WORKSHOP

18 MAGGIO > ORE 16.30

Thunkable come strumento di sviluppo a supporto dell'accessibilità

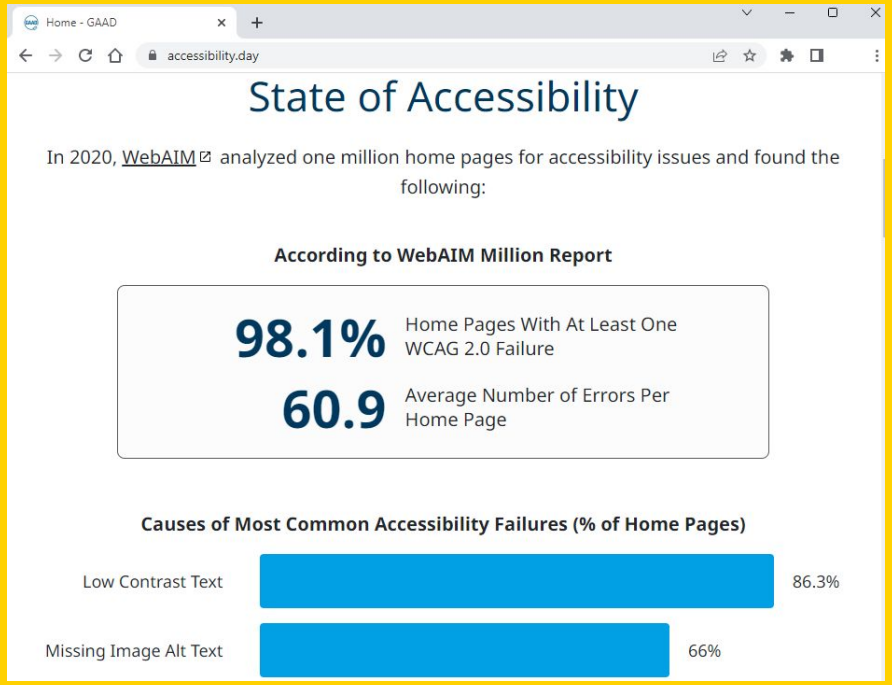
Barbara Leporini
Primo Ricercatore
@ CNR-ISTI

Giulio Galesi
Tecnico
@ CNR-ISTI

Alina Vozna
Assegnista di Ricerca
@ CNR-ISTI

ACCESSIBILITY DAYS 2023

18 Maggio Global Accessibility Awareness Day



https://accessibility.day/

Sviluppo di app senza codice

gartner.com/en/newsroom/press-releases/2021-11-10-gartner-says-cloud-will-be-the-centerpiece-of-new-digital-experiences

Low-Code and No-Code Technologies Use Will Nearly Triple by 2025

Application development will shift to application assembly and integration. The applications will be assembled and composed by the teams that use them. "The technological and organizational silos of application development, automation, integration and governance will become obsolete," said Govekar. "This will drive the rise of low-code application platforms (LCAPs) and citizen development."

By 2025, 70% of new applications developed by organizations will use low-code or no-code technologies, up from less than 25% in 2020. The rise of low-code application platforms (LCAPs) is driving the increase of citizen development, and notably the function of business technologists who report outside of IT departments and create technology or analytics capabilities for internal or external business use.

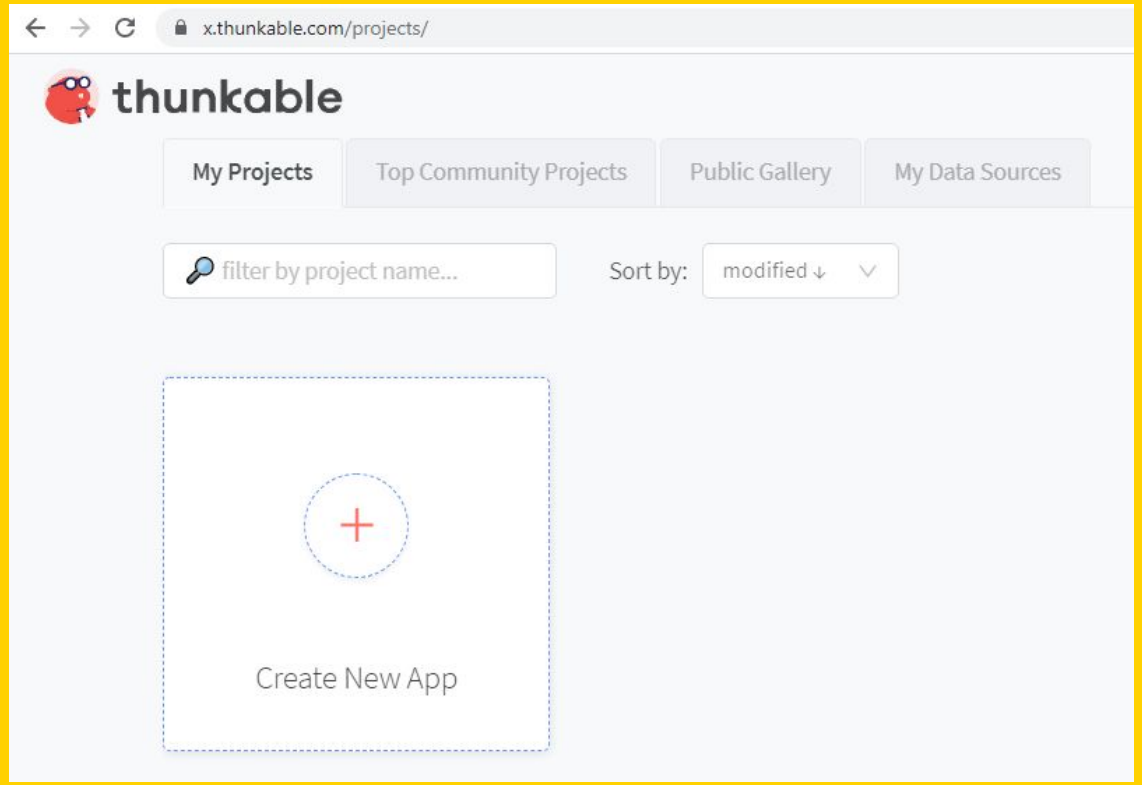
Thunkable

thunkable

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

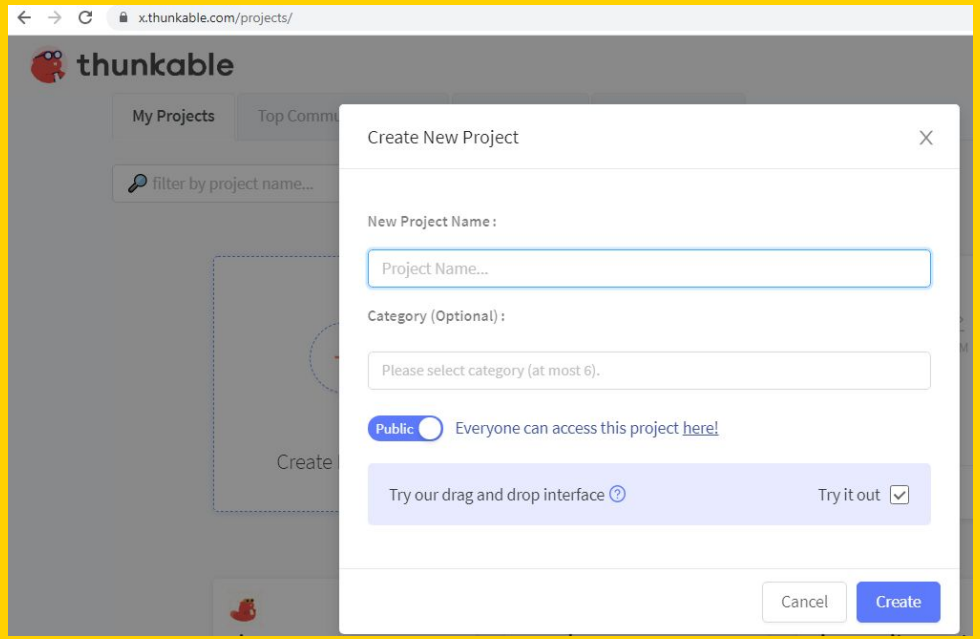
La versione gratuita consente di progettare app, realizzare velocemente dei prototipi, programmare complesse funzionalità e testare la propria app live sul proprio smartphone.
La versione Pro a pagamento (38$ mensili) consente di monetizzare le app inserendole su Google Play Store e sull'App Store di Apple.
E' anche possibile creare web-app, ovvero app che non è necessario scaricare e a cui è possibile accedere direttamente online.
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!

