In the App project my partner, Itaty Marquez, and I were tasked creating an app using the MIT app inventor. The app had to include notifier blocks, sound effects, background image/s, Procedures, a main screen, branding screen and about screen. We decided on making a quiz app that will challenge the player with five trivia questions about the Marvel Cinematic Universe and related Netflix shows. Challenges we overcame while designing our app include minimizing the number of screens used and randomizing question answers.

Originally I intended to have ten questions, but the MIT app inventor advises limiting the amount of screens to preserve functionality, so I had to find a way to fit all the questions into one screen. The solution to this was actually quite simple. First I declare a global level, starting at one. Then I call a procedure which uses if statements to search for the global level. When the screen initializes and calls the procedure it will recognize that the global level is one and call another procedure to set the textbox and buttons to the appropriate settings for question one. After calling question one the procedure will then add one to its total value. When the correct answer is chosen for question one it will run the procedure again, however, since the global level is set to two, the procedure will call question two’s procedure. It will work in this way up until question five when, once the correct answer is chosen the procedure will open the win screen. With this method all I have to do to create a new question it duplicate a previous question and it’s related procedures, variables and lists and change their data to match the new question.

A second obstacle we ran into while designing our app was randomizing the answers. The first thing I did was create a method for the correct answer to be placed randomly in one of the four buttons. I did this by first creating a variable with a random value og 1-4. Each number has a corresponding button. In an if statement the program will search for each number individually and, depending on the number, will override one of the buttons to show the answer and lead to the next question when clicked. Next we had to randomize all of the wrong answers. For this we only needed to worry about changing the text, not the function, of the button, since all buttons are initialized as wrong answers. The randomize the wrong answers I created a variable called “others” with a list containing four wrong answers. Then in the question procedure I set a variable called “global name” equal to a random item from the list. I then called for the first button to display global name. After that I called a procedure named “delete” which would find the value of global name and delete it from the list. This entire process is repeated for the remaining three buttons.

In conclusion, the App Project and the challenges we faced creating it allowed me to grow my knowledge and proficiency with block coding. The App Project was an enjoyable experience to create and I hope others find our work pleasing.