

Lesson 15: Keyboard Input

Overview

Question of the Day: How can our programs react to user input?

Following the introduction to booleans and *if* statements in the previous lesson, students are introduced to a new block called `keyDown()` which returns a boolean and can be used in conditionals statements to move sprites around the screen. By the end of this lesson, students will have written programs that take keyboard input from the user to control sprites on the screen.

Purpose

One common way conditionals are used is to check for different types of user input, especially key presses. Having a way for a user to interact with a program makes it more interesting and dynamic. Without interaction from the user it is very difficult to create a game. Therefore the introduction of conditionals and user inputs for decision making is a critical step toward creating games.

Assessment Opportunities

1. Use conditionals to react to keyboard input

See Level 6 in Code Studio.

2. Move sprites in response to keyboard input

See Level 6 in Code Studio.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

► **AP** - Algorithms & Programming

Agenda

Lesson Modifications

Warm Up (5 minutes)

Taking Input

Activity (35 minutes)

Keyboard Input

Objectives

Students will be able to:

- Move sprites in response to keyboard input
- Use conditionals to react to keyboard input

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

Introduced Code

- `keyDown(code)`

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)

Taking Input

Discuss: So far all of the programs you've written run without any input from the user. How might adding user interaction make your programs more useful, effective, or entertaining? How might a user provide input into your program?

Discussion Goal

The goal here isn't to get into the technical specifics of how programs can take input (students will get to that in the online portion of the lesson), but rather to get students thinking about how user input could change the programs they've made. Encourage students to think back to Unit 1 and the various computer inputs and outputs they explored then. Which inputs would be most useful for the types of programs they've been making?

Question of the Day: How can our programs react to user input?

Activity (35 minutes)

Keyboard Input

Transition: Send students to Code Studio

 1

Prediction

 2-4

Skill Building

2

3

4

 5

Practice

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

Wrap Up (5 minutes)

Adding Conditionals

Question of the Day: How can our programs react to user input?

Journal: Think back to all of the programs you've written so far; how might you use user interaction to improve one of your programs from past lessons? What condition would you check, and how would you respond to it?