Lesson 4: Shapes and Parameters

45 minutes

Overview

In this lesson, students continue to build skills and develop their familiarity with Game Lab by manipulating the width and height of the shapes they use to draw. The lesson kicks off with a discussion that connects expanded block functionality (e.g. different sized shapes) with the need for more block inputs, or "parameters". Students learn to draw with versions of `ellipse()` and `rect()` that include width and height parameters. They also learn to use the `background()` block. Throughout the lesson, students will need to reason about the x-y coordinate plane, consider the order of their code, and slightly increase their programs' complexity.

Question of the Day: How can we use parameters to give the computer more specific instructions?

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming

Agenda

Warm Up (5 minutes)
- **Shapes of Different Sizes**

Activity (35 minutes)
- **Programming with Parameters**

Wrap Up (5 minutes)
- **Journal**

Objectives

Students will be able to:

- Use and reason about drawing commands with multiple parameters

Preparation

- Review the level sequence in Code Studio
- Check the **"Teacher's Lounge"** forum for verified teachers to find additional strategies or resources shared by fellow teachers
- If you are teaching virtually, consider checking our **Virtual Lesson Modifications**

Links

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

For the teachers

- **Shapes and Parameters** - Resource
- **Shapes and Parameters** - Slides
  ▼ Make a Copy

Vocabulary

- **Parameter** - Additional information provided as input to a block to customize its functionality

Introduced Code

- `background(color)`
- `ellipse(x, y, w, h)`
- `rect(x, y, w, h)`

Teaching Guide

Warm Up (5 minutes)

Shapes of Different Sizes

Prompt: The `rect` block has two inputs that control where it's drawn - the x and y position. If you wanted these commands to draw different sizes of rectangles, what additional inputs would you need to give these blocks?

Discuss: Students should brainstorm ideas silently, then share with a neighbor, then share out with the whole class. Record ideas as students share them on the board.

Discussion Goal: This discussion introduces the vocabulary word "parameter" and also helps students understand the need for parameters. Students will be seeing versions of the `ellipse()` and `rect` block in this lesson that have additional parameters. Students may say they want inputs for the size of the shapes, their color, etc. During this conversation tie the behaviors the students want to the inputs the block would need. For example, if you want rectangles to be a different size, the block will need an input that lets the programmer decide how large to make it.

Remarks

> If we want our blocks to draw shapes in different ways they'll need more inputs that let us tell them how to draw. The inputs or openings in our blocks have a formal name, Parameters, and today we're going to be learning more about how to use them.

Key Vocabulary: Parameter - Additional information provided as input to a block to customize it's functionality

Question of the Day: How can we use parameters to give the computer more specific instructions?

Activity (35 minutes)

Programming with Parameters

Group: Put students into pairs for the programming activity.

Transition: Move students onto Code Studio.

Teaching Tip

Guide to Programming Levels: Additional guidance for programming levels is provided in the **CSD Guide to Programming Levels**. This document includes strategies and best-practices for facilitating programming levels with students.

1 **Predict**

Discussion Goal: The new parameters in this level will change the width and height of the rectangle block. Some students may guess this, but even if students are unable to make the initial prediction, that's okay! Once they run their code, they should be able to look at the result and notice that the rectangles are longer and wider than before. Using this new information, they should revisit their initial prediction and adjust it with this new information.

2-6 Skill Building

2 3 4 5 6

Teaching Tip

Normalizing Mistakes and Supporting Debugging: As programming levels become more complex, students may find themselves with bugs in their code that they need to untangle. If this happens frequently, this can be a demoralizing experience for students and can affect their self-perception of how capable they are in class.

To counter this, we recommend normalizing bugs and mistakes as something that happens to everyone - it's just part of the process. You can show students our **Debugging Video**, which includes several students normalizing mistakes and discussing debugging strategies that students can use. Additionally, consider displaying the **Student Guide to Debugging** for students to reference throughout the unit and having **Bug Report Quarter-Sheets** available for students to use.

7 Practice

8 Assessment

Assessment Opportunity

Formative Assessment: This level can be used as a formative assessment. A rubric is provided in the level, and written feedback can be given to students. **Click here to learn more about giving feedback to students**.

9 Challenges

Wrap Up (5 minutes)

Journal

Question of the Day: How can we use parameters to give the computer more specific instructions?

Prompt: You use parameters to control your shape's location and size. Can you think of any other situations in which parameters might be useful?

Share: Allow students to share out their ideas.

Discussion Goal: Student answers will vary, but they should all follow the pattern of giving more specific information about how to carry out a task. If students have trouble thinking of something, you may want to give some examples of things a computer could do, such as ring an alarm (with a parameter for a certain time) or play a song (with a parameter for the song title).

If you are interested in licensing Code.org materials for commercial purposes **contact us**.