Watch Your Behavior

Adding Behavior Scripts

Learning Objectives

1. Explain why a behavior script is like a light switch.
2. Describe how to add a behavior script to a lens experience
3. Identify the difference between local and global scope

Behavior Script

Now that you’ve mastered the "look" of your Lens by arranging images, it’s time to add some "brain power." In the world of AR, things don't just sit there—they react! To make your Lens interactive, we use something called the Behavior Script.

What is a Behavior Script?

Think of the Behavior Script as a bridge. It connects an Action (something the user does) to a Response (something the Lens does).

Instead of writing complex computer code line-by-line, Lens Studio gives you a "plug-and-play" tool where you just choose from a list of options. It's like filling out a Mad Libs story: *"When the user [Taps the Screen], then [Play a Sound] or [Hide an Image]."*

How it Works: Trigger → Response

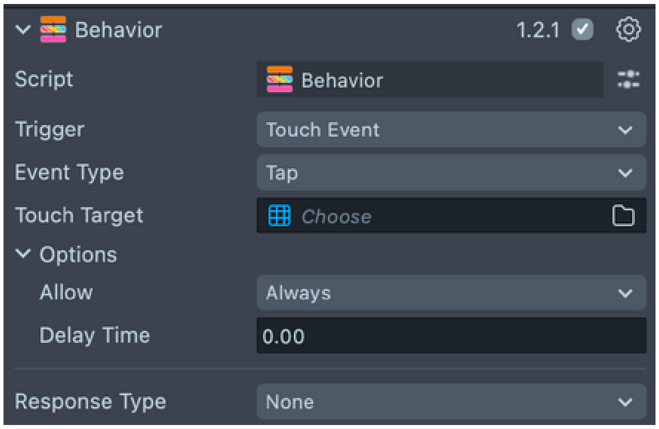
Every Behavior Script follows a simple two-step logic:

1. *The Trigger:* This is the "If." (If the user opens their mouth, if they tap the screen, or if the Lens starts).
2. *The Response*: This is the "Then." (Then show the hidden hat, then start the music, then make the star spin).

Why use it?

Without Behavior, a Lens is just a drawing. With Behavior, it becomes an Experience. You can use it to:

- Make a "Secret" image appear only when someone smiles.
- Change the background when the screen is tapped.
- Start an animation as soon as the Lens opens.



Behavior 1.2.1

Script: Behavior

Trigger: Touch Event

Event Type: Tap

Touch Target: Choose

Options

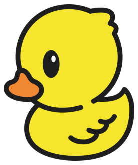
Allow: Always

Delay Time: 0.00

Response Type: None

Behavior is like a Light Switch.

- The Trigger is your finger flipping the switch.
- The Response is the light bulb turning on.
- The Behavior Script is the invisible wire behind the wall that connects the two!

