

@CSTANJ

@CS4NJ

#CSPDWeek2026

#CompSciNJ

@csteachersorg



New Jersey CSPDWeek

This material is based upon work supported by the National Science Foundation under Award No. 2607763.

Introducing Scratch for Grades 3-5

Meet Your Facilitators & Interactive Icebreaker



Jahaira Ortiz

Pre-K & Kindergarten Teacher

- Teacher for 10 Years
- STEM Summer Programs (PK-4)



Joann Case

K-4 Technology & PLTW Teacher

- Teacher for 28 Years
- STEMKamp (Grades 3-7)

@k2techiteacher

Interactive Poll

On a scale of 1-5, what is your experience level with block-based coding?

1

2

3

4

5

Welcome & Objectives

Learn to Create Your Own Games, Stories, and Animations!



Navigate the Scratch interface.



Understand pedagogical benefits.



Explore lesson plans for Grades 3-5.

What is GAME DESIGN?



EXAMPLE:

If you made a game where a puppy collects bones while avoiding puddles, you would decide:

Game design is when you create **your own game** for people to play. You get to imagine the **characters**, decide the **rules**, and think of **fun challenges**. You choose what happens when players **win**, **lose**, or earn **points**. Just like an author writes a story, a game designer creates an adventure for others to enjoy!

1

Who the character is



2

What the goal is



3

What obstacles are in the way



4

How to win



GAME DESIGNERS USE...



IMAGINATION



CREATIVITY



PROBLEM-SOLVING

to make games that are **FUN** for everyone to play!



— What Is Block-Based Coding?

Visual Building Blocks

A way of programming where learners create programs by **dragging and connecting visual blocks** instead of typing lines of code.

Think of It Like LEGO!

Each block represents an instruction. Learners **snap blocks together** to tell a computer, character, or robot what to do.

Interactive Code Example

When Green Flag Clicked

Move 10 Steps

Say "Hello!"

<https://scratch.mit.edu/projects/1365275890>

What is Scratch?

The world's largest visual coding platform for children

Visual Block-Based Coding

- ✓ Uses color-coded, drag-and-drop blocks instead of text syntax, eliminating typing errors and letting students focus on programming logic.

Creative & Interactive Medium

- ✓ Students can design their own characters, record sounds, and program fully interactive stories, animations, and arcade-style games.

Pedagogical Foundation

- ✓ Created by the MIT Media Lab specifically to cultivate essential 21st-century skills: collaboration, systematic reasoning, and design thinking.



This material is based upon work supported by the National Science Foundation under Award No. 2607763.

Coding Platforms & Robotics

Platforms & Software

- Scratch / Scratch 3.0
- ScratchJr
- mBlock 5
- Code.org

Robots & Devices

- BeeBot / BlueBot
- Sphero (indi, BOLT)
- litteBits
- Robot Mouse
- Codey Rocky
- mTiny Robot
- Ozobots
- Micro:Bit
- Strawbees
- Circuit Playground
- Codey Rocky
- Hands on Coding Blocks

This material is based upon work supported by the National Science Foundation under Award No. 2607763.

Building Computational Thinking Skills

- ✓ **Logical Reasoning & Troubleshooting**
Builds logical reasoning and structural troubleshooting skills.
- ✓ **Problem Breakdown & Algorithms**
Students learn to break down complex problems, recognize patterns, and design algorithms.
- ✓ **Cross-Subject Application**
Foundational logic translates across all academic subjects, enhancing overall problem-solving abilities.



Pedagogical Approach: Scaffolding Success

01

Logical Sequencing

Order instructions sequentially.

02

Creative Agency

Transform devices from consumption media to creation tools.

03

Problem-Solving

Develop early computational thinking in an accessible way.



The Scratch Interface

A creative platform for learning and sharing.

The Scratch Interface

Scratch has 4 main areas where you can create, code, and bring your ideas to life!

1 Code Blocks Palette
Find blocks of code sorted by color categories.
Drag blocks from here to build your program.

2 Scripts Area
Drag blocks here to build your program.
This is where your code comes to life!

3 Stage Area
This is where your project runs.
Watch your sprites move and interact based on your code.
Use the green flag to start and the red stop sign to stop.

4 Sprite List & Backdrop Area
See and manage your sprites (characters or objects) and backdrops (backgrounds).
Add new sprites or backdrops here.

Bonus: Top Bar
Create, save, share your projects, access tutorials, and more!

Tip: Explore each area and try dragging a block to the Scripts Area. Experiment, have fun, and let your creativity shine! ★

Interface Tour: Sprites, Stage, & Blocks

Sprites

The characters or objects that perform actions.

The Stage

The background or scene where the action takes place.

Blocks

The functional pieces of code that dictate behavior.

Block Categories

Motion (Blue) **Looks (Purple)**
Events (Yellow) **Control**
(Orange)

Classroom Implementation Strategies

Strategy 01

Technical Setup

Decide between online student accounts or the offline editor based on internet availability. Manage physical space for easy navigation.

Strategy 02

Social Strategy

Implement the 'Ask 3 Before Me' peer-to-peer debugging technique to foster collaboration and independent problem-solving.

Lesson 1

Animating a Single Character

✓ Cause-and-Effect

Introduces the concept using an 'Events' trigger (e.g., 'When Green Flag Clicked').

✓ Primary Objective

Make a Sprite move, display a speech bubble, and change its costume.



Lesson 1: Learning Outcomes

Primary Learning Outcome



Students immediately experience the satisfaction of controlling a digital object through sequential commands.

Lesson 2: Building a 'Catch' Game

Game Mechanics

✓ Motion & Controls

Implement arrow-key controls for horizontal movement.

✓ Variables

Introduce variables to track a simple score.

✓ Operators & Sensing

Program falling object mechanics using Y-axis coordinates.



Lesson 2: Game Design Logic

By combining conditional statements with variables, students learn the foundational logic behind interactive game design.

How to Create 'Catch the Fish!' Game

1

Motion & Controls

Set up your main character (like a Cat or Astronaut). Implement interactive arrow-key controls to enable smooth horizontal movement across the play screen.

