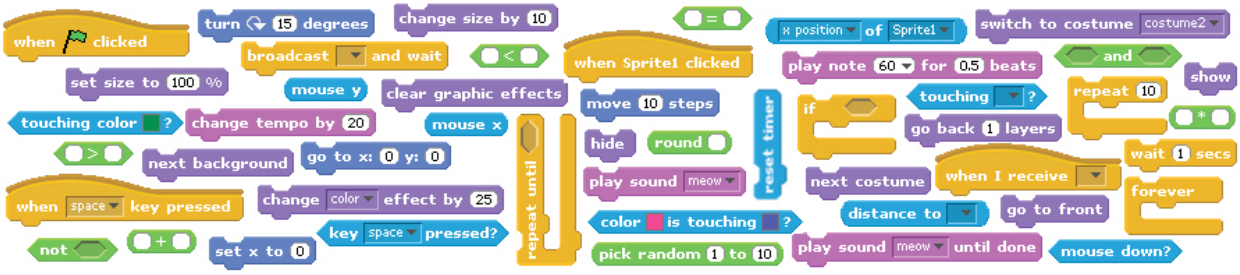


CREATIVE COMPUTING

a design-based introduction to computational thinking

DRAFT
Friday, September 23, 2011

Table of contents

Background

What is this guide?

This guide provides an introduction to *creative computing* with Scratch, using a *design-based learning* approach. It is organized as a series of twenty 60-minute sessions, and includes session plans, handouts, projects, and videos.

What is creative computing?

Creative computing is about *creativity*. Computer science and computing-related fields have long been perceived as being disconnected from young people's interests and values. Creative computing supports the development of personal connections to computing, by drawing upon creativity, imagination, and interests.

Creative computing is about *computing*. Many young people with access to computers participate as consumers, rather than designers or creators. Creative computing emphasizes the knowledge and practices that young people need to create the types of dynamic and interactive computational media that they enjoy in their daily lives.

Engaging in the creation of computational artifacts prepares young people for more than careers as computer scientists or as programmers. It supports young people's development as *computational thinkers* – individuals who can draw on computational concepts, practices, and perspectives in all aspects of their lives, across disciplines and contexts.

The activities in this guide are designed to explore computational thinking *concepts* (sequence, loops, parallelism, events, conditionals, operators, data), *practices* (working iteratively and incrementally, testing and debugging, reusing and remixing, abstracting and modularizing), and *perspectives* (expressing, connecting, questioning).

What is design-based learning?

Design-based learning is an approach that emphasizes *designing* (creating things, not just using or interacting with things), *personalizing* (creating things that are personally meaningful and relevant), *collaborating* (working with others on creations), and *reflecting* (reviewing and rethinking one's creative practices). As such, a design-based approach to learning is particularly well suited to creative computing, and forms the basis for the design of each session described in this guide.

Who is the guide for?

This guide is for any teacher who wants to support students' development of computational thinking through explorations with Scratch. Scratch is already being used by many educators

across a wide range of contexts, so we wrote the guide to be both subject-neutral and grade-neutral to accommodate different settings.

We rely on teachers to make the connections between the context of their learning environment and the activities that are described in this guide – and we hope to document some of these connections to share in future iterations of the guide.

What do I need in order to use this guide?

In addition to time, some important resources include:

- *computers with speakers (and, optionally, microphones and webcams):* for the computer-based design activities
- *projector or interactive whiteboard with speakers:* for sharing works-in-progress and for demonstrations
- *network connection:* for connecting to the Scratch and ScratchEd online communities
- *design notebooks (physical or digital):* for documenting, sketching, and brainstorming ideas and plans

How should I use this guide?

Feel free to design new activities and to remix the included activities. Of course, we'd love to learn about what you're doing, so we encourage you to document and share your experiences with us and with other educators via the ScratchEd community at http://scratched.media.mit.edu

Where did this guide come from?

This guide was written by Karen Brennan, with significant contributions from Michelle Chung and Jeff Hawson. Stephanie Gayle provided significant review and feedback.

The guide's content is based on four years of Scratch educator workshops, particularly the Google-funded 2009-2011 Creative Computing workshops that were co-hosted with Professor Mitchel Resnick and, more recently, NSF-funded ScratchEd workshops and meetups.

Many thanks to everyone who has made this guide possible, including the thousands of amazing workshop participants and ScratchEd online community members, and the members and friends of the ScratchEd and Scratch teams.

Structure

The 20 sessions presented in this guide are organized into five topics, as a way for students to explore different genres of creative expression and form, while developing familiarity and fluency with computational concepts and practices.

Topic	Description	# of Sessions
Introduction	Students are introduced to creative computing and Scratch, through sample projects and hands-on experiences.	2
Arts	Students explore the arts by creating projects that include elements of music, design, drawing, and dance. The computational concepts of sequence and loops, and the computational practices of being iterative and incremental are highlighted.	3
Stories	Students explore storytelling by creating projects that include characters, scenes, and narrative. The computational concepts of parallelism and events and the computational practices of reusing and remixing are highlighted.	3
Games	Students explore games by creating projects that define goals and rules. The computational concepts of conditionals, operators, and data, and the computational practices of testing and debugging are highlighted.	4
Final project	Students develop independent projects by defining a project to work on, collaborating with others to improve the project, and presenting the project and its development process. The computational practices of abstracting and modularizing are highlighted.	8

Daily Overview

Topic	Session #	Activity
Introduction	1	Planning: What is creative computing?
		Planning: Defining computational design processes
		Exploring: Something surprising
		Reflecting: Our discoveries
	2	Reflecting: Design notebook question
		Creating: About me
		Reflecting: My design process
Arts	3	Reflecting: Design notebook question
		Connecting: My favorite song
		Exploring: Programmed to dance
		Reflecting: Step by step
	4	Reflecting: Design notebook question
		Creating: Dance party
		Reflecting: How did you do that?
	5	Reflecting: Design notebook question
		Creating: Open-ended designing (arts)
Stories	6	Reflecting: Design notebook question
		Connecting: Six-word story
		Exploring: Performing scripts
		Reflecting: All together now
	7	Reflecting: Design notebook question
		Connecting: Creature construction
		Creating: Pass-it-on
	8	Reflecting: Design notebook question
		Creating: Open-ended designing (stories)
Games	9	Reflecting: Design notebook question
		Exploring: Debug it!
		Reflecting: Comparing debugging strategies
	10	Reflecting: Design notebook question
		Connecting: Games brainstorm
		Creating: A-maze-ing
	11	Reflecting: Design notebook question
		Creating: Maze extensions
		Reflecting: Here's what I figured out
	12	Reflecting: Design notebook question
		Creating: Open-ended designing (games)
Final project	13	Reflecting: Design notebook question
		Planning: Preparing for the final project
	14	Reflecting: Design notebook question
		Exploring: Special-interest groups

		Creating: Open-ended designing
	15	Reflecting: Design notebook question
		Creating: Open-ended designing
	16	Reflecting: Design notebook question
		Exploring: Critique groups
		Creating: Open-ended designing
	17	Reflecting: Design notebook question
		Creating: Open-ended designing
	18	Reflecting: Design notebook question
		Creating: Open-ended designing
		Planning: Preparing for the final project reflection
	19	Reflecting: Design notebook question
		Creating: Open-ended designing
	20	Reflecting: Design notebook question
		Reflecting: Celebration and final project reflections

Each session plan contains the following elements:

- *session description:* a brief summary of the session's activities
- *objectives:* a list of things that students will be able to know, do, or feel through the session's activities
- *session activities summary:* an outline of the session
- *resources:* a list of (required and optional) session resources
- *session description:* a detailed description of the session activities, including duration and student/facilitator actions
  - each session description begins with a reflective design notebook question, something that students can get started on as soon as they arrive
  - the remainder of the session description consists of different types of activities, including *planning*, *connecting*, *exploring*, *creating*, and *reflecting* activities
- *notes:* each session plan ends with a few reflective notes – points of common confusion, explanations of approaches, or suggestions for alternative strategies

Computational Thinking Connections

The following tables summarize the computational thinking framework and define its constituent components.

Computational Concepts

Concept	Description
sequence	identifying a series of steps for a task
loops	running the same sequence multiple times
parallelism	making things happen at the same time
events	one thing causing another thing to happen
conditionals	making decisions based on conditions
operators	support for mathematical and logical expressions
data	storing, retrieving, and updating values

Computational Practices

Practice	Description
being iterative and incremental	developing a little bit, then trying it out, then developing some more
testing and debugging	making sure that things work – and finding and fixing mistakes
reusing and remixing	making something by building on what others – or you – have done
abstracting and modularizing	building something large by putting together collections of smaller parts

Computational Perspectives

Perspective	Description
expressing	realizing that computation is a medium of creation “I can create.”
connecting	recognizing the power of creating with and for others “I can do different things when I have access to others.”
questioning	feeling empowered to ask questions about the world “I can (use computation to) ask questions to make sense of (computational things in) the world.”

Although the computational thinking concepts, practices, and perspectives are engaged throughout the curriculum guide activities, there are particular activities in which they are engaged more explicitly. The following tables highlight the activities in which computational concepts and computational practices are:

- introduced (marked with an I),
- discussed (marked with a D), and
- explored (marked with an E)

Computational Concepts

Session #	Activity	sequence	loops	parallelism	events	conditionals	operators	data
1	*Planning: What is creative computing?*							
	Planning: Defining computational design processes							
	Exploring: Something surprising							
	Reflecting: Our discoveries							
2	*Reflecting: Design notebook question*							
	Creating: About me							
	Reflecting: My design process							
3	*Reflecting: Design notebook question*	D						
	Connecting: My favorite song							
	Exploring: Programmed to dance	I	I					
	Reflecting: Step by step	D	D					
4	*Reflecting: Design notebook question*							
	Creating: Dance party	E	E					
	Reflecting: How did you do that?	E	E					
5	*Reflecting: Design notebook question*							
	Creating: Open-ended designing (arts)	E	E					
6	*Reflecting: Design notebook question*							
	Connecting: Six-word story							
	Exploring: Performing scripts			I	I			
	Reflecting: All together now			D	D			
7	*Reflecting: Design notebook question*							
	Connecting: Creature construction							
	Creating: Pass-it-on	E	E	E	E			
8	*Reflecting: Design notebook question*							
	Creating: Open-ended designing (stories)	E	E	E	E			
9	*Reflecting: Design notebook question*							
	Exploring: Debug it!					I	I	

	Reflecting: Comparing debugging strategies					D	D	
10	*Reflecting: Design notebook question*							
	Connecting: Games brainstorm							
	Creating: A-maze-ing					D	D	
11	*Reflecting: Design notebook question*							
	Creating: Maze extensions	E	E	E	E	E	E	I/E
	Reflecting: Here's what I figured out					D	D	D
12	*Reflecting: Design notebook question*							D
	Creating: Open-ended designing (games)	E	E	E	E	E	E	E
13	*Reflecting: Design notebook question*							
	Planning: Preparing for the final project							
14	*Reflecting: Design notebook question*							
	Exploring: Special-interest groups	D	D	D	D	D	D	D
	Creating: Open-ended designing	E	E	E	E	E	E	E
15	*Reflecting: Design notebook question*							
	Creating: Open-ended designing	E	E	E	E	E	E	E
16	*Reflecting: Design notebook question*							
	Exploring: Critique groups							
	Creating: Open-ended designing	E	E	E	E	E	E	E
17	*Reflecting: Design notebook question*							
	Creating: Open-ended designing	E	E	E	E	E	E	E
18	*Reflecting: Design notebook question*							
	Creating: Open-ended designing	E	E	E	E	E	E	E
	Planning: Preparing for the final project reflection							
19	*Reflecting: Design notebook question*							
	Creating: Open-ended designing	E	E	E	E	E	E	E
20	*Reflecting: Design notebook question*							
	Reflecting: Celebration and final project reflections							

Computational Practices

Session #	Activity	being iterative and incremental	reusing and remixing	testing and debugging	abstracting and modularizing
1	*Planning: What is creative computing?*				
	Planning: Defining computational design processes	I	I	I	I
	Exploring: Something surprising	E			
	Reflecting: Our discoveries	D			
2	*Reflecting: Design notebook question*				
	Creating: About me	E			
	Reflecting: My design process	D			
3	*Reflecting: Design notebook question*				
	Connecting: My favorite song				
	Exploring: Programmed to dance	E			
	Reflecting: Step by step	D			
4	*Reflecting: Design notebook question*			D	
	Creating: Dance party	E			
	Reflecting: How did you do that?	D			
5	*Reflecting: Design notebook question*				
	Creating: Open-ended designing (arts)				
6	*Reflecting: Design notebook question*			D	
	Connecting: Six-word story				
	Exploring: Performing scripts				
	Reflecting: All together now				
7	*Reflecting: Design notebook question*		D		
	Connecting: Creature construction		E		
	Creating: Pass-it-on		E		
8	*Reflecting: Design notebook question*				
	Creating: Open-ended designing (stories)	E	E		
9	*Reflecting: Design notebook question*			D	
	Exploring: Debug it!			E	
	Reflecting: Comparing debugging strategies			E	
10	*Reflecting: Design notebook question*				
	Connecting: Games brainstorm				
	Creating: A-maze-ing	E	E	E	
11	*Reflecting: Design notebook question*		D		
	Creating: Maze extensions	E	E	E	E
	Reflecting: Here's what I figured out				
12	*Reflecting: Design notebook question*				D

	Creating: Open-ended designing (games)	E	E	E	E
13	*Reflecting: Design notebook question*		D		
	Planning: Preparing for the final project				E
14	*Reflecting: Design notebook question*				D
	Exploring: Special-interest groups				
	Creating: Open-ended designing	E	E	E	E
15	*Reflecting: Design notebook question*				D
	Creating: Open-ended designing	E	E	E	E
16	*Reflecting: Design notebook question*				D
	Exploring: Critique groups				
	Creating: Open-ended designing	E	E	E	E
17	*Reflecting: Design notebook question*				D
	Creating: Open-ended designing	E	E	E	E
18	*Reflecting: Design notebook question*				D
	Creating: Open-ended designing	E	E	E	E
	Planning: Preparing for the final project reflection				
19	*Reflecting: Design notebook question*				D
	Creating: Open-ended designing	E	E	E	E
20	*Reflecting: Design notebook question*				
	Reflecting: Celebration and final project reflections				

Computational Perspectives

Computational perspectives are not addressed explicitly in the curriculum guide, but are introduced and explored indirectly through discussion questions and design journal prompts.