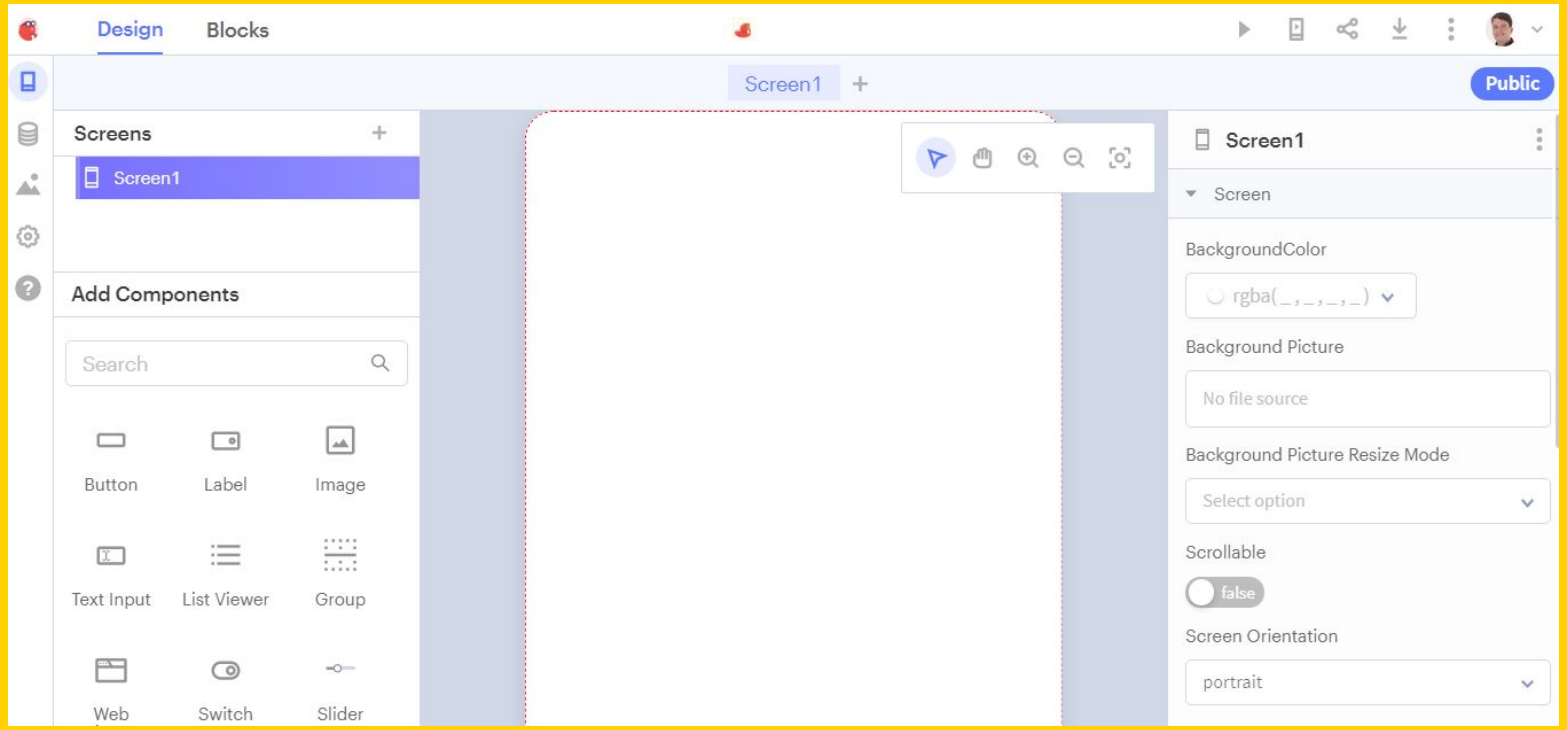


https://thunkable.com/

Creazione progetto usando l’interfaccia con drag and drop

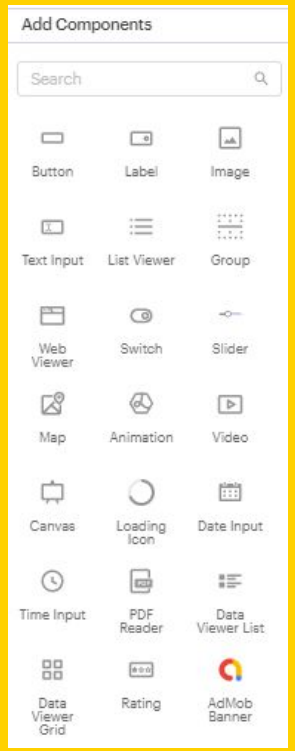


Design Page - Screens



Design Page - Components

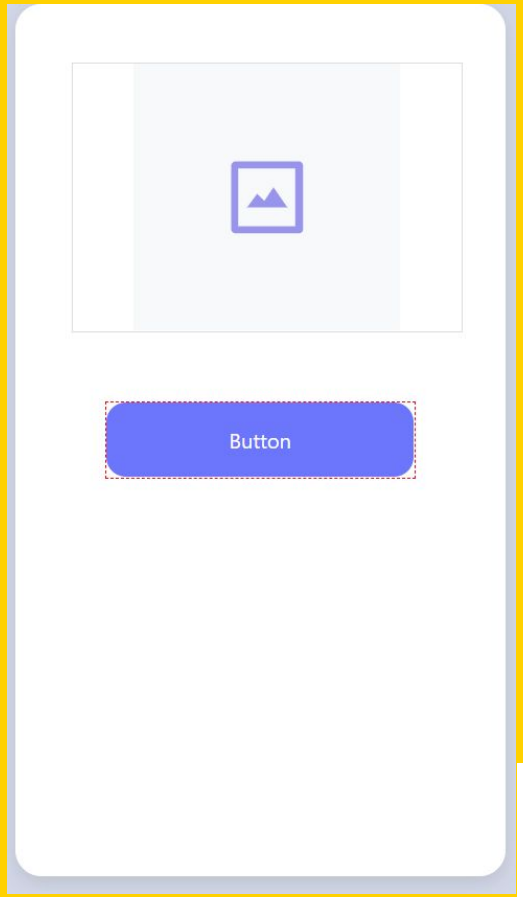
Si tratta delle componenti visibili dell'interfaccia utente:



- Button
- Label
- Image
- Text Input
- List Viewer
- Group
- Web Viewer
- Switch
- Slider
- Map
- Animation
- Video
- Canvas
- Loading Icon
- Date Input
- Time Input
- PDF Reader
- .. e molte altre

Realizziamo una semplice interfaccia su Thunkable

Aggiungiamo una immagine e un pulsante



Primo problema di accessibilità: su Thunkable manca la possibilità di inserire il testo alternativo per le immagini!

Se l'immagine che inseriamo è puramente decorativa dobbiamo sovrapporre all'immagine una etichetta (label) trasparente in cui inseriamo il testo alternativo che descrive l'immagine, utilizzando un font trasparente (non visibile).

altra soluzione

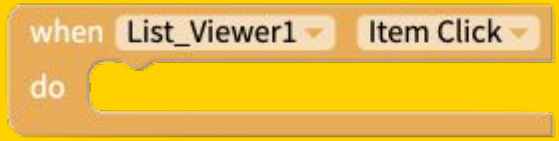
Se l’immagine deve poter essere cliccata, andiamo ad inserire un pulsante in cui impostiamo l’immagine come sfondo ed inseriamo come testo del pulsante la descrizione dell’immagine usando un font non visibile.

In questo modo la app creata sarà pienamente fruibile anche da chi usa uno screen reader.

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 funzionalità non visibili ma che sono in esecuzione in background.

- Eventi



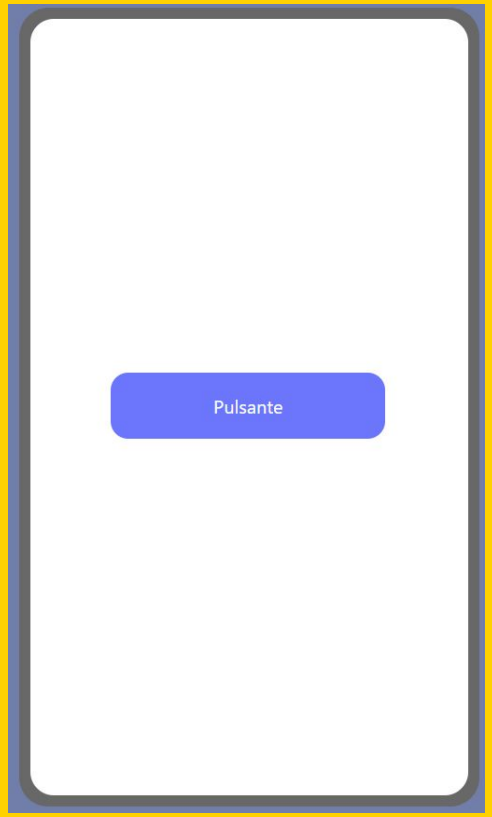
- Chiamate e Proprietà

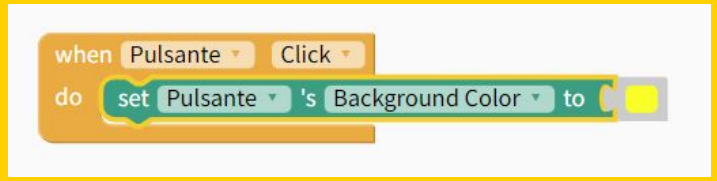
I blocchi principali (core)

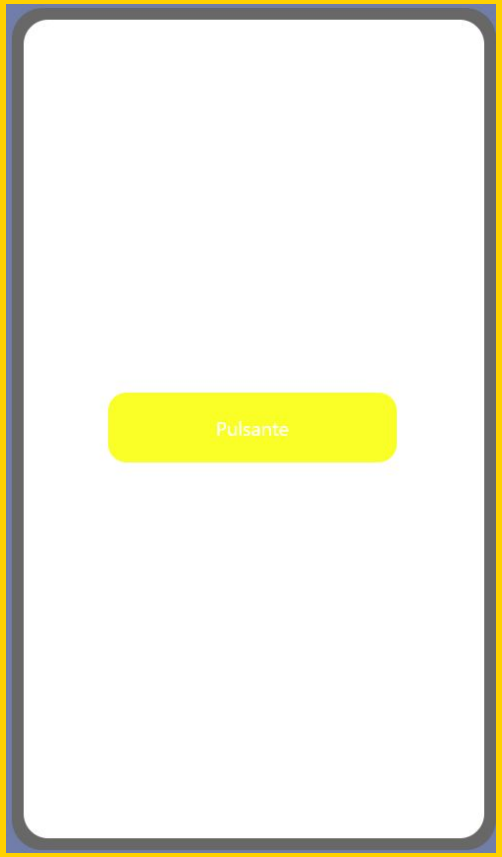
Abbiamo 10 categorie di blocchi principali che ci consentono di impostare regole e programmare la nostra app:

