

@CSTANJ

@CS4NJ

#CSPDWeek2026

#CompSciNJ

@csteachersorg



# New Jersey CSPDWeek

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



# **Coding in Early Childhood: Game Design with Scratch Jr.**

Empowering K-2 Educators through Creative Play &  
Computational Thinking

# Meet Your Presenters & 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)

@k2techieteacher  
[jcase@nhanover.com](mailto:jcase@nhanover.com)

### Interactive Poll

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

1

2

3

4

5

## Learning Objectives

By the end of this workshop, educators will:

- 1 Navigate ScratchJr interface comfortably.**
- 2 Understanding game design.**
- 3 Debug common errors.**
- 4 Adapt ScratchJr coding concepts for PreK-Grade 2 learners.**

# 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!



# 📺 Why ScratchJr? Coding in Early Childhood

## Logical Sequencing

Children learn to order instructions sequentially. They discover that computer programs execute instructions left-to-right, helping build reading orientation and logical reasoning patterns.

## Creative Agency

Scratch Jr transforms devices from consumption media to creation tools. Students paint custom characters, record voices, and implement personalized logic to direct active gameplay.

## Problem-Solving

Using block-based coding, children develop early computational thinking, logical sequencing, and problem-solving skills in a fun and accessible way.



# 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!"

# Block Categories

| <b>Category</b>   | <b>Scratch Jr. Blocks</b>                                                          | <b>Function</b>                    |
|-------------------|------------------------------------------------------------------------------------|------------------------------------|
| Movement Blocks   |  | Move Characters                    |
| Triggering Blocks |   | Start a Program                    |
| Looks Blocks      |  | Changes the Characters' Appearance |
| Sound Blocks      |   | Play a Sound                       |
| Control Blocks    |   | Controls Part of a Program         |
| End Blocks        |  | End a Program                      |

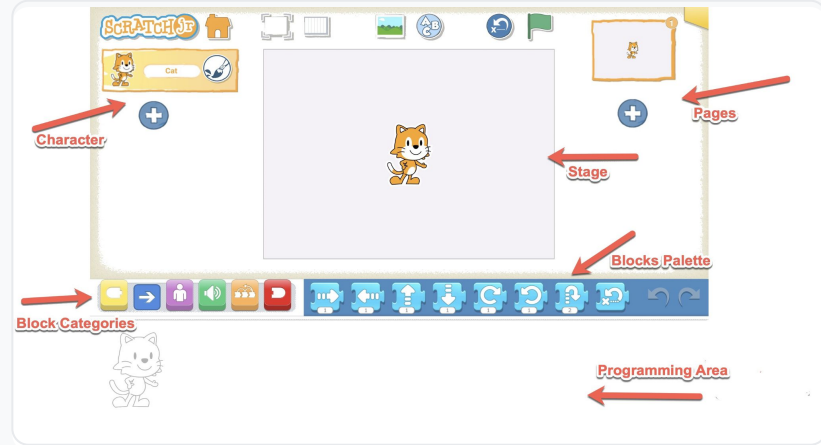
# Play: Exploring the ScratchJr Interface

- **Stage & Pages:** Where stories come to life across multiple scenes.
- **Characters (Sprites):** Choose, customize, or paint the actors of your project.
- **Programming Area:** Snap blocks together to build character scripts.
- **Block Categories:** Browse color-coded command blocks.

## Interactive Prompt

"Locate the green flag block. What category of block is it?"

## ScratchJr Interface Diagram



**Quick Tip:** Use the color-coded categories at the bottom left to filter and choose your coding commands.

# Differentiated Adaptations: K-2 Pathways

## Pre-K & Kindergarten

Exploratory focus

### Focus

Cause and effect. Keep scripts simple (1-3 blocks).

### Strategy

Use "pair programming" to encourage verbalizing their code.

## Grade 1

Foundational focus

### Focus

Sequencing and basic loops.

### Strategy

Introduce the "Repeat" block to simplify long chains of movement.

