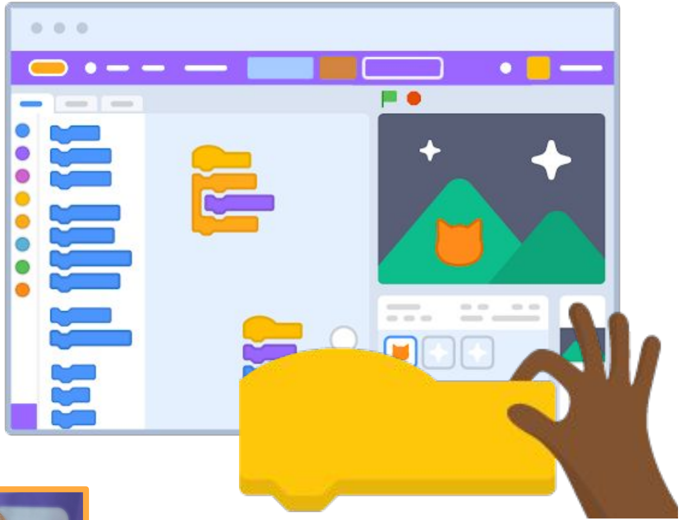
LEVEL UP WITH SCRATCH WORKSHOP SERIES

Super-Charge Your Game Design

SCRATCH™

Session Overview

- Brief Introduction to Creative Learning with Scratch
- What Is the Plan? Preparing to Create.
- Blank Page Blues? Start with a Remix!
- Cartesian Coordinates and Unique Uses of Coordinates
- Level Up Pong
- Adding Complexity
- Level Up Mazes and Platformers
- Gravity
- Clones
- Use Operators
- Animated Title
- Text Generator
- Use Hardware to Control
- Control a Sprite with Your Face!
- Design Considerations
- Add Emotion and Excitement with Pitch
- Health Bars and Timers
- Advanced Scrolling Background: User-Controlled
- Wrap Up - Debugging and Reflection



Facilitator: Maren Vernon

Scratch Learning Resource Designer
@algorithmar and @scratchlycaterton

SCRATCH™
FOUNDATION

Learning Goals

- Remix our starter projects to add personalized touches and additional elements and greater complexity (both for the programmer and the player)
- Discuss design considerations, like what inputs to use (keyboard keys, mouse, your face...) and how to make projects accessible to more players
- Consider how to add emotion, excitement, and urgency to games using sound, animated text, health meters, or timers
- Reflect on finalized projects and the creative process with peers
- Communicate and share projects with your learning community and the greater Scratch online community

Getting Started

Click “Create” or log in to your free account to save projects.

go to: scratch.mit.edu

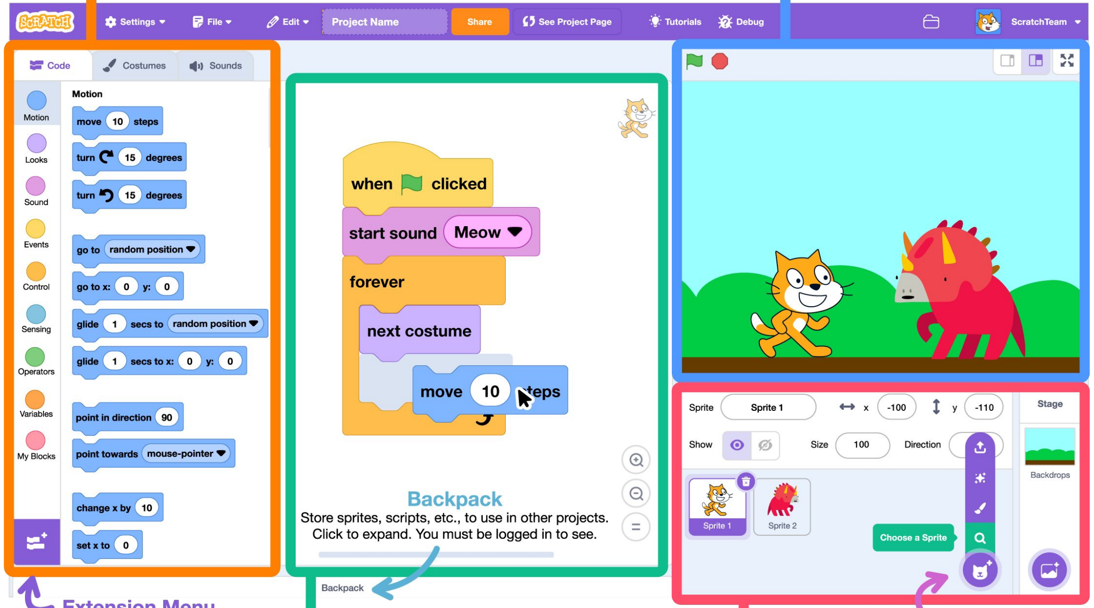
Set your language and block color mode.

Choose a sprite. Drag and drop code blocks to create a script.

scratchfoundation.org/learn/learning-library/getting-started

Block Palette
Blocks for coding your projects.

The Stage
Where your creations come to life.



Backpack
Store sprites, scripts, etc., to use in other projects. Click to expand. You must be logged in to see.

Extension Menu
Additional blocks available.

Sprite Menu
Expand for options.

Coding Area/Script Area
Drag in blocks and snap them together.

Sprite Area
Click the thumbnail of a sprite to select it.

SCRATCH™ FOUNDATION

Creative Learning

As facilitators, we want to support **playful learning and tinkering mindset values** so that participants can:

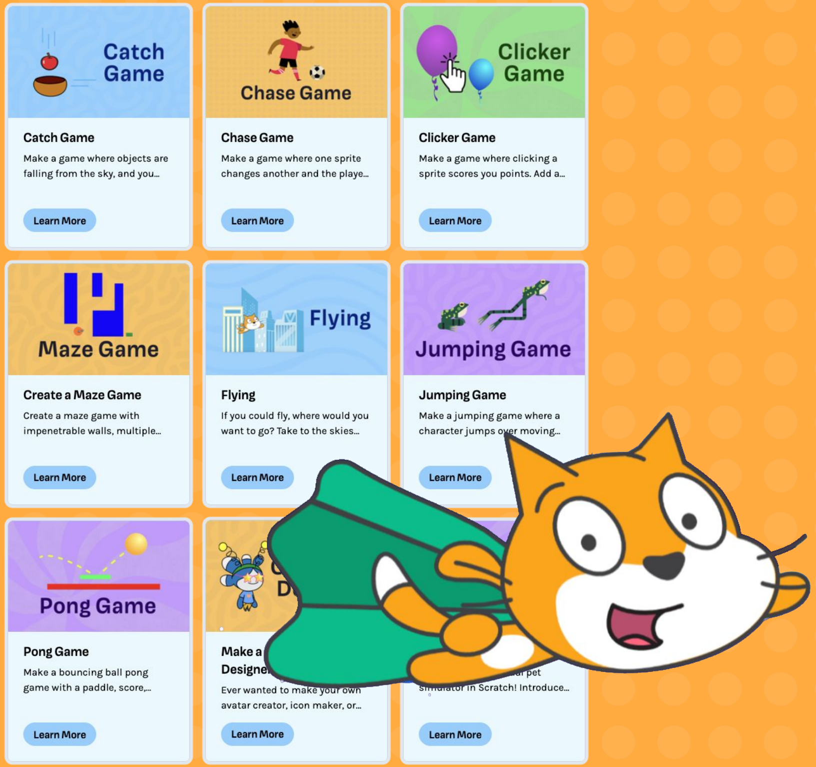
- Engage playfully in **projects** that are meaningful to them and elicit joy
- Collaborate with **peers** to experiment, share, and celebrate ideas
- Develop a mindset that is **comfortable with the discomfort** of getting stuck
- Develop a mindset that thinks critically about **strategies for getting unstuck**

scratchfoundation.org/learn/learning-library/scratch-creative-learning-philosophy

Scratch's Creative Learning Philosophy

Scratch's Creative Learning Philosophy

Scratch pioneered block-based programming, enabling young...



Catch Game

Make a game where objects are falling from the sky, and you...

Learn More

Chase Game

Make a game where one sprite changes another and the playe...

Learn More

Clicker Game

Make a game where clicking a sprite scores you points. Add a...

Learn More

Create a Maze Game

Create a maze game with impenetrable walls, multiple...

Learn More

Flying

If you could fly, where would you want to go? Take to the skies...

Learn More

Jumping Game

Make a jumping game where a character jumps over moving...

Learn More

Pong Game

Make a bouncing ball pong game with a paddle, score,...

Learn More

Make a
Designe

Ever wanted to make your own avatar creator, icon maker, or...

Learn More

ar pet
simulator in Scratch! Introduce...

Learn More

Let's Imagine…

What will you create?

What Is the Plan? Preparing to Create.

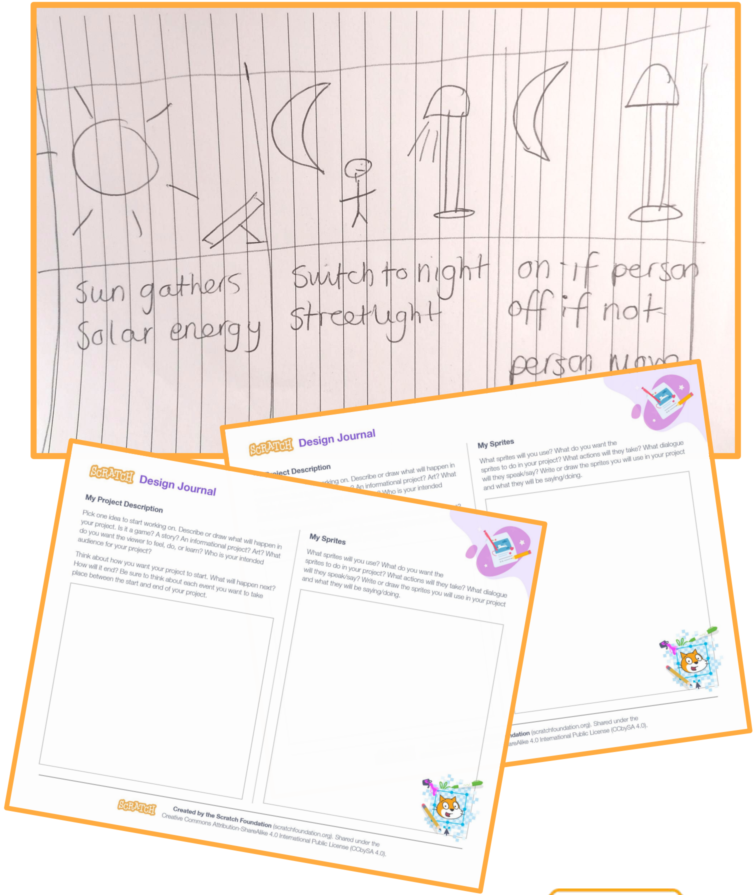
Some learners may want to jump right into creating their projects and enjoy the spontaneity of adjusting as they build. Others may feel most comfortable developing a comprehensive plan first.

