



Esercitazione

thinkable

Giulio Galesi, ISTI-CNR

Blocco funzione in Thinkable

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.

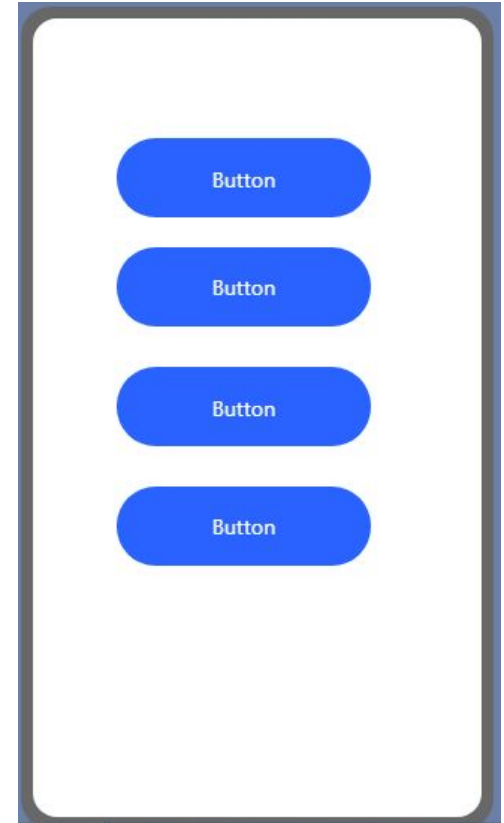
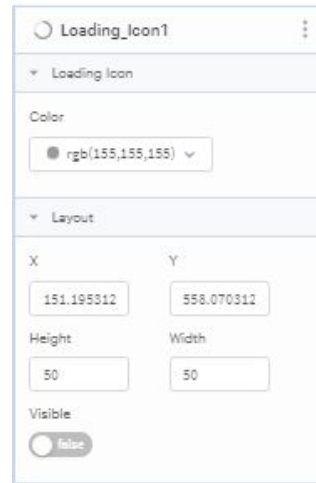
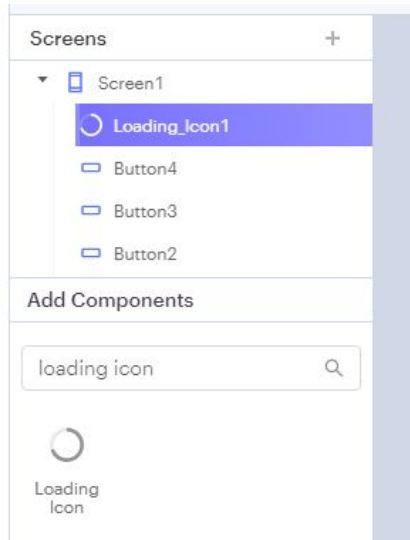
Proviamo a realizzare una semplice funzione che poi andremo a richiamare in altri blocchi.

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

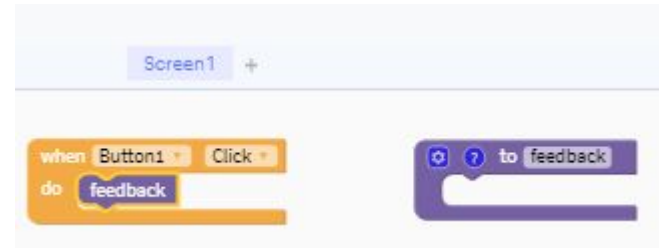
Inseriamo un blocco evento associato al click di uno dei 4 pulsanti.

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



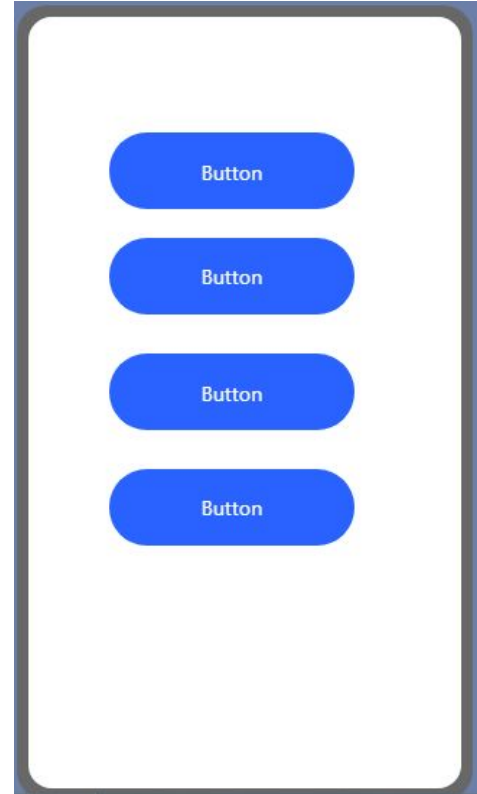
Creiamo una funzione 2

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.

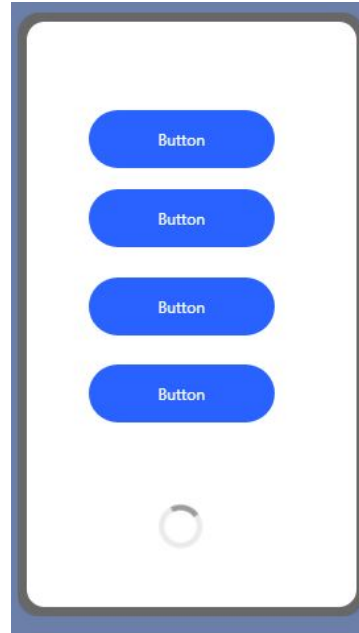
The screenshot displays a visual programming environment with a 'UI components' panel on the left and a main workspace. The 'UI components' panel lists 'Loading_Icon1' as the selected component. The workspace contains several code blocks for 'Loading_Icon1':

- set Loading_Icon1's color to [grey]
- Loading_Icon1's color
- set Loading_Icon1's size to [extra small]
- Loading_Icon1's size
- Loading_Icon1's Computed Height
- Loading_Icon1's Computed Width
- set Loading_Icon1's Visible to [true]
- Loading_Icon1's Visible

A 'when Button1 do feedback' block is also present. A callout box shows a zoomed-in view of the 'feedback' function block, which contains the block: 'set Loading_Icon1's Visible to [true]'. The 'UI components' panel also lists other components: Button4, Button3, Button2, Button1, and Screen1. The 'Core' panel lists categories: Control, Logic, Math, Text, Lists, Color, and Device.

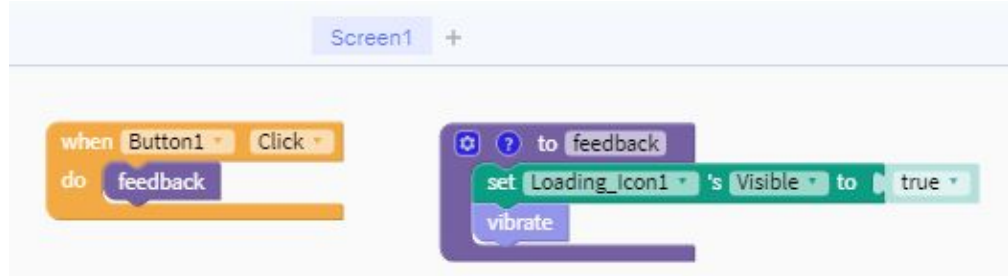
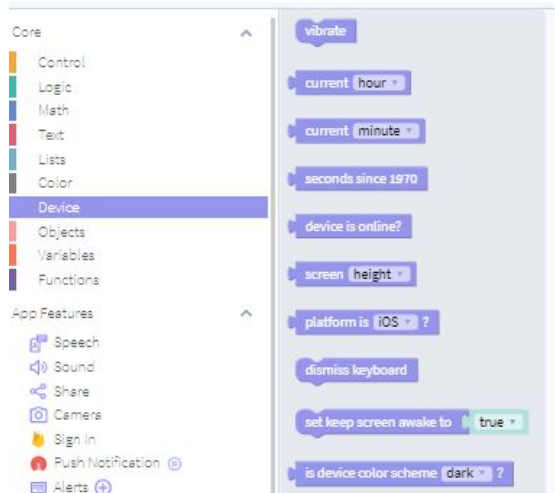
Proviamo la funzione con istruzione per feedback visivo

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



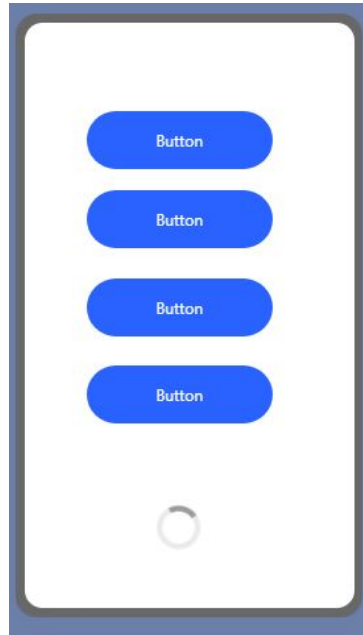
Aggiungiamo le istruzioni alla funzione - feedback tattile

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



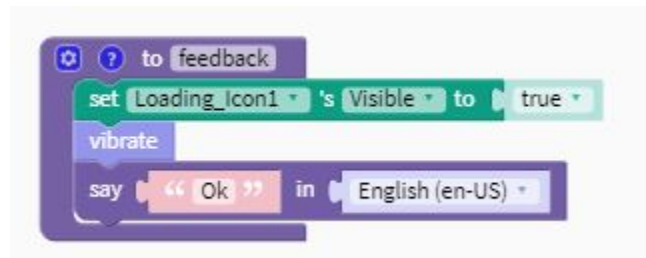
Proviamo la funzione a cui è stato aggiunto feedback tattile

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) .



