



Esercitazione thunkable

Giulio Galesi, ISTI-CNR

Blocco funzione in Thunkable

Il blocco funzione ci consente di raggruppare più istruzioni in un unico blocco che potrà essere richiamato più volte nella nostra app senza dover duplicare i blocchi istruzione di cui è costituito.

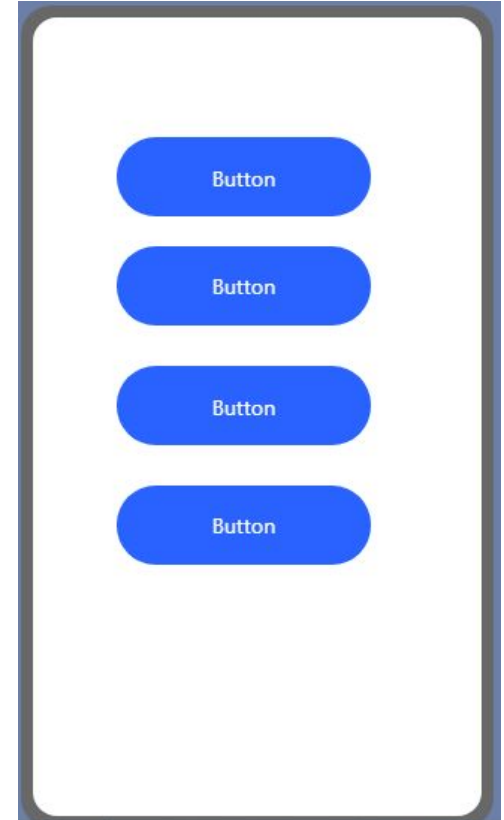
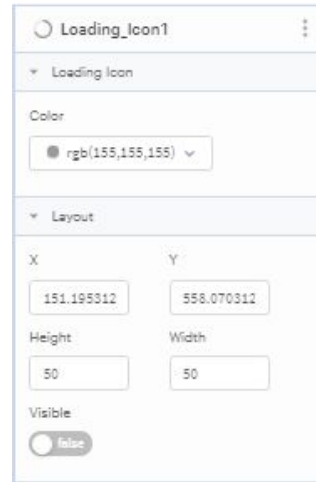
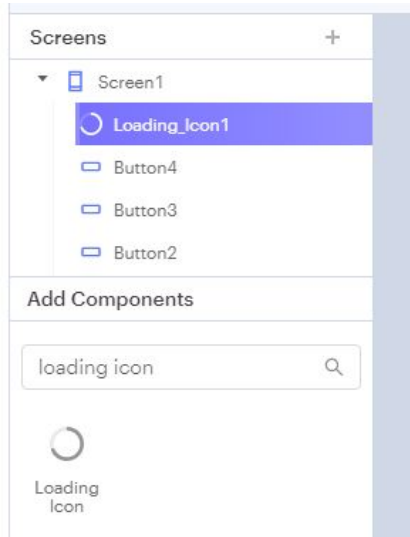
Proviamo a realizzare una semplice funzione che poi andremo a richiamare in altri blocchi.

La funzione che vogliamo realizzare, quando viene chiamata, deve eseguire 3 istruzioni:

- 1) fornire un feedback visivo
- 2) fornire un feedback tattile
- 3) fornire un feedback audio

Blocco funzione in Thunkable

Nel Design dell' interfaccia inseriamo 4 pulsanti e un componente "Loading Icon" (settiamo il suo attributo visible su false) e andiamo a lavorare sui blocchi.



Creiamo una funzione 1

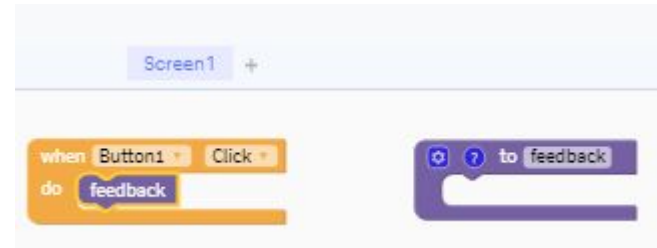
Inseriamo un blocco evento associato al click di uno dei 4 pulsanti.

Dai blocchi Core selezioniamo “Functions”, trasciniamo un blocco funzione nel desk e chiamiamolo “feedback”.



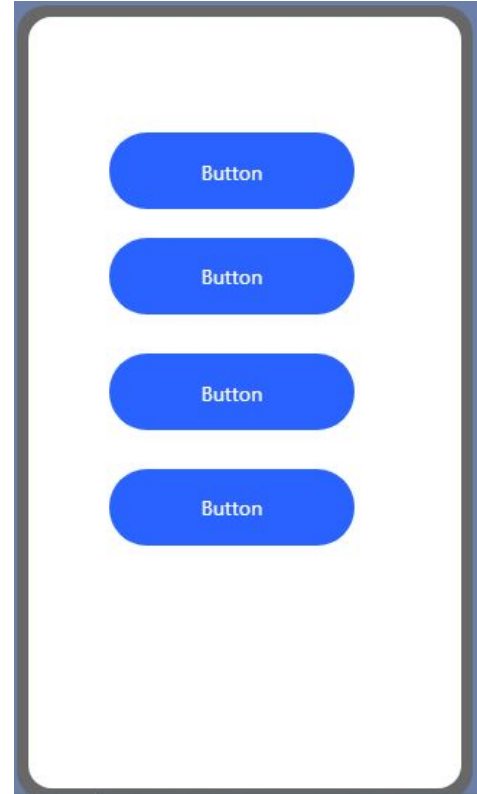
Creiamo una funzione 2

Tornando nel menu dei blocchi funzione notiamo che è stato generato un nuovo blocco funzione “feedback”. Questo blocco ci serve per richiamare le istruzioni contenute nella funzione. Selezioniamo questo blocco e lo trasciniamo all’interno del blocco evento di Button1.



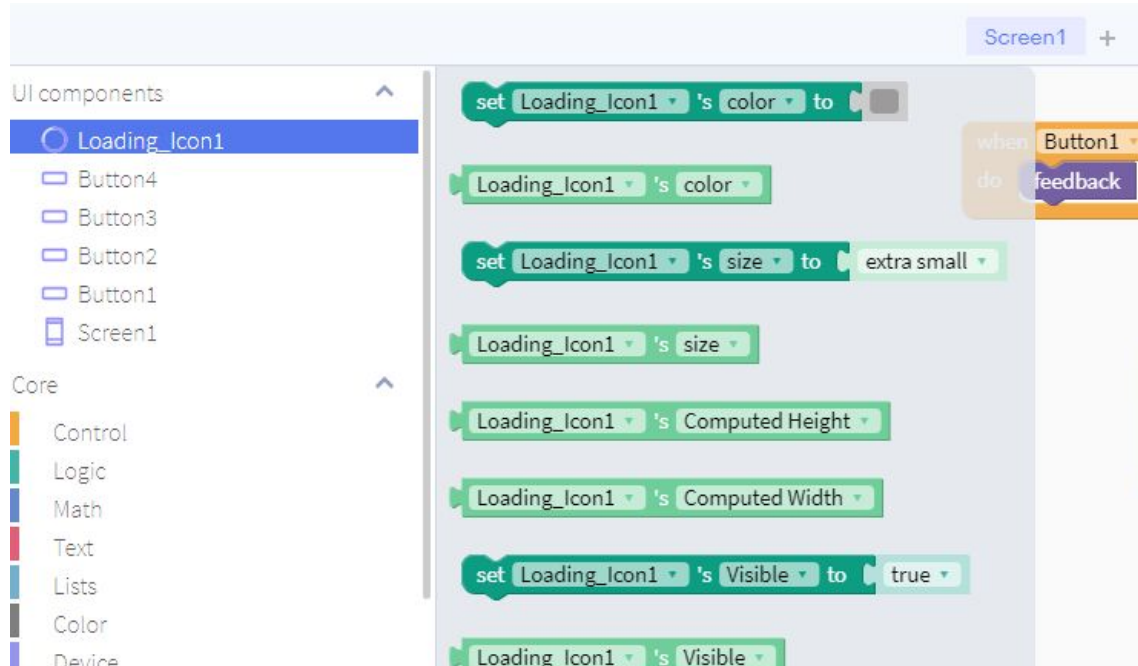
Proviamo la funzione senza istruzioni

Se proviamo la app ci accorgiamo che cliccando il pulsante non succede nulla, visto che non abbiamo inserito istruzioni nella funzione feedback.



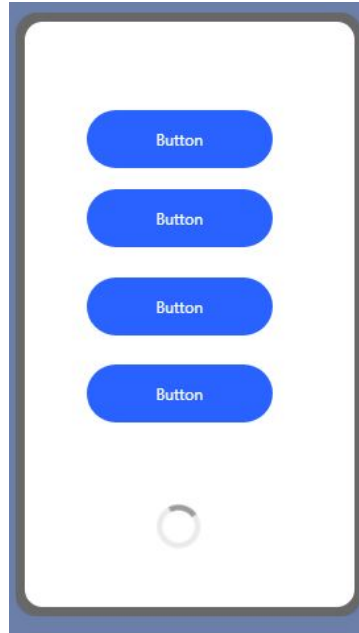
Aggiungiamo le istruzioni alla funzione - feedback visivo

Selezioniamo il blocco “set loading icon’s visible to true” e trasciniamolo all’interno del blocco funzione feedback.



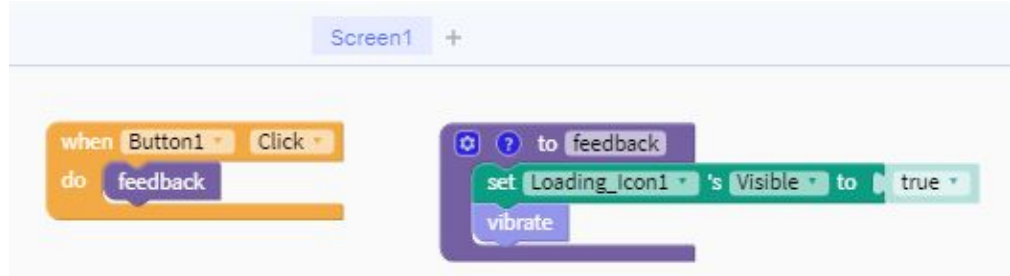
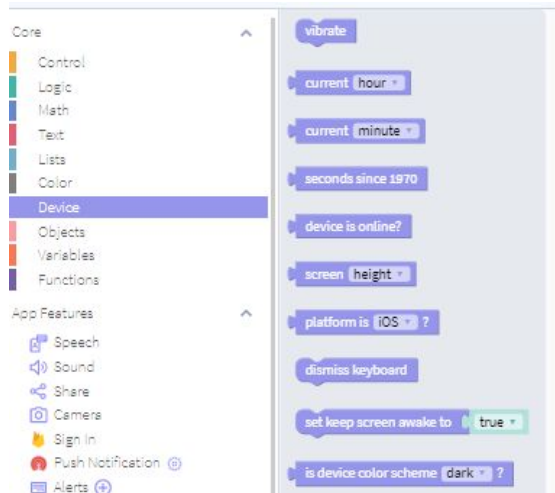
Proviamo la funzione con istruzione per feedback visivo

Se proviamo la app ci accorgiamo che cliccando il pulsante comparirà su schermo l'icona di caricamento.



