thunkable

Giulio Galesi, ISTI-CNR

24-25 Ottobre 2022

Corso di Laurea in Informatica Umanistica
617AA: Tecnologie assistive per la didattica (aa 2022/23)
Università di Pisa

Cos’è Thunkable?

Thunkable è una piattaforma web per lo sviluppo di app senza codice.

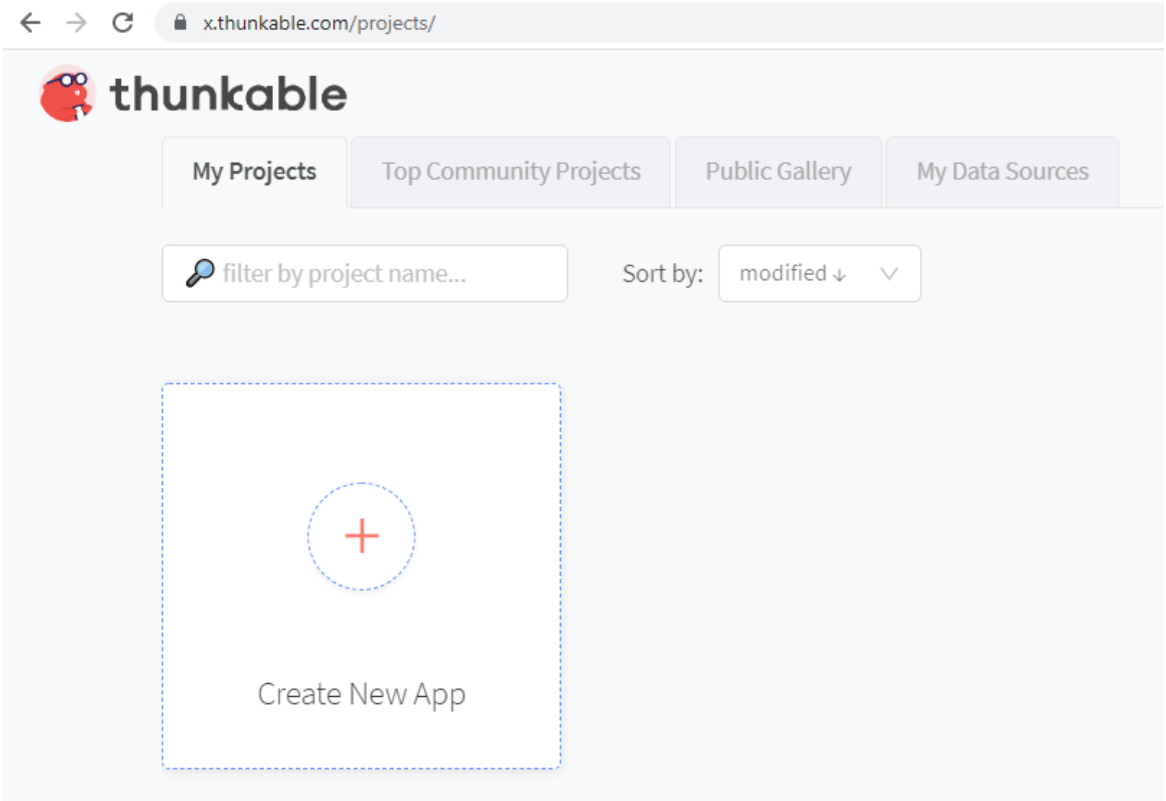
La versione gratuita consente a chiunque di progettare app, programmare potenti funzionalità con i blocchi e testare la propria app live sul proprio smartphone mentre con la versione Pro a pagamento (40€ mensili) è possibile caricare le app sul Google Play Store e sull'App Store di Apple.

E’ anche possibile creare web-app, ovvero app che non è necessario scaricare e a cui è possibile accedere direttamente online da browser.

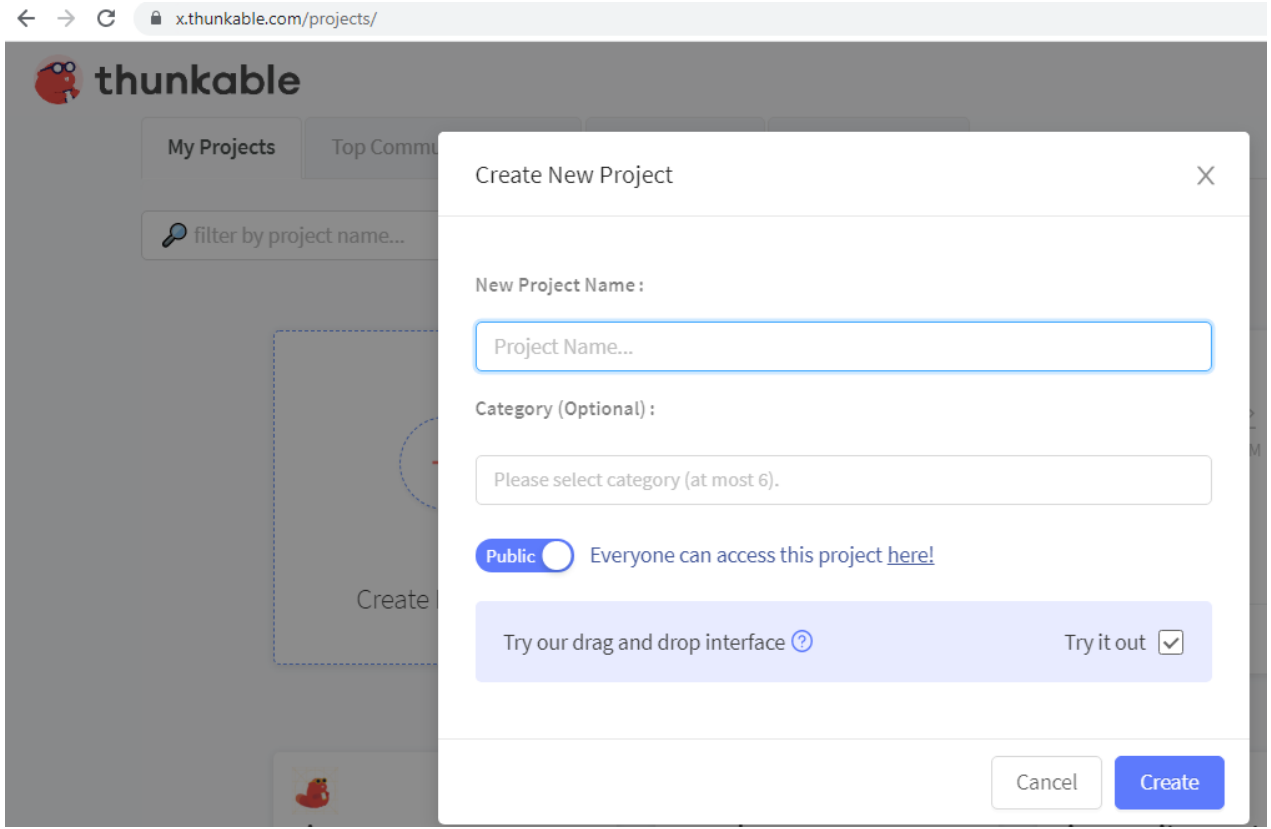
L’interfaccia è composta di una pagina di design in cui inserire le componenti dell’UI, e di una pagina di blocchi in stile puzzle per programmare il funzionamento della app.

Iniziamo ad usare Thunkable!

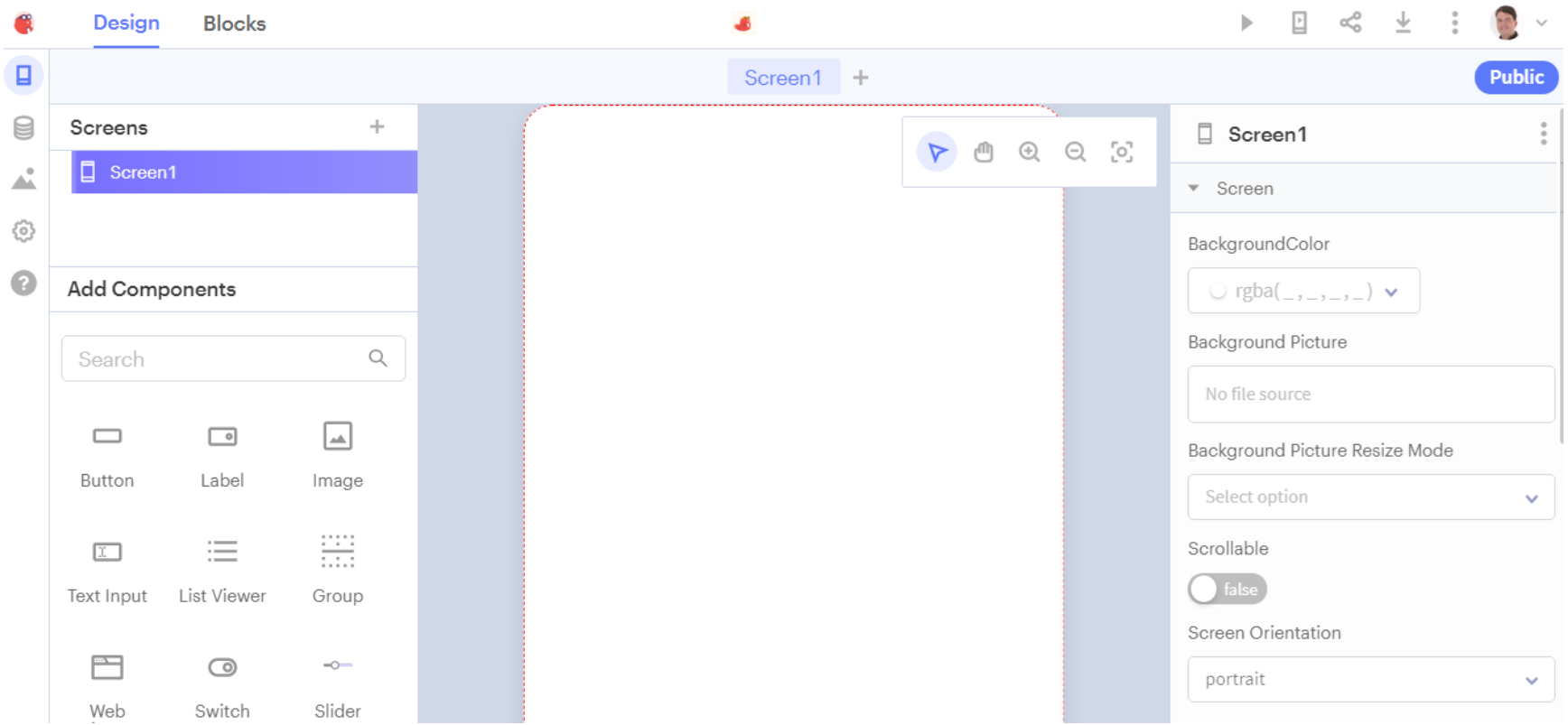
https://thunkable.com/



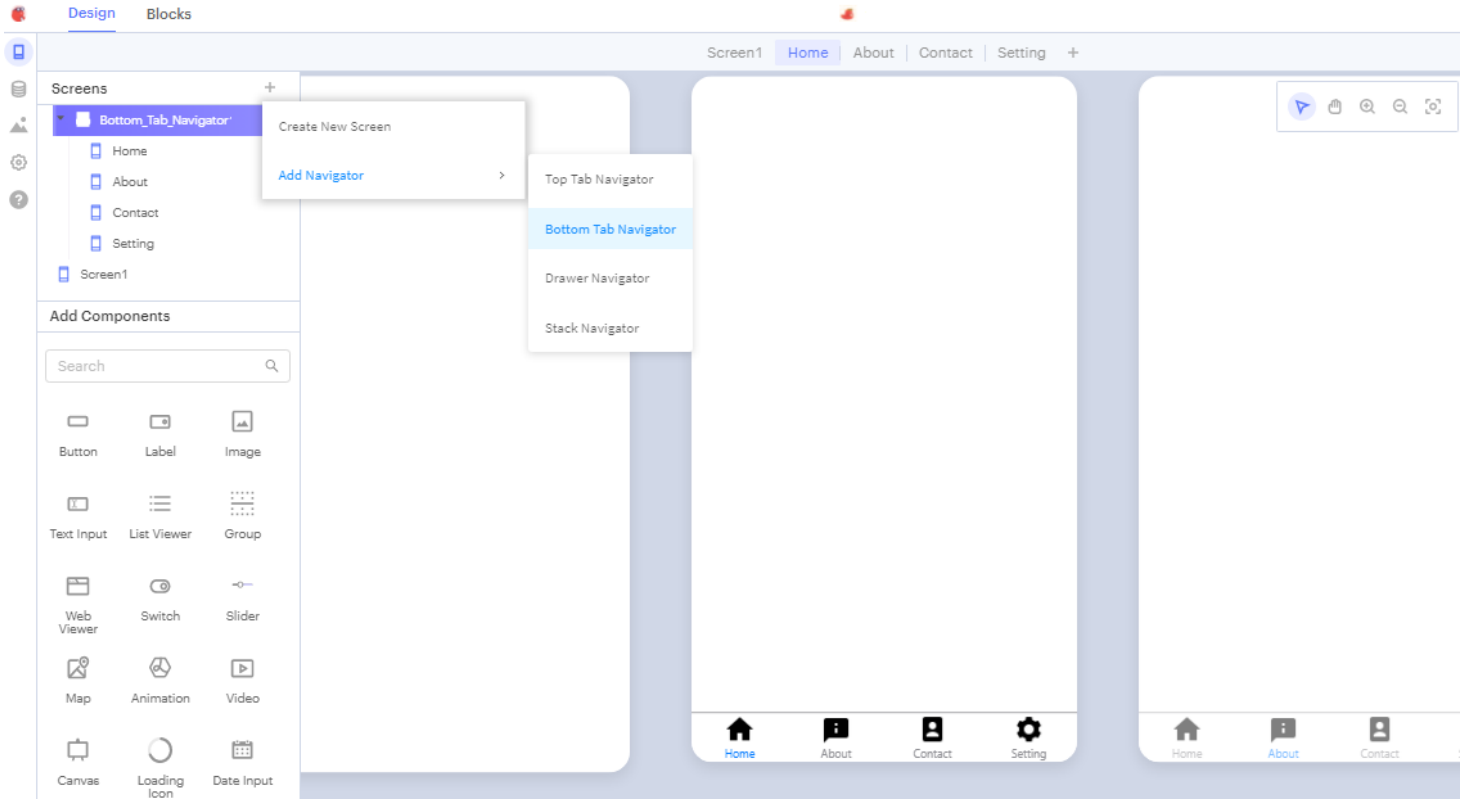
Creazione progetto usando l’interfaccia con drag and drop



Design Page - Screens

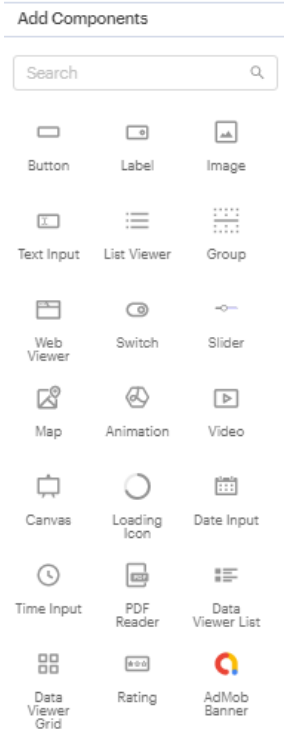


Design Page - Navigator



Design Page - Componenti

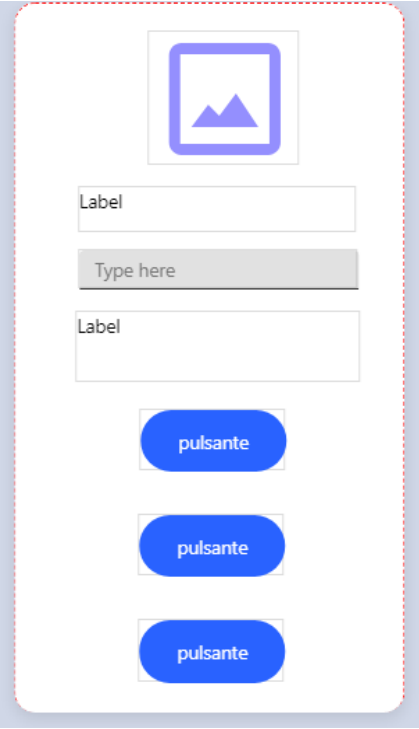
Si tratta delle componenti visibili dell'interfaccia utente:



- Pulsanti
- Etichette
- Immagini
- Testo
- Elenchi
- Gruppi
- Web Viewer
- Switch
- Slider

- Mappe
- Animazioni
- Video
- Canvas
- Loading Icon
- Date Input
- Time Input
- PDF Reader
- .. e molte altre

Proviamo a realizzare una interfaccia su Thunkable aggiungendo immagini, etichette e caselle di testo

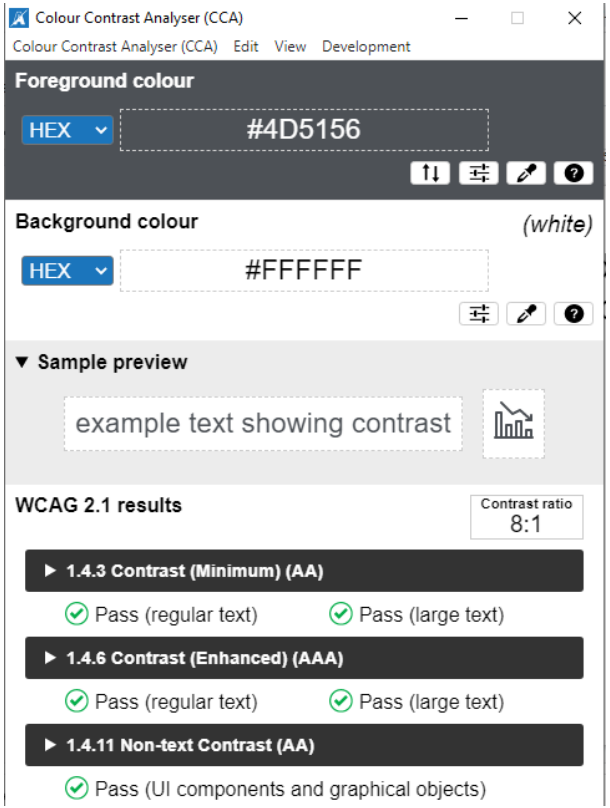


Accessibilità: in Thunkable manca la possibilità di inserire il testo alternativo per le immagini!

1) Se l'immagine che inseriamo è puramente decorativa andiamo a sovrapporre sopra l'immagine un label trasparente in cui inseriamo il testo alternativo che descrive l'immagine, utilizzando un font invisibile.
2) Se l'immagine deve poter essere cliccata, andiamo ad inserire un pulsante in cui impostiamo l'immagine come sfondo ed inseriamo come testo del pulsante la descrizione dell'immagine usando un font invisibile.

In questo modo la app creata sarà pienamente fruibile da tutti gli utenti.