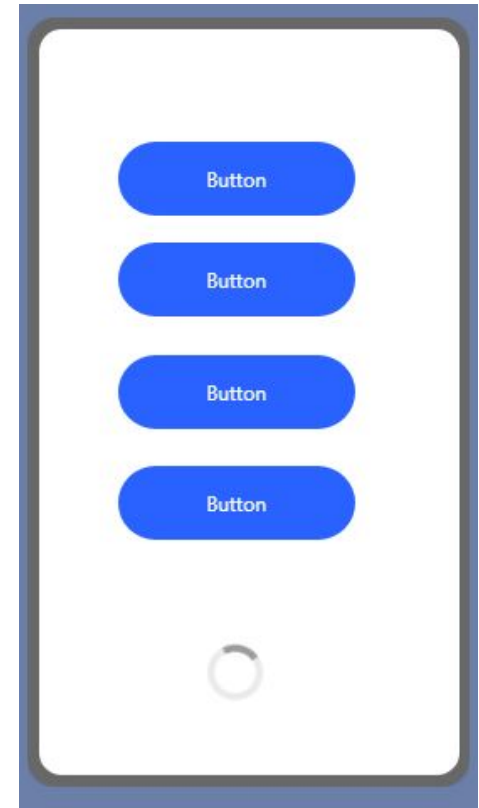
Aggiungiamo le istruzioni alla funzione - feedback audio

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..

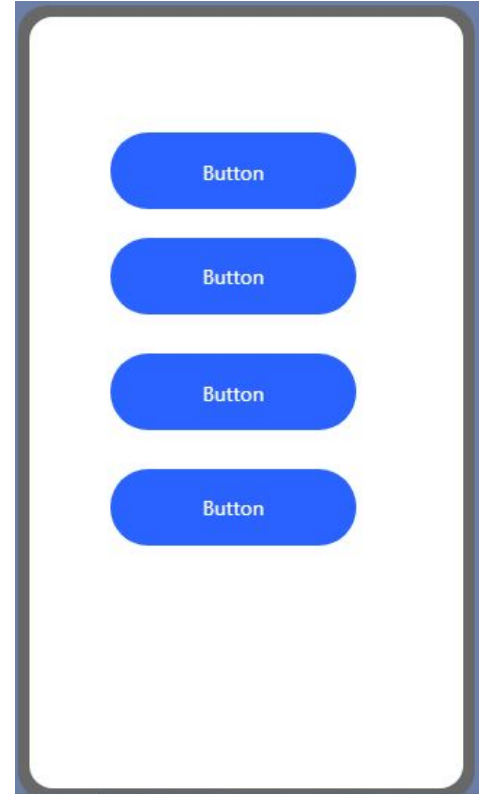
Inseriamo una pausa di 1 secondo nella funzione, duplichiamo il blocco dell'icona di caricamento e impostiamo l'attributo di visibilità su false in modo che l'icona di caricamento scompaia dopo che sono state eseguite le varie istruzioni.

The image shows a sequence of steps in the Scratch code editor to modify a function:

- Top Left:** A sidebar with categories: Core, Control, Logic, Math, Text, Lists. The 'Control' category is selected.
- Top Middle:** A 'do' block containing an 'else' block and a 'wait 1 seconds' block.
- Top Right:** A 'to feedback' function block containing:
 - 'set Loading_icon1's Visible to true' (green block)
 - 'vibrate' (purple block)
 - 'say "Ok" in English (en-US)' (purple block)
 - 'wait 1 seconds' (orange block)A context menu is open over the 'set Loading_icon1's Visible to true' block, showing options: Duplicate, Add Comment, Collapse Block, Delete 2 Blocks, Help, and Undo last delete.
- Bottom:** The final 'to feedback' function block, which has been updated to include a second 'set Loading_icon1's Visible to false' (green block) at the end of the sequence.

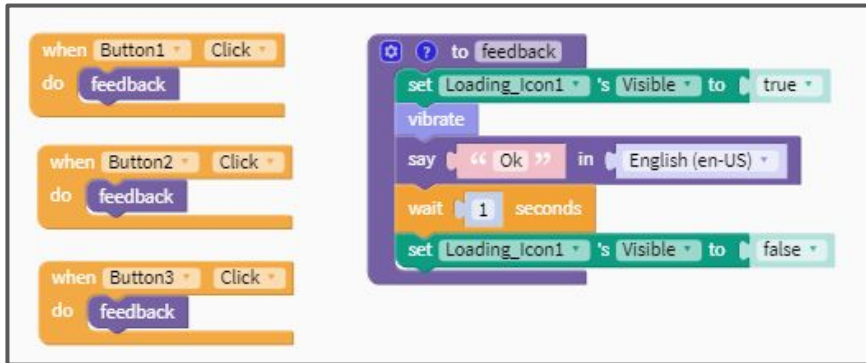
Proviamo la funzione a cui è stata aggiunta una pausa prima di disattivare la loading icon

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..

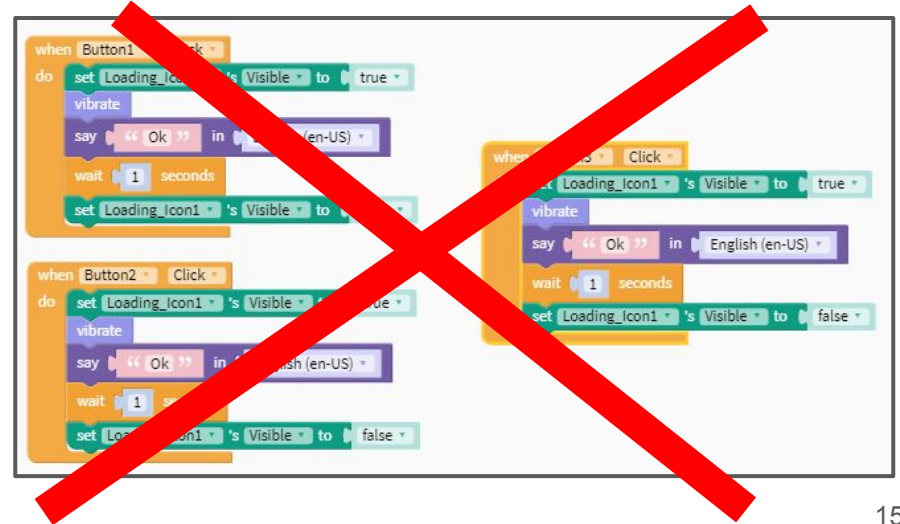


```
when Button1 Click
do feedback

when Button2 Click
do feedback

when Button3 Click
do feedback

to feedback
set Loading_Icon1's Visible to true
vibrate
say "Ok" in English (en-US)
wait 1 seconds
set Loading_Icon1's Visible to false
```



```
when Button1 Click
do
set Loading_Icon1's Visible to true
vibrate
say "Ok" in English (en-US)
wait 1 seconds
set Loading_Icon1's Visible to false

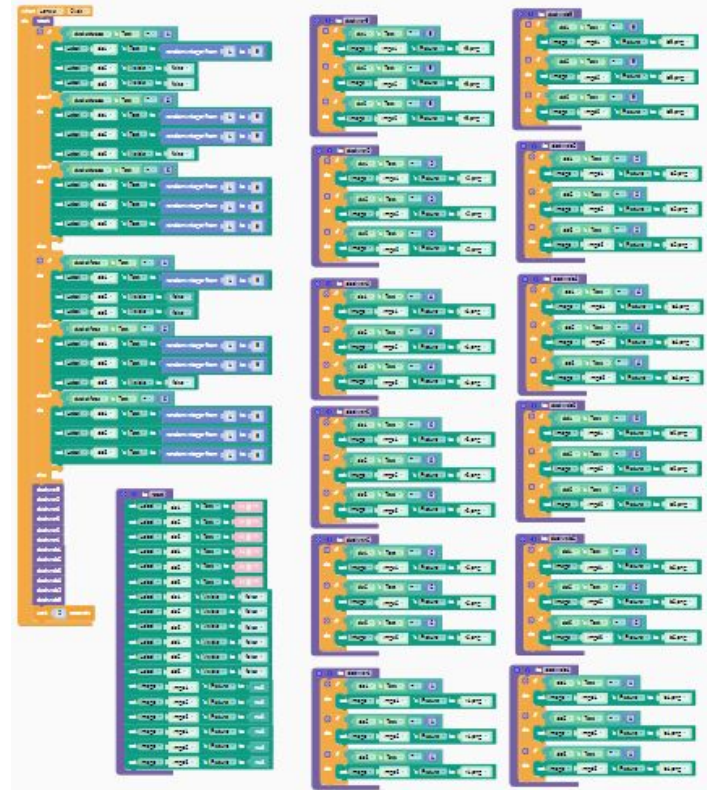
when Button2 Click
do
set Loading_Icon1's Visible to true
vibrate
say "Ok" in English (en-US)
wait 1 seconds
set Loading_Icon1's Visible to false

when Button3 Click
do
set Loading_Icon1's Visible to true
vibrate
say "Ok" in English (en-US)
wait 1 seconds
set Loading_Icon1's Visible to false
```

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 for handling button clicks across the UI.