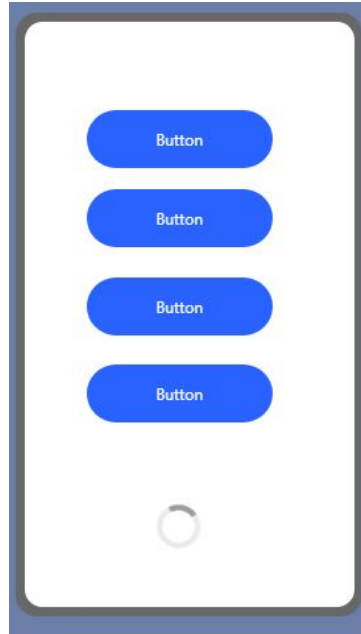
Aggiungiamo le istruzioni alla funzione - feedback tattile

Andiamo tra i blocchi del dispositivo (Device) e trasciniamo il blocco “vibrate” all’interno della funzione.



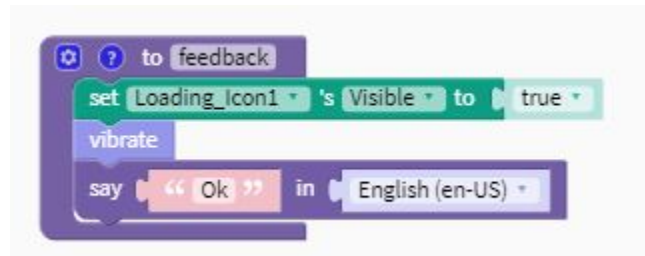
Proviamo la funzione a cui è stato aggiunto feedback tattile

Se proviamo la app adesso, cliccando il pulsante comparirà su schermo l'icona di caricamento (feedback visivo) e il dispositivo emetterà una vibrazione (feedback tattile) .



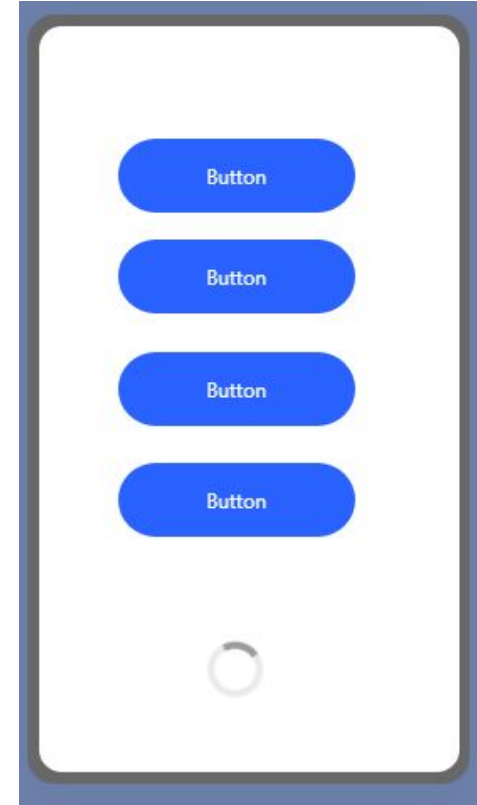
Aggiungiamo le istruzioni alla funzione - feedback audio

Andiamo tra le funzionalità di Speech e trasciniamo il blocco “say” all’interno della funzione.



Proviamo la funzione a cui è stato aggiunto feedback audio

Adesso la funzione è completa, cliccando il pulsante comparirà su schermo l'icona di caricamento (feedback visivo), il dispositivo emetterà una vibrazione (feedback tattile) e verrà riprodotto il messaggio vocale "Ok" (feedback audio).



Per concludere la funzione..

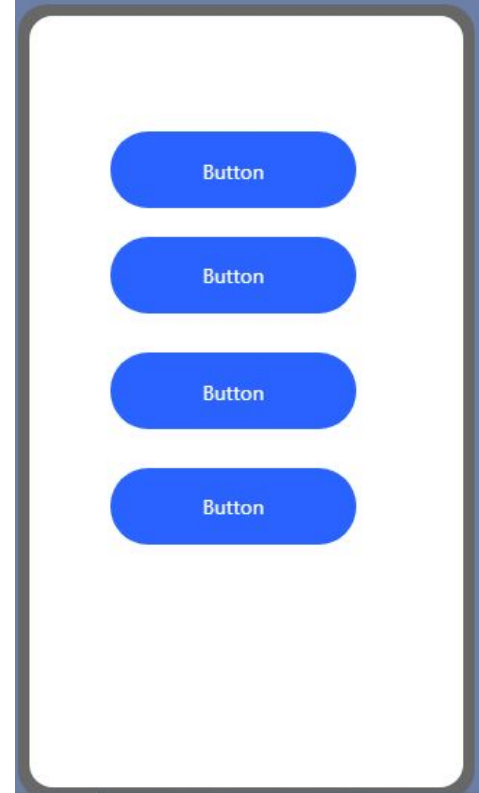
Inseriamo una pausa di 1 secondo nella funzione, duplichiamo il blocco dell'icona di caricamento e impostiamo l'attributo di visibilità su false in modo che l'icona di caricamento scompaia dopo che sono state eseguite le varie istruzioni.

The image displays the Scratch code editor with three panels illustrating the steps to complete a function:

- Top Left:** A sidebar showing the 'Control' category selected under the 'Core' heading.
- Top Middle:** A 'do' block containing an 'else' block with a 'wait 1 seconds' block.
- Top Right:** A 'to feedback' function block containing:
 - 'set Loading_Icon1's Visible to true'
 - 'vibrate'
 - 'say "Ok" in English (en-US)'
 - 'wait 1 seconds'A context menu is open over the 'wait 1 seconds' block, showing options: Duplicate, Add Comment, Collapse Block, Delete 2 Blocks, Help, and Undo last delete.
- Bottom:** The final 'to feedback' function block, which now includes an additional 'set Loading_Icon1's Visible to false' block at the end.

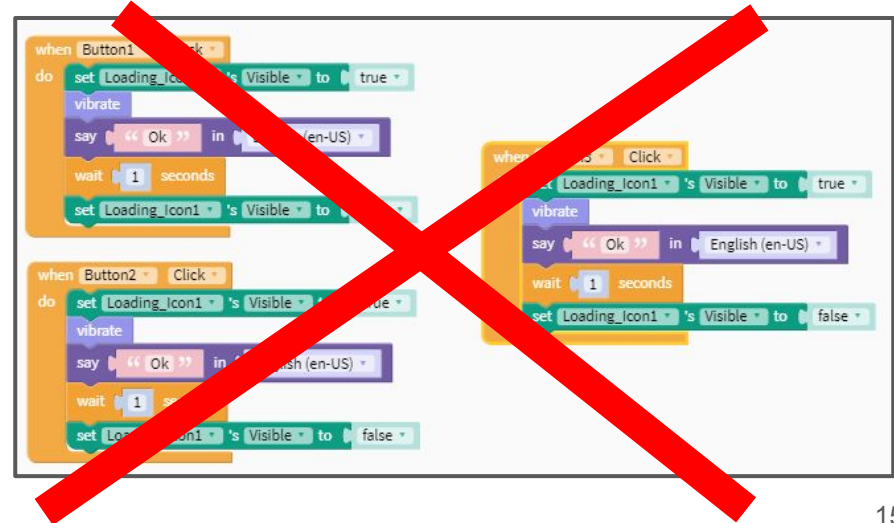
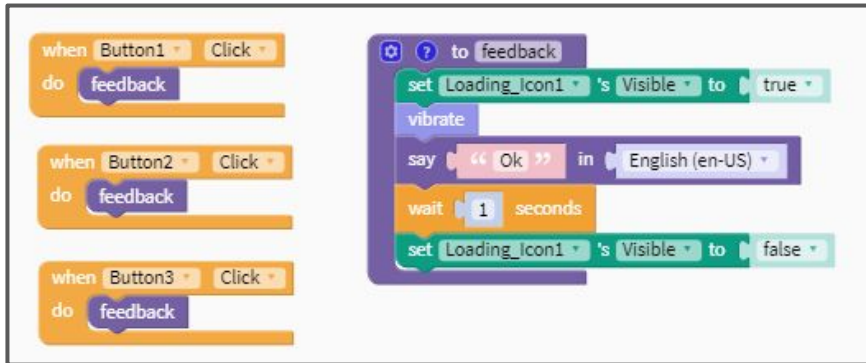
Proviamo la funzione a cui è stata aggiunta una pausa prima di disattivare la loading icon

Adesso la loading icon scompare una volta concluse tutte le istruzioni tattili e audio.



Esempi pratici che mostrano perchè conviene usare le funzioni 1

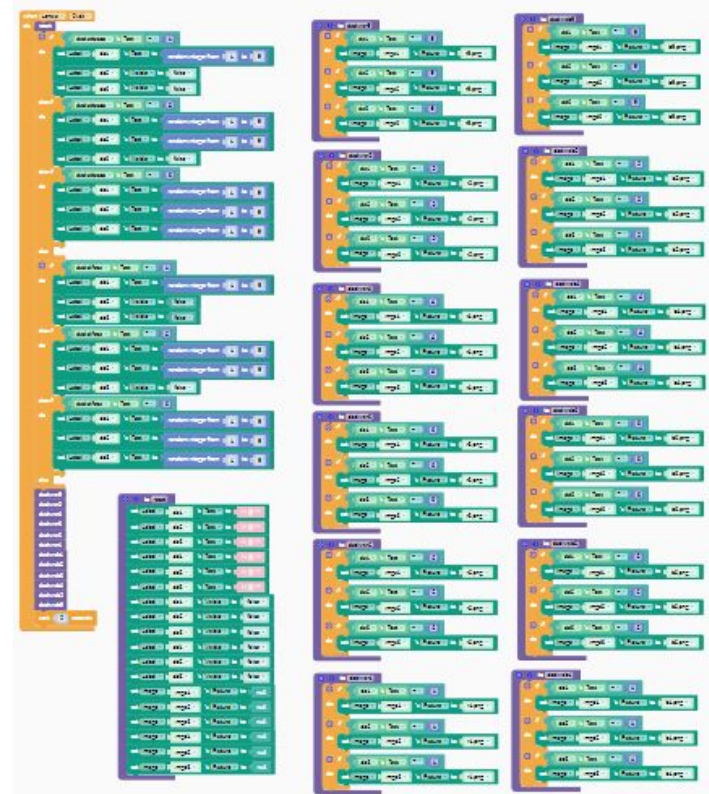
Se volessimo applicare il medesimo funzionamento anche ad altri pulsanti, basterebbe richiamare la funzione e avremmo una maggiore compattezza visiva ma soprattutto..



Esempi pratici che mostrano perchè dobbiamo usare le funzioni 2

..se dovessimo apportare delle modifiche alla nostra app basterebbe lavorare sulla singola funzione e le modifiche si ripercuoterebbero su tutti i blocchi in cui la abbiamo richiamata.

Immaginate invece se dovessimo andare a cercare tutti i punti in cui abbiamo utilizzato quelle stesse istruzioni che ora dobbiamo modificare, oltre a perdere molto tempo si rischierebbe di non arrivarne a capo e l'errore sarebbe a portata di mano!



