



Smithsonian
Science Education Center

A Weighty Problem

Teacher Scratch Guide

Smithsonian Science for Computational Thinking™
© Smithsonian Institution 2023

Welcome!

Project Goals

The Smithsonian Science Education Center has developed this Scratch-based project to give students the chance to exercise both their computational thinking skills and their creativity. Students will use the elements and blocks of code provided through this project, along with what is already available within Scratch, to create an animation that tells the story of an investigation they recently completed in class. Students will need to use three of these four computer programming code structures: sequences, events, loops, and conditionals. They will place and move at least one sprite, enter three variables from their investigation to be displayed on a prop and bar graph (using a premade function block), and incorporate textual output such as speech bubbles throughout the animation.

Fifth grade students are working towards this CSTA standard:

1B-1AP-10: Create programs that include sequence, events, loops and conditionals.

The student assignment includes these requirements:

- There should be at least one character
- That character should move at least once
- They should use at least three different code structures representing sequences, events, loops, and conditionals.

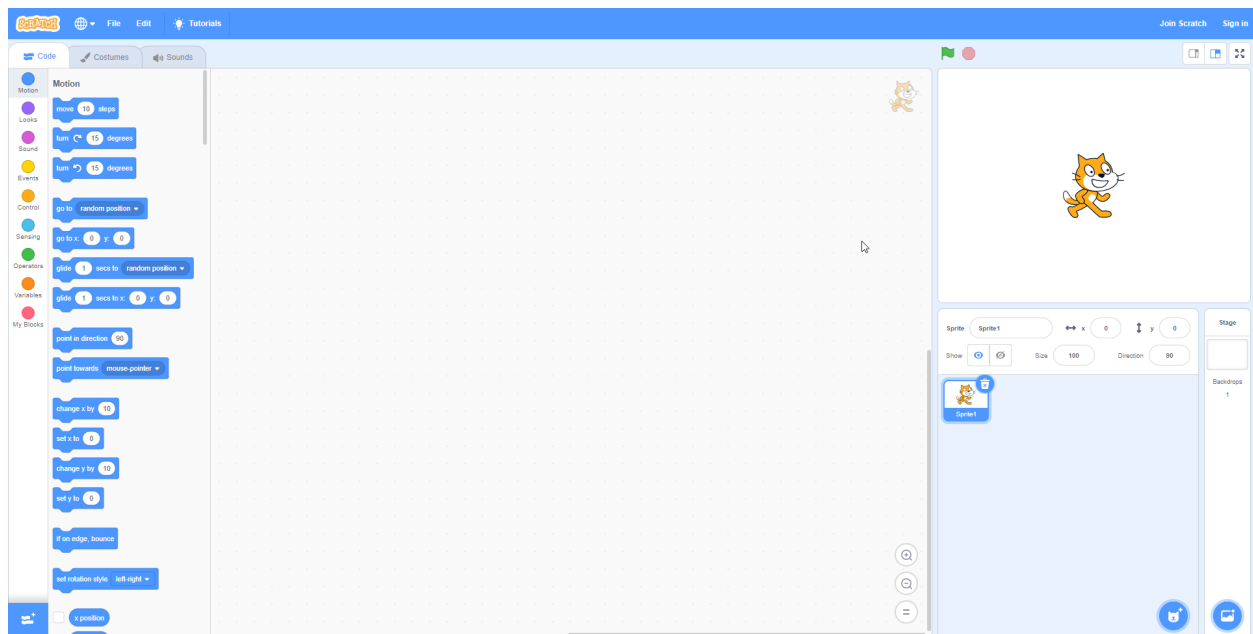
On the following pages, you'll find introductory information on the Scratch program, instructions on how to help your students through the experience, and a few additional tips and tricks that may come in handy.