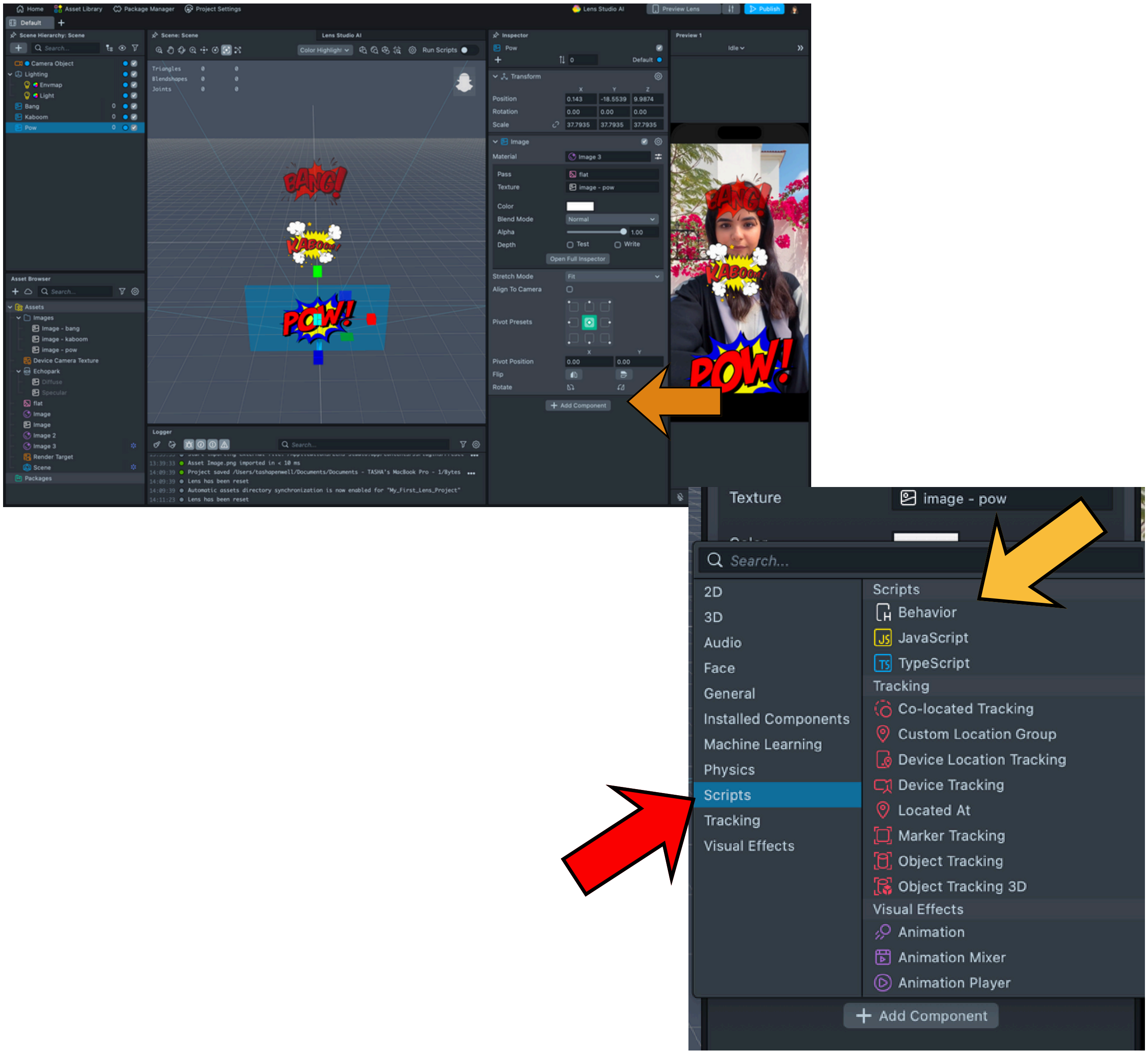


To add a Behavior Script, we are going to attach it directly to one of your objects. This tells Lens Studio, "This specific object now has a brain!"

How to Add the "Brain" (Behavior Script)

Follow these steps to find and attach the script.

