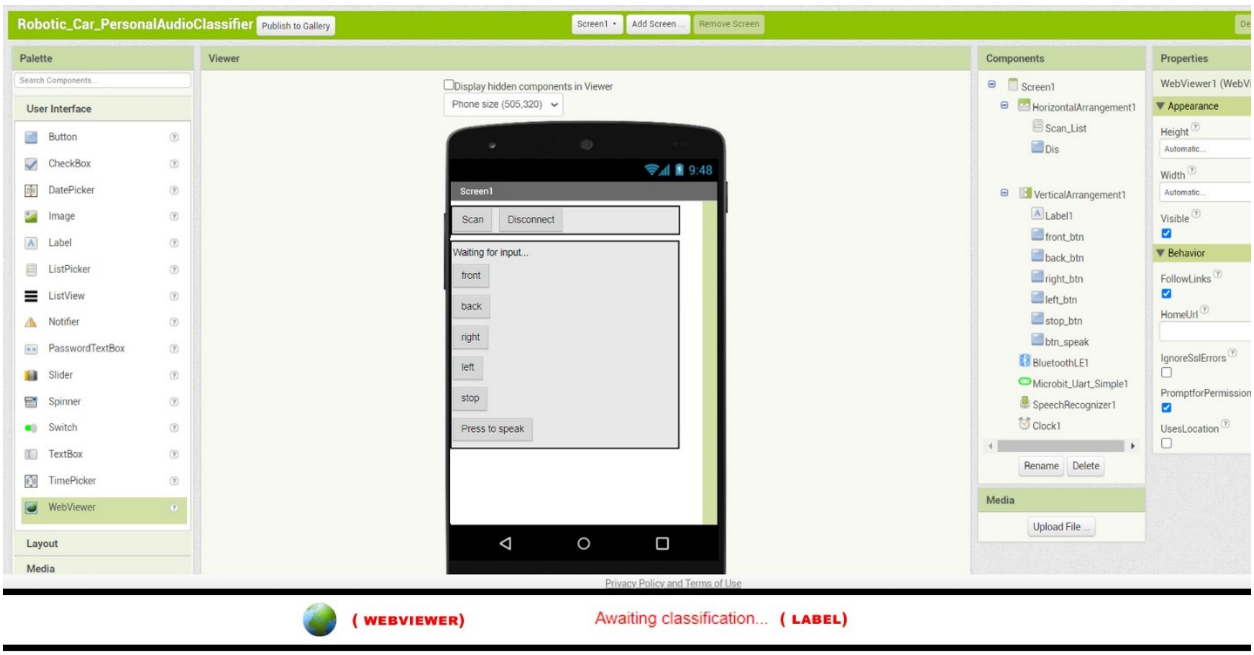
Worksheet for students Team:.....

Time for creating the application for receiving and classifying incoming voice commands through a smart device

Let's see how we can create an application that will integrate the trained model we produced by using the Personal Audio Classifier tool and installed it to a smart device that will allow us to record and classifying voice commands, and instruct the robotic car accordingly.

Step 1: Let's open the MIT App Inventor and continue working on our previous project.

The following image presents the Designer interface of MIT App Inventor. Place the Webviewer component and the Label that will inform us about the results of classification between the Horizontal and the Vertical Arrangement layout. Use the components and properties menus to modify and rename the added label.