E se la funzione si fosse dovuta applicare a tutti i pulsanti presenti nella UI anzichè solo ad alcuni?

In questo caso il modo più rapido e compatto sarebbe stato quello di usare il blocco “Any Button” che esegue le istruzioni su tutti i pulsanti.

The image displays three screenshots of an Android Studio interface, illustrating the use of the 'Any Button' block in a Kotlin script.

Top Left Screenshot: Shows the 'Advanced' tab of the 'Any Component' block. The 'Web APIs' and 'Bluetooth' options are visible. The 'Any Component' block is selected, and the 'Any Button' block is shown in the 'do' section of the script.

Bottom Left Screenshot: Shows a script snippet for the 'Any Button' block. The 'do' section contains a 'feedback' block. A 'component' label is visible next to the 'Any Button' block.

Right Screenshot: Shows a script snippet for the 'Any Button' block. The 'do' section contains a sequence of blocks: 'set Loading_Icon1's Visible to true', 'vibrate', 'say "Ok" in English (en-US)', 'wait 1 seconds', and 'set Loading_Icon1's Visible to false'. A 'component' label is visible next to the 'Any Button' block.

Sviluppiamo alcune app insieme passo per passo

- 1) App per insegnare ai bambini una lingua straniera
- 2) App per anziani con difficoltà di comunicazione
- 3) App dado parlante
- 4) app che scatti una foto, ne riconosca il contenuto e scriva in automatico la didascalia

App per insegnare ai bambini una lingua straniera

Creiamo una app che:

1. mostri una immagine
2. Mostri la relativa didascalia
3. Consenta tramite un pulsante di riprodurre il testo della didascalia
4. Consenta tramite un pulsante di tradurre il testo in altra lingua
5. Consenta tramite un pulsante di cambiare immagine



App per insegnare ai bambini una lingua straniera - UI

Inseriamo nel design:

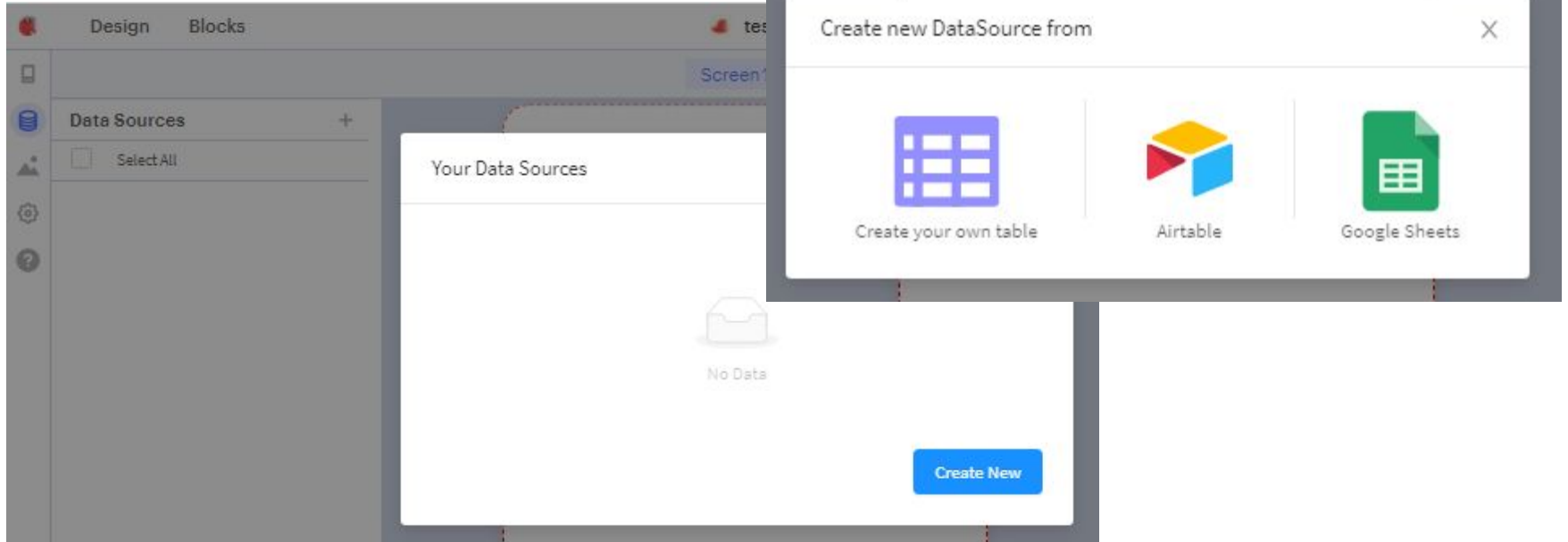
1. Un componente immagine
2. Due etichette
3. Quattro pulsanti

L'immagine mostra l'oggetto, la prima etichetta mostra il nome in italiano dell'oggetto mentre la seconda mostra il nome in altra lingua. Due pulsanti servono a riprodurre il contenuto delle etichette mentre un terzo pulsante serve per mostrare la traduzione in lingua straniera. Il quarto pulsante (che mostra l'icona standard per refresh/aggiornamento) serve per aggiornare l'immagine.



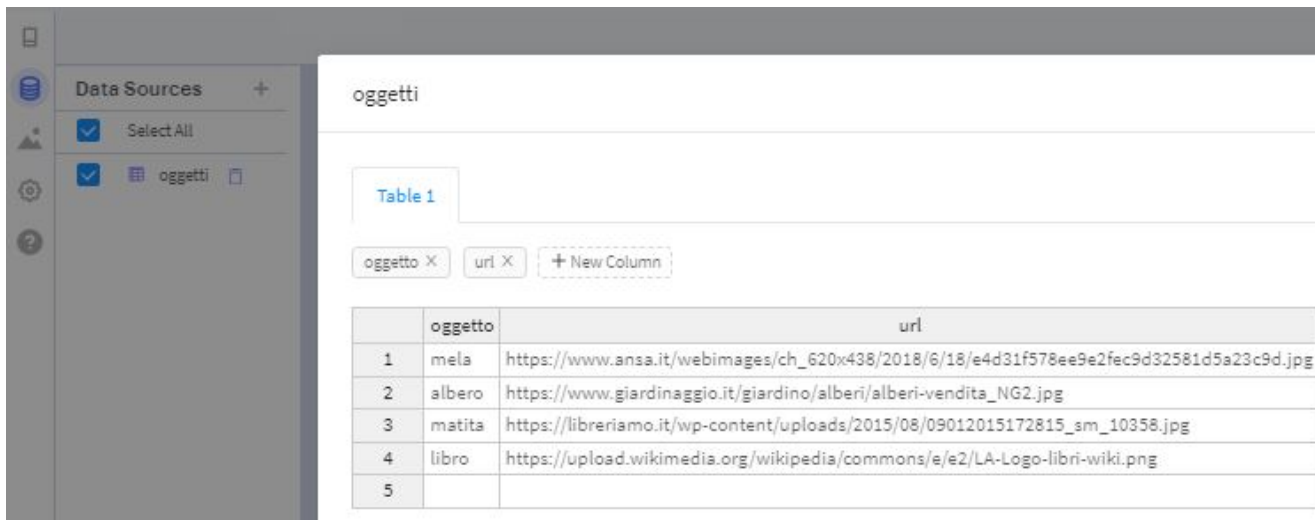
App per insegnare ai bambini una lingua straniera - DB

Creiamo una tabella da usare come sorgente di dati cliccando su “Create your own table”.



App per insegnare ai bambini una lingua straniera - creazione di una tabella dati

Costruiamo ora una nostra base di dati (chiamiamola ad esempio “oggetti”) composta da sole due colonne (oggetto e url) e andiamola a popolare di contenuti. Possiamo usare il servizio gratuito ImgBB (<https://it.imgbb.com/>) per caricare una immagine e ottenerne l'url.

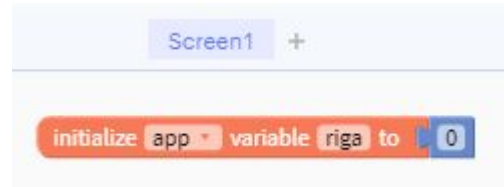


The screenshot shows a software interface for managing data sources. On the left, a sidebar titled 'Data Sources' contains a 'Select All' checkbox and a checkbox for 'oggetti'. The main area is titled 'oggetti' and displays a table with two columns: 'oggetto' and 'url'. The table contains five rows of data. Above the table, there are buttons for 'oggetto X', 'url X', and '+ New Column'. The table is labeled 'Table 1'.

	oggetto	url
1	mela	https://www.ansa.it/webimages/ch_620x438/2018/6/18/e4d31f578ee9e2fec9d32581d5a23c9d.jpg
2	albero	https://www.giardinaggio.it/giardino/alberi/alberi-vendita_NG2.jpg
3	matita	https://libreriamo.it/wp-content/uploads/2015/08/09012015172815_sm_10358.jpg
4	libro	https://upload.wikimedia.org/wikipedia/commons/e/e2/LA-Logo-libri-wiki.png
5		

App per insegnare ai bambini una lingua straniera - inseriamo una variabile

Andiamo a comporre i blocchi. Per prima cosa inizializziamo a zero una variabile che chiamiamo “riga” che ci servirà per tenere traccia del numero di riga della nostra base di dati a cui la nostra app dovrà attingere in maniera casuale (in modo da mostrare sempre una immagine differente).



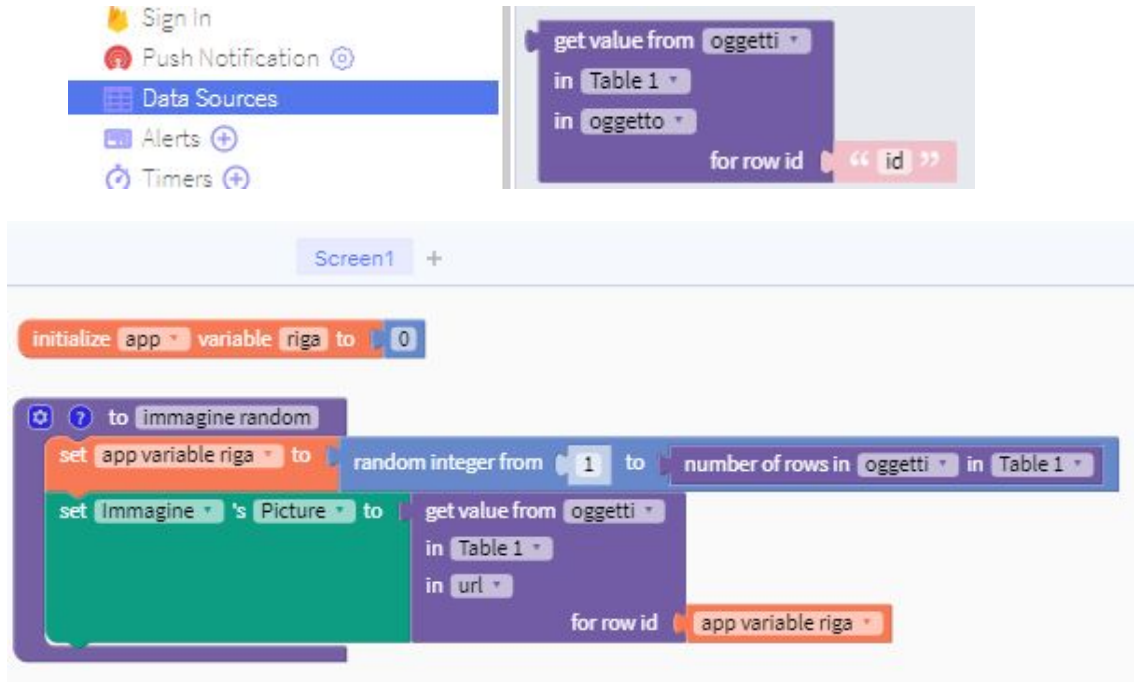
App per insegnare ai bambini una lingua straniera - random

Creiamo una funzione che vada a impostare casualmente un numero compreso da 1 al numero delle righe della nostra tabella. Possiamo chiamare la funzione “immagine random”.