Assessment

Our approach to assessment is process-oriented, with a focus on creating opportunities for students to talk about their own (and others') creations and creative practices. There are many forms of process-oriented data that could be collected and various strategies are suggested throughout the guide, such as:

- supporting conversations with and among students about their projects, recorded through audio, video, or text (such as the project planner handout from Session #13, the project critique handout from Session #16, or the project reflections handout from Session #18)
- examining portfolios of projects
- maintaining design journals

We view assessment as something that is done *with* students, to support their understanding of what they know and what they still want to know. Assessment can involve a variety of participants, including the creators, their peers, teacher, parents, and others.

Standards

The sessions and activities in this guide make connections to several different K-12 curriculum standards, including:

- *CSTA K-12 Computer Science Standards 2011*
  http://csta.acm.org/includes/Other/CSTAStandardsReview2011.pdf
  - Computational thinking – Algorithms (Levels 1A, 1B, 2)
  - Computational thinking – Problem solving (Levels 1A, 1B, 2)
  - Computational thinking – Abstraction (Levels 1B, 2)
  - Computational thinking – Connections (Levels 1B, 2)
  - Collaboration – Tools (Levels 1A, 1B, 2)
  - Collaboration – Endeavor (Levels 1A, 1B, 2, 3A)
  - Practice and programming – Learning (Levels 1A, 1B, 2)
  - Practice and programming – Tools for creation (Levels 1A, 1B, 2)
  - Practice and programming – Programming (Levels 1A, 1B, 2, 3A)
  - Practice and programming – Careers (Levels 1A, 1B, 2)
  - Community, Global, and Ethical Impacts – Responsible use (Levels 1A – 3B)

- *ISTE NETS Student Standards 2007*
  http://www.iste.org/standards/nets-for-students/nets-student-standards-2007.aspx
  - Creativity and Innovation – Students demonstrate creative thinking, construct knowledge, and develop innovative products and processes using technology. Students:
    - apply existing knowledge to generate new ideas, products, or processes.
    - create original works as a means of personal or group expression.
  - Communication and Collaboration – Students use digital media and environments to communicate and work collaboratively, including at a distance, to support individual learning and contribute to the learning of others. Students:
    - interact, collaborate, and publish with peers, experts, or others employing a variety of digital environments and media.
    - communicate information and ideas effectively to multiple audiences using a variety of media and formats.
    - contribute to project teams to produce original works or solve problems.
  - Research and Information Fluency – Students apply digital tools to gather, evaluate, and use information. Students:
    - plan strategies to guide inquiry.
    - locate, organize, analyze, evaluate, synthesize, and ethically use information from a variety of sources and media.
    - evaluate and select information sources and digital tools based on the appropriateness to specific tasks.
  - Critical Thinking, Problem Solving, and Decision Making – Students use critical thinking skills to plan and conduct research, manage projects, solve problems, and make informed decisions using appropriate digital tools and resources. Students:
    - identify and define authentic problems and significant questions for investigation.
    - plan and manage activities to develop a solution or complete a project.

- collect and analyze data to identify solutions and/or make informed decisions.
- use multiple processes and diverse perspectives to explore alternative solutions.

- Digital Citizenship – Students understand human, cultural, and societal issues related to technology and practice legal and ethical behavior. Students:
  - advocate and practice safe, legal, and responsible use of information and technology.
  - exhibit a positive attitude toward using technology that supports collaboration, learning, and productivity.
  - demonstrate personal responsibility for lifelong learning.
- Technology Operations and Concepts – Students demonstrate a sound understanding of technology concepts, systems, and operations. Students:
  - understand and use technology systems.
  - select and use applications effectively and productively.
  - troubleshoot systems and applications.

Sessions

Session #1

Session description
In this session, students are introduced to computational creation with the Scratch programming environment by viewing a collection of sample projects and engaging in an exploratory, hands-on experience.

Objectives
The students will:

- understand the concept of computational creation, in the context of Scratch
- be able to imagine possibilities for their own Scratch-based computational creation
- become familiar with resources that support their computational creation

Session activities summary

- Introduce the concept of computational creation and the Scratch environment
- Show sample Scratch projects
- Review design processes
- Explore the Scratch interface

Resources

- Scratch overview video (*optional*) http://vimeo.com/29457909
- Collection of sample projects
- Design notebooks (*may be digital*)
- Resources library items (*Scratch cards, etc.*)

Session description

~Min.	*Activities*

15	*Planning: What is creative computing?*
	• Ask students:
	  o What are the different ways you interact with computers?
	  o How many of those ways involve you *creating* with computers?
	• Explain that over the next several sessions they will be creating their own interactive computational media with Scratch.
	• Show a basic demo of Scratch, either through a live demo or through the Scratch overview video.
	  o You build projects by snapping blocks together, just as you can build things in the physical world by snapping LEGO bricks together.
	  o There are more than 100 blocks in 8 different categories.
	  o As a small example, let's make the cat do a dance.
	  o Start by dragging out the "move 10 steps" block from the "Motion" blocks palette to the scripting area. Every time you click on the block the cat moves a distance of 10. You can change the number to make the cat move a greater or smaller distance.

	○ From the "Sound" palette, drag out the "play drum" block. Click on the block to hear its drum sound. Drag and snap the "play drum" block below the "move" block. When you click on this stack of two blocks, the cat will move and then play the drum sound. ○ Copy this stack of blocks (either using the Duplicate toolbar item or by right-clicking the stack and selecting "duplicate") and snap the copy to the already-placed blocks. Change the second "move" block to -10 steps, so the cat moves backward. Every time the stack of four blocks is clicked, the cat does a little dance forward and back. ○ Go to the "Control" blocks palette and grab the "repeat" block. Wrap the "repeat" block around the other blocks in the scripting area. Now when you click on the stack, the cat dances forward and back 10 times. ○ Finally, drag the "when Sprite clicked" block and snap it to the top of the stack. Click on the cat (instead of the blocks stack) to make the cat dance. • Show the range of projects they will be able to create, by sharing some sample projects that students will find engaging and inspiring. The Scratch website (http://scratch.mit.edu) has many interesting examples.
15	*Planning: Defining the processes of computational design* • Introduce students to the other tools that they will have access to during their design activities: ○ Design notebook, for recording their ideas and plans, as well as for responding to the per-session design notebook question ○ Resource library, for accessing other forms of support, such as Scratch cards, or reminders of strategies for getting unstuck ○ Scratch website, for storing their projects and finding inspiration and help
10	*Exploring: Something surprising* • Give students 10 minutes to explore the Scratch interface in an open-ended way. One prompt is: "You have 10 minutes to make something surprising happen to a sprite." Students are encouraged to work together, ask each other for help, and share what they are figuring out during the 10 minutes.
20	*Reflecting: Our discoveries* • Ask for 3 or 4 volunteers to share with the entire group *one thing* that they discovered. • Optionally, after the volunteers have shared, offer several challenges to the students: ○ Did anyone figure out how to add sound? ○ Did anyone figure out how to change the background? ○ Did anyone figure out how to access the help screens for particular blocks?

Notes

A major goal of this session is to establish a culture of fearlessness, exploration, and peer collaboration. It is expected that students (and their teachers!) will not know everything ahead of time – and the environment becomes a space where everyone is learning together.

Session #2

Session description
In this session, students build on their initial explorations of the Scratch environment by creating an interactive project.

Objectives
The students will:

- become familiar with a wider range of Scratch blocks
- be able to create a Scratch project that is an interactive digital representation of their interests

Session activities summary

- Respond to design notebook question
- Create Scratch biography projects
- Share and discuss creations

Resources

- *About me* handout
- *About me* sample projects (*optional*)

Session description

~Min.	*Activities*
5	*Reflecting: Design notebook question* • What are three aspects of yourself that you could represent through images or sound?
40	*Creating: About me* • Introduce students to the concept of the interactive collage, a Scratch project that represents aspects of themselves through clickable sprites. Optionally, show a couple of different interactive "About me" projects. • Give students 35 minutes to work on their projects, with the "About me" handout available to provide guidance for blocks to experiment with.
15	*Reflecting: My design process* • Invite 2 or 3 students to share their "About me" projects and encourage others to ask questions about their design process: o What was your inspiration? o How did you do that? o What did you get stuck on? How did you get unstuck? o What are you most proud of? Why? o What might you want to do next? • Ask students to post their projects on the Scratch website. (*optional*)

Notes
Example projects can simultaneously inspire and intimidate, open the creative space and constrain it. Encourage a wide range of creations – diversity is great.

Session #3

Session description
In this session, students explore the concepts of instruction and sequence through the arts: music, design, drawing, and dance.

Objectives
The students will:

- learn to express a complex activity using a sequence of simple instructions

Session activities summary

- Respond to design notebook question
- Share favorite songs
- Express a sequence of dance moves using simple verbal instructions

Resources

- Dance videos
  http://vimeo.com/28612347
  http://vimeo.com/28612585
  http://vimeo.com/28612800
  http://vimeo.com/28612970

Session description

~Min.	*Activities*
5	*Reflecting: Design notebook question* • What are 5 situations where you use instructions? What are instructions good for?
10	*Connecting: My favorite song* • Explain that the next few sessions will explore computational creation within the genre of the arts – music, design, drawing, and dance. • Ask students to share one of their current favorite songs with the group.
20	*Exploring: Programmed to dance* • Ask for 8 volunteers – four people who don't mind being bossy and four people who don't mind being bossed. Create four bossy/bossed pairs. • For each bossy/bossed pair: o Have the bossed partner facing away from the display and the bossy partner (and the rest of the group) facing the display. o Show the video to the bossy partner and the group, but not to the bossed partner. o Ask the bossy partner to describe to their partner – using only words! – how to perform the sequence of dance moves shown in the video.
25	*Reflecting: Step by step* • After the four dances have been recreated, discuss the experience with the volunteers and the other students: o What was easy/difficult about being the bossy partner?

	○ What was easy/difficult about being the bossed partner? ○ What was easy/difficult about watching? ○ How does this activity relate to what we're doing with Scratch?

Notes
Like the two activities in this session, several of the activities in this guide will be computer-free. Stepping back from the computer can support fresh perspectives on and new understandings of computational concepts, practices, and perspectives.

Session #4

Session description
In this session, students explore computational creation within the genre of the arts by designing interactive dance party projects.

Objectives
The students will:
- be able to create a Scratch project that combines animation and music
- understand and practice incremental development

Session activities summary
- Respond to design notebook question
- Create Scratch dance party projects
- Share and discuss creations

Resources
- *Dance party* handout
- *Dance party* sample projects (*optional*)

Session description

~Min.	*Activities*
5	*Reflecting: Design notebook question* • What are two strategies that you use (or could use) when you get stuck while designing?

40	*Creating: Dance party*
	• Introduce students to the concept of a dance party, a Scratch project in which sprites get down with cool costumes and funky beats.
	• Demonstrate how to start a dance party, by adding a sprite with multiple costumes that responds by dancing when clicked. Ask students:
	  o What is the difference between a sprite and a sprite's costume?
	  o When might you want to use a sprite?
	  o When might you want to use a costume?
	• Encourage students to be incremental in their development, adding and testing small amounts of code at a time.
	• Have students work on their projects, with the "Dance party" handout available as a guide.

15	*Reflecting: How did you do that?* • Ask students to do a gallery walk of the dance party projects-in-progress. Encourage students to look at others' code and ask questions about unfamiliar code constructs. • Ask students to post their projects to the Scratch website. (*optional*)

Notes
The difference between sprites and costumes is often a source of confusion for Scratchers. The metaphor of actors wearing different costumes sometimes helps to clarify the difference.

Session #5

Session description
In this session, students have time to build on an already-started project or to start a new computational exploration within the genre of arts.

Objectives
The students will:

- develop greater fluency with computational concepts (i.e. sequence, loops, events) and practices (i.e. iterative and incremental development, testing and debugging, reusing and remixing, abstracting and modularizing) by working on a self-directed project

Session activities summary

- Respond to design notebook question
- Work on Scratch projects

Resources

- Projects from prior sessions
- *Arts starter projects* handouts
- *Arts starter* sample projects (optional)

Session description

~Min.	*Activities*
5	*Reflecting: Design notebook question* • Sketch an idea for an arts-themed project. What features does it have?
55	*Creating: Open-ended designing* • Explain to students that this session is an opportunity to go back to a project they started in a previous session or to start on a fresh idea. • Offer the arts starter projects handouts to (and/or have brainstorm session with) students who are looking for project ideas to work on, including: o *Square, circle: Create a project that includes an orange square and a purple circle.* o *Build-a-band: Create your own musical group by pairing sprites with sounds to make interactive instruments.* o *Automatic drawing: Create a self-generating drawing project.* • Partway through the session, encourage students to check in with their neighbor to share what they have been working on. • Ask students to post their projects on the Scratch website. (*optional*)

Notes
Open-ended design sessions provide an opportunity to check in with students who might need some additional attention or support.

Session #6

Session description
In this session, students explore the concepts of parallelism and events through performance and story.

Objectives
The students will:

- be able to explain what parallelism is and how it works in Scratch
- be able to explain what events are and how they work in Scratch

Session activities summary

- Respond to design notebook question
- Write six-word stories
- Perform parallel and event-driven activities

Resources

- Physical Scratch blocks (*optional*)
- Sticky notes

Session description

~Min.	*Activities*
5	*Reflecting: Design notebook question* • What was a challenge you overcame in your last project? What is something you still want to figure out?
10	*Connecting: Six-word story* • Explain that the next few sessions will explore computational creation within the genre of stories. • Ask students to compose six-word stories about some aspect of their lives on sticky notes. The six-word story format is attributed to Hemingway, who once said that his best story was "For sale: baby shoes, never worn." Share your own story or find others online as examples. • Post the six-word stories in a central location, to be viewed throughout the activities.
25	*Exploring: Performing scripts* • Ask for two volunteers. • Prompted by the facilitator, the two volunteers will act out a series of instructions (either by "programming" the volunteers through the Scratch interface or through the physical version of the Scratch blocks). The instructions highlight parallelism (things happening at the same time) and events (one thing causing another thing to happen): o Have one person do one thing (like walk across the room). o Have that person "reset." o Have that person do two things simultaneously (like walk across the room and talk). o Add the second person, by having the second person simultaneously (but independently) do a task, like talking.

	○ Have the second person do a dependent task, like responding to the first person instead of talking over.
20	*Reflecting: All together now* • After the five "scripts" have been performed, discuss the experience with the volunteers and the other students: ○ What were the different ways in which things were happening at the same time? ○ What are the mechanisms that enable parallelism in Scratch? ○ What are the different ways that actions were triggered? ○ What are the mechanisms for events in Scratch?