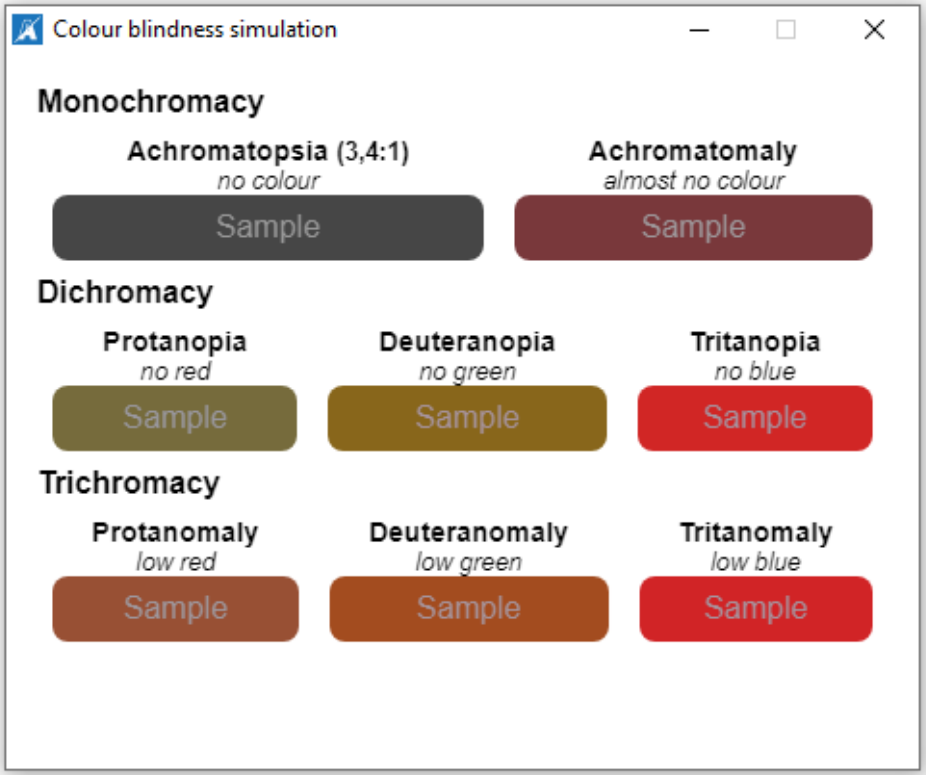
Verifica del contrasto con Colour Contrast Analyser

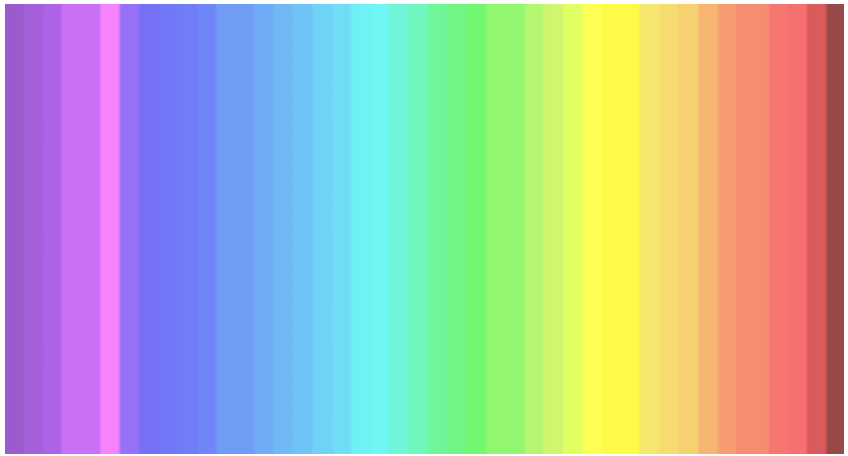


https://www.tpgi.com/color-contrast-checker/

- Uno strumento gratuito di controllo del contrasto cromatico che consente di determinare facilmente il rapporto di contrasto tra due colori permettendo di ottimizzare i contenuti, inclusi testo ed elementi visivi, per le persone con disabilità visive come daltonismo e ipovisione.
- Indicatori di conformità per le Linee guida per l'accessibilità dei contenuti Web 2.1 (WCAG 2.1)

Simulazione daltonismo con Colour Contrast Analyser





Se guardando l'immagine sopra (composta di 39 colonne con gradazioni di colore differente) una persona riesce a distinguere meno di 20 gradazioni di colore ha probabilmente un problema di daltonismo.

I blocchi associati alle componenti della UI

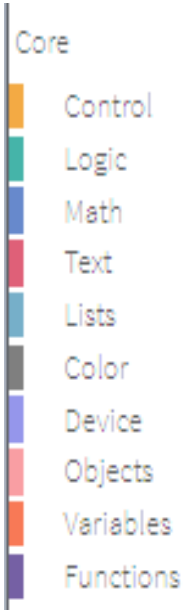
Dobbiamo considerare la app come una sequenza di eventi in cui sono coinvolte le componenti visibili dell'interfaccia utente nonchè le funzionalità non visibili ma che sono in esecuzione in background.
I blocchi relativi alle componenti inserite in una schermata non saranno usabili in altre schermate per cui bisogna utilizzare delle variabili per salvare le informazioni tra le schermate.

- Eventi

- Chiamate e Proprietà

I blocchi principali (core) per programmare una app

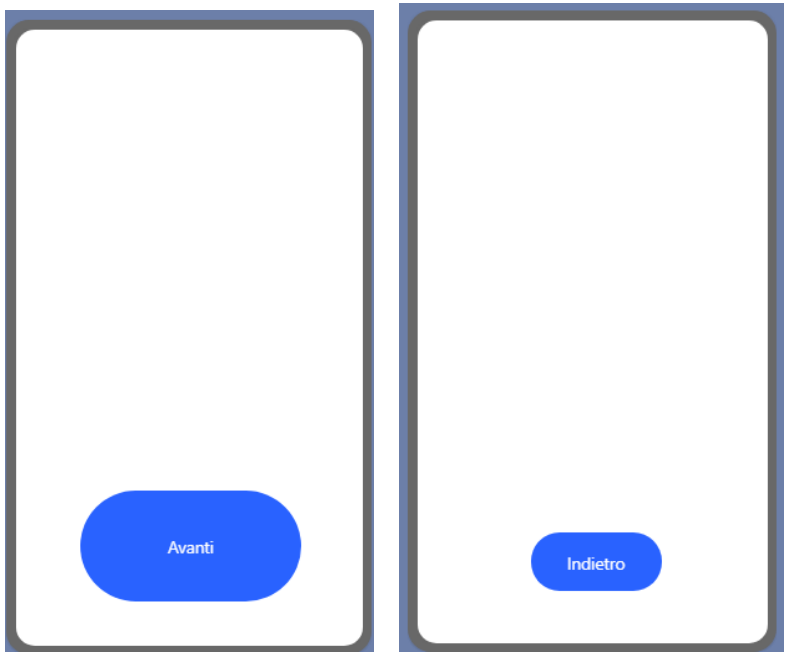
Abbiamo 10 categorie di blocchi principali che ci consentono di impostare regole e programmare la nostra app:



- Controllo (navigate, when->do, if->do, test, wait, loop,..)
- Logica (confronto, and-or, not, valori true,false,null)
- Matematica (operazioni con i numeri, valori random,..)
- Testo (impostare valori di testo e analisi testuali, join, contain, replace,..)
- Liste (creazione e modifica liste, insert, set,..)
- Colori (impostare colore da tavolozza o con valori rgb, hsv o esadecimali)
- Dispositivo (informazioni sul sistema operativo, dimensioni schermo,...)
- Oggetti (per gestire Web API che inviano dati in formato oggetto)
- Variabili (ove archiviare valori che poi verranno richiamati senza duplicarli)
- Funzioni

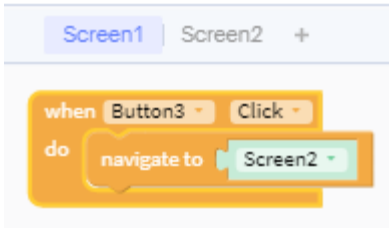
Blocco navigate

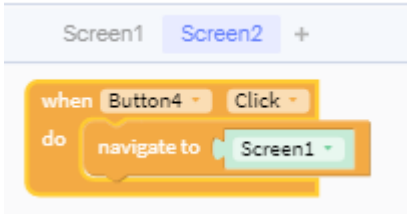
Consente di spostarci da una schermata all’altra; vediamo un esempio semplice creando due schermate ciascuna con un pulsante (schermata 1 con pulsante avanti e schermata 2 con pulsante indietro).
Realizziamo prima la UI e poi i blocchi.

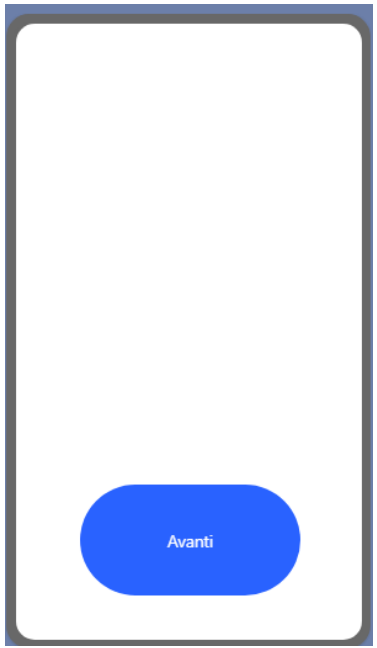


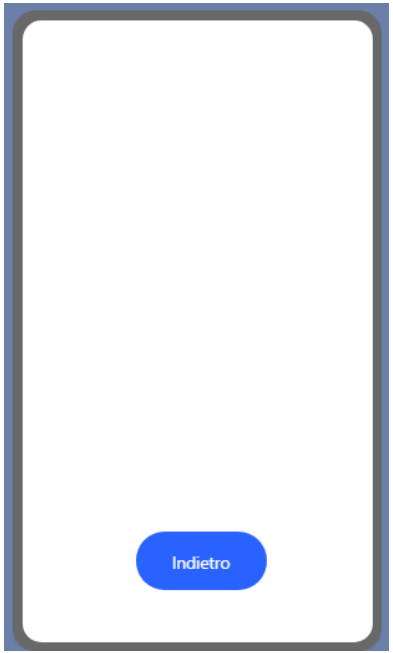
Blocco navigate

Cliccando sul primo pulsante veniamo portati nella seconda schermata

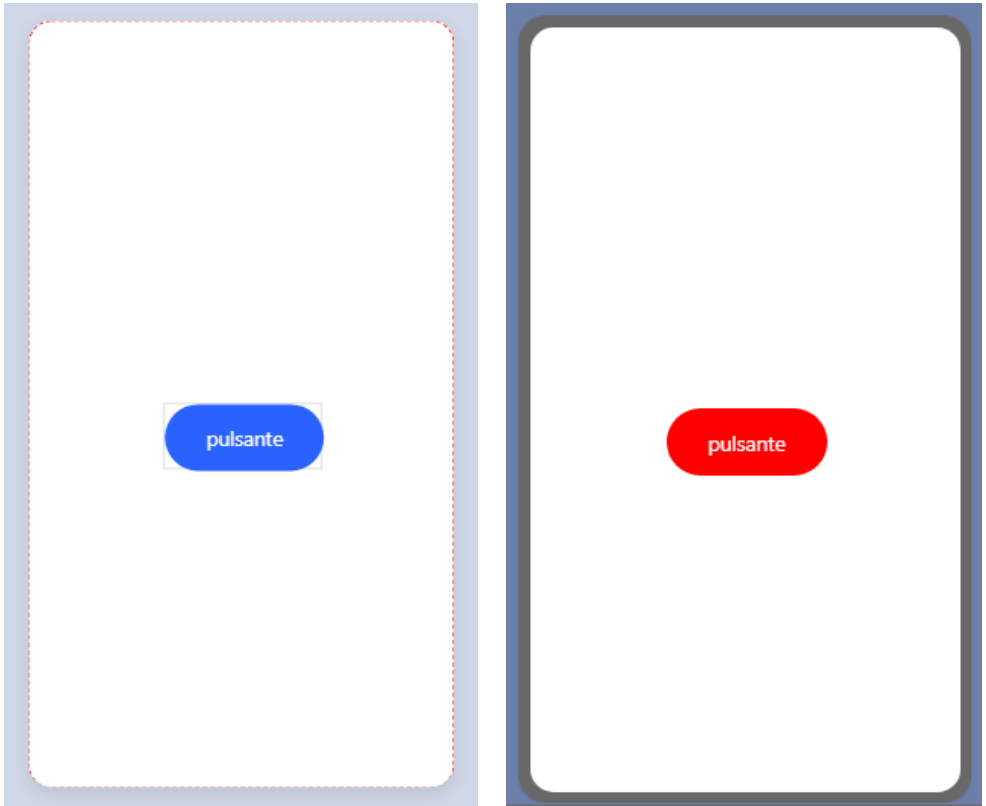




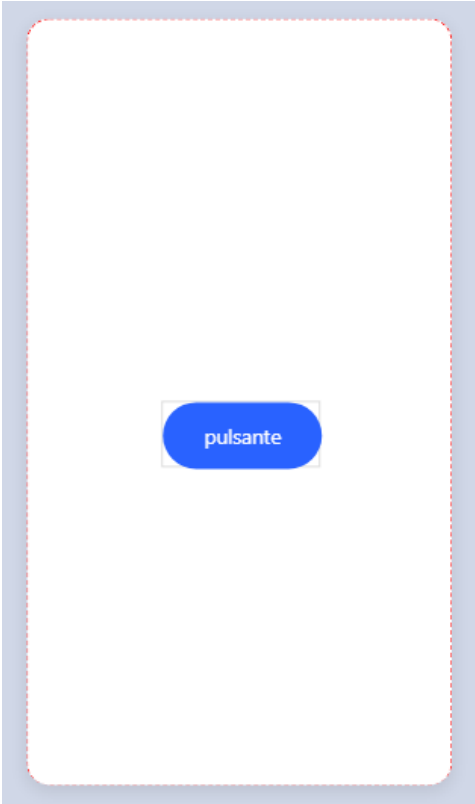


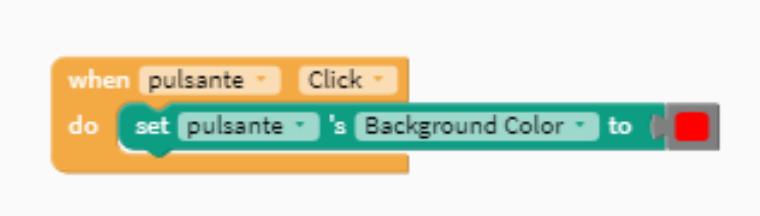


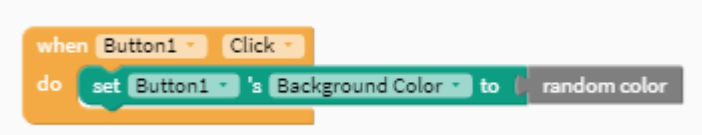
Creiamo un pulsante che quando viene cliccato cambia colore

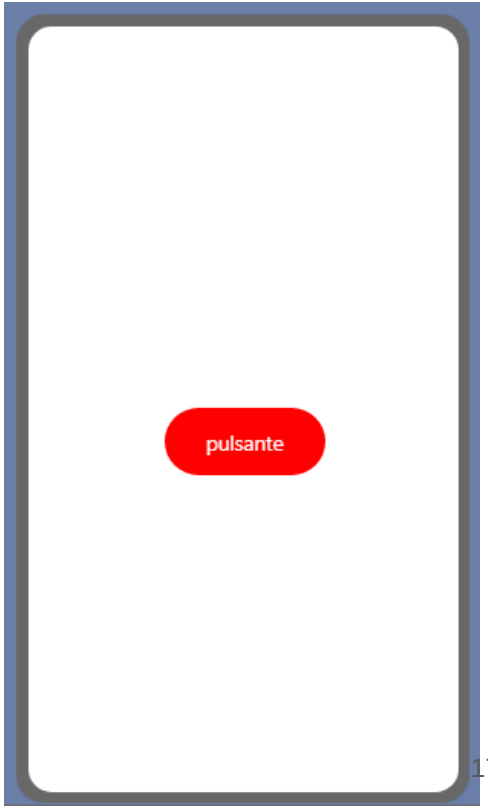


Creiamo un pulsante che quando viene cliccato cambia colore

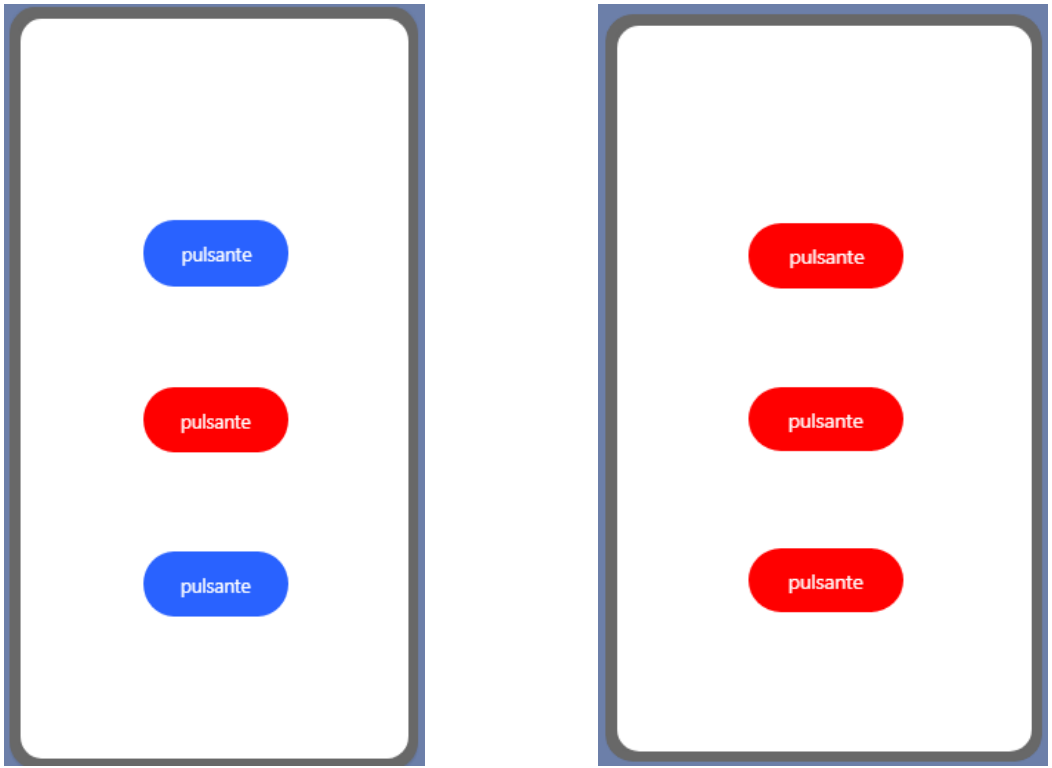








Aggiungiamo altri due pulsanti identici con la duplicazione



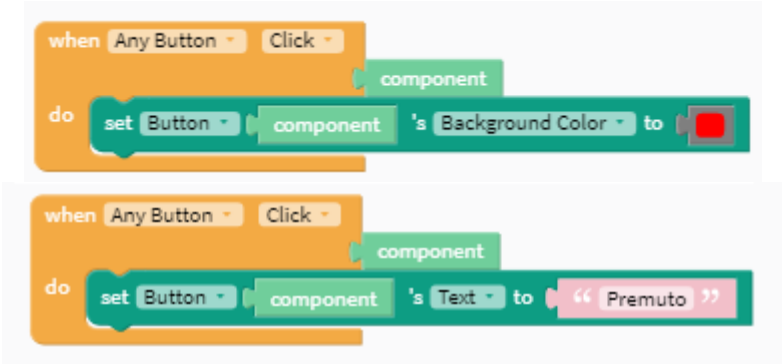
Aggiungiamo altri due pulsanti identici con la duplicazione

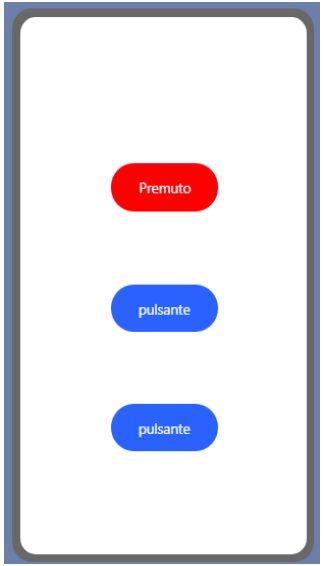
Usando la duplicazione delle componenti della UI, Thunkable ci avvisa che i relativi blocchi funzione non verranno duplicati quindi solo un pulsante cambierà colore una volta cliccato!