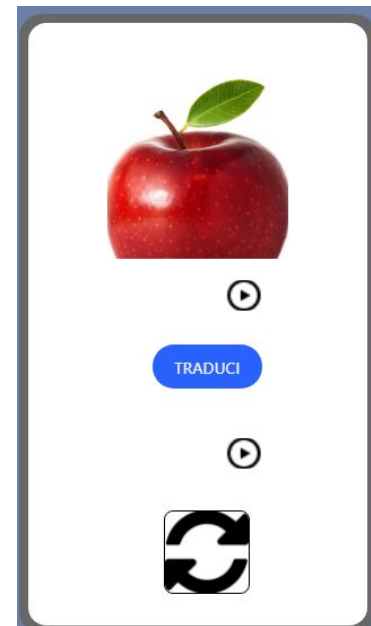
App per insegnare ai bambini una lingua straniera - recuperare valore da DB per impostare immagine

Inseriamo un blocco che usi l'url associato alla variabile riga in modo da mostrare una immagine.



App per insegnare ai bambini una lingua straniera - aggiungiamo un blocco evento

Inseriamo ora un blocco evento che esegua la funzione immagine random quando la schermata si apre. Se ora proviamo la app vedremo comparire una immagine.



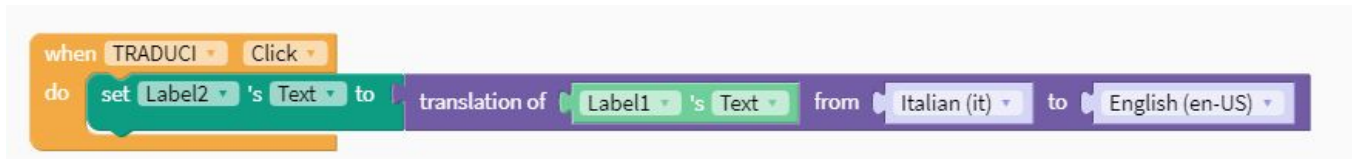
App per insegnare ai bambini una lingua straniera - recuperare valore da DB per impostare testo

Impostiamo anche il testo dell'etichetta alla stessa maniera, facendo attenzione a non riferirci alla colonna url ma alla colonna oggetto. Se ora proviamo la app vedremo comparire immagine e testo.



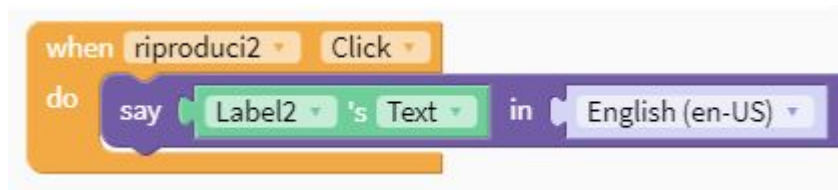
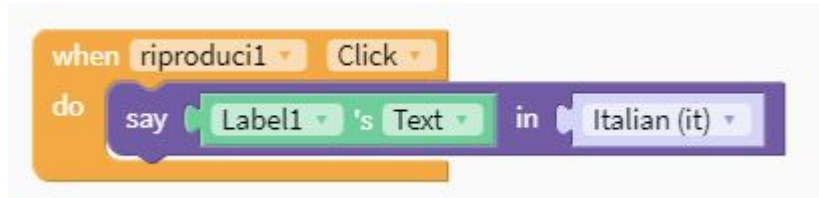
App per insegnare ai bambini una lingua straniera - pulsante traduci

Impostiamo il pulsante “traduci” perchè effettui la traduzione del testo della prima etichetta da italiano a inglese.



App per insegnare ai bambini una lingua straniera - pulsanti di riproduzione audio

Impostiamo i pulsanti “play” perchè riproducano in TTS il testo presente nelle etichette.

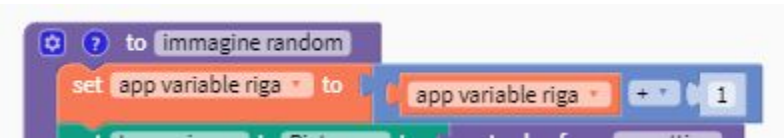


App per insegnare ai bambini una lingua straniera

Impostiamo il pulsante “aggiorna” perchè esegua la funzione “immagine random” e la app è pronta.

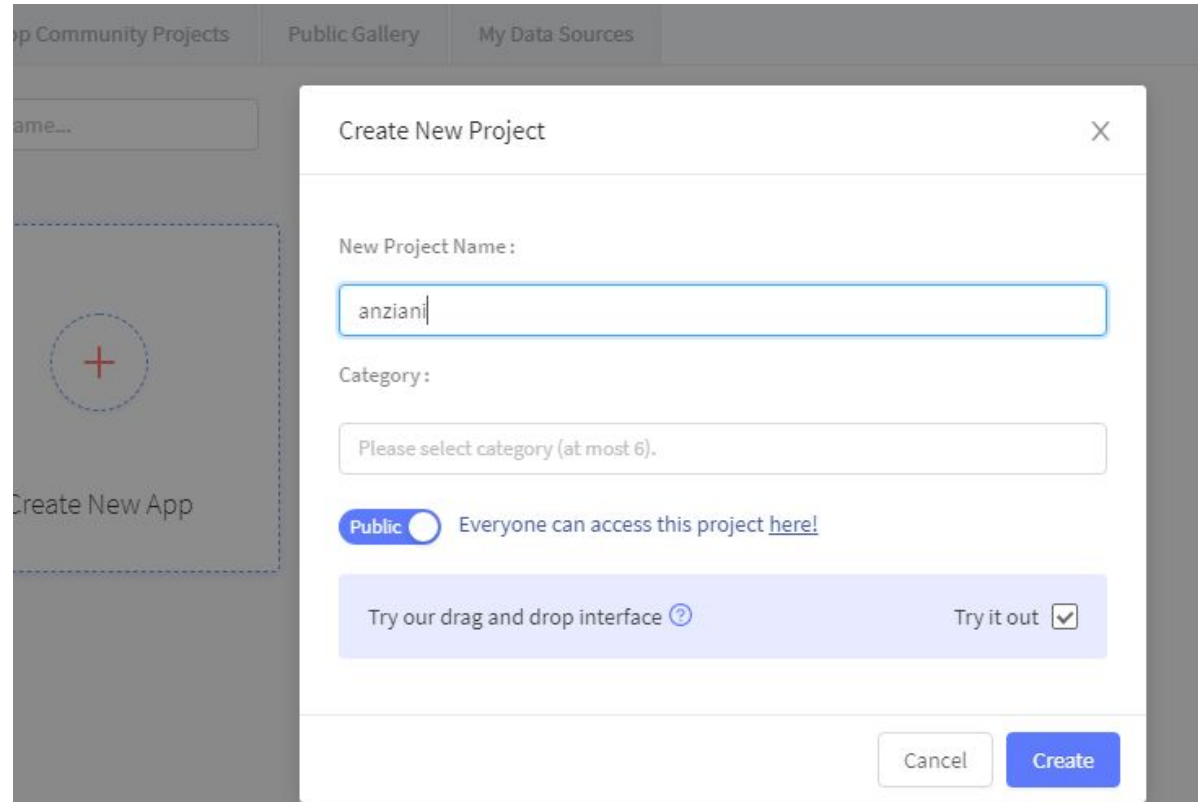


Se invece di usare un sistema casuale di scelta delle immagini volete usare un metodo sequenziale, basterà andare a modificare la funzione perchè invece di scegliere un numero a caso, si limiti ad incrementare di 1 ogni volta il valore della nostra variabile riga.



App per anziani

Creiamo un nuovo progetto

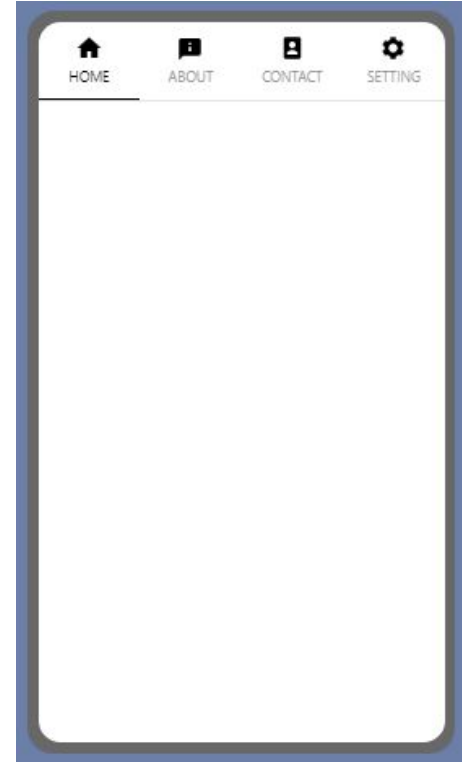
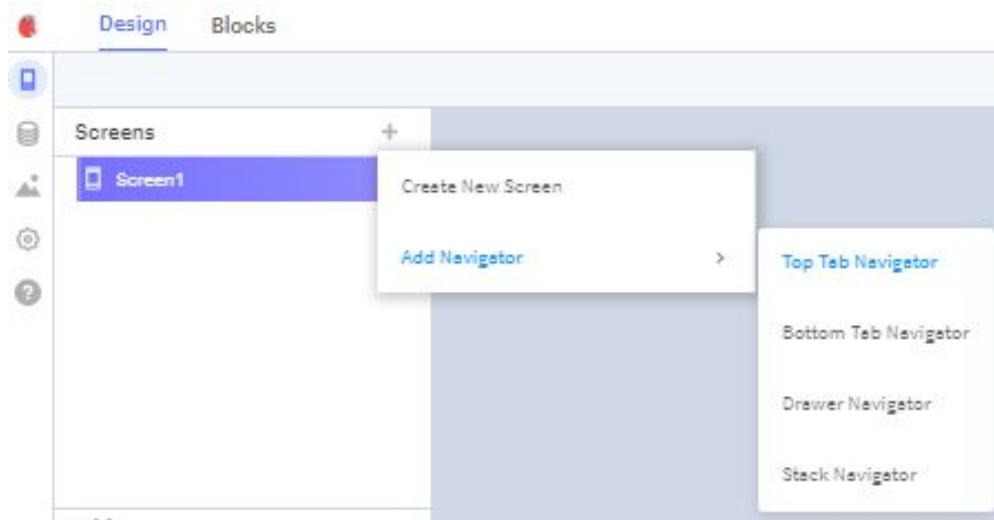


The image shows a 'Create New Project' dialog box overlaid on a software interface. The background interface has tabs for 'Community Projects', 'Public Gallery', and 'My Data Sources'. A 'Create New App' button with a red plus icon is visible. The dialog box contains the following elements:

- Title:** Create New Project (with a close button 'X' in the top right corner).
- New Project Name:** A text input field containing the text 'anziani'.
- Category:** A dropdown menu with the placeholder text 'Please select category (at most 6)'.
- Public:** A toggle switch set to 'Public', followed by the text 'Everyone can access this project [here!](#)'.
- Try our drag and drop interface:** A light blue box containing a question mark icon and the text 'Try it out' with a checked checkbox.
- Buttons:** 'Cancel' and 'Create' buttons at the bottom right.