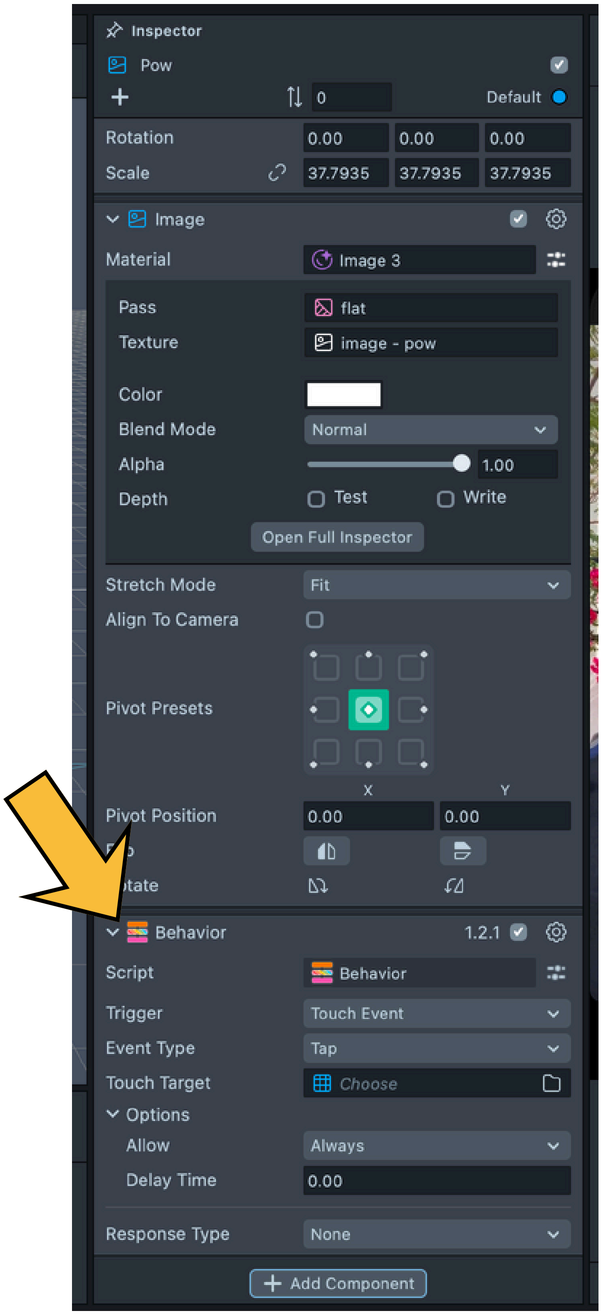
1. *Select Your Object*: In the Objects Panel, click on the image you want to make interactive (it will highlight in blue).
2. *The "Add Component" Button*: Look over at your Inspector Panel. At the very bottom, click the large blue + Add Component button.
3. *Follow the Categories*: Look at the list on the left side of the popup.
4. *Find "Script"*: Scroll down until you see the category labeled Script.
5. *Pick "Behavior"*: Hover your mouse over Script, and a second menu will slide out. Click on Behavior.



```
13:39:33 ● Asset Image.png imported in < 10 ms
14:09:39 ● Project saved /Users/tashopenwell/Documents/Documents - TASHA's MacBook Pro - 1/Bytes •••
14:09:39 ● Lens has been reset
14:09:39 ● Automatic assets directory synchronization is now enabled for "My_First_Lens_Project"
14:11:23 ● Lens has been reset
```

What to Look For

Once you click it, keep your eyes on the Inspector Panel. You will see a new gray box appear that says "Behavior" at the top.



Inspector

Pow ☑

+ 0 Default ●

Rotation 0.00 0.00 0.00

Scale 37.7935 37.7935 37.7935

Image ☑

Material: Image 3

Pass: flat

Texture: image - pow

Color

Blend Mode: Normal

Alpha: 1.00

Depth: □ Test □ Write

Open Full Inspector

Stretch Mode: Fit

Align To Camera □

Pivot Presets

Pivot Position: X 0.00, Y 0.00

F[illegible]

[illegible]tate

Behavior 1.2.1 ☑

Script: Behavior

Trigger: Touch Event

Event Type: Tap

Touch Target: Choose

Options

Allow: Always

Delay Time: 0.00

Response Type: None

+ Add Component

Notice that the object in your Objects Panel is still highlighted in blue. This is your way of telling Lens Studio: "I am giving these instructions ONLY to this specific image."