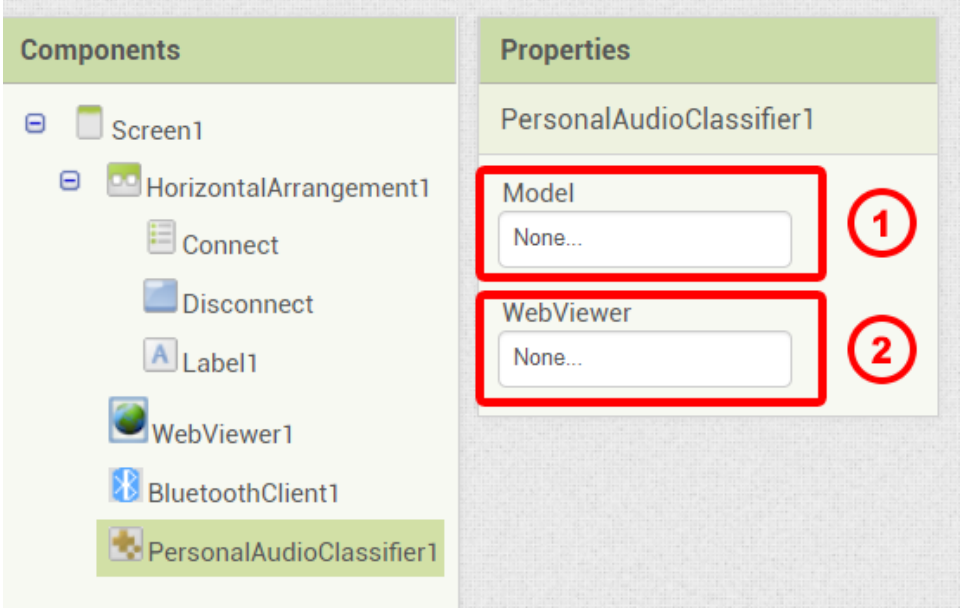


Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

Step 1: Let's add the Personal Audio Classifier extension.

Add the Personal Audio Classifier Extension, select the corresponding component on the Components menu. What modifications do you think you need to do on the fields 1 and 2, to enable the use of the trained model? *Discuss with your team, write your thoughts below and make the corresponding changes to your project.*



Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

Time for programming the application

Let's program the components that you have added to your application. To do that click on the App Inventor's "Blocks" interface. Here you can find all the commands that you need to create a functionable application and create your scripts by snapping the correct block commands in the scripting area.

3) Programming the Personal Audio Classifier component

(Before moving to the next step, make sure that you have uploaded the script created to the Makecode environment on your robotic car)

The Makecode script contains a number of messages. If these messages are received then the robotic car will act accordingly (i.e., with the message #forward# the robotic car will move forward etc.) Therefore, our aim is to enable our application to send each one of these messages in order to trigger the corresponding behavior. This will be managed through programming the Personal Audio Classifier component.

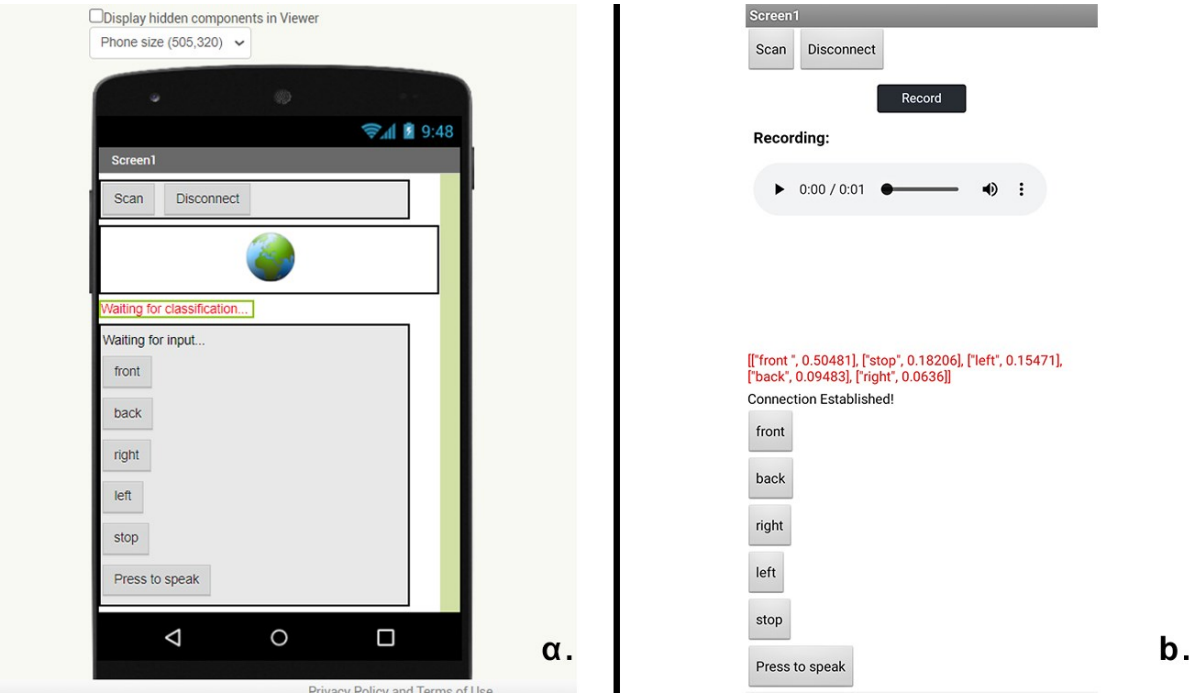
Have a look at the Mekecode script and note down to the following table which message will be transmitted to the robotic car, when a specific voice command is received:

Voice command received	*Message that should be sent*
"front"	
"back"	
"left"	
"right"	
"stop"	

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

The Personal Audio Classifier component is instantly adding a Record button to the application. This button is not visible on the design area (a.). It is only visible on the smart device (once the application has been programmed and installed to the smart device) (b.).