# Differentiated Adaptations: K-2 Pathways

## **Grade 2**

Advanced focus

### **Focus**

Conditional logic and project-based storytelling.

### **Strategy**

Challenge students to use the "Message" block to trigger multiple character interactions.

# Connecting Code to ELA & Math



## Math Integration

### Counting Steps & Patterns

Programming sequential motion using numeric arguments and identifying repeating logical patterns in instruction loops.



## ELA Integration

### Interactive Storytelling & Sequencing

Creating rich character arcs, mapping sequences of events (Beginning, Middle, End), and engaging in digital journaling.

### Language & Narrative Skills

Supports vocabulary growth, sentence structure, and narrative development as children plan, describe, and share stories with peers.

# Classroom Management: Coding with Early Learners

## Practical Tips

- 'Heads Up, Screens Down' rule.
- Peer troubleshooting ('Ask 3 Before Me').
- Organizing tablets.
- Using physical/tactile block printouts to design unplugged algorithms.

Provide students with tactile coding blocks to bridge abstract learning to the concrete application.

Explain that students can use the Say block to add text to speech bubbles to show what the characters say.





# Classroom Management: Supply Organization

Color Code the Materials



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

# Additional Ideas for All Learners

Provide students with coding blocks to plan out their game programs



Name: \_\_\_\_\_

Date: \_\_\_\_\_

# ★ MY GAME DESIGN MAP ★

Plan your game! Draw your ideas and write a few words.

Ages  
5-7

## 1 MY GAME TITLE

What is your game called?



## 2 CHARACTER

Who is the player?  
Draw your character.



Character name: \_\_\_\_\_

## 3 GOAL

What is the goal of your game?  
What does the player need to do?



## 4 OBSTACLES

What are the problems or things  
to avoid?  
Draw and write.



## 5 HOW TO PLAY

What are the rules?  
Write or draw the steps.



1

2

3

4

## 6 HOW TO WIN

How does the player win?  
Draw or write.



## 7 MY GAME SCREEN (What will your game look like?)

Draw your game scene.





