Lesson 13: Introduction to Arrays

Adapted from Code.org curriculum

Objectives:

- Identify an array as a data structure to store lists of information in programs
- Create arrays and access information stored within them using an index
- Manipulate an array using the append, insert, and remove operations
- Account for the fact that JavaScript arrays are zero-indexed when using them in a program

Discussion:

- What might we use lists for?
  - To organize information
  - Collect relevant information in one place
  - To order or prioritize ideas

Activity:

- App Lab – Code Studio Stage 13

Introduced Code:

- removeItem(list, index)
- insertItem(list, index, item)
- list.length
- var list = [“a”, “b”, “c”, “d”];
- var x = [1, 2, 3, 4]
- appendItem(list, item)

Wrap-up:

- Your app needs to store the following information. Decide whether you would use an array or a variable to store it:
  - All the messages a user has sent
  - The highest score a user has ever reached on the app
  - A username and password to unlock the app
- In general, when do you think you should store information in an array, and when should you use a variable?

Key Points:

- Variables store single pieces of information, while arrays store many
- An array can grow in size to accommodate more information
- Arrays are slightly more complex to use than variables. If you are only going to be storing a fixed amount of information, it is probably appropriate to use multiple variables

Vocabulary

- Array – A data structure in JavaScript used to represent a list
- List – A generic term for a programming data structure that holds multiple items

Lesson 14: Building an App: Image Scroller

Adapted from Code.org curriculum

Objectives:

- Use an array to maintain a collection of data in a program
- Create apps that allow user interaction through key events
- Refactor code in order to appropriately incorporate new functionality while maintaining readability and consistency

Getting Started

- When we want to add new functionality to our programs, we'll of course have to write new code.
- Sometimes, when we add new code to an existing program, we'll also have to make changes to the original components of our program. Why might this be the case?
  - Contradicting code
  - Redundant components
  - New, better code
- Coding is an iterative process

Introduced Code:

- playSound
- onEvent(id, type, function(event)){…}
- setImageURL

Key Points:

- Refactoring is the process of changing the way we wrote old code in order to keep programs consistent and readable while incorporating new functionality
- It is possible that refactoring code will not change the user's experience but will make the program easier to read and maintain
- Refactoring is a useful process, but it can be time consuming and challenging. We'd ideally not refactor code very often but it is sometimes necessary
- Good planning and design can help avoid refactoring. Good use of functions and an organized program means that at the very least we limit areas that need to be changed

Vocabulary:

- Key Event – in JavaScript an even triggered by pressing or releasing a key on the keyboard. For example: “key up” and “keydown” are event types you can specify. Use event.key – from the “event” parameter of the onEvent callback function – to figure out which key was pressed

Lesson 15: Processing Arrays

Adapted from Code.org curriculum

Objectives

- Use a for loop in a program to implement an algorithm that processes all elements of an array
- Write code that implements a linear search on an unsorted array of numbers
- Write code to find the minimum value in an unsorted list of numbers
- Explain how binary search is more efficient than linear search but can only be used on sorted lists

Getting Started:

- Remember the FindMin problem you wrote an algorithm for back in Unit 3 (with the cards) with the Human Machine Language?
- Today we will use the common pattern of using a loop to visit every element in the list, rather than the jump command

Getting Started

- We will start with looking back at the Minimum Card Algorithm
  - Now you can write using pseudocode
- The same kind of thinking that went into designing this algorithm can be applied to making working code as well.
- Today you'll get some practice writing code with loops and if-statements to process a list – skills that will help you write you own algorithms for lists.

Activity 1: App Lab

- Code Studio – Stage 15

Activity 2: Unplugged

- We will use the "Card Search Algorithm" Activity Guide
- Note:
  - What you programmed was a "linear search"
  - This activity is a "binary search," which is faster, but requires you to sort first…

Wrap-up

- When you talk about how “long” or how much “time” an algorithm takes to run, time is usually a measure of the number of operations a computer needs to perform to complete the task.
- You can measure the amount of time it takes to run an algorithm on a clock, but it’s often not a useful measure, because the speed of the computer hardware obscures whether the algorithm is good or not.

5 Statements

- "4.2.4 Evaluate algorithms analytically and empirically for efficiency, correctness, and clarity."
- Come up with a brief (60 second) explanation of the statement and relate it to something you experienced as part of this lesson
- You will have 3 minutes to discuss
- Whip-Around

5 Statements

The For Loop

- Consolidates all of the pieces we need to keep track of a counter
  - Counter variable, incrementing, and boolean condition (in one line)

Introduced Code:

- for(var i = 0; i<4; i++){ //code}
- function myFunction(n){ //code }

Vocabulary:

- For loop – A typical looping construct designed to make it easy to repeat a section of code using a counter variable. The for loop combines the creation of a variable, a boolean looping condition, and an update to the variable in one statement