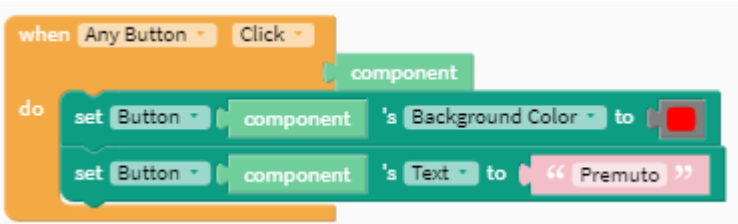
Cambiamo il testo del pulsante una volta premuto

Cambiamo il testo del pulsante una volta premuto e sfruttiamo il blocco avanzato “Any component”

Possiamo scegliere di modificare il testo del pulsante oltre che lo sfondo, andando a duplicare i blocchi e modificando il Text anzichè il Background Color.
Prestiamo attenzione a semplificare i blocchi quando possible, usando il blocco avanzato “Any component”.

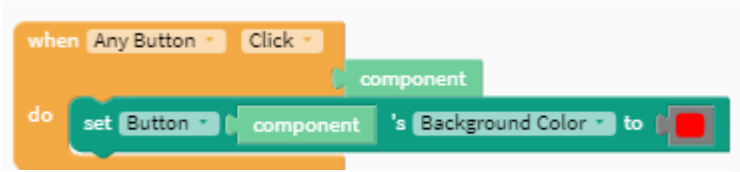


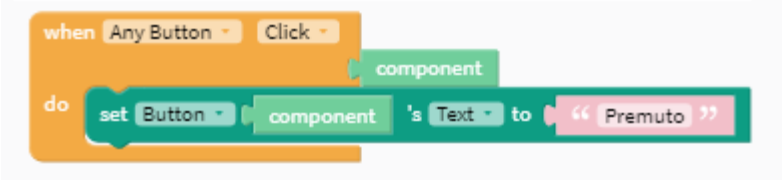


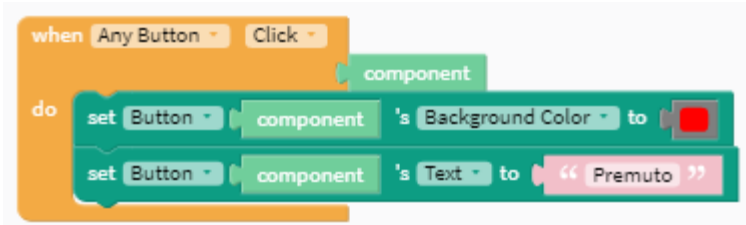


Usiamo il blocco avanzato “Any component”

Possiamo scegliere di modificare il testo del pulsante oltre che lo sfondo, andando a duplicare i blocchi e modificare la proprietà Text anzichè Background Color. Prestiamo attenzione a semplificare i blocchi quando possibile.

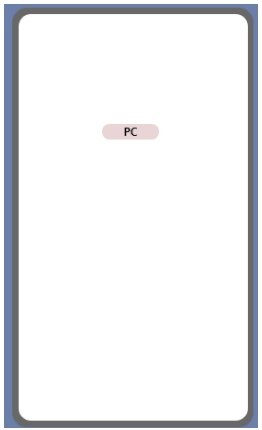


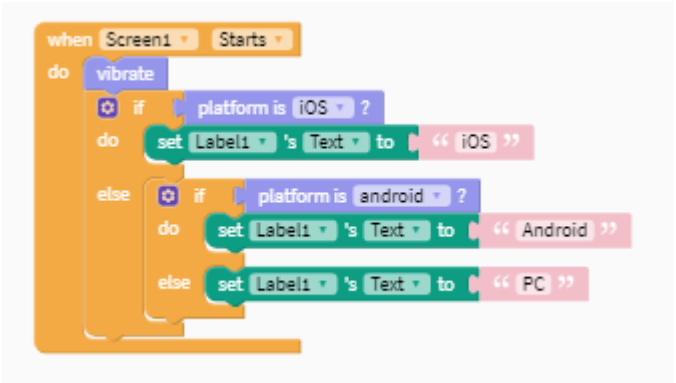


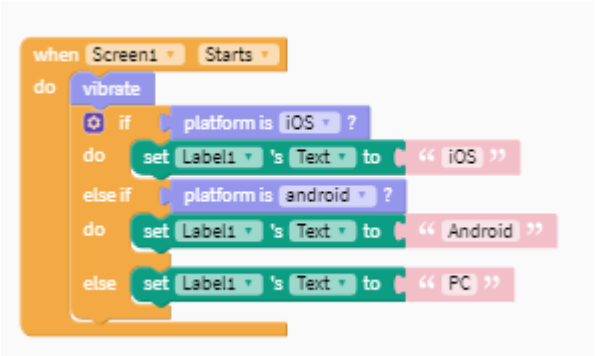


Blocchi if-do-else ed else-if

Il blocco if valuta la condizione e in base alla risposta (vero o falso) esegue l'istruzione associata a "do", altrimenti (else) esegue l'istruzione associata al blocco successivo. E' anche possibile annidare uno o più blocchi if all'interno di un blocco if o usare il blocco "else if". Creaiamo una app che emetta una vibrazione quando si avvia e che ci indichi semplicemente quale piattaforma stiamo utilizzando.

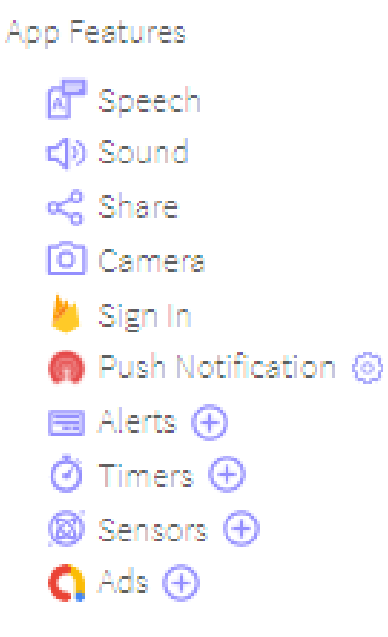






Le funzionalità (App features)

Abbiamo 10 categorie di funzionalità che ci consentono di usare blocchi con caratteristiche differenti:



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

Realizziamo una semplice app che riproduca un audio

Dalla pagina di design inseriamo solo un pulsante che, se cliccato, dovrà riprodurre un messaggio vocale tipo “Ho sete”.

Realizziamo una semplice app che riproduca un audio

Thunkable ha un suo modulo di Text to Speech (TTS)

Funziona correttamente ma notiamo che la qualità della pronuncia non è delle migliori.

Realizziamo una semplice app che riproduca un audio

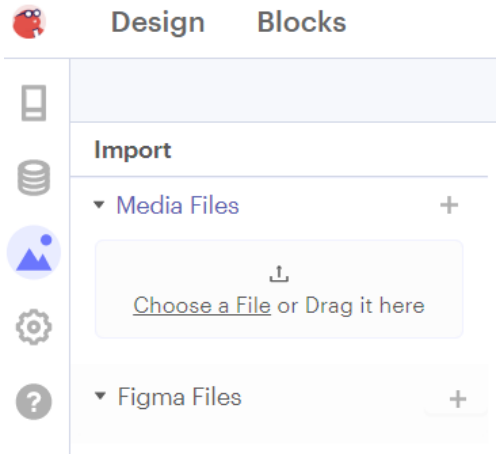
Free Text-To-Speech (TTS)

ttsmp3.com (https://ttsmp3.com/) consente di scaricare i file mp3 parlati dei testi che andiamo a digitare nella casella (fino a 3000 caratteri per volta).

Una volta digitato il testo da trasformare in parlato e scelta la lingua/voce da utilizzare, basterà premere il tasto ”Download as MP3” per ottenere il file.

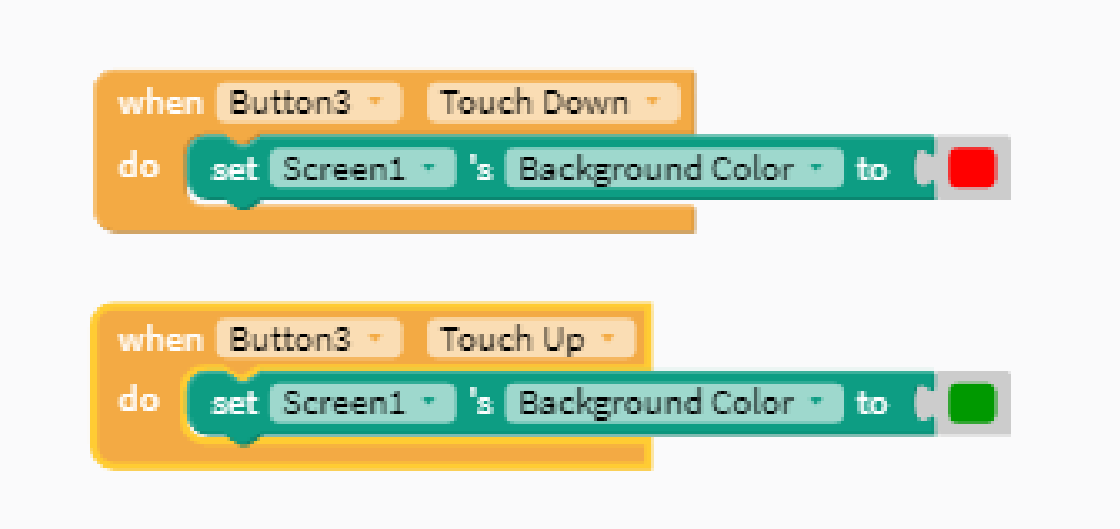
Realizziamo una semplice app che riproduca un audio

Andiamo a caricare il file audio in formato mp3 in Assets - Media Files e poi componiamo i blocchi per far funzionare la app in pochi secondi.



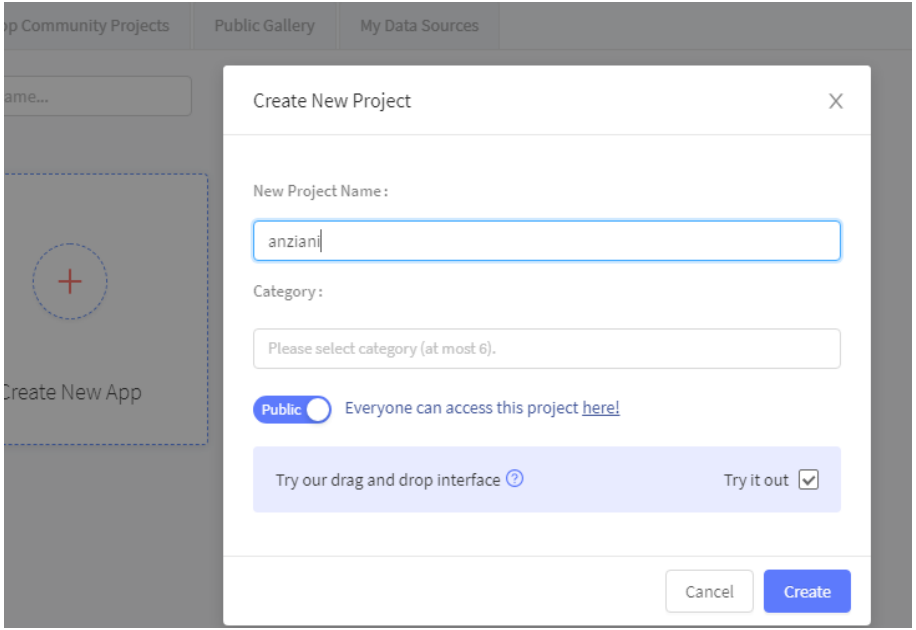
Usiamo le caratteristiche touch down e touch up del pulsante

Possiamo impostare un pulsante perchè effettui il cambio del colore di sfondo della schermata (o ad altro elemento della UI) solo mentre è premuto.



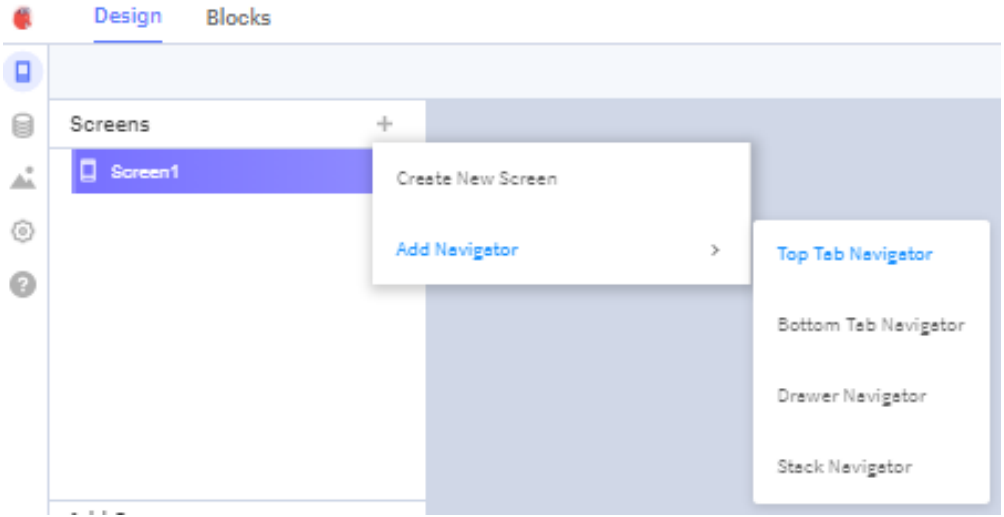
Creiamo una app utile per persone anziane che hanno difficoltà di comunicazione

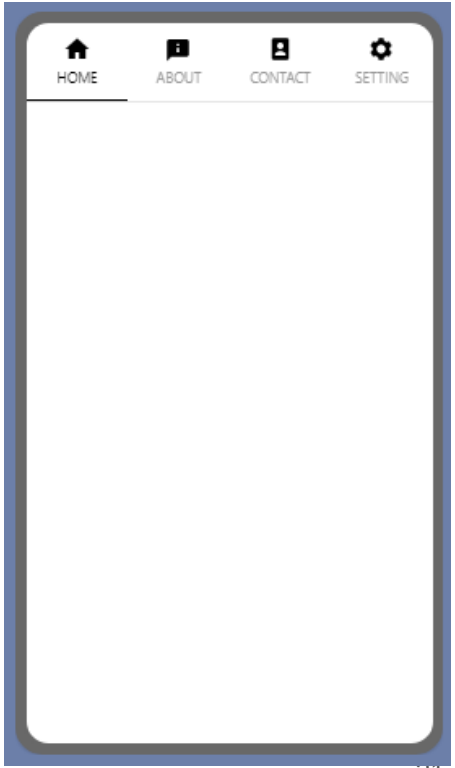
Creiamo un nuovo progetto



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.

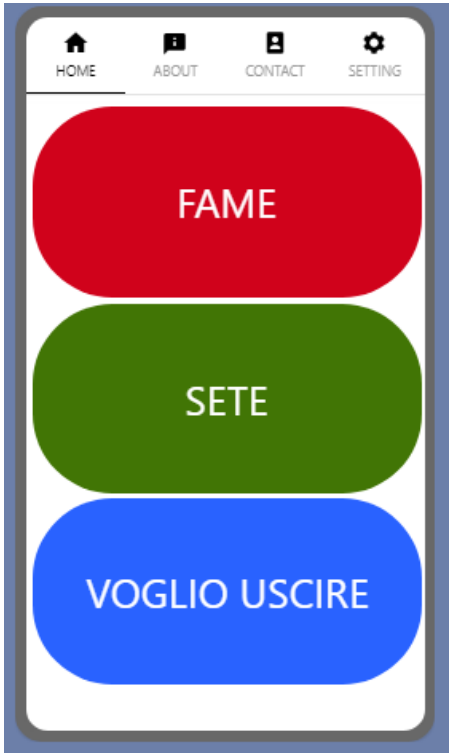




App per anziani

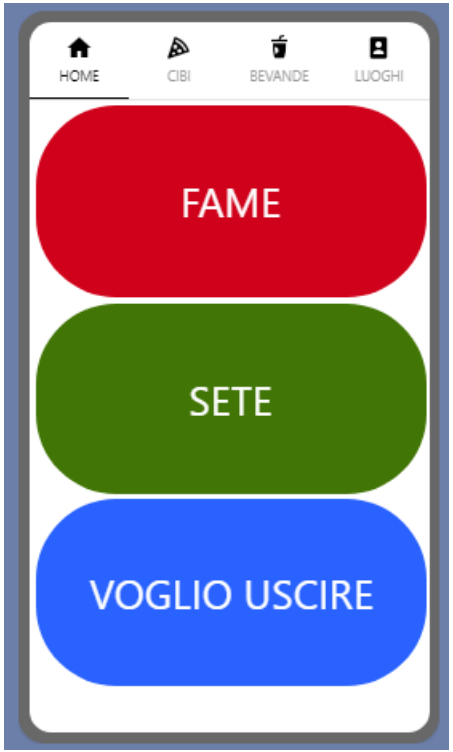
Aggiungiamo in Home tre pulsanti (inseriamo prima un button e lo duplichiamo due volte).

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



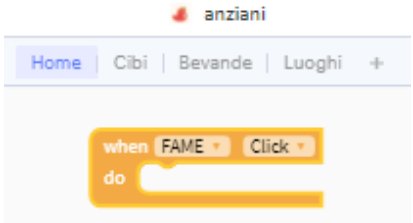
App per anziani

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



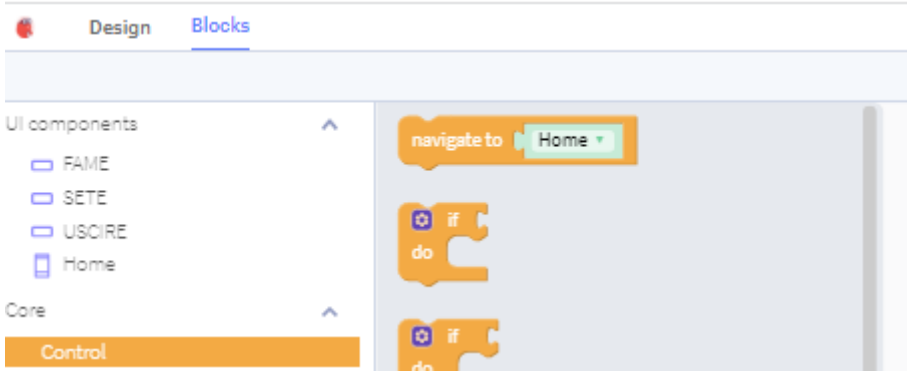
App per anziani

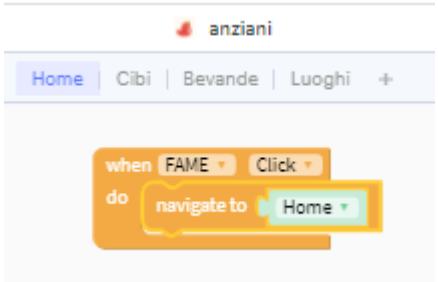
Passiamo a comporre i blocchi della nostra app. Selezioniamo il primo pulsante, scegliamo il blocco evento When-click e trasciniamolo nel desk.