The top-left screenshot shows the "Advanced" view of the "Any Button" block. The "do" block contains a "call Screen" block, which is configured to call the "ToggleDrawerMenu" method of the "Screen" component.

The bottom-left screenshot shows the "Any Button" block with a "do" block containing a "feedback" block. A "component" label is visible next to the "Any Button" block.

The right screenshot shows the "Any Button" block with a "do" block containing 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.

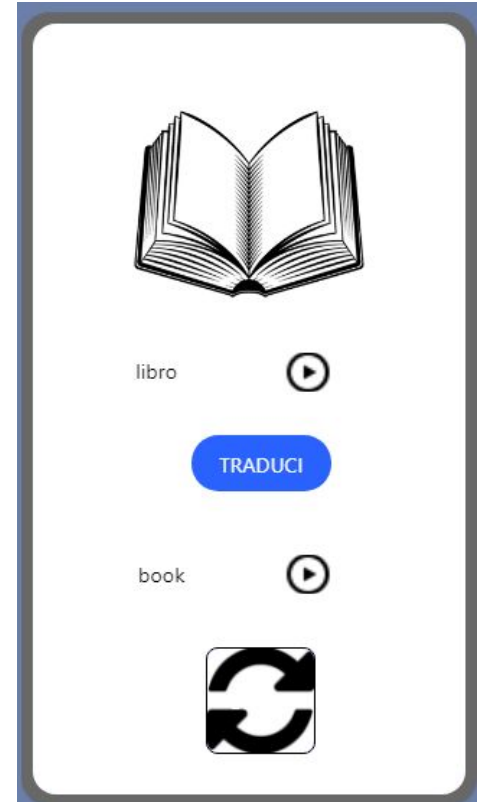
Sviluppiamo alcune app insieme passo per passo

- 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

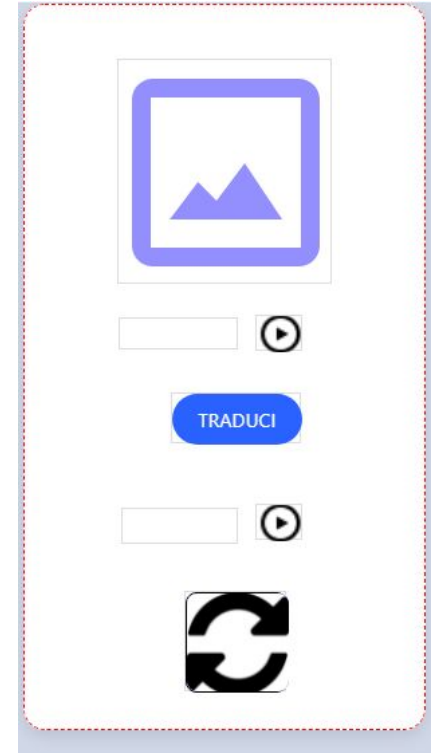


App per insegnare ai bambini una lingua straniera - UI

Inseriamo nel design:

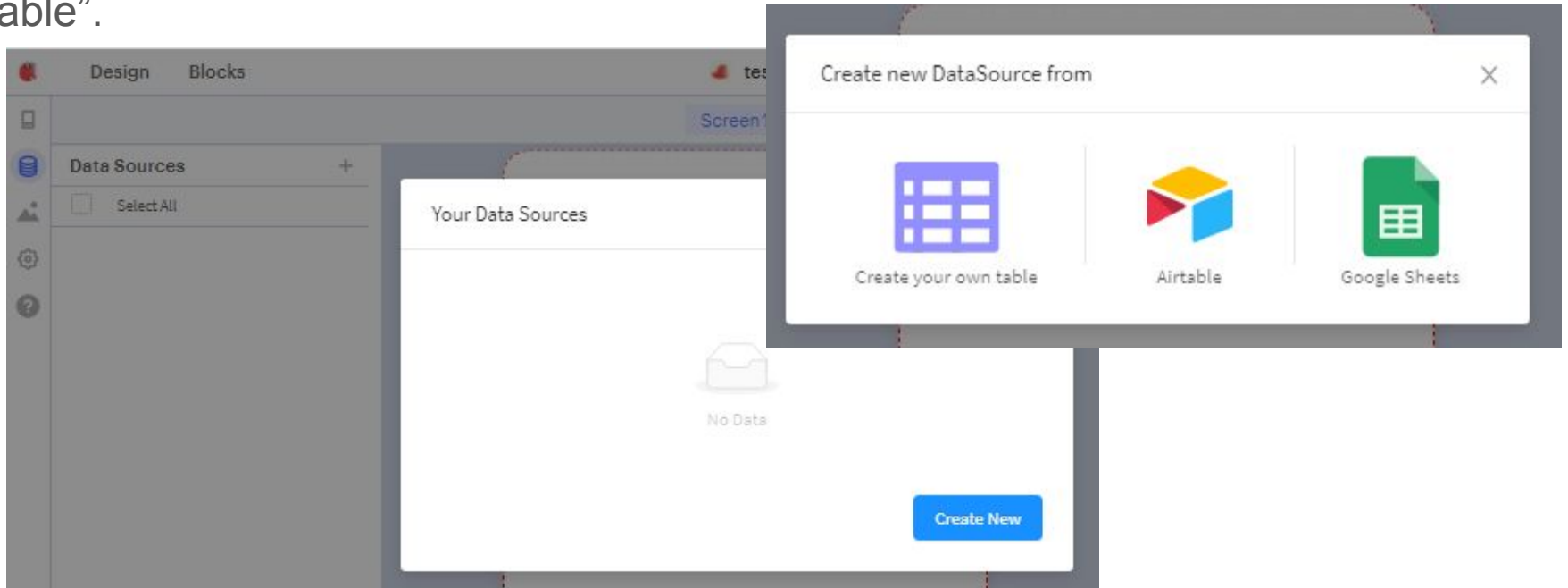
1. Un componente immagine
2. Due etichette
3. Quattro pulsanti

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.



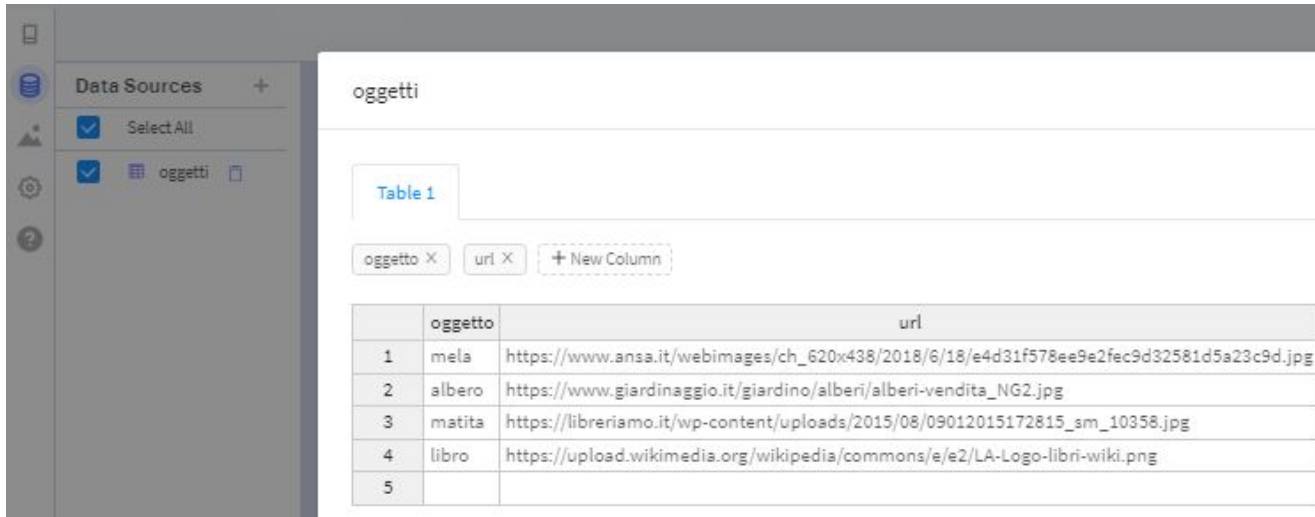
App per insegnare ai bambini una lingua straniera - DB

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.