SCRATCHjr





## My Projects



# Choose a Background



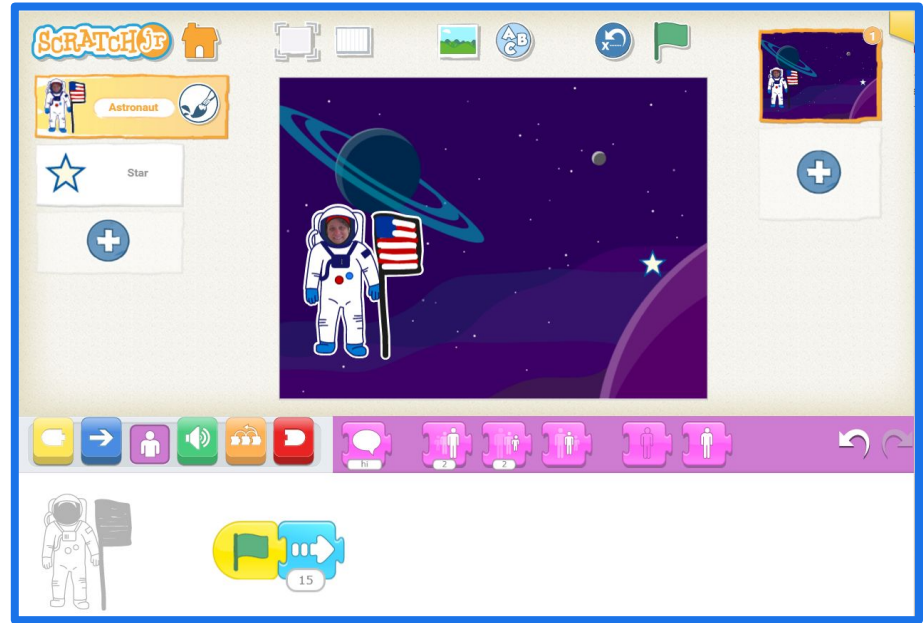
# Choose a Character



# Create a Basic “Catch a Star” Game

## Steps

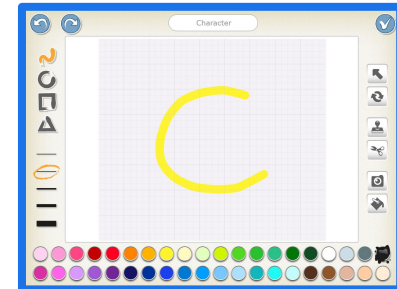
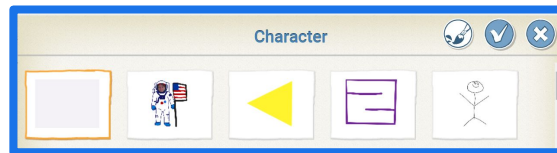
- Choose a background (I chose Space.)
- Choose 2 characters (sprites)
  - Edit and resize as needed
    - Astronaut and the Star (your choice)
    - Use the “Shrink” or “Grow” block to resize characters as needed.
- To Edit a Character
  - Select the Character
  - Click on the Paintbrush
    - Edit color as needed
    - Use the camera tool to add faces
  - Click the checkmark when done
- Write the programs for each character



# Create a Basic Matching Game

## Steps

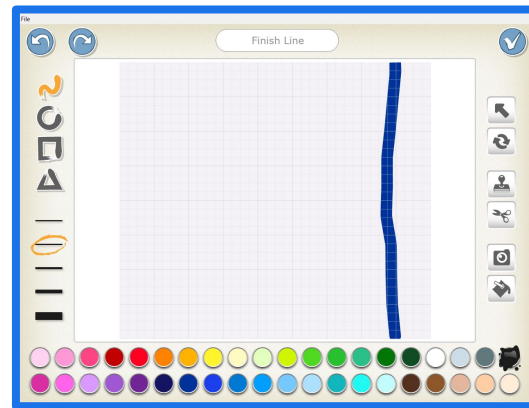
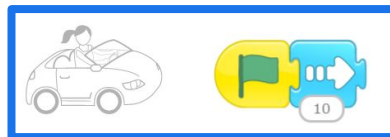
- Choose a background (I chose the Park.)
- Choose 2 characters (sprites)  
Edit and resize as needed
  - Cat and Dog (your choice)
  - Use the “Shrink” or “Grow” block to resize characters as needed.
- Make a New Character (Sprite):
  - Click the link for steps
  - Click to add a new character.
    - Click the white paper
    - Click the paintbrush
    - Use the tools to create one
  - Click the checkmark when done.
- Write the programs for each character



# Create a Race Game

## Steps

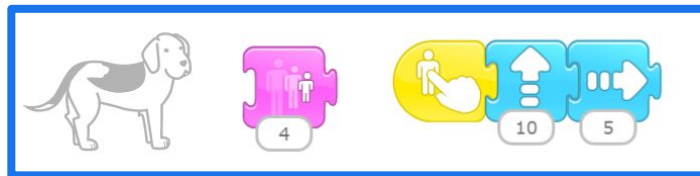
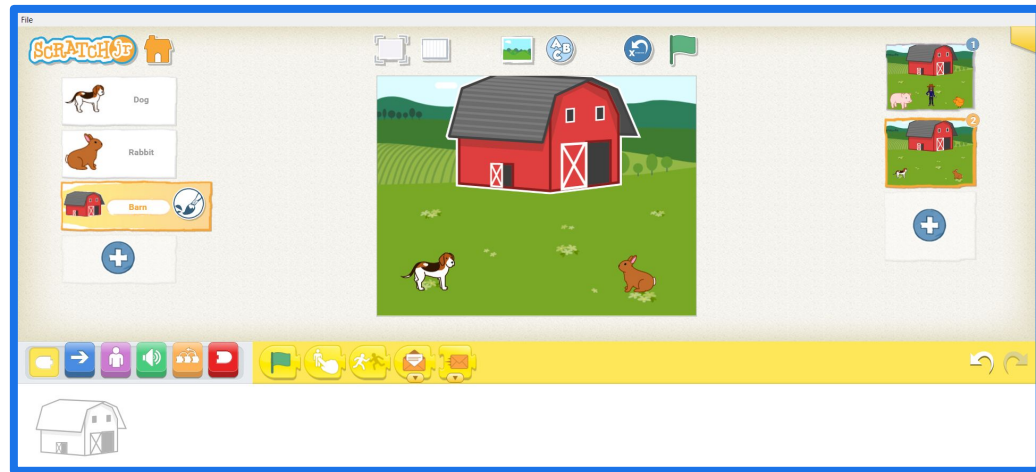
- Choose a background (I use the farm.)
- Choose 2 characters (sprites)  
Edit and resize as needed
  - Cars, bikes, boats
  - Animals, People, Insects
  - Use the camera tool to add faces
  - Use the “Shrink” block to resize characters to the same size.
- Make a new character: the finish line
  - Use the “Start On Bump” block
  - Use the green microphone block
    - record a message
- Write the programs for each character