App per anziani

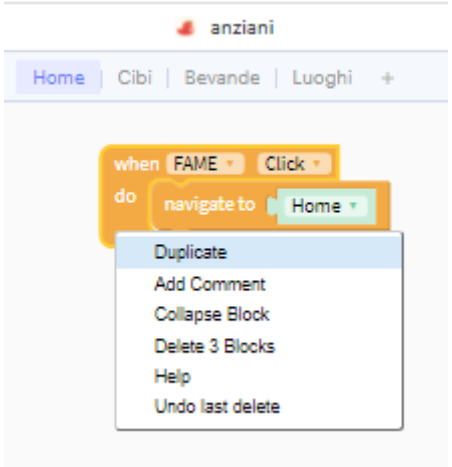
Scegliamo il blocco “navigate to” dalla sezione dei Core blocks - Control e trasciniamolo all’interno del nostro blocco evento.

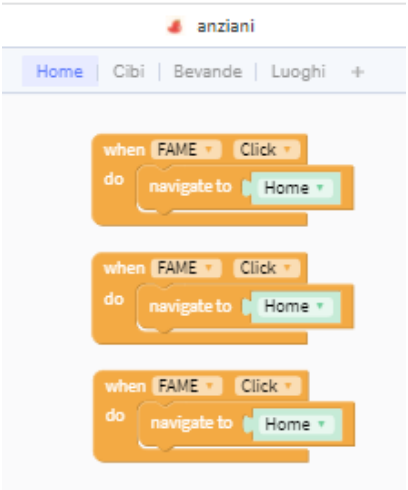


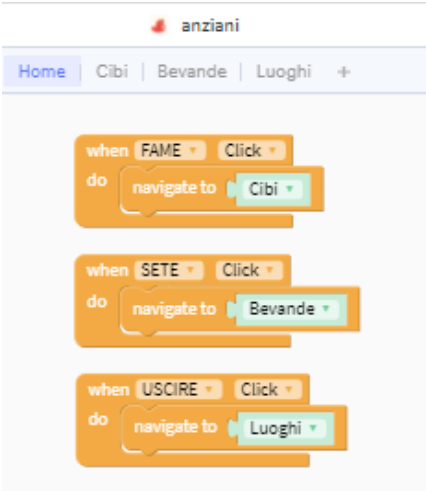


App per anziani

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.

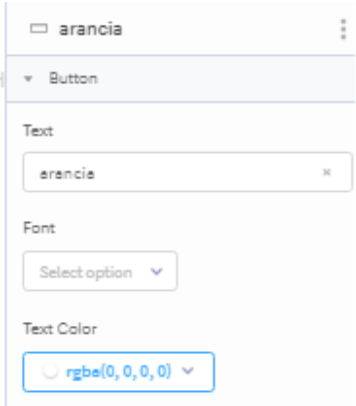


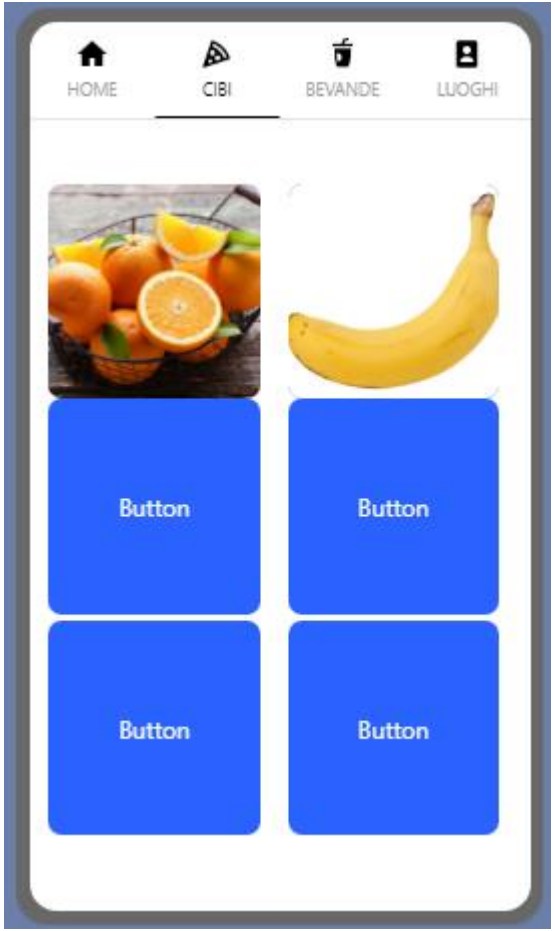




App per anziani

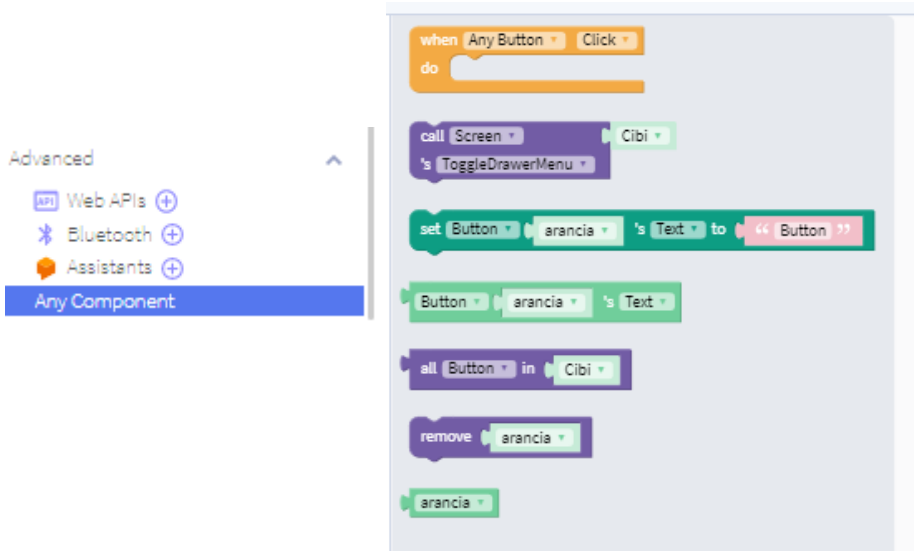
Torniamo sul design e nella schermata cibi andiamo ad inserire dei pulsanti a cui associamo i nomi di alimenti e sostituiamo lo sfondo con delle immagini associate al nome (ricordiamoci di impostare il colore del testo trasparente).

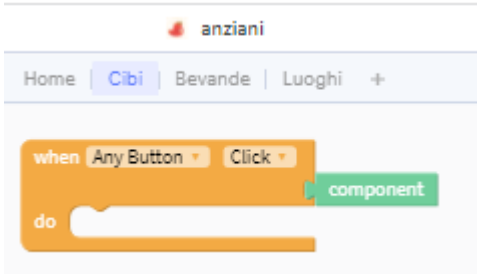




App per anziani

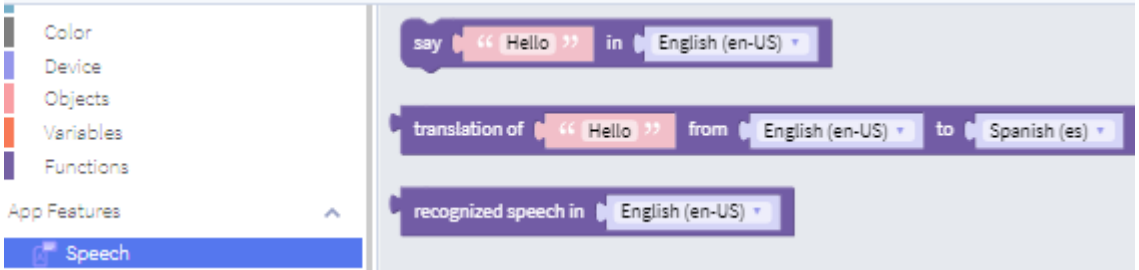
Spostiamoci sul desk della schermata Cibi, selezioniamo la categoria avanzata di blocchi “Any Component” e trasciniamo nel desk il blocco “when Any Button Click”.

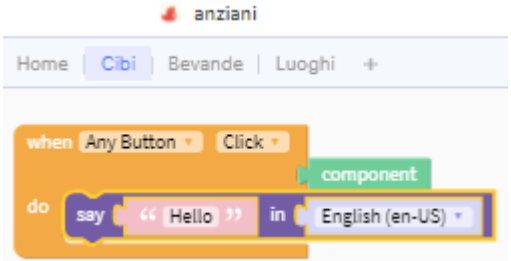




App per anziani

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



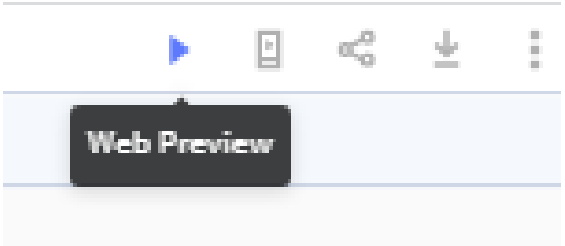


App per anziani

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

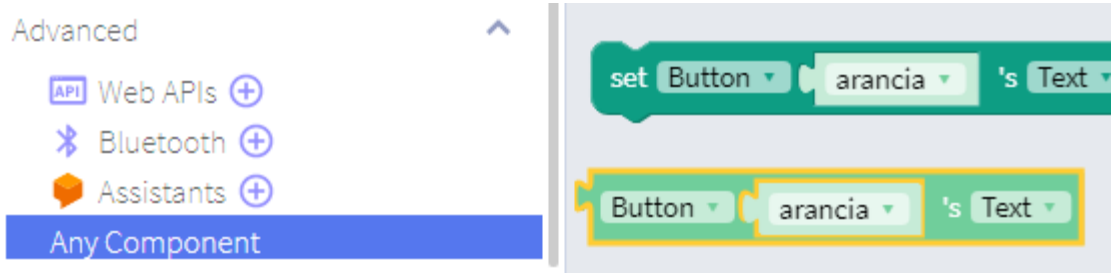
Notiamo che qualsiasi pulsante noi premiamo, viene riprodotto il saluto Hello!

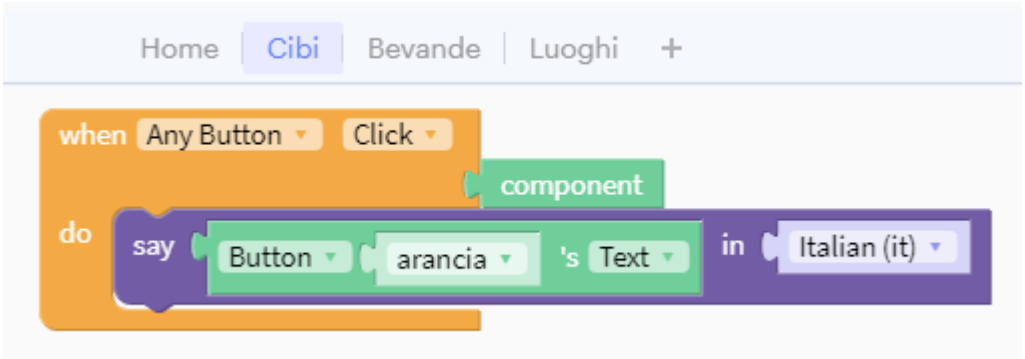
Dobbiamo ora modificare i blocchi per permettere ai pulsanti, una volta premuti, di riprodurre il messaggio associato al contenuto che veicolano.



App per anziani

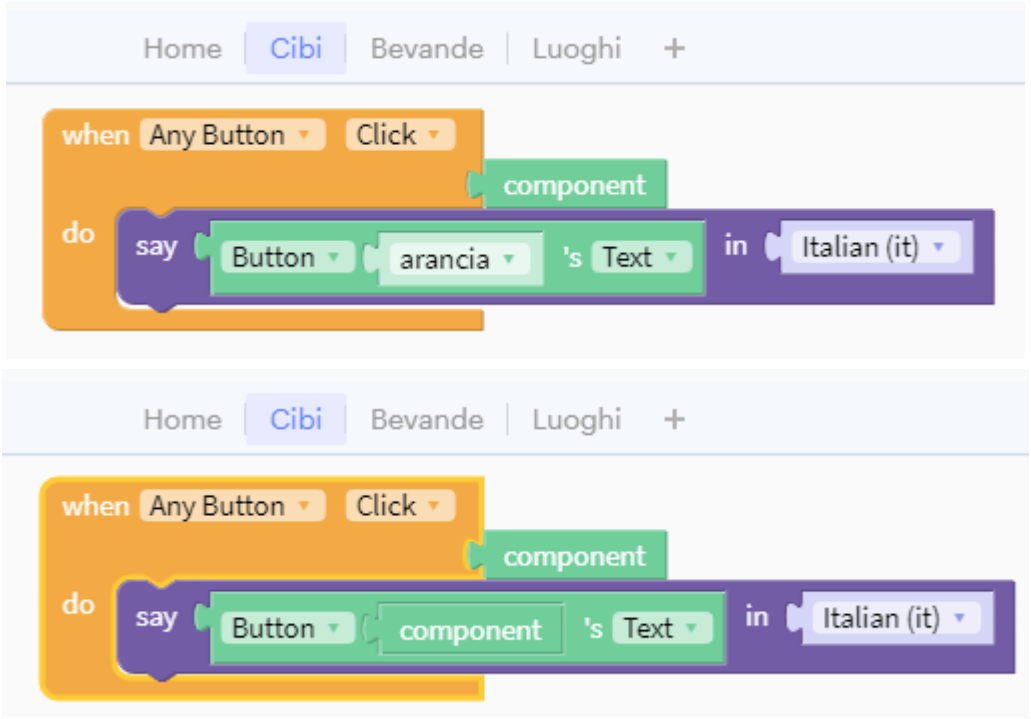
Da Any component andiamo a scegliere “Button xxx‘s text” e trasciniamo questo blocco all’interno del blocco “say”.





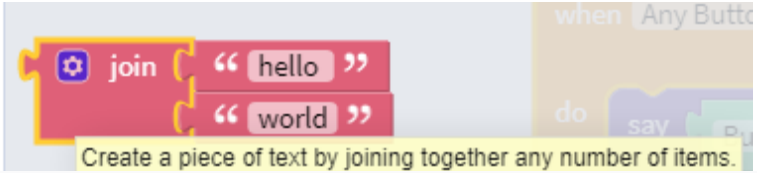
App per anziani

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



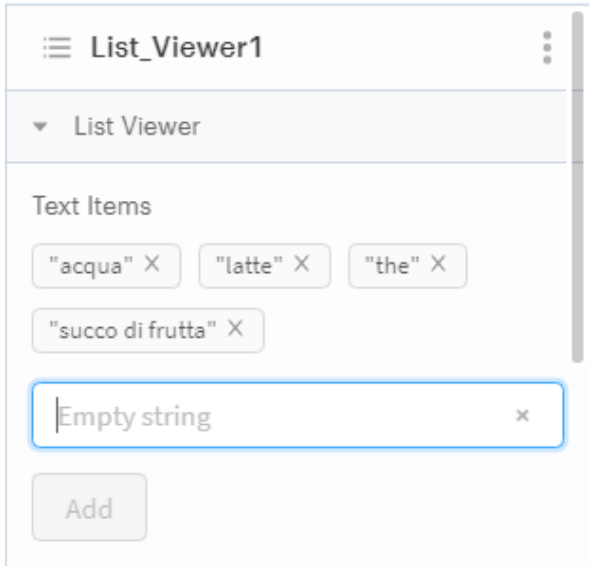
App per anziani

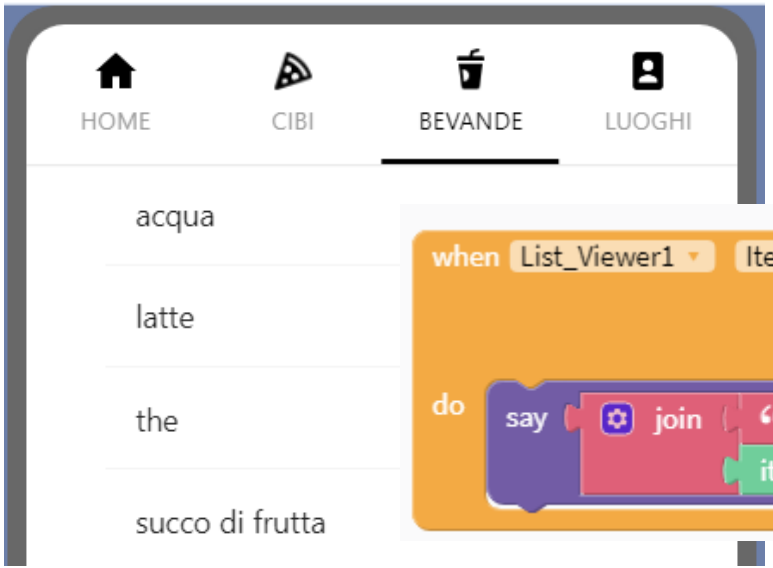
Se eseguiamo la app, adesso ciascun pulsante riproduce il testo associato. Utilizzando i blocchi con funzioni testuali possiamo migliorare la frase che viene letta; ad esempio, usando il blocco join, il sistema pronuncerà “voglio una arancia” anzichè dire solo “arancia”. Il testo statico impostato nella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



App per anziani

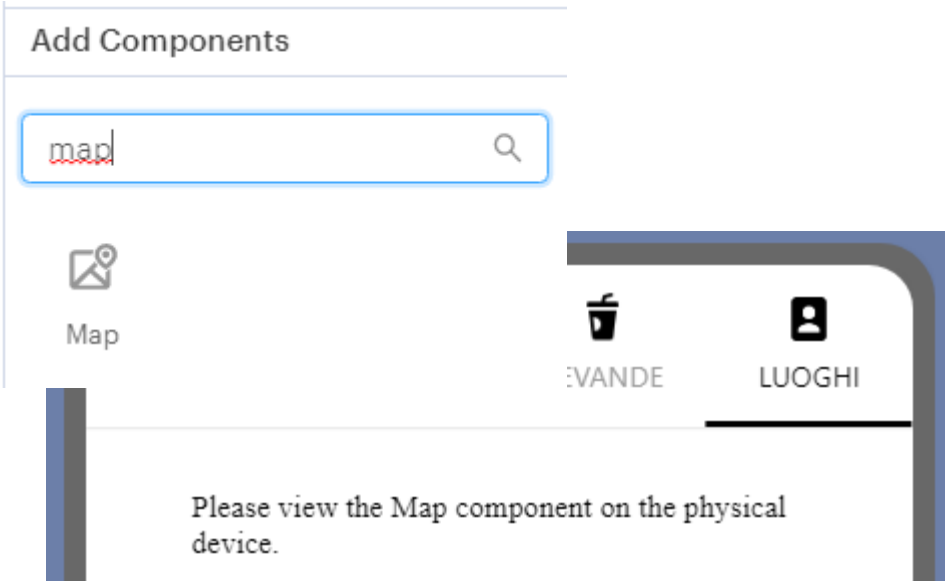
Possiamo fare altrettanto con la pagina delle bevande: anzichè usare i pulsanti con le immagini inseriamo un semplice elenco di elementi tra cui scegliere.

