Events: Gaming

Overview

In event-based programming, scripts are triggered when specific events occur (e.g., a sprite is clicked, a key is pressed). In this module, students learn to use a variety of events to trigger scripts that resize sprites or cause a sprite to say something.

Module Objectives

Students will be able to:

- Define an event.
- Write scripts to run when a particular key is pressed.
- Write scripts to run when a particular sprite is clicked.
- Change and set the size of sprites.
- Copy a script from one sprite to another.

Big Idea:

Any interactive system is programmed using events - associating code with each action the user might perform. The apps you use on your phone are programmed this way. Being able to think of a project as the things you want to happen for each user event will allow students to program this way.

Module Overview

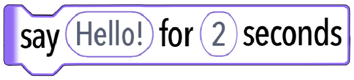
Lesson 1: Exploring Events (60-120 minutes)	Lesson 2: Creating with Events (120 minutes)
The concept of Events in programming is introduced. Students use and modify a Scratch project that handles basic events and responds with an action.	Students use events in an open-ended, create project about a topic of their choice.
Assessment: Events: Assessment Google doc PDF Form Answer Key	

Prior knowledge

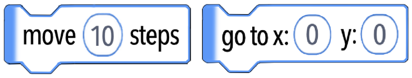
Events

when clicked

Looks

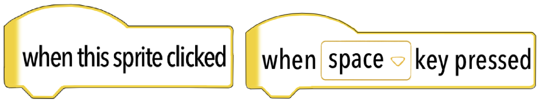


Motion

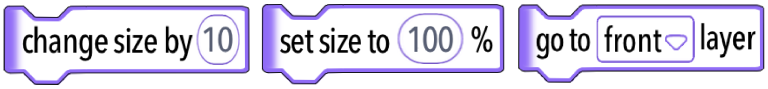


New Blocks

Events



Looks



Lesson 1: Exploring Events (60-120 minutes)

Lesson Overview

In this lesson, students create and modify scripts that handle basic events. A starter project is provided with sprites and example code that handles an event and then responds with an action.

Vocabulary: event, action

Lesson At-a-Glance

Engage (20-30 minutes)
Explore (30-65 minutes)
Reflect (10-25 minutes)

Lesson Preparation

Student Materials

Engage

- Events: L1 TIPP&SEE Sheet
  Google Doc PDF Form Answer Key

Explore

- Events: L1 Modify Sheet
  Google Doc PDF Form
- Events: L1 Modify Step-by-Step Sheet (only as needed)
  Google docs PDF

Scratch Project(s) & Other Resources

Engage & Explore

- Events Engage Video: Drive YouTube
- Events: Race in Space Scratch Projects
  Teacher Sample Student Modify

Engage

Introductory Discussion (10-15 minutes)

Use Question / Student Answer / Teacher Clarification to introduce events through this sequence of questions.

What is an event? *[A time when something happens - like a birthday party!]*

Events in programming are different from the types of events we think of in real life. They aren't something we think is special, they are an action someone takes when using a computer, and the computer does an action in response.

What do you do if someone taps you on the shoulder? *[You turn to look over your shoulder.]*

The tap on the shoulder is like an **event** in programming. When someone taps us on the shoulder, we turn to look to see who is there. Turning to look is the **action** you decided on to handle the event of being tapped on the shoulder.

What *event* have you already used in your Scratch projects? *[`when green flag clicked`]*

In programming, an event occurs when a user does something (e.g., clicks the green flag). The programmer writes scripts that handle those events and then do something - or carry out an action (e.g., When the green flag is clicked, a sprite says, 'Hi!').

What events have you used when you use a computer or play a video game? [*buttons pressed on video game controllers, mouse clicks, typing on keyboard, voice recognition with Siri, swiping or touching with finger on a tablet or phone*]

Scratch can only respond to some of those events. You will learn how to make your program respond to green flag clicks, typing on the keyboard, and mouse clicks.