2

Sensing & Spawning

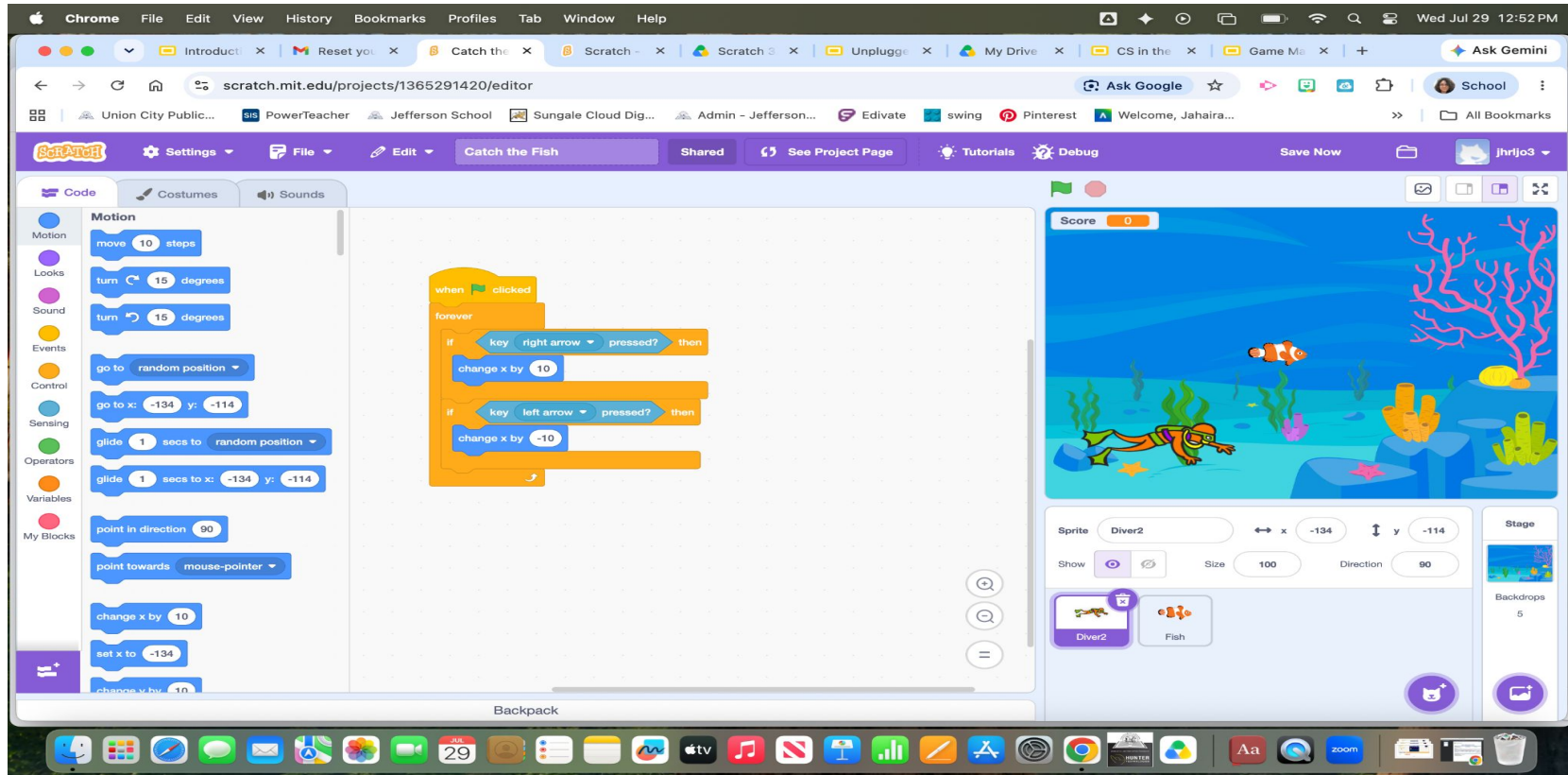
Program the falling star mechanics. Use Y-axis coordinate variables to configure the star to randomly appear, fall downwards, and disappear once caught or missed.

3

Score & Variables

Introduce dynamic score tracking variables. Set up trigger events so that whenever the active character touches a star, points are added and a fun celebration occurs.

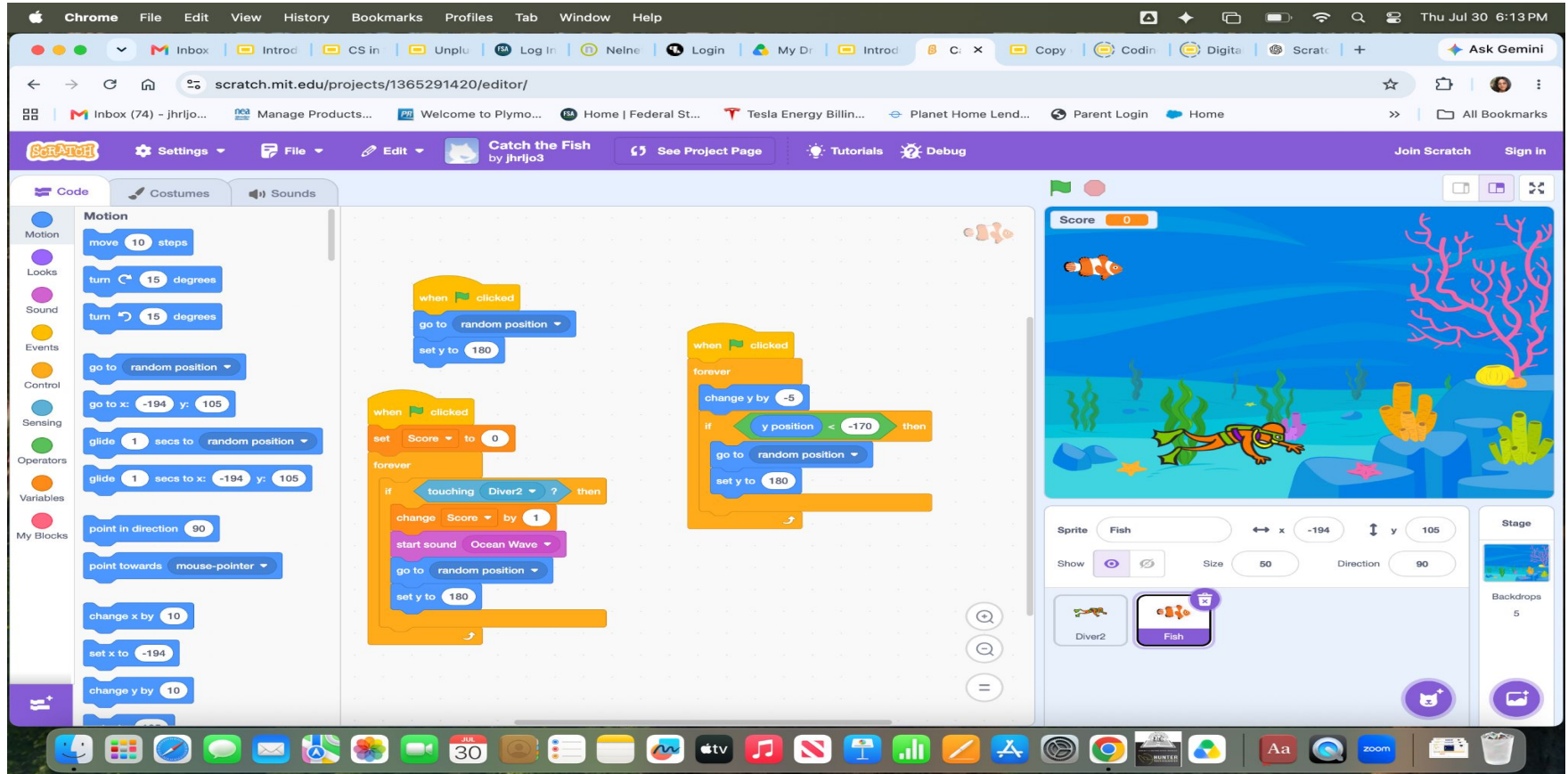
Catch a Fish Game



This material is based upon work supported by the National Science Foundation under Award No. 2607763.