- 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 (archiviare valori che poi verranno richiamati)
- Funzioni

Come funzionano i blocchi: creiamo un pulsante che quando viene cliccato modifichi il colore di sfondo


Pulsante

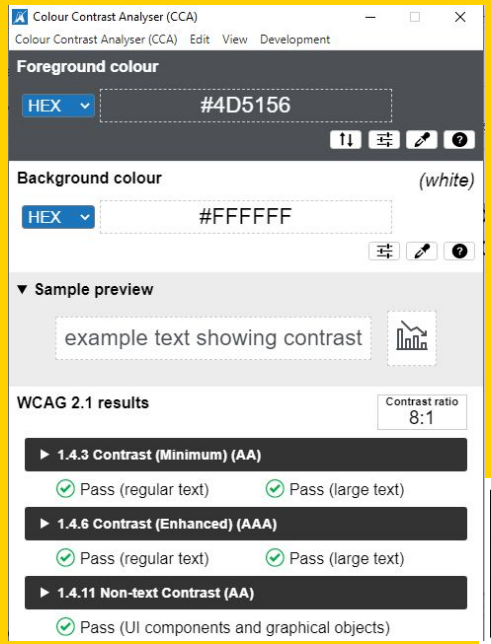

when Pulsante Click
do set Pulsante 's Background Color to


Pulsante

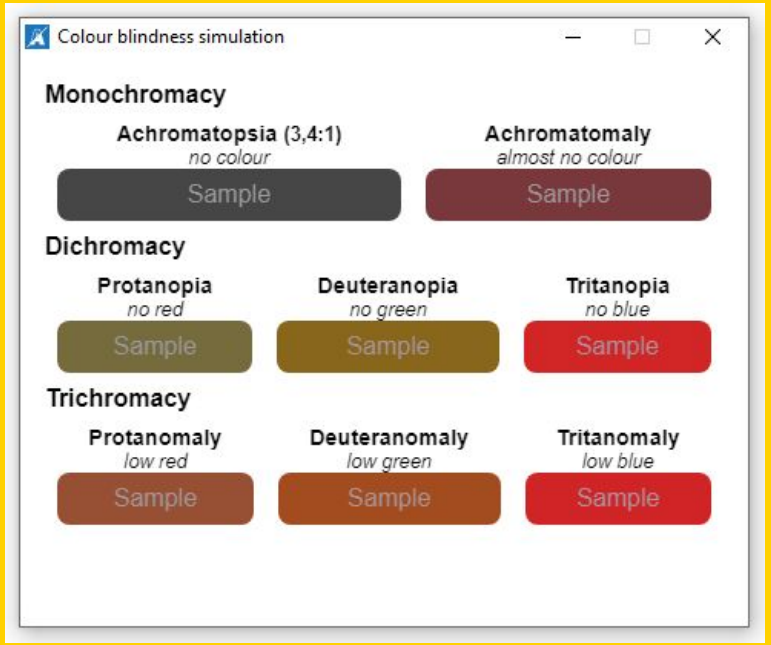
Verifica del contrasto con Colour Contrast Analyser

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)

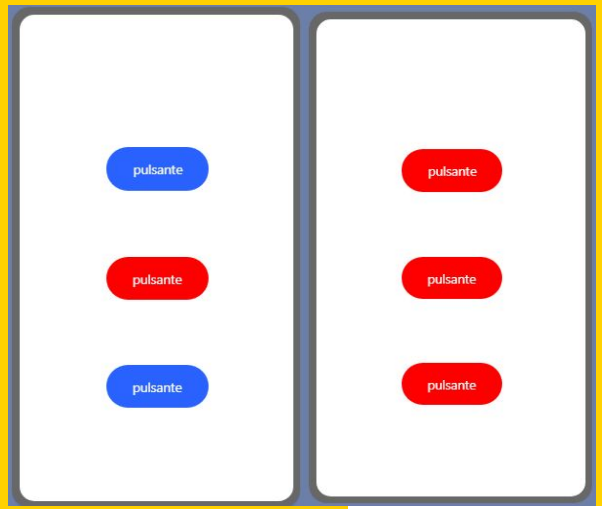


Colour Contrast Analyser consente anche di simulare le varie condizioni di daltonismo



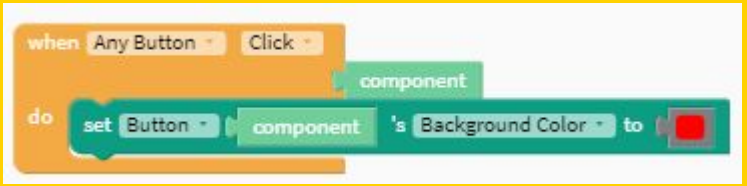
Aggiungiamo altri due pulsanti identici con la duplicazione

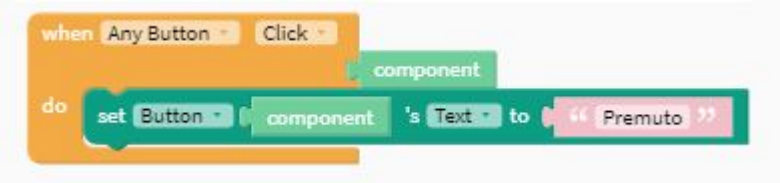
Usando la duplicazione delle componenti della UI, Thunkable ci avvisa che i relativi blocchi funzione non verranno duplicati quindi solo un pulsante cambierà colore una volta cliccato! Come procedere?



Usiamo il blocco avanzato “Any component”

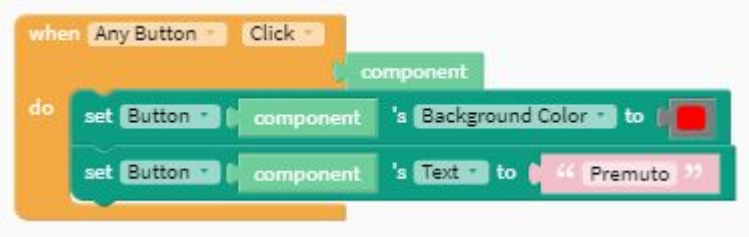
Possiamo scegliere di modificare il testo del pulsante oltre che lo sfondo, andando a duplicare i blocchi e modificare la proprietà Text anzichè Background Color.





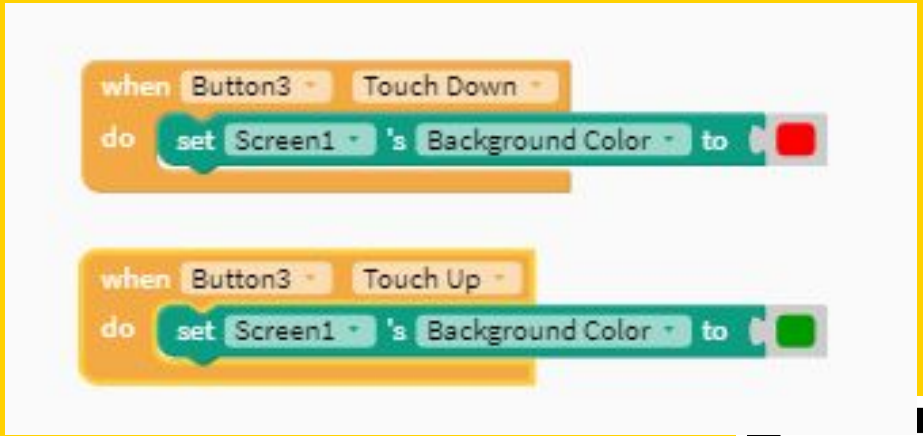
Usiamo il blocco avanzato “Any component”

Prestiamo attenzione a semplificare i blocchi quando possibile.



Possiamo usare anche 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.

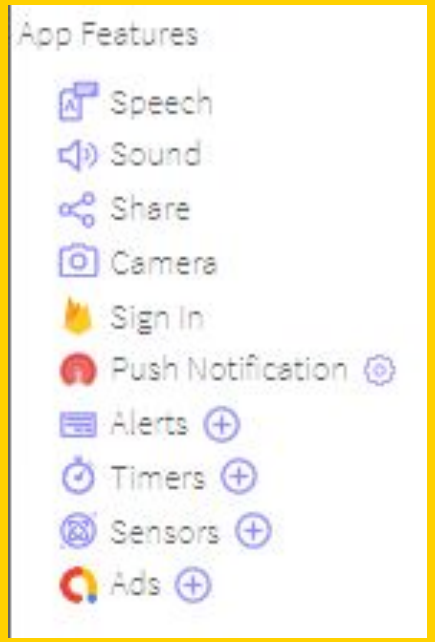


Proviamo a realizzare una semplice app che riproduce suoni...

