

Conditional Loops: Youth Culture

Overview

Within a conditional loop, repeated actions will continue until a specific condition becomes true. In Scratch, the condition that is fulfilled can relate to another sprite or a specific color. In this module, students will implement conditional loops that control the repeated actions of transportation vehicles. The car, bus, or train will stop moving when the condition of the loop becomes true due to the vehicle reaching a color or a sprite.

Module Objectives

Students will be able to:

- create an animation with one costume and repeated movement.
- write a script using a conditional loop that repeats an action until a condition is true.



Big Idea:

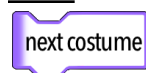
It is very often that loops need to execute a number of times that cannot be predetermined, therefore simple repeat loops cannot be used. Programmers need to think about what will decide when the loop stops. Here we step through both the design and implementation of such loops in Scratch.

Module Overview

<u>Lesson 1: Exploring Conditional Loops</u> <u>(120 minutes)</u> The concept of a conditional loop is introduced. Students stop the action within a program when a condition becomes true using sensing blocks.	<u>Lesson 2: Creating with Conditional Loops</u> <u>(120 minutes)</u> Students use conditional loops in a create project with moving animation that stops when a condition becomes true.
<u>Assessment:</u> Conditional Loops: Assessment Google doc PDF Form Answer Key	

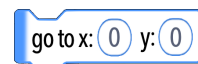
Prior knowledge

Looks



(Extension)

Motion

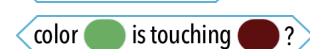
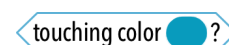


New Blocks

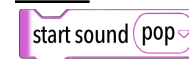
Control



Sensing

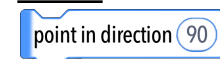


Sound



(extension)

Motion



Lesson 1: Exploring Conditional Loops (120 minutes)

Lesson Overview In this lesson, students learn how to stop an action when a condition becomes true by modifying a conditional loop using a <code>touching < >?</code> block. Vocabulary: conditional loop	Lesson At-a-Glance Engage (20 minutes) Explore (75 minutes) Reflect (25 minutes)
--	---

Lesson Preparation	
Student Materials Explore - Conditional Loops: L1 TIPP&SEE Sheet Google doc PDF Form Answer Key - Conditional Loops: L1 Modify Sheet Google doc PDF Form Answer Key - Conditional Loops: L1 Modify Step-by-Step Sheet (only as needed) Google docs PDF	Scratch Project(s) & Teacher Materials Engage & Explore - Conditional Loops Engage Video: Drive YouTube - Conditional Loops: Waiting for a Ride Student



Engage

Introductory Discussion (20 minutes)

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



What is a repeated action? *[Something that happens over and over again]*



Do repeated actions happen forever? *[No, all loops end. In animation we used repeated actions that only repeat a certain number of times]*

Repeated actions within computer program exist within loops. Loops can have a set number of repetitions, as seen in the animation module, can continue as long as the program is running, or can end based on a condition becoming true. In this module, we will focus on loops that run until a condition is true, these are called conditional loops.



Introduce students to the concept of a **conditional loop**, a repeated action that continues until a specific end condition becomes true. Explain that with a conditional loop, the action repeats an undetermined number of times because the action will not stop until the end condition is true. An example of this is running until you cross the finish line, walking until you reach a locked door, or reading until the end of a book.

Ask students to brainstorm other repeated actions and write them on the board. If actions without conditions come up, write them on the board as well, they will be teachable moments later in the discussion. Collect 5-8 ideas depending on how much time you have.

When all of the ideas are on the board, ask the students to identify the actions and the **end conditions** (what causes the action to stop). For each example that is on the board, ask, “What are the actions and what are the end conditions?” As students answer, underline the actions and **put a box around the end conditions**. For example running until you **cross the finish line** and walking until you **reach a locked door**. If students have volunteered any actions that do not have conditions, point them out as you go and demonstrate that they do not have an end condition, showing the difference between action and actions with end conditions. After discussing these as a class, erase them from the list.



[Conditional Loops Engage Video](#)

Use the recorded Conditional Loops 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.



Explore

Using TIPP&SEE (25 minutes)

Tell students that today they are going to use **TIPP&SEE** on their own. Point out the **TIPP&SEE** classroom poster and remind students that they can look there if they forget a part of the strategy. Remind students that **TIPP&SEE** allows them to get their head in the game, focus on the project, and explore the code.

Students open the Conditional Loops: Waiting for a Ride project (<https://scratch.mit.edu/projects/323733135/>) and complete the [Conditional Loops.YouthCulture: L1 TIPP&SEE Sheet](#) to work through the **TIPP&SEE** strategy.

After they have finished, review with students what happened with each of the Explore:Modify suggestions.

Modify Project (50 minutes)

Pass out the [Conditional Loops.YouthCulture: L1 Modify Sheet](#). Remind students to **R**o**a**R and **h**i**S**S (**R**eload, **R**emix, **S**hare and **+A**dd to **S**tudio) the Conditional Loops: Waiting for a Ride project (<https://scratch.mit.edu/projects/323733135/>). Remind students that they need to share the project so it is accessible to you and others. Remind students about using the **WHAT?!? Strategy** to help them debug and fix any problems. A [step-by-step building sheet](#) may be given to students who need additional help.

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

- Modify the script to stop the car using a different condition.
- Personalize by choosing a costume, adding a `say__for__secs` block, and changing the speed of the car.