# Create a Hide the Animals Game

## Steps

- Choose the Farm background
- Choose 3 characters (sprites)  
Edit, flip, and resize as needed
  - Dog
  - Rabbit
  - Barn
- Discuss layers. Grab the barn & cover the other animals. Place the barn back in place at the top of the stage.
- Write the programs for each character (sprite)

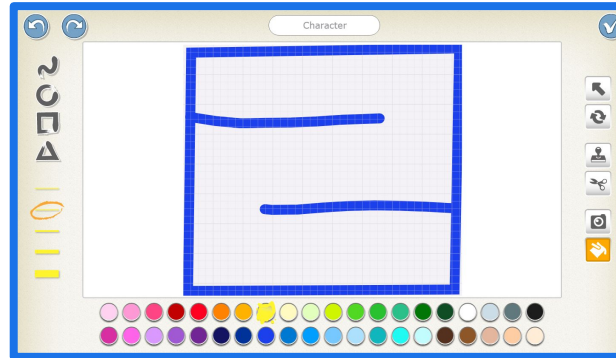
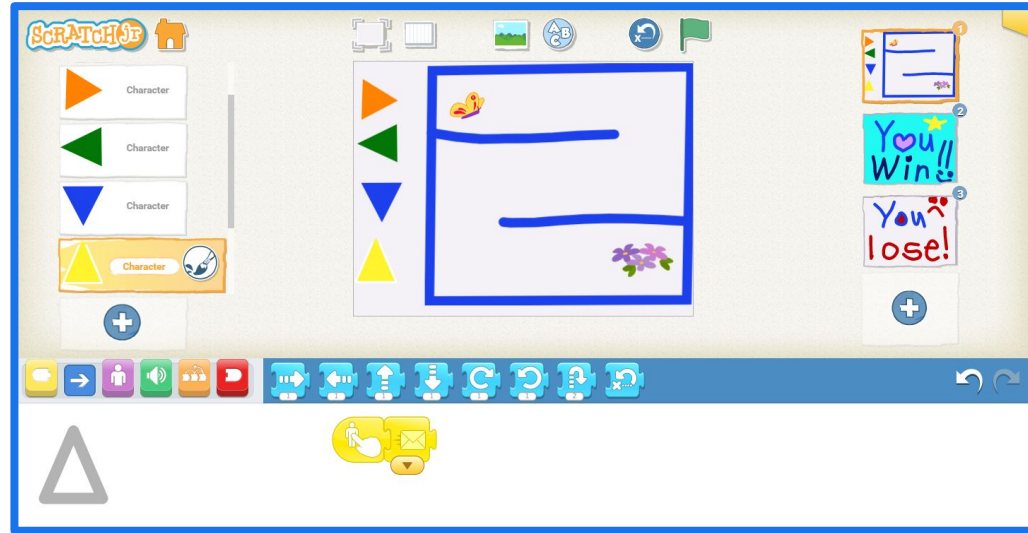