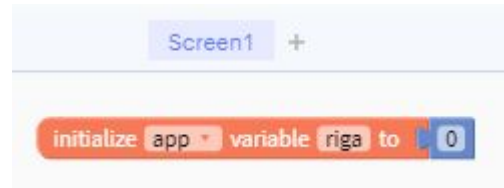


The screenshot shows a data table editor interface. On the left, there is a sidebar with 'Data Sources' and a table named 'oggetti'. The main area displays the table 'oggetti' with two columns: 'oggetto' and 'url'. The table contains five rows of data. Above the table, there are buttons for 'Table 1',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

	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”.

The image displays the MIT App Inventor interface. On the left, a sidebar shows the 'Math' category selected. The main workspace contains a code block for 'random integer from 1 to 100'. To the right, a 'Table 1' component is visible with a 'do' block containing an 'immagine random' block. Below this, a 'number of rows in oggetti in Table 1' block is highlighted. The bottom section shows a 'Screen1' component with an 'initialize' block setting 'app variable riga' to 0, and a 'to imagine random' block containing a 'set app variable riga to random integer from 1 to number of rows in oggetti in Table 1' block.

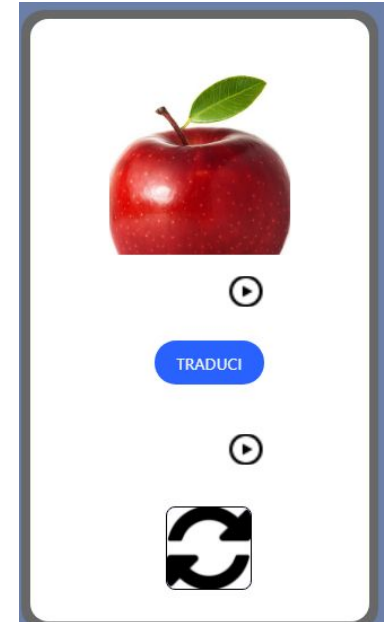
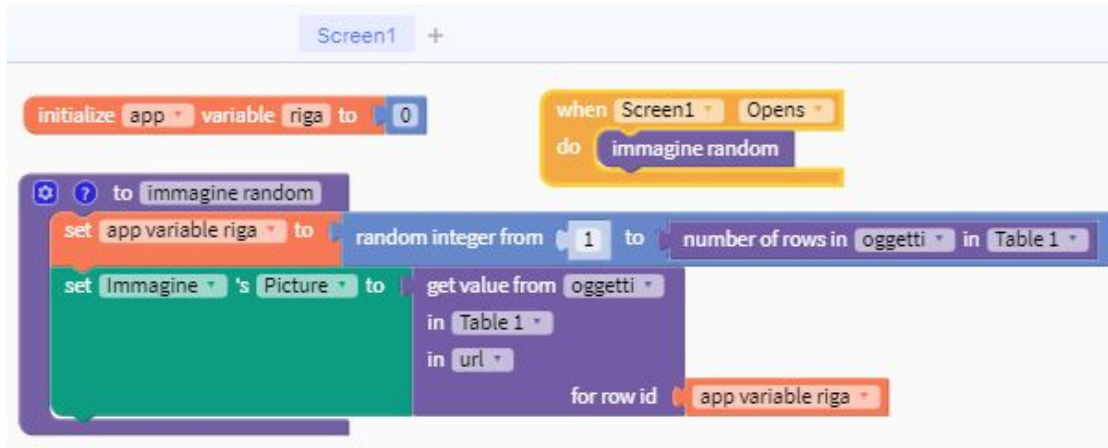
App per insegnare ai bambini una lingua straniera - recuperare valore da DB per impostare immagine

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

The image shows a development environment with a sidebar on the left containing menu items: Sign In, Push Notification, Data Sources (highlighted), Alerts, and Timers. The main workspace displays two code snippets. The top snippet is a purple block for 'get value from oggetti' in 'Table 1' for row id 'id'. The bottom snippet is a larger block starting with 'to immagine random', followed by 'set app variable riga to random integer from 1 to number of rows in oggetti in Table 1', and 'set Immagine's Picture to get value from oggetti in Table 1 in url for row id app variable riga'.

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.

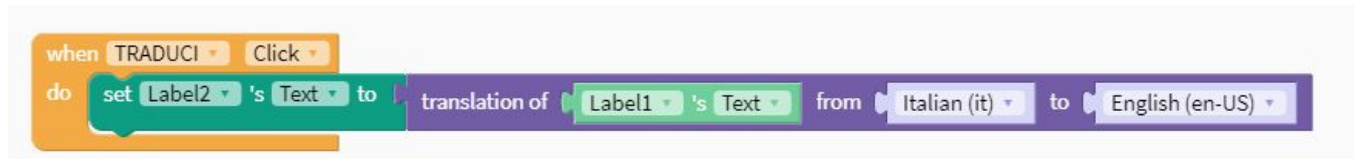
The screenshot shows the code editor for a screen named "Screen1". The logic is as follows:

- when Screen1 Opens** (orange block) does:
 - initialize app variable riga to 0** (orange block)
 - immagine random** (purple block)
- to immagine random** (purple block) contains:
 - set app variable riga to random integer from 1 to number of rows in oggetti in Table 1** (orange block)
 - set Immagine 's Picture to get value from oggetti in Table 1 in url** (purple block)
 - set Label1 's Text to get value from oggetti in Table 1 in oggetto** (purple block)

The "for row id" blocks are connected to the "set" blocks, with the "app variable riga" variable used as the row identifier.

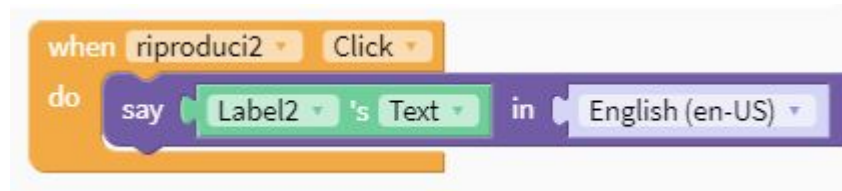
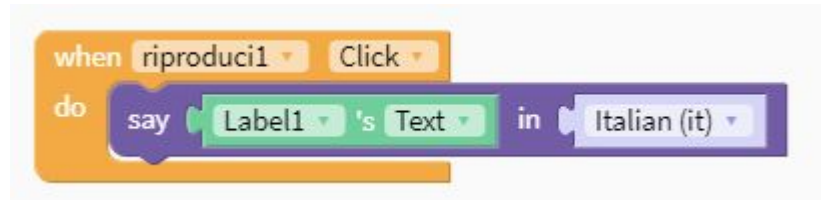
App per insegnare ai bambini una lingua straniera - pulsante traduci

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

Community Projects Public Gallery My Data Sources

ame...

Create New App

Create New Project

New Project Name:

anziani

Category:

Please select category (at most 6).

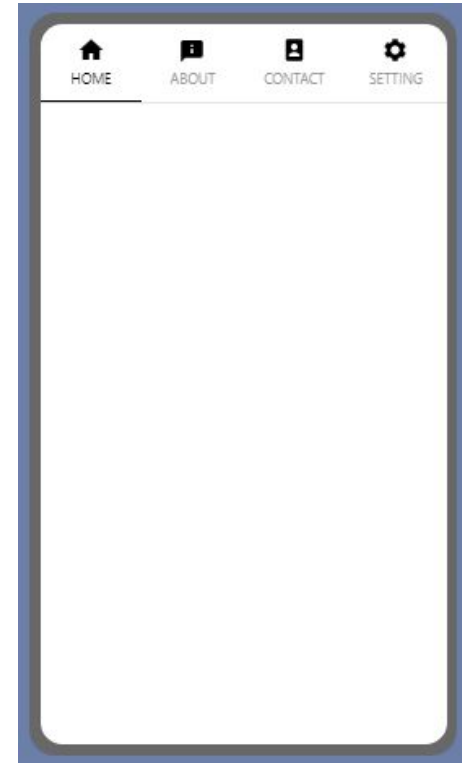
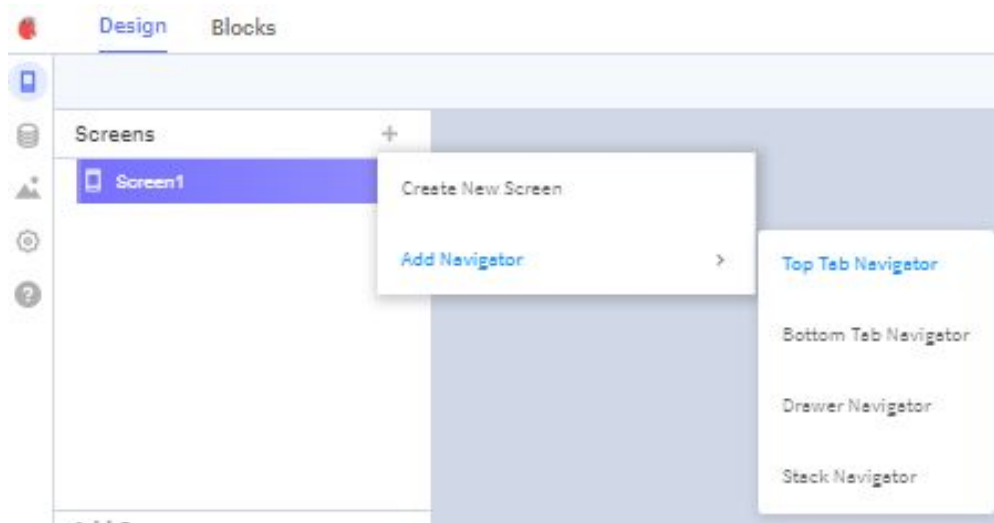
Public Everyone can access this project [here!](#)

