

# Lesson 21: Complex Sprite Movement

## Overview

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

Students learn 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. In particular, students learn how to simulate gravity. They then reflect on how they were able to get new behaviors by combining blocks and patterns that they already knew.

## Purpose

This lesson does not introduce any new blocks and in fact only uses patterns students have seen in Chapter 1. Instead, it demonstrates how combining these tools, in particular the abstractions students learned in the previous two lessons, allows them to build new behaviors for their sprites. This highlights the broader point that abstractions not only simplify code, but also can themselves be used as building blocks of even more complex behavior.

## Assessment Opportunities

1. **Use sprite velocity with the counter pattern to create different types of sprite movement**

See Level 6 in Code Studio.

2. **Explain how individual programming constructs can be combined to create more complex behavior**

In the wrap up discussions, check that students can explain how they are creating new types of movement without new blocks.

## Standards

Full Course Alignment

**CSTA K-12 Computer Science Standards (2017)**

► **AP** - Algorithms & Programming

## Agenda

## 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

## Links

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

For the teachers

- **CSD Unit 3 - Interactive Animations and Games** - Slides
- **Velocity and the Counter Pattern** - Resource

## Lesson Modifications

**Warm Up (5 minutes)**

**Activity (35 minutes)**

**Wrap Up (5 minutes)**

# Teaching Guide

## Lesson Modifications



**Attention, teachers!** If you are teaching virtually or in a socially-distanced classroom, please **click here** to access modifications that can be used during this lesson.

## 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.



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

You can use this level as a formative assessment for students. Click inside the level to view a rubric and leave feedback to your 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

**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.