App per anziani

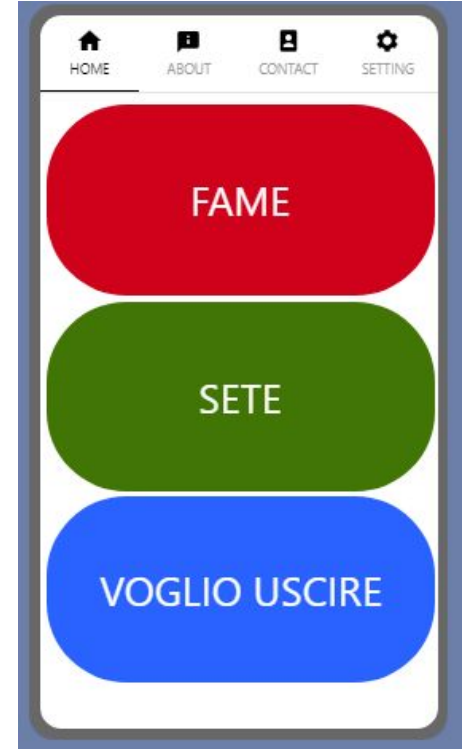
Per prima cosa aggiungiamo un Top Tab navigator cliccando sul “+” ed eliminiamo Screen1. Vengono create in automatico quattro schermate con un menu in alto con 4 icone di navigazione.



App per anziani

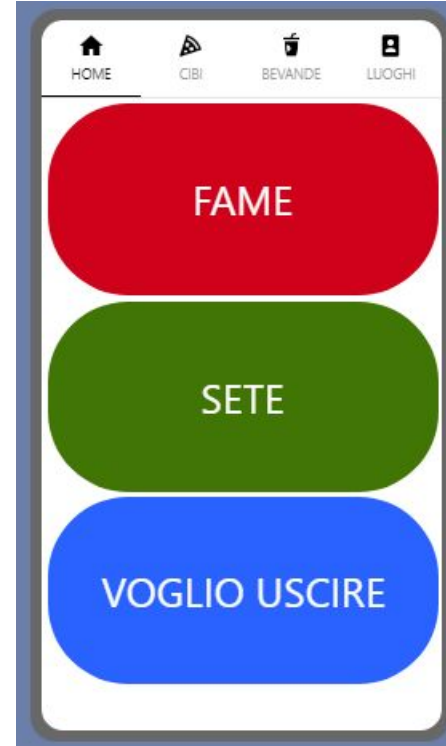
Aggiungiamo in Home tre pulsanti (inseriamo prima un button e lo duplichiamo due volte).

Rinominiamo i pulsanti e diamogli dei colori di sfondo differenti tenendo in considerazione il contrasto tra testo e sfondo.



App per anziani

Rinominiamo anche i pulsanti di navigazione e cambiamo le icone del menu cercando quelle più adatte su <https://icons8.com/>



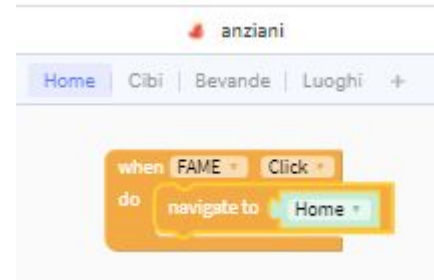
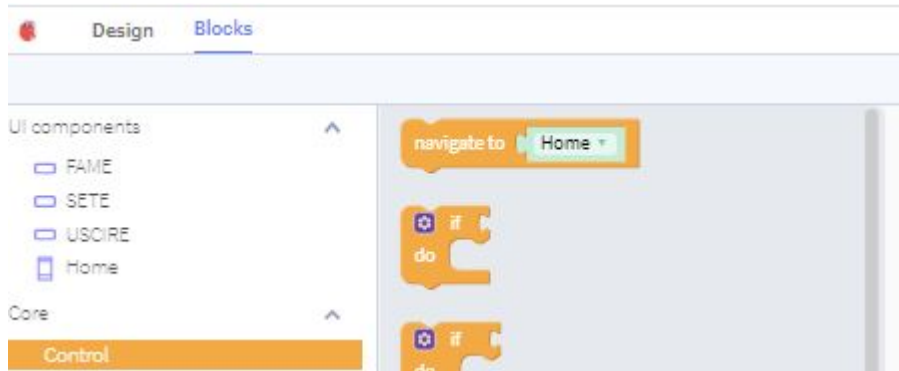
App per anziani

Passiamo a comporre i blocchi della nostra app. Selezioniamo il primo pulsante, scegliamo il blocco evento When-click e trasciniamolo nel desk.



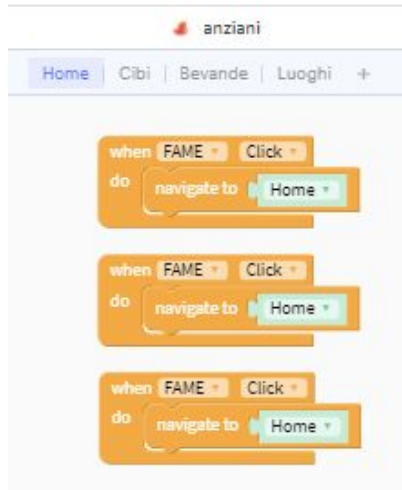
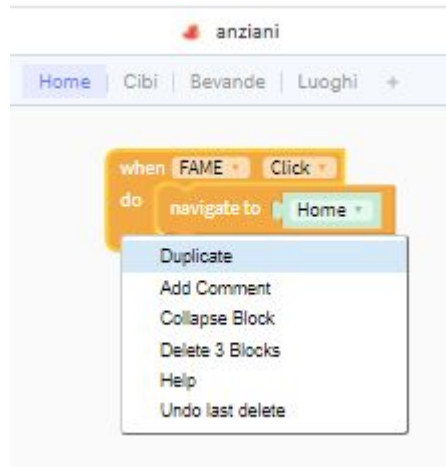
App per anziani

Scegliamo il blocco “navigate to” dalla sezione dei Core blocks - Control e trasciniamolo all’interno del nostro blocco evento.



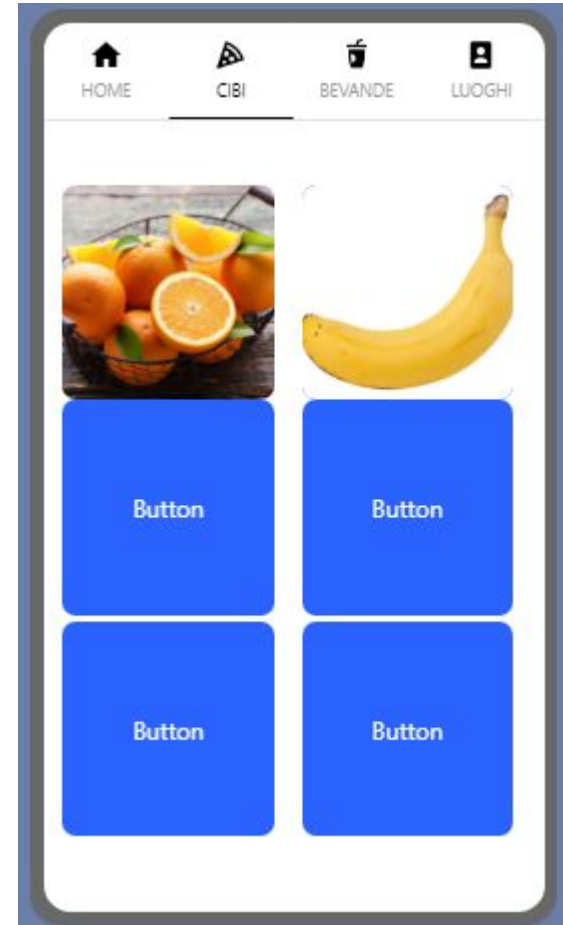
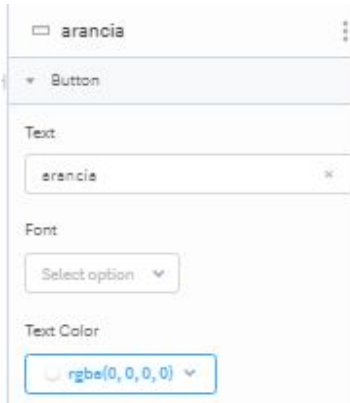
App per anziani

Duplichiamo il blocco per due volte cliccando col tasto destro del mouse sul blocco e scegliendo quindi Duplicate. Andiamo ora a modificare i riferimenti dei blocchi associandoli correttamente.



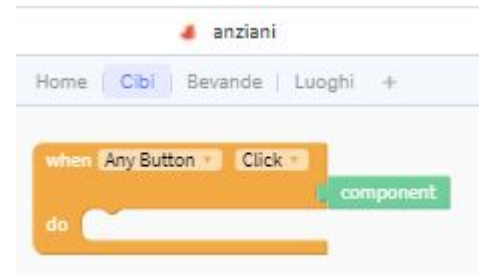
App per anziani

Torniamo sul design e nella schermata cibi andiamo ad inserire dei pulsanti a cui associamo i nomi di alimenti e sostituiamo lo sfondo con delle immagini associate al nome (ricordiamoci di impostare il colore del testo trasparente).



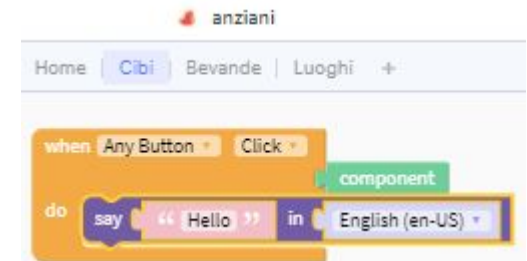
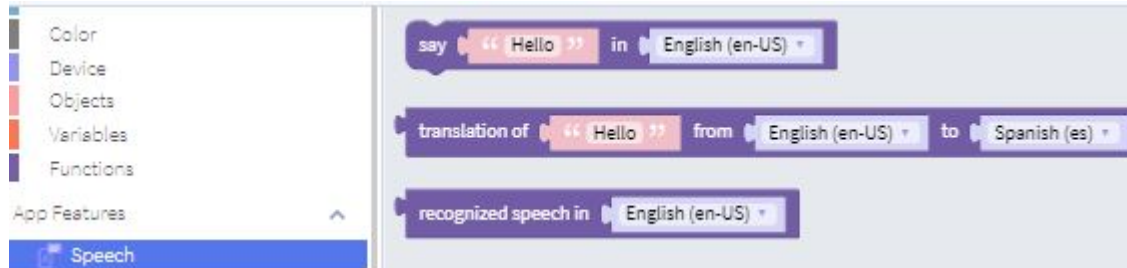
App per anziani

Spostiamoci sul desk della schermata Cibi, selezioniamo la categoria avanzata di blocchi “Any Component” e trasciniamo nel desk il blocco “when Any Button Click”.