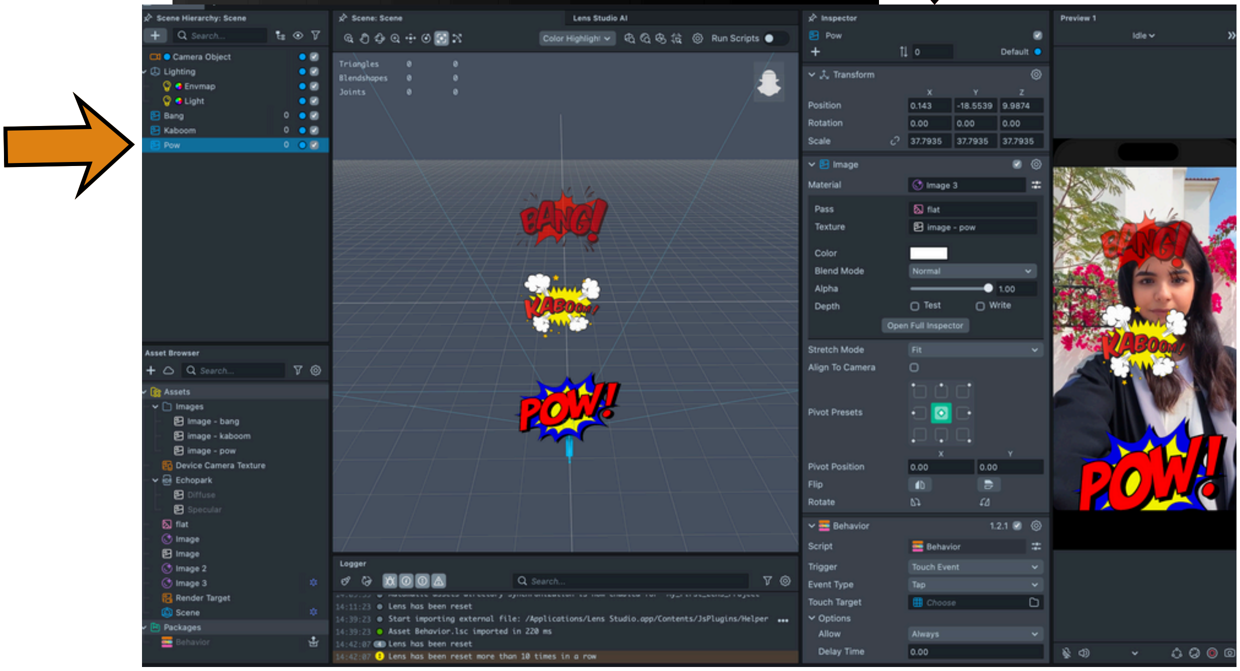
Local vs. Global: The Secret of "Scope"

In professional programming (like JavaScript, which is the language behind Lens Studio), we call this Scope. It's the difference between a "Local" command and a "Global" command.

- *Local* (What you are doing now): When you add a Behavior Script to one image, it is Local to that object. If you tell that script to "Turn Green," your other images won't change. They are minding their own business!
- *Global*: A Global command would be like a "Master Switch" that talks to every single object in the Lens at the same time (like the sun rising or music playing for the whole scene).

Why This Matters for Your Project

If you want three different images to react when they are tapped, you can't just add one script and hope for the best.



Inspector: Pow

Transform

	X	Y	Z
Position	0.143	-18.5539	9.9874
Rotation	0.00	0.00	0.00
Scale	37.7935	37.7935	37.7935

Image
Material: Image 3
Pass: flat
Texture: image - pow
Blend Mode: Normal
Alpha: 1.00
Depth: □ Test □ Write
Open Full Inspector
Stretch Mode: Fit
Align To Camera: □
Pivot Presets
Pivot Position: X 0.00 Y 0.00
Flip
Rotate

Behavior 1.2.1
Script: Behavior
Trigger: Touch Event
Event Type: Tap
Touch Target: Choose
Options
Allow: Always
Delay Time: 0.00

Logger

```
14:11:23 ● Lens has been reset
14:39:23 ● Start importing external file: /Applications/Lens Studio.app/Contents/JsPlugins/Helper •••
14:39:23 ● Asset Behavior.lsc imported in 220 ms
14:42:07 ● Lens has been reset
14:42:07 ! Lens has been reset more than 10 times in a row
```

Think of your objects like Students in a Classroom.

- Local Scope is like a teacher whispering a secret to one student. Only that student knows what to do.
- Global Scope is like the Principal on the Loudspeaker. Everyone in the whole school hears the announcement and follows the instruction at the same time!

Now that your object has a "brain" (the Behavior Script), we need to give it a Trigger. This is the "If" part of our logic: If the user does X, then do Y.