Use the Scratch Design Journal to record brainstorming and sketches as they imagine, plan, and iterate throughout their project's development.

How many sprites will you need?

What is the goal of the project/object of the game?

All Blocks posters are available to remind students of block options while working offline or away from their device.


Sun gathers Solar energy
Switch to night streetlight
on if person off if not person move

SCRATCH Design Journal

My Project Description

Pick one idea to start working on. Describe or draw what will happen in your project. Is it a game? A story? An informational project? Art? What do you want the viewer to feel, do, or learn? Who is your intended audience for your project?

Think about how you want your project to start. What will happen next? How will it end? Be sure to think about each event you want to take place between the start and end of your project.

My Sprites

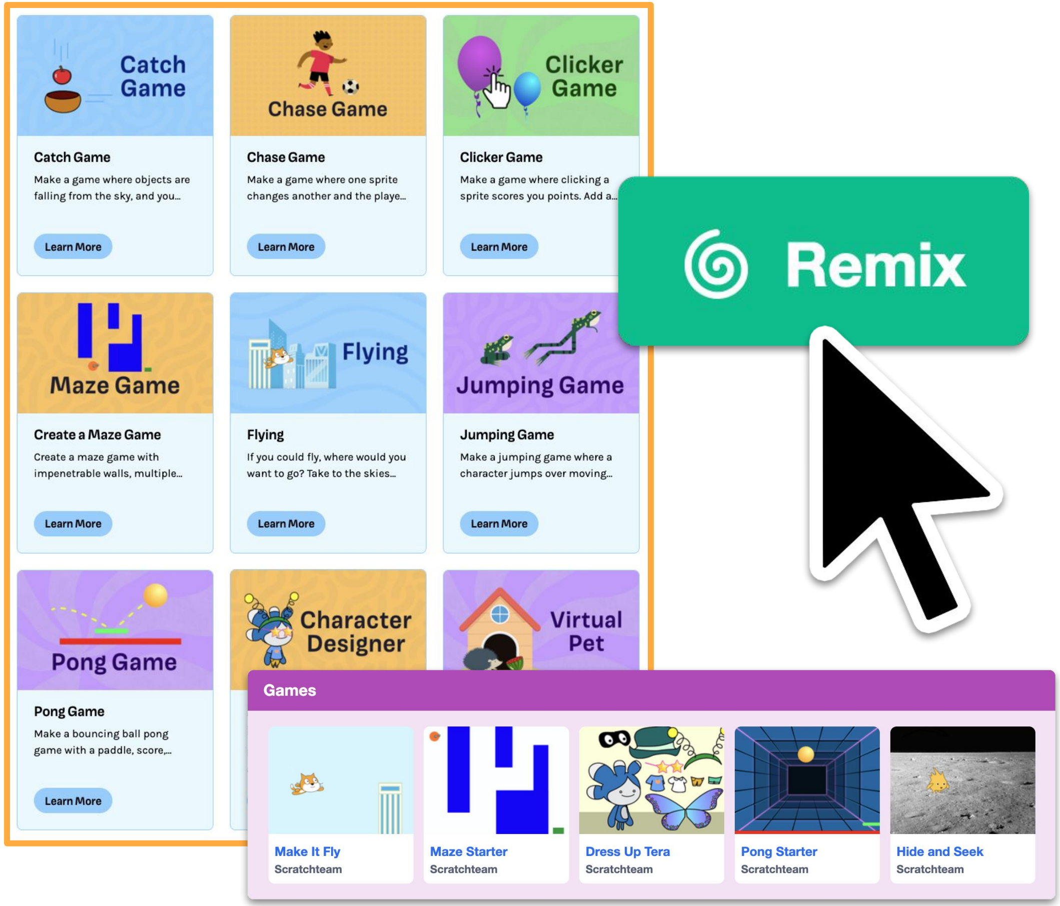
What sprites will you use? What do you want the sprites to do in your project? What actions will they take? What dialogue will they speak/say? Write or draw the sprites you will use in your project and what they will be saying/doing.

Created by the Scratch Foundation (scratchfoundation.org). Shared under the Creative Commons Attribution-ShareAlike 4.0 International Public License (CCbySA 4.0).

Blank Page Blues?
Start with a Remix!

Starting from a blank slate can be intimidating for some learners. It might be helpful to utilize an Interactive Tutorial or explore and remix a Starter Project.

Filtering by "Games" as the topic in our Learning Library brings up many resources related to elements of game development and game tutorials to explore.



Catch Game

Make a game where objects are falling from the sky, and you...

Learn More

Chase Game

Make a game where one sprite changes another and the playe...

Learn More

Clicker Game

Make a game where clicking a sprite scores you points. Add a...

Learn More

Create a Maze Game

Create a maze game with impenetrable walls, multiple...

Learn More

Flying

If you could fly, where would you want to go? Take to the skies...

Learn More

Jumping Game

Make a jumping game where a character jumps over moving...

Learn More

Pong Game

Make a bouncing ball pong game with a paddle, score,...

Learn More

Games

Make It Fly	Maze Starter	Dress Up Tera	Pong Starter	Hide and Seek
Scratchteam	Scratchteam	Scratchteam	Scratchteam	Scratchteam

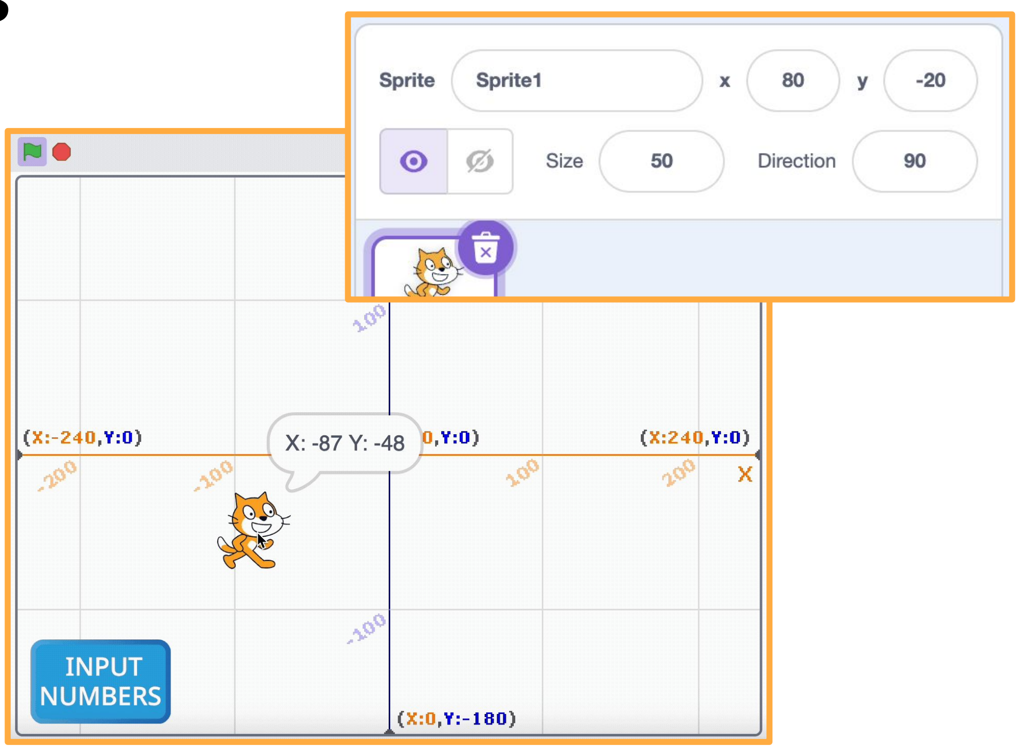
Achievery Unit

Cartesian Coordinates

Often, one important element of a game is knowing and controlling where objects are on the Scratch stage.

Check out our Achievery Units "Cartesian Coordinates" and "Coordinate Reporter," and experiment with our "X and Y Coordinates" starter project.

Which blocks control the sprite's position, and what are the differences between them? Experiment!



Sprite: Sprite1 x: 80 y: -20

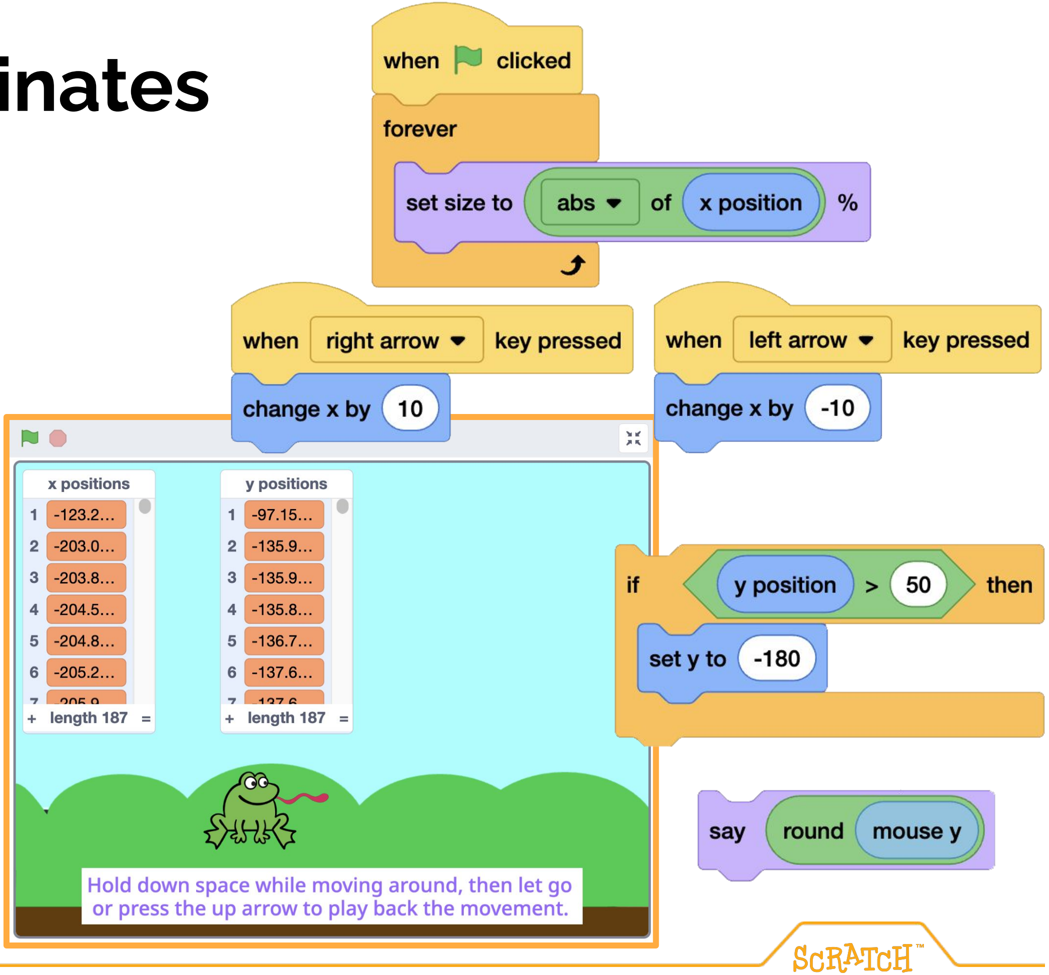
Size: 50 Direction: 90

Unique Uses of Coordinates

Coordinates can be utilized in unique ways in projects. For example:

- Record x and y positions as you move a sprite around, then play it back as an animation. Think of it like CGI motion capture!
- Use the position of a sprite or the mouse on the stage to change a sprite's size, a sound's pitch, and more.