Extensions:

- Add another sprite and have it stop at another sprite or a color.
- Add a sound when the sprite stops moving.
- Have a sprite go back to the start after it stops moving.



Reflect & Discuss

Reflect (10 minutes)

New Blocks and Concepts and Reflection Questions are included on the [Conditional Loops.YouthCulture: L1 Modify Sheet](#).

New Blocks and Concepts:

- `repeat until`
- `touching <sprite>?`
- `touching <color>?`
- `color <color> is touching <color>?`
- Conditional Loop

Journal Questions:

- Why is a conditional loop a useful tool in programming?
- How would you use sensing blocks in another project?
- What other actions could you place inside a conditional loop?
- What happens when you use a negative number in a move block?
- How do you make a sprite look to the left?

Share & Discuss (15 minutes)

Set up a gallery walk with projects on each computer and allow students to circulate around the room to visit their classmate's projects and view them. If there is time, consider leading a discussion about how conditional loops with sensing blocks could be used in their projects.

Assessment:

Student projects should be graded based on completion of all required elements.

Integration:

Social Studies - societies

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 Conditional Loops (120 minutes)

Lesson Overview In this lesson, students animate a sprite to move across the stage and use a conditional loop to stop the sprite when a condition becomes true using a sensing block.	Lesson At-a-Glance Engage (15 minutes) Explore (80 minutes) Reflect (25 minutes)
---	---

Lesson Preparation	
Student Materials Explore - Conditional Loops: L2 Create Sheet Google doc PDF Form Assessment - Conditional Loops: Assessment Google doc PDF Form Answer Key	Scratch Project(s) Explore None



Engage

Introductory Discussion (15 minutes)

Use a discussion to review the blocks and concepts covered in the previous lesson.



What is a conditional loop? *[A loop that continues until a condition is determined to be true.]*



What block do you use in Scratch to create a conditional loop? *[repeat until]*



How does the `repeat until` block work? *[Anything inside the block will continue to happen until the condition is determined to be true]*



What do you use with `repeat until` blocks? *[sensing blocks - (touching <sprite>?, touching <color>?, color <color> is touching <color>?)]*



What did the `point in direction` block that we discovered during our exploration do? *[pointed the sprite in the selected direction so that it would move the right way.]*



Based on what you learned in the previous lesson, how would you make a sprite move from right to left rather than from left to right? *[They could move backwards to the left by using negative numbers in the `move_steps` block or they could turn around and move forwards by using `point in direction` to turn around and positive numbers in the `move_steps` block.]*

In Scratch, the `move ___ steps` block moves the sprite depending on the direction that the sprite is facing. In order to have the sprite move from right to left rather than left to right, the `point in direction ___` block is used to turn the sprite to face the other direction so that forward movement for that sprite goes right to left.



Explore

Create Project (80 minutes)

Introduce today's activity: To create a project using animation that moves across the screen and a conditional loop to stop when a condition becomes true. Pass out the [Conditional Loops: L2 Create Sheet](#) to give students guidance throughout the project.

To complete this task, students will:

- Select a backdrop and at least 3 sprites for their project.
- Move the people sprites to logical locations within the stage
- Build a script on their chosen sprite to make it move (including the `go to x: __ y: __` block to start at a specific location and a `repeat until __` conditional). If students choose to make it move from right to left, they will need to use a `point in direction` block.
- Add a sensing block to stop their chosen transportation: either `touching <sprite>?, color <color> is touching <color>?, or touching <color>?` and identify the condition that will make the movement stop.
- Add a say or sound block to signal that the sprite has stopped.

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

- Give a walking tour of your neighborhood
- A dance off with different sprites moving to the center
- A topic of your choice!



Note: If students want to give a walking tour of their neighborhood, they can either use a built-in backdrop from Scratch, find a picture on the internet of part of their neighborhood and upload it as a backdrop, or take a picture themselves to upload as a backdrop. It might be helpful to talk to students about this option ahead of time so they come to class prepared with their picture if that is the option they want to choose.

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

Extensions:

- Add `repeat` and `move` blocks to animate another sprite.
- Add a sprite from the sprite library (resize if needed) to be used in the sensing block as the condition that stops the sprite.
- Have the sprite stop in different places when it is triggered by different events.



Reflect & Discuss

Reflect (10 minutes)

Reflection Questions are included on the [Conditional Loops: L2 Create Sheet](#).

New Blocks: point in direction__	Journal Questions: - What was challenging about this project? - How did you work through the challenges you faced? - How did you choose which sensing block to use in your project? - How is moving a sprite from right to left different from moving the sprite from left to right?
---	---

Share & Discuss (15 minutes)

Consider having students present their projects to the class or set up a gallery walk with projects on each computer and allow students to circulate around the room to visit their classmate's projects and view them. If there is time, lead a discussion about challenges, questions, and new discoveries identified while completing their project.

Assessment: - Student projects should be graded based on completion of all required elements. - Students will complete the Conditional Loops: Assessment based on the concepts covered in this module.	Integration: Social Studies - societies
Differentiation: - Prepare a block definition sheets to add to journals - block names listed (students fill in the definitions) or blocks and the definitions completed, 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. (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.
- c. develop, test and refine prototypes as part of a cyclical design process.
- 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.
- d. publish or present content that customizes the message and medium for their intended audiences.