App per anziani

Dalle funzionalità andiamo su “Speech” e trasciniamo un blocco “say” all’interno del blocco evento “Any Button”.

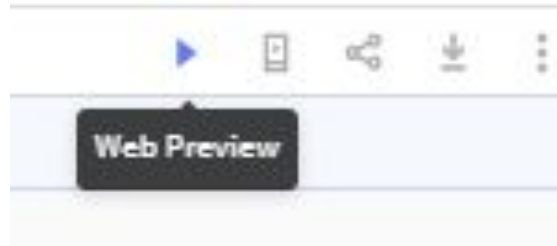


App per anziani

Proviamo ad avviare ora la nostra app cliccando sull'icona play posizionata in alto a destra.

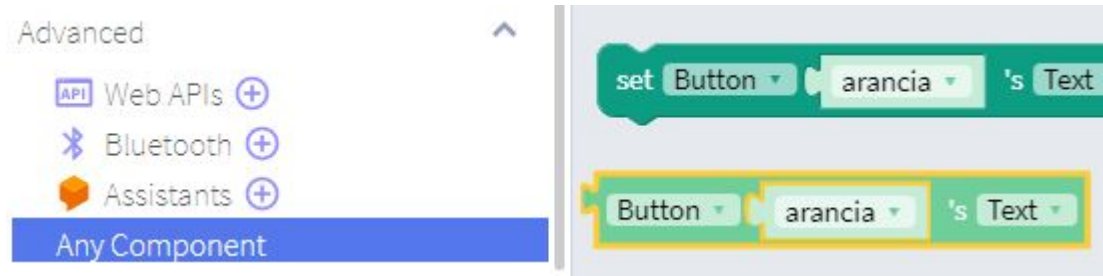
Notiamo che qualsiasi pulsante noi premiamo, viene riprodotto il saluto Hello!

Dobbiamo ora modificare i blocchi per permettere ai pulsanti, una volta premuti, di riprodurre il messaggio associato al contenuto che veicolano.



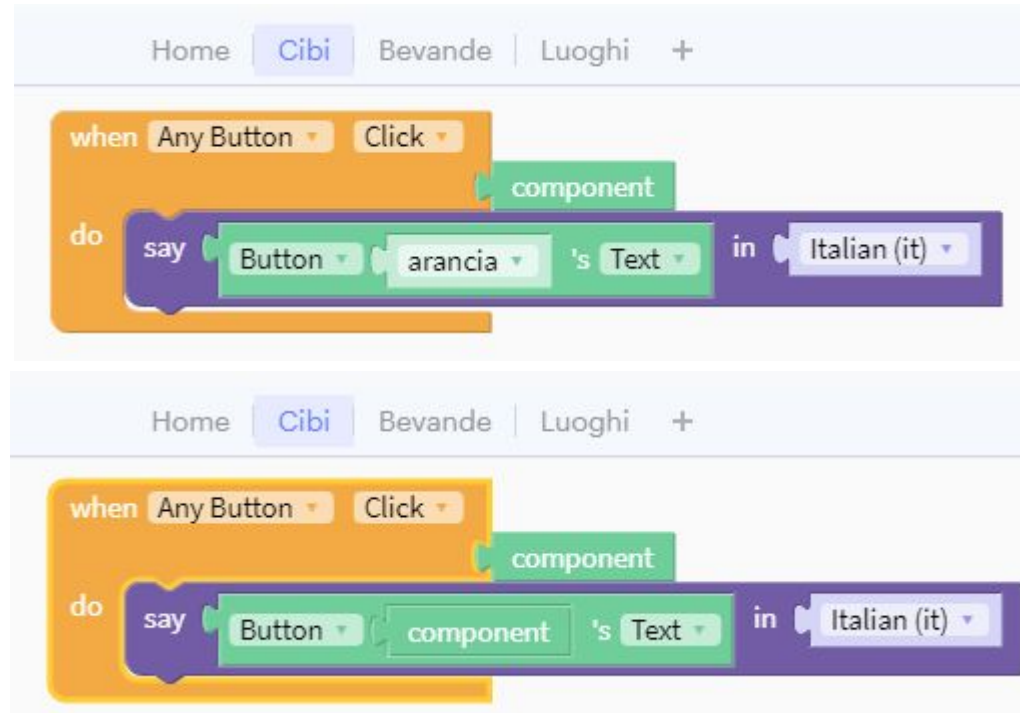
App per anziani

Da Any component andiamo a scegliere “Button xxx’s text” e trasciniamo questo blocco all’interno del blocco “say”.



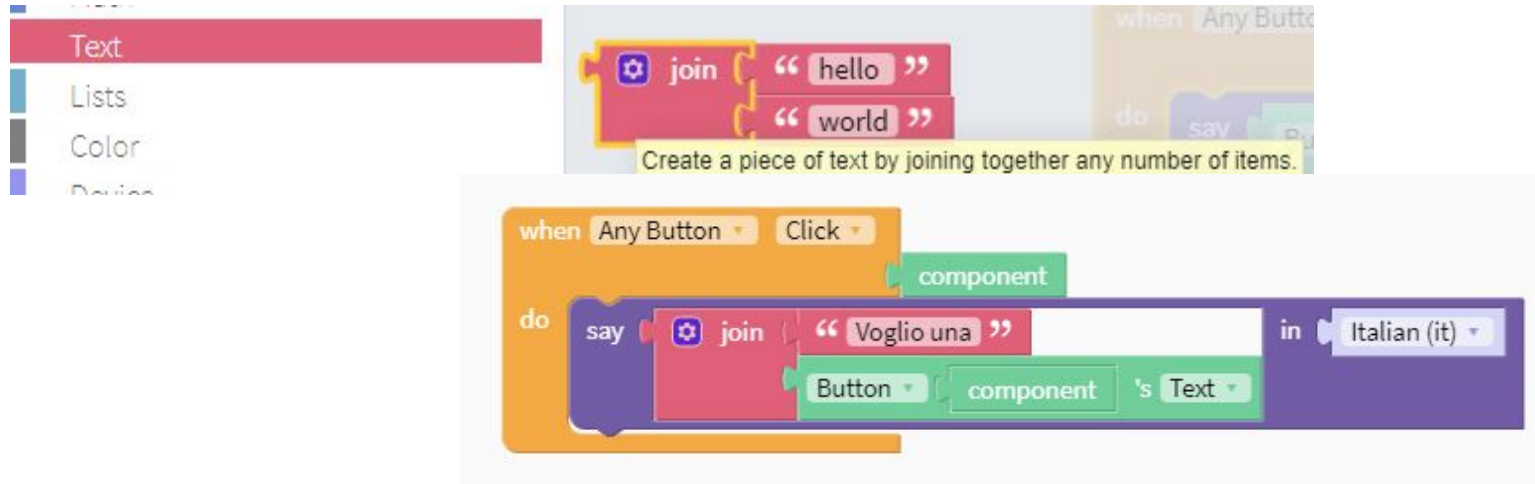
App per anziani

Selezioniamo il blocco “component” e trasciniamolo al posto del blocco “arancia”.



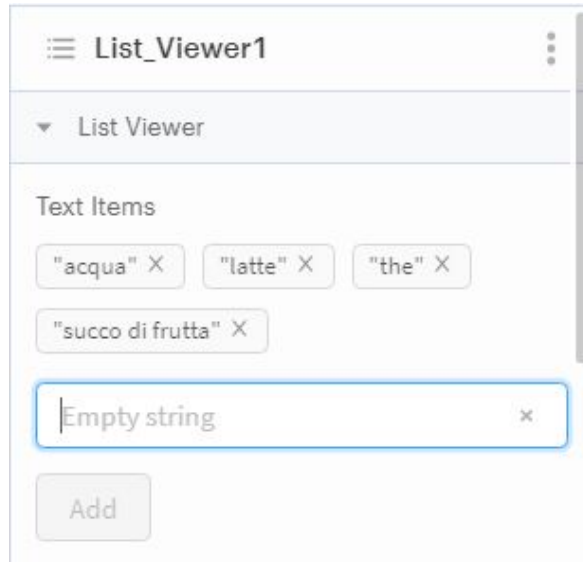
App per anziani

Se eseguiamo la app, adesso ciascun pulsante riproduce il testo associato. Utilizzando i blocchi con funzioni testuali possiamo migliorare la frase che viene letta; ad esempio, usando il blocco join, il sistema pronuncerà “voglio una arancia” anzichè dire solo “arancia”. Il testo statico impostato nella join (unione) sarà “voglio una”, mentre la parola seguente varierà in base a quale pulsante viene premuto.



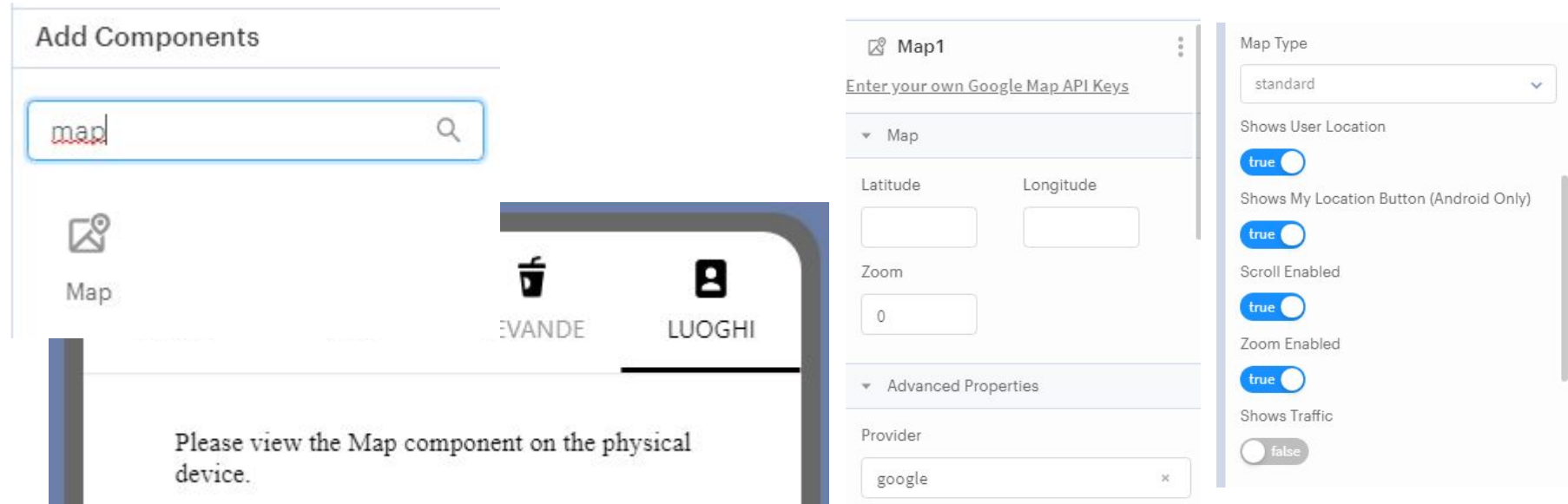
App per anziani

Possiamo fare altrettanto con la pagina delle bevande: anzichè usare i pulsanti con le immagini inseriamo un semplice elenco di elementi tra cui scegliere.