Possiamo pensare ad una app che verrà usata da una persona anziana per comunicare con il proprio caregiver.
All'avvio della app possiamo inserire un suono di benvenuto.
Dalla pagina di design inseriamo solo 3 pulsanti, ciascuno dei quali, una volta cliccato, riprodurrà un messaggio vocale differente.
Andiamo ad inserire 3 file audio in formato mp3 in Assets - Media Files e poi componiamo i blocchi per far funzionare la app in pochi secondi.

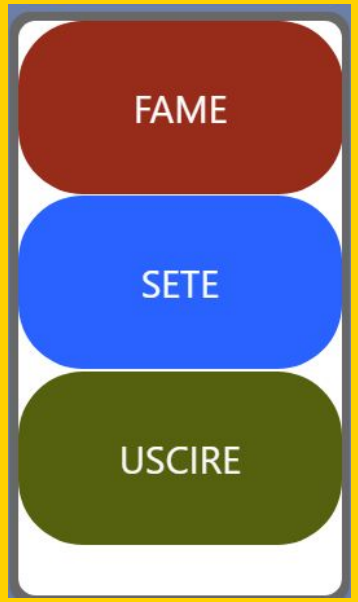
Le funzionalità (App features)

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



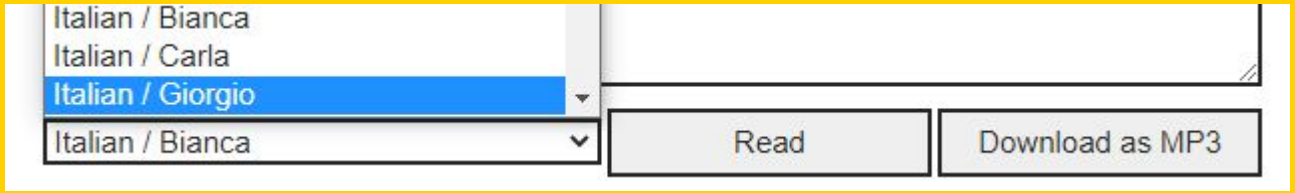
- Speech
- Sound
- Share
- Camera
- Sign In
- Push Notification
- Alerts
- Timers
- Sensors
- Ads

Proviamo a realizzare una semplice app che riproduce suoni...



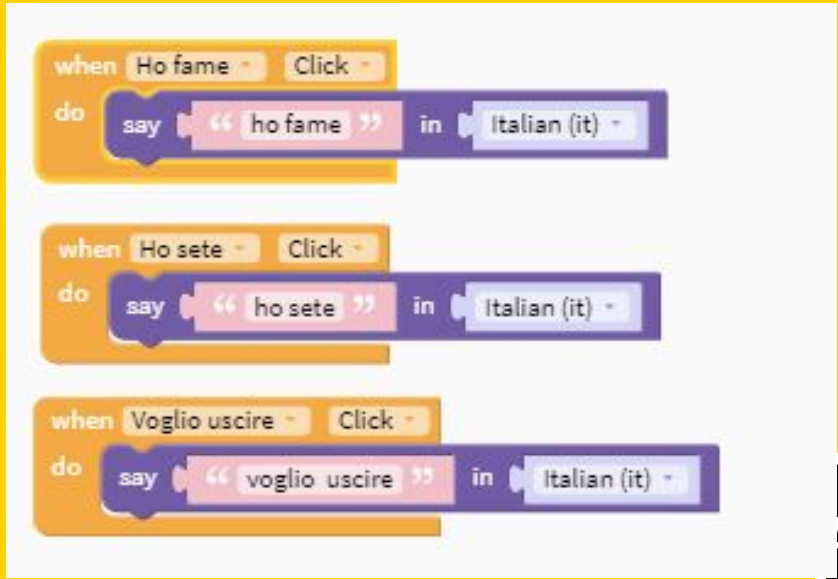
Free Text-To-Speech

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.



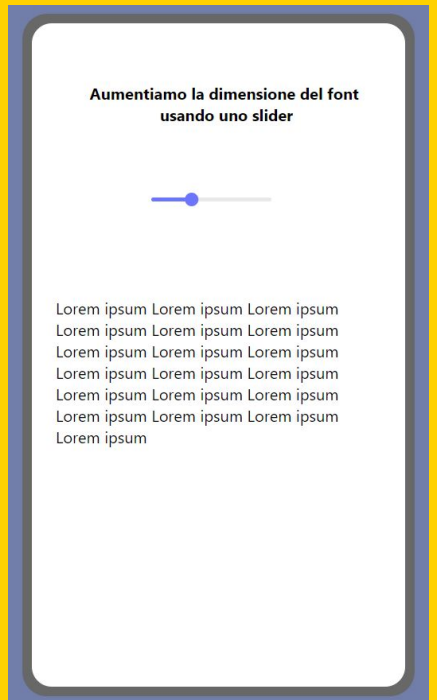
Anche Thunkable ha un suo modulo TTS

Modifichiamo la app realizzata usando i blocchi del text-to-speech di Thunkable anzichè i file audio mp3.
Funziona correttamente ma notiamo che la qualità della pronuncia è peggiore.



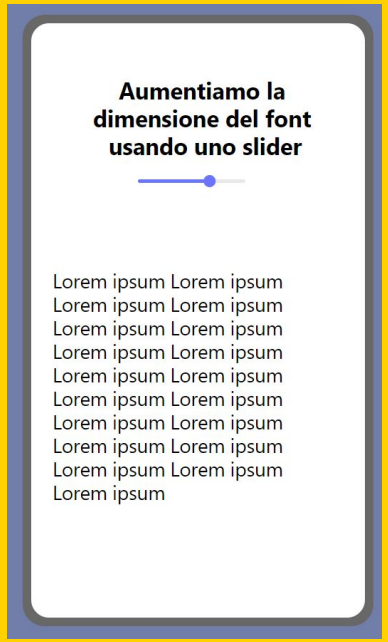
Aumentiamo la dimensione del carattere usando uno slider

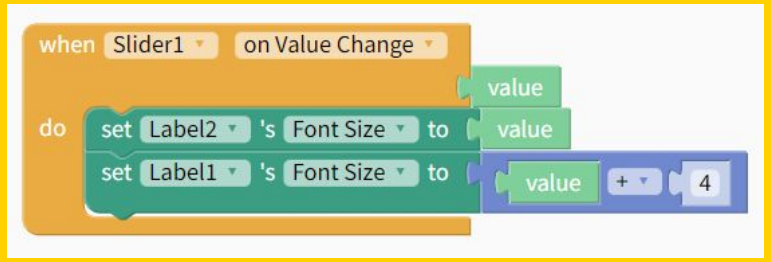
Inseriamo due etichette e al centro posizioniamo uno slider

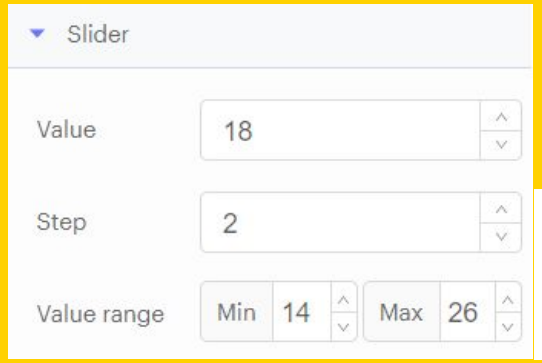


Aumentiamo la dimensione del carattere usando uno slider

il font della etichetta principale deve avere un font superiore rispetto a quello dell’altra etichetta

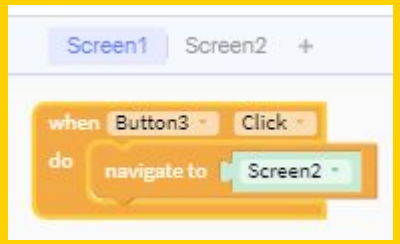


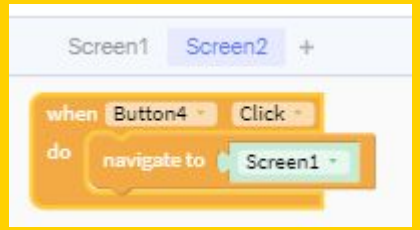


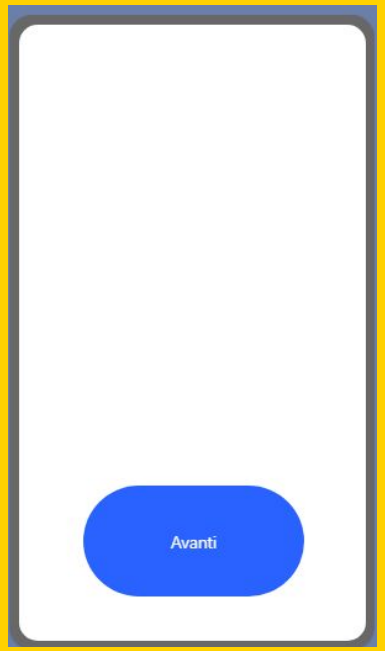


Blocco navigate

Consente di spostarci da uno screen all'altro, vediamo un esempio semplice creando due schermate ciascuna con un pulsante (schermata 1 con pulsante avanti e schermata 2 con pulsante indietro)

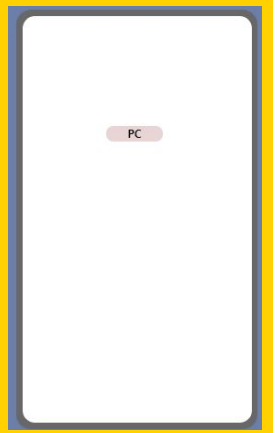


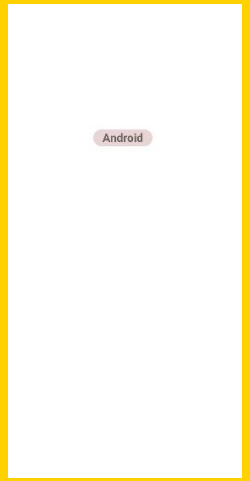




Blocchi if-do-else ed else-if

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". Creiamo una app che emetta una vibrazione quando si avvia e che ci indichi semplicemente quale piattaforma stiamo utilizzando.