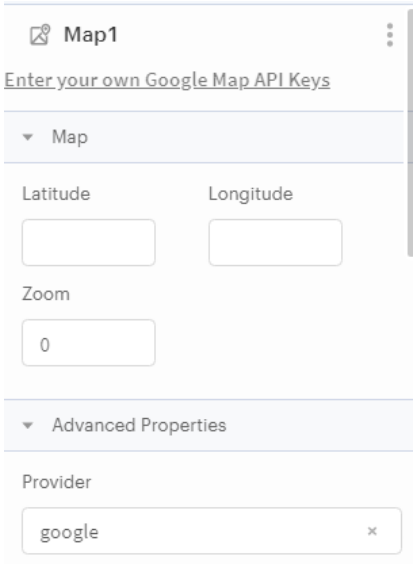


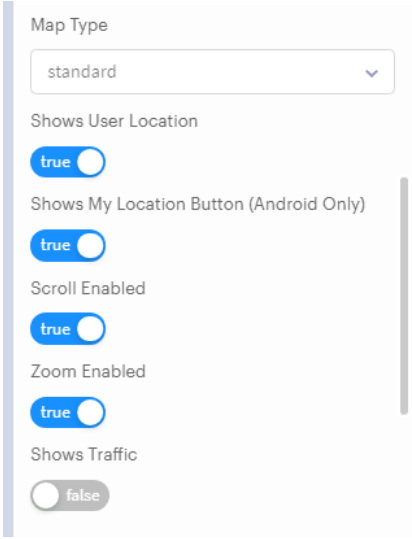


App per anziani - Luoghi

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

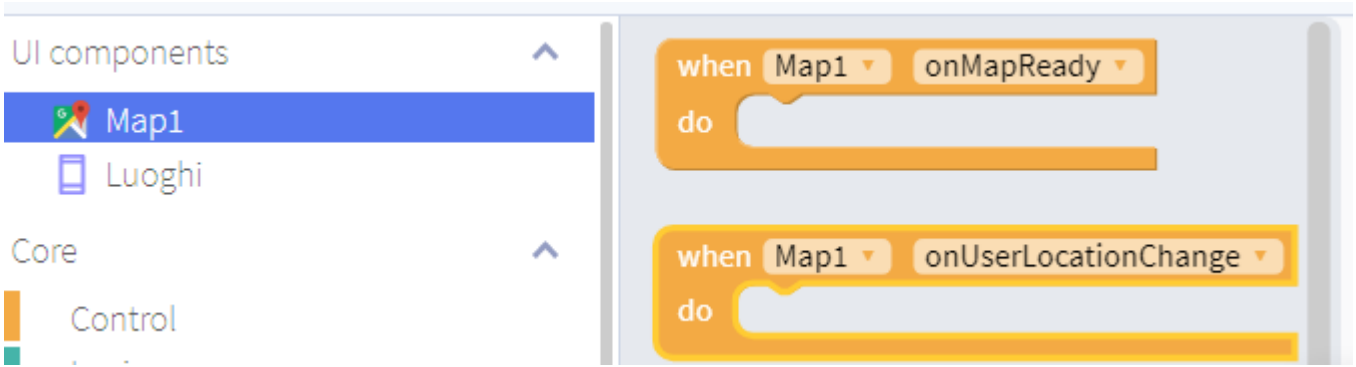






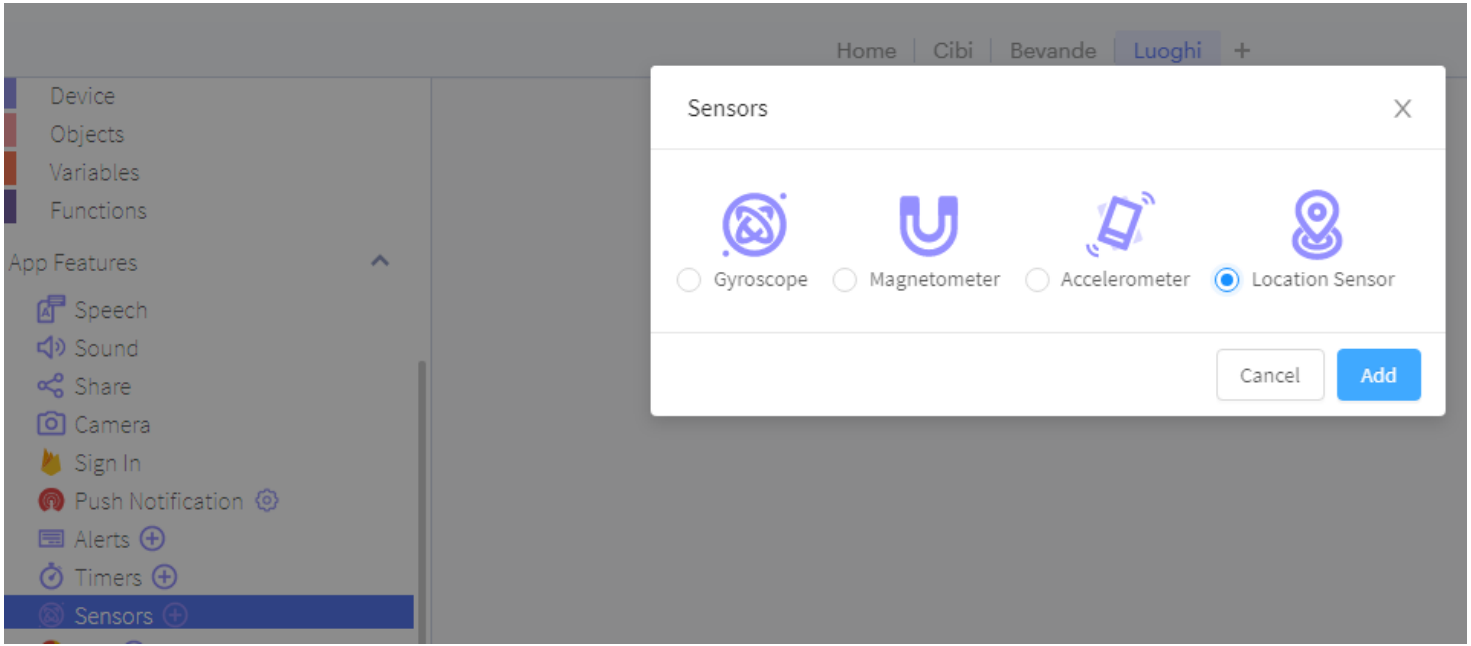
Blocco Mappa

Andiamo a trascinare nel desk un blocco evento associato alla mappa



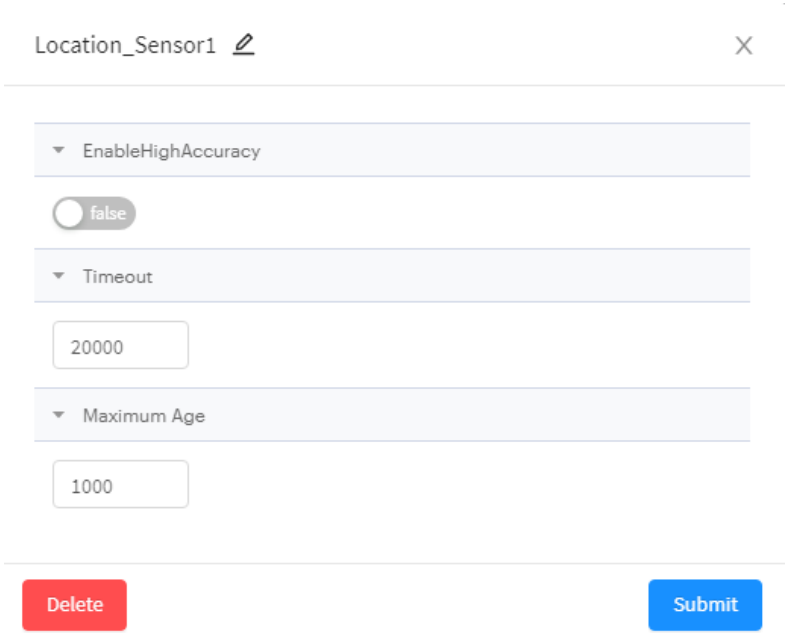
Sensore di posizione

Inizializziamo il location sensor (gps del dispositivo)



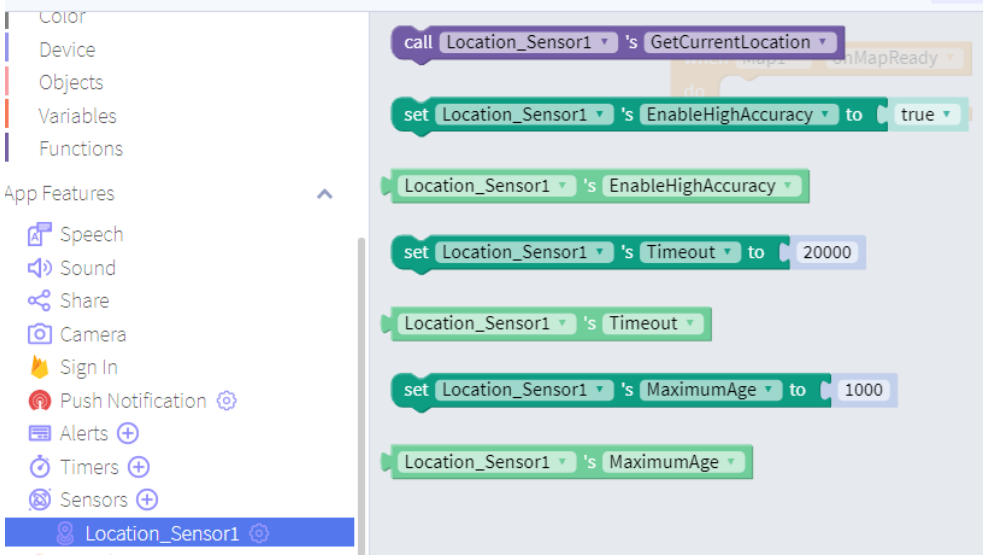
Sensore di posizione - parametri

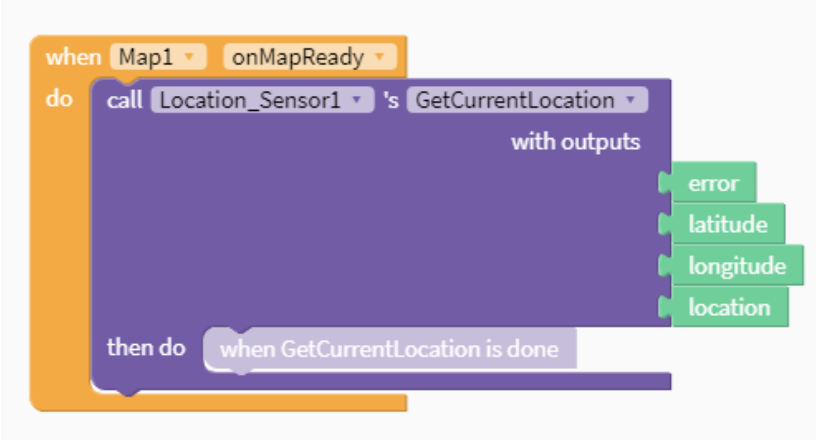
Impostiamolo con i parametri di default



Chiamata del sensore di posizione nel blocco mappa

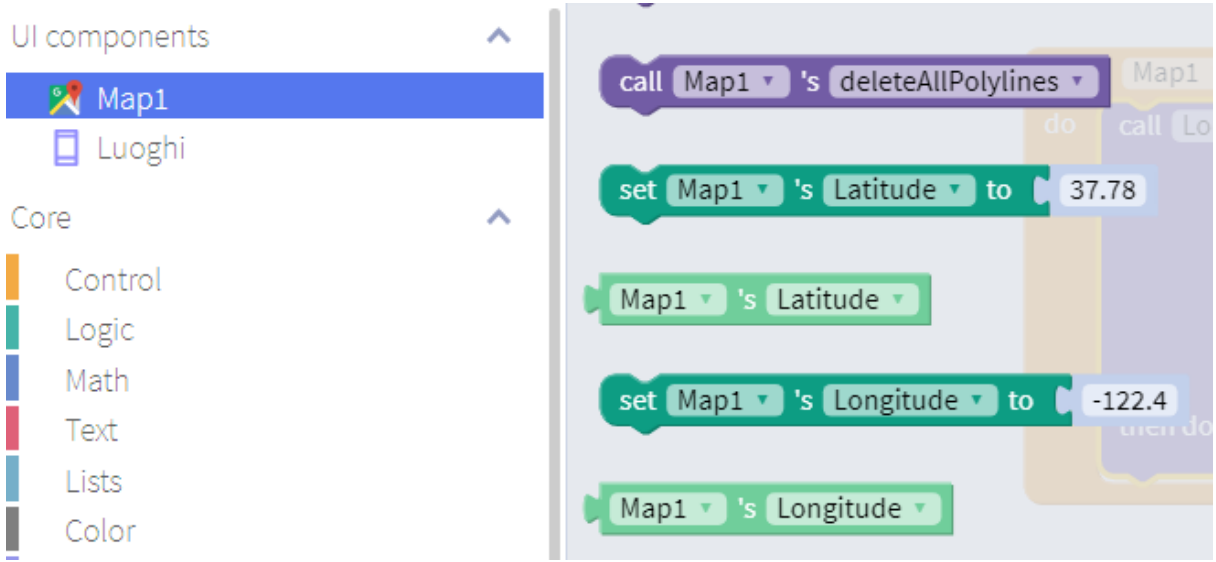
Ora trasciniamo la chiamata del sensore di posizione all'interno del blocco evento. Il sensore di posizione fornisce 4 output: errore, latitudine, longitudine e posizione.





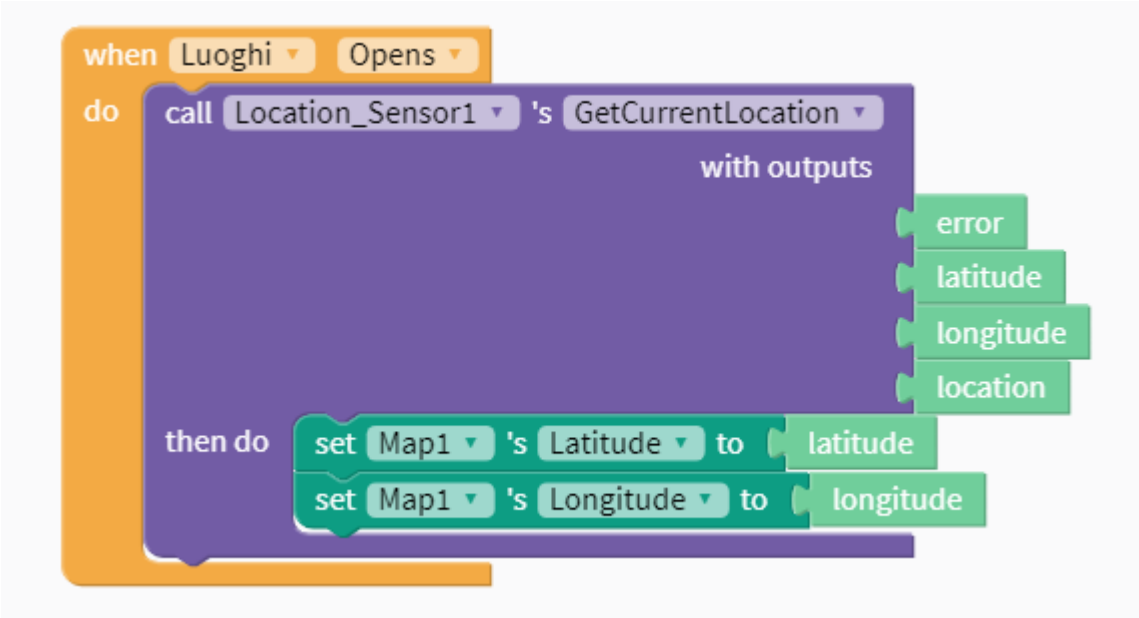
Sensore di posizione

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.

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

Inseriamo una immagine, una etichetta e un pulsante.

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

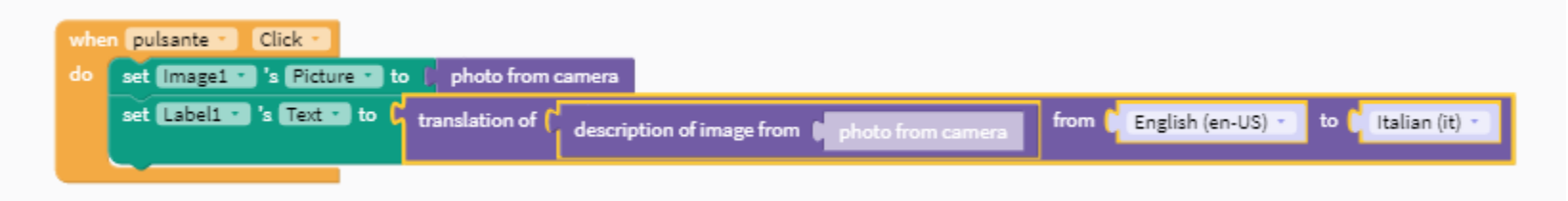
Nei blocchi usiamo il modulo fotocamera e il riconoscitore (Microsoft Image Recognition)

Traduttore

Modifichiamo la app perchè scriva la didascalia in italiano anzichè in inglese

Traduttore

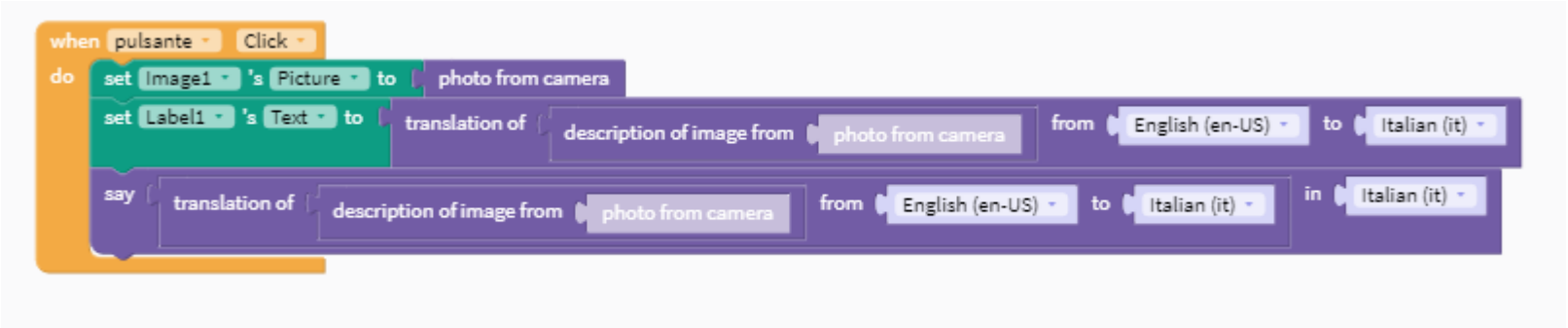
aggiungiamo il modulo traduttore da inglese a italiano



Traduttore

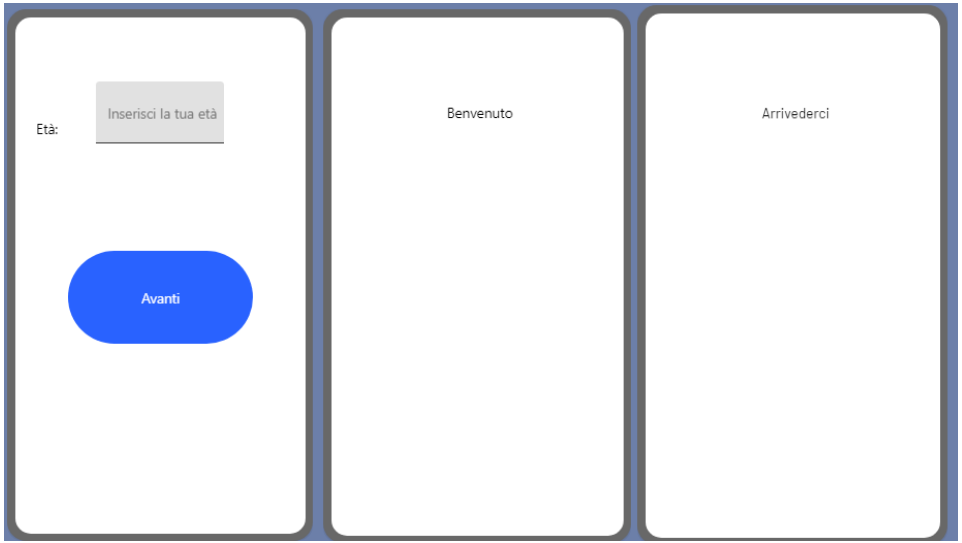
attiviamo il modulo TTS cosicchè la app legga ad alta voce la didascalia.