App per anziani - location

Possiamo realizzare un localizzatore gps che mostra sulla mappa dove ci troviamo. Inseriamo nella UI un componente “Map”



Add Components

map

Map

EVANDE

LUOGHI

Please view the Map component on the physical device.

Map1

Enter your own Google Map API Keys

Map

Latitude

Longitude

Zoom

0

Advanced Properties

Provider

google

Map Type

standard

Shows User Location

true

Shows My Location Button (Android Only)

true

Scroll Enabled

true

Zoom Enabled

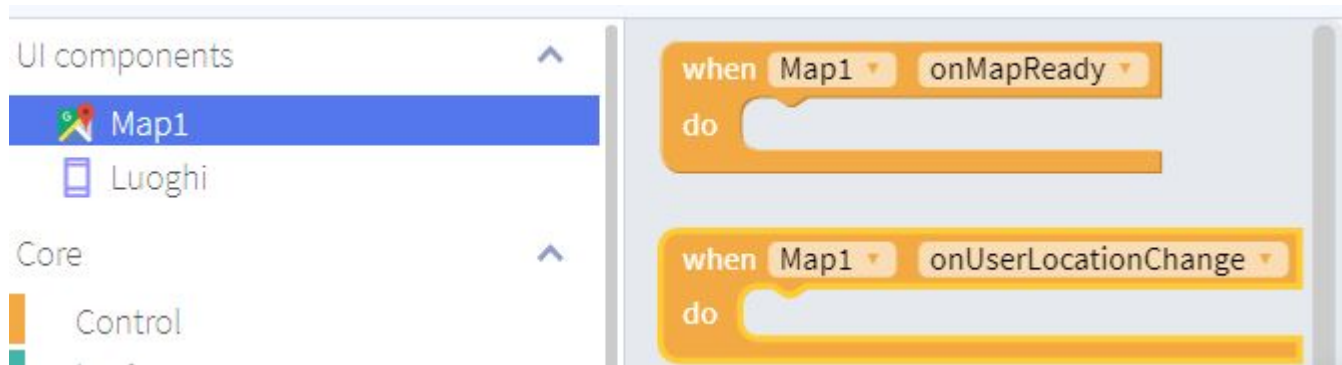
true

Shows Traffic

false

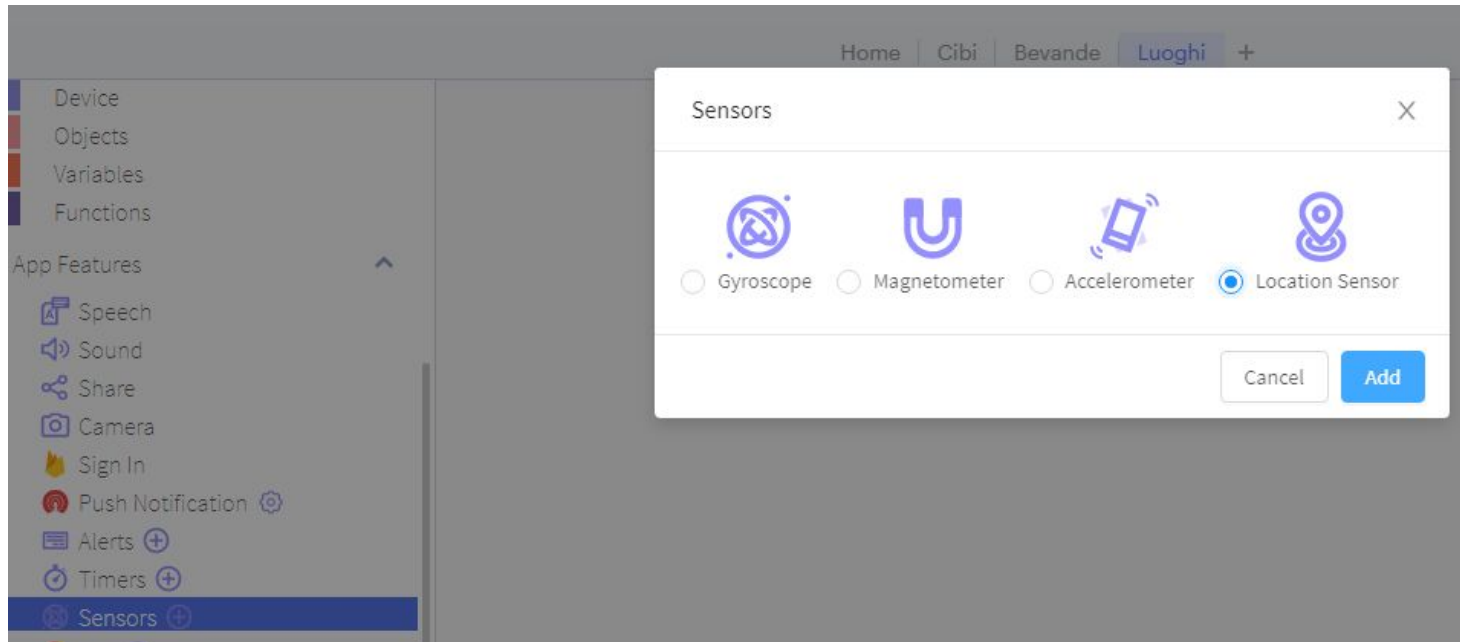
Blocco Mappa

Andiamo a trascinare nel desk un blocco evento associato alla mappa



Sensore di posizione

Inizializziamo il location sensor (gps del dispositivo)



Sensore di posizione - parametri

Impostiamolo con i parametri di default

Location_Sensor1 

▼ EnableHighAccuracy

☐ false

▼ Timeout

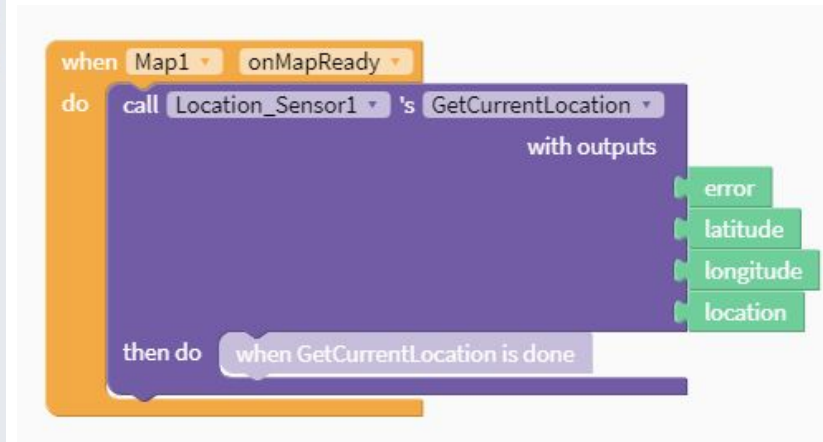
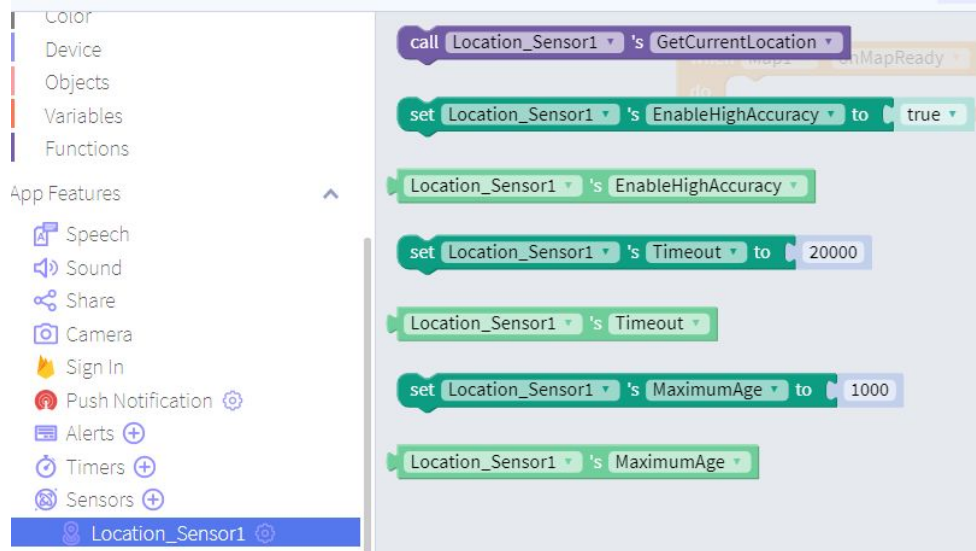
▼ Maximum Age

Delete

Submit

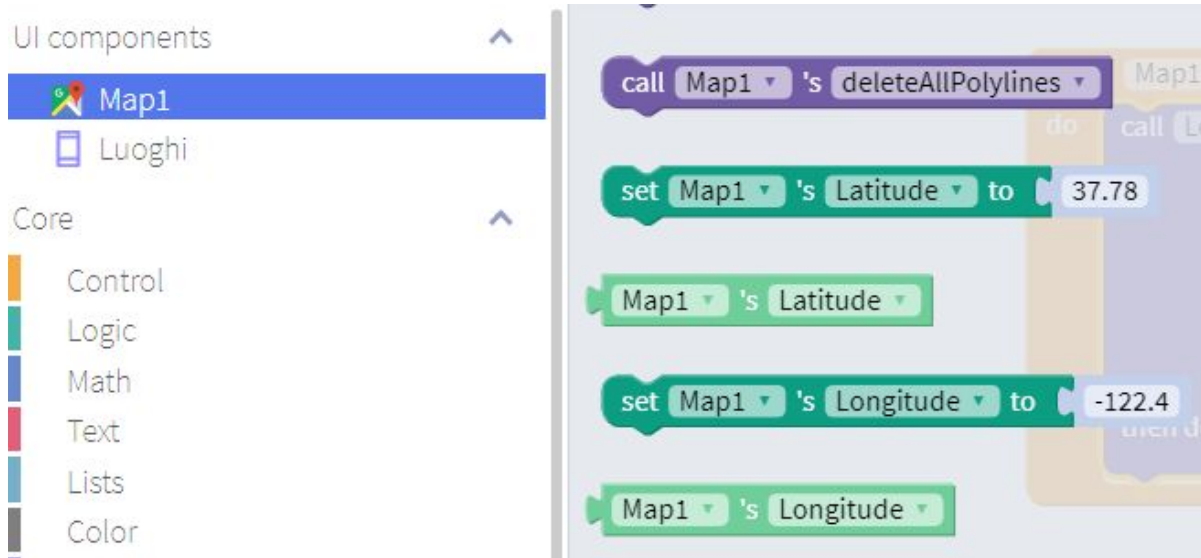
Chiamata del sensore di posizione nel blocco mappa

Ora trasciniamo la chiamata del sensore di posizione all'interno del blocco evento. Il sensore di posizione fornisce 4 output: errore, latitudine, longitudine e posizione.