Therefore, we need to determine what the application should do with the results of the recording, after the incoming voice command has been classified (i.e., instruct the robotic car to perform the corresponding movement).

We will do that by assembling a script that will check if the recorder result is classified as one of the categories/labels contained to the trained model.

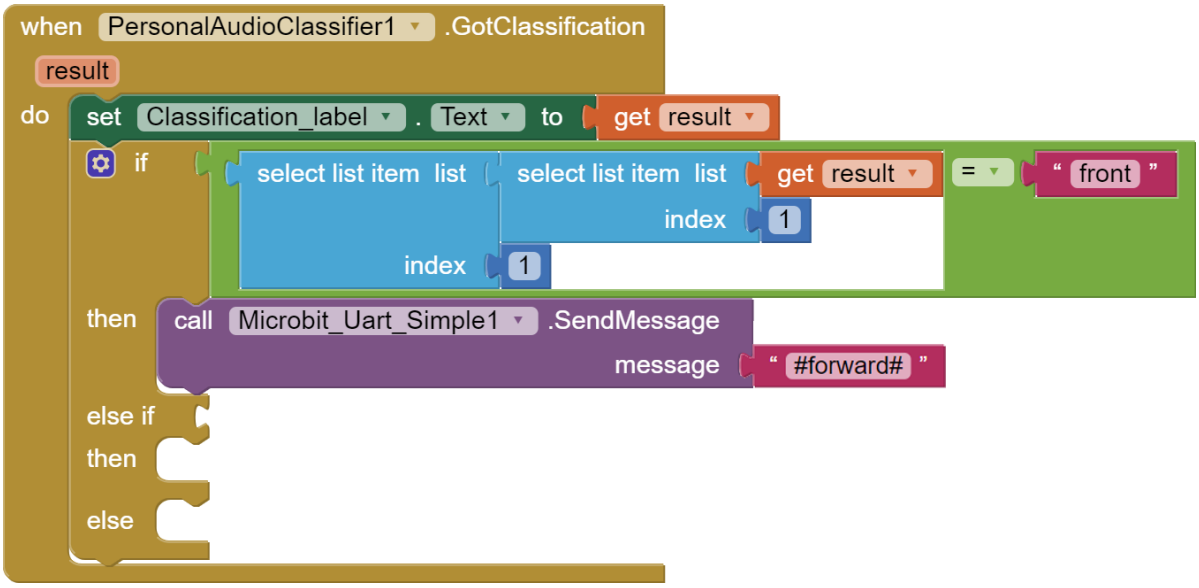
If so, then the script will send – through the Microbit_Uart_Simple.Send Message - the corresponding message to our robotic car.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

The following script is semi-structured. After getting the result of the classification, the script searching which one of the labels contained in the model is the best match.

So, if the result, pulled out from the list of labels, matches to the "front" label/category, then the application sends the message "#forward#" to the electronic device, and the robotic car moves forward.



Note: Index [1] means that the first item in this list will be pulled out. So, if the results of the classification of a sound are "*[front, 0.50], [stop, 0.18], [left, 0.15],etc.*", this means that the first item in this list is the front voice command. Therefore, our application will send the corresponding message to the robotic car (the message "#forward#" in our example above).

Try to repeat the same process for all the labels/categories that you have created (e.g., back, left etc.) keeping in mind to insert the correct message in the text field.

Which part of the script do you need to repeat on the final "else" statement? *Discuss with your team*

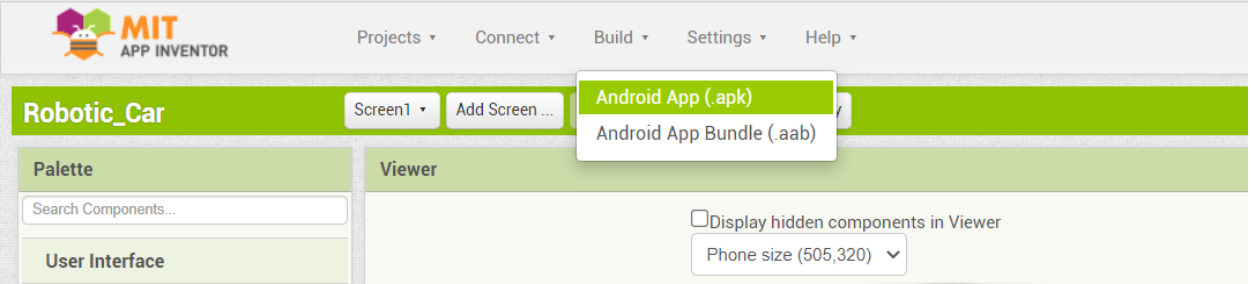
Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