Example projects: 962767941 and 1226183538, 1111534003, and 1228985642



```
when 🏁 clicked
forever
    set size to (abs of (x position)) %
↺
```

```
when [right arrow] key pressed
change x by (10)
```

```
when [left arrow] key pressed
change x by (-10)
```

x positions	
1	-123.2...
2	-203.0...
3	-203.8...
4	-204.5...
5	-204.8...
6	-205.2...
7	-205.9
+ length 187 =	

y positions	
1	-97.15...
2	-135.9...
3	-135.9...
4	-135.8...
5	-136.7...
6	-137.6...
7	-137.6
+ length 187 =	

Hold down space while moving around, then let go or press the up arrow to play back the movement.

```
if <(y position) > (50)> then
    set y to (-180)
```

```
say (round (mouse y))
```

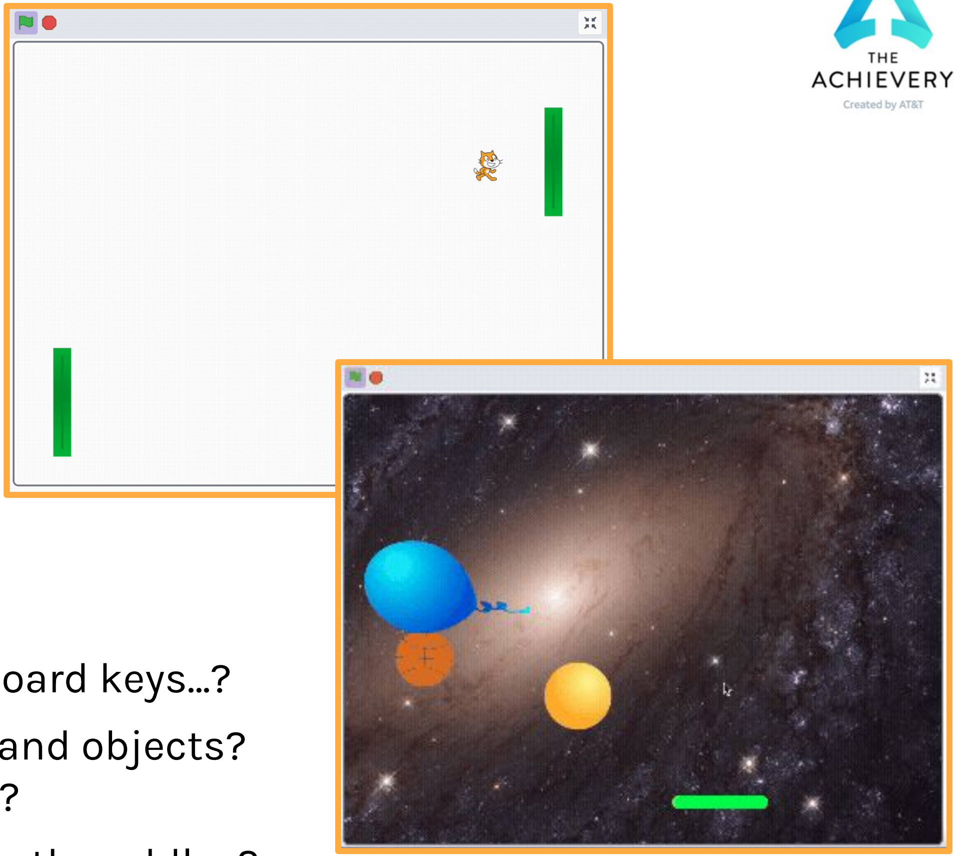
Achievery Unit

Level Up Pong

Remix our starter project “Pong Game,” or use our Achievery Units “Create a Pong Game” or “Coding a Ball” to create a moving/bouncing object you interact with using a paddle.

How can you create a unique remix? Examples might include having two paddles or multiple objects to bounce.

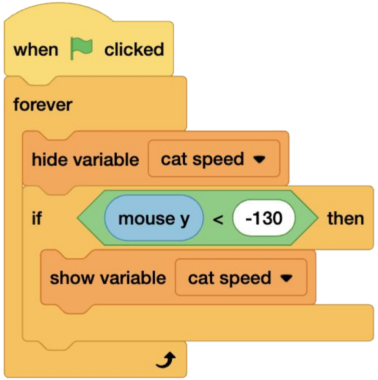
- How to control the paddle: mouse, keyboard keys…?
- How to control the speed of the paddle and objects? Do you want to introduce modes/levels?
- What if a single movement controlled both paddles?



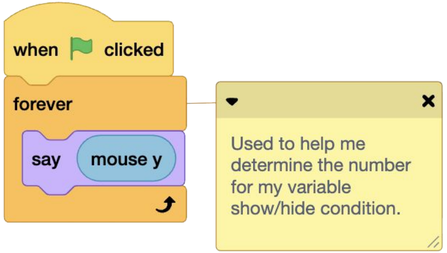
Adding Complexity

Expanding on the idea of one paddle controlling another offers an example of cross-curricular connections and using more advanced blocks in your code: absolute values in math.

In this example project, “Opposite Paddle Cat Pong,” I’ve used operators to help me reverse mirror a paddle’s behavior, show and hide stage monitors that give the user control over the difficulty level, and commented my code (an important part of your practice; teaching others reinforces for yourself).

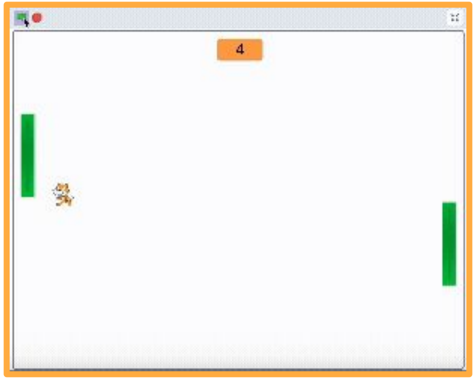


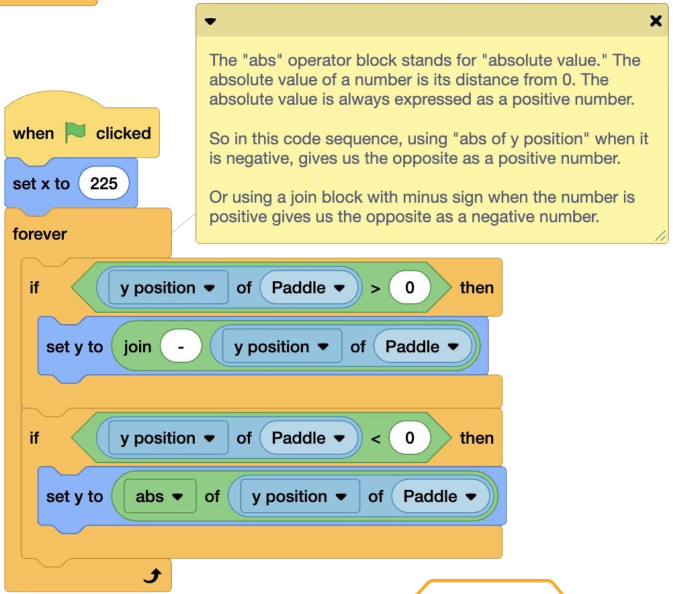
```
when 🏁 clicked
forever
    hide variable [cat speed ▾]
    if <(mouse y) < (-130)> then
        show variable [cat speed ▾]
    end
end
```



```
when 🏁 clicked
forever
    say (mouse y)
end
```

Used to help me determine the number for my variable show/hide condition.





```
when 🏁 clicked
set x to (225)
forever
    if <([y position ▾] of [Paddle ▾]) > (0)> then
        set y to (join (-) ([y position ▾] of [Paddle ▾]))
    end
    if <([y position ▾] of [Paddle ▾]) < (0)> then
        set y to ([abs ▾] of ([y position ▾] of [Paddle ▾]))
    end
end
```

The "abs" operator block stands for "absolute value." The absolute value of a number is its distance from 0. The absolute value is always expressed as a positive number.

So in this code sequence, using "abs of y position" when it is negative, gives us the opposite as a positive number.

Or using a join block with minus sign when the number is positive gives us the opposite as a negative number.

