Lesson 14: Conditionals

Overview

Question of the Day: How can programs react to changes as they are running?

This lesson introduces booleans and conditionals, which allow a program to run differently depending on whether a condition is true. Students start by playing a short game in which they respond according to whether particular conditions are met. They then move to Code Studio, where they learn how the computer evaluates Boolean expressions, and how they can be used to structure a program.

Purpose

This lesson follows closely the booleans model that students first experienced in the Booleans Unplugged lesson. As before, we start with using booleans directly before using booleans to trigger *if* statements. In the following lesson we will introduce some boolean producing blocks, such as `keyDown()`, which can be used in place of simple boolean comparisons to write programs that respond to user input.

Assessment Opportunities

1. **Use conditionals to react to changes in variables and sprite properties**

   See Level 9 in Code Studio.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- **AP** - Algorithms & Programming

Agenda

Lesson Modifications

Warm Up (5 minutes)
Introducing Conditionals

Activity (35 minutes)
Conditionals

Wrap Up (5 minutes)

Objectives

Students will be able to:

- Use conditionals to react to changes in variables and sprite properties

Links

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

For the teachers

- **Booleans and Comparison Operators** - Resource
- **CSD Unit 3 - Interactive Animations and Games** - Slides
- **If Statements** - Resource

For the students

- **Boolean Expressions** - Video (**Download**)
- **Conditional Statements** - Video (**Download**)

Vocabulary

- **Boolean Expression** - in programming, an expression that evaluates to True or False.
- **Condition** - Something a program checks to see whether it is true before deciding to take an action.
- **Conditionals** - Statements that only run when certain conditions are true.

Introduced Code

- __ != __
- __ < __
- __ <= __

Considering Conditions	• `__ == __` • `__ > __` • `__ >= __` • `if (__) { __; }`

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)

Introducing Conditionals

Display: Display the following questions on the board and ask students to answer them.

Teaching Tip

The last two prompts ask students to react to information that may change as they are answering the questions. In order to drive this point home, consider moving from the front of the room and back, as well as alternately tapping and not tapping a pencil, so that students realize that their answers may change according to when they are answering each question.

1. If your last name has more than five letters, draw a square on your paper.
2. If your last name less than seven letters, draw a circle.
3. If you are wearing anything green, add 3 + 2.
4. If the teacher is tapping their pencil, draw an 'X'.
5. If the teacher is in the front of the room, fold your paper in half.

Prompt: When we program, we give the computer instructions on what to do. How are the instructions you just followed different from the instructions that we have been giving in Game Lab?

Think-Pair-Share: After students have had a chance to write down their answers, allow them to talk to a partner before sharing out as a class.

Discussion Goal

This discussion should prepare students to investigate programs that react to changing conditions. Students should realize that there is no "one way" that these questions will be answered, and that the answers will change according to changing conditions.

Remarks

> These instructions were a little different because we first had to decide whether something was true or not before we knew what we should do. In programming, instructions that depend on whether or not something is true are called **conditionals**, and the thing that is checked is called the **condition**.

> something is true are called **conditionals**, and the thing that is checked is called the **condition**.
>
> Conditionals are especially useful when we want the program to react to situations that change while the program is running.

Key Vocabulary:

- **Condition** - Something a program checks to see whether it is true before deciding whether to take an action.
- **Conditionals** - Statements that only run under certain conditions.

Question of the Day: How can programs react to changes as they are running?

Activity (35 minutes)

Conditionals

Transition: Send students to Code Studio.

1 **Prediction**

Questions to Consider with Video:

- What is a Boolean expression?
- What's an expression that would evaluate to true?
- What's an expression that would evaluate to false?

2 **Video: Booleans**

Discussion Goal

Goals:

Key Vocabulary:

- **Boolean Expression** - in programming, an expression that evaluates to True or False.

Students should be able to explain that a Boolean expression is something that is either true or false, similar to a yes or no question. The more formal way to say this is that Boolean expressions evaluate to either true or false. That means that when the computer processes a Boolean expression, it checks to see whether the expression describes a situation that is true or false, and then uses the value of either true or false wherever the expression is found.

Some examples of Boolean expressions that evaluate to true are `3 > 1` and `4 <= 7`, but press students to think of expressions that might be better represented by variables, such as `studentAge < 70` or `sizeOfClass > 2`.

Some examples of Boolean expressions that evaluate to false are `4 == 7`, `schoolName == "Hogwarts"`, and `currentYear < 1000`.

3 Quick Check

4-5 Skill Building

4 5

Question to Consider with Video:

- When would you want to use an if statement?

6 Video: Conditional Statements

Discussion Goal

Goals:

The broad point of this question is that programmers use if statements when they want the program to run differently in response to different situations. Encourage students to think of particular situations in which this would be the case. For example, they might want their characters to move faster when a "bonus" is in effect, or maybe they want more enemies to appear when the player reaches a certain level. Maybe they want the program to react in some way if a user presses a key or clicks the mouse, or they want a character to change animations if it touches a particular item.

7 Skill Building

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

8 Practice

9 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

10 Challenges

Wrap Up (5 minutes)

Considering Conditions

Question of the Day: How can programs react to changes as they are running?

Key Vocabulary:

- **Condition** - Something a program checks to see whether it is true before deciding to take an action.
- **Conditionals** - Statements that only run under certain conditions.
- **Boolean Expression** - In programming, an expression that evaluates to True or False.

Prompt: Now that you know how conditionals work, where you do think that they are used in games or other programs and apps that you already use?

Discuss: Have students share responses.

Discussion Goal

This discussion should get students thinking about how conditionals are used in games they have already seen, and help them connect those ideas to how they might want to use conditionals in their own programs. Student responses might include:

- If my username and password are correct, log me into Facebook
- If Pacman has collected all the balls, start the next level
- If my keyboard or mouse hasn't moved in 10 minutes, turn on the screensaver