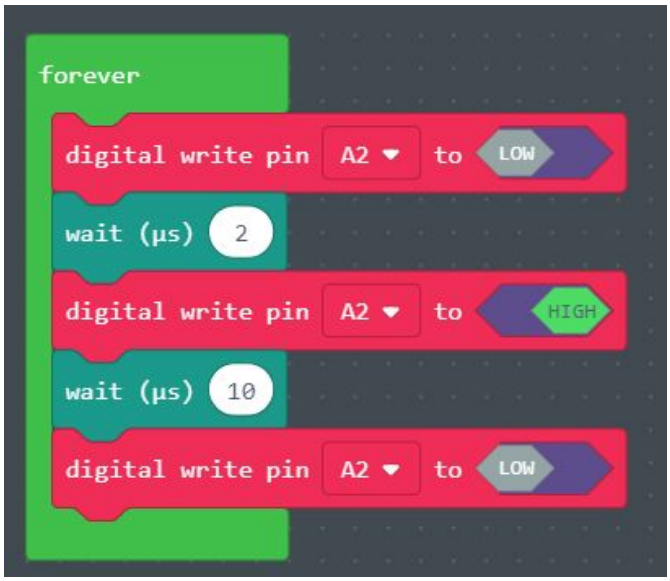
- We want to continuously check if the information on pin A2 is on a pulse pattern that is, goes from low to high and high to low.

9.3. Set all the pins to “A2” as the distance sensor is connected to pin A2 on the CPX. Then set the “write pin” block to “HIGH”.



- After each “digital write” block, the CPX should wait for some micro seconds to ensure commands don’t run over each other. To do so, we use the “wait” block.

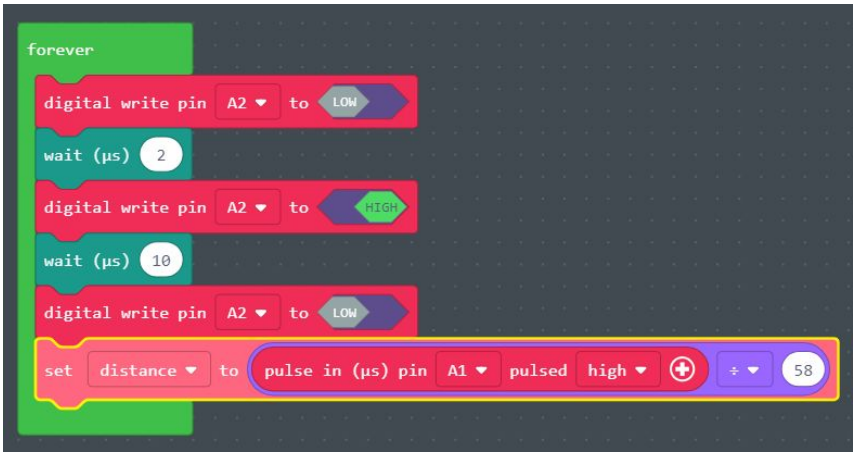
9.4. Go to “Advanced” → “Control” and select the “wait” block



Add the wait between each write pin to make sure the commands don’t run over each other.

Step 10: Update the Distance Variable

- Yesterday, we made a variable called **“Distance”.** Now we’re going to use this variable to keep track of how far away the distance sensor reads the objects surrounding it to be.
- The pulse block is used to read the signal from the sensor: a HIGH pulse whose duration represents the time (in microseconds) it took to send the signal and receive its echo off of an object.
- Dividing by 58 converts the metrics from the speed of sound to cm.