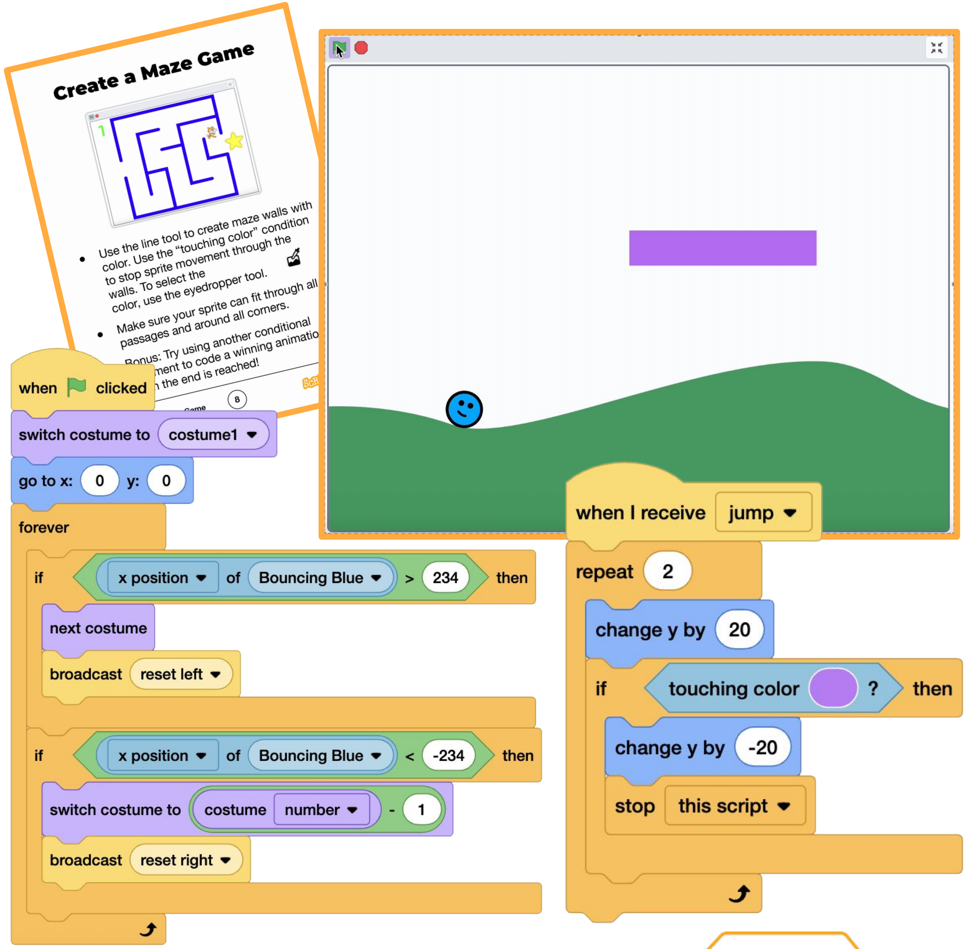
Starter Project

Level Up Mazes and Platformers

Many students first encounter conditional statement blocks when making a game like our “Maze Starter” starter project.

Touching color and reversing movement gets more complex in platformer games with raised platforms the sprite shouldn’t pass through, like in this “Platformer Starter.” Or use conditional statements to cycle through costumes not just forward (“next costume”) but backward as well.

More on Conditional Statements



Create a Maze Game

- Use the line tool to create maze walls with color. Use the “touching color” condition to stop sprite movement through the walls. To select the color, use the eyedropper tool.
- Make sure your sprite can fit through all passages and around all corners.

Bonus: Try using another conditional
ment to code a winning animatio
n the end is reached!

Game 8

```
when [flag] clicked
switch costume to (costume1 ▾)
go to x: (0) y: (0)
forever
    if <(x position ▾) of (Bouncing Blue ▾) > (234)> then
        next costume
        broadcast (reset left ▾)
    if <(x position ▾) of (Bouncing Blue ▾) < (-234)> then
        switch costume to ((costume (number ▾)) - (1))
        broadcast (reset right ▾)
```

```
when I receive (jump ▾)
repeat (2)
    change y by (20)
    if <touching color ( ) ?> then
        change y by (-20)
        stop (this script ▾)
```

Starter Project

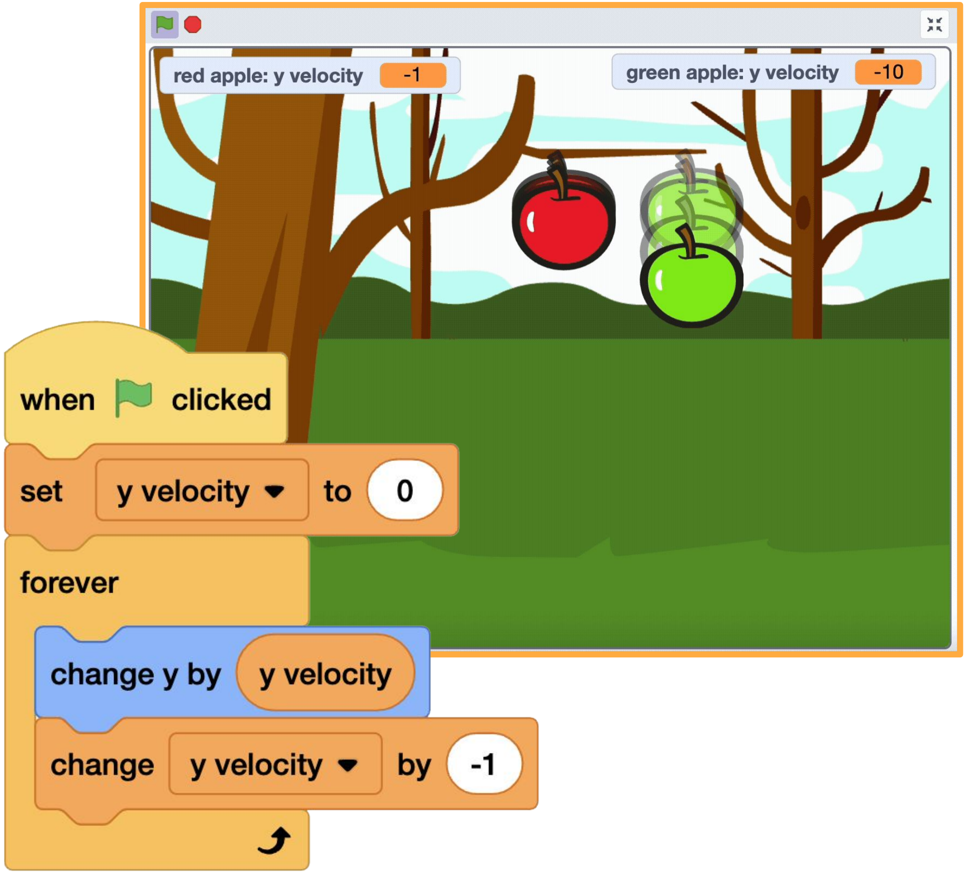
Gravity

Check out our starter project “Gravity Example” and our Gravity resources, including student-facing coding cards.

With gravity, the object gains speed as it falls down the stage (instead of falling at a consistent rate of speed), making for a more realistic feel.

See the example project “Gravity Plus Forward Movement” for an example of making a more realistic bouncing ball.

Some example projects: 1215320440, 964424785, 1145141063, 963989317



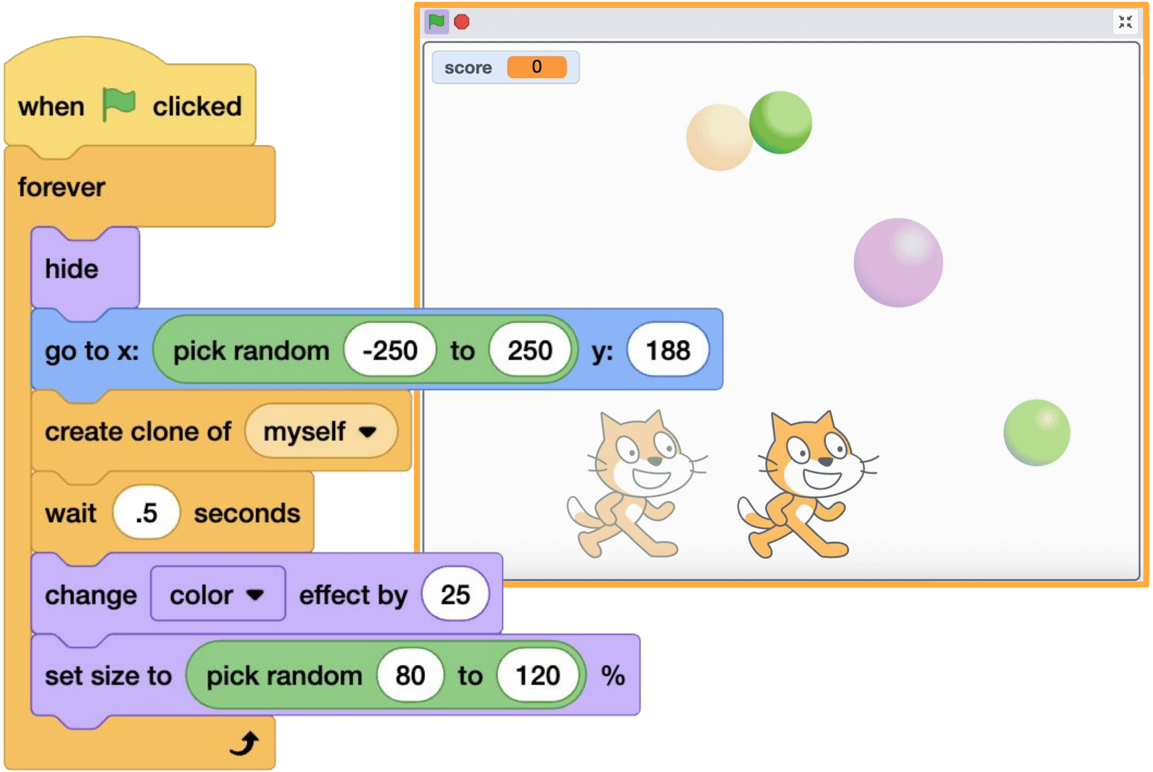
```
when 🏁 clicked
set y velocity ▾ to 0
forever
    change y by (y velocity)
    change y velocity ▾ by -1
↺
```

Clones

See our resources on Clones, which let you create multiple copies of your sprite while your project is running.

When each clone is produced, it has the same costumes, sounds, scripts, and variables as the original, but it is otherwise independent.