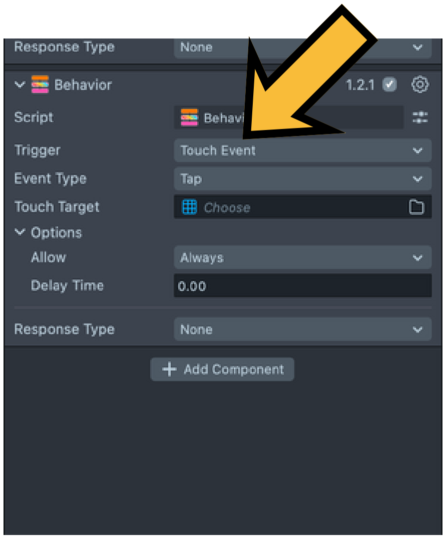
The "Touch" Trigger: Screen Tap

The most common way people interact with AR is by tapping their phone screen. Let’s set up our script to listen for that specific touch.

Setting the Trigger

In your Inspector Panel, look at the top of the Behavior Script box:

- *Trigger Dropdown*: Click the box next to Trigger. Select the drop down to find and select “Touch Event” (it may already be selected).
- *Event Type Dropdown:* Event types are based on the Trigger selected. Choose "Tap":
- *Touch Target:* tells the script where the user has to tap. Leaving it at default means user can tap anywhere.
- *Options* > *Allow:* controls how many times the user can trigger the action.
- *Options* > *Delay Time:* How long before behavior begins.
- *Response Type:* Provides a variety of options to set as instructions on what happens when a triggered event (such as tapping the screen) takes place.

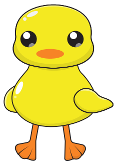


Response Type	None
Behavior	1.2.1
Script	Behavi
Trigger	Touch Event
Event Type	Tap
Touch Target	Choose
Options	
Allow	Always
Delay Time	0.00
Response Type	None

Real-World Logic: Why "Tap"?

In computer science, we call this Event-Driven Programming. The computer (or phone) is "listening" for a specific event to happen.

- *The Event*: A finger touching the glass screen.
- *The Listener*: Your Behavior Script.



Pro-Tip: Tap vs. Touch

You might have noticed other options like "Touch Start" or "Touch End."

- Touch Start happens the instant your finger hits the glass.
- Tap only happens if you touch and let go quickly. For most games and stickers, Tap is the most "natural" feeling choice for users!

Now that we have the Trigger (the "If"), it is time to set the Response Type (the "Then"). This is the actual action that will change your image's look when the screen is tapped.

We are going to use “Set Scale” to make your image grow or shrink instantly!

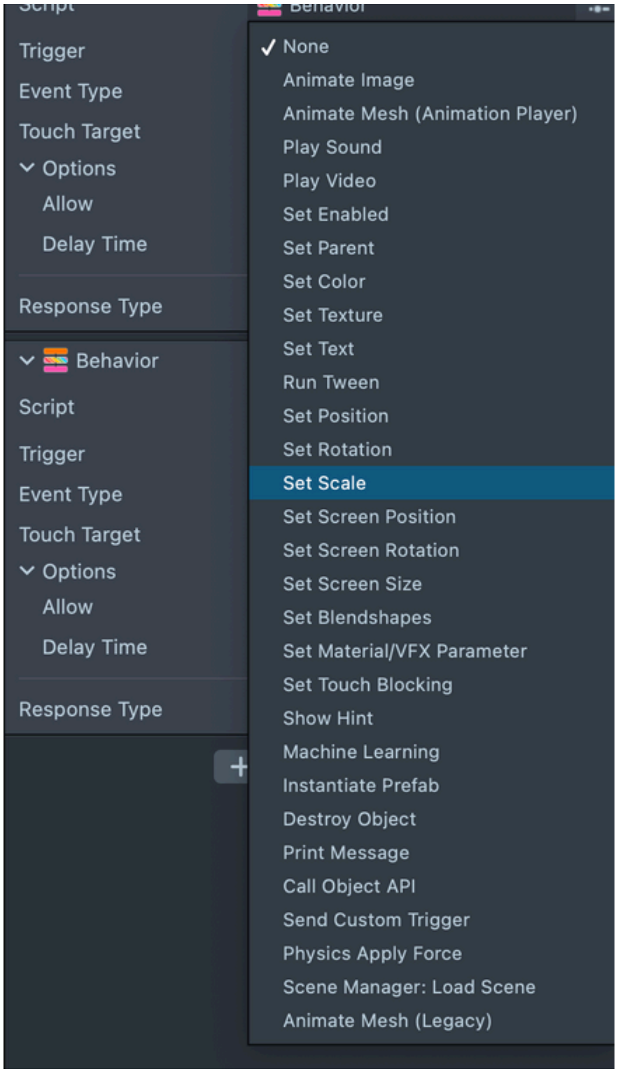
Response Type: Set Scale

Set Scale tells the Behavior Script to change the size of an object to a specific new measurement.

