2A: In the creation of my app I used code.org's App Lab. The built in program language in the App Lab is JavaScript. The purpose of my program was to create a game that someone could play for entertainment purposes, that would allow you to race two cars. My program functions by allowing the player to select a color for their car, that information is then stored in a variable that is referred to each time that car is referenced. Next a random number is generated and the car travels that number of spaces. The end goal is to beat the white car or the "enemy" car to the end of the game board. My video illustrates how the game is played from start to end, and the two different ending scenarios (win or lose).

2B: I started by planning, then I created a basic game board, and an algorithm named "movePiece" which moved a single car. My first difficulty was when I tried testing with two cars. The cars would move forward based on the location of the opponent's car. This was because there was only one variable keeping track of the number of spaces the cars had moved. To fix this I created the variables "pawnTotal" and "enemyTotal" which stored a player's current location when it wasn't their turn. Next I created my functions "variableAssignment" and "movePiece". Near the end of my coding I had an issue with my "finish" function which checked if a car had reached the final space. This function checked the Y-position of the car, but I discovered it only recognized if the "enemy" car reached the end. I tested the function checking for the X-location of the cars, but since the cars passed the same X-location more than once this didn't work. I then realized that it didn't work originally because the cars ended at different Y-locations. I changed the algorithm so it checked if the cars were at either y=25 or y=50 and my program worked.

2C: My main algorithm is "movePiece" which includes the sub-algorithms, "variableAssignment" and "totalChange". This algorithm helps achieve my program's purpose by creating the movement of the cars and allowing them to race by switching which car is moving. Individually my algorithm "movePiece" uses a for loop to move the car from the current location (carTotal) to its end location (total). The "movePiece" algorithm uses logic to check the car's location so it can face right or left, to make sure that the cars move to the correct Y-position, and to ensure that the loop ends once the car reaches the end. On its own "variableAssignment" uses a logical if-else statement to check whose turn it is, and assign the values associated with that player to the variables, "car", "car2", and "carTotal". My algorithm "totalChange" uses logic to determine whose turn it is and store the end total (final location) into that player's individual total and set the "total" variable with the opponent's current location. Together "variableAssignment" provides the values for the variables in "movePiece", and "totalChange" rearranges the totals so that each individual total stores their current location, and "total" can be used to update the current player's new location.

2D: My algorithm "variableAssignment" is a sub-algorithm in "movePiece" and is very important to managing the complexity of my program. It is responsible for the reassignment of the values of the variables, "car" (right-facing car), "car2" (left-facing car), and "carTotal" (car location at the start of each turn) which are used in the algorithm "movePiece". This algorithm uses a logical if-else statement to check whose turn it is. If "turn" stores the value "enemy", then "car", "car2", and "carTotal" will hold the information associated with the "enemy" car. If that statement is false then the else statement will have "car" hold the ID associated with the color selected at the start of the game. The algorithm will also create the ID that refers to that same color car, but facing the left, by concatenating a "2" to the string and storing the ID in "car2". This algorithm is important in taking in information on whose turn it is, and creating and storing the correct IDs into variables that can be used in the "movePiece" function. Without this function there would need to be seven functions for movement of the cars, depending on whose turn it is and the color car selected.