Notes

Several weighty ideas are explored in this light-hearted activity. First, the notion of reset is something Scratchers struggle with as they get started. You program *everything* in Scratch and if you want things to start in a particular location, with a particular orientation, etc., you are completely responsible for those setup steps. Second, there are multiple levels of parallelism with Scratch. A single sprite can do multiple things at once, and multiple sprites can also perform actions simultaneously. Finally, there are different approaches to coordinating action within/across sprites. Many beginners use a central event (like the green flag) and wait blocks to control timing – there is a lot of power and excitement in learning the "broadcast" and "when I receive" block pair.

Session #7

Session description
In this session, students explore computational creation within the genre of stories by designing collaborative narratives.

Objectives
The students will:
- understand the benefits of remixing while designing
- be able to create a Scratch project that tells a story by building on the work of others

Session activities summary
- Respond to design notebook question
- Create creature characters
- Collaboratively create Scratch story projects through remixing
- Share and discuss creations

Resources
- Blank paper (approximately 8.5" by 11"), folded into thirds
- Things to sketch with

Session description

~Min.	*Activities*
5	*Reflecting: Design notebook question* • What is remixing?
10	*Connecting: Creature construction* • Give each student a tri-folded sheet of blank paper. • Ask students to draw a "creature" in three parts. • First, each student has a minute to draw a "head" for their creature. They fold the paper over so that the head is hidden, with little prompts for where to continue the drawing. After the head is hidden, pass the creature to another student. • Next, each student has a minute to draw a "middle" for their creature, using the guides from the head, but without peeking! After the middles are hidden (and prompts drawn), pass the creatures. • Finally, each student has a minute to draw a "bottom" for their creature. Once the creatures are complete, unfold the papers to see the collaboratively constructed creatures.
45	*Creating: Pass-it-on* • Divide the group into pairs. • Introduce students to the concept of a pass-it-on-story, a Scratch project that is started by a pair of people, and then passed on to two other pairs to extend and reimagine. • Encourage students to start in whatever way they want – focusing on characters, scene, plot, or whatever element excites them. • Each pair has 10 minutes to work on their contribution to the collaborative project

	before the group rotates. • Leave time for students to return to the other projects they contributed to, to see how the projects developed. • Ask students to post their projects to the Scratch website. (*optional*)

Notes
Being able to read others' code is a valuable skill and is critical for being able to engage in the practices of reusing and remixing.

Session #8

Session description
In this session, students will have time to build on an already-started project or to start a new computational exploration within the genre of stories.

Objectives
The students will:

- develop greater fluency with computational concepts (i.e. parallelism, events) and practices (i.e. iterative and incremental development, testing and debugging, reusing and remixing, abstracting and modularizing) by working on a self-directed project

Session activities summary

- Respond to design notebook question
- Work on Scratch projects

Resources

- Projects from prior sessions
- *Stories starter projects* handouts
- *Stories starter* sample projects (optional)

Session description

~Min.	*Activities*
5	*Reflecting: Design notebook question* • How was working with someone else different from your prior experiences of designing your Scratch projects?
55	*Creating: Open-ended designing* • Explain to students that this session is an opportunity to go back to a project they started in a previous session or to start on a fresh idea. • Offer the starter projects handouts to (and/or have brainstorm session with) students who are looking for project ideas to work on, including: o *Conversation: Get two characters talking to each other. Use the say and wait blocks to coordinate the conversation.* o *Scenes: Use the broadcast and when I receive blocks to create a multi-scene story.* o *Slideshow: Create your own slideshow – a collection of background images accompanied by audio narration.* • Partway through the session, encourage students to check in with a neighbor to share what they have been working on. • Ask students to post their projects on the Scratch website. (*optional*)

Notes
This open-ended design session provides an opportunity to check in with students who might need some additional attention or support, particularly around the "broadcast" and "when I receive" blocks.

Session #9

Session description
In this session, students explore a range of concepts (including conditionals and operators) through the practices of testing and debugging.

Objectives
The students will:
- be able to explain testing and debugging practices
- develop a list of strategies for testing and debugging Scratch projects

Session activities summary
- Respond to design notebook question
- Debug several Scratch projects
- Design a debugging scenario
- Share and discuss debugging strategies

Resources
- *Debug it!* handouts

Session description

~Min.	*Activities*
5	*Reflecting: Design notebook question* • Look at your response to the session #4 design question. What new strategies have you learned for dealing with getting stuck?
35	*Exploring: Debug it!* • Divide the group into teams of four people. • Give each team the *Debug it!* handout set, which contains 5 programs to debug and a prompt to design a new debugging challenge.
20	*Reflecting: Comparing debugging strategies* • Gather the group to discuss and compare the various approaches to fixing the bugs. o *What was the problem?* o *How did you identify the problem?* o *How did you fix the problem?* o *Did others have alternative approaches to fixing the problem?* • Ask one or more of the groups to share their new debugging challenge with the group.

Notes
Testing and debugging is probably the most common activity of programmers. Things rarely work as planned, so developing a set of testing and debugging strategies will be beneficial to any computational creator.

Session #10

Session description
In this session, students explore computational creation within the genre of games by designing a maze.

Objectives
The students will:

- be able to identify some common design elements of games
- be able to use Scratch to create a maze game

Session activities summary

- Respond to design notebook question
- Brainstorm popular games
- Identify common game design elements
- Create a maze

Resources

- *Maze* handout

Session description

~Min.	*Activities*
5	*Reflecting: Design notebook question* • What is a game?

10	*Connecting: Games brainstorm*
	• In small groups, ask students to generate a list of games that they like.
	• After a few minutes, ask them about the list:
	  o What do the games have in common?
	  o What features of their design make them a *game*?

45	*Creating: A-maze-ing* • Have students follow along for a maze-creating activity, as described in the *Maze* handout. • Start by designing the maze layout, by drawing a maze-like background with single-colored walls and a different single-colored end marker. • Add a sprite that will navigate the maze. Keep this simple – a single-colored square will work fine. • Add the interactive navigation to the sprite, having it move up, down, right, and left with the arrow keys, using the "point in direction" block and the "move 10 steps" block. • Add the initial state, by having the sprite jump to the beginning of the maze using the "when green flag clicked" block and the "go to x y" block. • Have the sprite bounce off the maze walls using conditionals (the "if" block) and sensing (the "touching color" block). • Define the ending condition, using the "wait until" and "touching color" blocks.

Notes

There has been little direct instruction recommended so far in this guide. One of the goals of this guide is to demonstrate ways of engaging with computational creation, and direct instruction can certainly be included as part of the design of the learning environment.

Session #11

Session description
In this session, students explore the concepts of conditionals and data through common game mechanics.

Objectives
The students will:

- be able to describe what a variable is and why variables are useful

Session activities summary

- Respond to design notebook question
- Develop extensions to the maze projects
- Help others learn about variables using one of the extensions as an example

Resources

- *Maze extension* projects
  http://scratch.mit.edu/galleries/view/138300

Session description

~Min.	*Activities*
5	*Reflecting: Design notebook question* • What did you like about the maze project? How might you want to change it?
25	*Creating: Maze extensions* • Divide the group into teams of three people. • Assign each team one of the *Maze extension* projects to explore: o *Score: Demonstrates how to set and change a score. Receive 10 points every time the Scratch cat is clicked.* o *Timer: Demonstrates how to use a timer. Use the mouse to navigate the Scratch cat to Gobo.* o *Enemies: Demonstrates how to add an enemy. Avoid the crazy tennis ball using the up and down arrow keys.* o *Levels: Demonstrates how to change levels. Score increases by 1 every time the space bar is pressed. Level increases by 1 for every 10 points.* o *Rewards: Demonstrates how to collect items. Use the arrow keys to move the Scratch cat around to collect items for his quest.*
30	*Reflecting: Here's what I figured out* • After studying the extension projects, ask students from each extension project to teach the rest of the class what they learned. o *What was the project?* o *How could it be used to extend the maze?* o *How does it use variables?*

Notes

Variables are an important mathematical concept and also an important computational concept. Students are taught about variables in their math classes at school, but many students have a difficult time learning them. Games are one way to make the usefulness of variables more concrete.

Session #12

Session description
In this session, students will have time to build on an already-started project or to start a new computational exploration within the genre of games.

Objectives
The students will:

- develop greater fluency with computational concepts (i.e. conditionals, operators, data) and practices (i.e. iterative and incremental development, testing and debugging, reusing and remixing, abstracting and modularizing) by working on a self-directed project

Session activities summary

- Respond to design notebook question
- Work on Scratch projects

Resources

- Projects from prior sessions
- *Games starter projects* handouts
- *Games starter* sample projects (optional)

Session description

~Min.	*Activities*
5	*Reflecting: Design notebook question* • What is a variable? What is it good for?
55	*Creating: Open-ended designing* • Explain to students that this session is an opportunity to go back to a project they started in a previous session or to start on a fresh idea. • Offer the starter projects handouts to (and/or have brainstorm session with) students who are looking for projects ideas to work on, including: o *Collide: Help Scratch cat navigate a field of Gobos.* o *Catlibs: Create an interactive word game.* o *Scrolling: Create the foundation for a side-scrolling game.* • Partway through the session, encourage students to check in with a neighbor to share what they have been working on. • Ask students to post their projects on the Scratch website. (*optional*)

Notes
This open-ended design session provides an opportunity to check in with students who might need some additional attention or support, particularly around the concepts of conditionals (e.g. if), operators (e.g. arithmetic, logical), and data (e.g. variables, lists).

Session #13

Session description
In this session, students will work on outlining their final projects.

Objectives
The students will:
- identify an appropriately-scoped project to work on
- develop an outline of activities or tasks required to complete the project
- generate a preliminary list of resources required to complete the project

Session activities summary
- Respond to design notebook question
- Brainstorm final projects
- Review project planner elements
- Complete project planners
- Work on final project (if time permits)
- Collect project planners

Resources
- *Projects planner* handouts

Session description

~Min.	*Activities*
5	*Reflecting: Design notebook question* • What has been your favorite Scratch project to work on so far? What are three ideas for something you could work on next?
55	*Planning: Preparing for the final project* • Divide the group into teams of three or four people. • Give the teams 10 minutes to brainstorm possible final projects. • Gather the teams together and have each student share one idea for a final project that they might want to develop. • Distribute a project planner to each person. • Review the different planner elements (outline of tasks, list of resources, storyboards/wireframes). • Ask students to start filling out the project planners. • Students who already have a clear concept and plan are welcome to start working on their project design. • Collect the project planners at the end of the session to prepare for the special-interest groups in the next session.

Notes
The final project is an opportunity for students to pursue their interests and explore their already-developed capacities in a self-directed way. Taking some time at the start of the final project to explore ideas, identify tasks involved in completing the project, and list what is (and isn't)

already known can be very beneficial for successful project completion. Although planning is helpful, it shouldn't be all-consuming or the only way of doing things. Different students will want and need to plan and tinker to different extents – and different phases of the project will require different approaches. Multiple design and development styles should be encouraged and accommodated.

Session #14

Session description
In this session, students will break out into groups to develop capacities necessary for designing their projects and work on their independent projects.

Objectives
The students will:
- identify areas where they need support
- provide guidance and support to peers

Session activities summary
- Respond to design notebook question
- Review project planner elements
- Gather into special-interest groups
- Work on final project

Resources
- List of potential breakout groups, based on student needs and interests

Session description

~Min.	*Activities*
5	*Reflecting: Design notebook question* • What part of your project will you be working on today? What might you need help with in order to make progress?
25	*Exploring: Special-interest groups* • Before this session, generate a list of potential special-interest/breakout groups, based on the types of projects students are planning to create. • Post the list of topics for the group the view. • Ask students to sign up to participate in a group or to suggest other topics for special-interest groups.
30	*Creating: Open-ended designing* • Explain to students that the remainder of this session is time to work on their final projects. • Partway through the session, encourage students to check in with a neighbor to share what they have been working on. • Ask students to post their projects-in-progress on the Scratch website. (*optional*)

Notes
Having many students each exploring different paths poses an interesting challenge for a facilitator – how do you support a large number of people? Students can be enormously valuable in providing support and guidance to each other throughout all of the Scratch sessions, and particularly during the final project sessions. Giving young people opportunities to teach other

makes things easier for the facilitator, but can also significantly deepen creators' learning and understanding.

Session #15

Session description
In this session, students will work on their final projects.

Objectives
The students will:

- use computational concepts and practices to further develop a Scratch project of their choosing

Session activities summary

- Respond to design notebook question
- Review project planner elements
- Seek additional support as needed
- Work on final project

Resources

- Additional resources to support student projects

Session description

~Min.	*Activities*
5	*Reflecting: Design notebook question* • What part of your project will you be working on today? What might you need help with in order to make progress?
55	*Creating: Open-ended designing* • Explain to students that this session is time to work on their final projects. • Introduce and distribute additional support resources as needed. • Partway through the session, encourage students to check in with a neighbor to share what they have been working on. • Ask students to post their projects-in-progress on the Scratch website. (*optional*)

Notes
Challenges will emerge as project development proceeds. In addition to peer support, having a collection of readily-available support resources can help students continue to make progress. Finding sample projects on the Scratch website (http://scratch.mit.edu) can be a source of ideas, as can finding additional resources on ScratchEd (http://scratched.media.mit.edu/resources).

Session #16

Session description
In this session, students will work together in small critique groups to give each other preliminary feedback on their projects.

Objectives
The students will:
- test projects-in-progress
- formulate and share feedback for others

Session activities summary
- Respond to design notebook question
- Review project planner elements
- Gather into critique groups
- Work on final project

Resources
- *Project feedback* handouts
- Additional resources to support student projects

Session description

~Min.	*Activities*
5	*Reflecting: Design notebook question* • What aspects of your project could someone give you feedback about?
30	*Exploring: Critique groups* • Divide the group into teams of three people. • Distribute two *Project feedback* handouts to each person. • Review the different critique handout elements. • Ask students to spend eight minutes reviewing each project in the team and completing the critique handout for the project. • At the end of the critique groups, ask students to give the completed critique handouts to their project creators.
25	*Creating: Open-ended designing* • Explain to students that the remainder of this session is time to work on their final projects. • Partway through the session, encourage students to check in with a neighbor to share what they have been working on. • Ask students to post their projects-in-progress on the Scratch website. (*optional*)

Notes
Different people will provide different perspectives on the project-in-progress. Create opportunities for creators to get feedback from a variety of sources, including themselves!

Session #17

Session description
In this session, students will work on their final projects.

