

Lesson 3: Drawing in Game Lab

Overview

Question of the Day: How can we communicate to a computer how to draw shapes on the screen?

Students are introduced to Game Lab, the programming environment for this unit, and begin to use it to position shapes on the screen. They learn the basics of sequencing and debugging, as well as a few simple commands. At the end of the lesson, students will be able to program images like the ones they made with the drawing tool in the previous lesson.

Purpose

The main purpose of this lesson is to give students a chance to get used to the programming environment, as well as the basic sequencing and debugging that they will use throughout the unit. Students begin with an introduction to the GameLab interactive development environment (IDE), then learn the three commands (`rect` , `ellipse` , and `fill`) that they will need to code the same types of images that they created on paper in the previous lesson. Challenge levels provide a chance for students who have more programming experience to further explore Game Lab.

Assessment Opportunities

1. **Use a coordinate system to place elements on the screen.**

See level 9 in Code Studio, in particular the placement of the square in the picture.

2. **Sequence code correctly to overlay shapes.**

See level 9 in Code Studio, in particular that the square is displayed in front of the circles.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- **AP** - Algorithms & Programming

Agenda

Lesson Modifications

Objectives

Students will be able to:

- Sequence code correctly to overlay shapes.
- Use a coordinate system to place elements on the screen.

Preparation

- Read the Forum
- Prepare projector or other means of showing videos if you wish to watch as a class

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
- **Drawing Shapes** - Resource

For the students

- **Drawing in Game Lab - Part 1** - Video (**Download**)
- **Drawing in Game Lab - Part 2** - Video (**Download**)

Vocabulary

- **Bug** - Part of a program that does not work correctly.
- **Debugging** - Finding and fixing problems in an algorithm or program.
- **Program** - An algorithm that has been coded into something that can be run by a machine.

Introduced Code

Warm Up (5 minutes)
Programming Images

Activity (35 minutes)
Simple Drawing in Game Lab
Share Drawings

Wrap Up (5 minutes)

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

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)

Programming Images

Prompt: Based on what you know about computers, what do you think will be different between telling a person about your image and telling a computer about your image?

Share: Allow students time to think individually and discuss with a partner, then bring the class together and write their ideas on the board.

Discussion Goal

Goal From Unit 2, students may remember that a computer can only understand what to do if you use a particular language in order to communicate with it. For Web Development, that language was HTML. Computers can't "figure out" what you mean in the same way that a human can, and they are usually much more specific in how they follow instructions.

Remarks

In order to give instructions to a computer, we need to use a language that a computer understands. In the last unit, we used HTML, which is great for making web pages. To make our animations and games, we will use a version of Javascript that uses blocks. The environment that we'll be programming in is called Game Lab.

Question of the Day: How can we communicate to a computer how to draw shapes on the screen?

Activity (35 minutes)

Simple Drawing in Game Lab

Group: Place students in pairs to program together.

Teaching Tip

Pair programming is a great way to increase student confidence and foster the practices of collaboration and communication. You can read more about how to use Code Studio's pair programming feature [here](#).

Transition: Send students to Code Studio.



1

Introduction to Game Lab

💡 Teaching Tip

Tour of Game Lab

Depending on the age and comfort level of your students, you may choose to use this level to tour the environment as a whole class. Make sure that students can find the level instructions, coding area, display area, and block drawers. This is also a good opportunity to point out some of the useful resources like documentation and the blocks to text button.

Questions to Consider with Video:

- Where can you find more information about how to use the blocks?
- What's an advantage of using block mode?



2

Video: Drawing in Game Lab - Part 1

💬 Discussion Goal

Goal

Make sure students understand how to access the block documentation by clicking on the blocks inside the toolbox and clicking "see examples".

As students think of ideas of why they might prefer to use block mode, make sure that they understand that the block-based version of the programming language is just as legitimate as the text-based version. Students may offer that blocks make it easier to remember the exact commands or that they don't have to worry about the details of the parentheses or semicolons.

(Alternatively, advantages to text might be that it's easier to edit or that the text takes up less space.)



3

Using the Grid

Questions to Consider with Video:

- What's the difference between stroke and fill?



4

Video: Drawing in Game Lab - Part 2

💬 Discussion Goal

Goal: Stroke controls the border color of the shape, and fill controls the color inside of it.



5-7

Skill Building

5

6

7



8

Practice

💡 Teaching Tip

Level 8a: This debugging task tests the student's ability to sequence code correctly.

Level 8b: This debugging task helps students understand how to place elements on a screen using a coordinate plane.

Level 8c: This debugging task tests the student's ability to sequence code in regards to the `fill` command.



9



Assessment

✔ Assessment Opportunity

This is a good puzzle to use as an assessment of whether or not students understand the basics of sequencing and positioning shapes in Game Lab.



10

Challenges

💡 Teaching Tip

Level 10a: This task introduces students to the `point` block.

Level 10b: This task gives students practice with the `strokeWeight` block.

Level 10c: This task gives students practice with the `noFill` block.

Support: As students work on the levels, you can help them, but encourage them to try to spend some time figuring things out themselves first. If you need help supporting students, see the exemplars in the teacher answer viewer. When students hit the **challenge levels**, they can choose to pursue one or more of the challenges, return to improve upon previous levels, or help a classmate.

💡 Teaching Tip

Students that are new to programming often have some common misconceptions they run into. In order to prevent those, keep reminding students about the following things:

- One command per line

- Commands run in order from top to bottom
- Order of inputs into shape commands matter
- Each input into shape commands are separated by commas
- (0,0) is in the upper left corner of the display
- All x and y values on the display are positive

Share Drawings

Share: Once students have completed their drawings, have them share them with the class. One way to do this is with a gallery walk.

Wrap Up (5 minutes)

Question of the Day: How can we communicate to a computer how to draw shapes on the screen?

Prompt: Today you learned how to draw in Game Lab for the first time. What type of advice would you share with a friend who was going to learn about drawing in Game Lab to make it easier for them?

Share: Allow students to share out their responses with the class.