Traduttore

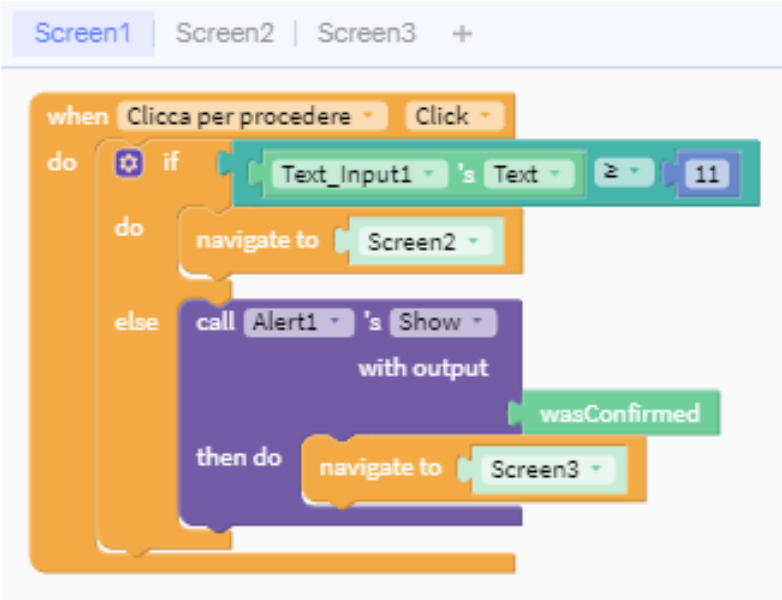

when pulsante Click
do
set Image1 's Picture to photo from camera
set Label1 's Text to translation of description of image from photo from camera from English (en-US) to Italian (it)
say translation of description of image from photo from camera from English (en-US) to Italian (it) in Italian (it)

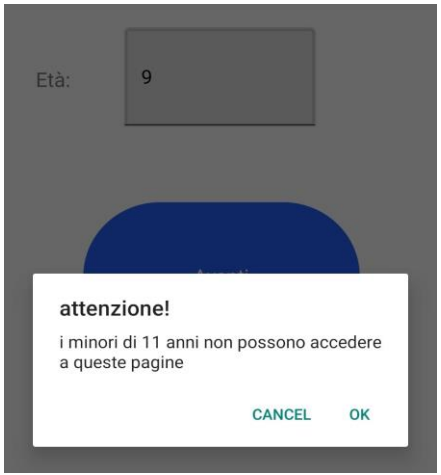
Esempio di app con alert

Proviamo a creare una app che non consenta a chi ha meno di 11 anni di usare la app. Possiamo chiedere l'età nella prima schermata e se questa è maggiore o uguale a 11 anni il sistema consentirà di passare alla schermata successiva, altrimenti aprirà un alert che avvisa della limitazione attiva.



Esempio di app con alert



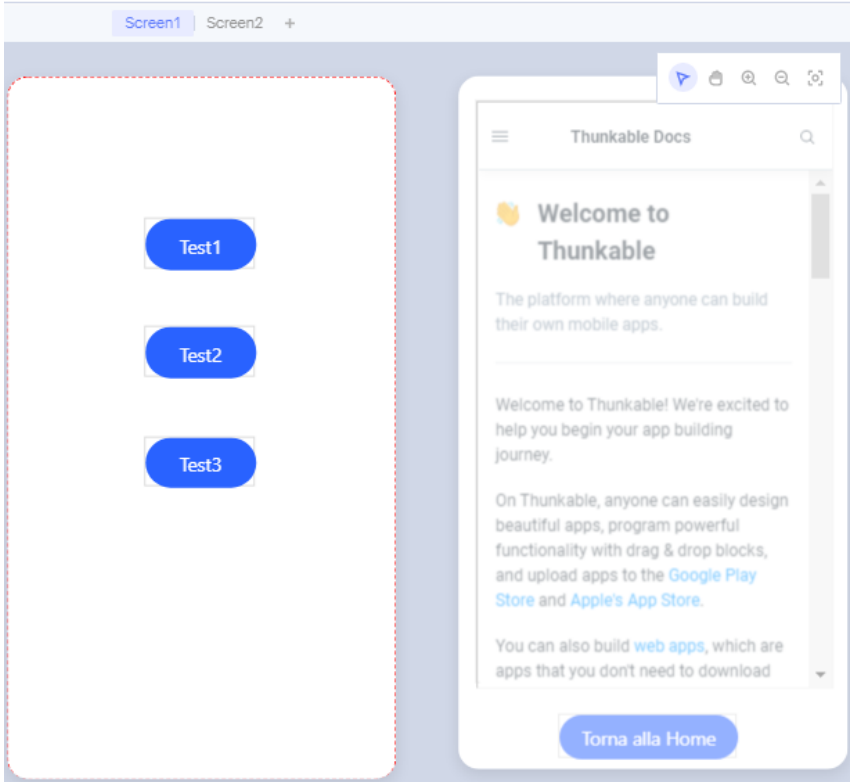


Utilizziamo le variabili

Esempio di una app che ci consente di visualizzare siti web con url predefiniti

Nella pagina di design:

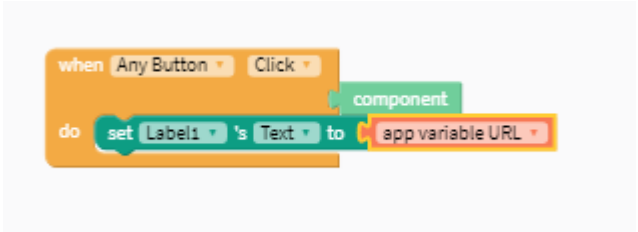
1. creiamo due schermate
2. Nella prima inseriamo tre pulsanti
3. Nella seconda inseriamo un pulsante e un componente web viewer

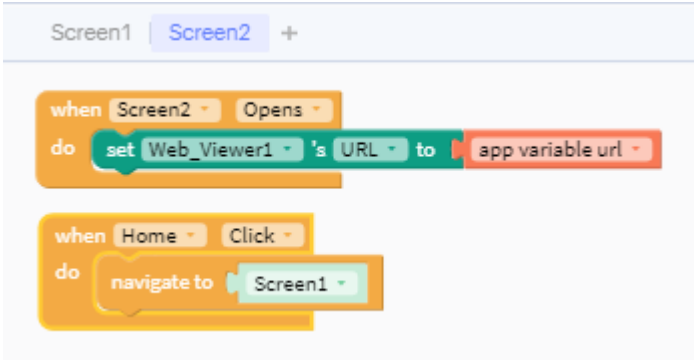


Variabili in Thunkable

Nella pagina dei blocchi di Screen1 inizializziamo la variabile url e attiviamo i blocchi eventi dei tre pulsanti associando un url diverso a ciascun pulsante.

Nella pagina dei blocchi di Screen2 attiviamo il blocco eventi del visualizzatore web associando al suo attributo URL la variabile url.



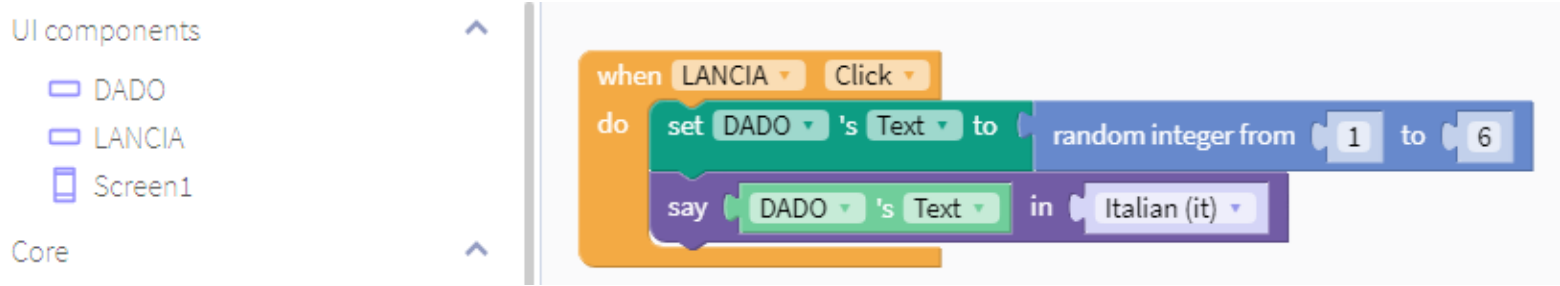


App dado parlante

Creaiamo una app che cliccando su 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.

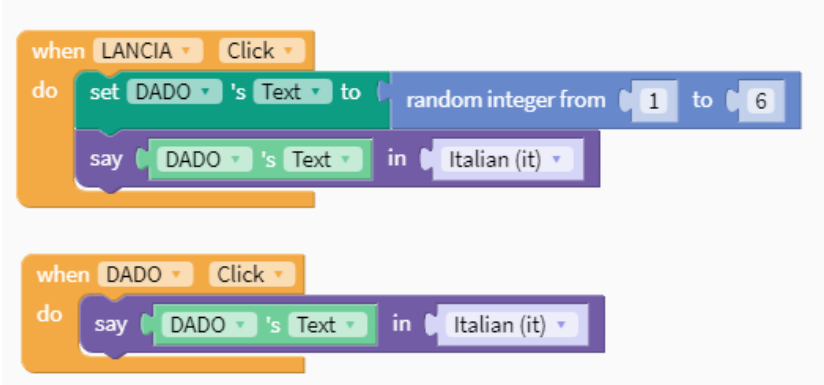
App dado parlante

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

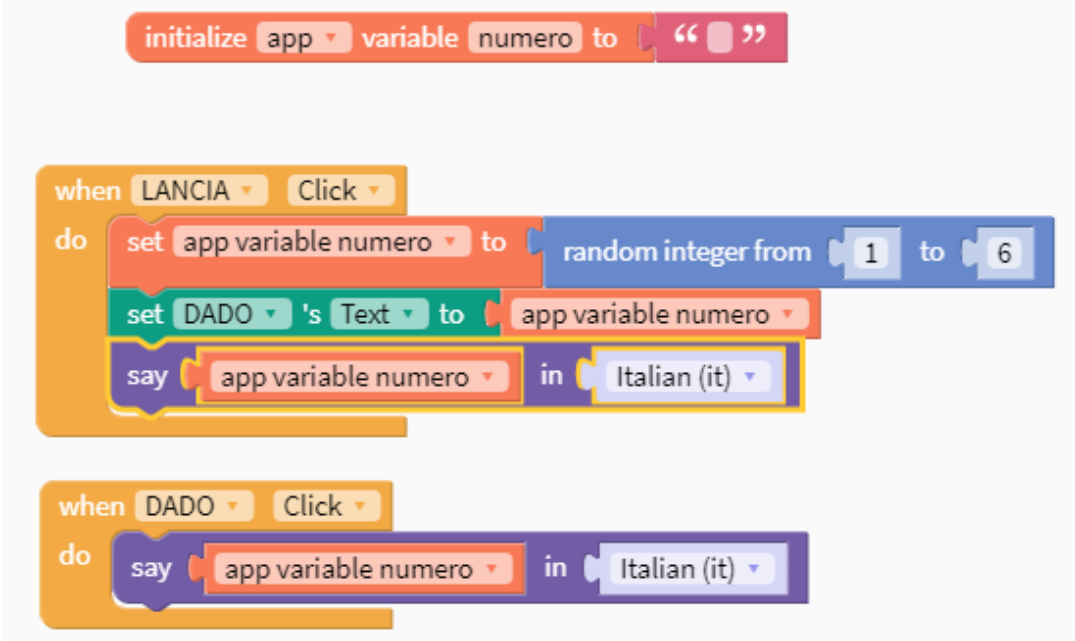


App dado parlante

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



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.

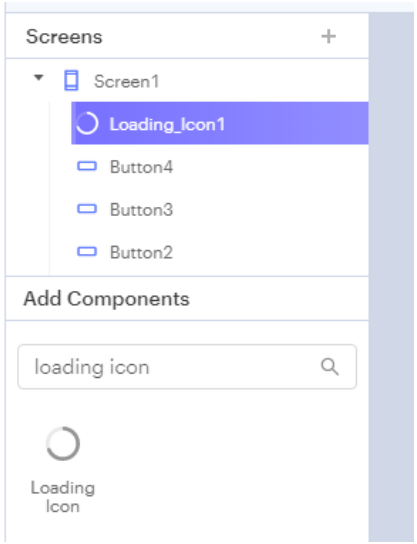
Proviamo a realizzare una semplice funzione che poi andremo a richiamare in altri blocchi.

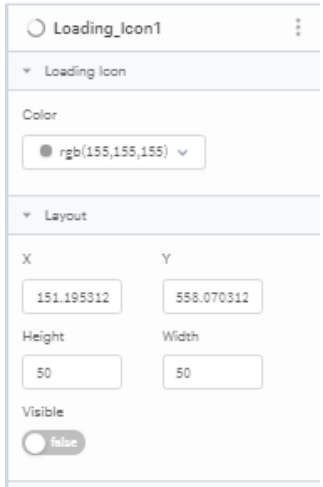
La funzione che vogliamo realizzare, quando viene chiamata, deve eseguire 3 istruzioni:

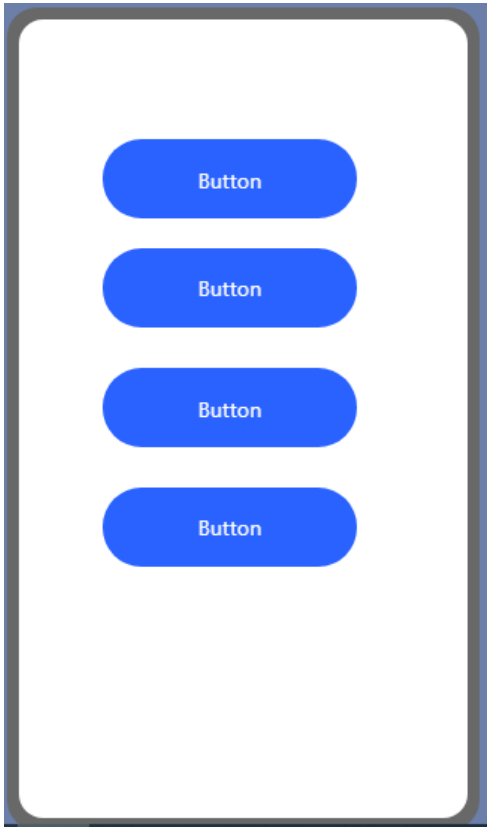
1) fornire un feedback visivo
2) fornire un feedback tattile
3) fornire un feedback audio

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.



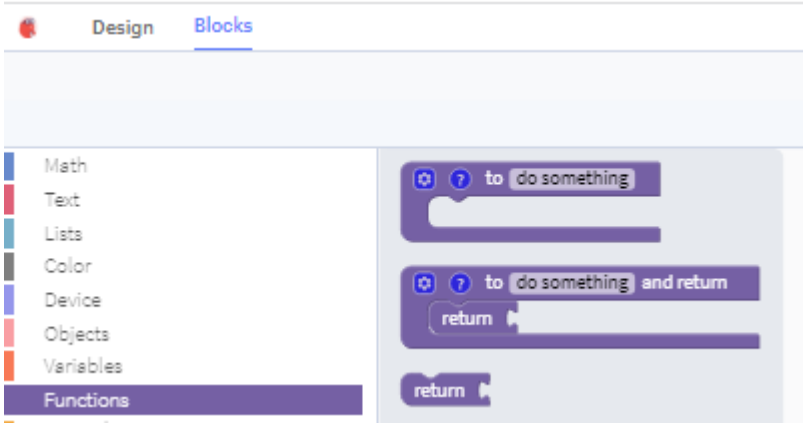


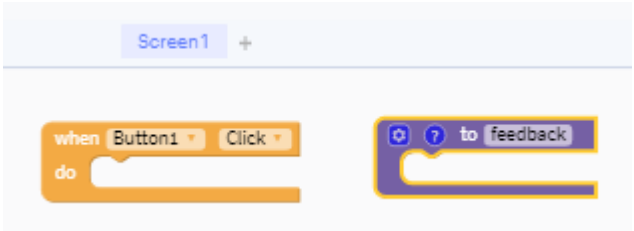


Creiamo una funzione 1

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

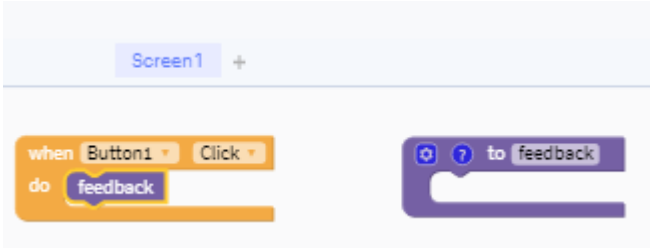
Dai blocchi Core selezioniamo “Functions”, trasciniamo un blocco funzione nel desk e chiamiamolo “feedback”.





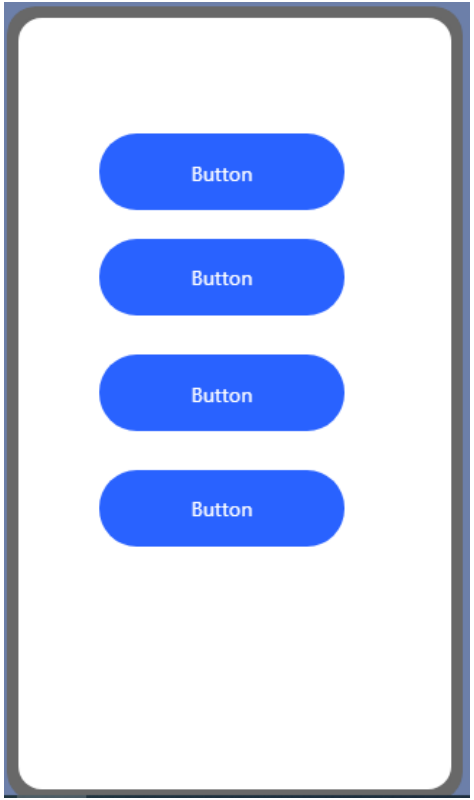
Creiamo una funzione 2

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.



