

# Lesson 22: Complex Sprite Movement

45 minutes

## Overview

This lesson does not introduce any new blocks and in fact, only uses patterns students have seen in Chapter 1 and demonstrates how combining these patterns, in particular the abstractions students learned in the previous two lessons, allows them to build new behaviors for their sprites. Specifically, this lesson has students learn how to combine the velocity properties of sprites with the counter pattern to create more complex sprite movement. After reviewing the two concepts, they explore various scenarios in which velocity is used in the counter pattern and observe the different types of movement that result, such as simulating gravity. They then reflect on how they were able to get new behaviors by combining blocks and patterns that they already knew.

**Question of the Day:** How can previous blocks be combined in new patterns to make interesting movements?

## Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

► AP - Algorithms & Programming

## Agenda

Warm Up (5 minutes)

Activity (35 minutes)

Wrap Up (5 minutes)

## Objectives

Students will be able to:

- Explain how individual programming constructs can be combined to create more complex behavior
- Use sprite velocity with the counter pattern to create different types of sprite movement

## 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

- [Complex Sprite Movement](#) - Slides  
▼ Make a Copy
- [Velocity and the Counter Pattern - Digital Resource](#) - Resource
- [Velocity and the Counter Pattern - Unplugged](#) - Resource

## Teaching Guide

### Warm Up (5 minutes)

**Display:** Show the two images of a frog jumping to the class.

**Prompt:** Here are two images of a frog jumping. The first is from the side scroller. Do you have any ideas for how to make the second type of jumping?

**Think-Pair-Share** Allow students to discuss their ideas with a classmate before sharing with the entire class.

**Discussion Goal:** It is unlikely that students will come up with the solution on their own, but encourage them to think of as many ideas as possible, and that they will keep working on the problem throughout the class.

### *Remarks*

We've learned a lot of new blocks that have helped us create some fun animations. Today, we're going to look at how we can use the blocks that we already know in new ways to make more interesting types of movements. By the end of the class, you'll be able to code the new type of jumping with blocks that you already know.

**Question of the Day:** How can previous blocks be combined in new patterns to make interesting movements?

## Activity (35 minutes)

**Transition:** Move 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**

 Teaching Tip

This level introduces the primary new programming pattern of this lesson, combining the counter pattern with sprites' velocity properties. Encourage students to take seriously their predictions before actually running the code.

 2-4

**Skill Building**

2

3

4

5

Practice

6

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.](#)

7

Challenges

## Wrap Up (5 minutes)

**Share:** Have students share with their classmates what additions they made to their final flyer game. Have students focus not just on how the game works, but on what the code to create that kind of functionality looks like.

**Prompt:** On your paper make two lists. First, make a list of new things you can program sprites to do after today's lesson. On the second list write down all the new blocks you learned today.

**Discussion Goal:** This conversation should highlight that students did not learn any new blocks in today's lesson, they just learned new ways to combine blocks and patterns they had learned previously. The broader point here is that programming is not always about learning new blocks but being creative about combining the tools you already know how to use.

**Discuss:** Have students share their lists with classmates. Afterwards share lists as a class. They should hopefully have listed many new sprite movements but students haven't actually learned any new blocks in this lesson.

**Prompt:** Today we built lots of new sprite movements like gravity and jumping, but none of this required us to learn new blocks. How were you able to do new things without learning any new blocks?

**Discuss:** Lead a quick follow-up to your initial discussion about this point.

✓ Assessment Opportunity ▲

Check that students can explain the new programming structures and algorithms that they were able to use to get the new behaviors in the program.

## Remarks

We're going to keep learning a few more tools in Game Lab, but as we do, remember what we saw today. To create new kinds of programs you don't always need to learn new blocks. Most of the time the creativity of programming comes from learning to combine things you already know in new and creative 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**.