Setting the Response

In the Response Type section of your Behavior Script:

1. Click the Dropdown: By default, it might say "None" or "Set Visible." Click the box to see the list.
2. Choose "Set Scale": Click on Set Scale.

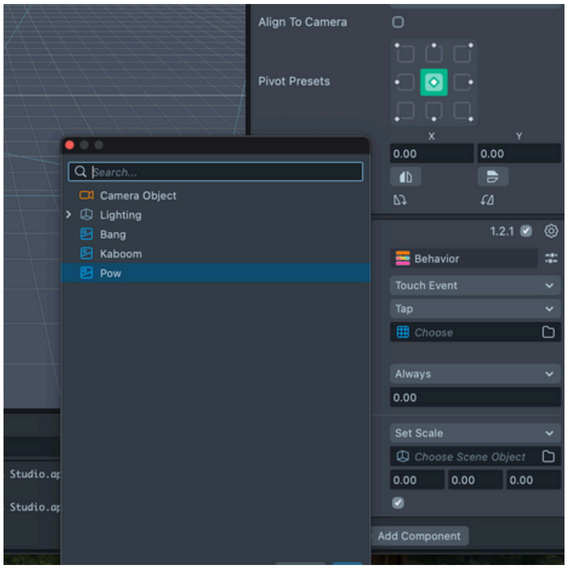


The "Target" Logic

Once you pick Set Scale, you need to set some more parameters.

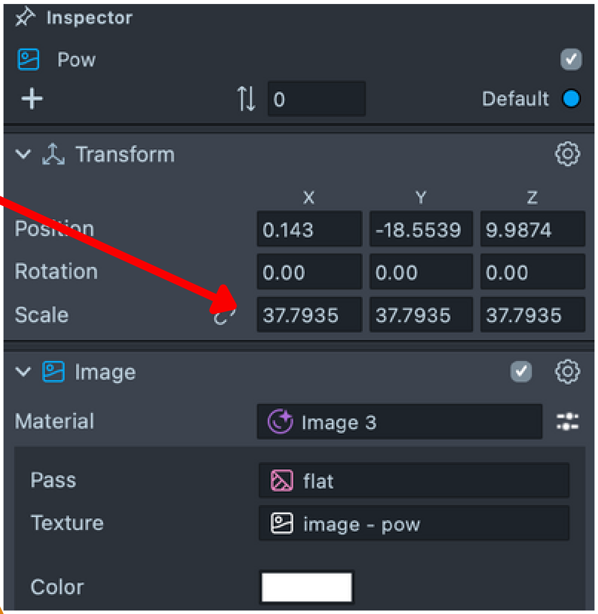
Object to Scale: Click to display options. For simplicity’s sake, select the image that aligns with the object you’re on. You can choose any object if you’re in the mood to experiment though.

Scale: You will see three boxes representing X, Y, and Z values for the location of the object.



Look in the Transform component at the top of the Inspector Panel to see the size of the object before any trigger is activated.

Set the Scale (X,Y,Z) to a size that is larger or smaller than the original size of the image.



Inspector

Pow ☑

0 Default

Transform

	X	Y	Z
Position	0.143	-18.5539	9.9874
Rotation	0.00	0.00	0.00
Scale	37.7935	37.7935	37.7935

Image ☑

Material: Image 3

Pass: flat

Texture: image - pow

Color

In the world of 3D design, every object has two different ways of looking at its position and size: World and Local.

