

# Lesson 13: Sprite Movement

45 minutes

## Overview

This lesson builds on the draw loop that students learned previously to create programs with *purposeful* motion. Students learn how to control sprite movement using a construct called the counter pattern, which incrementally changes a sprite's properties. Students first brainstorm different ways that they could animate sprites by controlling their properties, then explore the counter pattern in Code Studio. After examining working code, students try using the counter pattern to create various types of sprite movements. The skills that students build in this lesson lay the foundation for all of the animations and games that they will make throughout the rest of the unit.

**Question of the Day:** How can we control sprite movement in Game Lab?

## Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ AP - Algorithms & Programming

## Agenda

**Warm Up (5 minutes)**

**Reviewing Sprite Properties**

**Activity (35 minutes)**

**Wrap Up (5 minutes)**

**Journal**

## Objectives

Students will be able to:

- Identify which sprite properties need to be changed, and in what way, to achieve a specific movement
- Use the counter pattern to increment or decrement sprite properties

## Preparation

- Check the "**Teacher's Lounge**" forum for verified teachers to find additional strategies or resources shared by fellow teachers
- If you are teaching virtually, consider checking our **Virtual Lesson Modifications**

## Links

**Heads Up!** Please make a copy of any documents you plan to share with students.

For the teachers

- **Sprite Movement** - Slides  
▼ Make a Copy

For the students

- **Sprite Movement** - Video  
(**Download**)

# Teaching Guide

## Warm Up (5 minutes)

### Reviewing Sprite Properties

**Review:** On a piece of scratch paper, list out all of the sprite properties you can think of and what aspect of a sprite they affect.

**Prompt:** What kinds of animations could you make if you could control these properties? Consider adding and subtracting from properties, or even updating multiple properties at the same time.

**Discuss:** Allow students to share their ideas with a partner before sharing with the entire class.

**Discussion Goal:** The purpose of this discussion is to start students thinking about how they might use the various sprite properties they've seen so far to make animations with purposeful motion. If students struggle to come up with ideas, you can narrow down the question to specific properties. For example:

- What would happen to a sprite if you constantly increased its **x** property?
- What would happen to a sprite if you constantly increased its **y** property?
- What about other properties, or combining multiple properties?

**Question of the Day:** How can we control sprite movement in Game Lab?

## Activity (35 minutes)

**Transition:** Send students to Code Studio.

💡 Teaching Tip ▲

**Guide to Programming Levels:** Additional guidance for programming levels is provided in the [CSD Guide to Programming Levels](#). This document includes strategies and best-practices for facilitating programming levels with students.

🖥️ 1

Prediction

📺 **Video:** Show students the [Sprite Movement](#) video in the slides.

💡 Teaching Tip ▲

To encourage active engagement and reflection, use one or more of the strategies discussed in the [Guide to Curriculum Videos](#).

**Questions to Consider with Video:**

- What is the counter pattern?
- How does the counter pattern move sprites across the screen?

**Discussion Goal:**

- *What is the counter pattern?*

- Students may describe the counter pattern in various ways. Make sure that they go beyond just stating the blocks or code that the counter pattern uses.
- They should understand that the counter pattern allows the programmer to update the value of a variable in a pattern that counts up (or down) on every iteration of the draw loop.
- This can be used for many different things, such as spinning, growing, shrinking, or timers, but it's most often used to move sprites across the screen.
- *How does the counter pattern move sprites across the screen?*
  - When a sprite's x or y property is updated in a counter pattern, its position changes in a consistent way over time, causing it to move across the screen.
  - Students should be able to explain that movement for an animation is just a change in position and that changing a sprite's x position will cause it to move horizontally, and changing a sprite's y position will cause it to move vertically.



2-5

Skill Building

2

3

4

5



6

Practice



7



Assessment



Assessment Opportunity ▲

**Formative Assessment:** This level can be used as a formative assessment. A rubric is provided in the level, and written feedback can be given to students. [Click here to learn more about giving feedback to students.](#)



8

Challenges

## Wrap Up (5 minutes)

### Journal

#### Question of the Day: How can we control sprite movement in Game Lab?

**Prompt:** You've seen two ways to create animations with the draw loop: random numbers and the counter pattern. What is one type of movement that you'd want to use random numbers for? What is one type of movement that you would want to use the counter pattern for? Are there any movements that might combine the counter pattern and random numbers?

**Discuss:** Allow students to discuss with a partner before sharing with the entire class.

**Discussion Goal:** Student answers may vary, but movements such as shaking use random numbers, and movements that are more purposeful use the counter pattern. It's less important that students have a specific answer than that their answers reflect a growing understanding of how the different programming

constructs work.

Students may have seen random numbers used on one property (rotation) and the counter pattern used on another (x position) in one of the challenges, but if there is time, encourage them to think about what would happen if you used the counter pattern to add or subtract a random number to a sprite property, or to combine the two constructs in other ways.



This work is available under a **Creative Commons License (CC BY-NC-SA 4.0)**.

If you are interested in licensing Code.org materials for commercial purposes **contact us**.