Cloning can speed up your game creation process by generating multiple obstacles or objects to collect.



```
when 🏳 clicked
forever
    hide
    go to x: (pick random (-250) to (250)) y: (188)
    create clone of (myself ▾)
    wait (.5) seconds
    change (color ▾) effect by (25)
    set size to (pick random (80) to (120)) %
↺
```

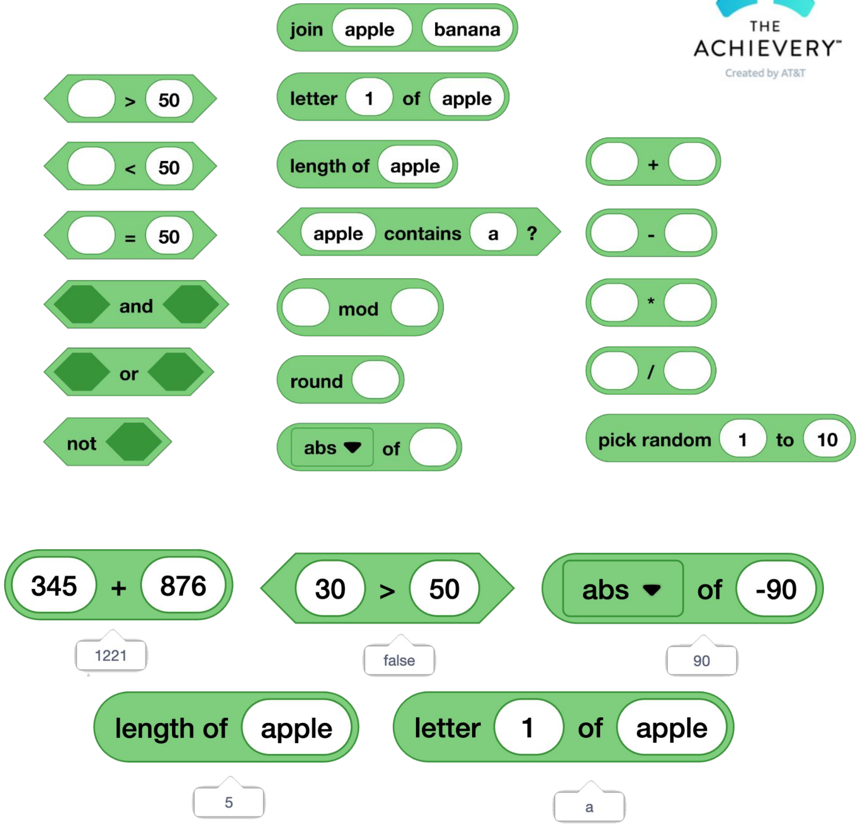
Example project: 1010672973

Achievery Unit

Use Operators

Arithmetic operators in Scratch are mathematical tools that perform calculations, report back if a mathematical statement is true or false, or perform functions on numbers to create unpredictable values.

See our Achievery Unit "Arithmetic Operators" to learn more. Did you know that you can quickly see equation results, by simply adding an arithmetic operator block to your script area, typing in the numbers, and clicking on the block? Blocks like greater than or less than are an example of a Boolean block, which reports if something is true or false.



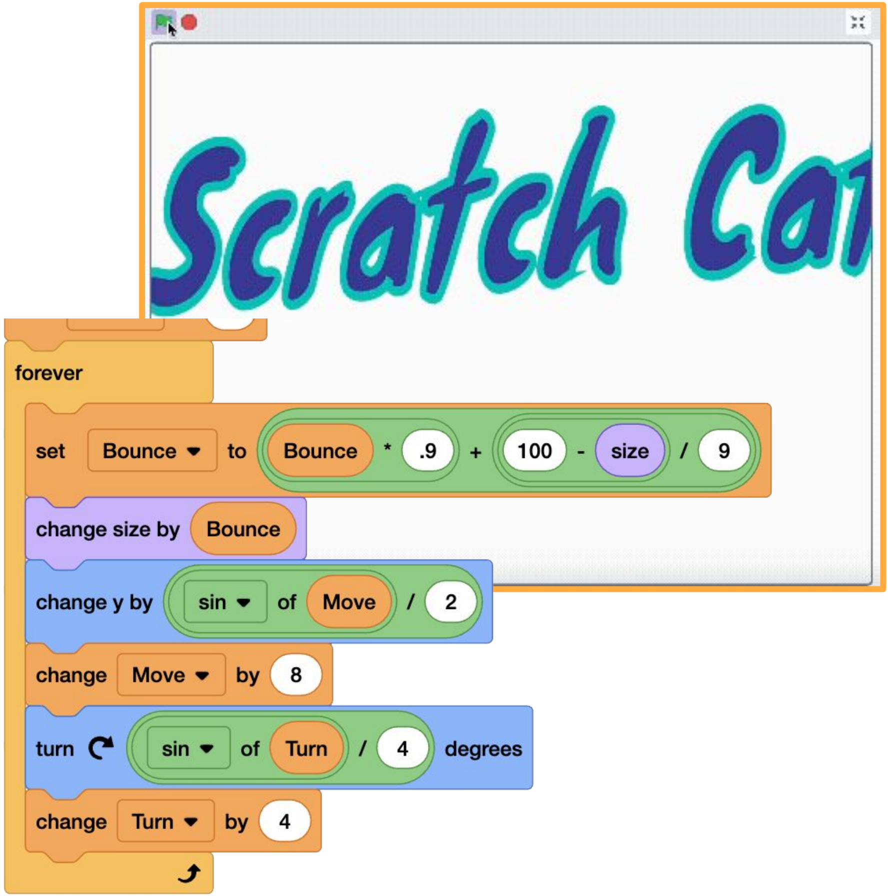
Achievery Unit

Animated Title

Text in an animation or game can be very important to communicate information and guide players through the experience: display instructions, show scores or levels. tell a story through visible character dialogue, or use visually exciting text or animated text to create different moods.

Check out our Achievery Units "Create a Text Sprite," "Revealing a Sprite," and "The Bounce Algorithm."

Example project: 865927349



```
forever
    set Bounce to ((Bounce * .9) + ((100 - size) / 9))
    change size by Bounce
    change y by ((sin of Move) / 2)
    change Move by 8
    turn ↻ ((sin of Turn) / 4) degrees
    change Turn by 4
```

THE ACHIEVERY™
Created by AT&T

SCRATCH™
FOUNDATION

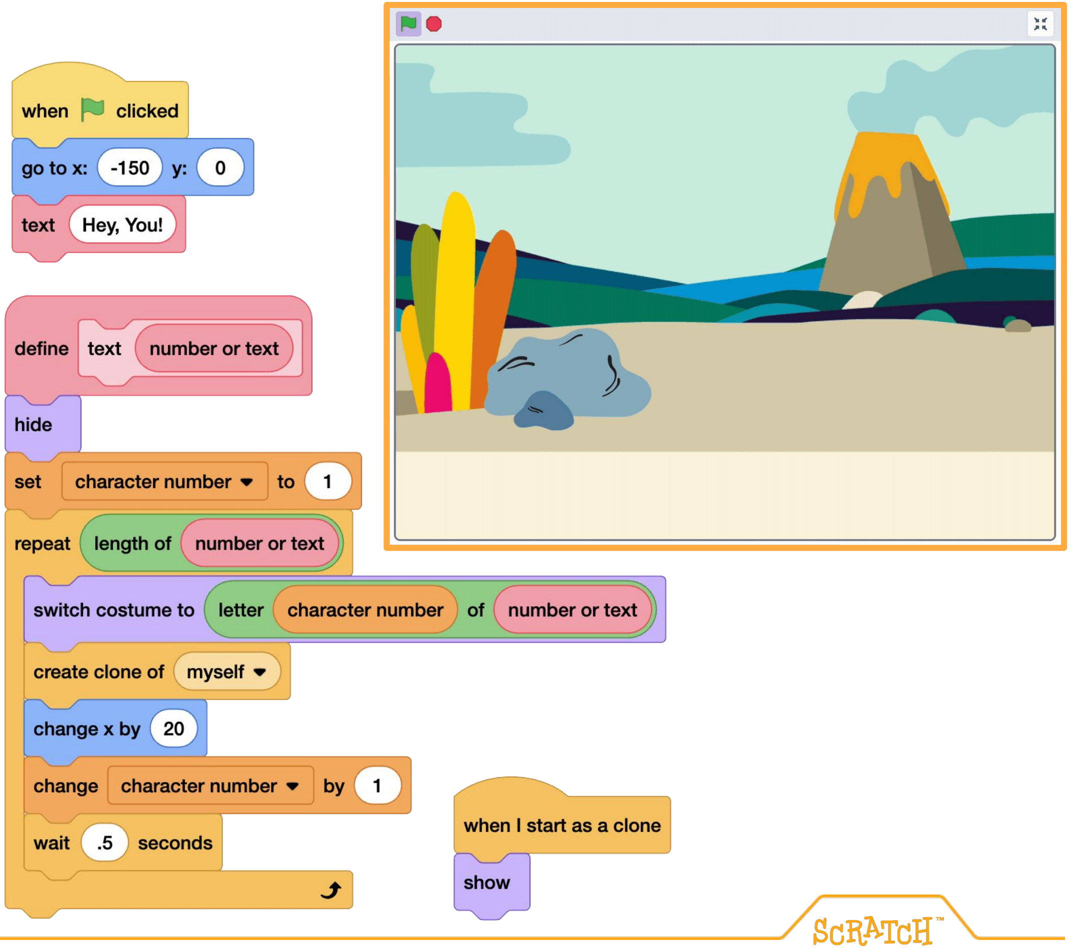
Text Generator

Use text rendering/a text generator to print text onto the stage (outside of using static text on costumes, or “say” or “think” blocks). Imagine closed captioning on your Scratch project.

See our Text Rendering/Text Generator resources, including student-facing coding cards.

Learn techniques, like how to iterate through a text string or use custom My Blocks for efficiency.

Some example projects: 1138479378, 985124543

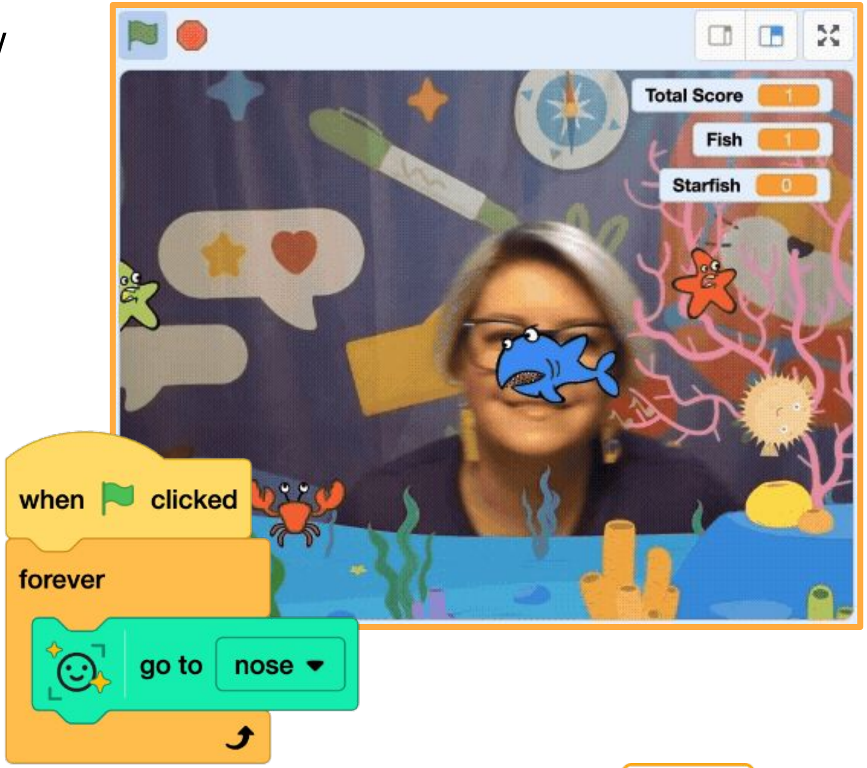


```
when 🏴 clicked
go to x: (-150) y: (0)
text [Hey, You!]

define text (number or text)
hide
set [character number ▾] to (1)
repeat (length of (number or text))
    switch costume to (letter (character number) of (number or text))
    create clone of [myself ▾]
    change x by (20)
    change [character number ▾] by (1)
    wait (.5) seconds
end

when I start as a clone
show
```

Control a Sprite with Your Face!