Learning the Basics

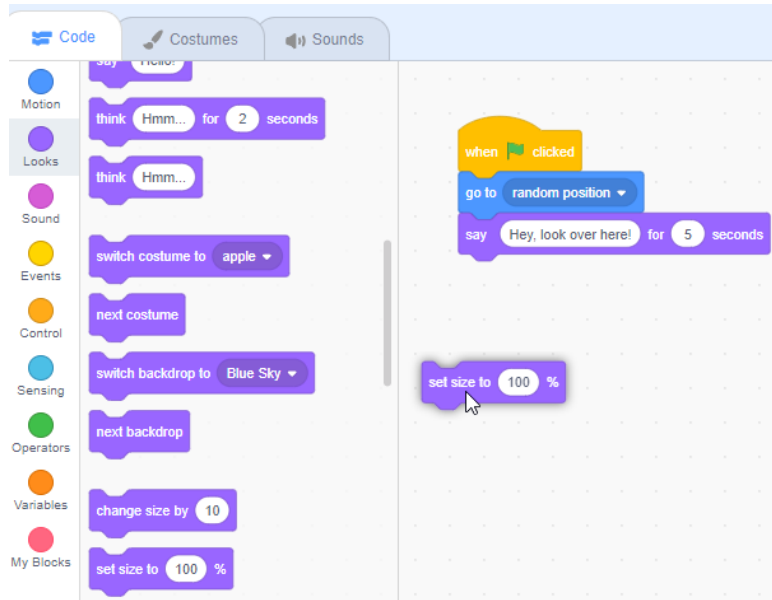
Keep in mind as your students work that it's ok to make mistakes! Nothing your student does with their project will impact their classmates' projects, and they can always restart or undo actions if the editor becomes cluttered.

Scratch Code Editor

Scratch is a visual programming language that utilizes lego-style code blocks and sprites to create animations, video games, and interactive media.



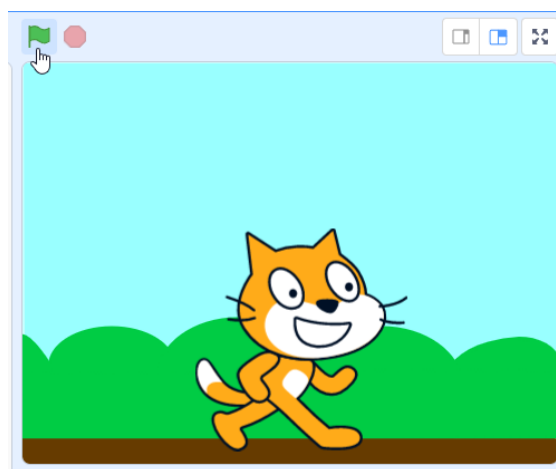
The empty white canvas above is where code blocks are placed. You can drag and drop code blocks from the left hand "Code" menu into your canvas to begin programming your animation.



Code blocks in Scratch run from top to bottom, typically starting with some sort of Event block, such as “when flag clicked.” Following this event, the code blocks will affect your sprites in order of appearance in the block stack. You can use Control blocks to adjust the flow of your code. More information about Control blocks is provided later in the document.

The Viewport

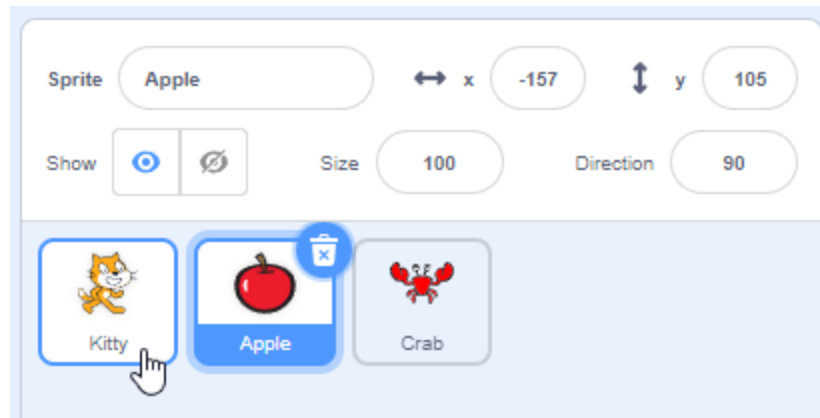
On the right hand side of the window, you will see your viewport. The viewport can be considered your “stage”- where the sprites you add act out the code you develop. Above the viewport is a green flag icon, and a red stop icon.



Clicking on the green flag icon will begin executing your code from the very beginning. At any time you can stop your program by clicking the stop icon.

Sprite Manager

Below your viewport is a list of sprites in your project. Every object used in the animation - from characters to props - will be represented as sprites in this section. Selecting a sprite from this manager will swap your canvas to show the code affiliated with that specific sprite. Below the viewport, you will also see information on the selected sprite's X and Y position, size, and rotation.



Stages

On the absolute far right, next to your sprite manager, you will see a section labeled "Stage." Click on this section, then go to the "Backdrops" tab (next to "Code") to see the available list of scene backgrounds.



