

Lesson 12: The Draw Loop

45 minutes

Overview

In this lesson, students are introduced to the draw loop, one of the core programming paradigms in Game Lab. To begin the lesson students look at some physical flipbooks to see that having many frames with different images creates the impression of motion. Students then watch a video explaining how the draw loop in Game Lab helps to create this same impression in their programs. Students combine the draw loop with random numbers to manipulate some simple animations with dots and then with sprites. Students should leave the lesson understanding that the commands in the draw loop are called after all other code but are then called repeatedly to create animation. Students will have a chance to continue to develop an understanding of this behavior in the next two lessons, but laying a strong conceptual foundation in this lesson will serve them well for the rest of the unit.

Question of the Day: How can we animate our images in Game Lab?

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- AP - Algorithms & Programming

Agenda

Warm Up (5 minutes)

Activity (35 minutes)

Wrap Up (5 minutes)

Objectives

Students will be able to:

- Explain how the draw loop allows for the creation of animations in Game Lab
- Use the draw loop in combination with the `randomNumber()` command, shapes, and sprites to make simple animations

Preparation

- Print and assemble the manipulatives
- Prepare the video
- 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

- [Flipbook Example - by Marnic Bos](#) - Video
- [Random Dot Flipbook](#)
- [Random Sprite Flipbook](#)
- [The Draw Loop](#) - Resource
- [The Draw Loop](#) - Slides

▼ Make a Copy

For the students

- [Introduction to the Draw Loop](#) - Video ([Download](#))

Introduced Code

- `World.frameRate`
- `function draw() {}`

Teaching Guide

Warm Up (5 minutes)

Display: Show [Flipbook Example - by Marnic Bos - Video](#).

Prompt: This video shows a flipbook to make animation. In your own words, how is it working? Why does it "trick our eyes" into thinking something is moving?

Discuss: Have students write their ideas independently, then share with partners, then share as a whole group.

Discussion Goal: This discussion should introduce some key understandings about animation. Students should understand that the key is seeing many pictures in a row that are slightly different. Introduce the vocabulary word "frame" as being one of those many pictures--a single image within an animation. Then transition to the fact that soon students will be creating animations of their own.

Remarks

We're going to start learning how to make animations, not just still images. In order to do this we need a way to make our programs draw many pictures a second. Our eyes will blur them together to make it look like smooth motion. To do this, though, we're going to need to learn an important new tool.

Question of the Day: How can we animate our images in Game Lab?

Activity (35 minutes)

 **Video:** Show students the [Introduction to the Draw Loop](#) video in the slides.

Teaching Tip

To encourage active engagement and reflection, use one or more of the strategies discussed in the [Guide to Curriculum Videos](#).

Question to Consider with Video:

- What does the draw function do?

Discussion Goal: The draw function runs the same code over and over in a loop, which is why it's often called "the draw loop". This creates an animation effect by changing what is drawn by a small amount each time the draw function runs. This can be compared to a physical flip book, which can animate an image by changing it slightly on each page.

Misconception Alert!

When the draw loop runs, it does not "clear" out the previous drawing, so it will continue to show anything that has not been covered up by the new draw loop. If students do not add a "background" at the beginning of the draw loop, they will still see all the images drawn to the screen previously.

Demo: If you choose, use the provided manipulatives to provide a hands-on demo for exploring how the draw loop and animations are connected.

Do This: Send students to Code Studio.

The first prediction level program is the first time they will see the "draw loop" in action.

Discuss: Students should consider and discuss:

- What will be drawn on the screen do and why?

Discussion Goal: Predictions will vary but after viewing the Draw Loop video, students should be able to predict that the program will draw an ellipse over and over and over to the screen. Students may also predict and should notice once they run the program that the ellipse is drawn in random locations on the screen. One last thing to point out is the fact that the code to give the ellipse a color is *outside* the draw loop and runs *before* the program enters the draw loop.



1

Predict



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.



2

Skill Building

The second prediction level program has one small difference that will make it run a little differently.

Discuss: Students should consider and discuss:

- What will this program do?

Discussion Goal: Notice that the blue background is never visible because it is immediately drawn over by the red background. Also, there's only ever one yellow dot visible since as the draw loop runs over and over it is placing down new backgrounds.



3

Predict



Teaching Tip



Inside versus outside the draw loop

This is a good time to make sure students understand that code **outside** the draw loop is used to **set up the program**. It is for how you want your program to start. Code **inside** the draw loop is for things that are **changing as the program is running**.



4-5

Skill Building

4

5



6

Practice



7



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.](#)



8

Challenges

Wrap Up (5 minutes)

Question of the Day: How can we animate our images in Game Lab?

Prompt: How does the draw loop help us make animations?



Assessment Opportunity ▲

Key Understandings: There are many common misconceptions with the draw loop. Make sure students understand the following:

- The draw loop is run after all other code in your program. It does not actually matter where it is located in your program.
- The draw loop is run by Game Lab at a constant frame rate of 30 frames per second. You do not actually need to call the function yourself.
- The "frames" in Game Lab can be thought of as transparency sheets. Unless you draw a background, then all your new shapes or sprites will simply appear on top of your old ones.
- You should only have one draw loop in your program.

Review: Return to the resources students saw at the beginning of the lesson (Video, physical flip books) or under Help and Tips during the lesson. Address misconceptions that have arisen in the lesson.



This work is available under a [Creative Commons License \(CC BY-NC-SA 4.0\)](#).

If you are interested in licensing Code.org materials for commercial purposes [contact us](#).