Want a challenge that gets your body moving? Try using our Face Sensing blocks and the features of your face to control the player sprite!

These AI-powered blocks use a machine learning model to detect if they see a face and where a nose, eyes, ears, mouth, etc., are. When you use Face Sensing blocks, **only your computer can sense your face. None of your data is stored or sent to Scratch** or any other site, making it a safe, fun, and creative way to explore the possibilities of AI.



```
when ⚑ clicked
forever
    go to nose ▾
```

More on Face Sensing, including coding cards.
Some example projects: 1210061611, 1217332929, 1216390004, 1216389964, 1216390066 and studio: 50854499

Use Hardware to Control

Any game using keyboard keys or a mouse click can be operated with a **Makey Makey**. Larger targets to press or click (like large foil-covered cardboard keys) also add to accessibility.

Have the program check the **micro:bit**’s tilt angle and move the sprite, or close the circuit of a micro:bit pin and GND to trigger an action.

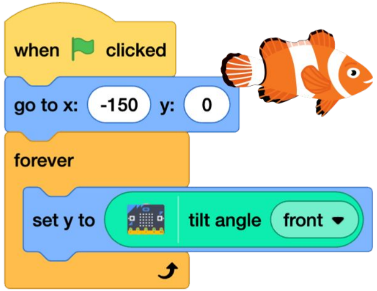
Spin and push the **Vernier Go Force sensor** to steer a ship.

More on micro:bit, Vernier Go Direct, and Makey Makey

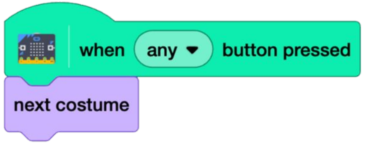
Some example projects: 239075973, 1159171779, 301385331

Maren’s Makey Makey controller

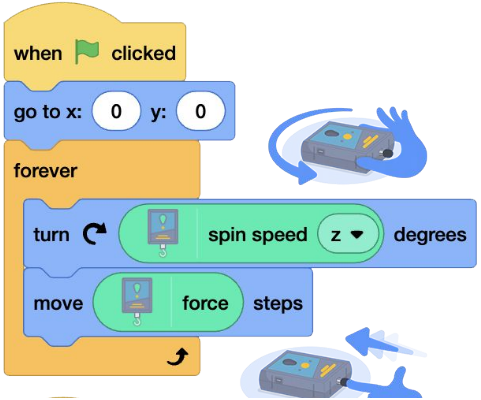
Maren’s micro:bit holder



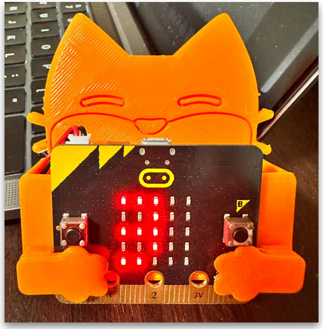
```
when ⚑ clicked
go to x: -150 y: 0
forever
    set y to tilt angle front ▾
```

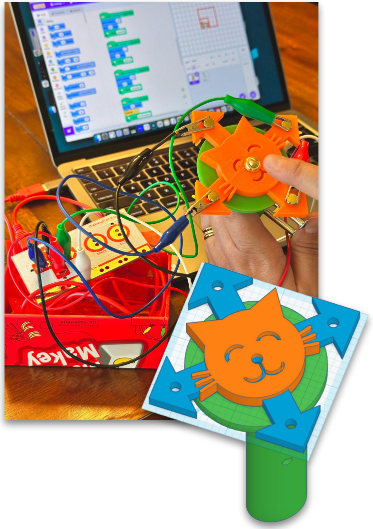


```
when any ▾ button pressed
next costume
```



```
when ⚑ clicked
go to x: 0 y: 0
forever
    turn ↻ spin speed z ▾ degrees
    move force steps
```



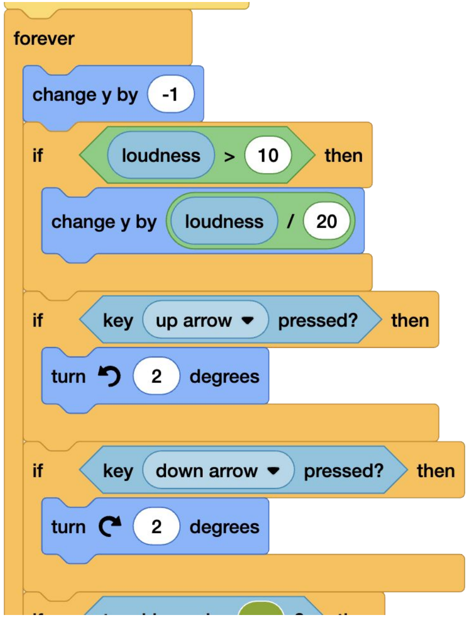


Design Considerations

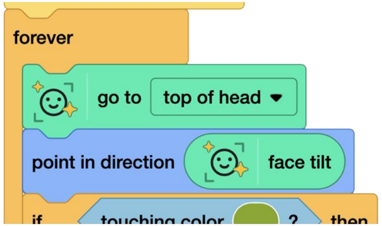
Many projects could work using a variety of inputs, like:

- keyboard
- mouse
- sound/loudness
- external button (like a Makey Makey or micro: bit)
- Face Sensing blocks
- video motion

Reflect on which inputs to choose for a project. What are the pros and cons? Who is your audience/who are you designing for? Are there benefits to having multiple input options?



```
forever
  change y by -1
  if loudness > 10 then
    change y by loudness / 20
  if key up arrow ▾ pressed? then
    turn ↺ 2 degrees
  if key down arrow ▾ pressed? then
    turn ↻ 2 degrees
```



```
forever
  go to top of head ▾
  point in direction face tilt
  if touching color ? then
```

Example projects:
Fishy Fishy 217332929 vs 554771225
Vietnamese Boat Float 1216389964 vs 1060227699
Vietnamese Food Quiz 1216390004 vs 1060254080
Vietnamese Drum Bounce 1216390066 vs 1060254615

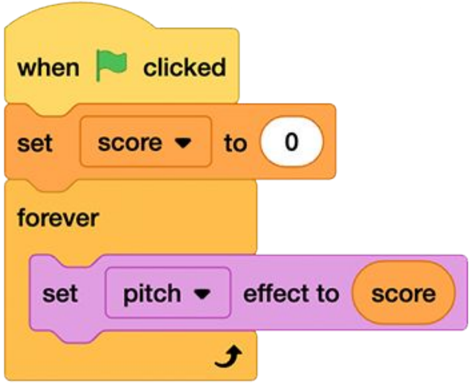
Starter Project

Add Emotion and Excitement with Pitch

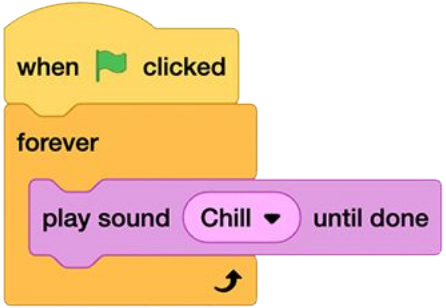
Check out our starter project "Catch the Fish, Increase the Pitch." Explore and remix this project.

Notice as you click on all 30 fish that the music gets higher in pitch creating a feeling of urgency.

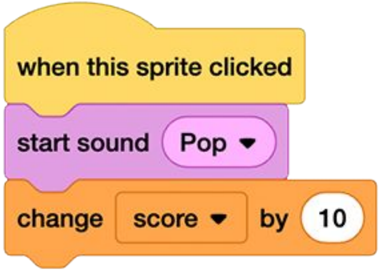
How can sound add an emotional component to a project?



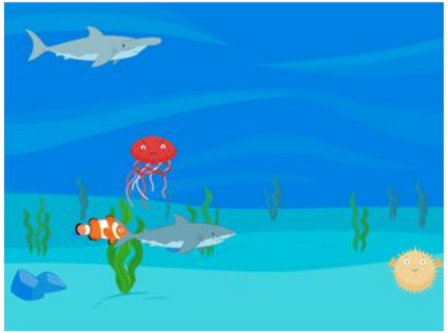
```
when 🏁 clicked
set score ▾ to 0
forever
    set pitch ▾ effect to score
```



```
when 🏁 clicked
forever
    play sound Chill ▾ until done
```



```
when this sprite clicked
start sound Pop ▾
change score ▾ by 10
```



An example project: 1216952432

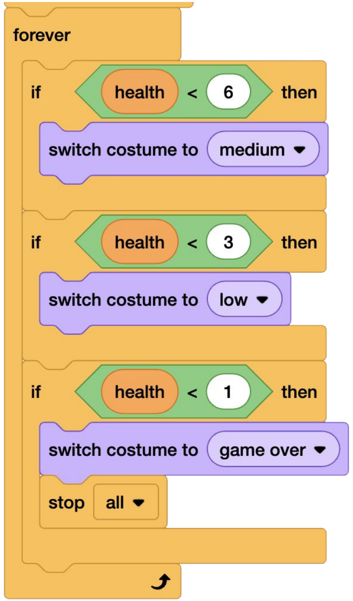
Achievery Unit

Health Bars and Timers

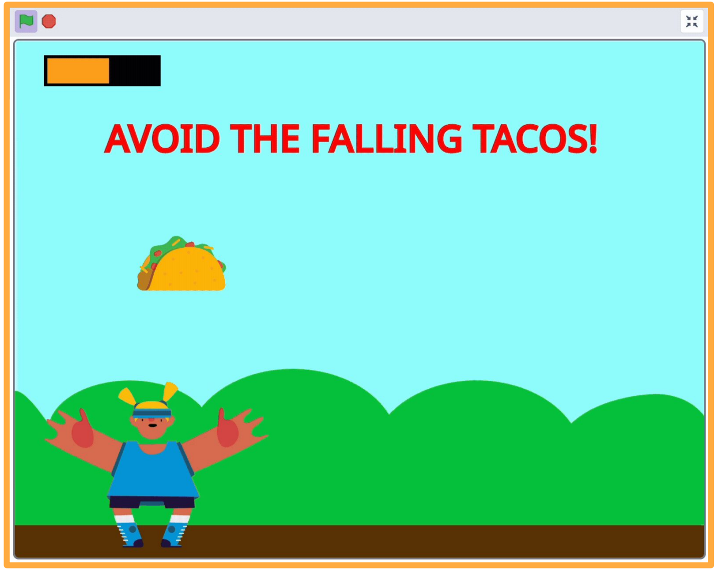
Looking for other ways to add a sense of urgency and excitement to your project? Try adding a health bar and limit the number of “lives” your player has. Or try using a timer.