<https://scratch.mit.edu/projects/1365291420>

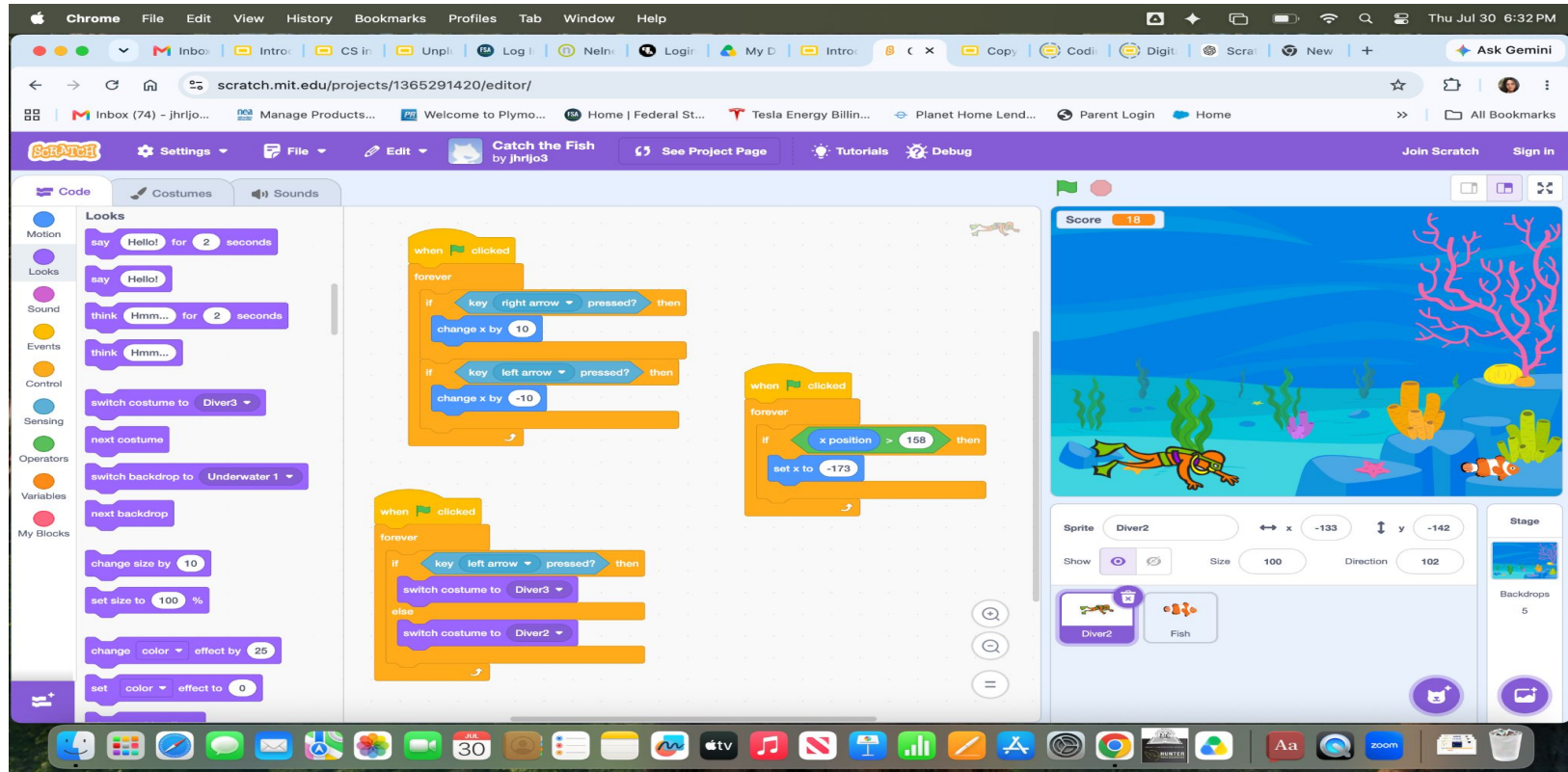
Catch a Fish Game



This material is based upon work supported by the National Science Foundation under Award No. 2607763.

<https://scratch.mit.edu/projects/1365291420>

Catch a Fish Game



This material is based upon work supported by the National Science Foundation under Award No. 2607763.

<https://scratch.mit.edu/projects/1365291420>

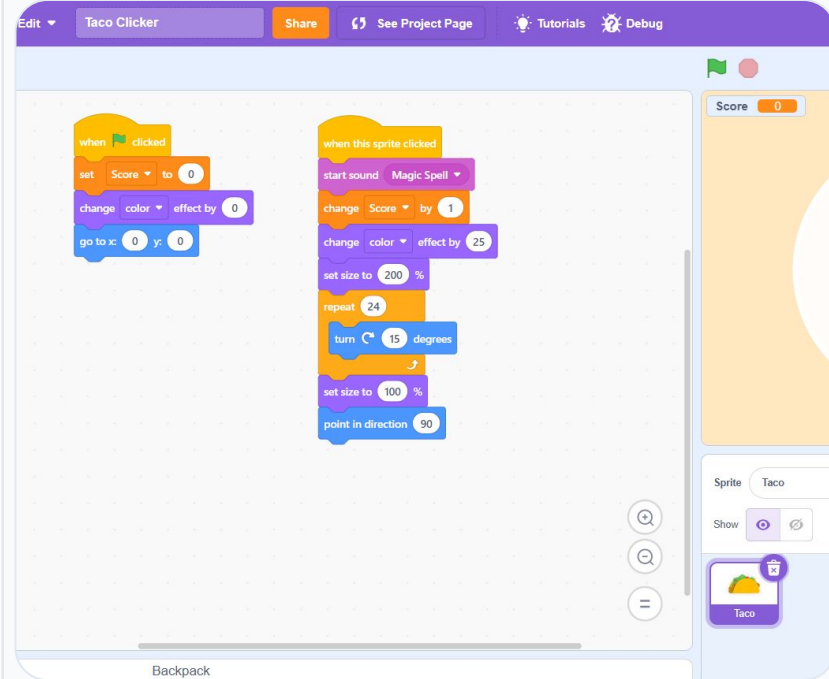
Clicker Game

What is a Clicker Game?

A clicker game (or incremental game) is an interactive experience where players click on an object on the screen—like a taco, a button, or a character—to earn points or score.