# Create an Easy Maze

## Steps

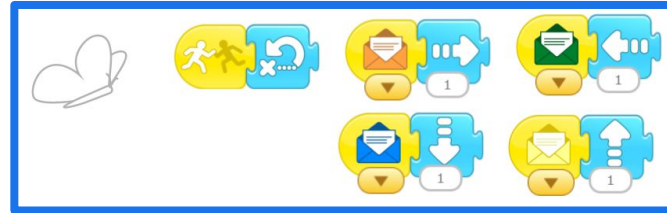
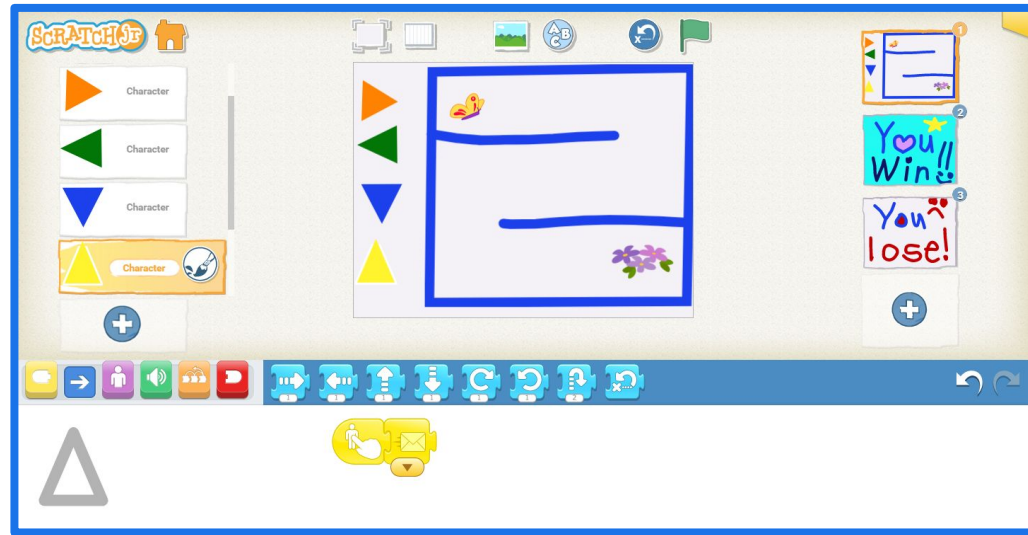
- Create Characters in the Paint Editor
  - The Maze (Do NOT Fill In)
  - Orange Triangle (Facing Right)
  - Green Triangle (Facing Left)
  - Blue Triangle (Facing Down)
  - Yellow Triangle (Facing Up)
- Add and Edit Characters as Needed
  - Fill, rotate and resize
- Choose the butterfly and the flower or 2 characters of your choice.
- Write the programs for each character



# Create an Easy Maze

## Steps

- Write Character Programs
  - Orange Triangle (Facing Right)
  - Green Triangle (Facing Left)
  - Blue Triangle (Facing Down)
  - Yellow Triangle (Facing Up)
  - Butterfly
- Discuss Blocks Needed
  - Start On Tap
  - Start On Bump
  - Start On Message (Trigger)
  - Send Message (Trigger)
  - Go Home (Movement)



| Sprite                                                                              | Arrows                                                                               |
|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
|  |   |
|  |   |
|  |   |
|  |  |