Objectives
The students will:

- use computational concepts and practices to further develop a Scratch project of their choosing

Session activities summary

- Respond to design notebook question
- Review project planner elements
- Seek additional support as needed
- Work on final project

Resources

- Additional resources to support student projects

Session description

~Min.	*Activities*
5	*Reflecting: Design notebook question* • What part of your project will you be working on today? What might you need help with in order to make progress?
55	*Creating: Open-ended designing* • Explain to students that this session is time to work on their final projects. • Introduce and distribute additional support resources as needed. • Partway through the session, encourage students to check in with a neighbor to share what they have been working on. • Ask students to post their projects-in-progress on the Scratch website. (*optional*)

Notes
All design activities are constrained – by time, by resources, by our own abilities at a given moment – and compromises may need to be made. The open-ended designing sessions are a great opportunity to have conversations with students about the essential elements of their projects. What are the most important aspects of the projects? What can reasonably be accomplished in the remaining time?

Session #18

Session description
In this session, students will work on their final projects and prepare for the final project reflection.

Objectives
The students will:

- use computational concepts and practices to further develop a Scratch project of their choosing
- think about how to share their project process

Session activities summary

- Respond to design notebook question
- Review project planner elements
- Seek additional support as needed
- Work on final project
- Prepare for the final project reflection

Resources

- Additional resources to support student projects
- *My final project reflections* handout

Session description

~Min.	*Activities*
5	*Reflecting: Design notebook question* • What part of your project will you be working on today? What might you need help with in order to make progress?
40	*Creating: Open-ended designing* • Explain to students that this session is time to work on their final projects. • Introduce and distribute additional support resources as needed. • Partway through the session, encourage students to check in with a neighbor to share what they have been working on. • Ask students to post their projects-in-progress on the Scratch website. (*optional*)
15	*Planning: Preparing for the final project reflection* • Remind students that they will be sharing their projects with each other (and possibly guests). • Share the *My final project reflection*s handout with students and discuss the *what?, so what?, now what?* framework as a way for them to present their experiences to others.

Notes
Sharing is an important part of the creative process. It is an opportunity to acknowledge the hard work that has taken place – and to reflect on the experience.

Session #19

Session description
In this session, students will work on their final projects.

Objectives
The students will:

- use computational concepts and practices to further develop a Scratch project of their choosing

Session activities summary

- Respond to design notebook question
- Review project planner elements
- Seek additional support as needed
- Work on final project

Resources

- Additional resources to support student projects
- *My final project reflections* handout

Session description

~Min.	*Activities*
5	*Reflecting: Design notebook question* • What part of your project will you be working on today? What might you need help with in order to make progress?
55	*Creating: Open-ended designing* • Explain to students that this session is time to work on their final projects. • Introduce and distribute additional support resources as needed. • Partway through the session, encourage students to check in with a neighbor to share what they have been working on. • Ask students to post their projects-in-progress on the Scratch website. (*optional*)

Notes
Students may be feeling anxious or stressed about completing their projects. This is an opportunity to remind them that: (1) this experience is just a waypoint on their paths as computational creators, and (2) some types of stress can be good, helping us to focus on our goals and get things done!

Session #20

Session description
In this session, students will share their final projects and reflect on their project development process and computational creation experiences.

Objectives
The students will:
- present their design work to others

Session activities summary
- Respond to design notebook question
- Share final projects
- Discuss experiences with computational creation

Resources
- Snacks (*optional*)

Session description

~Min.	*Activities*
5	*Reflecting: Design notebook question* • Look through your design notebook. What types of notes did you take? Which notes were most helpful?
55	*Reflecting: Celebration and final project reflections* • Invite students to share their work with others. The sharing can take place in a variety of ways: individuals presenting to the entire group, concurrent subsets of students presenting, live demos, accessing projects from the web, etc. • Make student progress visible by having design notebooks and prior projects available. • Create a celebratory mood in the space with documentation, guests, music, decorations, and/or snacks.

Notes
The portfolios of projects, design journals, final project feedback handouts, and final project reflections handouts are only a few (of many different possible) forms of assessment that can be conducted by and with different stakeholders, including the creators, their peers, teacher, parents, and others. Why did you select Scratch as a tool for you and your students to work with? What artifacts could you or did you collect that align with your goals? What questions would you like to explore with your students?

Appendix: Links

A summary of links to curriculum guide resources:

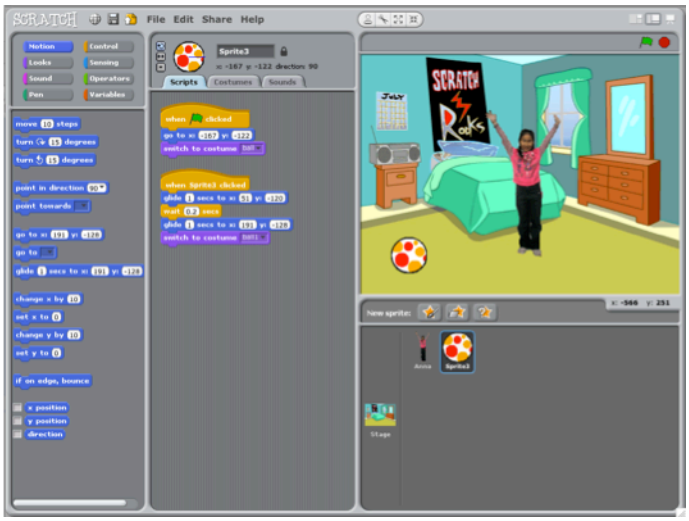
Type	Description	Link
Video	Intro to Scratch video	http://vimeo.com/29457909
Video	Dance #1	http://vimeo.com/28612347
Video	Dance #2	http://vimeo.com/28612585
Video	Dance #3	http://vimeo.com/28612800
Video	Dance #4	http://vimeo.com/28612970
Project	About me	http://scratch.mit.edu/projects/ScratchEdTeam/2041660
Project	Dance party	http://scratch.mit.edu/projects/ScratchEdTeam/2041671
Project	Square, circle	http://scratch.mit.edu/projects/ScratchEdTeam/2042075
Project	Build-a-band	http://scratch.mit.edu/projects/ScratchEdTeam/2042276
Project	Automatic drawing	http://scratch.mit.edu/projects/ScratchEdTeam/2042282
Project	Conversation	http://scratch.mit.edu/projects/ScratchEdTeam/2042349
Project	Scenes	http://scratch.mit.edu/projects/ScratchEdTeam/2042673
Project	Slideshow	http://scratch.mit.edu/projects/ScratchEdTeam/2042695
Project	Debug it #1	http://scratch.mit.edu/projects/ScratchEdTeam/2042697
Project	Debug it #2	http://scratch.mit.edu/projects/ScratchEdTeam/2042703
Project	Debug it #3	http://scratch.mit.edu/projects/ScratchEdTeam/2042706
Project	Debug it #4	http://scratch.mit.edu/projects/ScratchEdTeam/2042712
Project	Debug it #5	http://scratch.mit.edu/projects/ScratchEdTeam/2042724
Project	Maze	http://scratch.mit.edu/projects/ScratchEdTeam/2042736
Project	Maze Extension: Score	http://scratch.mit.edu/projects/ScratchEdTeam/2042755
Project	Maze Extension: Timer	http://scratch.mit.edu/projects/ScratchEdTeam/2042761
Project	Maze Extension: Enemies	http://scratch.mit.edu/projects/ScratchEdTeam/2042763
Project	Maze Extension: Levels	http://scratch.mit.edu/projects/ScratchEdTeam/2042764
Project	Maze Extension: Rewards	http://scratch.mit.edu/projects/ScratchEdTeam/2042770
Project	Collide	http://scratch.mit.edu/projects/ScratchEdTeam/2042778
Project	Catlibs	http://scratch.mit.edu/projects/ScratchEdTeam/2042781
Project	Scrolling	http://scratch.mit.edu/projects/ScratchEdTeam/2042861
Gallery	Sample Scratch projects	http://scratch.mit.edu/galleries/view/137903
Gallery	Sample arts projects	http://scratch.mit.edu/galleries/view/138296
Gallery	Sample stories projects	http://scratch.mit.edu/galleries/view/138297
Gallery	Sample games projects	http://scratch.mit.edu/galleries/view/138298
Gallery	Maze extensions	http://scratch.mit.edu/galleries/view/138300
Gallery	About me sample projects	http://scratch.mit.edu/galleries/view/138381
Gallery	Dance party sample projects	http://scratch.mit.edu/galleries/view/138382
Gallery	Maze sample projects	http://scratch.mit.edu/galleries/view/138299

Appendix: Handouts

This appendix includes the following handouts:

Session #	Handout
2	About me
4	Dance party
5	Useful blocks for arts-themed projects
	Square, circle
	Build-a-band
	Automatic drawing
8	Useful blocks for stories-themed projects
	Conversation
	Scenes
	Slideshow
9	Debug it!
10	Maze
12	Useful blocks for games-themed projects
	Collide
	Catlibs
	Scrolling
13	Plans for my final project
	Sketches of my final project
16	Project feedback
18	My project reflections

ABOUT ME



How can you combine interesting images and sounds to make an interactive collage about yourself?

STEP BY STEP...

1. Add a sprite

paint your own sprite

choose a downloaded or library sprite

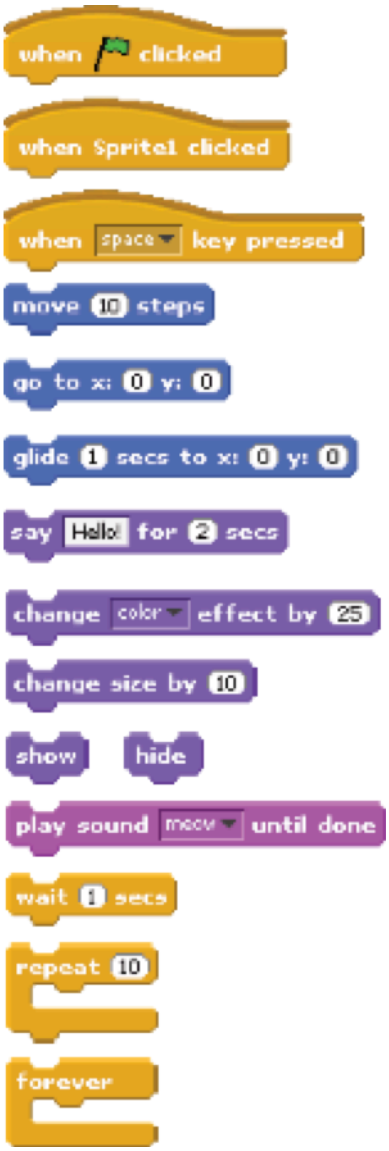
get a surprise sprite

2. Make it interactive

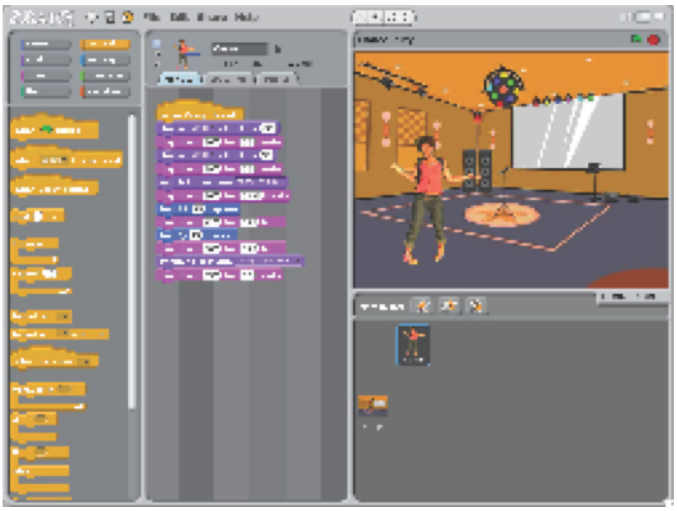
make your sprite interactive by adding scripts that have the sprite respond to clicks, key presses, and more

3. Repeat!

BLOCKS TO PLAY WITH...



DANCE PARTY



Create your own interactive dance party where sprites get down with cool costumes and funky beats.

STEP BY STEP...

1. Add a sprite

paint your own sprite

choose a downloaded or library sprite

get a surprise sprite

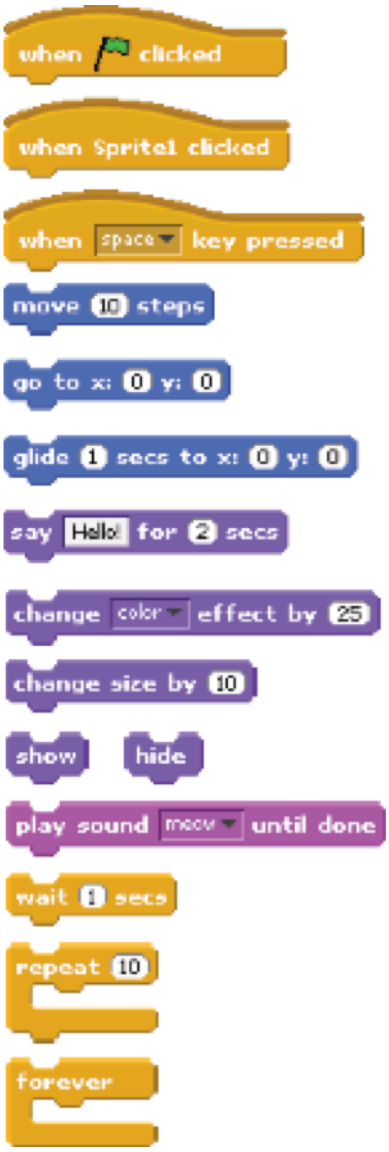
2. Make it interactive

snap blocks together to make your sprite dance

make your sprite interactive by adding scripts that have the sprite respond to clicks, key presses, and more

3. Repeat!

BLOCKS TO PLAY WITH...



ARTS

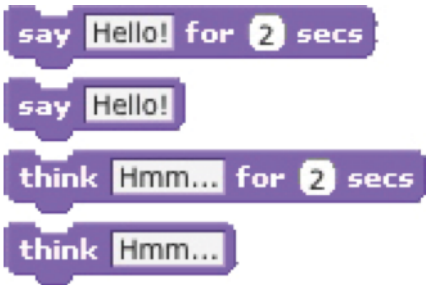
Here are some blocks that can be useful in arts-themed projects.

