# Lesson 5.5: 13 Colonies - Final Touches

## Objectives

In this lesson, students will:

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

## Agenda

| | |
|---|---|
| 1. Status and Coding Plan | 10 mins |
| 2. Display a Colony Name | |
| 3. Student Activity: Fun | 20 mins |
| Facts for All Colonies | 15 mins |
| 4. Wrap Up and Reflections | 10 mins |

## Resources & Links

❏ Lesson 5 starter project (in case students were absent or could not finish the prior lesson): https://scratch.mit.edu/projects/545825110

## Preparation

❏ Projector for class demonstration and instruction

## 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 | Done |
| Check if the user clicked the correct colony | Done |
| Keep Score and Display Fun Facts for Massachusetts | Done |
| Keep Score and Display Fun Facts for All Colonies | Done |
| **Respond to 'success':** | When the correct colony is identified:<br>1. The Label sprite hides<br>2. Delete the currently displayed colony from the colonies list so it does display again<br>3. Reset the counter variable so users cannot repeatedly click on the colony to increase their score |
| **Winning the Game** | When the player wins the game, make something fun happen. |

## 2. Respond to "success"

**Engage** students in an interactive demonstration and instruction:

Which sprites need to respond to the "success" broadcast message because the correct colony was identified?

1. The "Labels" sprite:   It must hide the label
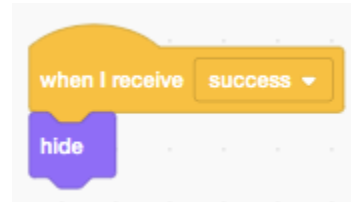


2. The "Game Instructions" sprite must delete the currently displayed label and reset the counter.

Why this sprite?

The *Game Instructions* sprite has all the initialization code for the list and we need to update our lists.  Therefore, it makes sense to make any list changes in the *Game Instructions* sprite.

What to do *When I receive success*:

**First**, If we choose a random colony from the colonies list, is it possible that the same colony could be picked?    Yes, the random number generator could pick the same number.
To avoid this,  when there is a "success", we delete the colony number from the list so it is not displayed again after the user successfully picks it.

In the "Game Instructions" sprite add this code:
The counter variable holds the item number of the colony displayed.  The delete block deletes that item, the colony number, from the list.



**Second**, The value of colony must be set to 0 again to prevent the player from boosting their score artificially by clicking "Massachusetts" sprite repeatedly.
You can invite students to try this during the activity before adding the code.

**Student Activity: Respond to "success"**

Students code the scripts for the response to the "success" broadcast message.

Display the following pseudocode (available in Exhibit A) and instruct students to write the code for the given pseudocode. You can also print and distribute the pseudocode.

For the *Labels* sprite:
> When the sprite receives the "success" broadcast message
> > Hide

For the *Game Instructions* sprite:
> When the sprite receives the "success" broadcast message
> > Delete the currently displayed colony (stored in *counter*) from the *colonies* list
> > set counter to 0

## 3. Student Activity:   Winning the Game

**Engage** students in a brief class discussion followed by a discussion between project partners.

What do you want to see when the player wins?
What do you want to happen when the player did not score 13 points?
Maybe you want somebody to win with 12 or 11 points?

Here are some suggestions when the player wins:
> Say "You won. Congratulations!"  and reset to start
> You can create a sprite that does something
> You can change the backdrop to something you create

**Tell students to discuss with their partner** some ideas of what to do when the player wins and what to do if the player does not win.

Time to get started!
**Tell** students to open their project and code their ideas to win the game or to try again.
Students should then play their game repeatedly to test it and fix any remaining bugs.

You can use the list of ***Troubleshooting and Debugging Tips*** at the end to help students during the activity.

## 4. Wrap Up and Reflections

Tell students to share their project so that the team member whose Scratch account was not used can remix it into their scratch account.

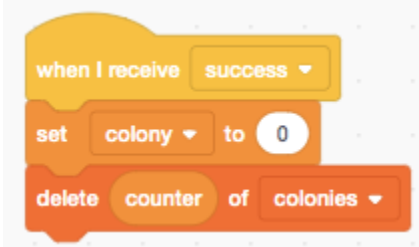| Reflection Points: |
| --- |
| <ul><li>What did you learn during this project?</li><li>What did you find challenging?</li><li>What was fun?</li><li>Describe the process of developing our project.<br>1. Specification- What needs to be done<br>2. Design the project incrementally<br>3. Write the algorithm for each step<br>4. Implement the algorithm<br>5. Test the implementation<br>6. Fix bugs<br>7. Repeat steps 5 and 6 until all bugs are fixed<br>8. Repeat steps 3-7 until project is complete as per specification<br>9. Enhance your project with new features<br>10. Share your project and continue to maintain your project</li></ul> |

Troubleshooting and Debugging Tips

| Error | Cause | Solution |
|-------|-------|----------|
| Score does not go up when the right colony is clicked | The colony=<number> boolean condition has an incorrect number | Input the correct number in |
| Colony labels that have already been identified are not deleted from the list | Check that the variable "counter" is deleted from list "colonies" when message "success" is received | |
| When we click on a colony repeatedly score keeps increasing | After the colony was correctly identified the value of "colony" should be initialised to 0 | |
| The fun fact of a colony won't show | The message sent corresponding to that colony was not received by the funfact sprite of the colony | Check if the funfact sprite of the colony is responding to the message received from the corresponding colony sprite. Also check the spelling and case sensitive nature of the message |
| Fun facts from some other colony also shows | The message is received incorrectly by the funfact sprite of the other colony | Correct the other funfact sprite to respond to the message received from its corresponding colony sprite only. For ex: Funfacts Massachusetts must respond to the Massachusetts message sent from the Massachusetts sprite, not any other sprite. |

Exhibit A

## For the *Label* sprite:

When the sprite receives the "success" broadcast message
- Hide

## For the *Game Instructions* sprite:

When the sprite receives the "success" broadcast message
- Delete the currently displayed colony (stored in *counter*) from the *colonies* list
- set counter to 0