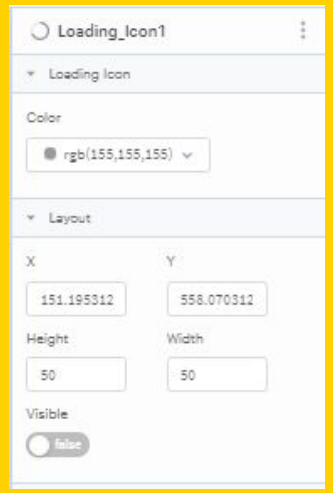
Blocco funzione in Thunkable

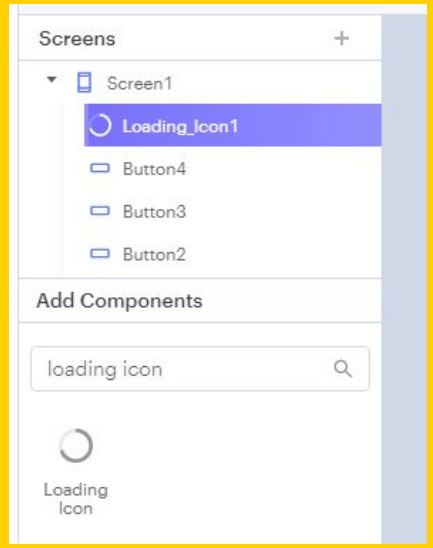
Il blocco funzione ci consente di raggruppare più istruzioni in un unico blocco che potrà essere richiamato più volte nella nostra app senza dover duplicare i blocchi istruzione di cui è costituito.
Proviamo a realizzare una semplice funzione che poi andremo a richiamare in altri blocchi.
La funzione 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.



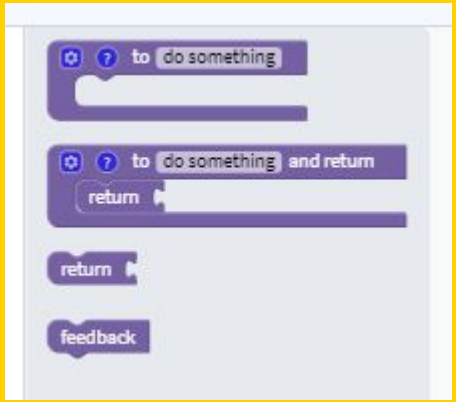


Creiamo una funzione 1

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”.

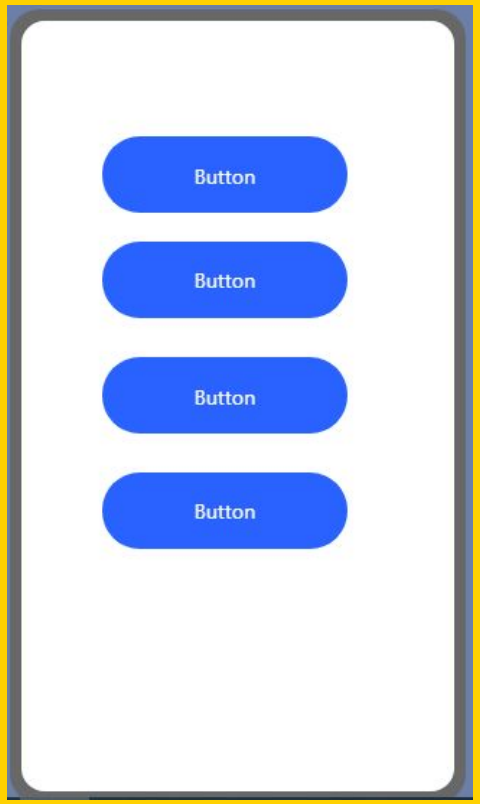
Creiamo una funzione 2

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

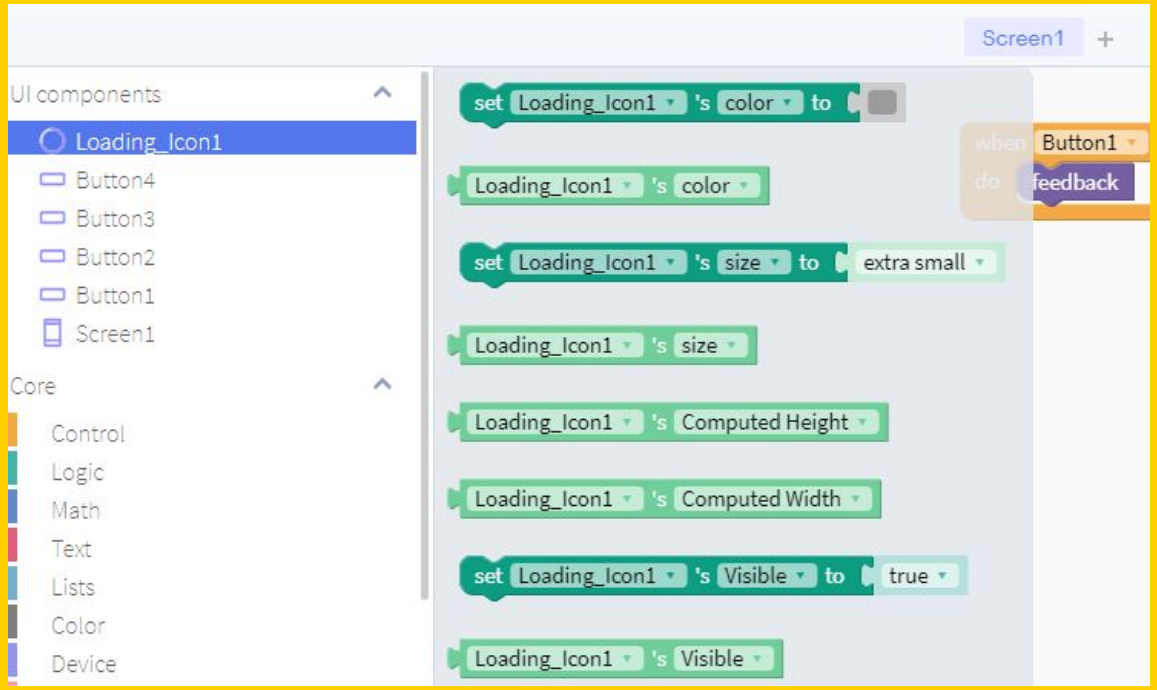
Se proviamo la app ci accorgiamo che cliccando il pulsante non succede nulla, visto che non abbiamo inserito istruzioni nella funzione feedback.

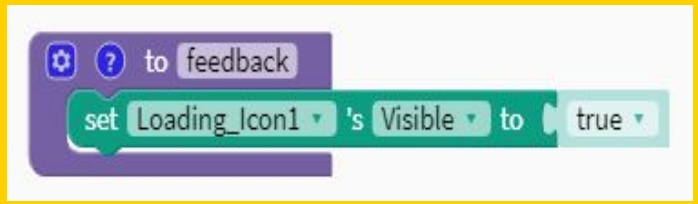


Aggiungiamo le istruzioni alla funzione - feedback visivo

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

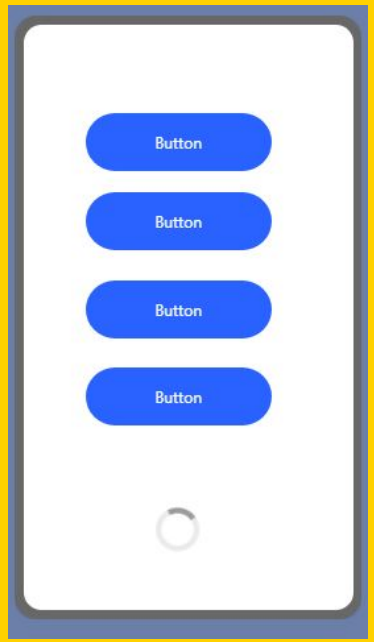
Aggiungiamo le istruzioni alla funzione - feedback visivo





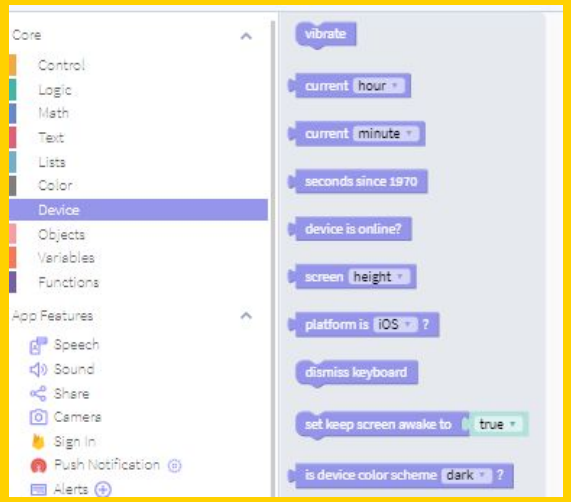
Proviamo la funzione con istruzione per feedback visivo

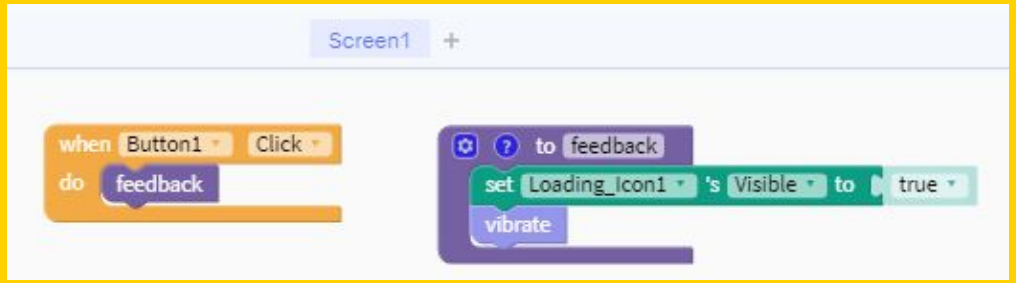
Se proviamo la app ci accorgiamo che cliccando il pulsante comparirà su schermo l'icona di caricamento.