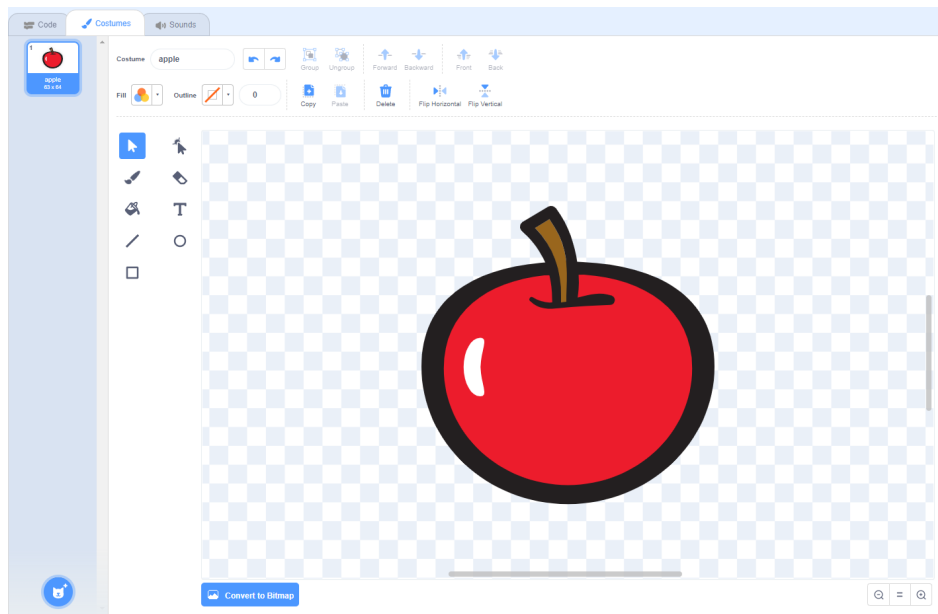
Just like with sprites, stages can contain code blocks, albeit a limited collection.

Costumes and Sounds

Back on the left side of your editor, next to the "Code" menu, you'll see two more tabs. When a sprite is selected, these tabs are "Costumes" and "Sounds". When the stage is selected, these tabs are "Backdrops" and "Sounds."

The "Costumes" menu will allow you to customize sprites and apply different styles. By using Scratch's built-in photo editing tools, uploading an image, or selecting

other preset costumes, you can then use the *Looks > Switch Costume to X* or *Looks > Next Costume* blocks to change the look of your sprite dynamically.



Similarly, the “Backdrops” menu is where you will go to customize the look of your stages and upload new backgrounds.

The “Sounds” menu will allow you to select a sound effect, or upload your own, for use in your project.

To test your code, you can click on the green flag listed above your viewport. Remember, this will run the code from the beginning. You can click on a block of code in your canvas to test one specific part of your program.