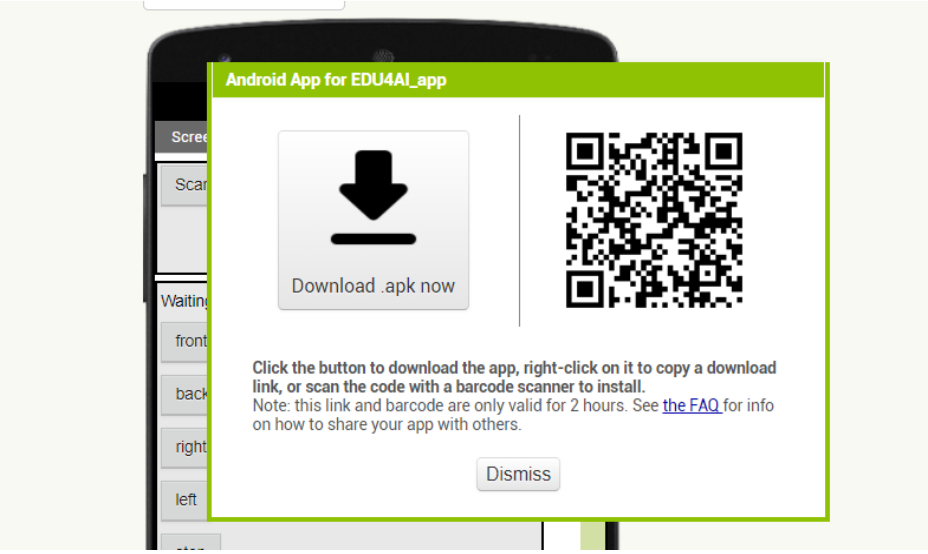
Testing the application

Let's try out our application to check if everything works properly.

After completing all the aforementioned steps, go to the Build menu and from the floating menu click on the Android App (.apk) selection.



A progress bar will appear, indicating that your application is generated. This process might take a couple of minutes. When this process ends, a window will pop up informing you that you can either download the application or you can scan the generated QR code to install the application to your smart device. Choose what is more convenient for your device and after finishing the installation process, open the application to your device and try it out.



Run the application and start recording sounds. Observe the classifications results and see if you would like to change anything (e.g., create more categories/labels, refine your trained model etc.)

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

You can also organize your observations to a table like the following, by noting down the results of the classification and the confidence level (i.e., how sure is the application that the recorded sound belongs to a certain category):

	Result and Confidence level				
Recorded command	front	back	left	right	Stop
Front command	0.50	0.09	0.15	0.06	0.18
…					
….					

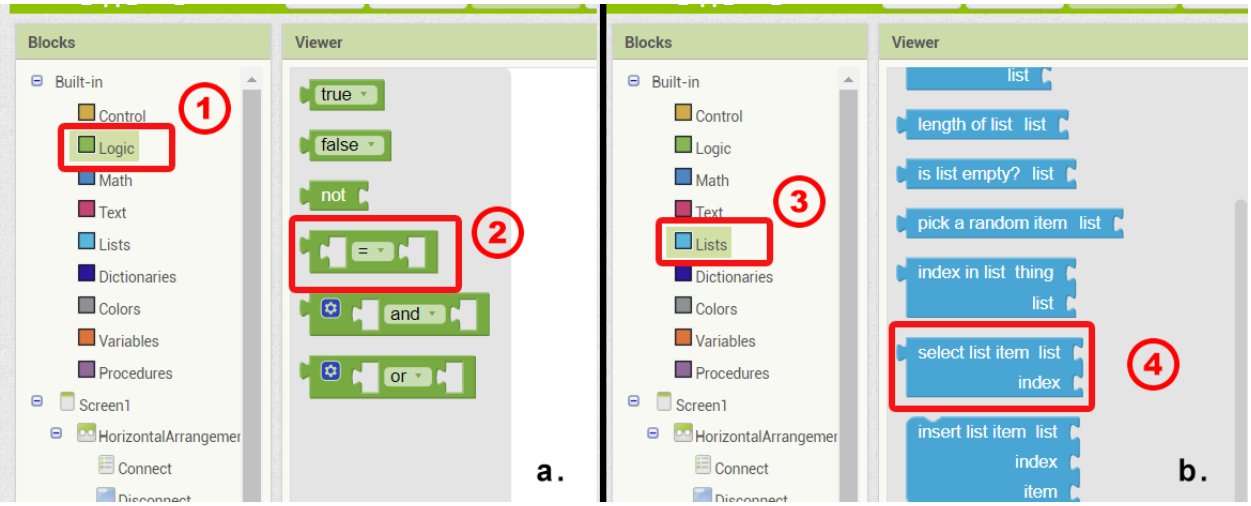
Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