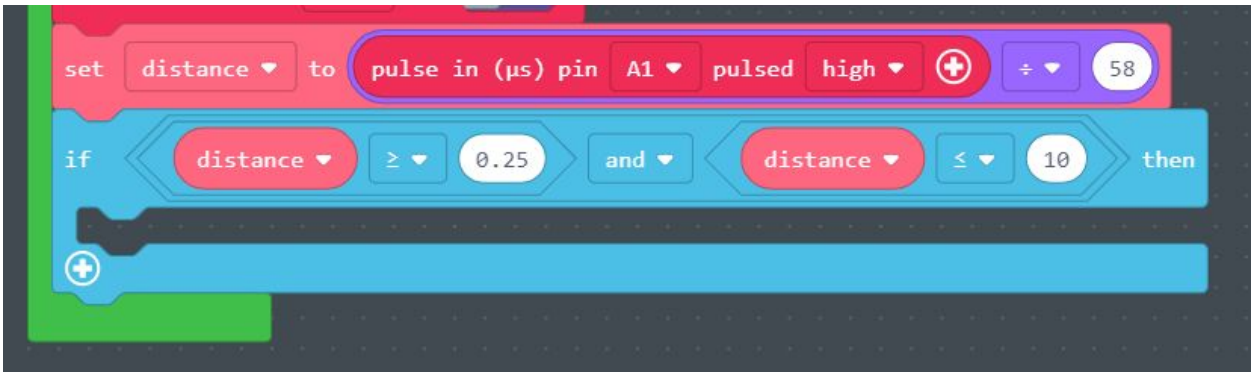
10.1. Go to variables so you can set the value of distance.

10.2. Go to math and get the division block.

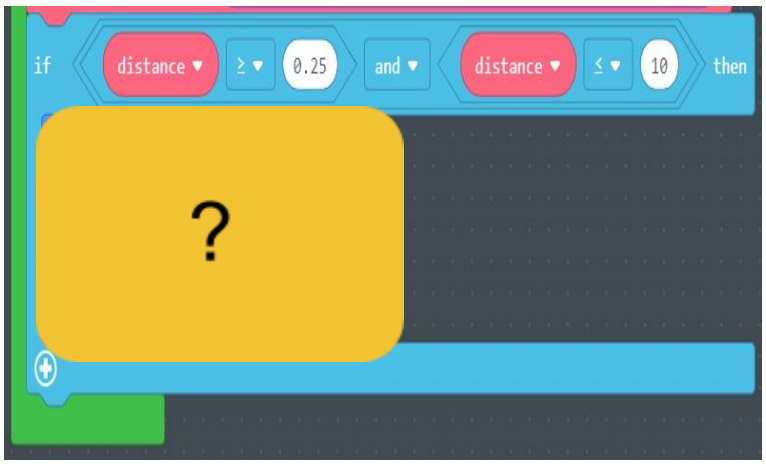
10.3. Go to “Advanced” → “Pins” and set the pulse in pin to “A1” and pulsed to “high”.

Step 11: Check if a Player was Tagged

11.1. Using an if-statement we want to check if the distance of some object or person is within a certain range of centimeters from the sensor



We chose 10cm to be the farthest someone can be to get "tagged", but this can be any number up to 80 cm.



11.2. Change the last neopixel to a different color to show the loss of a point visually.

11.3. Deduct a point from the player's score when they are signaled as "tagged". (remember the "Points" variable we set yesterday?)

Step 12: Game Reset Functionality

We want to allow a player to start a game over if they press the **button B.**

- A player's points has to be reset which means, set **points** to 0
- The light pixels have to be reset which means set to our starting color purple.
- Maybe you can add a special sound to play?

12.1. Choose button B to be the button that resets your points and add the light and sound blocks you want inside

Test Your Code!

- Now that we've done more of the initial coding of our game we want to make sure that it works!
- To test our code we might need to make a slight modification to our code.

1. Make sure the points variable in "On Start" is set to "10" if you forgot to reset it after our testing on Day 2.

- Now let's get the code onto the circuit playground.

1. Press the download button in the lower left corner of MakeCode.
2. Find your downloads folder, and find the download you just created of your CPX code.
3. Plug your CPX into the computer.
4. Hit the "Reset" button on your CPX
5. Drag the file to the side of the File Explorer where you should see something like "CPLAYBOOT"

6. Have fun playing with your laser tag game, or ask for help is something isn't working properly.

Step 13: Combine the Physical Pieces Together

Use velcro to connect the CPX and distance sensor to a battery pack so you can easily carry it around.