Try our drag and drop interface [?](#) Try it out

Cancel Create

App per anziani

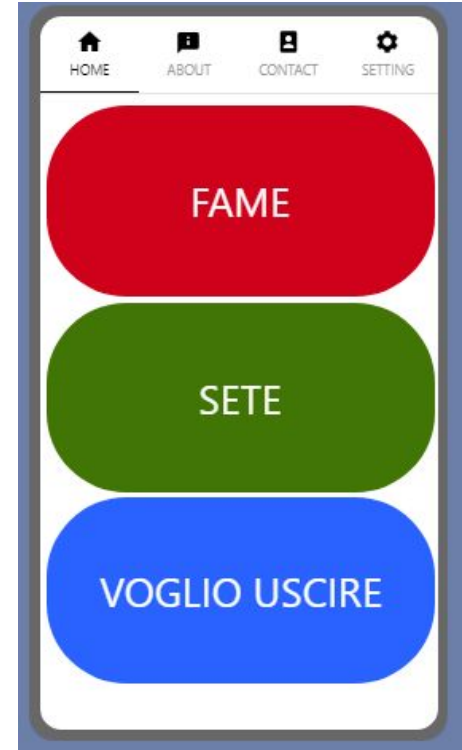
Per prima cosa aggiungiamo un Top Tab navigator cliccando sul “+” ed eliminiamo Screen1. Vengono create in automatico quattro schermate con un menu in alto con 4 icone di navigazione.



App per anziani

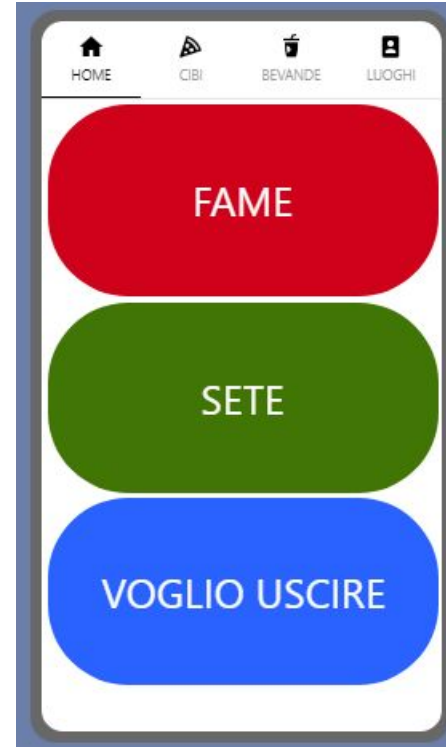
Aggiungiamo in Home tre pulsanti (inseriamo prima un button e lo duplichiamo due volte).

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/>



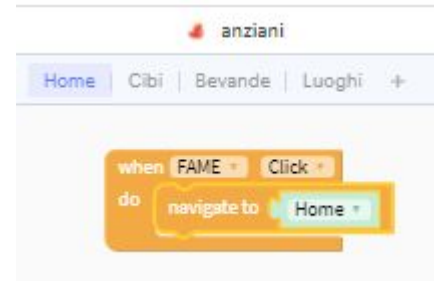
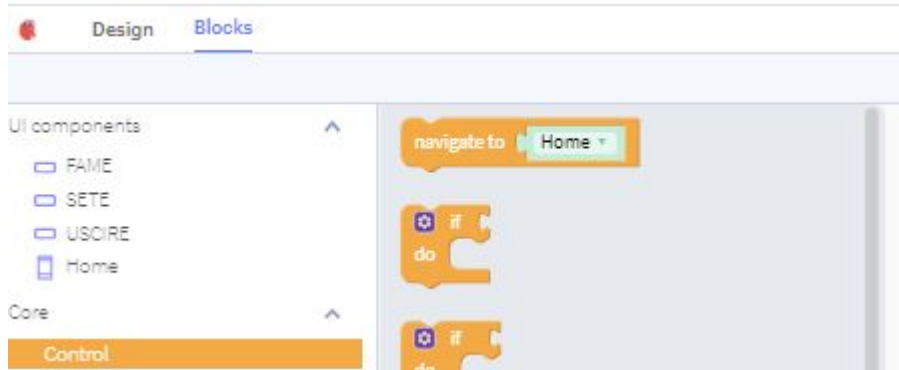
App per anziani

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



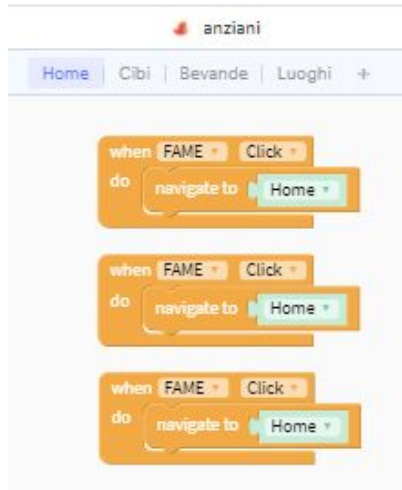
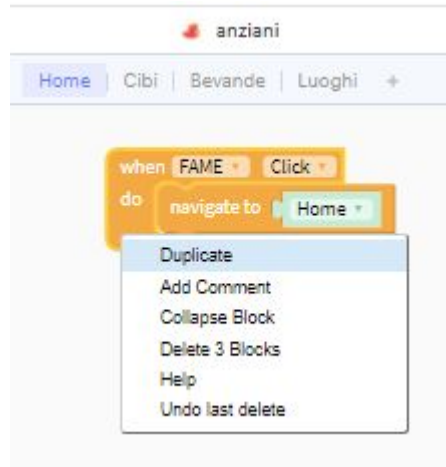
App per anziani

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



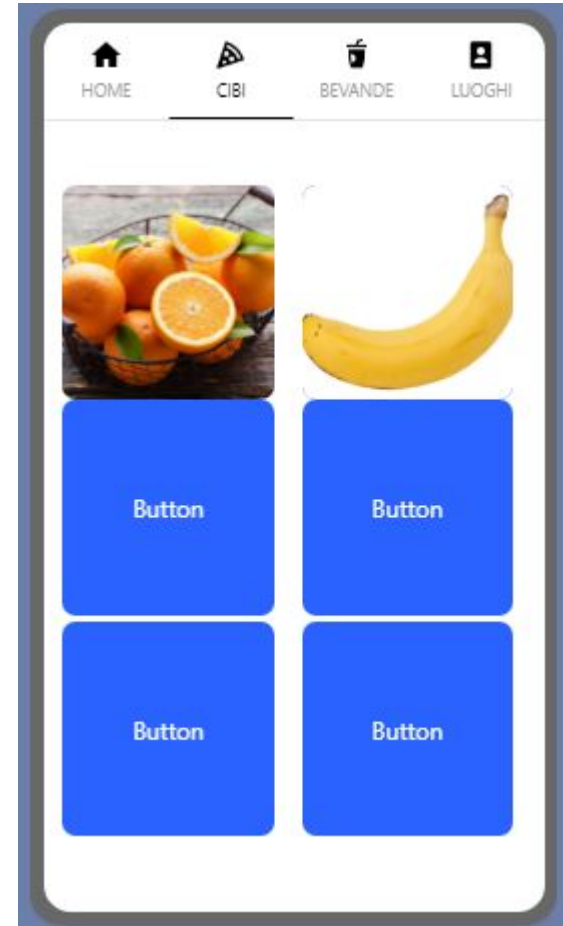
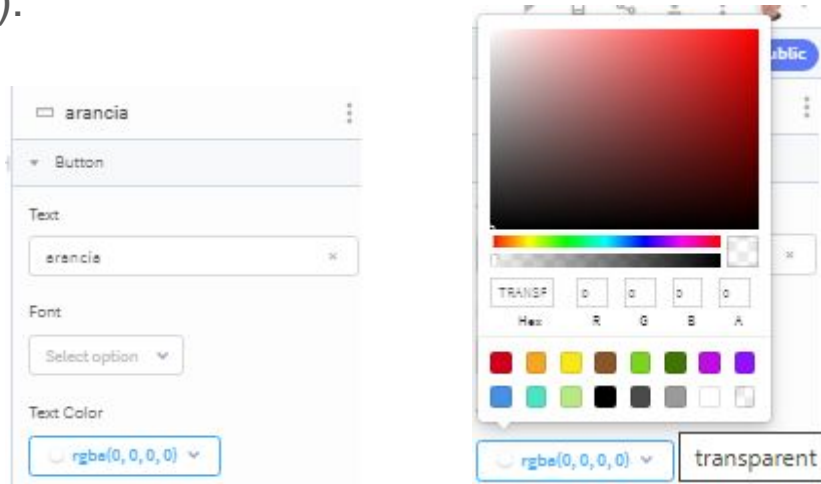
App per anziani

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.