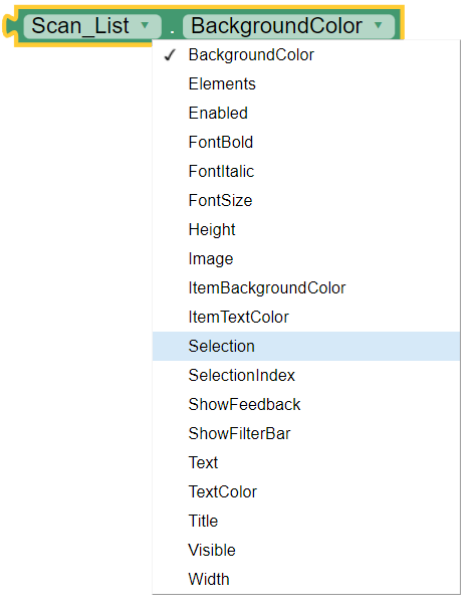
Tip Zone

Finding the commands:

To find some of the needed commands, click on the corresponding item or category (i.e., Logic or ""Lists" in the following example) and search on the popping up floating menu.

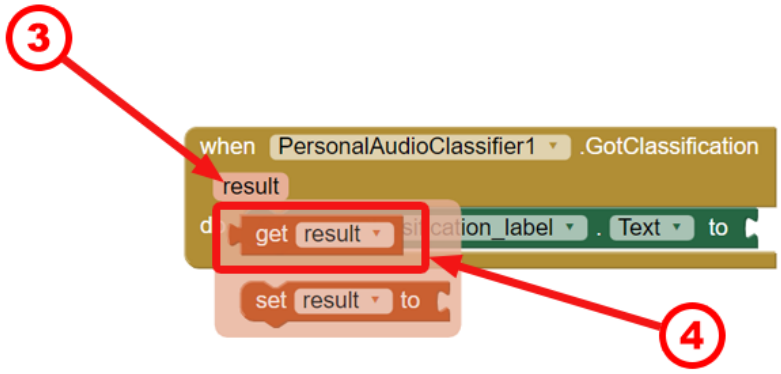


Some commands contain more than one option. The following images presents such commands.

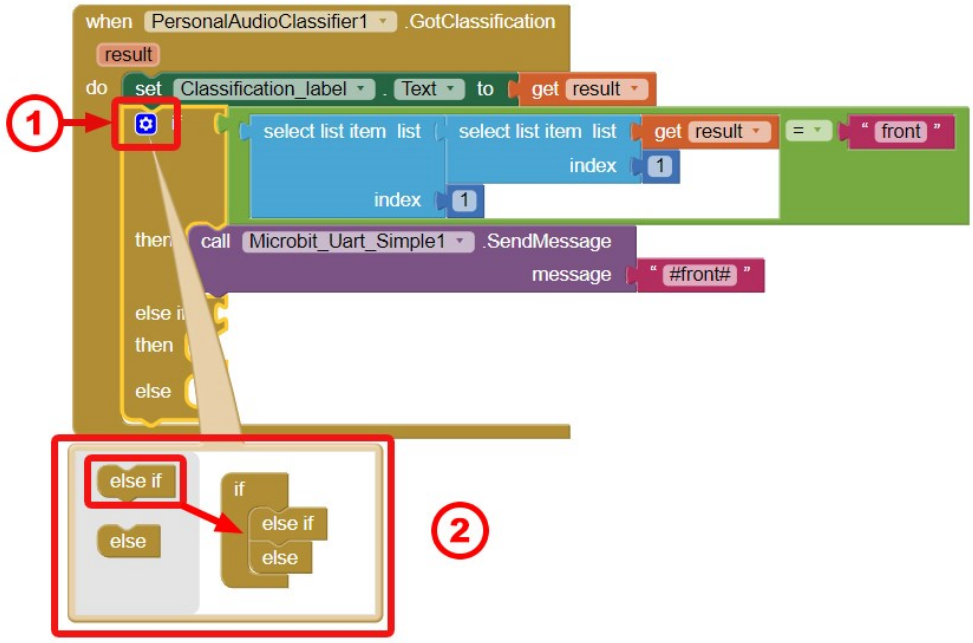


Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.



if you want to add more “else if” conditions click on the blue gear (1) next to if statement and from the floating menu that will appear (2), drag and drop as many new else if conditions you need



Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.