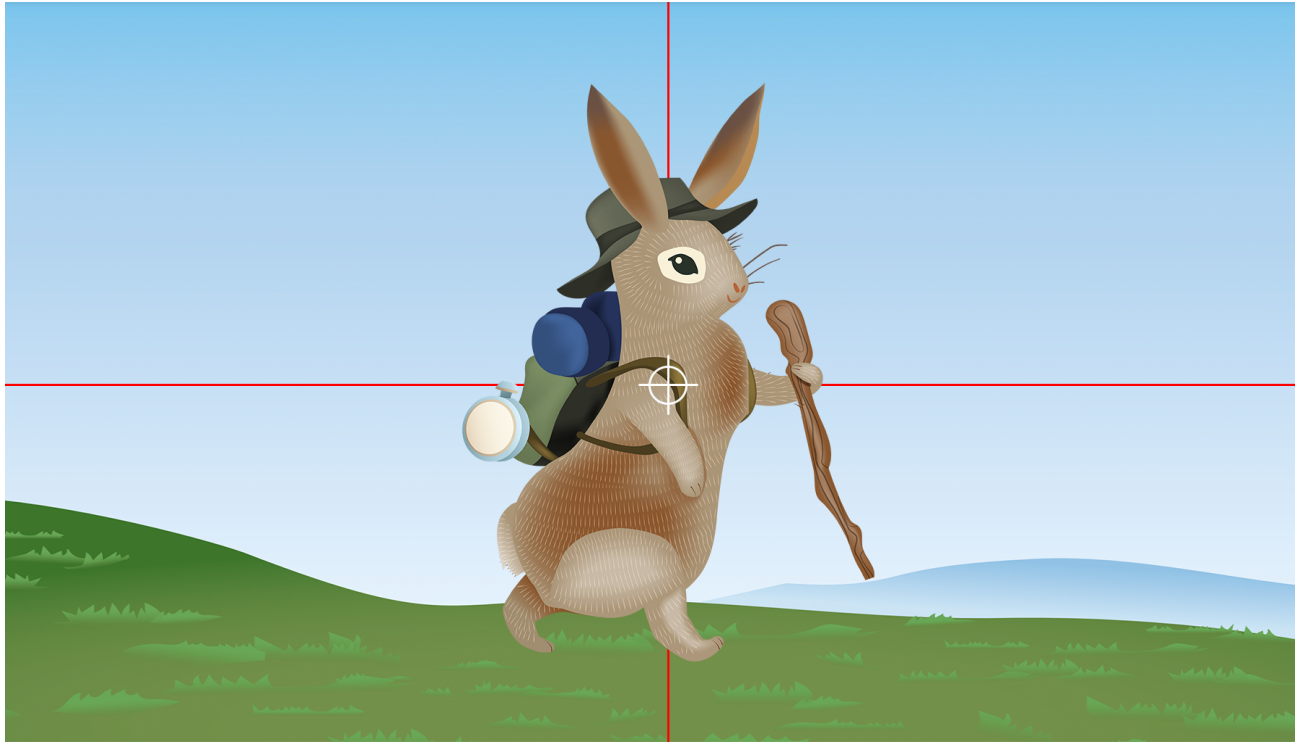
XY Coordinates

Students are expected to move a character around the scene, so they will need to know how to position a sprite within the scene.

The viewport is 480 x 360 in resolution. The minimum X value a sprite can be positioned at is -248, while the maximum X value is 247. The minimum Y value a sprite can be positioned at is -187 while the maximum Y value is 187.

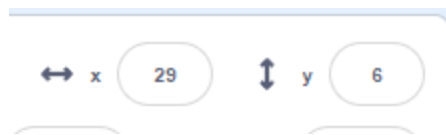
With Scratch, sprites are positioned on to the viewport using an X/Y coordinate system. You can think of this just like any other grid system! The origin of the viewport coordinate system is at the center of the viewport.

The coordinates of a sprite refer to the center point of the sprite’s image. You can manually adjust the position of your sprite’s origin points (the “center” of the sprite) by clicking and dragging it in your sprite’s “Costume” menu.



The origin point or “center” of your sprite is shown as a crosshair icon on your canvas in the Costume menu.

To position a sprite properly onto your viewport, you can simply click and drag it anywhere on your stage. **This will automatically update the X and Y value** listed on your sprite manager. Alternatively, if you know the exact X and Y coordinate you want the sprite placed at, you can enter these values in your sprite manager.



Once you have your sprite in a spot you like, make sure to update your code blocks to ensure it won't move away when you click the green flag! The *Motion > Go to X/Y* block is used to place the sprite.

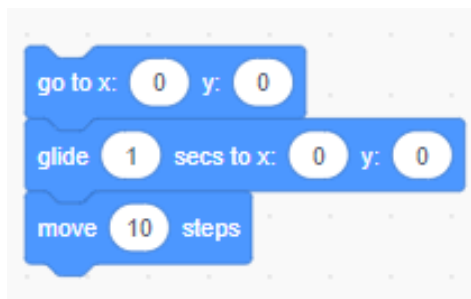
Starting Blocks

There are a wide variety of code blocks that students will be able to utilize in their program. Blocks are organized by type and are color coded. Below are some of the most commonly used blocks.

Motion (Blue)

This group of code blocks influence the movement and position of your sprites. Motion blocks are only available to sprites, not to backdrops.

- Go to X/Y
 - This block will set the position of your sprite. It requires you to input an X and Y coordinate.
- Glide x secs to X/Y
 - This block will move your sprite to the inputted X and Y coordinate. It will slide to this position in the duration inputted.
- Move x steps
 - This will move the sprite by a specific, user determined amount on the X axis. One step = +1X, i.e. 50 steps would move you from X=0 to X=50.

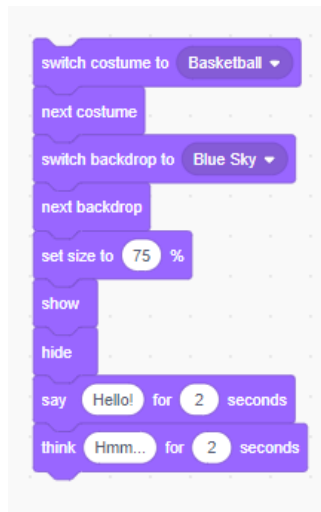


Looks (Purple)

This group of code blocks allows students to adjust the visuals of sprites, by changing costumes, applying effects, adjusting the size, and more. Sprites can also have speech bubbles or thought bubbles appear for dialogue.