World vs. Local: The Perspective Shift

World (Checkbox Unchecked)

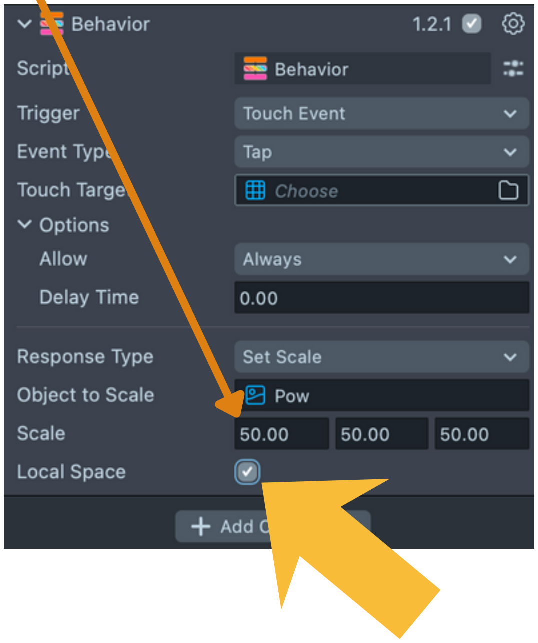
If the box is unchecked, the script uses "Global" coordinates.

- It looks at the entire 3D scene as its map.
- If you set the scale to 2.00, it says, "I don't care how big you were before; I want you to be exactly 2 times the size of a standard starting object in this world.”

Local (Checkbox Checked)

If the box is checked, the script only cares about the object's parent.

- It looks at the object's current state as the starting point.
- It’s like saying, "Double the size of what I am right now," rather than "Double the size of a standard unit."



Behavior 1.2.1 ☑

Script: Behavior

Trigger: Touch Event

Event Type: Tap

Touch Target: Choose

Options

Allow: Always

Delay Time: 0.00

Response Type: Set Scale

Object to Scale: Pow

Scale: 50.00 50.00 50.00

Local Space: ☑

Why use Local?

For most beginners, Local is the "safer" bet when you are resizing objects.

If you have an image that you already scaled down to 0.5 in the Scene Panel because it was too big, and then you trigger a World scale of 1.0, your image will suddenly "pop" back to its original giant size! If you use Local, it respects the work you've already done to arrange your scene.

Knowledge Check

Behaviors in AR

Instructions: Answer the following questions to verify your understanding of how AR "thinks" and reacts.

1. The Light Switch Analogy

Why is a Behavior Script often compared to a simple light switch?

A. Because it can only be used to turn the lights on and off in your 3D scene.
B. Because it waits for a specific Trigger (like a tap) to "flip" an action into motion.
C. Because it requires a physical battery to be connected to your computer.
D. Because it makes the entire screen turn bright white when activated.

2. Adding the Brains

Where do you go first to add a new Behavior script to your lens experience?

A. The Logger Panel
B. The [+] button in the Objects Panel
C. The Preview Panel
D. The "Help" menu at the top of the screen

3. Understanding "Scope"

In coding, what does the term Scope refer to?

A. The physical size of the 3D grid in Lens Studio.
B. The area of the code where a specific variable can be "seen" and used.
C. The magnifying glass tool used to zoom in on textures.
D. The total number of images in your Asset Browser.

Stop and Think

Who is the “Brain”?

As you build your AR experience, it’s easy to feel like the computer is "smart" because it knows exactly when you tap the screen. But there is a big difference between a computer that follows a plan and a computer that makes a guess.

The Great Distinction: Programming vs. AI

In 2026, we hear the word "AI" everywhere. But in this workbook, you are playing two different roles: the Architect and the Trainer.

1. The Script (The Architect's Plan)

The Behavior scripts you are adding are Deterministic. This is a fancy way of saying "100% predictable."

- *Logic*: If the user taps, the "POW" sticker appears. Every. Single. Time.
- *The "Brain":* YOU are the brain. You wrote the rule, and the computer is just a loyal follower.

2. The AI (The Trainer's Guess)

The part of Lens Studio that "sees" your face or "finds" the ground is Artificial Intelligence (AI).

- *Logic*: The computer looks at millions of pixels and says, "I am 98% sure this is a human nose."
- *The "Brain"*: The Computer is the brain. It learned from millions of examples to make an educated guess.

Critical Thinking Question: If your "POW" sticker appears in the wrong place, is it because the AI failed to find your face correctly, or because your Script gave the wrong instructions for where to put it?

____

____

____

____

____