

# The Big Game - Variables

Lesson time: 30-60 Minutes

## LESSON OVERVIEW

Students get their first look at the inside of their own video games. They will start development by substituting in new Images, Strings, and Numbers for existing variables.

## LESSON OBJECTIVES

Students will:

- Substitute new values into existing variables of an existing program and describe the effects.
- Examine the structure of an existing program.

## ANCHOR STANDARD

Common Core Math Standards

- **6.EE.4:** Identify when two expressions are equivalent (i.e., when the two expressions name the same number regardless of which value is substituted into them). For example, the expressions  $y + y + y$  and  $3y$  are equivalent because they name the same number regardless of which number  $y$  stands for.

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

### TEACHING SUMMARY

#### Getting Started

- 1) [Vocabulary](#)
- 2) [Teaching Notes](#)

#### Activity: The Big Game - Variables

- 3) [Online Puzzles](#)

# TEACHING GUIDE

---

## GETTING STARTED

### 1) Vocabulary

This lesson has three new and important words:

- **Troubleshooting** - when a program generates an unexpected result, a programmer must examine the code to determine the source of the unexpected results (usually an unanticipated input or incorrect handling of an expected input). Sometimes called **debugging**.
- **Mod** - Short for modification. Games in the real world are often a mod of another game. Othello (or Reversi) is usually considered a mod of the ancient game of "Go". A mod of a program is one that has been altered to do something slightly different than its original purpose.
- **Stub** - A function whose domain and range have been designated, but the process to transform the domain into the range has not yet been defined.

### 2) Teaching Notes on the Big Game

The students will create a mod of an existing game. As they make changes to the game, it is possible that they will add code that will either "break" the program (cause nothing to happen) or cause an unexpected outcome. If either of these conditions exist, they will need to troubleshoot or debug the code to determine how to get it working in the proper way. **If things go terribly awry and finding a problem is too frustrating, use the Clear Puzzle button in the upper right corner of the workspace.** This button will clear your game back to its initial state, so it should only be used as a last resort.

This exercise is a simplified version of a very common real world programming task. Programmers often create mods of programs about which they know very little. They slowly unravel which pieces require further understanding in order to make the mod work the way they want, while leaving other parts of the program completely unexplored.

Many programs and functions are customizable through their arguments (which can be variables or values). When a function is called, its arguments are passed in as variables into the function. In other cases, variables that someone might want to change (sometimes called constants) are often at the top of a piece of code. Having access to the code allows the programmer to change the way the program behaves by setting these variables to different values.

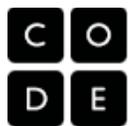
In this lesson, we are creating the mod by changing the variables inside the code. The student has access to the game code and is changing the initial value of the Title, Subtitle, Player, Danger, and Target. As a reminder, the **ultimate** goal of this game will be to manipulate the player through pressing keys, to avoid the danger, and to make contact with the target. The **current** lesson has no motion or interactivity. It only changes the look of the game. The motion and interactivity function stubs, such as "update-target" and "danger?", will be completed in later lessons.

The blocks menu displays a few new items (Boolean, Cond, and Functions) which will be examined in more detail in future lessons. The students should be encouraged to explore each of the sub-menus. However **the only navigation required for this level is editing the five color blocks at the top of the function:** Title, subtitle, bg (background), player, target, and danger. The difference between the color and black blocks will also be explained in a future lesson.

## ACTIVITY: THE BIG GAME - VARIABLES

### 3) Online Puzzles

In this stage you'll define and modify variables to changes how some games function. Head to [CS in Algebra stage 7](#) 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>