- Switch costume to x
 - This block will let you swap to a separate costume of your choice.
- Next costume
 - This block will swap your sprite to the next costume in order on your “Costume” menu.
- Switch backdrop to x
 - This block will let you swap to a different backdrop of your choice.
- Next backdrop
 - This backdrop will swap your backdrop to the next costume in order on your “Costume” menu
- Set size to x%
 - This block will let you set the size of your sprite, based on a percentage value.
- Show
 - This block will show your sprite on the stage.
- Hide
 - This block will hide your sprite from appearing on stage.
- Say x for x seconds

- This block will allow you to make a speech bubble appear above your sprite. You can enter any text you would like into the speech bubble, and it will disappear after the inputted amount of seconds.
- Think x for x seconds
 - Similar to the “Say x for x seconds”, this block functions similarly, with the only exception being instead of a speech bubble, a thought bubble houses your text instead.



Sound (Lavender)

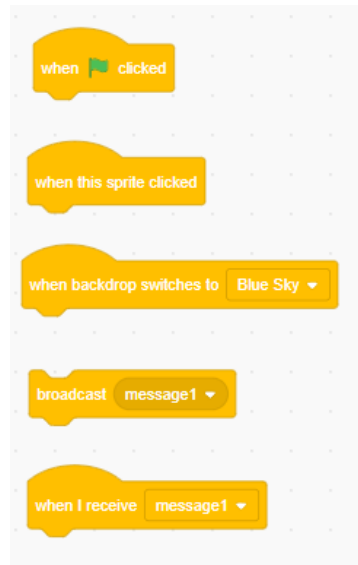
This project does not utilize any sound blocks.

Events (Yellow)

This group of code blocks act as starting points for your scripts. These blocks will detect when specific events occur throughout the runtime of your program, and trigger other code.

- When green flag clicked
 - This code block will begin running any code placed under it when the green flag is clicked.
- When this sprite clicked
 - This code block will begin running any code placed under it when the sprite is clicked on.
- When backdrop switches to x
 - This code block will begin running any code placed under it when the backdrop switches to a selected backdrop.
- Broadcast message x
 - This code block will transmit a preset message globally throughout your project. Any sprite can insert a “When I receive x” block to begin running code upon transmission.
- When I receive x

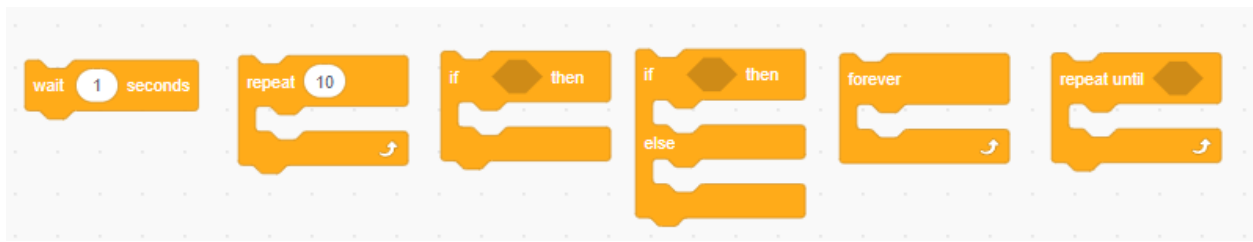
- This code block will begin running any code placed under it when a preset message is received, either from another sprite or another function.



Control (Orange)

This group of code blocks act as filters, allowing you to manage and create loops, sequences, and conditional statements to further develop your program. Many control blocks require you to use operators, which let you quickly manipulate variables, to form these conditionals.

- Wait x seconds
 - This block will pause your code and wait the inputted amount of time before running the next code block.
- Repeat x
 - This block will run any code inserted inside of it the amount of times specified.
- If x then
 - This block will check the value of an inserted variable, and, if it meets the user's defined condition, will run any code inserted inside of it.
- If x then, else
 - Similar to the above code block, this code block will instead run other code if the condition isn't met.
- Forever
 - This block will repeat the code inserted inside of it infinitely.
- Repeat until x
 - This block will repeat the code inserted inside of it until the user's defined condition is met.



Sensing (Light Blue)

This project does not utilize any sensing blocks.

Operators (Green)

Operators are a way to quickly manipulate variables. Operators are used by Control blocks to create conditions to check and verify before continuing with runtime execution. Operators can also be used to easily combine two variables together.

Variables (Burnt Orange)

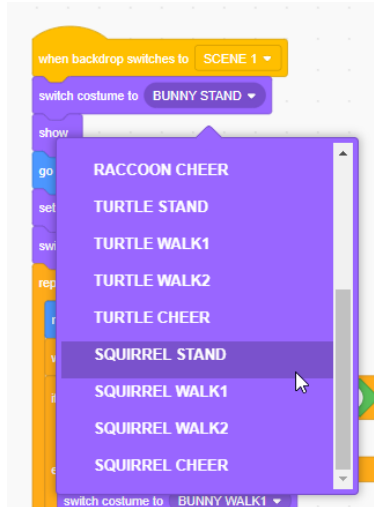
Variables are elements within Scratch that hold a value, like a word, a number, or true/false value. Variables can be limited to one sprite, or accessible by all.