Aggiungiamo le istruzioni alla funzione - feedback tattile

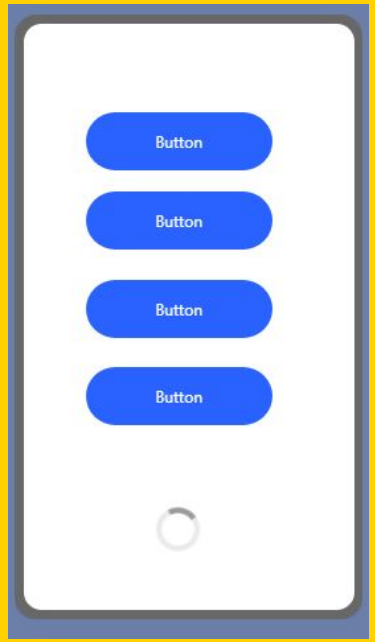
Andiamo tra i blocchi del dispositivo (Device) e trasciniamo il blocco “vibrate” all’interno della funzione.





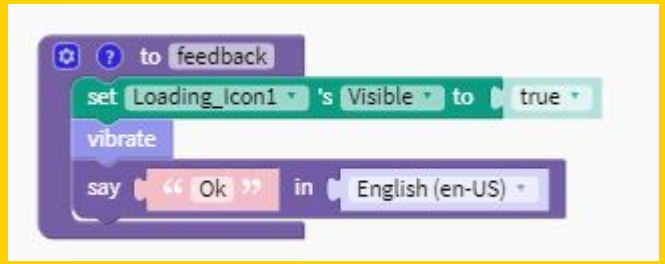
Proviamo la funzione a cui è stato aggiunto feedback tattile

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) .



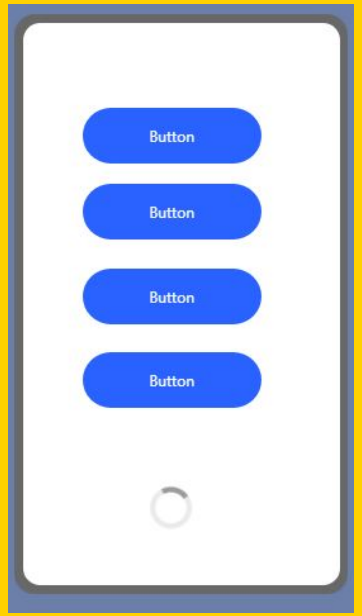
Aggiungiamo le istruzioni alla funzione - feedback audio

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



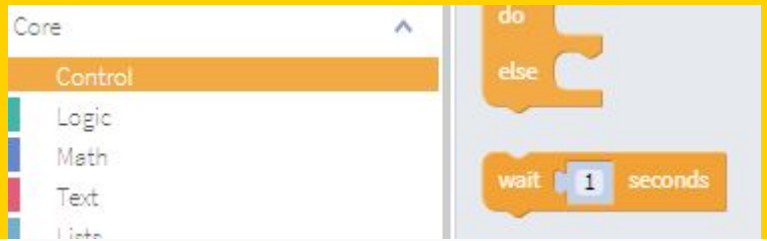
Proviamo la funzione a cui è stato aggiunto feedback audio

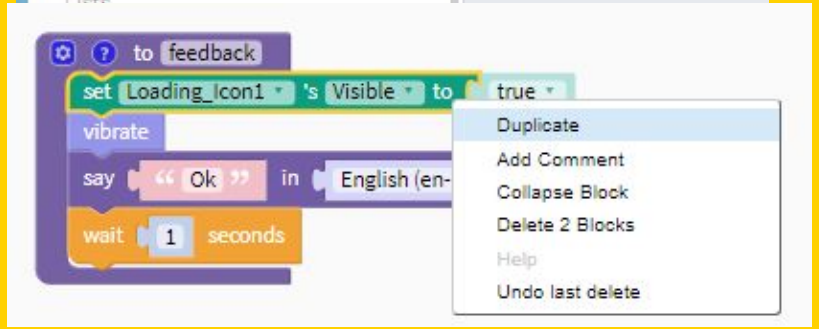
Adesso la funzione è completa, cliccando il pulsante comparirà su schermo l'icona di caricamento (feedback visivo), il dispositivo emetterà una vibrazione (feedback tattile) e verrà riprodotto il messaggio vocale "Ok" (feedback audio).

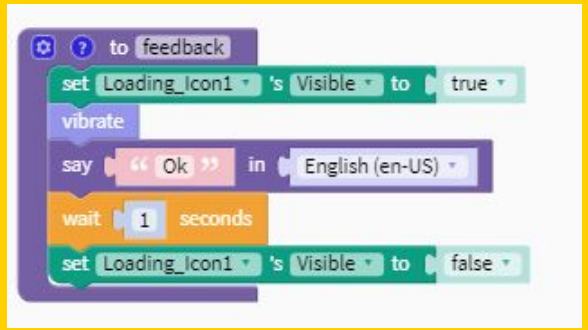


Per concludere la funzione..

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.

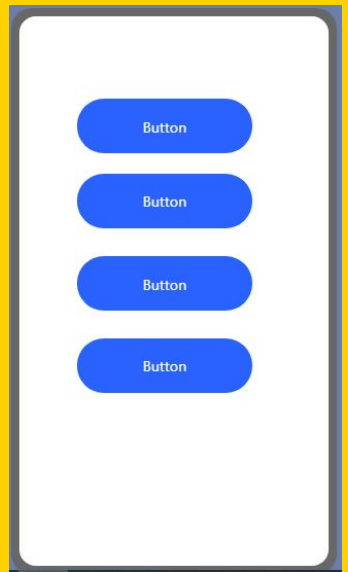






Proviamo la funzione a cui è stata aggiunta una pausa prima di disattivare la loading icon

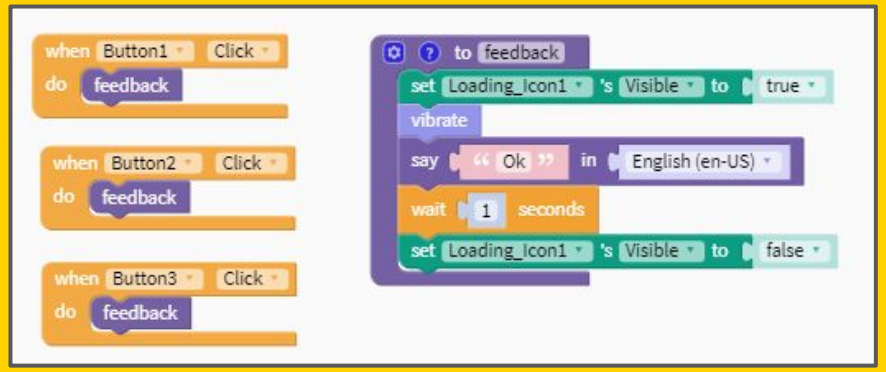
Adesso la loading icon scompare una volta concluse tutte le istruzioni tattili e audio.