Core Mechanics

Clicking the main sprite triggers fun visual feedback, plays sounds, and increments the score. Over time, players use accumulated points to unlock upgrades, change costumes, or trigger automated clicks.



Clicker Game

The image shows the Scratch IDE interface for a project titled "Taco Clicker". The project is in the "Code" tab. The code area contains two scripts:

- When green flag clicked:**
 - move 10 steps
 - turn 15 degrees
 - turn 15 degrees
 - go to random position
 - go to x: 0 y: 0
 - glide 1 secs to random position
 - glide 1 secs to x: 0 y: 0
 - point in direction 90
 - point towards mouse-pointer
 - change x by 10
 - set x to 0
- When this sprite clicked:**
 - start sound Magic Spell
 - change Score by 1
 - change color effect by 25
 - set size to 200 %
 - repeat 24
 - turn 15 degrees
 - set size to 100 %
 - point in direction 90

The stage area shows a taco sprite on a white circular background. The score is 0. The sprite is named "Taco" and is currently at x: 0, y: 0, with a size of 100 and a direction of 90. The "Backpack" button is visible at the bottom center.

This material is based upon work supported by the National Science Foundation under Award No. 2607763.

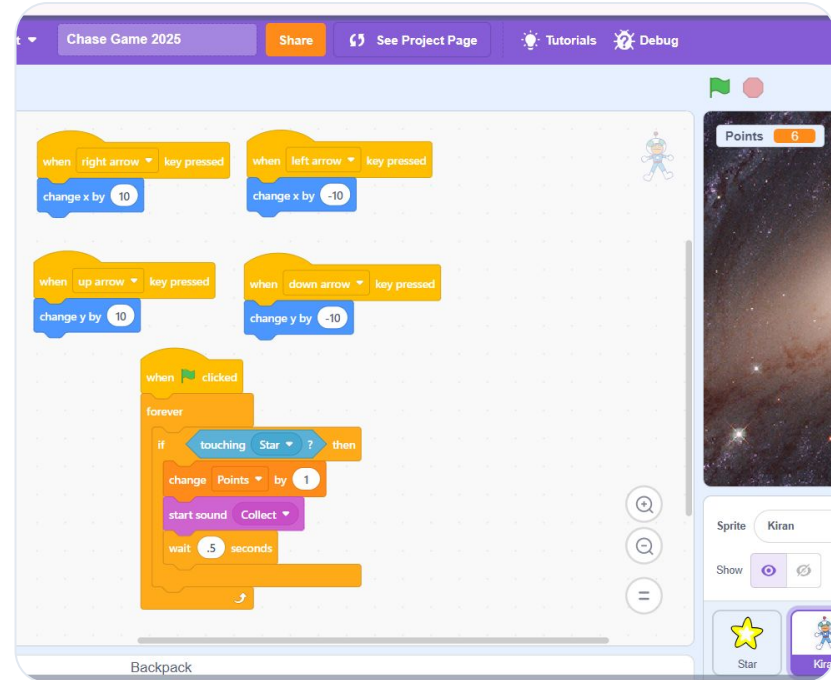
Chase Game

What is a Chase Game?

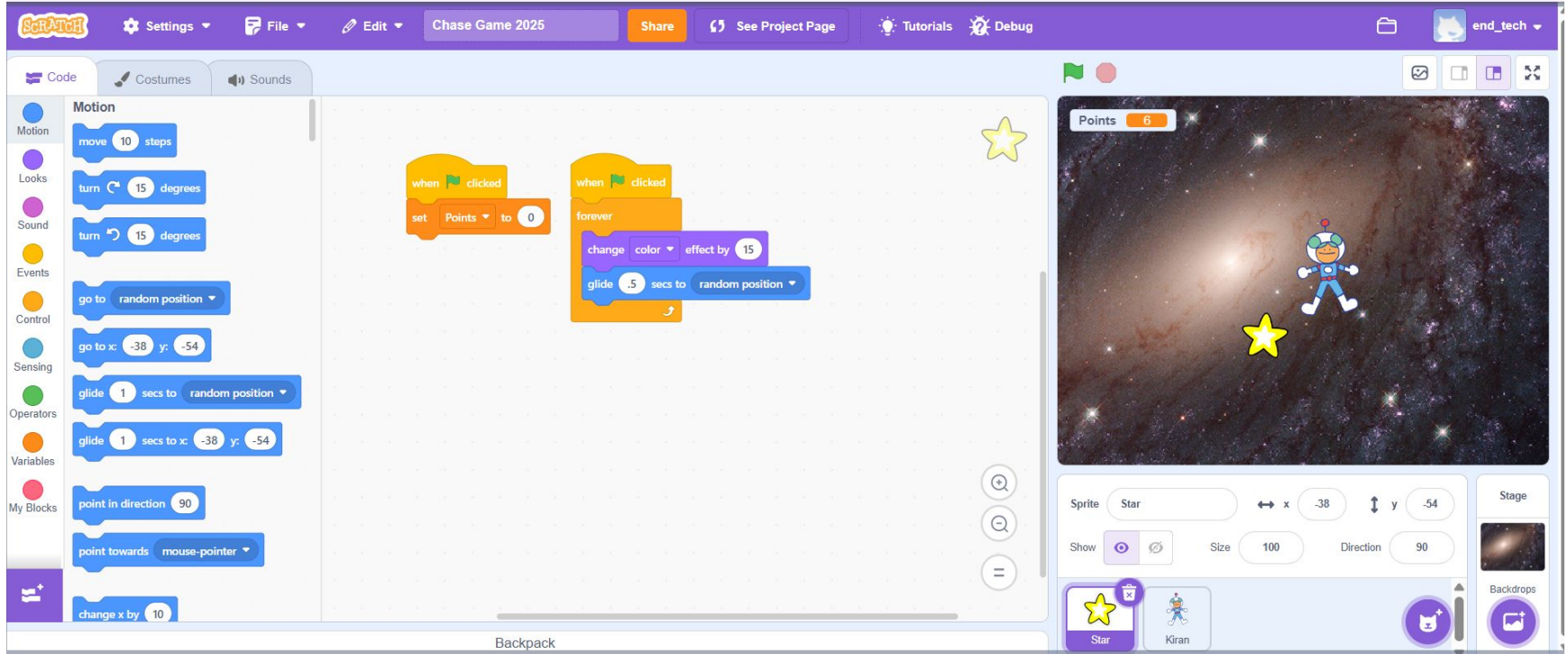
A chase game on Scratch is an interactive game where players control a character (the chaser) to pursue and catch a moving target (the prize) to score points.

Core Mechanics

Players use keyboard controls (like arrow keys) to guide their sprite across the screen. Coding logic triggers scoring events and sound feedback whenever the sprite touches the target, which then teleports to a new random position.