Characters and Creativity

Scratch makes it easy to unleash your creativity. Kids will be able to alter this animation in a wide variety of ways to complete the curriculum requirements. To test your code, you can click on the green flag listed above your viewport. Remember, this will run the code from the beginning. You can click on a block of code in your canvas to test one specific part of your program.

Change Characters Entirely

The Main Actor and Side Actor are the primary sprites defined in the project. The woodland animals designed for the animation are costumes that are available for these two sprites. The characters available in the project are the Bunny, Turtle, Raccoon, and Squirrel. You can swap characters using the *Looks > Switch costume to x* block. If you, for example, wish to change the Main Actor into the Turtle, all existing *Looks > Switch costume to BUNNY STAND* blocks should be changed to *Looks > Switch costume to TURTLE STAND*. This is true for all other poses (WALK1, WALK2, and CHEER) too.

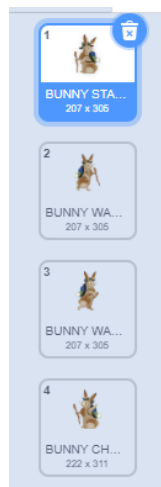


Please note that the Main Actor has all characters, with all poses, and the Side Actor has all characters with only the standing and cheering poses.

Make the Character Move

With the right sequence of code blocks, you can create the illusion of more advanced movement.

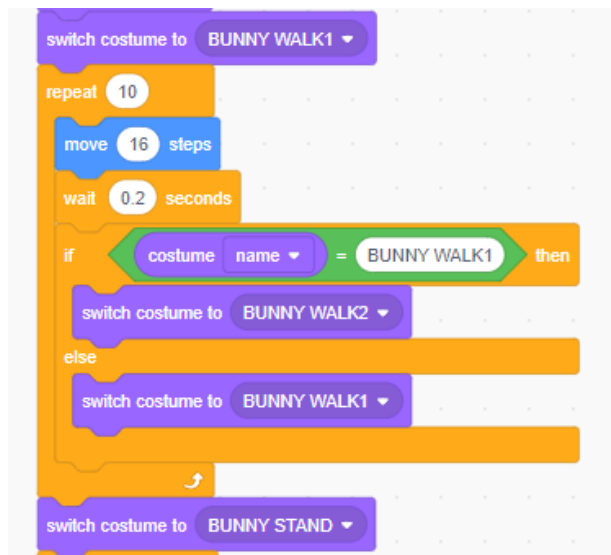
The Main Actor sprite has 16 costumes - 4 sets of different animals with each costume showing a different pose. One standing, two walking, and one cheering pose.



To create the illusion that our bunny is walking, we must move our bunny forward, while also swapping between our two walking costumes.

An easy walking animation can be created by first starting with a *Looks > Switch costume to x block*, with x being BUNNY WALK1. Next, add a *Control > Repeat x* block. Inside this block, start the sequence by moving our bunny a set amount on the X axis. Next, using a *Control > Wait x seconds* block, pause for a moment- 0.2 seconds will make this animation look believable.

Lastly, we'll need to swap between our two walking poses each time the repeat sequence runs. Using a *Operator > x = y* block, and a *Control > If x then, else* block, we can quickly check what costume is on display, and swap accordingly. At the end, the code should look something like this:

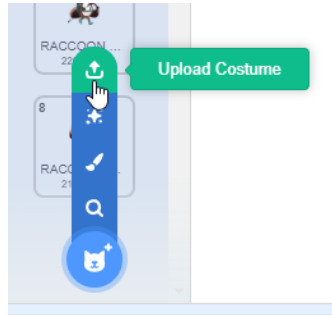


Encourage your students to experiment with different values or costumes. Alternatively, you could discover your own way to animate this walk cycle.

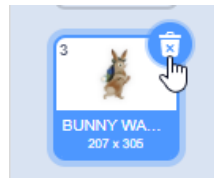
Editing Your Sprite & Adding/Removing Costumes

If you want to adjust the look of any sprite, or upload your own costumes, go to the “Costumes” menu.

At the bottom left of your window, click on the “Choose a Costume” button and then select “Upload Costume” - any general image file will work, but preferably, use a transparent .png or .svg file.