Events Engage Video

Use the recorded Events Engage Video to introduce content for students who miss class when the concept is introduced, when teaching online or asynchronously, or for students who benefit from multiple exposures to content.

Modeling TIPP&SEE (10-15 minutes)

Tell students that today you are going to ***model part*** of **TIPP&SEE**, and then they will do the rest themselves. Point out the **TIPP&SEE** classroom poster and remind students that they can look there if they forget a part of the strategy.

Ask students to raise their hands if they remember what **TIPP&SEE** is. Explain to students that today you are going to look at a project together.

- Pass out the Events.Gaming: L1 TIPP&SEE Sheet
- Project the Events: Race in Space project https://scratch.mit.edu/projects/323032226/
- Ask students questions to lead them through **TIPP&SEE** as a class.
  First, we will get a Tip from the project page.
  - **Title:** Read the title, what does it tell us?
  - **Instructions:** Read the instructions, what are we going to do? (*read instructions*)
  - **Purpose:** What is the purpose of looking at this code? (*read purpose*)
  - **Play**: Run the code to see what it does. What should I do to run the program? [*green flag*] What other events should I try? [*space bar, click on pictures*]. (*demonstrate only one of the events this time*)

  Now, we will SEE inside to learn more
  - **Sprites:** What are the Sprites? What are their names and pictures? Do they have multiple costumes? Do they have scripts?

- **Events:** What events trigger something to happen?
- **Explore:** Now let's look around the code for any blocks we don't recognize and see if we can learn about them by exploring! How do I learn by exploring? [*Change numbers in the blocks, remove or add blocks*] Click on the purple spaceship "Beep" Sprite and look at the scripts. What block here do I not recognize? [*go to front, change size by, set size to*] What happens if I remove the `go to front` block? (remove from `when this sprite is clicked script`, click on Beep (purple) a few times). Beep stays behind Bop. Put the block back and show again.

Scratch Skill: Copying scripts from one sprite to another with drag and drop
Project the Events: Race in Space Project (same as above): https://scratch.mit.edu/projects/323032226/. Click on "Bork" (pink alien) Sprite to show there are no scripts. Click on "Beep" (purple spaceship) and drag the `when the green flag is clicked` script from Beep (purple) to Bork (pink). Show Bork's scripts again. The `go to x:_ y:_` would need to be updated.

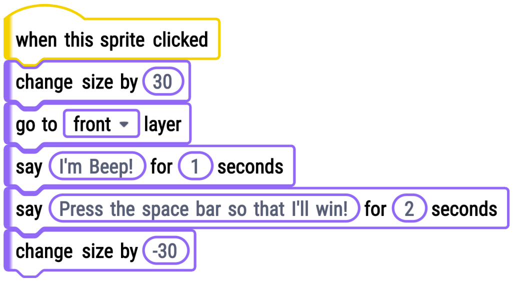
Explore

Using TIPP&SEE (10-20 minutes)

Students open the Events: Race in Space Student Modify Project: (https://scratch.mit.edu/projects/323032226/) and complete the Events.Gaming: L1 TIPP&SEE Sheet to work through the **TIPP&SEE** strategy in more detail on their own.

Modify Project (20-45 minutes)

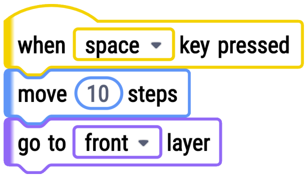
Pass out the Events.Gaming: L1 Modify Sheet. Remind students to **R**oa**R** and hi**SS** (**R**eload, **R**emix, **S**hare and +Add to **S**tudio) the Events: Race in Space Student Modify project: https://scratch.mit.edu/projects/323032226/. Remind students that they need to share the project so it is accessible to you and others. A step-by-step building sheet may be given to students who need additional help.



```
when this sprite clicked
change size by (30)
go to [front ▾] layer
say (I'm Beep!) for (1) seconds
say (Press the space bar so that I'll win!) for (2) seconds
change size by (-30)
```

