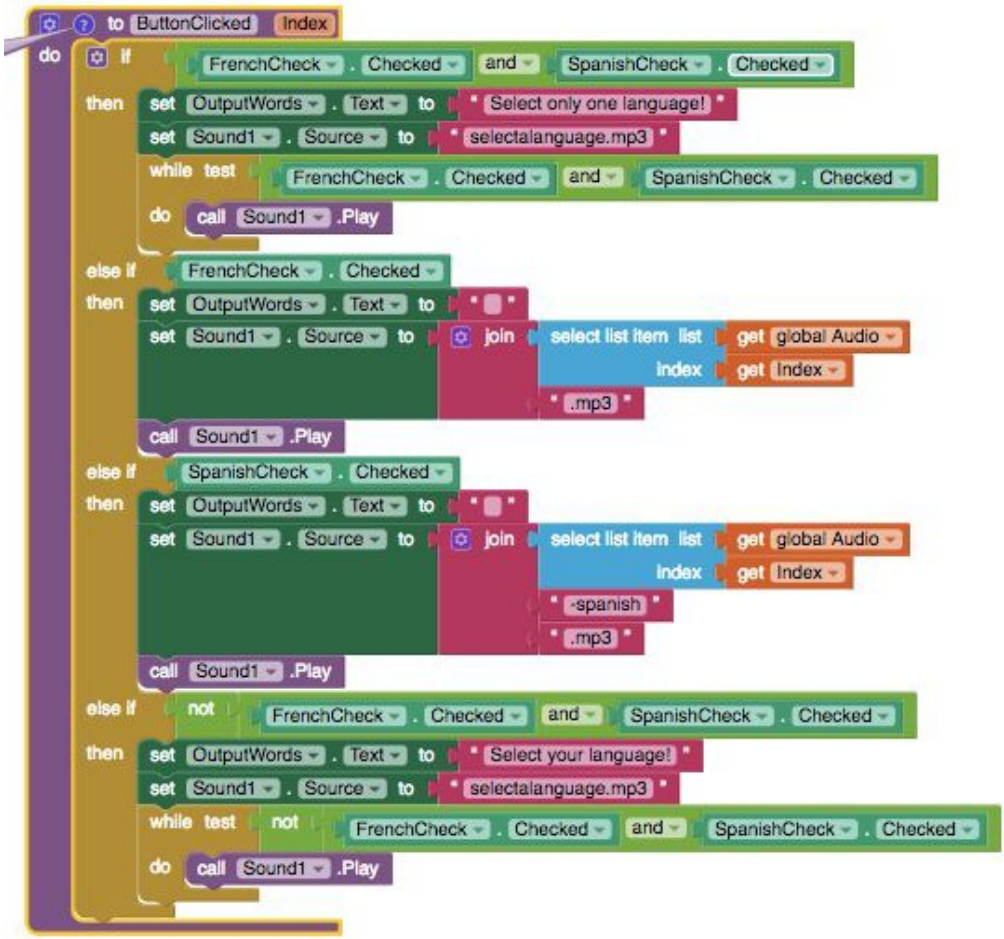
AP Computer Science Principles: 2020 Create Task
Student Sample A

3a.