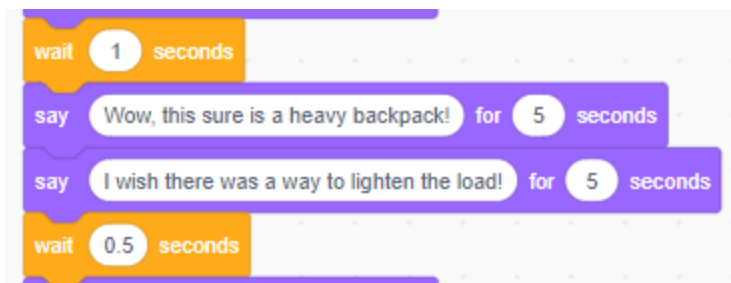


If you want to delete a specific costume from your sprite, simply click on the trash can icon on the top right of the costume.



Make the Character Speak

Dialogue can be edited easily, by simply typing in whatever you would like into the existing *Looks > Say x for x seconds* blocks. New dialogue can quickly be slotted in by adding more *Looks > Say x for x seconds* blocks.



It's important to note if you use a *Looks > Say x* block, the character will have a speech bubble appear until another similar block is triggered.

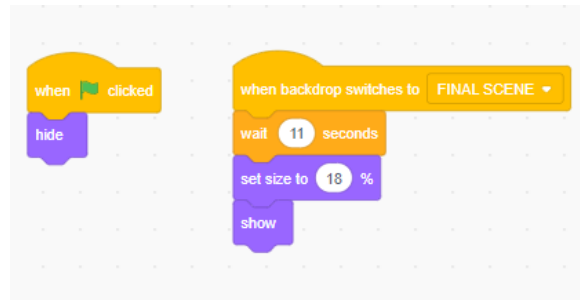
Add Props to the Scene

New sprites can be added at any time. To make them appear, start with an *Events > When backdrop switches to x* block, with x being the scene you'd like the sprite to appear on.

Add a *Looks > Show* block to this event, and position the sprite where you want it to appear. Set the position with a *Motion > Go to x and y* block. If you want the sprite to appear on the next scene, you can simply stop here, but if there's a scene you do not

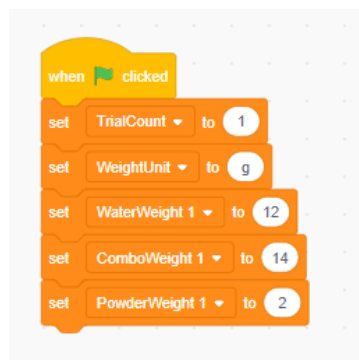
want the sprite to appear on, make sure to add another *Events > When backdrop switches to x* block, set the proper scene, and add a *Looks > Hide* block.

You can see this in action on the “Prop” sprite.



Changing the Weights Displayed on the Graph

To change the weight of the bottle of water, the powder, or the mixed drink, use a *Variables > Set x to y* block.



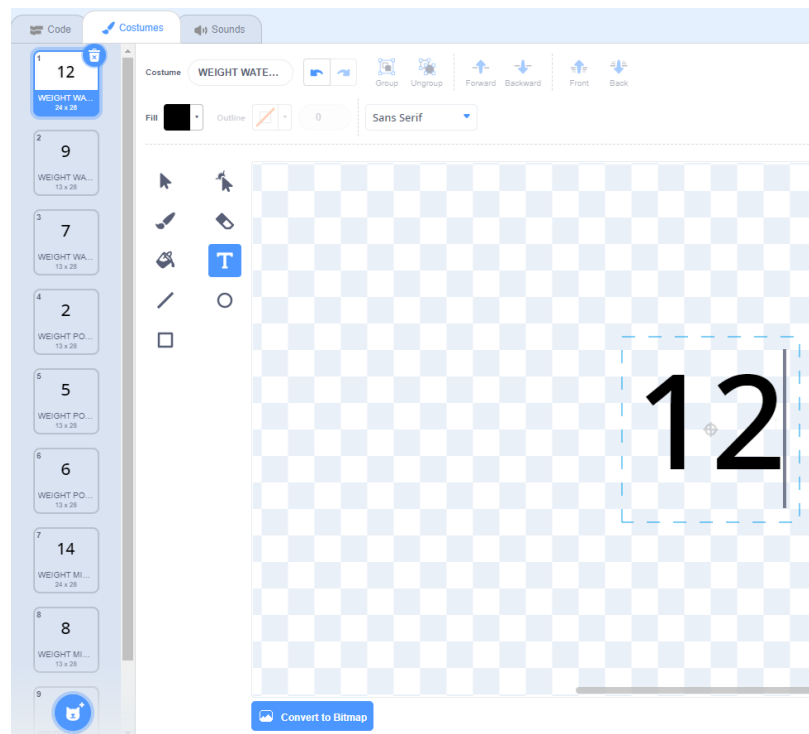
The Main Actor sprite already contains three *Variables > Set x to y* blocks that will set WaterWeight1, PowderWeight1, and ComboWeight1 to set weights. Students are able to enter any numerical value into these blocks to have the weights updated.