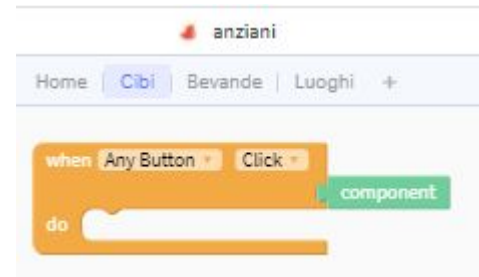
App per anziani

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).



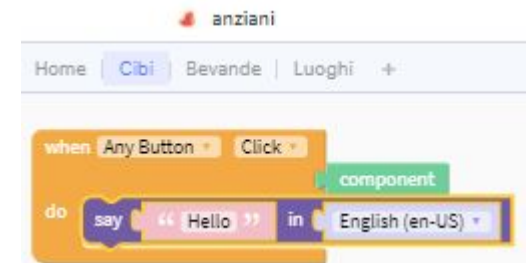
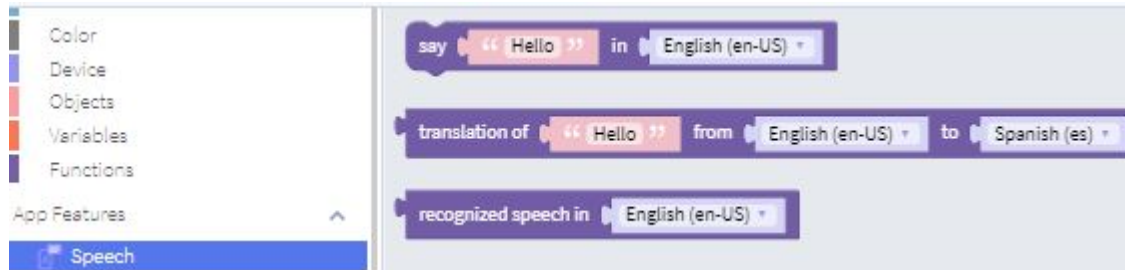
App per anziani

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”.

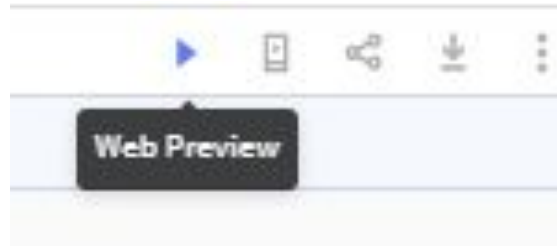


App per anziani

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

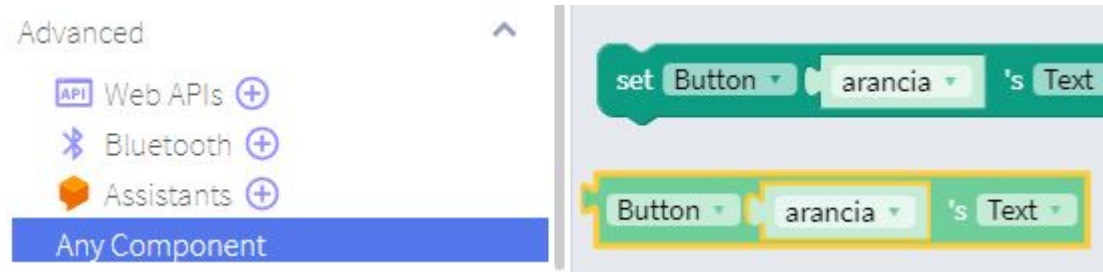
Notiamo che qualsiasi pulsante noi premiamo, viene riprodotto il saluto Hello!

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”.



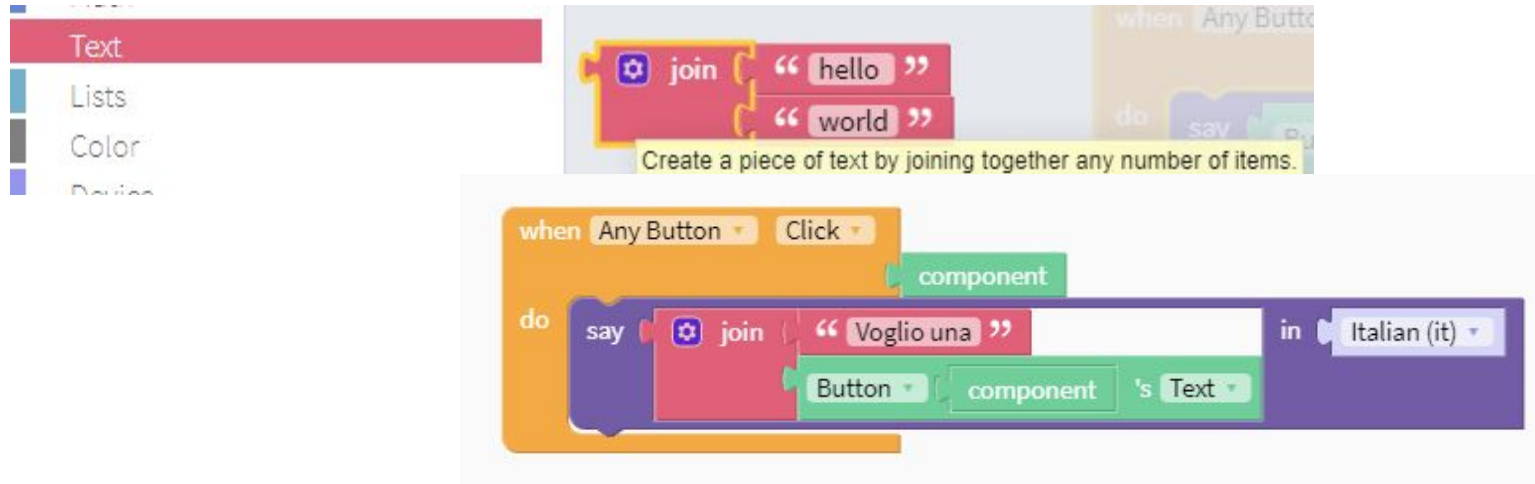
App per anziani

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



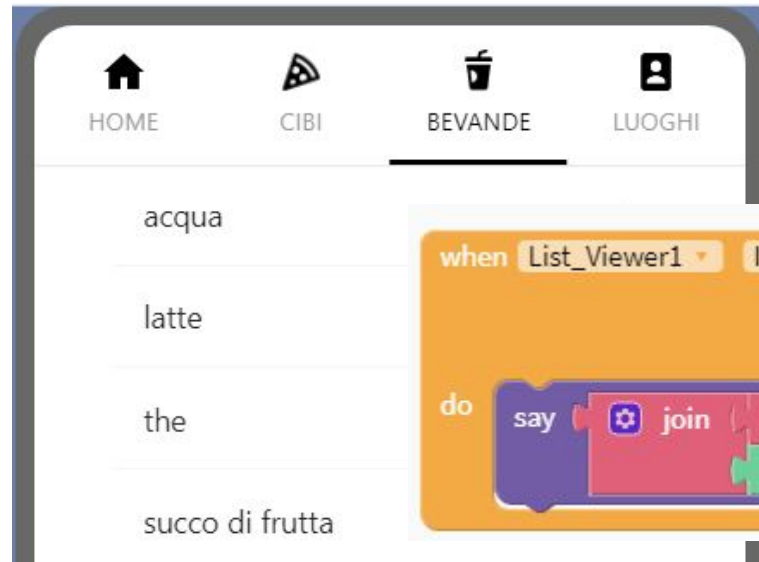
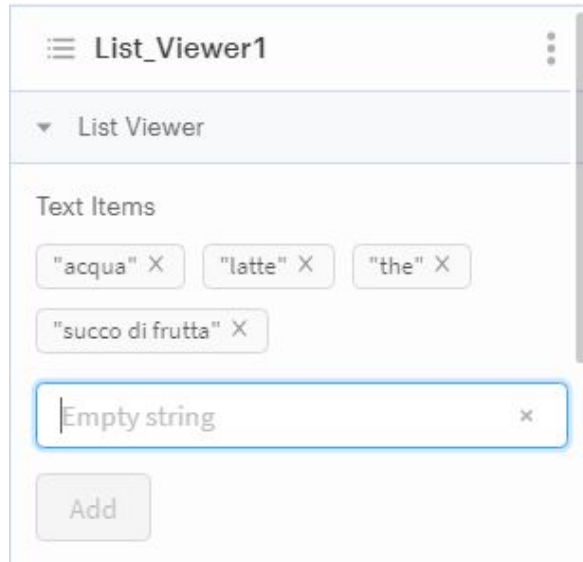
App per anziani

