FIRIA LABS

MISSION 3: Light Show

Time: 30 (to 45) minutes

Overview:

This mission will teach students to turn on (and off) the four NeoPixels on the CodeX. They can be set to any color using RGB triplets, called tuples. The basic lesson will use built-in colors.

As an extension, students can learn about RGB lights in the pixels and how to control the amount of each light. Then students can modify their code to make the NeoPixels any color, not just built-in colors.

Objectives:

- I can assign **data** to a **variable**.
- I can use **variables** to make code more efficient.
- I can set the pixels using built-in colors.

Standards:

2-CS-03 Systematically identify and fix problems with computing devices and their components

2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.

2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.

CSP Framework:

Computational Thinking Practices:

4.C Identify and correct errors in algorithms and programs, including error discovery through testing.

6.A Collaborate in the development of solutions.

Key Concepts:

- Follow instructions in the **Objective panel** carefully.
- The CodeX has four NeoPixels that can turn on any color represented by RGB.
- The NeoPixels are numbered 0, 1, 2 and 3.
- When setting a pixel, the pixel number, color and optional brightness need to be given.
- A variable can be defined and used in code.

Preparation:

Make a copy of the assignment or put it in the LMS.
Prepare any formative assessments you want to use in the wrap-up
Each student/pair needs

- a computer/ Chrome
- a CodeX & USB cable
- assignment guide

Links:

- Mission reminders slides
- Assignment
- CodeX and RGB slides
- Color picker: https://htmlcolorcodes.com/color-picker/
- Programming journal
- Daily reflection form (use your form link)

Agenda:

- Warm-up (5 minutes)
- Mission 3 (20 minutes)
- Extension (15 minutes)
- Wrap-up & Assessment (5 minutes)

Vocabulary:

- **RGB:** Red, Green, Blue; the colors that make up a single pixel on the screen
- **Sequential:** Executing code line by line, one after another, in order
- **Literal:** a specific value, like 1 or “hello”
- **Variable:** a name you assign to some data that you use in code instead of the literal, or actual values
- **Assign:** bind a name to a value; give a variable a value

Assessment:

- Daily reflection journal or Google form (use the link to your form)
- Review Kahoot: https://create.kahoot.it/share/firia-labs-ap-csp-mission-1-2-3/5be3baab-3370-49ae-8912-adf30394f2bd
- More suggestions listed below in the Walk-Through Wrap-up

Teaching Guide

Warm-up (5 minutes)

🧑‍🤝‍🧑 **Discuss** – Use a discussion strategy, like journaling, working at boards, selecting random students, or a form of think-pair-share.

Talk to students about primary colors. They will probably say “red, yellow and blue.” Mention that those are primary colors for paint, but for light primary colors are red, green and blue. All color on your screen is made by mixing those three lights.

💡 **Teaching tip – Reminders for the beginning of mission:**
These reminders are organized on a short slide deck that can be shown to students at the beginning of class

- Always start a new program by creating a new file and naming it appropriately. If you don’t, you will lose all your previous work! Using descriptive file names is essential to finding the program later!
- You are making a project – not just working random problems. Focus on the *project-based* objectives and avoid rushing through the material too quickly.
- Test your understanding along the way by “coloring outside the lines”. Try stuff!
- Collect all the Tools you find! (They’re indicated with a wrench icon)
- Read carefully – usually the answer is right there in front of you!

Activity – Mission #3 (20 minutes)

💻 Randomly group students into pairs for pair programming.

Students log in to one computer. Two computers can be used if they want to see instructions on one computer and work on the other computer. However, the assignment document requires snippets, so it will need to be open on the same computer as CodeSpace.

Students go to sims.firialabs.com and should be at the beginning of Mission 3

💡 **Teaching tip – Objective 5 & 6:**

These objectives introduce the debugger, and then have the students use the debugger. This can be a little confusing the first time (and even third, fourth, etc.). You may want to demonstrate this with the students. The important thing to remember is that the code that is HIGHLIGHTED happens AFTER the next “step in”.

The students click debug once in Objective 5, but they have to click it AGAIN for Objective 6.

💡 **Teaching tip – Objective 7:**

This is a good time to remind students to keep track of their mistakes. The last question in the assignment is to record their bugs and how they fixed them.

💡 **Teaching tip – Objective 8:**

Students will create a variable and use it in code. They may have trouble realizing they need to use it in four places – everyone there is a sleep() command.

💡 **Teaching tip – Objective 9:**

Students will create another variable for the color and use it in code. They may have trouble realizing they need to use it after each group of pixels are set to a color. They can use CodeTrek for ideas, but their code doesn’t have to match it exactly.

Activity – Extension (15 minutes)

💻 Students remain with their partners and continue to work on the assignment (time permitting).

Use the slide deck “CodeX and RGB” to go through important vocabulary and show the code for setting your own colors.

- The slide deck includes a short video from Code.org that introduces RGB and pixels. It covers the values for RGB used in coding. Start at 0:45 and go to 2:35. https://www.youtube.com/watch?v=15aqFQQVBWU&t=88s
- Website for selecting a color: https://htmlcolorcodes.com/color-picker/

✅ Assignment is complete and ready to turn in. Both students should include their names on the document. Program can be turned in with a code snippet pasted to the assignment, or any other way you choose as a teacher (Google Classroom, observation, download file, etc.)

Wrap-Up (10 minutes)

✅ **IMPORTANT!!**

Show students how to “clear” their CodeX by creating a program file without code. **These instructions are included in the Clearing the CodeX slide deck (and also at the end of CodeX and RGB slides, in case you already have them open).**

- Students create a new file called “Clear”
- Students type the code to clear the display
- Run the code at the end of the class period to clear the code for the next person.

✅ If this lesson is completed in one class period, the following can be used as a wrap-up. If you are on a block schedule and continuing to the next lesson, a wrap-up isn’t necessary.

Formative Assessment:

- Daily reflection journal or Google form (use your own link)
- Kahoot (in class or individual): https://create.kahoot.it/share/firia-labs-ap-csp-mission-1-2-3/5be3baab-3370-49ae-8912-adf30394f2bd
- Exit ticket on vocabulary (RGB, literal, variable)
- Group review on vocabulary (RGB, literal, variable)
- Students add vocabulary words to their programming journal, and they can start their debugging chart
- Students add to their vocabulary canvas with vocabulary words (RGB, literal, variable). This concept is adapted from Code.org AP CSP curriculum. The vocabulary canvas slide deck is created by Code.org.

SUCCESS CRITERIA:

□ Define RGB, literal, and variable
□ Define and use a variable used in sleep()
□ Define and use a variable for color that is changed and used multiple times
□ Debug any errors in the code and keep a debugging table
□ Write a program, run it, and save it to the CodeX