Chase Game



This material is based upon work supported by the National Science Foundation under Award No. 2607763.

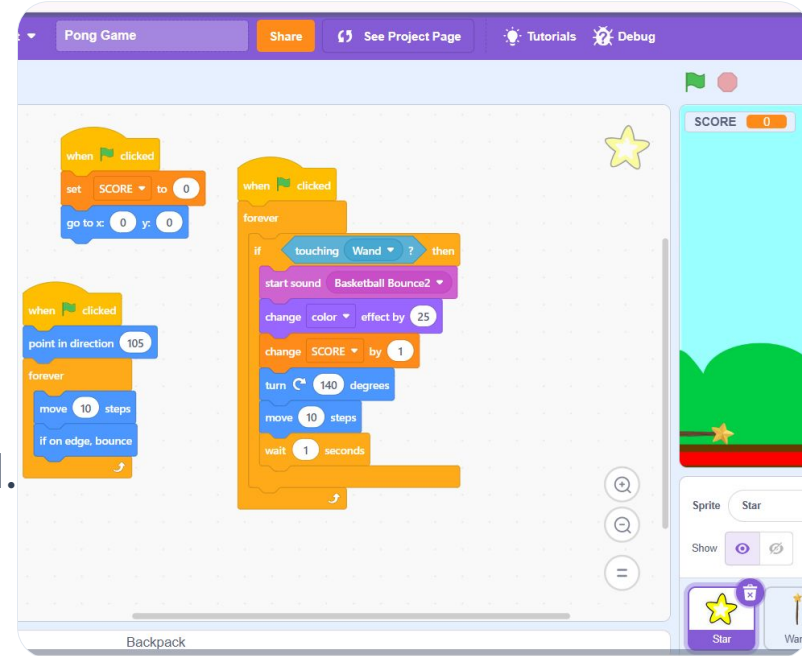
Pong Game

What is a Pong Game?

A Pong game on Scratch is a classic arcade-style game where players control a paddle to bounce a ball and prevent it from hitting the ground.

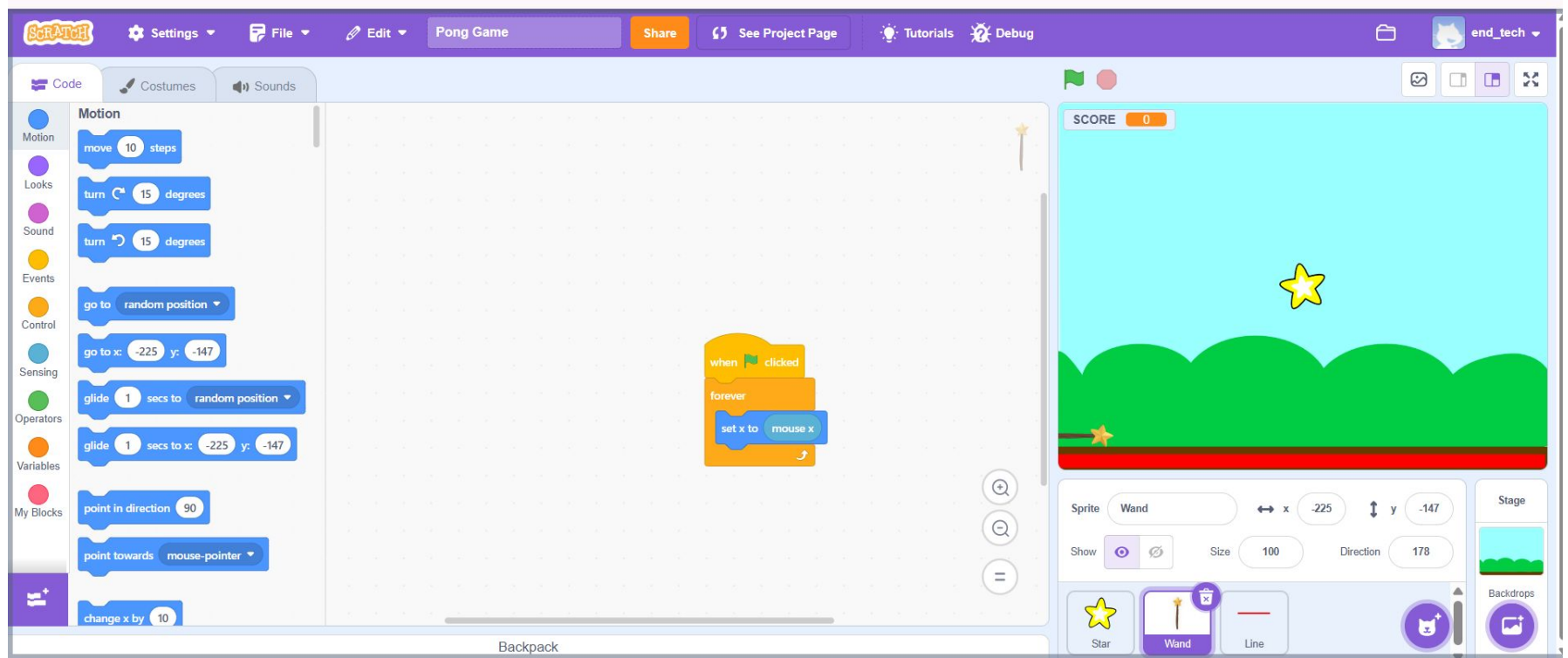
Core Mechanics

- **Paddle Control:** Players slide the paddle horizontally, often tracked to the mouse pointer's X position, to block the falling ball.
- **Ball Physics:** The ball moves continuously, bouncing off the edges and the paddle at custom angles.
- **Game Over Logic:** If the ball misses the paddle and touches a red line at the bottom, the script stops and the game ends.