To change weights for other trials, you will need to add extra *Variables > Set x to y* blocks to the *Events > When green flag clicked* block, for WaterWeight2, WaterWeight3, ComboWeight2, ComboWeight3, PowderWeight2, and PowderWeight3. Make sure to also update the TrialCount variable to either 2 or 3 respectively.

Please note that any value under 0, and any value over 998 may result in the graph displayed on Scene 4 to become distorted and incorrect.

Changing the Weights Displayed on the Scale

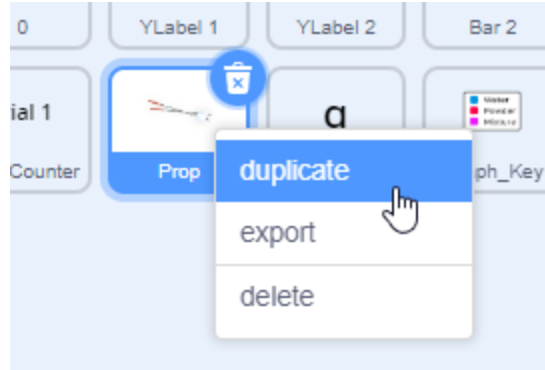
The weight values that appear on the scale are simple text blocks that can be edited within the “Costumes” menu when selecting the Scale Label sprite. While in the current setup the scale scene only plays once; students could code this scene to repeat other weights if they please. Be sure to center your text to the middle of the canvas using the arrow tool and the crosshair in the middle of the screen!



Due to the limitations of Scratch, this value cannot be automatically updated when students adjust their weight variables. **Remember to change these values separately!**

Extra Tips and Tricks

Sprites can be cloned- if, especially for the ending scene, a student wishes to have more than one prop shown on screen, they can quickly clone the Prop sprite by right clicking and pressing “Duplicate”

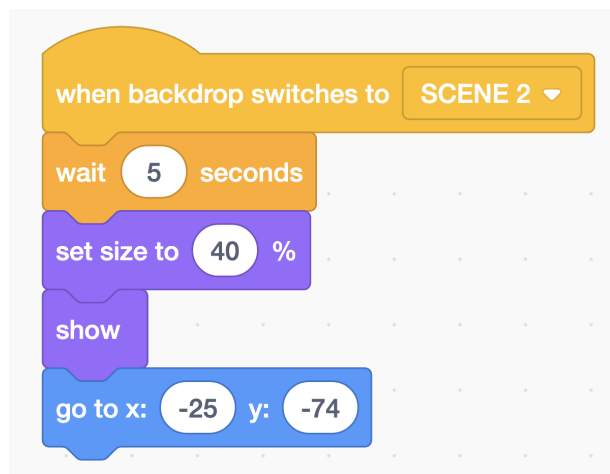


This will copy over the code and costumes available on the previous sprite.

Scratch allows you to copy and paste code blocks, so by using CTRL + C and CTRL + V, you can quickly use your animation elsewhere on the same sprite. This will let you experiment without losing your good progress.

Scratch allows sprites to be placed on top of one another, with a layering system. Unfortunately, there is no quick way to visually see what sprite is on what layer, but you can use *Looks > go to x layer* to quickly bring a sprite to the front of the screen on top of other sprites, or the back of the screen so that other sprites appear on top of it.

Did you add a character or prop to your scene in the code editor, but don't see it when you start your animation? You are likely missing the "Show" block. Every character or prop needs this block - otherwise it won't appear.



If you change the value of the scale label, and find that it's overlapping the weight unit symbol or isn't positioned correctly, you can go to the Costumes menu, and using the cursor tool, move the text to the left or right to correct it.

Project Links and Resources

Drink Mix Animation Scratch Project:

A prebuilt project showcasing what the animation could look like:

- <https://scratch.mit.edu/projects/787794297/>

Drink Mix Animation Student Project:

This paired-down version of the example project will be your students' starting point:

- <https://scratch.mit.edu/projects/787796199/>

Scratch Tutorial:

A simple tutorial and introduction to Scratch can be found here:

- <https://scratch.mit.edu/projects/editor/?tutorial=getStarted>

Scratch Wiki:

A community forum with insight into Scratch's coding language, discussions/idea sharing, etc:

- <https://en.scratch-wiki.info/>