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

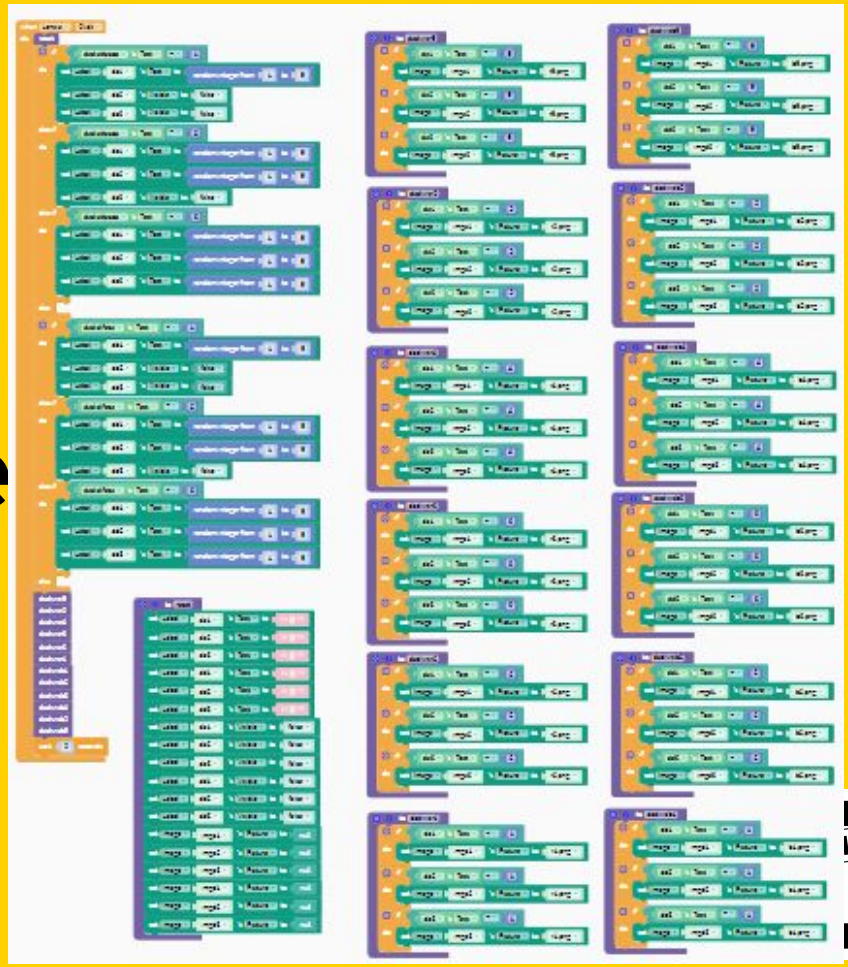
Esempi pratici che mostrano perchè conviene usare le funzioni 1



Esempi pratici che mostrano perchè dobbiamo usare le funzioni 2

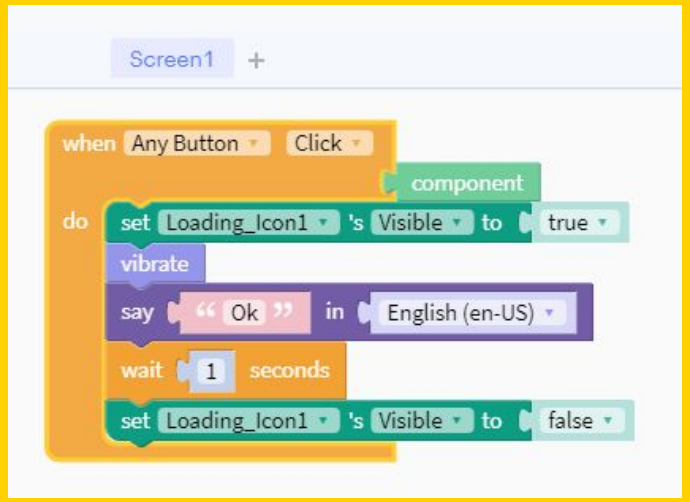
.. ma soprattutto se dovessimo apportare delle modifiche alla nostra app basterebbe lavorare sulla singola funzione e le modifiche si ripercuoterebbero su tutti i blocchi in cui la abbiamo richiamata.
Immaginate invece se dovessimo andare a cercare tutti i punti in cui abbiamo utilizzato quelle stesse 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!

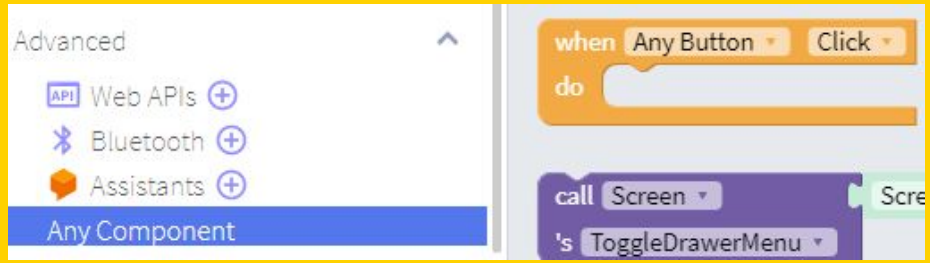
Esempi pratici che mostrano perchè dobbiamo usare le funzioni 2

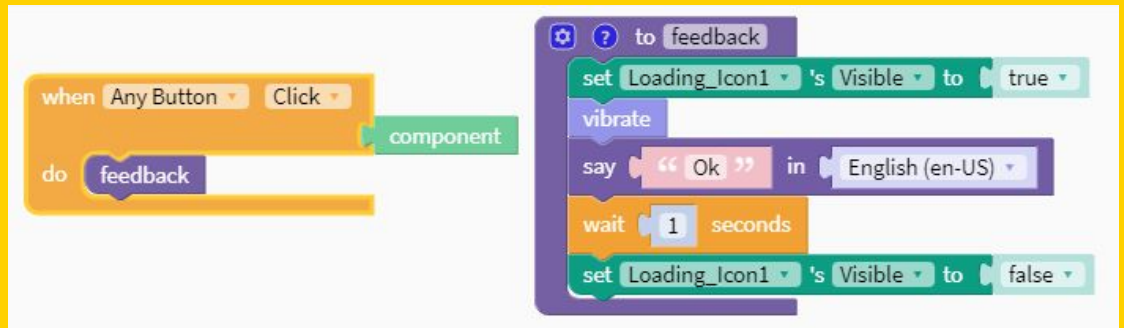


E se la funzione si fosse dovuta applicare a tutti i pulsanti presenti nella UI anzichè solo ad alcuni?

bisogna usare il blocco “Any Button” che esegue le istruzioni su tutti i pulsanti.



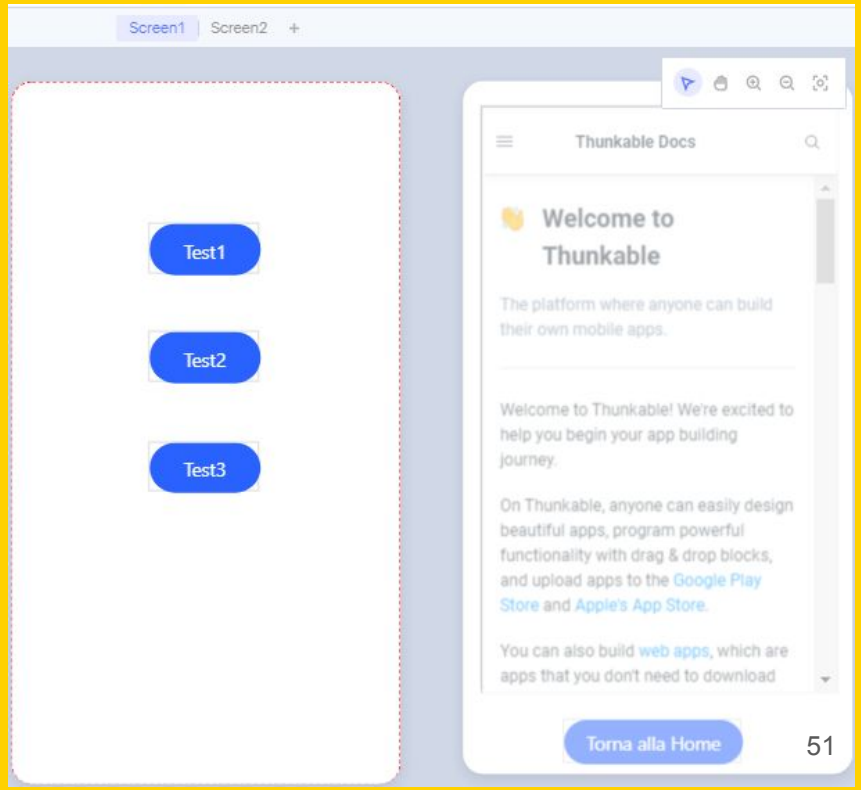




Utilizziamo le variabili in Thunkable

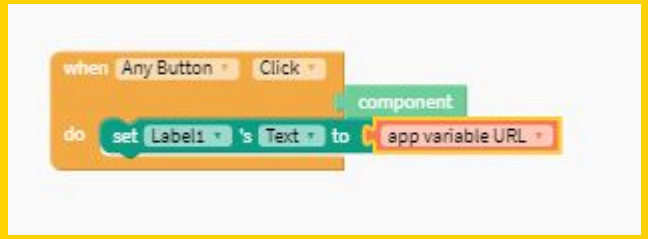
Esempio di una app che ci consente di visualizzare siti web con url predefiniti
Nella pagina di design:

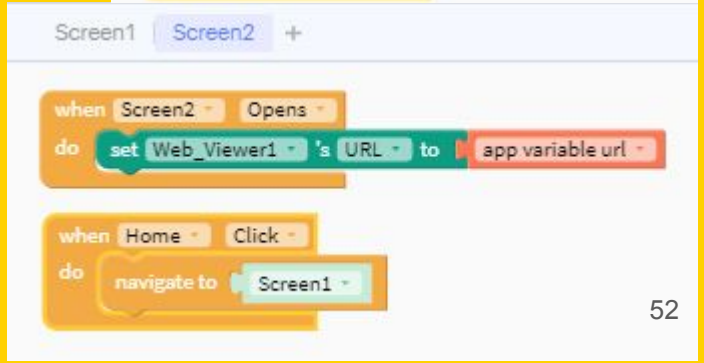
1. creiamo due schermate
2. Nella prima inseriamo tre pulsanti
3. Nella seconda inseriamo un pulsante e un componente web viewer