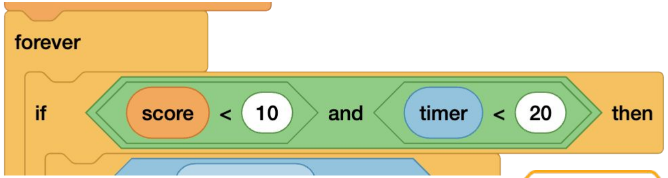
Check out our Achievery Units “Catch Game Design” and “Player Health Bar.”

Some example projects: 868464368 and 1010672973



```
forever
  if health < 6 then
    switch costume to medium ▾
  if health < 3 then
    switch costume to low ▾
  if health < 1 then
    switch costume to game over ▾
    stop all ▾
```



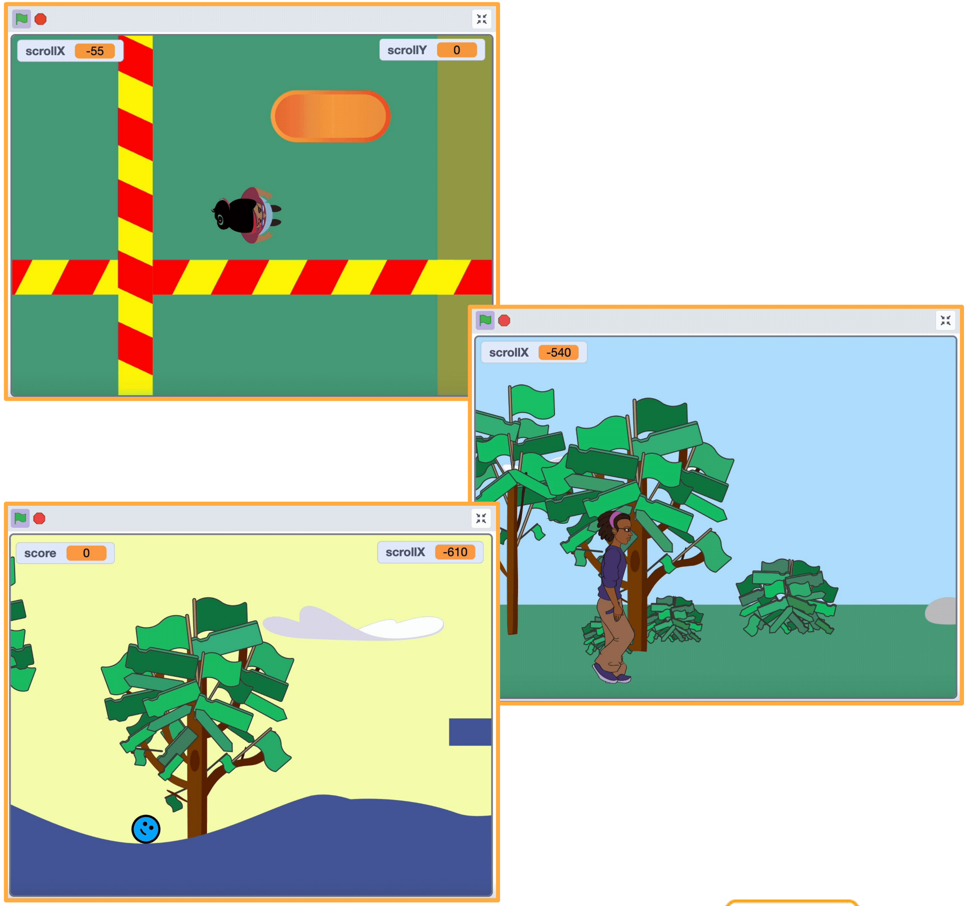


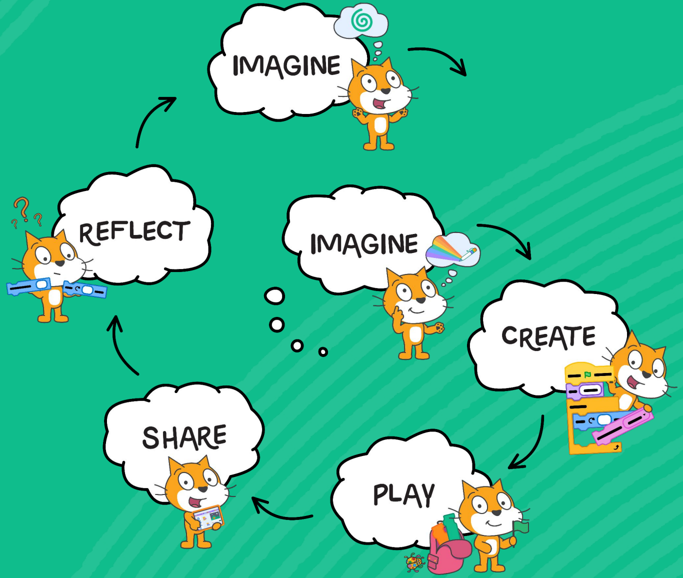
```
forever
  if score < 10 and timer < 20 then
```

Advanced Scrolling Background: User-Controlled

You can create a more advanced version of a scrolling background, where the user controls the background movement and position via keyboard keys to explore the scene!

See our Video Tutorial | Advanced Topics: User-Controlled Scrolling Background and Coding Cards.





Debug, Share, and Reflect

Continue Along the Creative Learning Spiral

Debugging

Debugging strategies to suggest include:

- Read Aloud/Explain the Code Step-By-Step
- Break Long Sequences Apart
- Add Temporary Waits to Slow Action
- Tinker with the Block Order
- Is There a Similar but Different Block Option?
- Check the Values/Inputs

See our Debugging resources for more

Prompts to Try

- "Ask Three Before Me," ask three peers before asking a facilitator.
- I don't know, but let's see if anyone else in the room might know/find out together.
- Which category do you think would be helpful?
- Can you say more about that?
- Let's test it out. What do you observe?
- Walk me through your code. What does it say?

Pair Programming

Have groups of different experience levels? Try pair programming! One person serves as "**driver**" (creating scripts), while the other is a "**navigator**" (reviewing, advising, etc.) and roles are switched frequently.

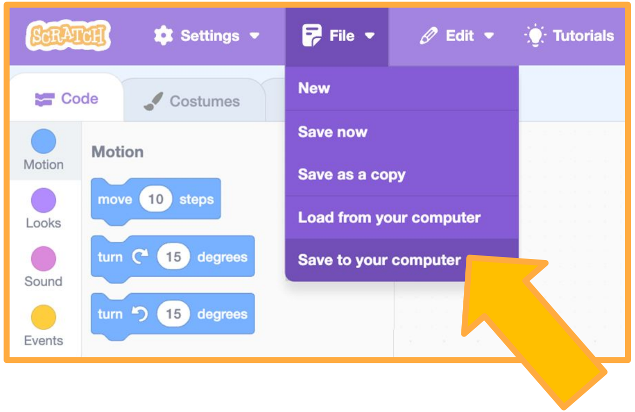
For a game project, pairing up users who are interested in drawing or storytelling with users who have more experience coding, for example, can create rich and dynamic projects in addition to opportunities for them to teach and learn from each other.

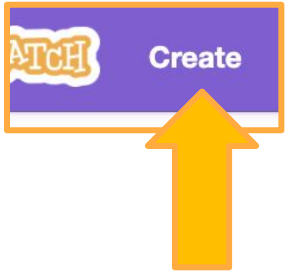
Saving

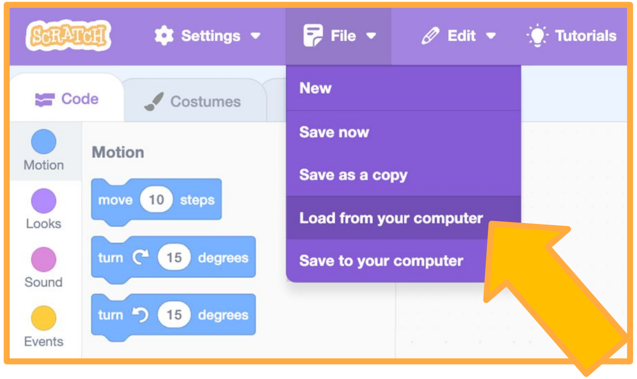
If you have a Scratch account, your project will save automatically.

If you don’t have a Scratch account yet, you can save your project to your computer. Click “**File**,” then choose “**Save to your computer**.”

Next time you want to work on your project, go to scratch.mit.edu and click “Create.” Then click “**File**,” choose “**Load from your computer**,” and upload your project.