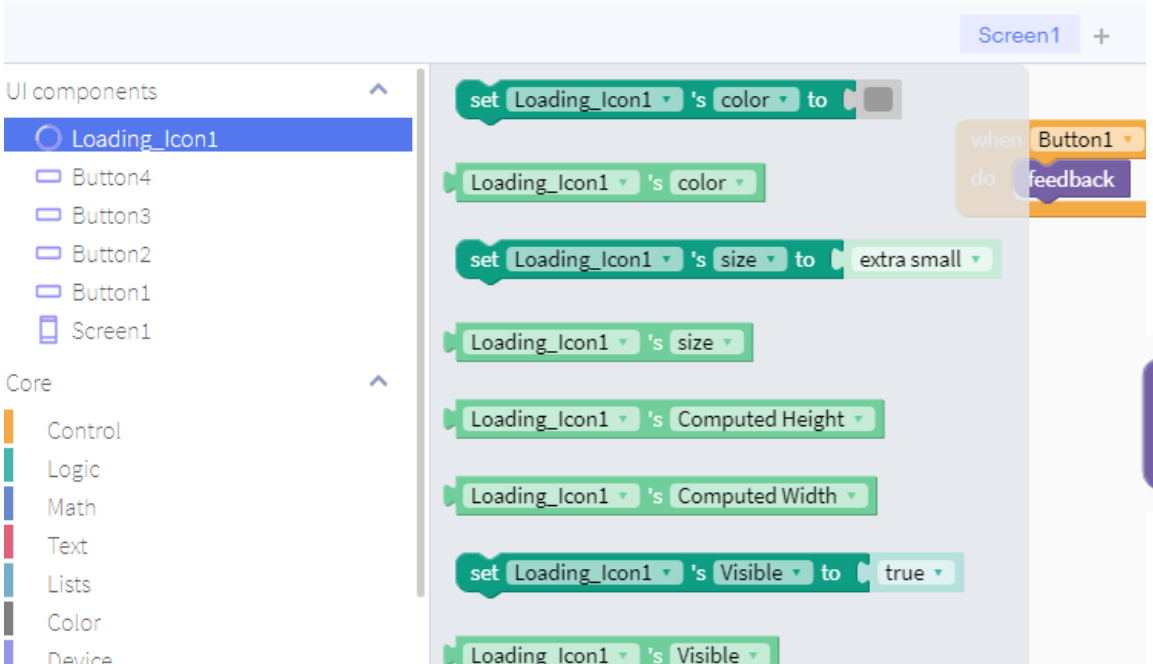
Proviamo la funzione senza istruzioni

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



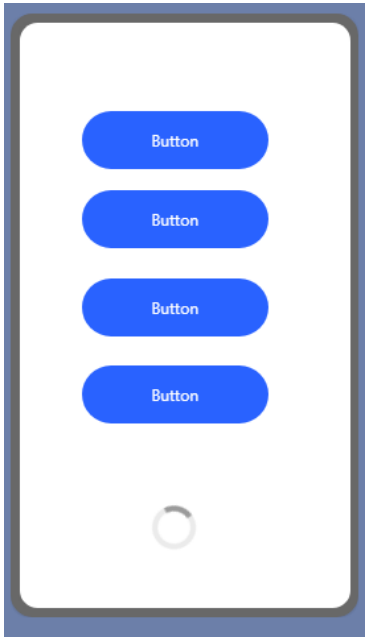
Aggiungiamo le istruzioni alla funzione - feedback visivo

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



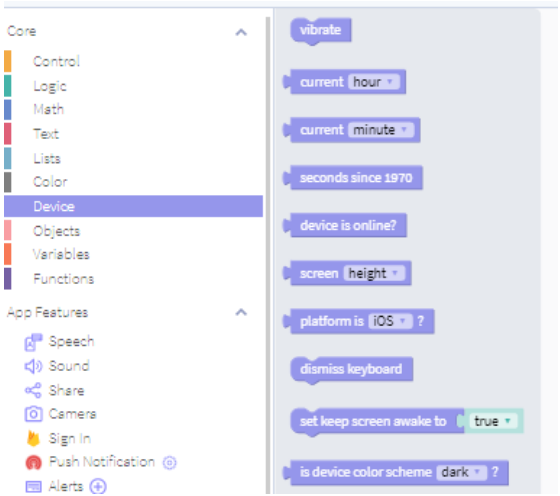
Proviamo la funzione con istruzione per feedback visivo

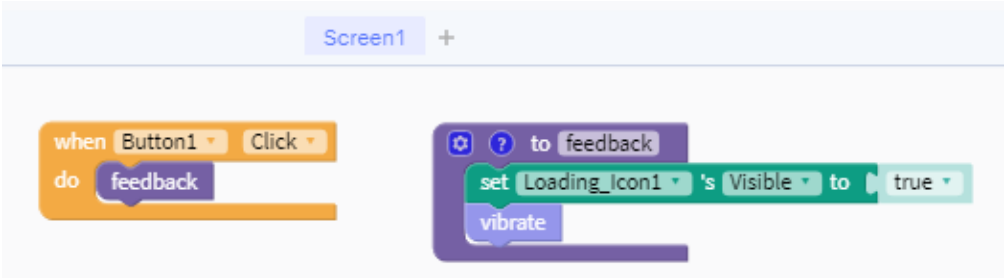
Se proviamo la app ci accorgiamo che cliccando il pulsante comparirà su schermo l’icona di caricamento.



Aggiungiamo le istruzioni alla funzione - feedback tattile

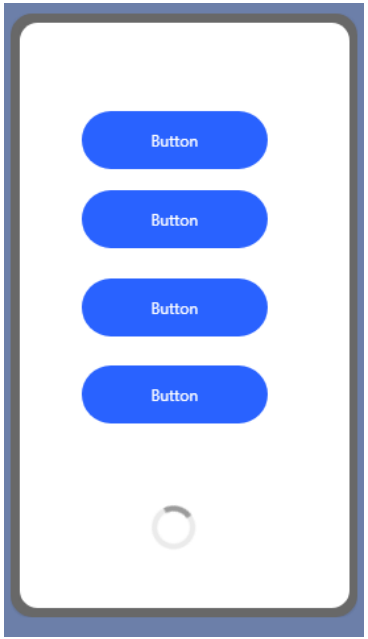
Andiamo tra i blocchi del dispositivo (Device) e trasciniamo il blocco “vibrate” all’interno della funzione.





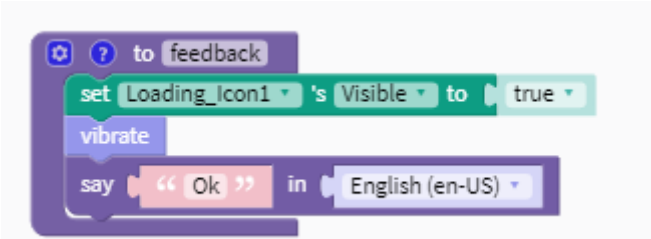
Proviamo la funzione a cui è stato aggiunto feedback tattile

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) .



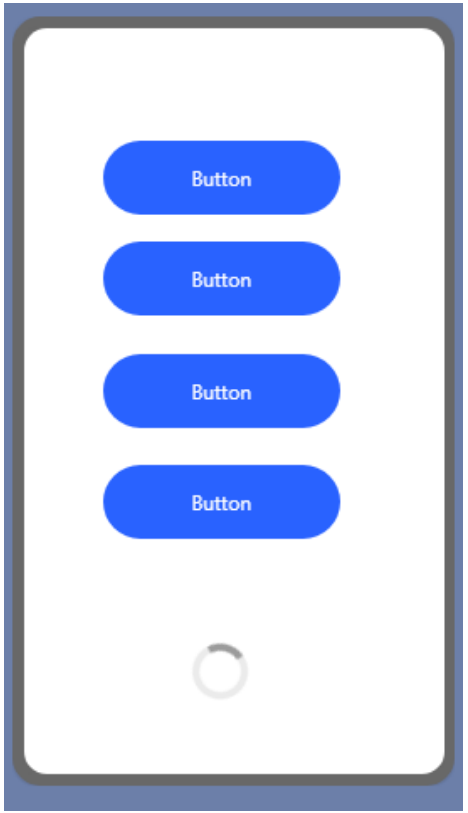
Aggiungiamo le istruzioni alla funzione - feedback audio

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



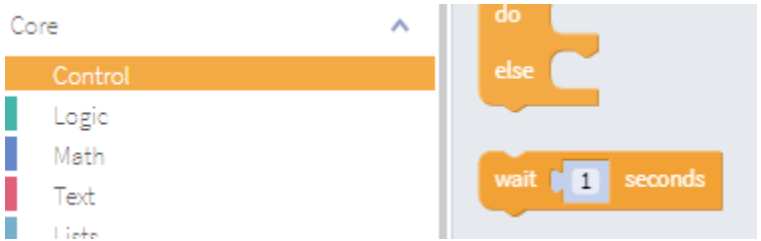
Proviamo la funzione a cui è stato aggiunto feedback audio

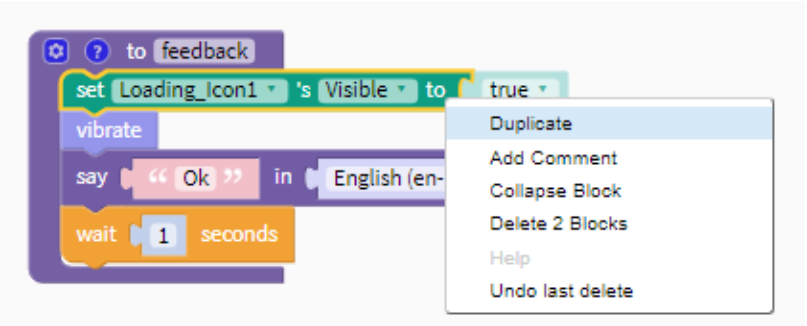
Adesso la funzione è completa, cliccando il pulsante comparirà su schermo l’icona di caricamento (feedback visivo), il dispositivo emetterà una vibrazione (feedback tattile) e verrà riprodotto il messaggio vocale “Ok” (feedback audio).

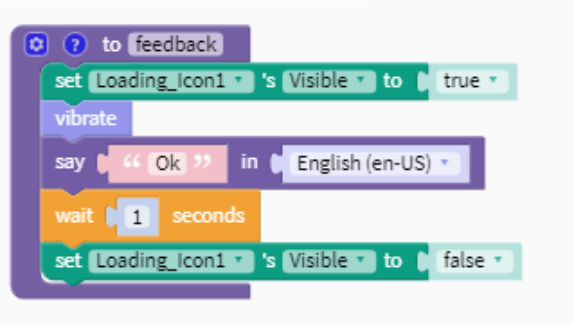


Per concludere la funzione..

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.

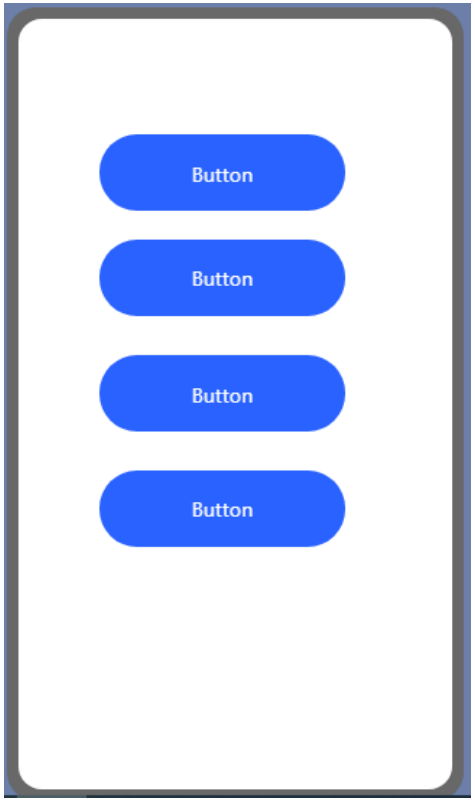






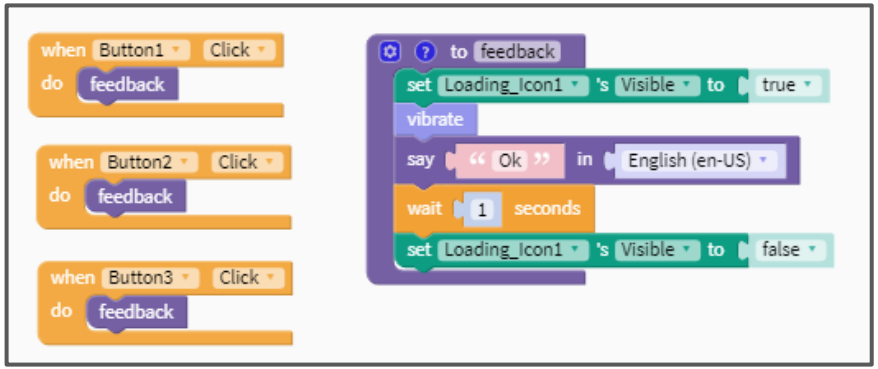
Proviamo la funzione a cui è stata aggiunta una pausa prima di disattivare la loading icon

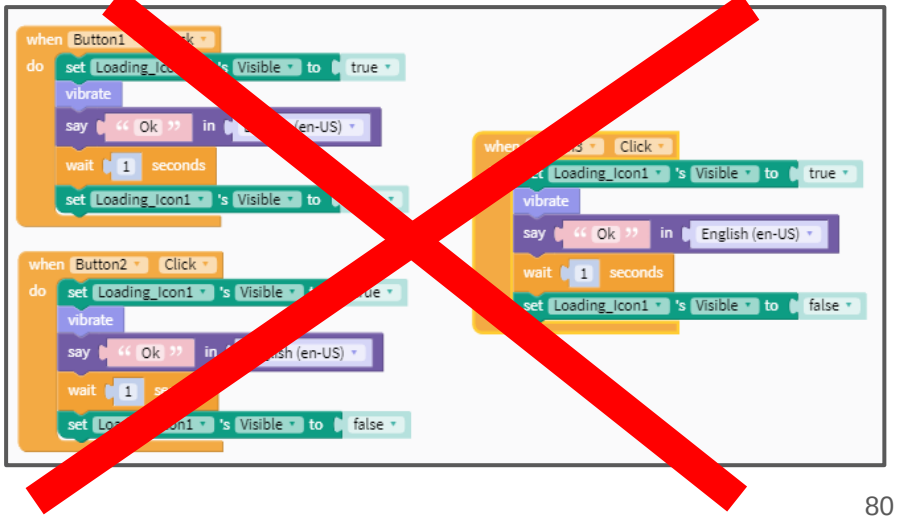
Adesso la loading icon scompare una volta concluse tutte le istruzioni tattili e audio.



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..

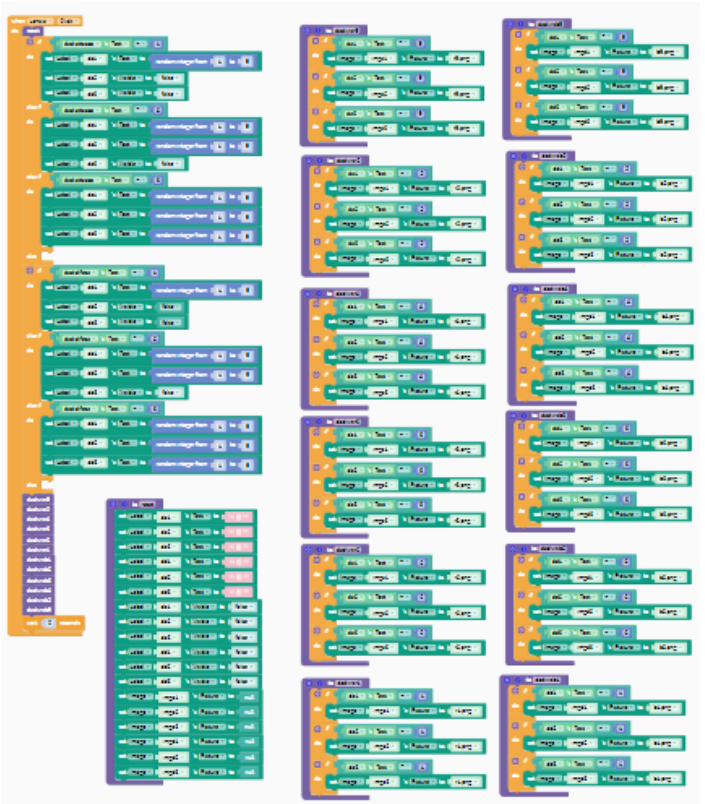




Esempi pratici che mostrano perchè dobbiamo usare le funzioni 2

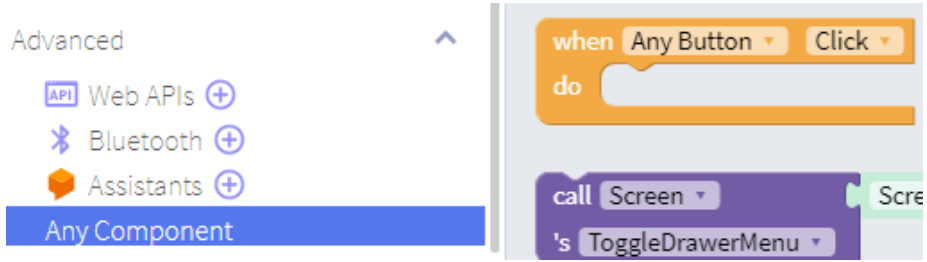
..se dovessimo apportare delle modifiche alla nostra app basterebbe lavorare sulla singola funzione e le modifiche si ripercuoterebbero su tutti i blocchi in cui la abbiamo richiamata.

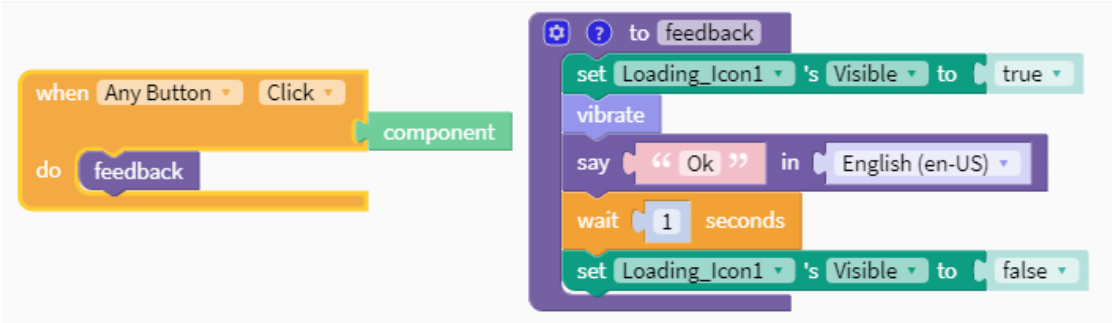
Immaginate invece se dovessimo andare a cercare tutti i punti in cui abbiamo utilizzato quelle stesse istruzioni che ora dobbiamo modificare, oltre a perdere molto tempo si rischierebbe di non arrivarne a capo e l'errore sarebbe a portata di mano!

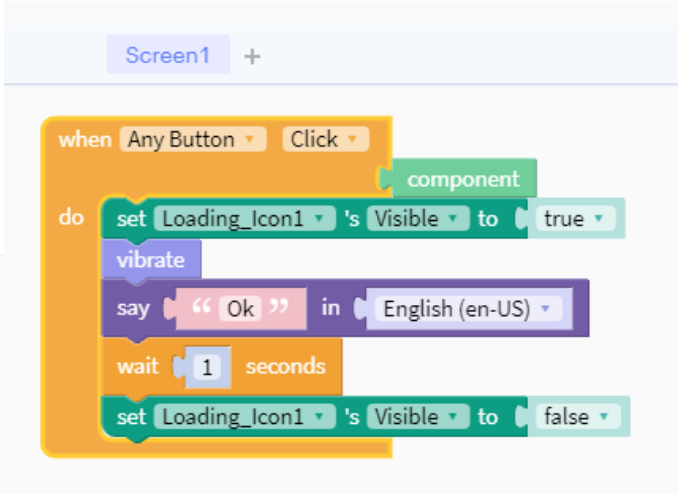


E se la funzione si fosse dovuta applicare a tutti i pulsanti presenti nella UI anzichè solo ad alcuni?