WAIT

Insert a pause

SAY/THINK

Have a speech or thought bubble appear over a sprite



SOUNDS

Play recorded and synthesized audio

VISIBILITY

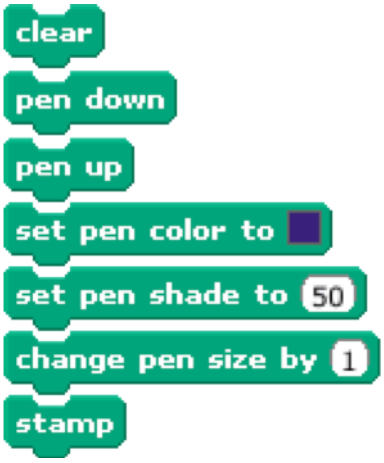
Make a sprite appear or disappear

LOOP

Repeat stacks of blocks

PEN

Leave visual lines and sprite stamps on the stage



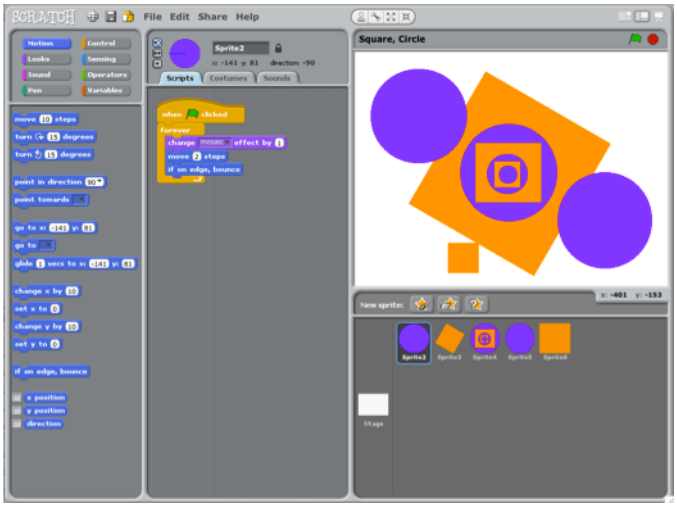
RANDOM

Get a computer-generated number from a specified range

KEY PRESS

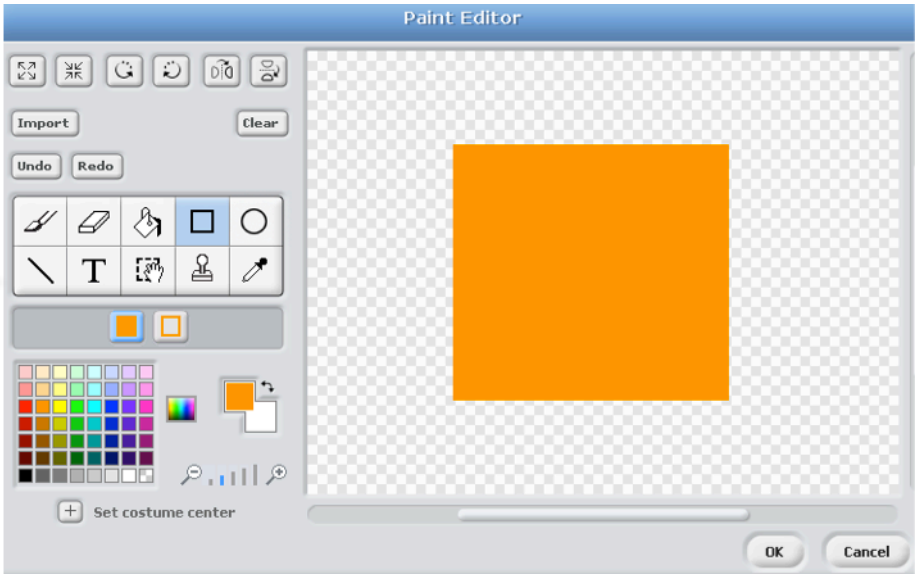
Make a sprite respond when different keys are pressed

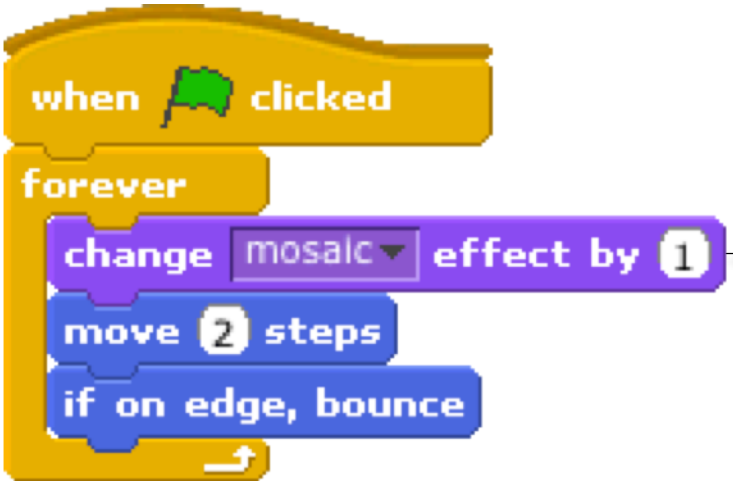
SQUARE, CIRCLE



What project can you create that includes an orange square and a purple circle?

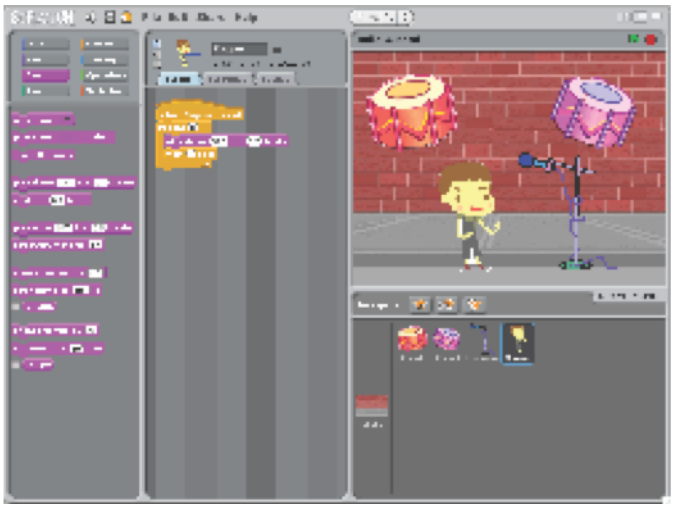
use the paint editor to design sprites that incorporate an orange square and a purple circle



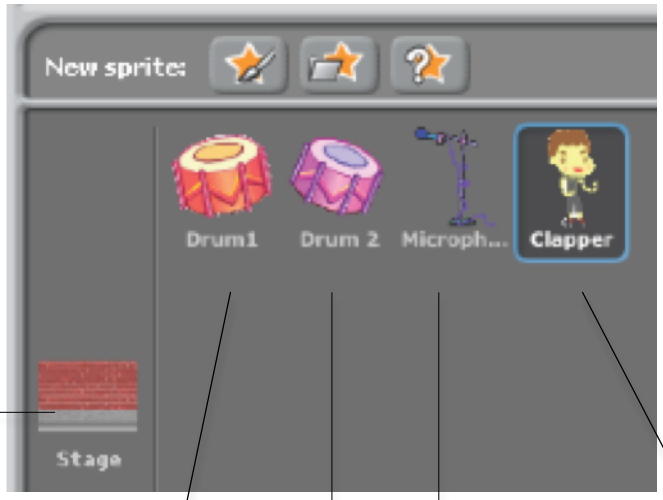


experiment with different motion and looks blocks to bring your sprites to life

BUILD-A-BAND



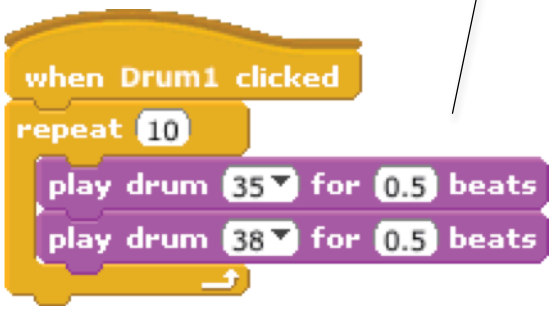
Create your own musical group by pairing sprites with sounds to make interactive instruments.

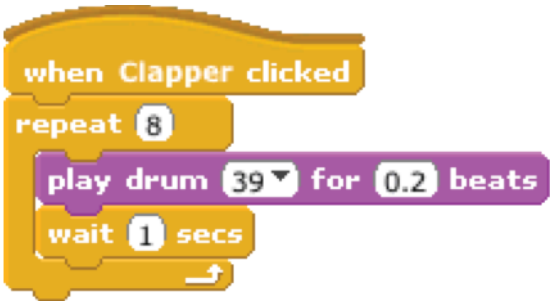


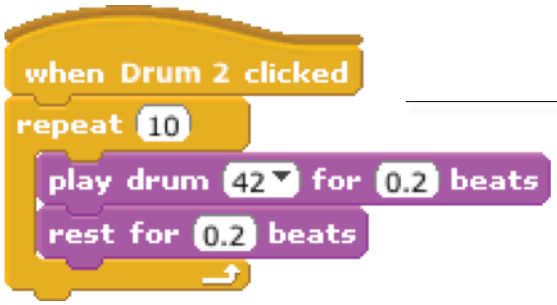
use the repeat block to make a sound play more than once

experiment with different music and beats, or import your own sounds

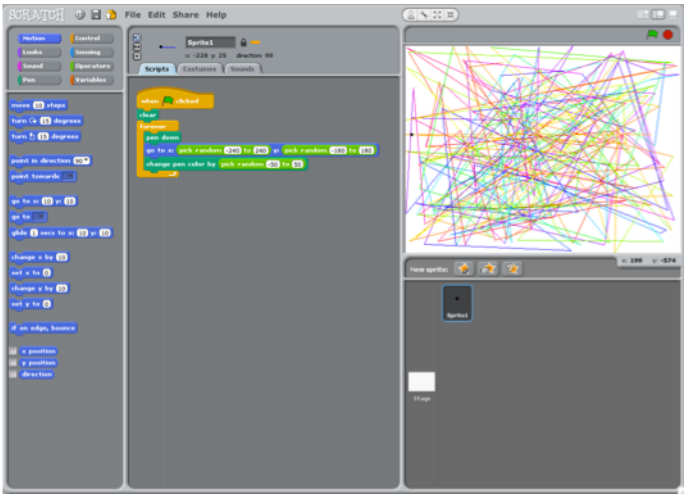
no scripts





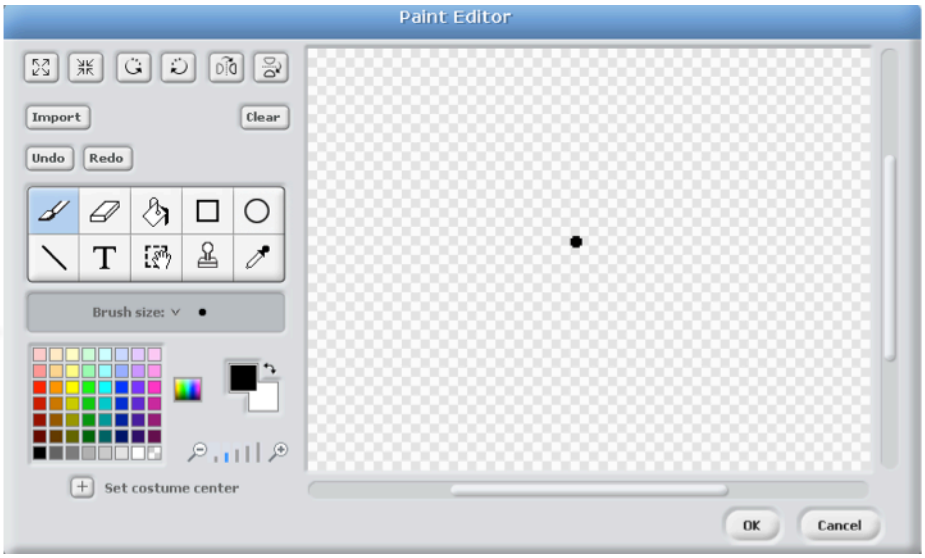


AUTOMATIC DRAWING



Make a project that draws something on its own.

create a sprite that is just a small dot



STORIES

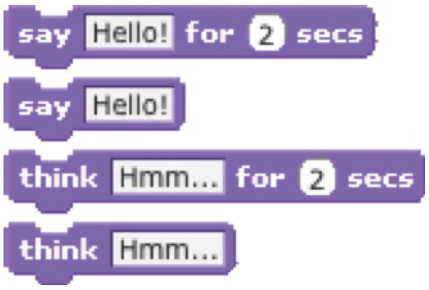
Here are some blocks that can be useful in story-themed projects.

WAIT

Insert a pause

SAY/THINK

Have a speech or thought bubble appear over a sprite



SOUNDS

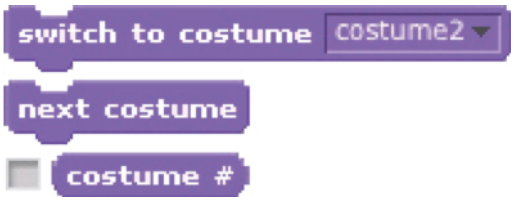
Play recorded audio

VISIBILITY

Make a sprite appear or disappear

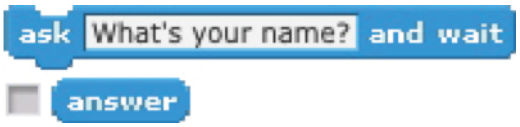
COSTUMES

Change the appearance of your sprite



ASK

Get input to use in a project



STRINGS

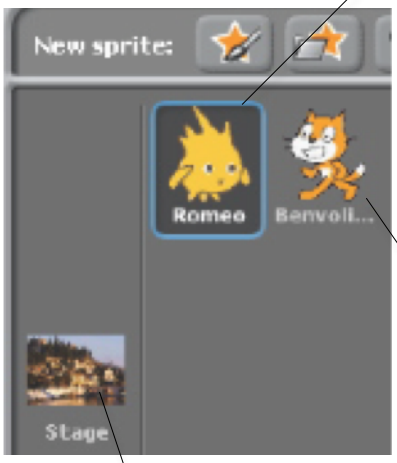
Test, access, and change words and sentences

COORDINATE

Synchronize actions between and within sprites

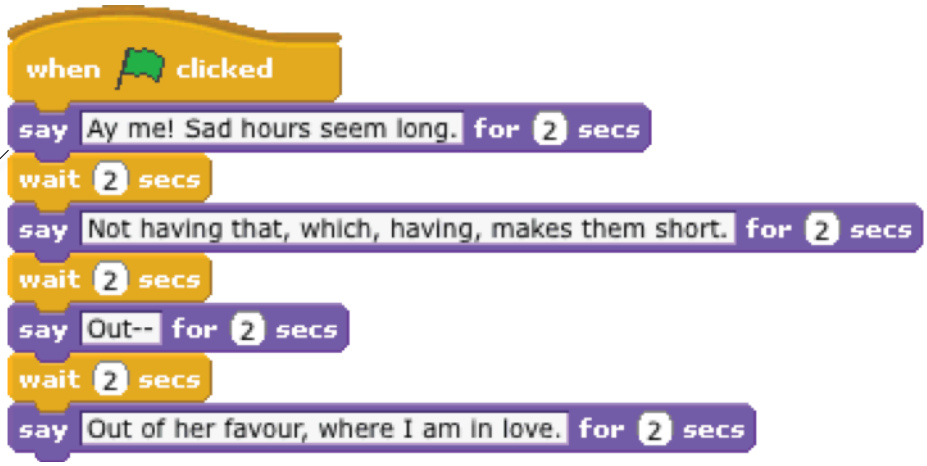
CONVERSATION

Get two characters talking to each other.
Use the **say** and **wait** blocks to coordinate the conversation.

