

Lesson 5.3: 13 Colonies - Display a Colony Name

Objectives

In this lesson, students will:

- ❖ Practice assessing the status and next steps of a larger programming project
- ❖ Learn to approach a more complex project with planned tasks and algorithms
- ❖ Practice iteratively building upon smaller programming steps to create a large program
- ❖ Learn to identify which sprite should contain which script
- ❖ Practice using list commands

Agenda

- | | |
|-----------------------------|---------|
| 1. Status and Coding Plan | 10 mins |
| 2. Display a Colony Name | 30 mins |
| 3. CWrap Up and Reflections | 10 mins |

Resources & Links

- ❑ Lesson 3 starter project (in case students were absent or could not finish the prior lesson (lesson 2)): <https://scratch.mit.edu/projects/545830212>

Preparation

- ❑ Projector for class demonstration and instruction
- ❑ Print student activity worksheet, one per student pair

1. Status and Coding Plan



Display your screen and discuss the status of the program and next steps using the Colonies Coding Plan.

Colonies Coding Plan:	
Task	Next Steps Algorithm:
Game Setup: Create the Game Instructions	Done
Game Setup: Create the Colony Labels	Done
Create needed variables: score and the list of colonies	Done
Initializations	Done
Display Colony Name	<ol style="list-style-type: none"> 1. When the spacebar is clicked: 2. If the list of colonies is not empty, <ul style="list-style-type: none"> Pick a random colony to be displayed 3. Display the colony name 4. If the list is empty, check player's score

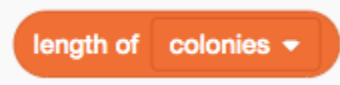
2. Display a Colony Name



Prompt students to answer these questions given the **colonies** list. You can optionally do it as an exercise for all students to do on their own. Review the answers with students.

colonies	
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
+ length	13 =

What do these return ?

1. 

2. 

3. 



4. Which item is deleted by this block?



If you run the block a second time, which item is deleted?

(It might be helpful to demonstrate this. You can quickly create this list in the lesson 2 project by clicking on the script that initializes the list)

Answers:

1. 13
2. A random number between 1 and 13
3. It returns the item from the colonies list at a random location between 1 and 4. If random returns 3, it returns 3. The item in the list at location 3 which also happens to be the number 3. Explain that when the list is full, it seems that the counter number is always the same as the item. But what if we remove items from the list. The item at location 3 may be 5.

- The value 3 is deleted (representing colony # 3). What item is now at location 3?



Demonstrate the code to display a colony label in the **Game Instructions** sprite by remixing the Lesson 2 project. Explain each instruction as you go prompting students to help you code it.

- When the spacebar is pressed, we want to hide the game instructions because we are going to show a label.

- We use an **if-then-else** conditional statement to check if we still have colonies to display. If the colonies list is not empty, meaning we still have colonies to display, we pick a random one to display, if we don't have colonies left in the list, we display every single one and the game ends after we check the score. The **else** part will be coded later.

- Check if the length of the colonies is greater than 0 (meaning we still have colonies to display).
- If True, set a random number between 1 and the length of the colonies list to the variable counter. This is the item #.
- Set the **colony** variable to the colony number from the list at the item # in counter (this is a mouthful, so it is helpful to break it down).
- Let the colony **Label** sprite know it is time to show the chosen colony name by a broadcast message.

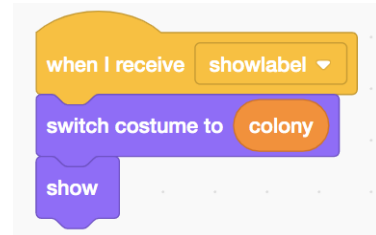
What do you think happens when the length of the list is 0? When the length of the list is 0, it means the list is empty and every colony name was displayed.

Time to check the player's score. If the player's score is 13, they won because they identified all colony locations correctly. We also need to do something if they did not score 13, so we use a conditional

statement with 2 options. Which block will serve that purpose? An IF-THEN-ELSE block.

We will add the winning code later after we add the code to keep score.

Finally, when the colony label receives the broadcast to show one of it's labels (costume), switch the costume to the value in **colony**. That is why the order of the colony names is important, so that the colony number is matched up with its location.



Student Activity: Display a Colony Name



Explain to students that in this activity they will be given pseudocode to guide them in writing the code for the activity. Pseudocode is a notation resembling a simplified programming language. It is often used by computer scientists to write down an algorithm without being tied to a particular programming language.

Instruct students to code and test the script to display the colony name. (The pseudocode is available for display in Exhibit A)

Remind students to think about which sprite each script should be added to.

After the activity, **review** the solution code with students.

3. Wrap Up and Reflections



Reflection Points:

- What was fun about today's lesson?
- How does an **if-then-else** statement work?
- What does the pick random block allow you to do?
- Which list commands did you use today?
- How was the plan helpful during today's lesson?

Exhibit A

(you must choose the correct sprite for each script)

Script 1

When the spacebar is pressed:

If the list of colonies is not empty,

- Set the variable **counter** to a random number between 1 and the length of the **colonies** list. (this is the item #)
- Set the colony variable to the colony number from the **colonies** list at the item # stored in **counter**. (hint: you need 2 Scratch blocks and the **counter** variable)
- Broadcast a message “showlabel”

Else

 If score = 13 then

 <code to be added later>

 else

 <code to be added later>

Script 2

When this sprite receives the message “showLabel” :

- switch to the costume number corresponding to the value of “colony”
- show the label