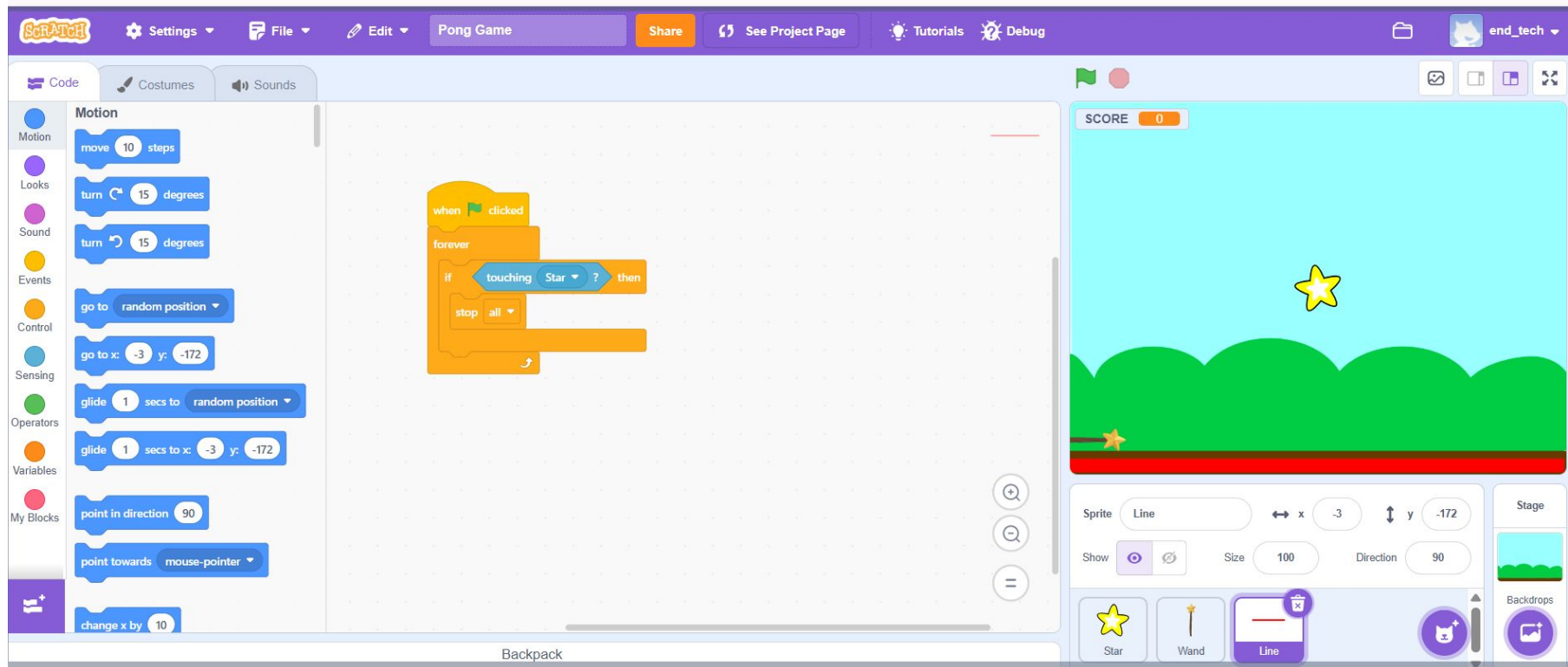
This material is based upon work supported by the National Science Foundation under Award No. 2607763.

Pong Game



This material is based upon work supported by the National Science Foundation under Award No. 2607763.

Pong Game



A screenshot of the Scratch web application showing a Pong game project. The interface includes a top menu bar with options like Settings, File, Edit, and Share. The left sidebar contains a 'Code' tab with a block palette for Motion, Looks, Sound, Events, Control, Sensing, Operators, Variables, and My Blocks. The main workspace displays a script starting with 'when clicked', followed by a 'forever' loop containing an 'if touching Star?' condition and a 'stop all' block. The right panel shows a stage with a light blue sky, green hills, and a red ground. A yellow star sprite is positioned on the ground. The 'Sprite' panel at the bottom right shows the 'Line' sprite selected, with its x and y coordinates set to -3 and -172 respectively. The 'Backdrops' panel shows a green hill backdrop selected.

Scratch

Settings File Edit Pong Game Share See Project Page Tutorials Debug

Code Costumes Sounds

Motion

- move 10 steps
- turn 15 degrees
- turn 15 degrees
- go to random position
- go to x: -3 y: -172
- glide 1 secs to random position
- glide 1 secs to x: -3 y: -172
- point in direction 90
- point towards mouse-pointer
- change x by 10

when clicked

forever

if touching Star ? then

stop all

SCORE 0

Sprite Line x: -3 y: -172

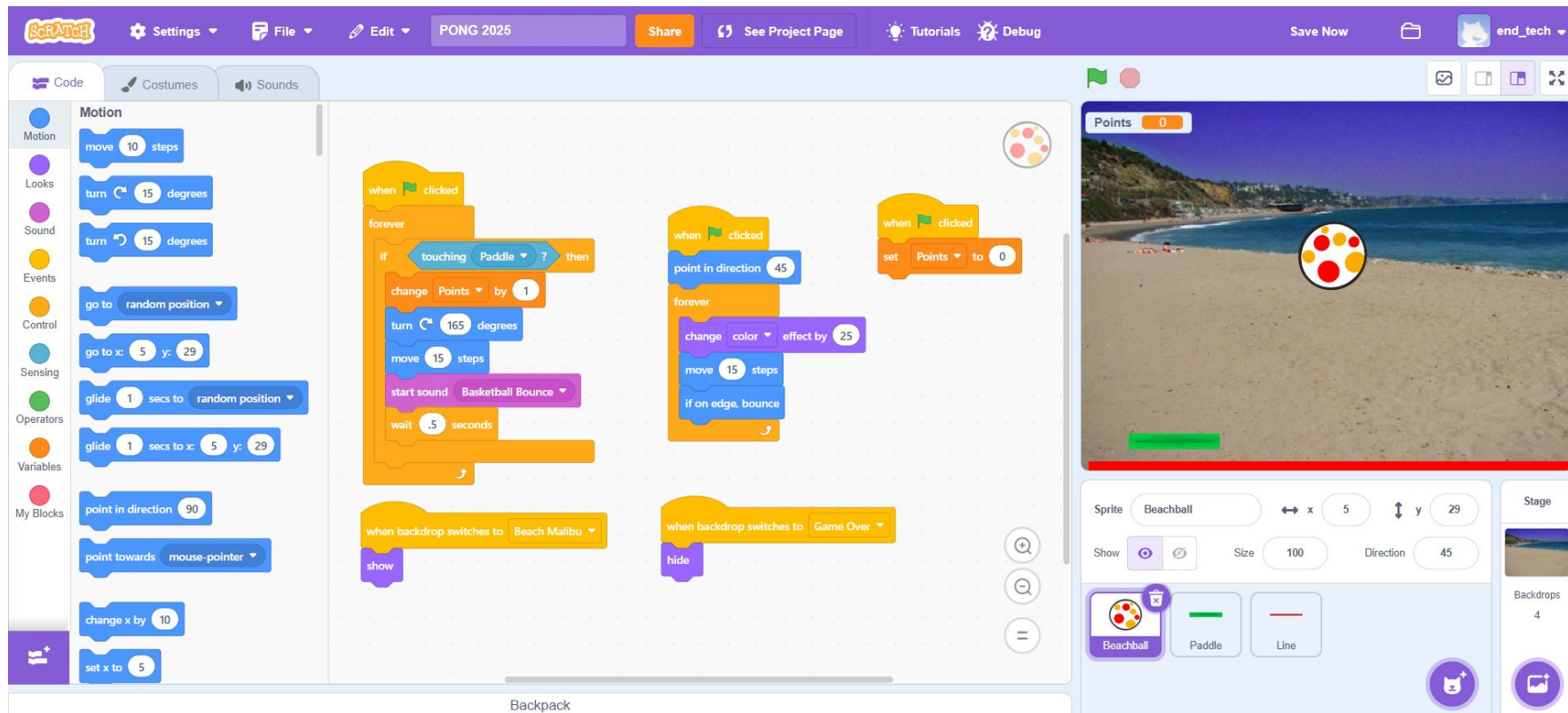
Show Size 100 Direction 90

Star Wand Line

Backpack

This material is based upon work supported by the National Science Foundation under Award No. 2607763.