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”

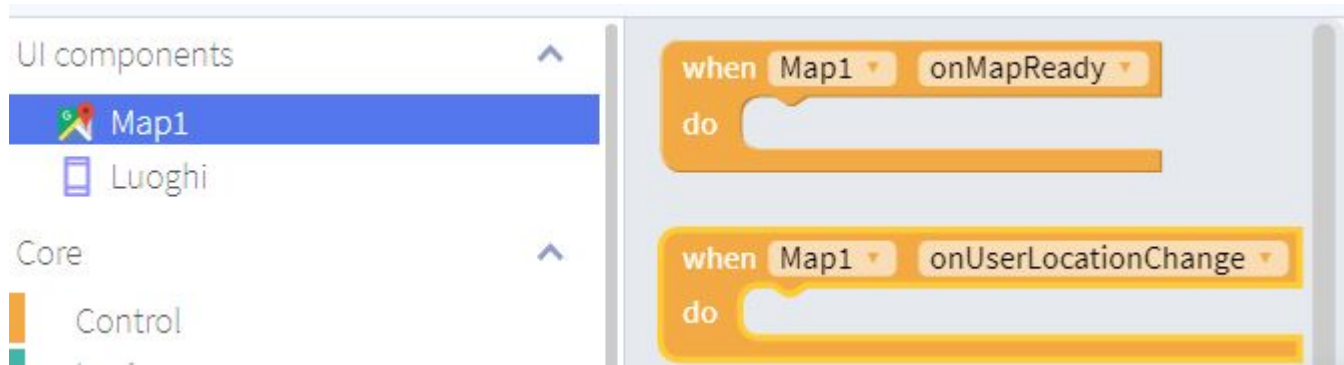
The image shows the Android Studio interface for adding a component to an app. The 'Add Components' dialog is open, and the search bar contains the text 'map'. Below the search bar, the 'Map' component is selected, indicated by a blue highlight. The 'Map' component is also visible in the 'Add Components' list, with a trash icon and a person icon next to it. The 'Map' component is also visible in the 'Add Components' list, with a trash icon and a person icon next to it. The 'Map' component is also visible in the 'Add Components' list, with a trash icon and a person icon next to it.

The 'Map1' properties panel is visible, showing options for API keys, map type, and user location. The 'Shows User Location' and 'Shows My Location Button (Android Only)' options are checked.

Please view the Map component on the physical device.

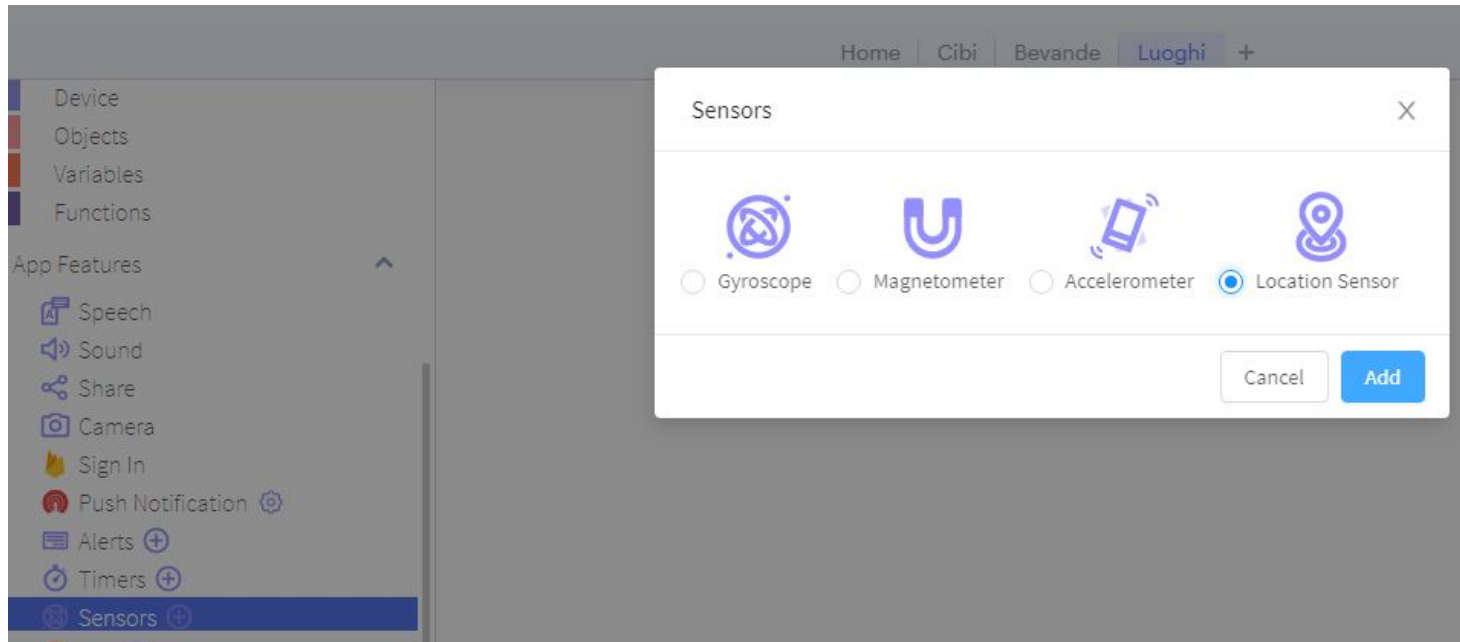
Blocco Mappa

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.

Color
Device
Objects
Variables
Functions

App Features

- Speech
- Sound
- Share
- Camera
- Sign In
- Push Notification
- Alerts
- Timers
- Sensors
- Location_Sensor1