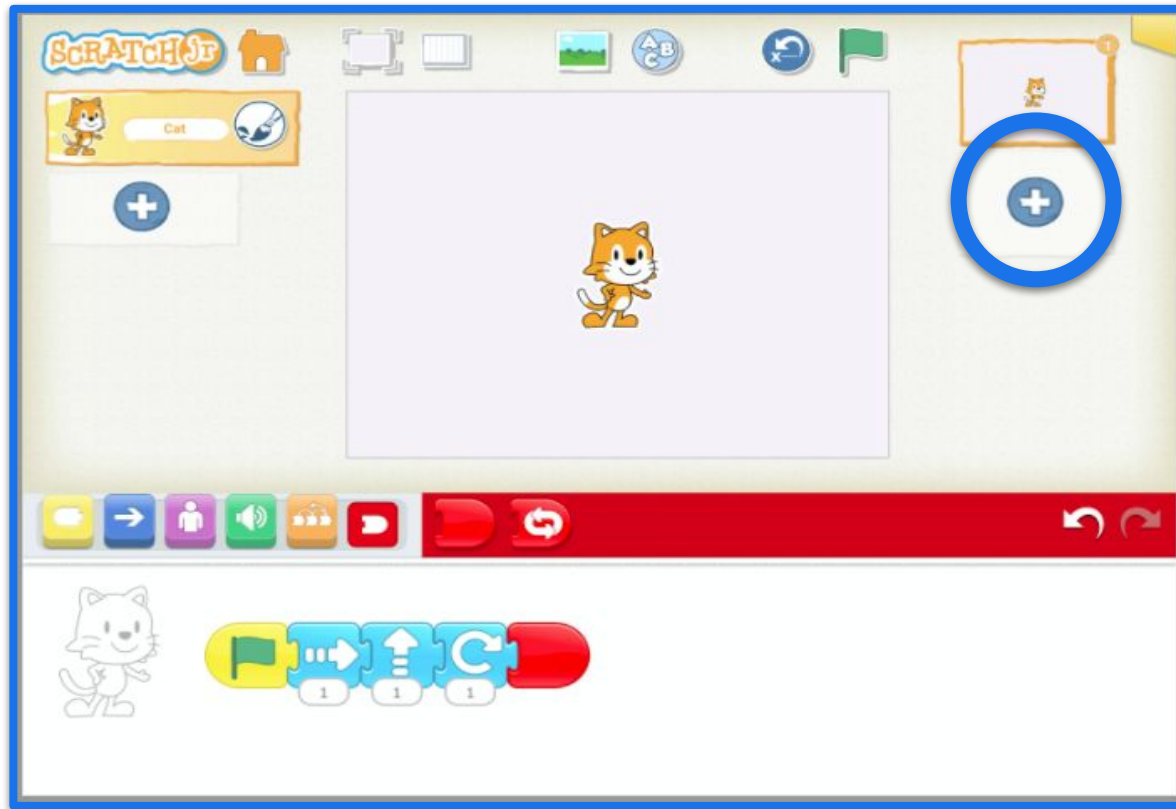
# Create an Easy Maze

## Steps

- Write Character Programs
  - Maze
  - Flower
    - Discuss the End Blocks
- Add 2 Blank Pages (See Slide 22)
  - Delete the sprite
- Create 2 Backgrounds (1 for each page)
  - Click on the background Icon
    - Click on the white paper
    - Click on the paintbrush
  - In the Paint Editor create a
    - "You Win!" page
    - "You Lose!" page