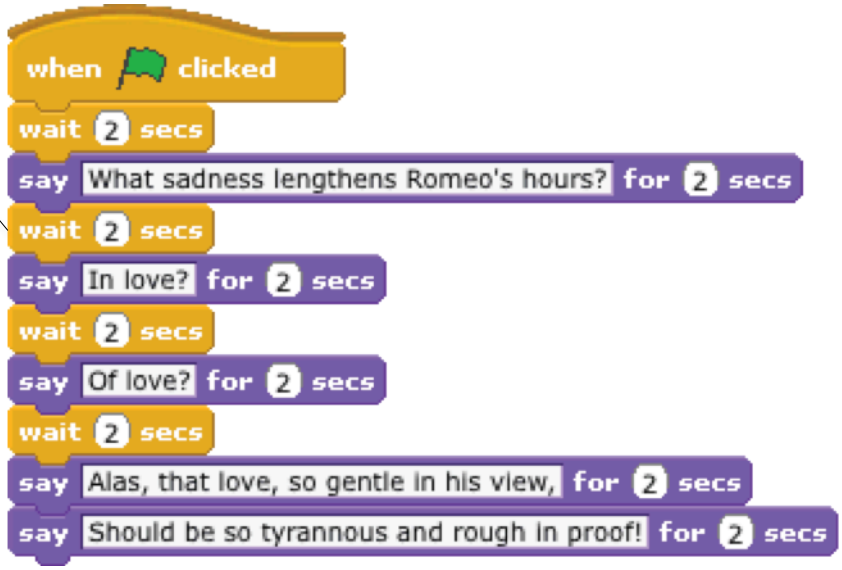


no scripts
add a background

1 script



1 script



SCENES

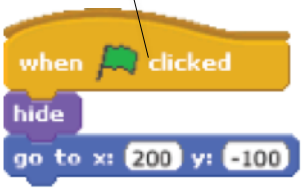
Use the **broadcast** and **when I receive** blocks to create a multi-scene story.

add 3 background images

4 scripts

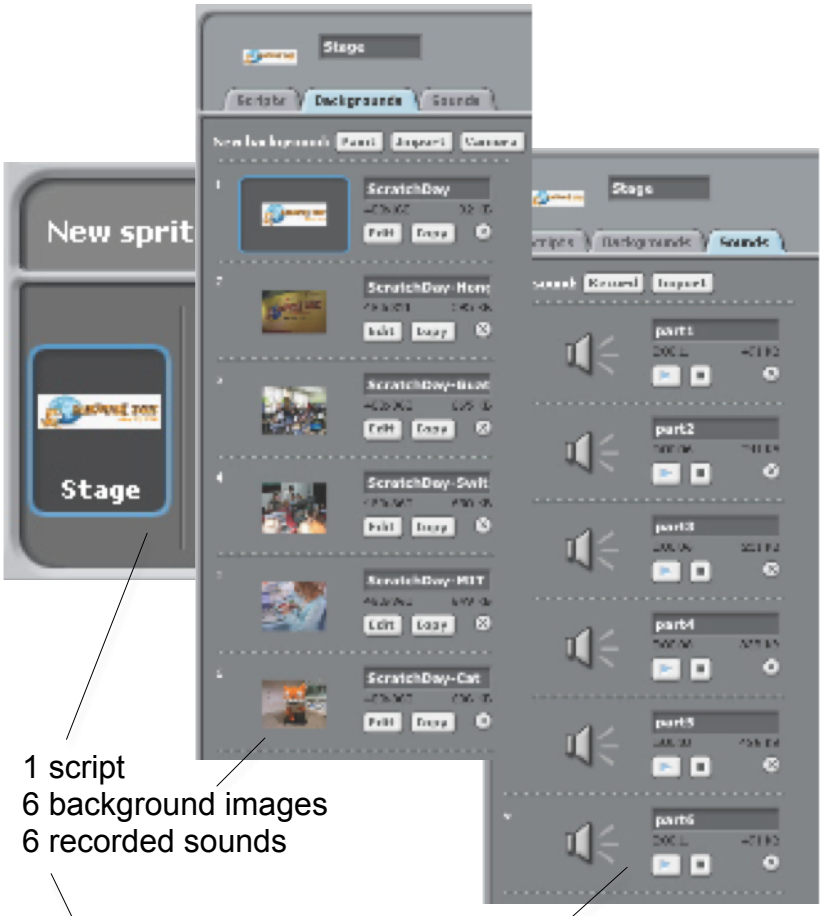
4 scripts

2 scripts

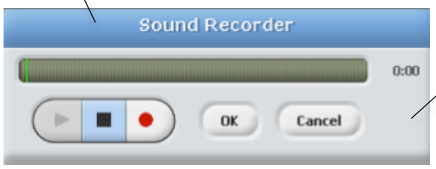


SLIDESHOW

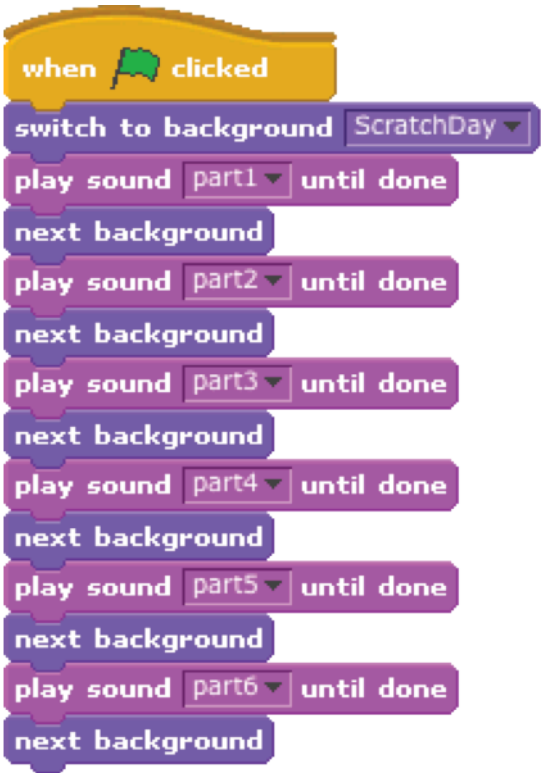
Create your own slideshow – a collection of background images accompanied by audio narration.



1 script
6 background images
6 recorded sounds

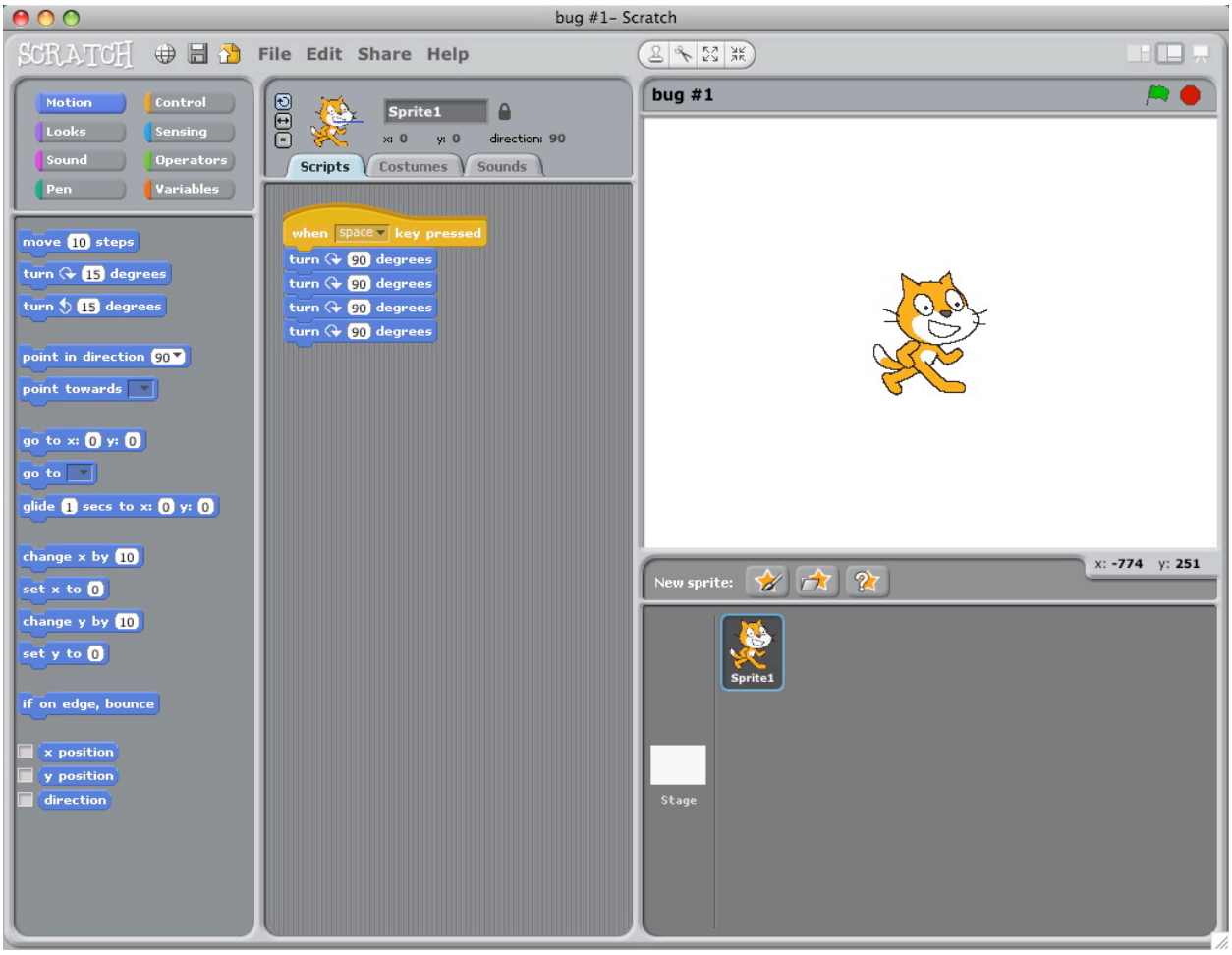


record your narration right in Scratch with the sound recorder



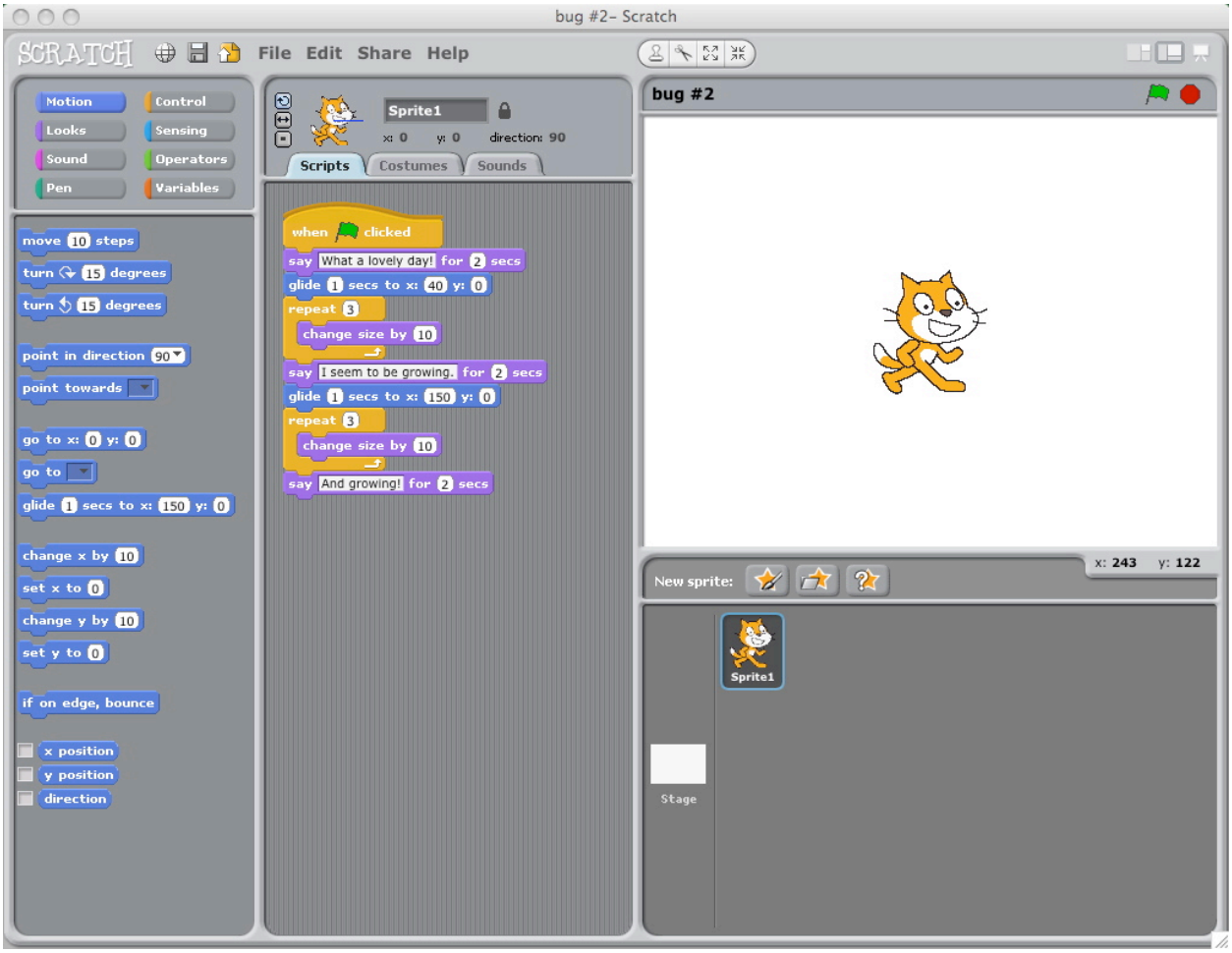
BUG #1

Farooq wants his cat to rotate when he presses the space bar. But the cat isn't moving! What's going on?