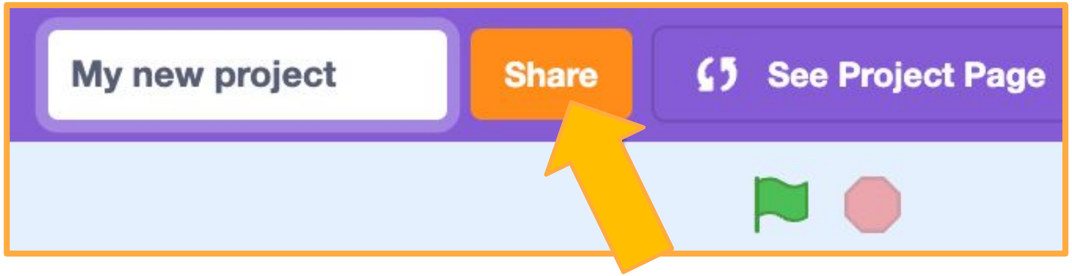
Share Your Project

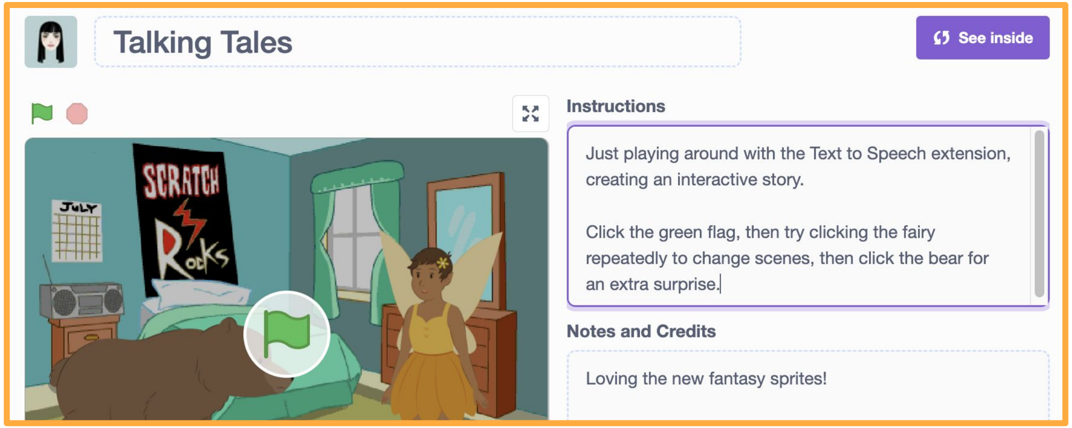
If you have a Scratch account, you can share your project and add it to studios.

Click the orange “**Share**” button at the top of the Scratch editor to share your project with the Scratch community.

Click the “See Project Page” button to go to the project page. This is where you can **add instructions and notes** about your project.

Now other Scratchers can see and interact with your project!





Just playing around with the Text to Speech extension, creating an interactive story.

Click the green flag, then try clicking the fairy repeatedly to change scenes, then click the bear for an extra surprise.

Loving the new fantasy sprites!

Reflection

- What was fun about this activity?
- What struggles or frustrations did you have during this activity?
- Many pathways, many solutions: Compare your code with other solutions. Was your solution similar or different? Why did you choose the blocks you did?
- If you had more time what would you add or change?

See our Reflection and Sharing resources for more

Prompts to Try

- I love it! What is it?
- What are your next steps for this project? What do you want to do in the future?

“Turn & Talk” is one technique to reflect and share in a physical environment.

Breakout rooms are an option for small group reflection in virtual spaces.

Record reflections using Scratch’s sound editor. Then, add to a reflection project.

After-Activity Reflection

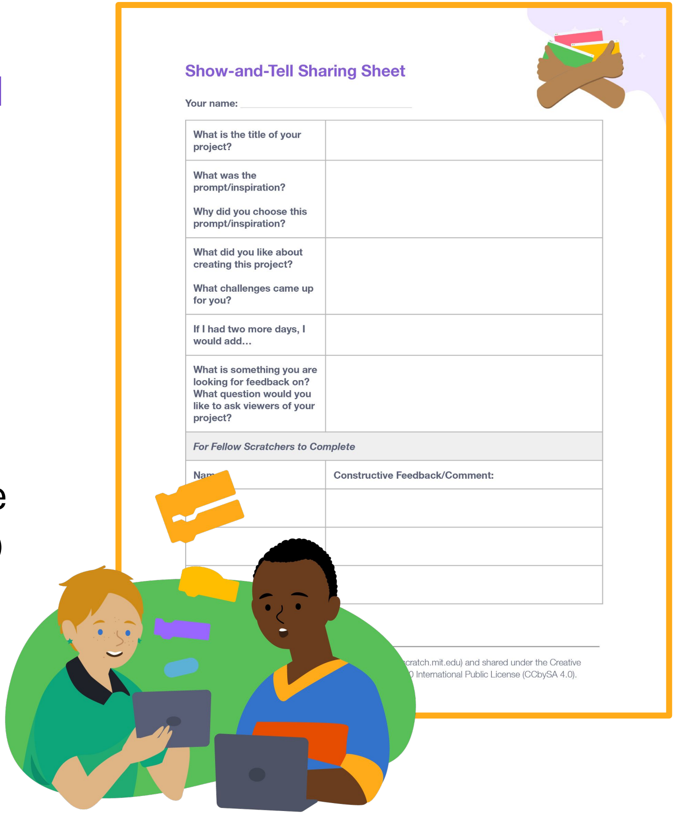
Share Option #1: Create a Class Studio to Gather Shared Projects

Studios are a space on Scratch where users can come together to make, share, and collect projects related to a particular theme, idea, or prompt.

Share Option #2: Gallery Walk

Have your project open on your computer. Walk around the room (or take turns sharing your screen in a virtual space) to experience each other's creations. Take time to look at projects and read/listen/interact with them to learn more about your peers.

More on Teacher Accounts, Studios, and our Reflection and Sharing resources



Show-and-Tell Sharing Sheet

Your name:

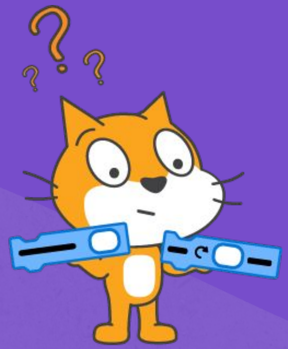
What is the title of your project?	
What was the prompt/inspiration? Why did you choose this prompt/inspiration?	
What did you like about creating this project? What challenges came up for you?	
If I had two more days, I would add...	
What is something you are looking for feedback on? What question would you like to ask viewers of your project?	
For Fellow Scratchers to Complete	
Nam	Constructive Feedback/Comment:

cratch.mit.edu) and shared under the Creative
0 International Public License (CCbySA 4.0).

SCRATCH™
FOUNDATION

Wrapping Up

Reflecting on Our Session, Resources, Next Steps



Get a copy of our Creative Learning Materials!

In addition to the resources shared throughout these slides:

- See our Learning Library at scratchfoundation.org/learn/learning-library to find lesson plans, coding cards, tutorial videos, and more! For this session, filtering by "Games" as the topic would bring up perfect resources to explore.
- Getting Started with Scratch
- Scratch Creative Learning Philosophy

Find help, inspiration, and information:

- Visit scratch.mit.edu/ideas and scratch.mit.edu/starter-projects
- Click "Tutorials" to see in-editor guides
- Watch tutorial videos on our channel youtube.com/c/ScratchTeam

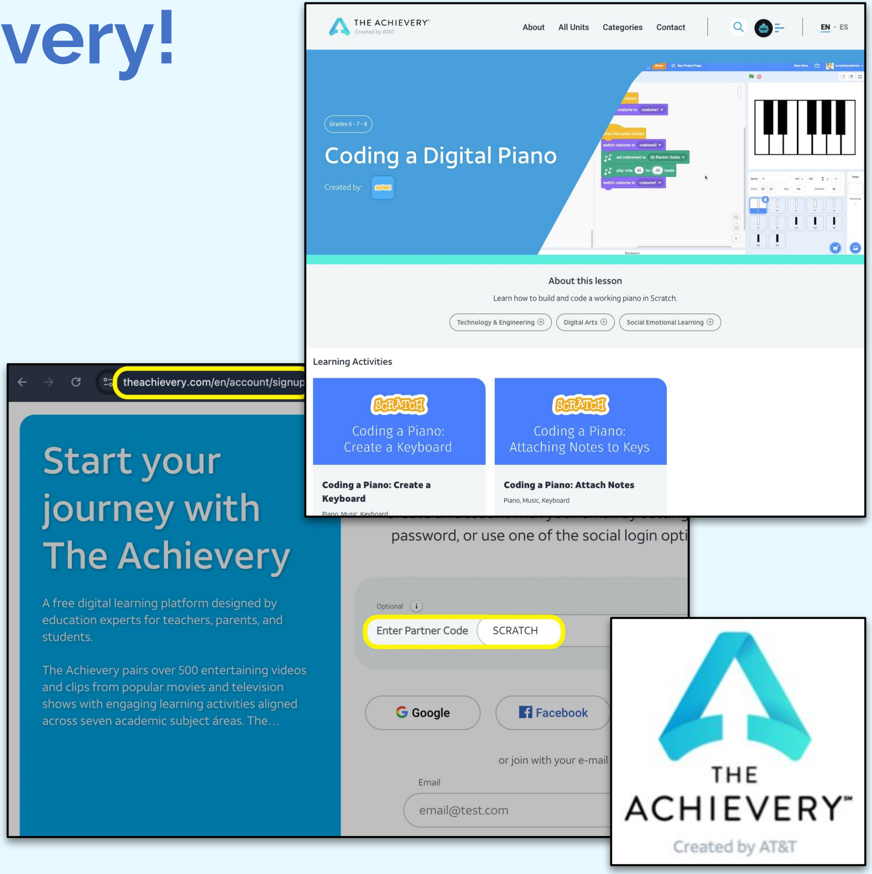
Find Scratch on The Achievery!

The Achievery platform connects K-12 students to a new world of digital learning.

Scratch Foundation has teamed up with The Achievery to provide free beginner and intermediate creative coding lesson plans on a variety of topics for educators, caregivers, and learners.

Sign up (for free!) by using our custom code "SCRATCH" when you register to support our work!

<u>**theachievery.com/account/signup**</u>



Grades 6 - 7 - 8

Coding a Digital Piano

Created by:

About this lesson

Learn how to build and code a working piano in Scratch.

Learning Activities

Coding a Piano: Create a Keyboard

Coding a Piano: Create a Keyboard

Piano, Music, Keyboard

Coding a Piano: Attaching Notes to Keys

Coding a Piano: Attach Notes

Piano, Music, Keyboard

theachievery.com/en/account/signup

Start your journey with The Achievery

A free digital learning platform designed by education experts for teachers, parents, and students.

The Achievery pairs over 500 entertaining videos and clips from popular movies and television shows with engaging learning activities aligned across seven academic subject áreas. The...

password, or use one of the social login opti

Optional

Enter Partner Code: SCRATCH

or join with your e-mail

Email: email@test.com

Thank you!

Be sure to subscribe to our Scratch Foundation YouTube channel for Educators (@scratchfoundation).

Keep an eye on our Event page for additional opportunities: scratchfoundation.org/get-involved/events

Helpful Links:

- Scratch Application: scratch.mit.edu
- Learning Library: scratchfoundation.org/learn/learning-library
- Email Signup: scratch.mit.edu/connect
- Follow us on Instagram and Facebook @ScratchTeam
- Also see our YouTube channel @scratchteam for tutorials