In this activity, students will modify the project to do the following:

- Copy when this sprite clicked script from purple spaceship "Beep" Sprite to the blue spaceship "Bop" Sprite and pink alien "Bork" Sprite.
- Personalize by choosing a costume and modifying say blocks for "Bop" and "Bork" sprites to include the planet they are from.
- Copy when space key pressed script from Beep to Bop and modify the event to when right arrow key pressed.



```
when [space ▾] key pressed
move (10) steps
go to [front ▾] layer
```

Remind students to save and share their work before exiting Scratch.

Extensions:

- Students explore by changing the scripts that handle events to see how the functionality changes.
- Students add scripts to handle additional events. (e.g., students make sprites say additional things when other keys are pressed, students make sprites move when an event occurs).

Reflect & Discuss

Reflect (5-10 minutes)

New Blocks and Concepts and Reflection Questions are included on the Events.Gaming: L1 Modify Sheet.

New Blocks and Concepts:	**Journal Questions:**
• `when this sprite clicked` • `when <key> key pressed` • `go to front` • `change size by__` • `set size to __%` • Event • Action	• How would you explain what an event is in computer programming? • How is having different types of Events blocks helpful when coding?

Share & Discuss (5-15 minutes)

Consider having students present their projects to the class (or make time for a walk around showcase where students run each other's projects on individual computers). If there is time, consider leading a discussion about how events could be used in their projects.

Assessment:	**Integration:**
Student projects should be graded based on completion of all required elements.	Math - Initializing sprite to 100% and increasing size by 75% uses percentages. Decreasing sprite size uses negative numbers.

Differentiation:

- Prepare block definition sheets for students to add to their journals. Options include: List the block names (students fill in the definitions), block names and definitions provided (students match block names to the definitions).
- Provide a list of the type and number of blocks needed to complete each activity.

Lesson 2: Creating with Events (120 minutes)

Lesson Overview	Lesson At-a-Glance
Students use events in an open-ended, create project. Depending on your students' prior experience and comfort with the content in Lesson #1, this lesson could be skipped to save time.	**Engage** (15 minutes) **Explore** (80 minutes) **Reflect** (25 minutes)

Lesson Preparation	
Student Materials Explore ● Events: L2 Create Sheet Google Doc PDF Form Assessment ● Events: Assessment Google docs PDF Form Answer Key	Scratch Project(s) Engage ● Events: Gaming A Race in Space Teacher Sample

Engage

Demonstration (15 minutes)

Ask if anyone remembers what an event is in programming [*Something a user does*]. Ask if they remember what three event blocks were used in the last lesson? *[when green flag clicked, when this sprite clicked, when <key> key pressed]*

Project the Events: Race in Space Teacher Sample project (from Lesson 1): https://scratch.mit.edu/projects/323037594/.
Click See inside, then click on the purple spaceship "Beep" sprite to reveal the blocks. Show what happens when you click "Beep" (on the stage) a few times (the sprite gets bigger, introduces itself, and tells you how to make it move). Ask, "What block makes the sprite bigger each time the sprite is clicked?" *[change size by __]*

Click on the sprite again and stop the project using the stop sign while the sprite is talking so that it is stopped while it is big. Then, click the green flag to show that the sprite returns to its original size.

Ask the following questions to clarify the difference between change size by and set size to:

- "What block makes the sprite return to its original size?" *[set size to __ %]*

- "How are `change size by` and `set size to` different?" *[`change size by __` makes something bigger (or smaller) each time an event occurs; `set size to __%` sets the sprite to a specific size only]*

Demonstrate these things enough times that it becomes clear that *`set size to __%`* always sets the sprite to the same size while *`change size by __`* continually makes a sprite bigger (or smaller) each time an event occurs.

