

# The Big Game - Collision Detection

Lesson time: 30-60 Minutes

## LESSON OVERVIEW

To finish up their video games, students will apply what they have learned in the last few stages to write the final missing functions. We'll start by using booleans to check whether keys were pressed in order to move the player sprite, then move on to applying the Pythagorean Theorem to determine when sprites are touching.

## LESSON OBJECTIVES

Students will:

- Apply the Distance Formula to detect when two points on a coordinate plane are near each other.

## ANCHOR STANDARD

Common Core Math Standards

- **8.G.7:** Apply the Pythagorean Theorem to determine unknown side lengths in right triangles in real-world and mathematical problems in two and three dimensions.

*Additional standards alignment can be found at the end of this lesson*

### TEACHING SUMMARY

#### Getting Started

- 1) [Introduction](#)

#### Activity: The Big Game - Collision Detection

- 2) [Online Puzzles](#)

# TEACHING GUIDE

---

## MATERIALS, RESOURCES, AND PREP

### For the Student

- [Line-length Design Recipe](#) (in the student workbook)
- [Distance Design Recipe](#) (in the student workbook)
- [Collide? Design Recipe](#) (in the student workbook)

## GETTING STARTED

### 1) Introduction

Let's get back into that Big Game from stages 7, 12, and 16.

Previous work with the game has created movement for the danger and target characters, using Booleans to check if they have left the screen. The last time students worked on their game they used a conditional to check which key was pressed and make the player move accordingly. At this point the only thing left to do is to decide when the player is touching either the target or danger. Once students have successfully completed the **distance** and **collide?** functions, their score will increase when the player touches the target, and decrease when it touches the danger.

The Pythagorean Theorem studied in the last lesson will be used to determine when the characters have made contact. Students are not required to write their own [line-length](#) function, but you may ask them to complete the Design Recipe for it anyway.

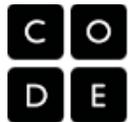
Students will first complete the [distance](#) function so that it measures the distance between two points,  $(px, py)$  and  $(cx, cy)$ . After the students implement the distance formula, they will need to implement the tests in the [collide?](#) function.

Once these last functions are put into place, scoring will automatically update based on collisions between target and danger.

## ACTIVITY: THE BIG GAME - COLLISION DETECTION

### 2) Online Puzzles

Return to your Big Game to use collision detection logic so that you know when your player is touching the target or the danger. Head to [CS in Algebra stage 20](#) in Code Studio to get started programming.



Derived from

**BOOTSTRAP**  
[www.bootstrapworld.org](http://www.bootstrapworld.org)

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

If you are interested in licensing [Code.org](https://code.org) materials for commercial purposes, contact us: <https://code.org/contact>