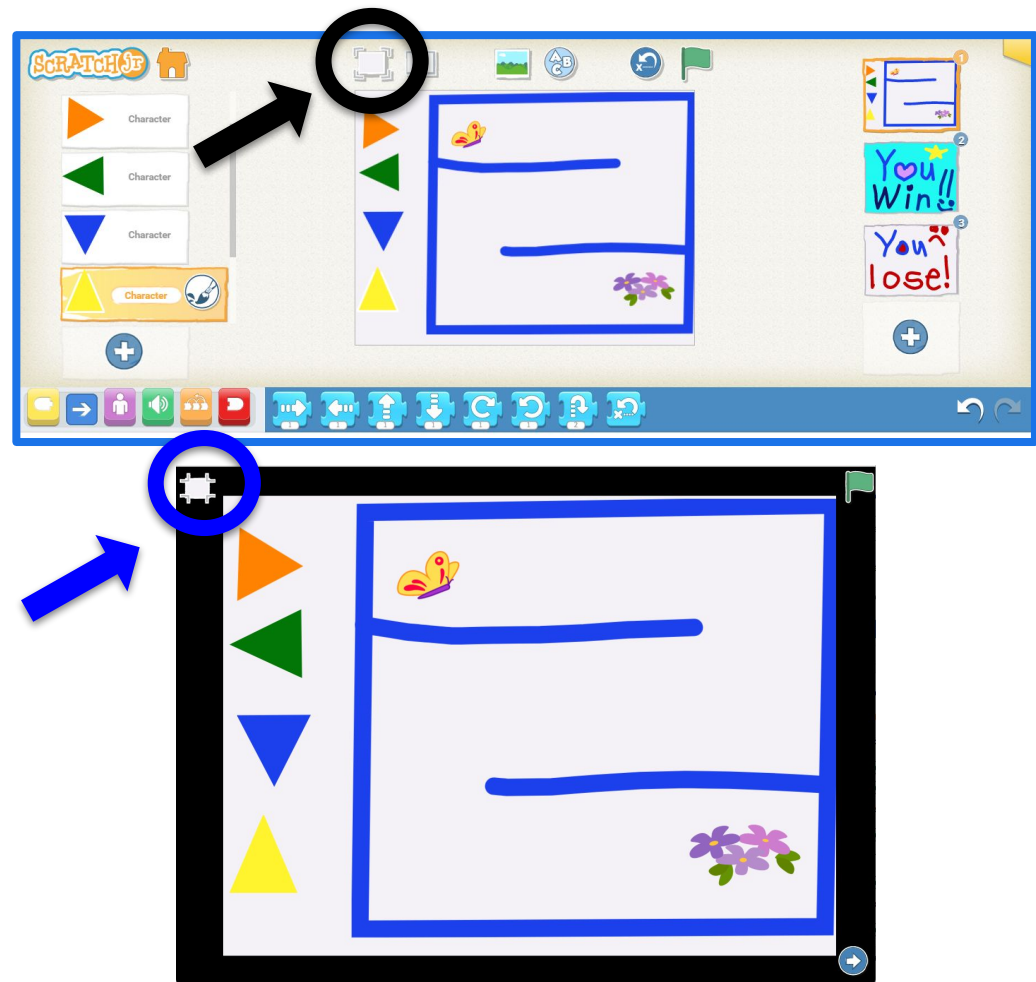
# Add a Page



# Create an Easy Maze

## Steps

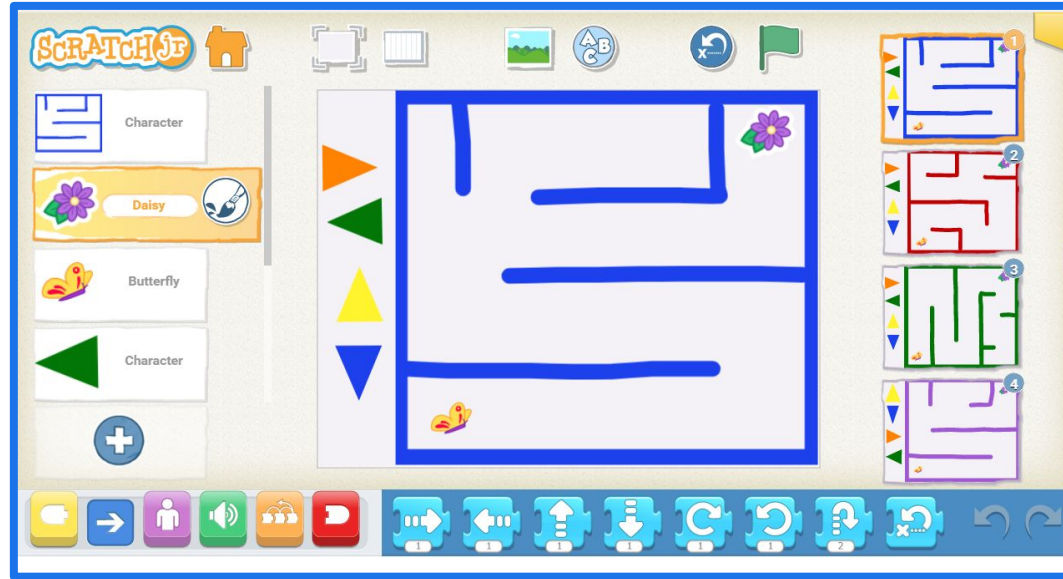
- Click on the Full Screen Icon
- Play to Test the Programs
- Debug as Needed
- Click on the Exit Full Screen Icon to return to the ScratchJr. interface