Another Pong Game



A screenshot of the Scratch 3.0 interface showing a Pong game project titled "PONG 2025". The interface includes a top menu bar with options like Settings, File, Edit, Share, See Project Page, Tutorials, Debug, Save Now, and a user profile. The left sidebar shows the "Code" tab selected, with a "Motion" category highlighted. The main workspace contains several scripts and a stage preview. The scripts include: a "when clicked" event triggering a "forever" loop that checks if the ball is touching the paddle, changes points, turns 165 degrees, moves 15 steps, starts a "Basketball Bounce" sound, and waits 0.5 seconds; a "when clicked" event setting points to 0; a "when clicked" event setting the ball's direction to 45; a "when clicked" event changing the ball's color effect by 25; a "when backdrop switches to Beach Malibu" event showing the backdrop; and a "when backdrop switches to Game Over" event hiding the ball. The stage preview shows a beach scene with a ball and a paddle. The bottom right shows the "Sprite" and "Stage" panels with the "Beachball" sprite selected and the "Stage" panel showing the "Beach Malibu" backdrop.

Scratch

Settings File Edit PONG 2025 Share See Project Page Tutorials Debug Save Now end_tech

Code Costumes Sounds

Motion

when clicked

forever

if touching Paddle ? then

change Points by 1

turn 165 degrees

move 15 steps

start sound Basketball Bounce

wait 0.5 seconds

when clicked

point in direction 45

forever

change color effect by 25

move 15 steps

if on edge, bounce

when clicked

set Points to 0

when backdrop switches to Beach Malibu

show

when backdrop switches to Game Over

hide

point in direction 90

point towards mouse-pointer

change x by 10

set x to 5

Points 0

Sprite Beachball x 5 y 29

Show Size 100 Direction 45

Beachball Paddle Line

Stage Backdrops 4

Backpack

This material is based upon work supported by the National Science Foundation under Award No. 2607763.

Another Pong Game

The image displays the Scratch project editor for a game titled "PONG 2025". The interface includes a top menu bar with options like Settings, File, Edit, Share, See Project Page, Tutorials, and Debug. The left sidebar shows the "Code" tab with a list of motion blocks. The main workspace contains two scripts: one triggered by a click event that enters a forever loop to set the ball's x-position to the mouse x, and another triggered by backdrop switches to "Beach Malibu" that shows the ball and switches to "Game Over" that hides it. The right panel shows a preview of the game stage with a beach backdrop, a ball sprite, and a "Points" score of 0. The bottom right panel shows the sprite list with "Beachball", "Paddle", and "Line" sprites.

Scratch Project: PONG 2025

Code:

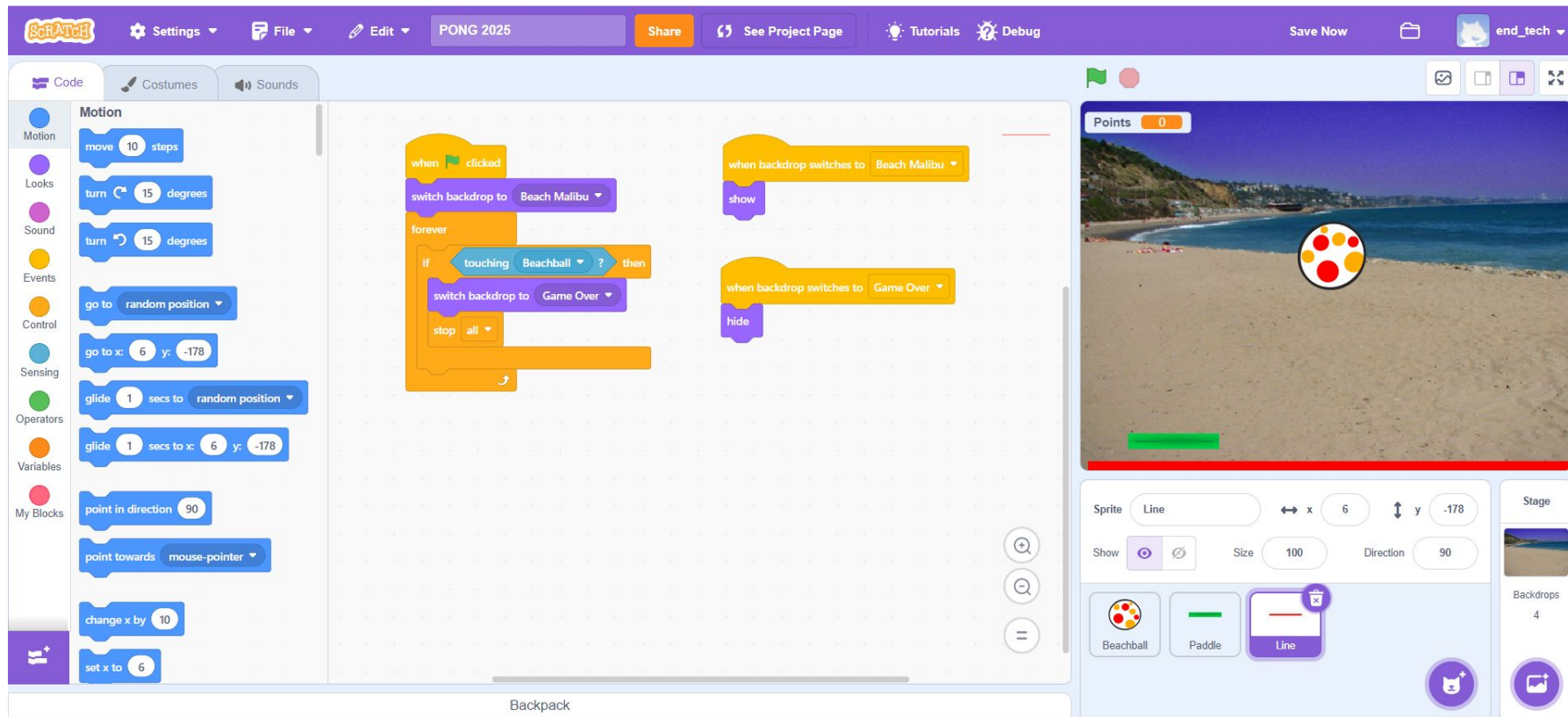
- Script Area:**
 - when green flag clicked
 - forever loop
 - set x to mouse x
 - when backdrop switches to Beach Malibu
 - show
 - when backdrop switches to Game Over
 - hide

Sprite Area:

- Sprite: Paddle
 - x: -149, y: -151, Size: 100, Direction: 90
- Backdrops: 4
 - Beachball
 - Paddle
 - Line

This material is based upon work supported by the National Science Foundation under Award No. 2607763.