call Location_Sensor1's GetCurrentLocation

set Location_Sensor1's EnableHighAccuracy to true

Location_Sensor1's EnableHighAccuracy

set Location_Sensor1's Timeout to 20000

Location_Sensor1's Timeout

set Location_Sensor1's MaximumAge to 1000

Location_Sensor1's MaximumAge

when Map1 onMapReady

do

call Location_Sensor1's GetCurrentLocation

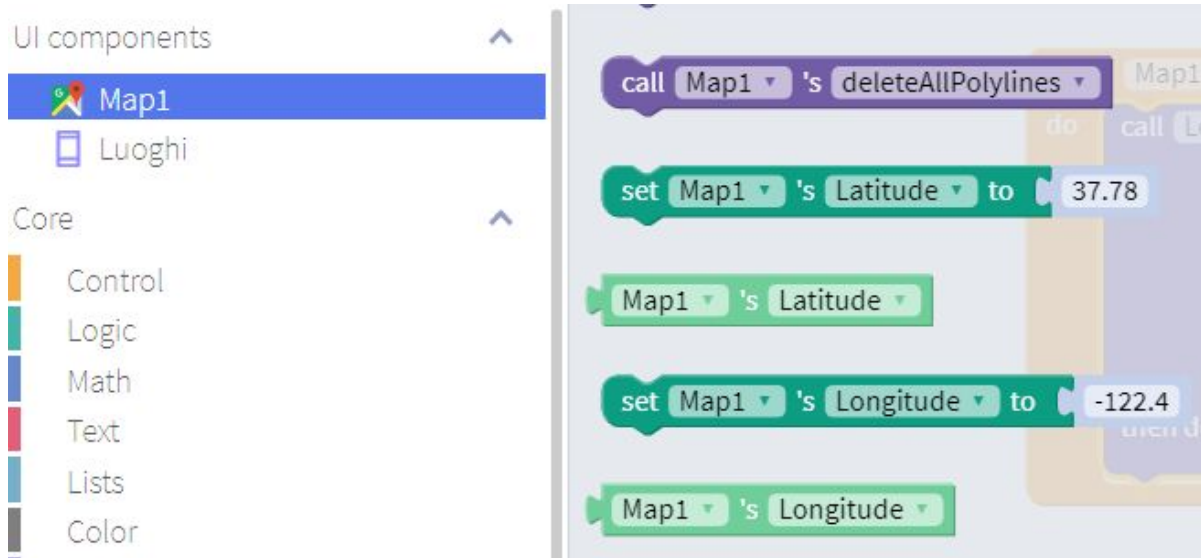
with outputs

- error
- latitude
- longitude
- location

then do when GetCurrentLocation is done

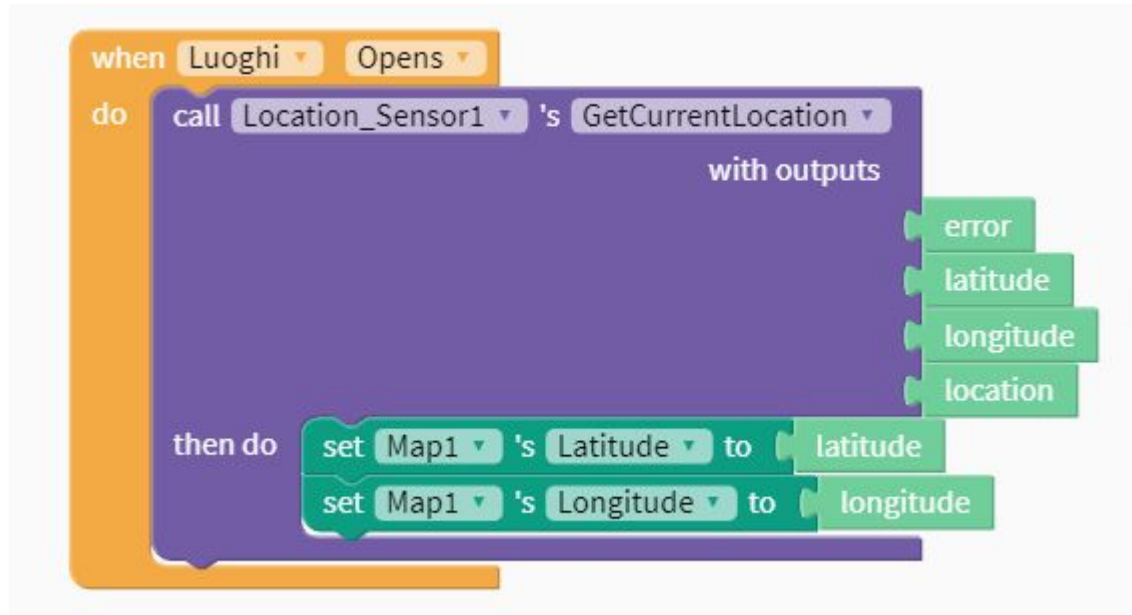
Sensore di posizione

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

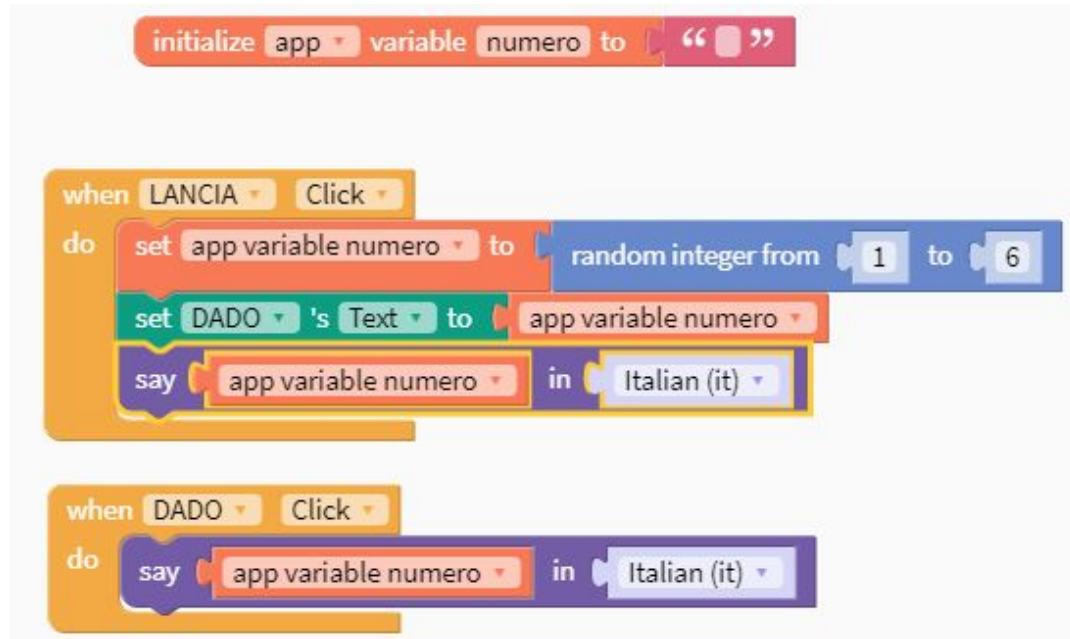


App dado parlante

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



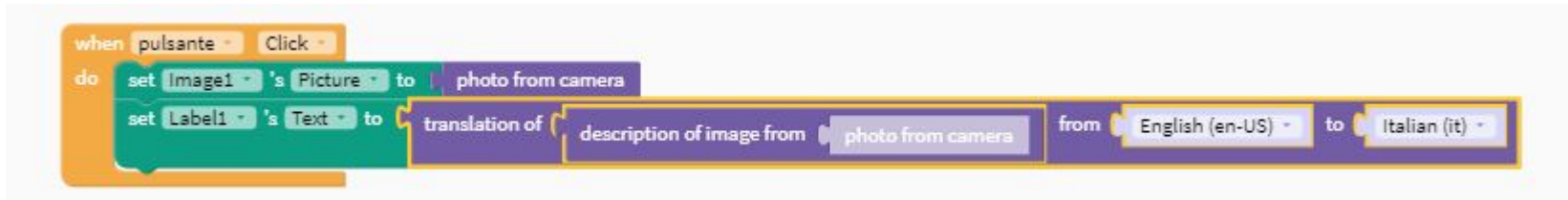
Inseriamo una immagine, una etichetta e un pulsante.

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

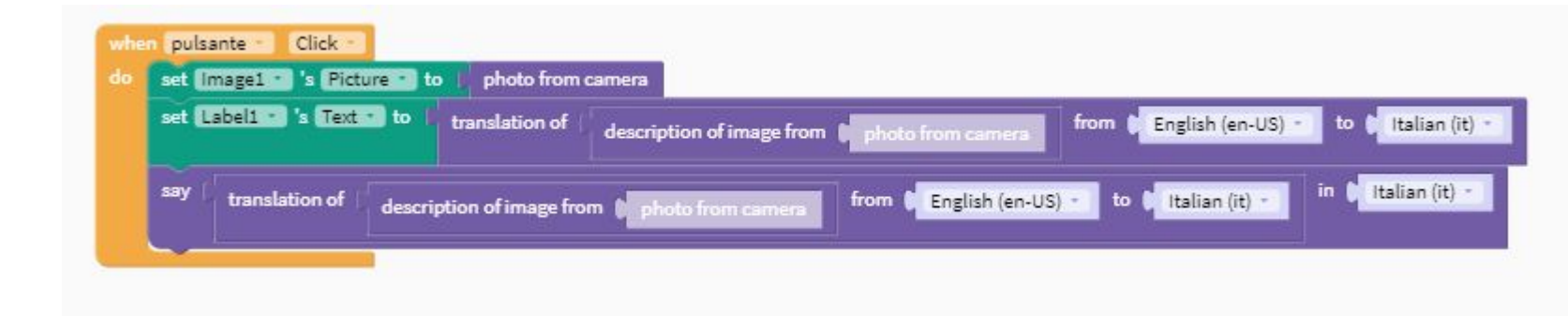


Traduttore

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.



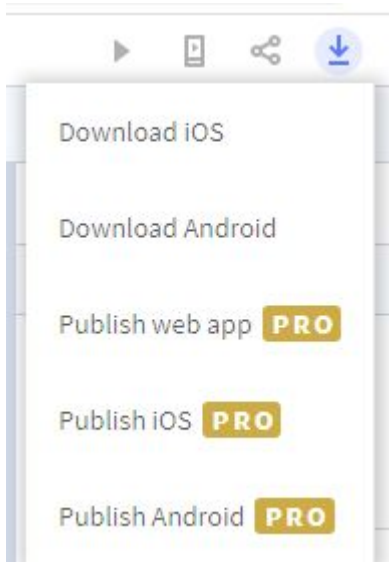
```
when pulsante - Click -  
do  
  set Image1 - 's Picture - to photo from camera  
  set Label1 - 's Text - to translation of description of image from photo from camera from English (en-US) - to Italian (it) -
```



```
when pulsante - Click -  
do  
  set Image1 - 's Picture - to photo from camera  
  set Label1 - 's Text - to translation of description of image from photo from camera from English (en-US) - to Italian (it) -  
  say translation of description of image from photo from camera from English (en-US) - to Italian (it) - in Italian (it) -
```

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 Thinkable](#). You only need to do this once.

Documentazione utile

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