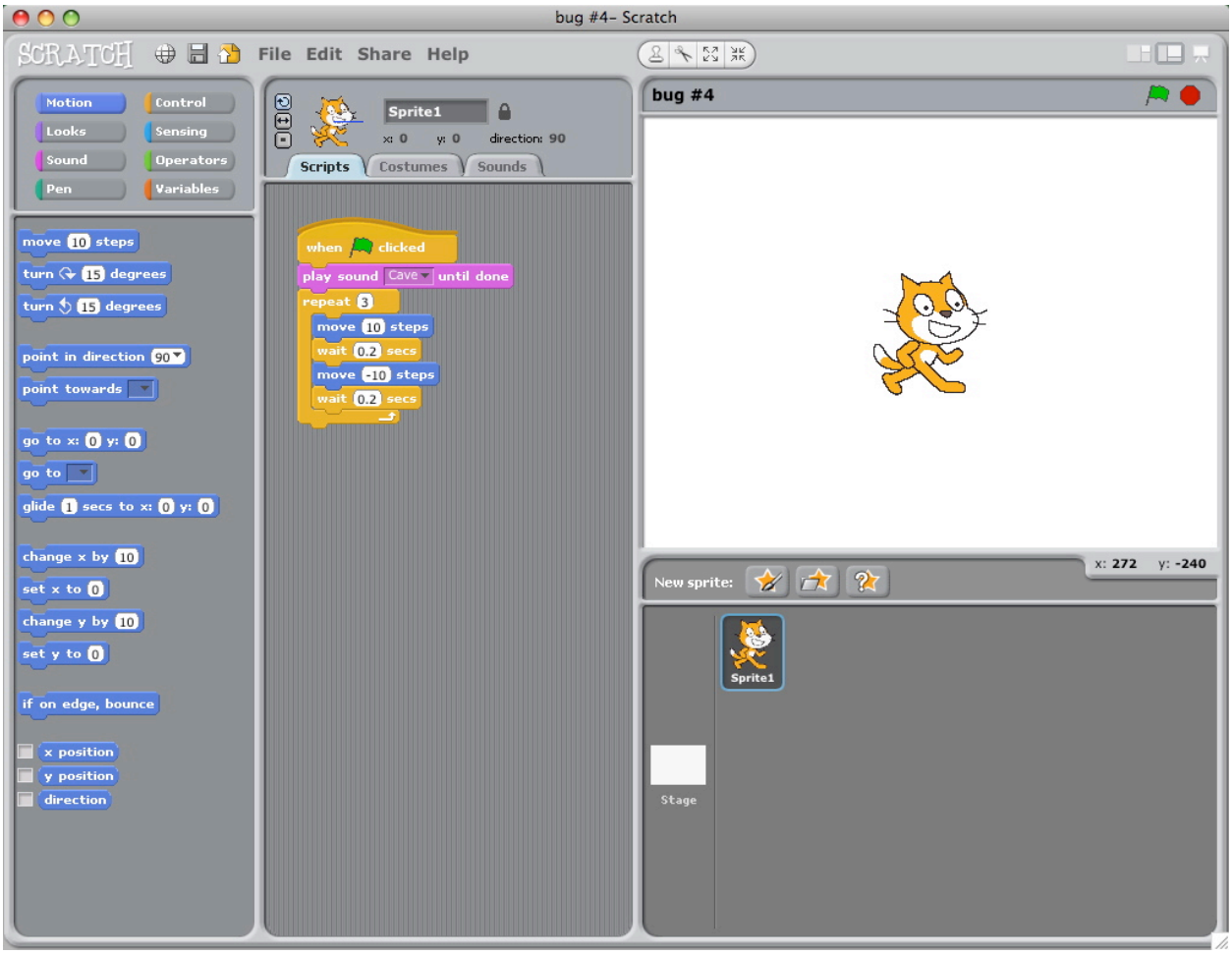
BUG #2

Michelle wants the cat to start in the middle of the stage, then move across the stage and grow. It works the first time she clicks the green flag – but not when she clicks it again! What's going on?



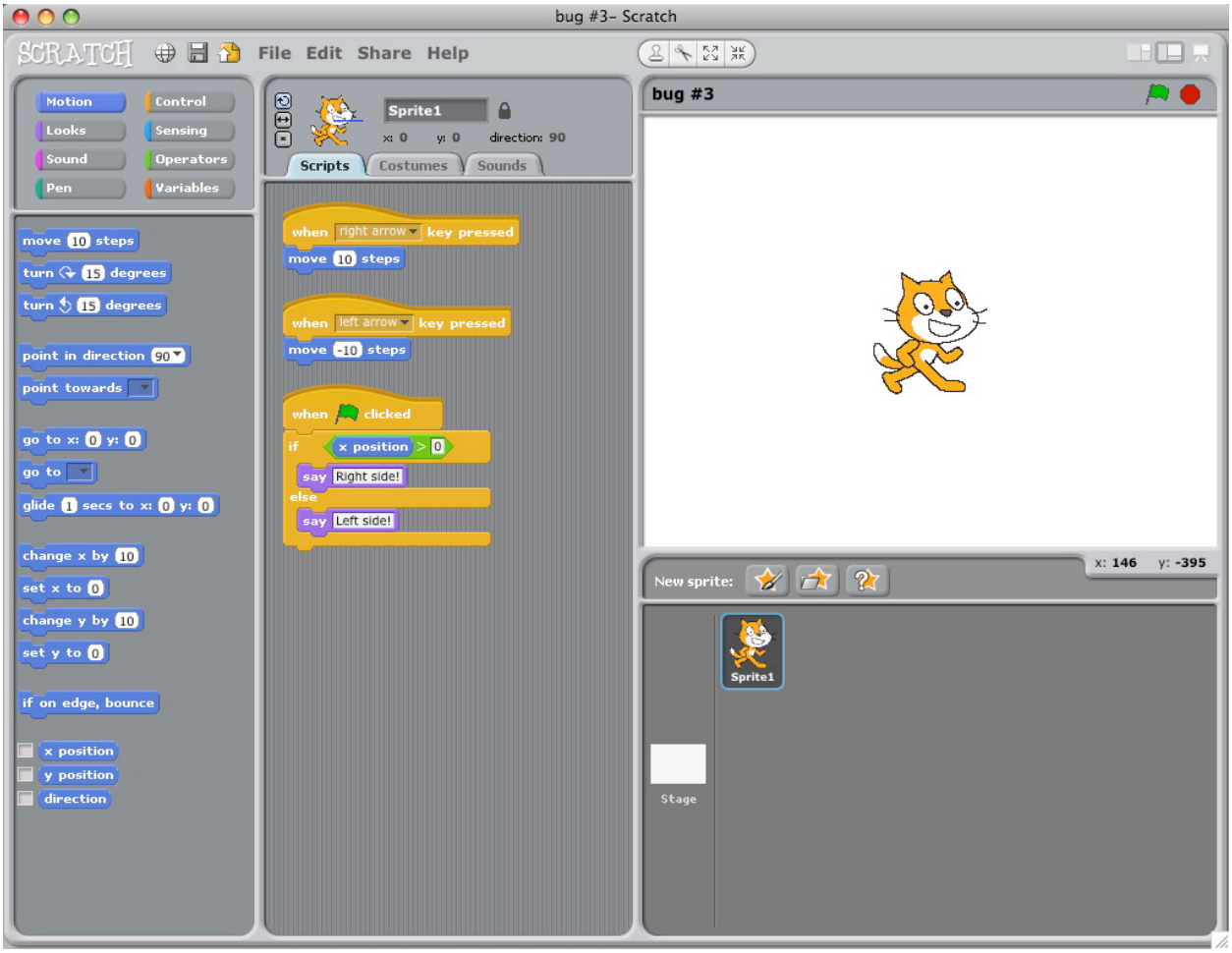
BUG #3

Alex wants his cat to dance to some music. But the cat is dancing after the music is over! What's going on?



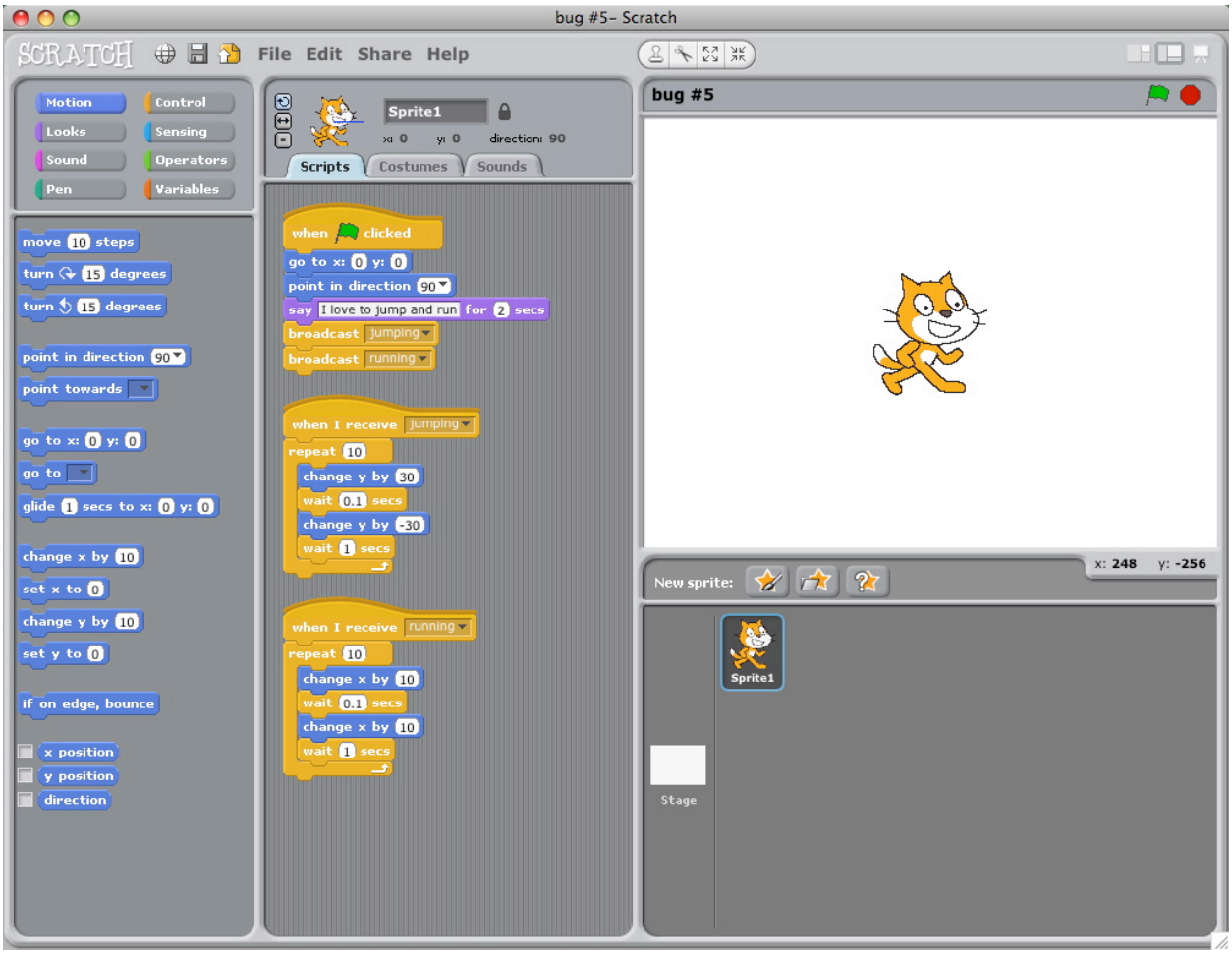
BUG #4

Praneetha wants to control the cat's x-position with the keyboard: right arrow moves the cat right, left arrow moves the cat left. She also wants the cat to say if it's on the right side or the left side, depending on its x-position. The cat's moving, but not saying its position correctly! What's going on?



BUG #5

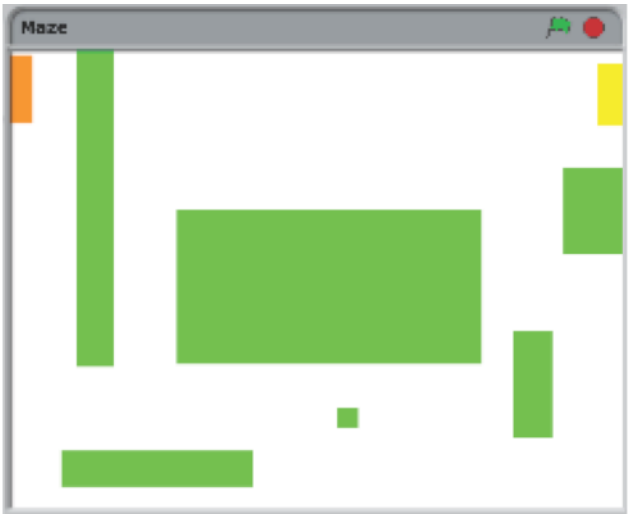
Nobuyuki wants his cat to jump up and down, and then move across the screen. But the cat's hopping across the screen! What's going on?



BUG #6

Design your very own puzzling Scratch program to debug.

MAZE



Create a game where you get a sprite from the start to the end of a maze without touching the walls.

7 scripts total

no scripts,
draw a maze-like background with colored walls and a differently colored end marker

move the sprite around

have the sprite bounce off the green walls

players wins when sprite reaches the yellow end marker

GAMES

Here are some blocks that can be useful in games.