# Differentiate and Create a Hard Maze

## Steps

- To begin, follow the Easy Maze directions (Slides 25-29)
- Add 3 More Pages (Slide 28)
  - Create a page for additional mazes.
    - Choose a different color for each maze.
  - Delete the sprite.
  - Add Characters
- Drag and drop characters from one page to another page
  - See directions. Go to Slide 75
  - All programs will transfer over
  - Edit as needed





## Algorithms & Programming (AP)

### **8.1.2.AP.3** (Sequences & Loops)

Create programs with sequences and simple loops to accomplish tasks.

### **8.1.2.AP.4** (Sequence of Steps)

Break down a task into a sequence of steps.

### **8.1.2.AP.6** (Debugging)

Debug errors in an algorithm or program with sequences/loops.



## Computing Systems (CS)

### 8.1.2.CS.1 (Devices & Tools)

Select and operate computing devices that perform a variety of tasks.

**Workshop: Navigating tablet interface & ScratchJr**

### 8.1.2.CS.3 (Troubleshooting)

Describe basic hardware and software problems using accurate terminology.

**Workshop: "Ask 3 Before Me" peer-troubleshooting**

# 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 new computer science education 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**

# CSTA: New 2026 Standards

## Standards Documents

- **PreK/K Standards**
- **Grade 1 Standards**
- **Grade 2 Standards**
- **Grade 3+ Standards**
  - Locate at the bottom of the CSTA resource page

# **Apply: Developing Your Action Plan**



## **Classroom Action Plan**

**Write down one upcoming lesson plan where ScratchJr can be integrated.**

## **Reflection Gallery**

**Share your drafted plan to gather feedback and inspire others.**

## **My Notes / Draft Space**

---

---

---

---

## **INTERACTIVE CHAT**

**"What is one project topic you are excited to try in your classroom?"**

# 📌 Workshop Wrap-Up & Digital Exit Ticket

## Key Takeaways

### Creative Learning

ScratchJr empowers young learners through creative play and computational thinking.

### Logical Sequencing

Block-based coding builds logical sequencing and problem-solving skills.

## Digital Exit Ticket

### Virtual Poll

How confident do you feel starting ScratchJr in your classroom next week?



**Super Confident!**



**Needs Practice**



### Emojis of Understanding

Share an emoji to rate your comfort level with today's **#ScratchJrWorkshop**

# Takeaway Resources & Support



## Additional Resources

-  **Large Scratch Blocks**  
Printable offline manipulative cards
-  **STEMinGames Coding Lessons Playlist**  
Video playlist and lesson materials
-  **ScratchJr Activities Guide**  
Official quick-start guides and posters
-  **ScratchJr. K-2 Lesson Plans**  
Curriculum guides and teacher resources

## Support & Contact



**Email Support**



**Official Website**

## Funding Acknowledgment

Supported by the National Science Foundation (NSF). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

# Additional Resources



- [Digital Storytelling with ScratchJr \(PreK-2\)](#)
- [Scratch Jr. Website](#)
- [Block Images, 6 Block Images](#)
- [Printable Blocks for Stickers](#)
- [Cross-Curricular Coding with Scratch Jr.](#)
- [Scratch Jr. Curriculum Ideas](#)
- More Game Ideas
  - [Foster Computational Thinking by Building Games in ScratchJr! CSTA 2023](#)
- [Score Keeper Soccer ScratchJr. Game](#)
- [Race the Grid ScratchJr. Game](#)

# Survey

---

## 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**.

## Thank You!

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

@CSTANJ

@CS4NJ

#CSPDWeek2026

#CompSciNJ

@csteachersorg



# New Jersey CSPDWeek

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