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.







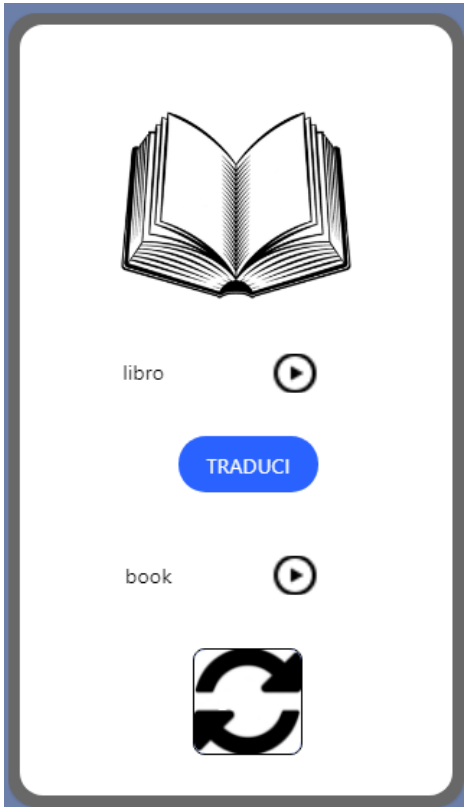
Sviluppiamo alcune app insieme passo per passo

1) App per insegnare ai bambini una lingua straniera
2) App dado parlante
3) 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

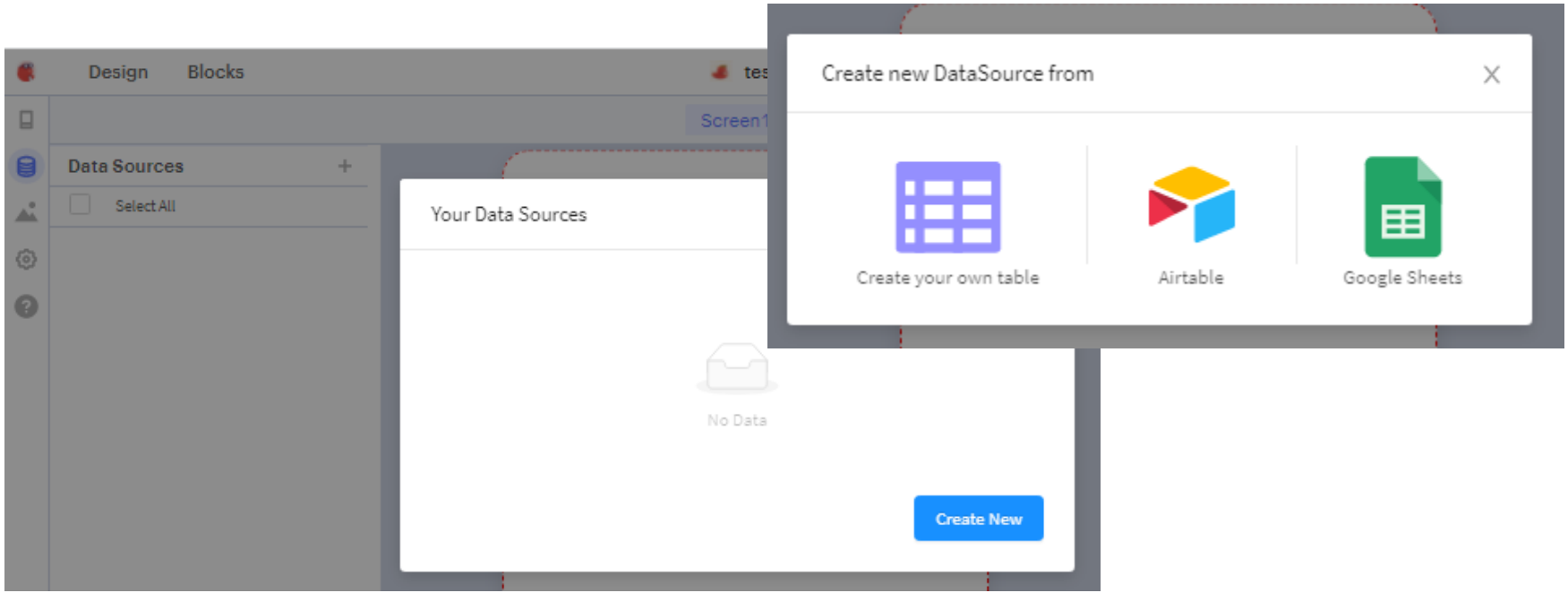
Inseriamo nel design:

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.

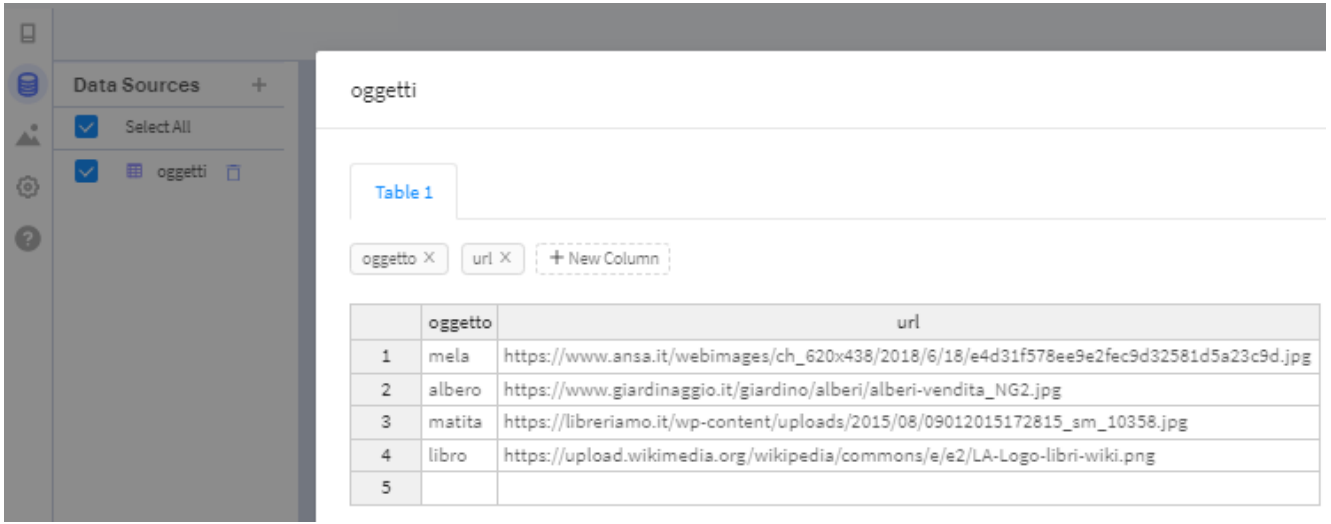
App per insegnare ai bambini una lingua straniera - DB

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

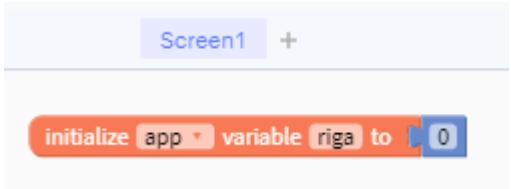
Costruiamo ora una nostra base di dati (chiamiamola ad esempio “oggetti”) composta da sole due colonne (oggetto e url) e andiamola a popolare di contenuti. Possiamo usare il servizio gratuito ImgBB (https://it.imgbb.com/) per caricare una immagine e ottenerne l’url.



	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).

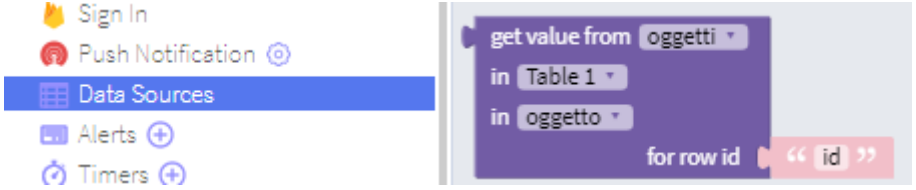


App per insegnare ai bambini una lingua straniera - random

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”.

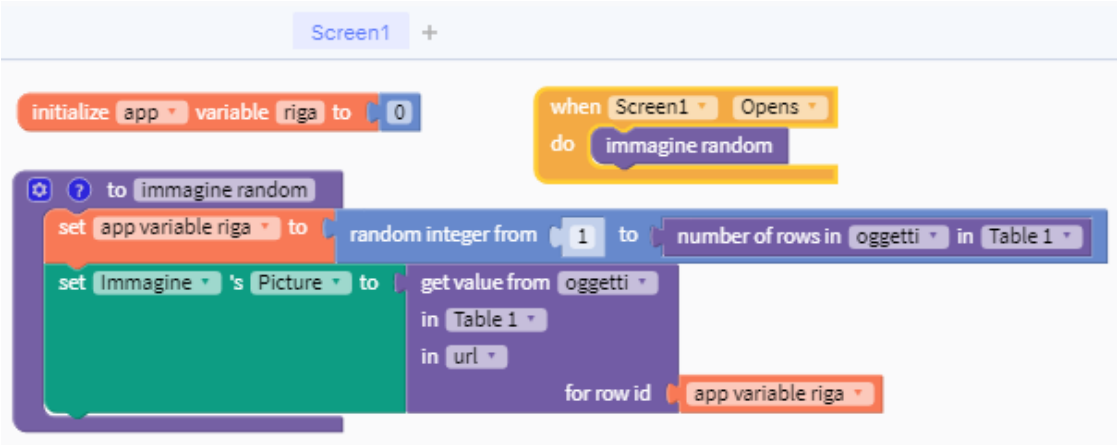
App per insegnare ai bambini una lingua straniera - recuperare valore da DB per impostare immagine

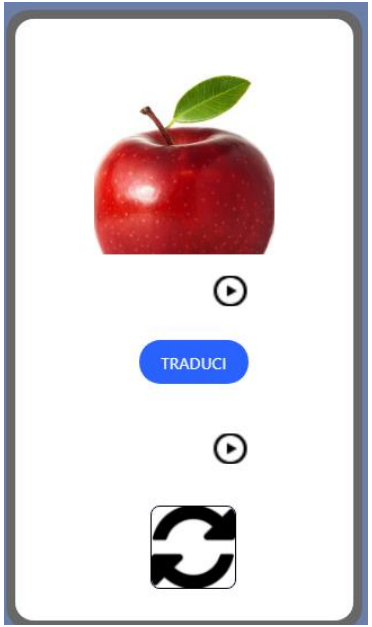
Inseriamo un blocco che usi l'url associato alla variabile riga in modo da mostrare una immagine.



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

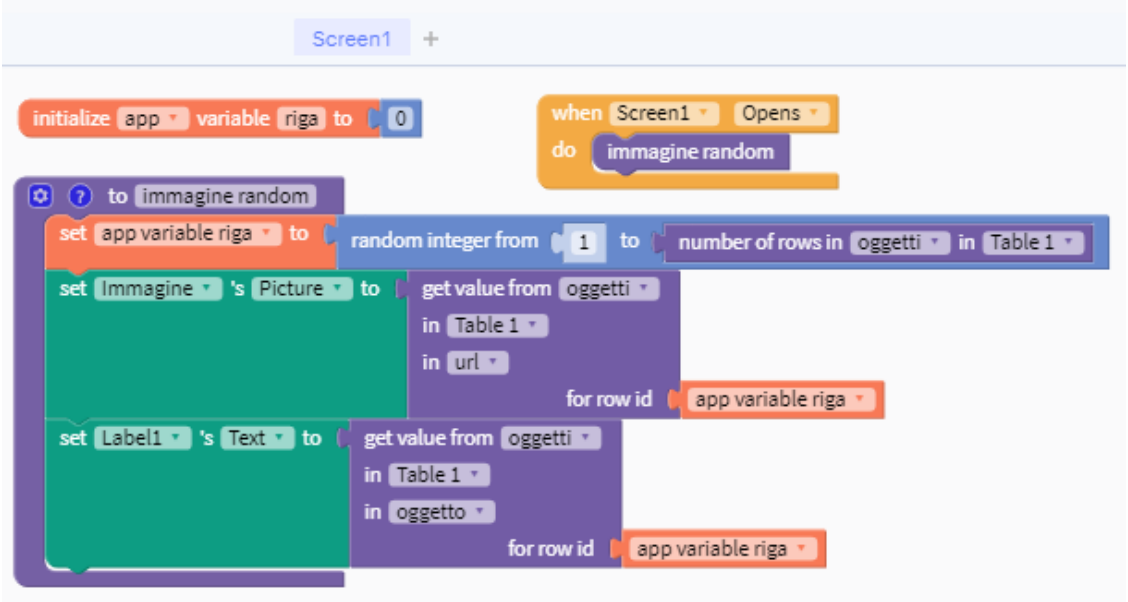
Inseriamo ora un blocco evento che esegua la funzione immagine random quando la schermata si apre. Se ora proviamo la app vedremo comparire una immagine.





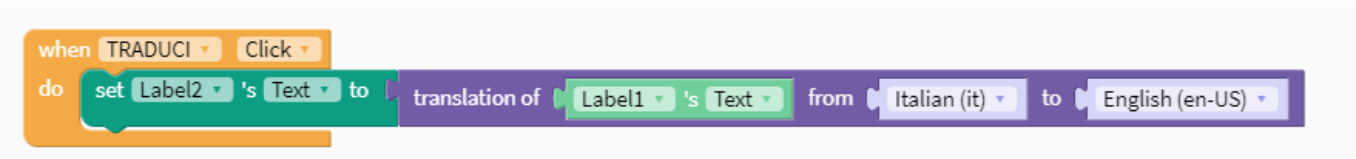
App per insegnare ai bambini una lingua straniera - recuperare valore da DB per impostare testo

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.



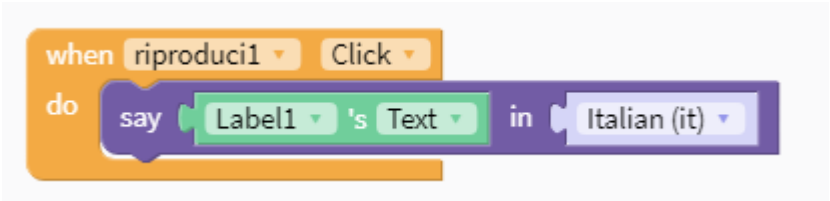
App per insegnare ai bambini una lingua straniera - pulsante traduci

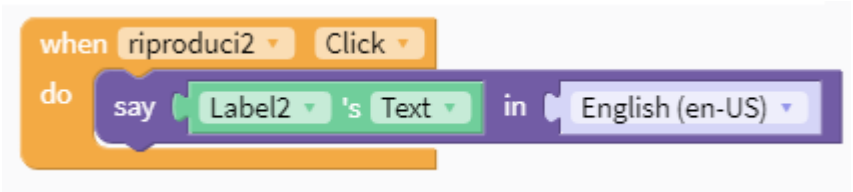
Impostiamo il pulsante “traduci” perchè effettui la traduzione del testo della prima etichetta da italiano a inglese.



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.

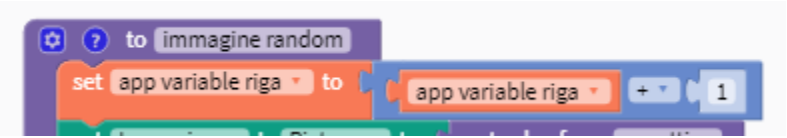




App per insegnare ai bambini una lingua straniera

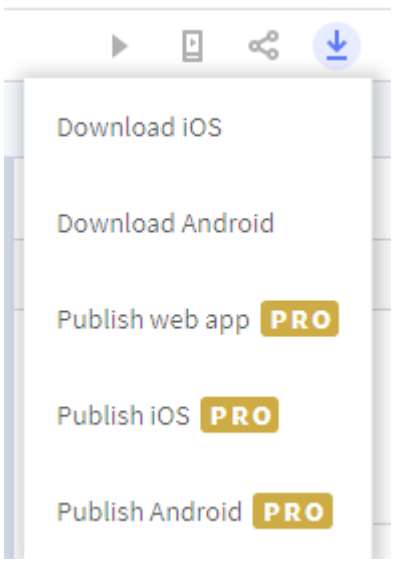
Impostiamo il pulsante “aggiorna” perchè esegua la funzione “immagine random” e la app è pronta.

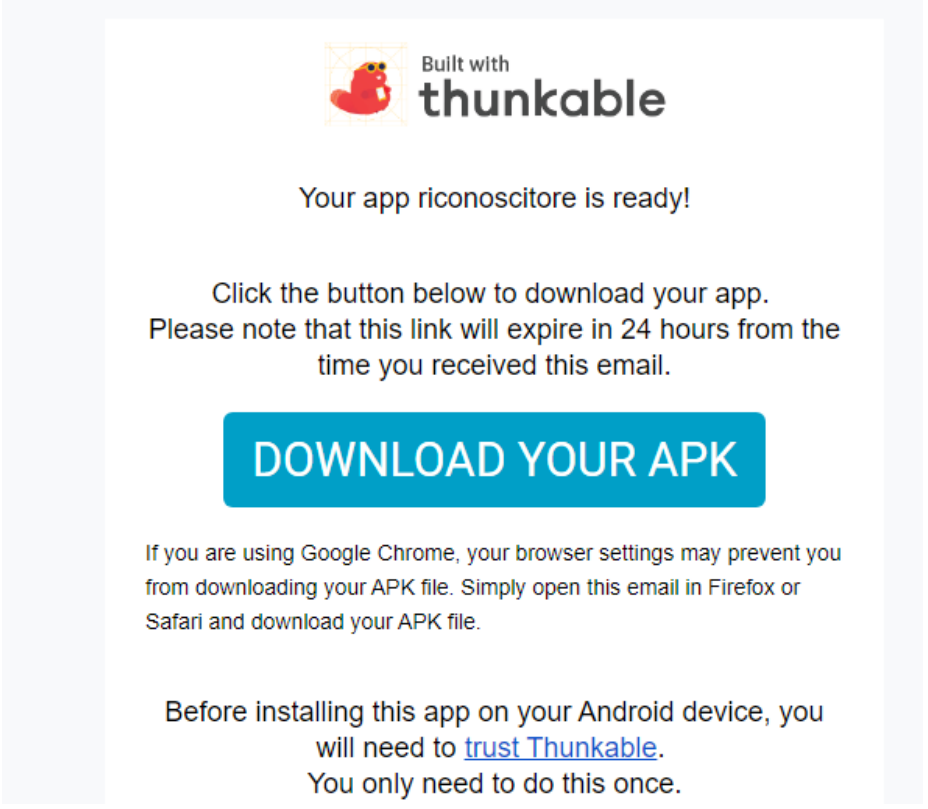
Se invece di usare un sistema casuale di scelta delle immagini volete usare un metodo sequenziale, basterà andare a modificare la funzione perchè invece di scegliere un numero a caso, si limiti ad incrementare di 1 ogni volta il valore della nostra variabile riga.



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.





Documentazione utile

https://docs.thunkable.com/