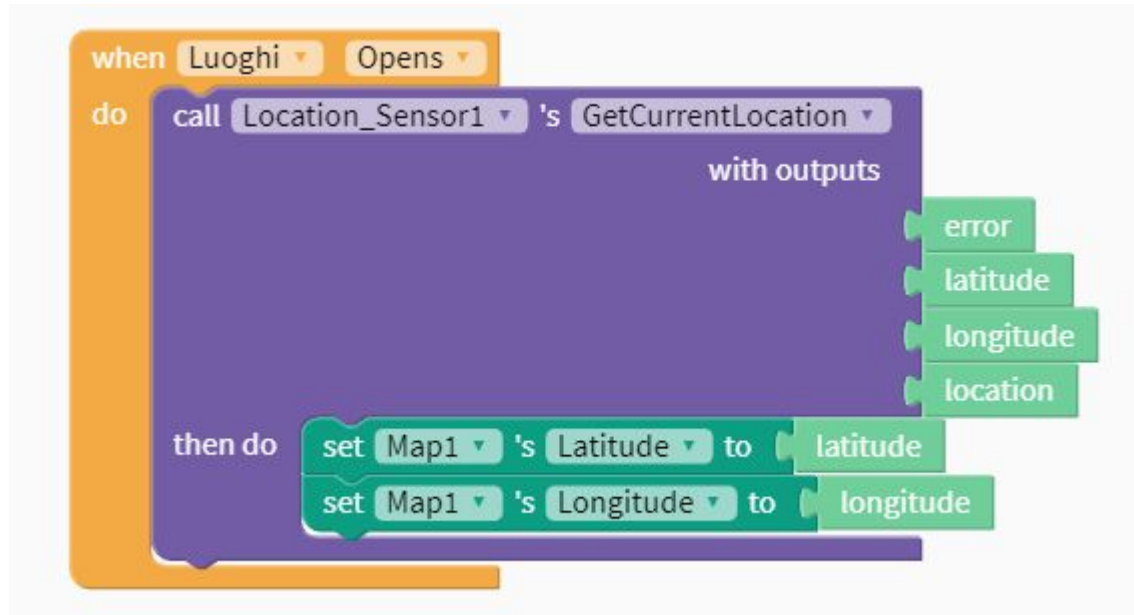
Sensore di posizione

Selezioniamo il blocco “set Map1’s Latitude to xx.yy” e trasciniamolo nel desk.
Facciamo lo stesso per la longitudine.



Mappa - blocchi

Impostiamo la mappa perchè sia centrata sulle coordinate ricevute dal sensore gps (trascinando i blocchi latitude e longitude provenienti dal sensore di posizione come attributi della mappa)



Mappa funzionante in app

Proviamo la app e vediamo che viene mostrato un pallino blu sulla mappa in corrispondenza della nostra posizione (coordinate gps di latitudine e longitudine). Componendo altri blocchi è possibile creare delle regole o impostare su mappa un percorso o dei punti di interesse.



App dado parlante

Create una app che cliccando un pulsante effettui il lancio casuale di un dado a 6 facce, ci dica il numero ottenuto e visualizzi il numero ottenuto in un altro componente dell'interfaccia.

Basta aggiungere allo screen un pulsante che chiamiamo “lancia” e un altro componente (ad esempio etichetta) che chiamiamo “dado”.

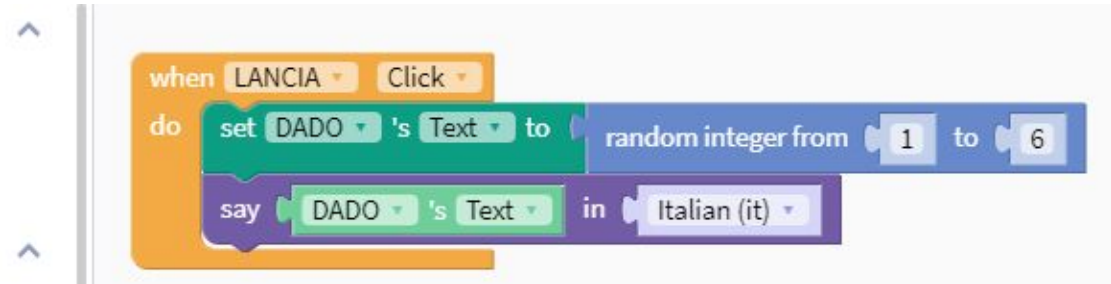
UI components

DADO

LANCIA

Screen1

Core

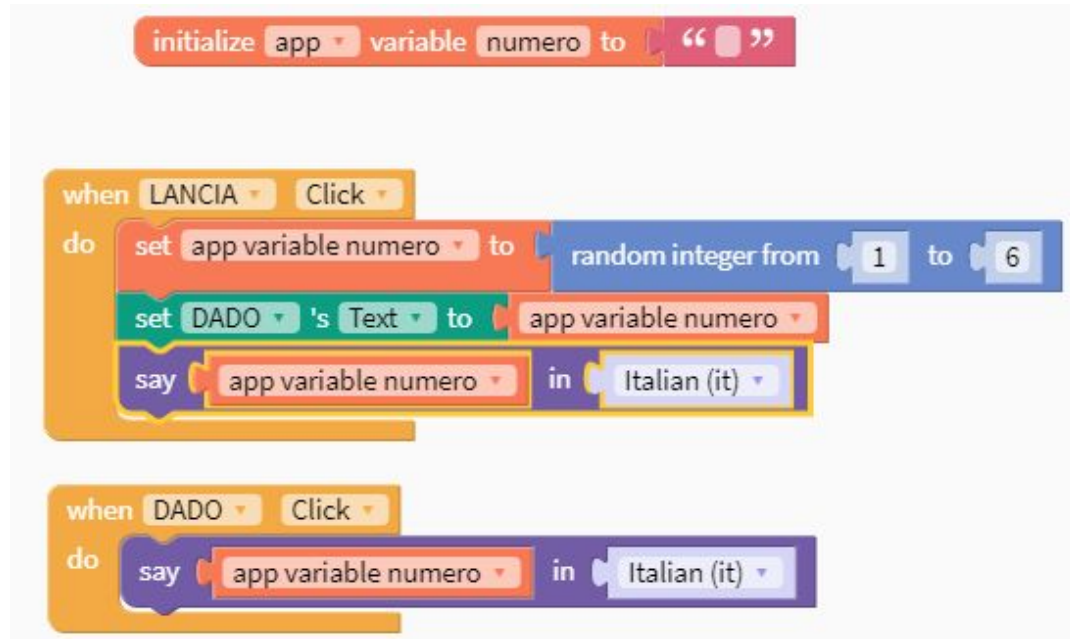


App dado parlante

Se non si è sentito il numero uscito e si vuole riascoltare senza dover rilanciare il dado si può inserire un evento sul secondo componente: cliccando viene riletto il numero che era uscito.



App dado parlante - altra soluzione per chi avesse inizializzato una variabile



Realizziamo una app che scatti una foto e scriva la didascalia in automatico



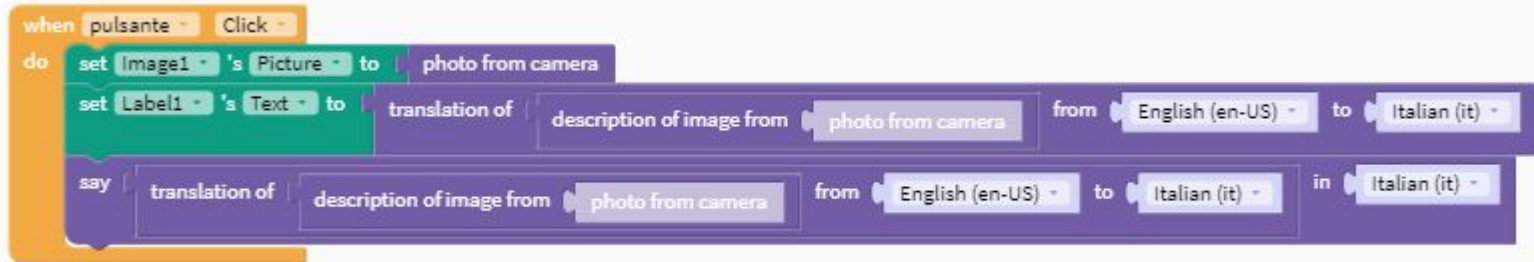
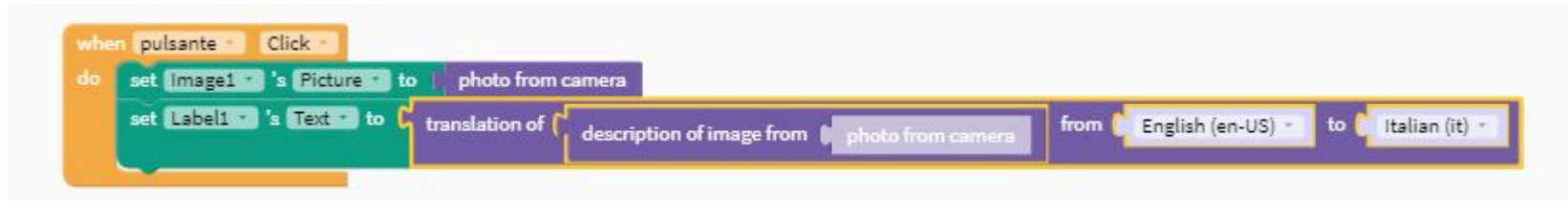
Inseriamo una immagine, una etichetta e un pulsante.

Nei blocchi usiamo il modulo fotocamera e il riconoscitore (Microsoft Image Recognition)



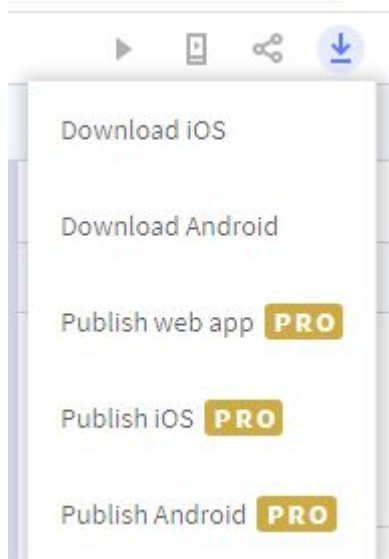
Traduttore

Modifichiamo la app perchè scriva la didascalia in italiano anzichè in inglese; poi aggiungiamo il modulo traduttore da inglese a italiano e attiviamo il modulo TTS cosicchè la app legga ad alta voce la didascalia.



Download della app su smartphone Android o iOS

Potete ricevere la vostra app via email per condividerla con altri dispositivi. La versione free è limitata a 2 progetti scaricati al mese.



Your app riconoscitore is ready!

Click the button below to download your app.
Please note that this link will expire in 24 hours from the time you received this email.

DOWNLOAD YOUR APK

If you are using Google Chrome, your browser settings may prevent you from downloading your APK file. Simply open this email in Firefox or Safari and download your APK file.

Before installing this app on your Android device, you will need to [trust Thunkable](#).
You only need to do this once.

Documentazione utile

<https://docs.thunkable.com/>