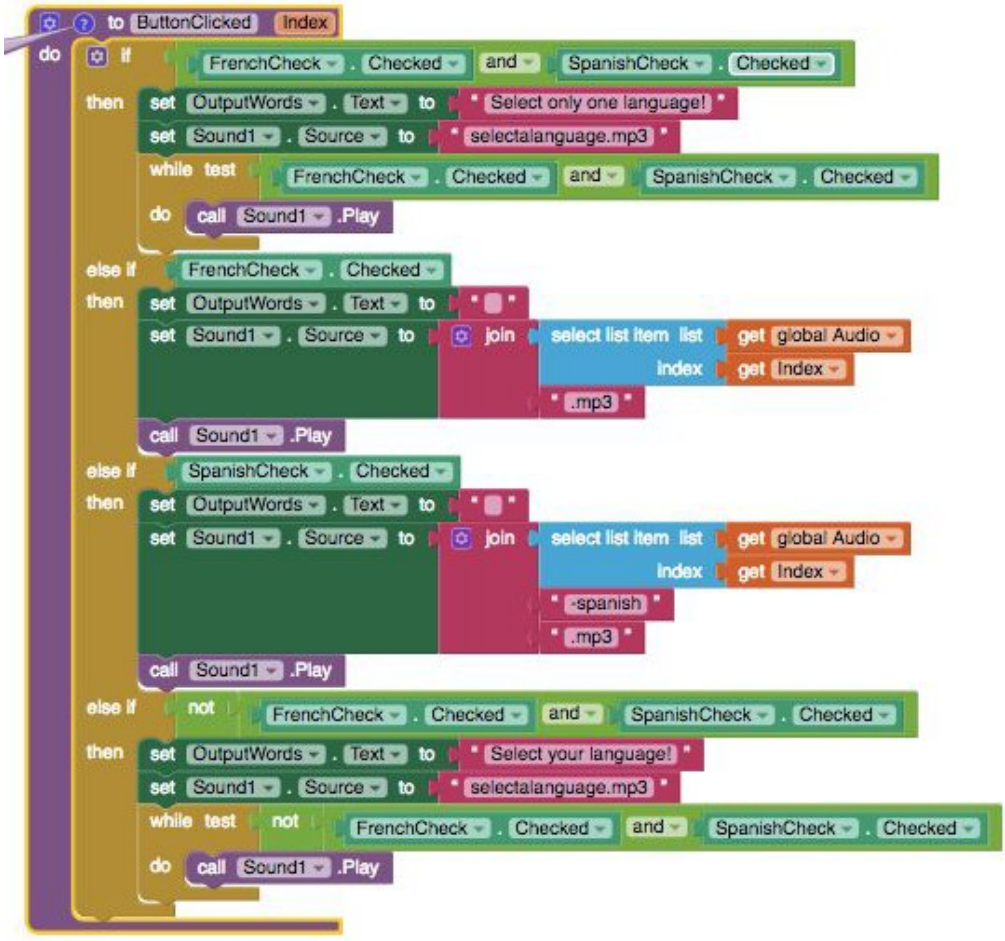
This program was created in MIT App Inventor to address the issue of learning new languages. Here it teaches the user how to say different colors, where the user inputs what language it wants to hear, either Spanish or French, and then taps on a color, prompting the program to output the audio for that certain color. This allows users to quickly learn how to say colors in another language through interaction and output of audio. In the video, it shows an example of the user clicking on the Spanish checkbox and playing the audio for red and blue. The user can hear what it sounds like, and thus learn how to say it correctly. If the user accidentally inputs no language or both, the program will catch the error and notify the user.

3b.



The data contained in the list Audio is the list of available color names. It represents all the colors a user can pick for the program in English. These are used to create the corresponding Spanish or French audio files based on which language is selected. When a button is pressed, it will get the English color word from the index of the color in the list. Here, the language the user has chosen does not matter. The program will then create the audio file name for the Spanish or French audio based on what language the user has check marked by manipulating the text (for example, adding“-spanish” to the end if they selected spanish) and then adding “.mp3” to the end in order to call the correct audio file. For the program to function without lists in general, each button will have to call the individual audio file, meaning we would need to have buttons for every color and language combination making the code and user interface more complex.

3c.



An example call to the ButtonClicked procedure. There is a call for each button.

This procedure helps to contribute to the overall functionality of the program by selecting the correct audio file based on what language and color the user has selected. Since this algorithm is needed every time a user presses a button, the procedure helps the overall efficiency of the program by having the code located in one location that the program repeatedly calls on.

When a user presses a certain color button, the button returns an index pertaining to the main audio list of colors. Then, the procedure takes the index of the color as a parameter and selects the respective list element, which is a string that contains the certain color. Then, depending on the language, the procedure will append a language identifier (for example, “-spanish”) and then add “.mp3”. It will then use this to call a certain color from the database of audio files that is named accordingly. The procedure is able to take parameters and inputs from the user and then output them as such by constructing certain audio file names and then pulling them from the database to play. This action demonstrates selection, and sequencing is when the procedure is able to order tasks accordingly and call from a list. Iteration is used when the procedure senses that there are no languages or both languages selected, and thus repeats the audio of “Select a language!” until the user does so.

3d.
The test cases are based on the conditions of what language is selected and what button for what color is pressed, which is represented by the parameter “index”. We would want to check the program for both Spanish and French. For the language Spanish, we could select one of the colors, for example Orange, and the program should give us the correct translation for Spanish. For the language French, we could select one of the colors, for example Orange, and the program should give us the correct translation for French. To further test the program, we should select another color, for example green, and the program should give us the correct translation for the pre-selected language.
Each of these test cases executes different parts of the algorithm, going by the condition of which checkboxes are checked for which languages.