Explore

Create Project (80 minutes)

Introduce today's activity: To create a project using events. Pass out the [Events: L2 Create Sheet] and give students guidance throughout the project.

To complete this task, students will:

- Create a new Scratch project
- Brainstorm content for the project and plan how the project is to be implemented
- Add a backdrop
- Select three sprites.
- Write scripts to handle all three types of events:
    - `when green flag clicked`
    - `when this sprite clicked`
    - `when <key> key pressed`
- Use `say ___ for ___ sec` blocks to introduce themselves, say something about their topic, and describe what "event" will cause them to move.
- Use `change size by __` block to change size, `set size to ___` to initialize it when green flag clicked

Explain to students that they can create any project that is interesting to them, but some ideas are:

- Telling about a holiday that the student likes to celebrate
- Describing a family celebration and what makes that celebration unique and special to the student's family
- Introducing the student's favorite place around your city
- A topic of their choice!

Remind students to run their project to test that it works as expected. Debug and re-test if necessary. Students should **S**hare and +Add to **S**tudio before exiting Scratch.

Extensions:

- Have students add scripts to handle additional events (e.g., students make the sprites move in different ways when an event occurs; sprites get bigger when one event occurs, but then get smaller when a different event occurs).

Reflect & Discuss

Reflect (10 minutes)

Reflection Questions are included on the Events: L2 Create Sheet.

Journal Questions:

- What was challenging about this project?
- How did you work through the challenges you faced?
- What is the difference between the `change size by__` and `set size to __% blocks?`

Share & Discuss (15 minutes)

Consider having students present their projects to the class (or make time for a walk around showcase where students run each other's projects on individual computers). If there is time, lead a discussion about how events could be used in other projects (e.g. map movements in a game to arrow keys, map different sounds to each sprite when it is clicked).

Assessment:

- Student projects should be graded based on completion of all required elements.
- Students will complete the Events: Assessment based on the concepts covered in this module.

Integration:

Math - Initializing sprite to 100% and increasing size by 75% uses percentages. Decreasing sprite size uses negative numbers.

Differentiation:

- Prepare block definition sheets for students to add to their journals. Options include: List the block names (students fill in the definitions), block names and definitions provided (students match block names to the definitions).
- Provide a list of the type and number of blocks needed to complete each activity.

Standards

CSTA Standards

1B-AP-10 Create programs that include sequences, events, loops, and conditionals. (P5.2)

1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features. (P5.3)

1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs. (P5.2, P7.3)

1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. (P6.1, P6.2)

1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations. (P7.2)

2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution. (P4.2, P5.2, P7.3)

1B-IC-20 Seek diverse perspectives for the purpose of improving computational artifacts. (P1.1)

1B-IC-21 Use public domain or creative commons media, and refrain from copying or using material created by others without permission. (P7.3)

ISTE Standards

1. **Empowered Learner:** Students leverage technology to take an active role in choosing, achieving and demonstrating competency in their learning goals, informed by the learning sciences. Students:

    c. use technology to seek feedback that informs and improves their practice and to demonstrate their learning in a variety of ways.

    d. understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies.

2. **Digital Citizen**: Students recognize the rights, responsibilities and opportunities of living, learning and working in an interconnected digital world, and they act and model in ways that are safe, legal and ethical. Students:

    b. engage in positive, safe, legal and ethical behavior when using technology, including social interactions online or when using networked devices.

4. **Innovative Designer:** Students use a variety of technologies within a design process to identify and solve problems by creating new, useful or imaginative solutions.

    a. know and use a deliberate design process for generating ideas, testing theories, creating innovative artifacts or solving authentic problems.

d. exhibit a tolerance for ambiguity, perseverance and the capacity to work with open-ended problems.

6. **Creative Communicator:** Students communicate clearly and express themselves creatively for a variety of purposes using the platforms, tools, styles, formats and digital media appropriate to their goals.

b. create original works or responsibly repurpose or remix digital resources into new creations.