Another Pong Game



The image shows the Scratch IDE interface for a Pong game. The top bar includes the Scratch logo, settings, file, edit, and share options, along with the project name "PONG 2025". The left sidebar shows the "Code" tab selected, with a "Motion" category highlighted. The main workspace contains the following code blocks:

- when green flag clicked**
 - switch backdrop to Beach Malibu
 - show
- forever loop**
 - if touching Beachball ? then
 - switch backdrop to Game Over
 - stop all
- when backdrop switches to Beach Malibu**
 - show
- when backdrop switches to Game Over**
 - hide

The right sidebar shows the stage preview, which displays a beach scene with a ball in the center. The stage properties are set to x: 6, y: -178, size: 100, and direction: 90. The stage backdrop is "Beach Malibu". The bottom right shows the "Backpack" section with a "Line" sprite.

This material is based upon work supported by the National Science Foundation under Award No. 2607763.

Another Pong Game

The image displays the Scratch IDE interface for a project titled "PONG 2025". The interface includes a top menu bar with options like Settings, File, Edit, Share, See Project Page, Tutorials, and Debug. Below the menu is a toolbar with tabs for Code, Costumes, and Sounds. The left sidebar contains a palette of block categories: Motion, Looks, Sound, Events, Control, Sensing, Operators, Variables, and My Blocks. The main workspace shows three scripts for a "Beachball" sprite. The first script, triggered by a "when clicked" event, contains a "forever" loop with an "if touching Paddle?" condition. If true, it increments the "Points" variable by 1, turns the sprite 165 degrees, moves it 15 steps, plays a "Basketball Bounce" sound, and waits 0.5 seconds. The second script, also triggered by "when clicked", sets the "point in direction" to 45 and enters a "forever" loop where it changes the "color" effect by 25 and moves 15 steps, with an "if on edge, bounce" action. The third script, triggered by "when backdrop switches to Beach Malibu", shows the sprite. A fourth script, triggered by "when backdrop switches to Game Over", hides the sprite. The right sidebar shows the "Stage" area with a "Points" variable set to 4 and a "Game Over" backdrop. The "Sprite" area shows the "Beachball" sprite with x: -74, y: -145, size: 100, and direction: -15. The "Backdrops" area shows the "Game Over" backdrop with a count of 4. The "Beachball" sprite is currently visible on the stage.

This material is based upon work supported by the National Science Foundation under Award No. 2607763.

Curriculum & Management Resources

- ✓ **Teacher Accounts:** Manage classes and monitor student progress easily.
- ✓ **Block Cards:** Provide printable block cards as physical reference guides.
- ✓ **Starter Projects:** Transition to Starter Projects to encourage self-directed exploration and remixing.



NJSLS Computer Science Standards: Grades 3-5

ALGORITHMS & PROG

8.1.5.AP.1

Algorithm Selection

Compare and refine multiple algorithms for the same task and determine which is most appropriate.

8.1.5.AP.2

Variables & Data

Create programs that use clearly named variables to store and modify data.

CONTROL & PROCESS

8.1.5.AP.3

Control Structures

Create programs that include sequences, events, loops, and conditionals.

8.1.5.AP.4

Problem Decomposition

Break down problems into smaller, manageable sub-problems to facilitate program development.

NJSLS Computer Science Standards: Grades 3-5

CREATION & ITERATION

8.1.5.AP.5

Remixing & Modifying

Modify, remix, or incorporate pieces of existing programs into one's own work to add features.

8.1.5.AP.6

Iterative Development

Develop programs using an iterative process, implement designs, and test to ensure correct functionality.

Interactive Reflection: Brainstorming Applications

Step 1

Draft Your Plan

Write down one upcoming lesson plan where Scratch can be integrated.

Step 2

Collaborate & Share

Share your drafted plan to gather feedback and inspire others.

Scratch Lesson Plan Example



Scratch Lesson: Create an Interactive Story!
Grades 3–5 | Code. Create. Share.

LEARNING OBJECTIVES

- Use basic Scratch blocks
- Create a sequence of instructions
- Use events and dialogue
- Solve problems and debug
- Be creative and tell a story

YOU WILL USE

- Events
- Looks
- Motion
- Sound (optional)

CHALLENGE

Create an interactive story with two characters that talk and interact!

SHARE & REFLECT

Test your story.
Fix what you need.
Share with a classmate!

OUR PROCESS

PLAN → CODE → TEST → DEBUG → SHARE

PLAN: Think about your story.
CODE: Use blocks to bring it to life.
TEST: Run your project and check it.
DEBUG: Fix problems and improve.
SHARE: Show your story and celebrate!

The graphic illustrates a Scratch lesson plan for creating an interactive story. It features a central image of a Scratch project on a laptop screen, showing a cat and a dog character with dialogue boxes. The project includes a script area with blocks like 'when clicked', 'say Hello! for 2 seconds', 'move 10 steps', 'say Nice to meet you! for 2 seconds', 'wait 1 seconds', 'move -10 steps', and 'say Let's go on an adventure! for 2 seconds'. The stage area shows the cat and dog characters. The background is a green field with trees. The Scratch interface includes a menu bar (File, Edit, Tutorials), a sidebar with categories (Events, Motion, Looks, Sound, Control, Sensing, Operators, Variables, My Blocks), and a stage area with a sprite (Cat) and a backdrop (Stage). The lesson plan includes learning objectives, a list of tools to be used, a challenge, a share and reflect section, and a five-step process: Plan, Code, Test, Debug, and Share.



Lesson Plan Resource

Access the complete, printable PDF lesson plan to guide your classroom through creating an interactive story in Scratch.

[Scratch Lesson Plan.pdf](#)

CSTA: New 2026 Standards

Explore the New Guidelines

The Computer Science Teachers Association (CSTA) has updated its standards for 2026.

Access the official interactive portal to browse, search, and implement these benchmarks in your PreK-12 curriculum.

Official CSTA Portal

Use the interactive database tool to quickly filter standards by grade band, concept, or keyword.



[Interactive Tool](#)

Survey & Feedback

Requirements

After Each Session

Please complete the feedback survey after you finish each session.

End of the Day

Complete one final survey at the end of the day to ensure you receive your **STIPEND**.

Volunteer Activism

With Gratitude

Thank You!

Thank you for choosing CS in your classrooms and dedicating your effort to computational thinking!



Q&A & Support Channels

What questions do you have about implementing Scratch in your classroom?

@CSTANJ

@CS4NJ

#CSPDWeek2026

#CompSciNJ

@csteachersorg



New Jersey CSPDWeek

This material is based upon work supported by the National Science Foundation under Award No. 2607763.