TOUCHING

See if two sprites are touching or if a sprite is touching a color

VISIBILITY

Make a sprite appear or disappear

RANDOM

Get a computer-generated number from within a specified range

TIMING

Have the computer keep track of time for you

STRINGS

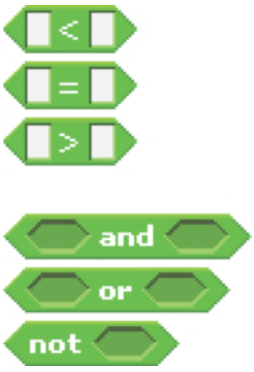
Test, access, and change words and sentences

VARIABLES

Store a number or string in a container to access later

COMPARE

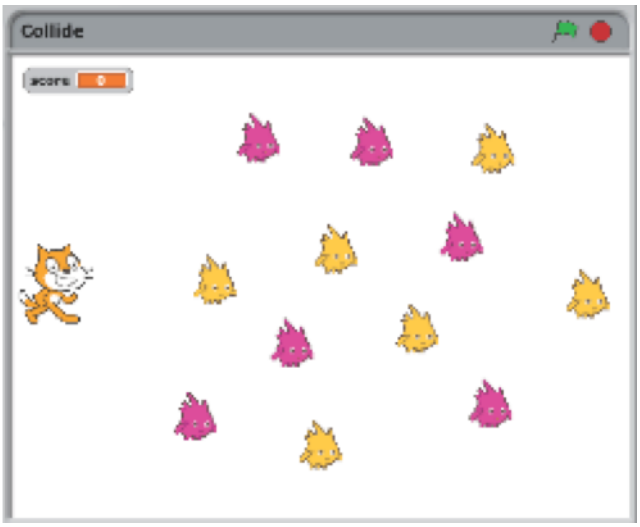
Compare values to help make decisions within your game



KEY PRESS

Make a sprite respond when different keys are pressed

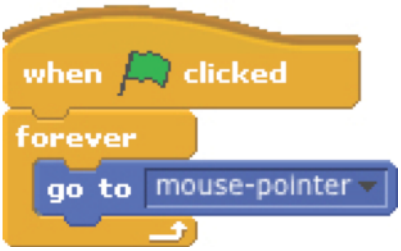
COLLIDE



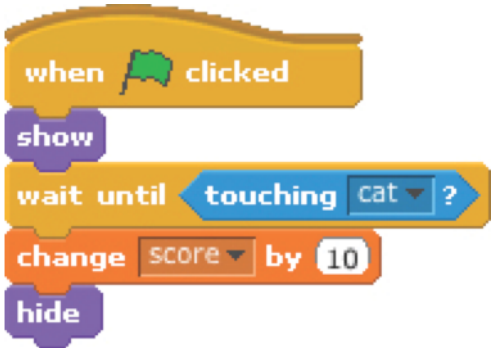
Help the cat navigate a field of Gobos. Collect yellow gobos to earn points, avoid pink gobos to avoid losing points.

no scripts

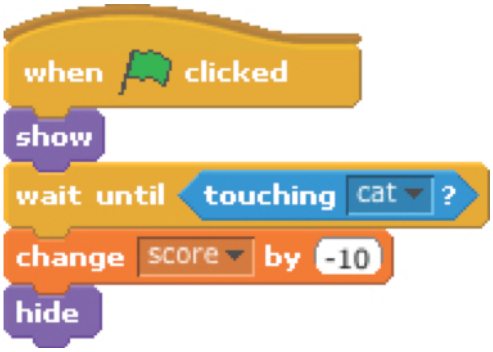
reset the cat's position and the score



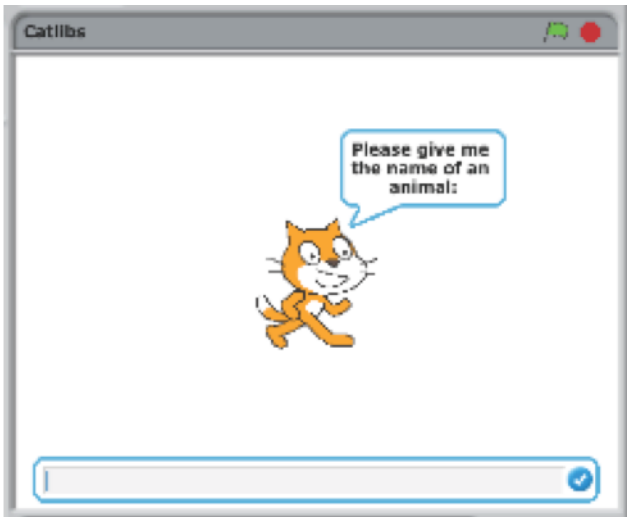
when the cat collides with a yellow gobo, the gobo disappears and the score increases by 10



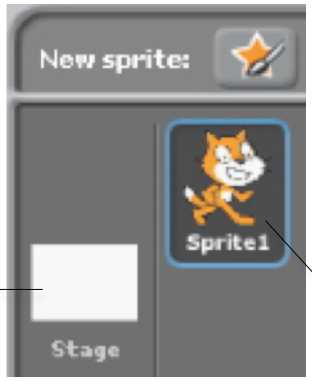
when the cat collides with a pink gobo, the gobo disappears and the score decreases by 10



CATLIBS

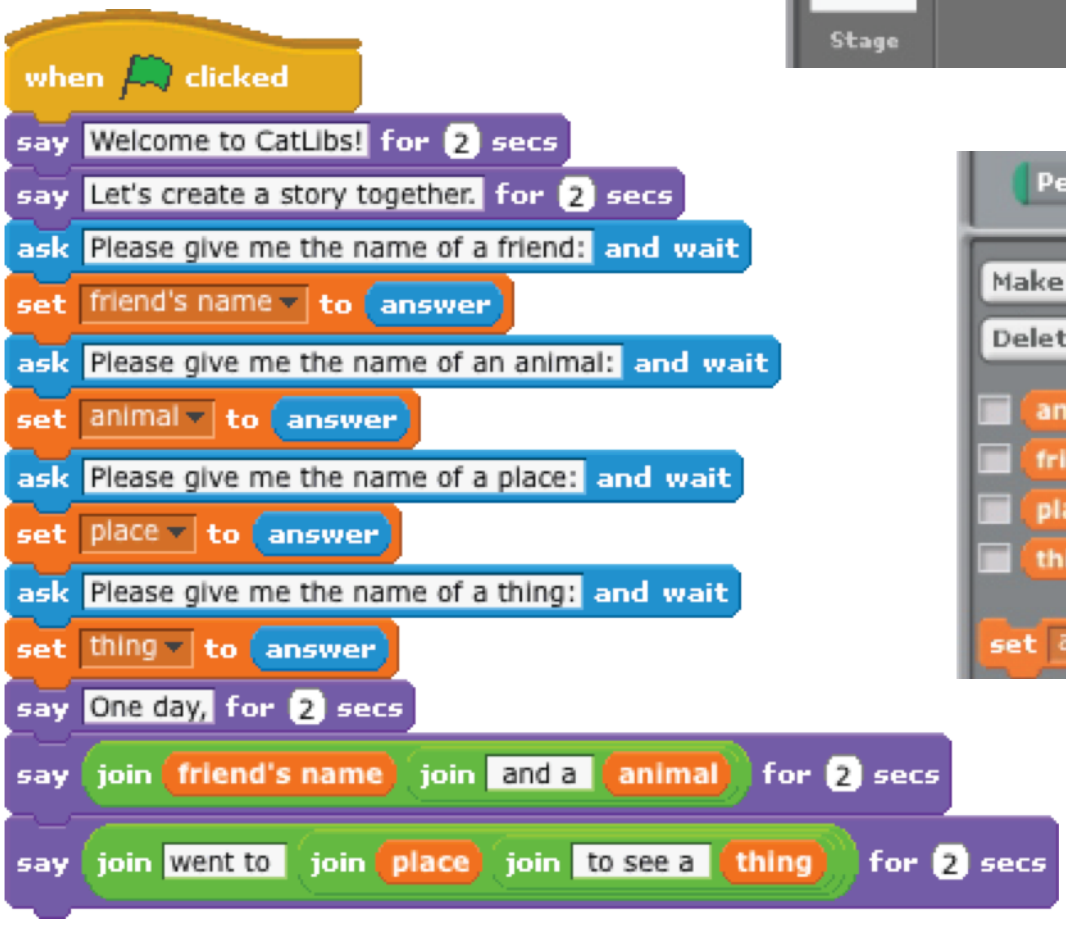


Create a unique Madlib story by collecting user input.



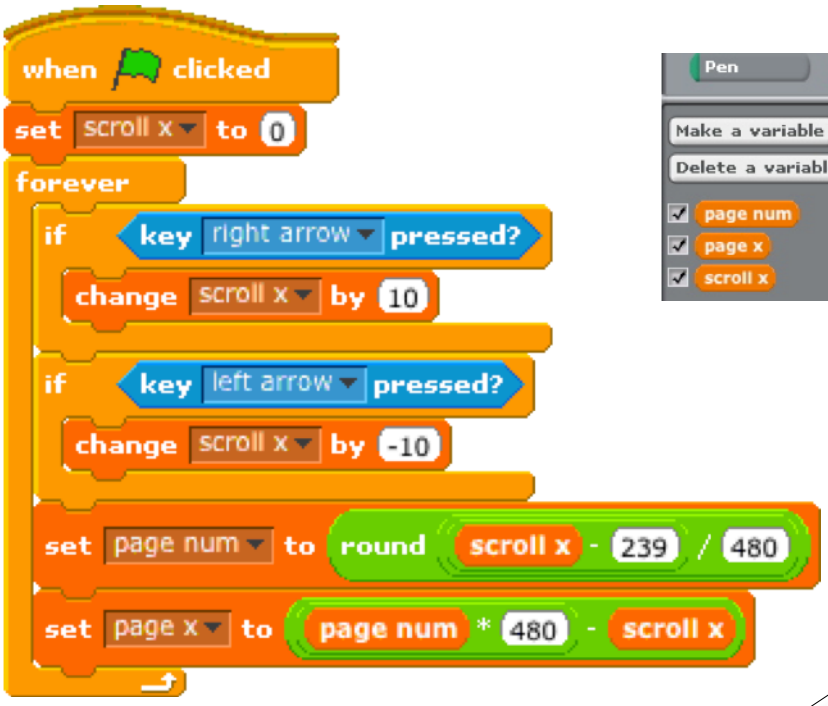
no scripts

1 script
4 variables



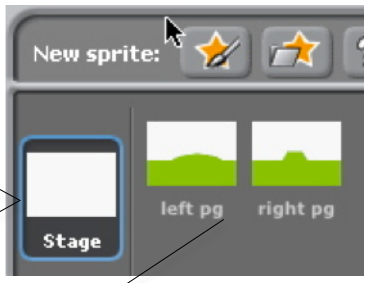
SCROLLING

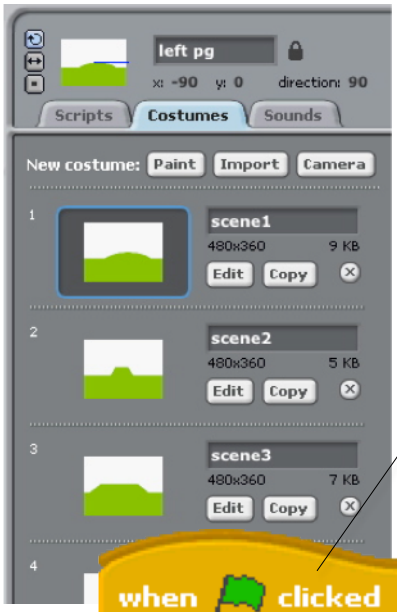
Create the foundation for a side scrolling game.



3 variables

1 script





create two sprites:
one for the left background panel,
one for the right background panel

add the same (2 or more) costumes to each sprite

add this script to the left sprite

add this script to the right sprite

Plan for my final project

Name: _____

A description of the project that I want to create:

The steps I will take to develop my project:

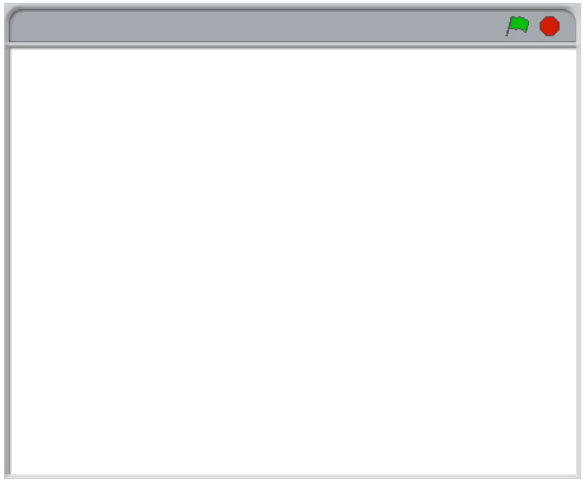
The resources (e.g. people, sample projects) I already have to develop my project:

The resources (e.g. people, sample projects) I might need to develop my project:

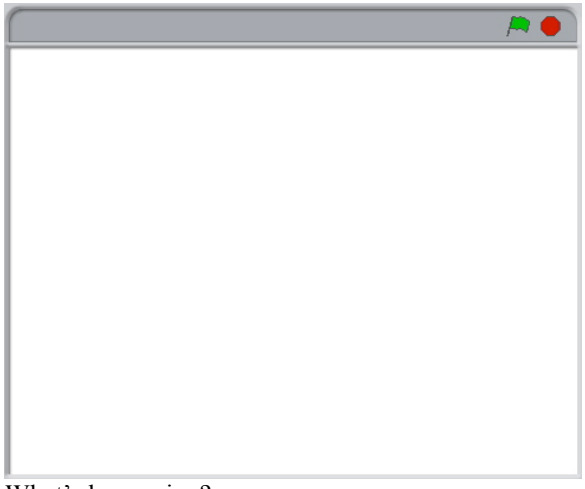
Sketches of my final project

Name: _____

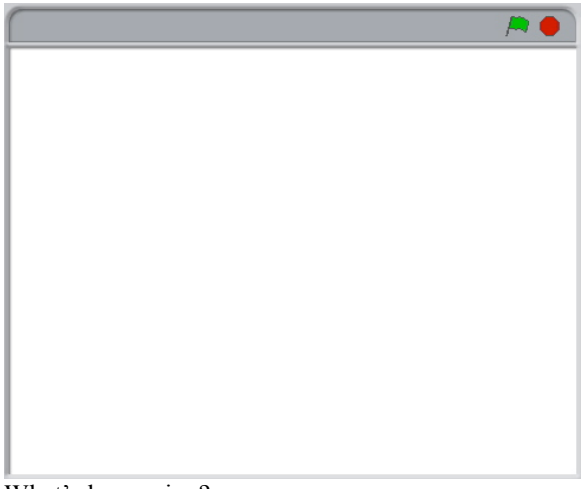
What's happening?
What are the important elements?



What's happening?
What are the important elements?



What's happening?
What are the important elements?



What's happening?
What are the important elements?

Project feedback

Feedback for: _____

Feedback by: _____

Parts of the project that I really liked	*Parts of the project that I would fix or change or add*

Parts of the project that might be helpful to think about:

- *Clarity*: Did you understand what the project is supposed to do?
- *Features:* What features does the project have? Does the project work as expected?
- *Appeal:* How engaging is the project? Is it interactive, original, sophisticated, pretty, funny, or interesting? How did you feel as you interacted with the project?

My project reflections

Name: _____

What?
What is your project?
How does it work? How did you come up with the idea?

So what?
What was your process for developing the project?
What was interesting, challenging, and surprising? Why?
What did you learn?

Now what?
What are you most proud of about your project? What would you change?
What do you want to create next? Why?