Variabili in Thunkable

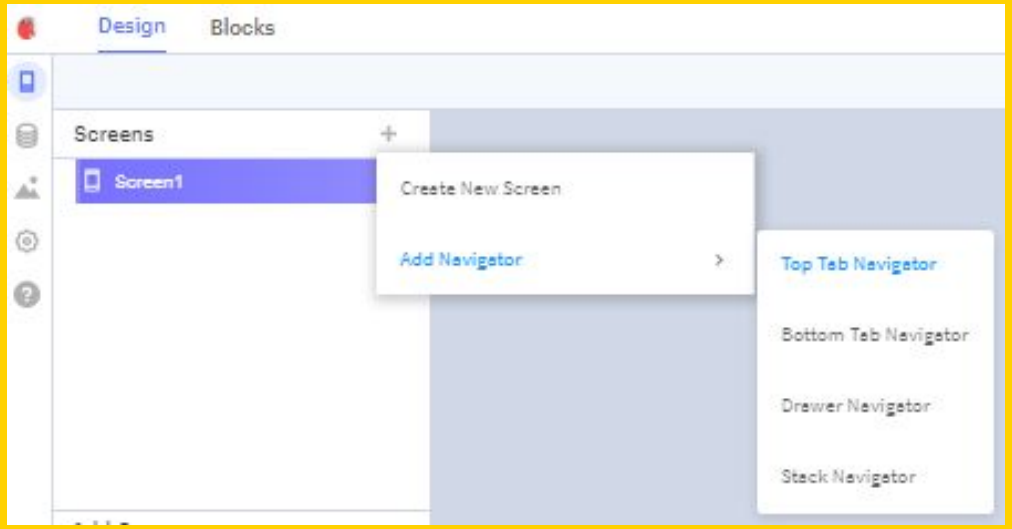
Nella pagina dei blocchi di Screen1 inizializziamo la variabile url e attiviamo i blocchi eventi dei tre pulsanti associando un url diverso a ciascun pulsante.
Nella pagina dei blocchi di Screen2 attiviamo il blocco eventi del visualizzatore web associando al suo attributo URL la variabile url.





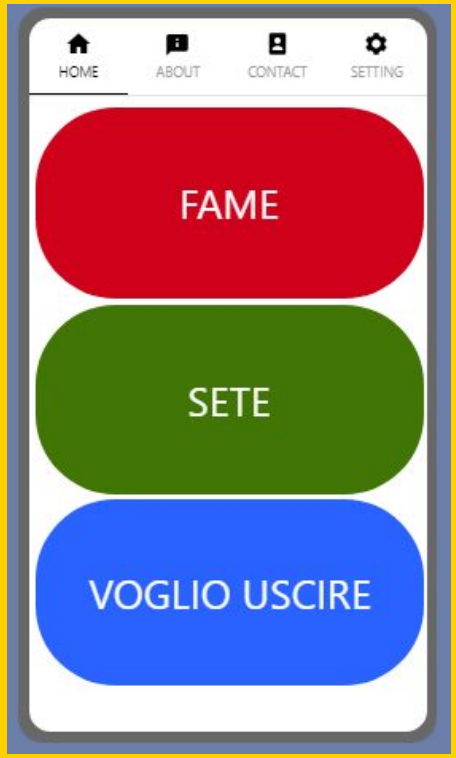
Sviluppiamo una app per anziani

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.



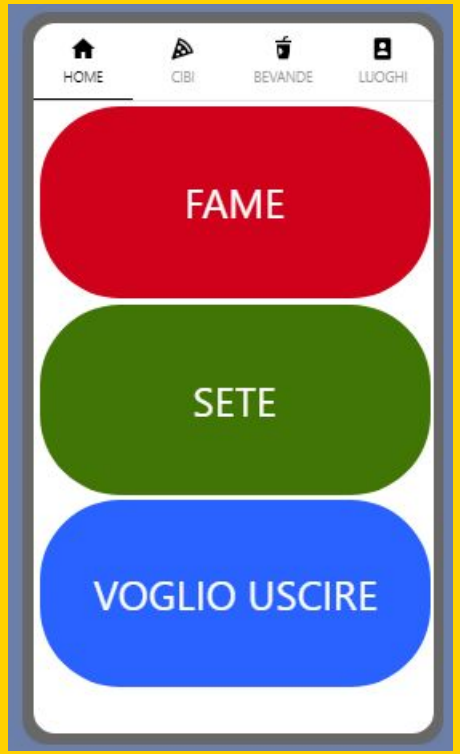
App per anziani

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.



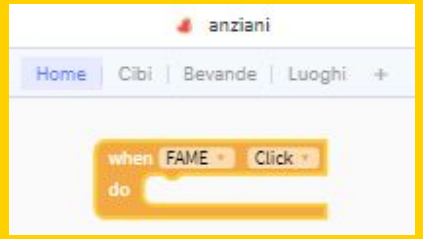
App per anziani

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



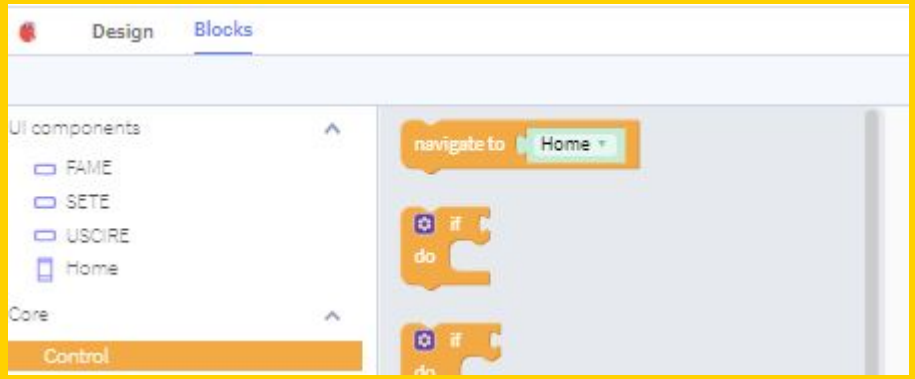
App per anziani

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



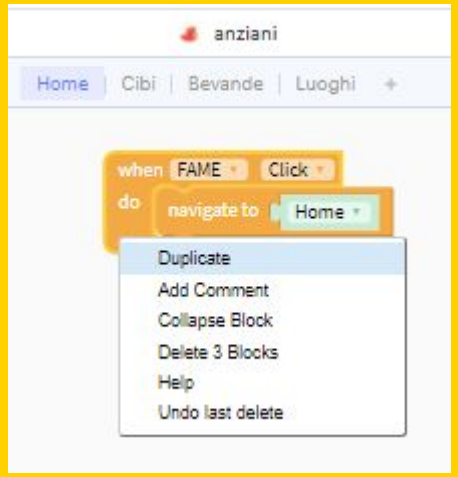
App per anziani

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



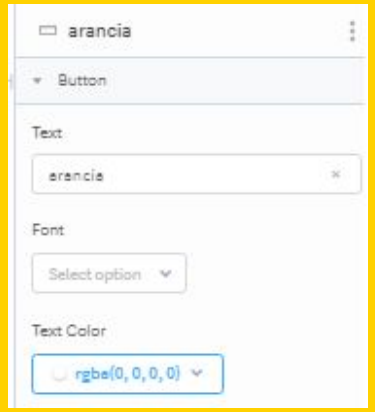
App per anziani

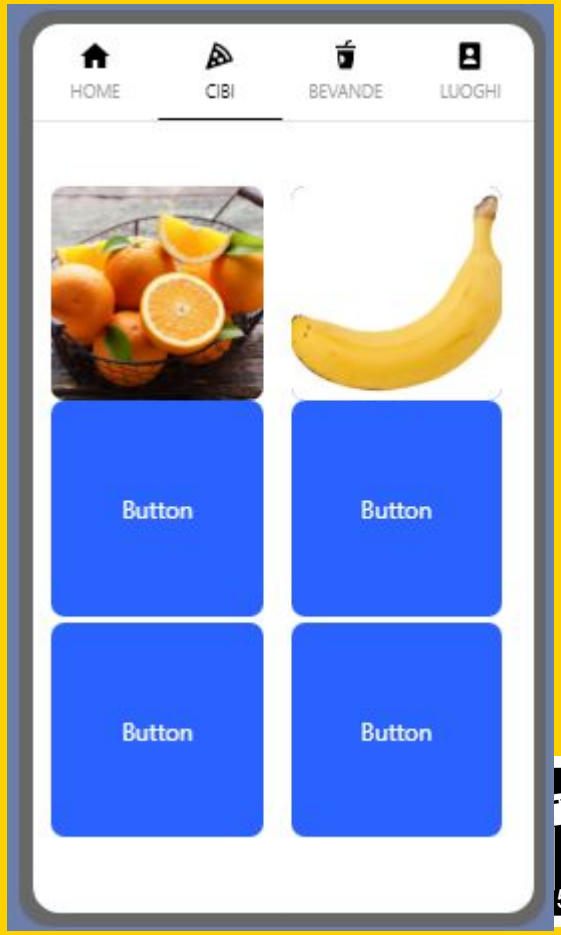
Duplichiamo il blocco per due volte cliccando col tasto destro del mouse sul blocco e scegliendo quindi Duplicate. Andiamo ora a modificare i riferimenti dei blocchi associandoli correttamente.



App per anziani

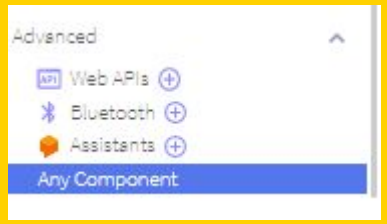
Torniamo sul design e nella schermata cibi andiamo ad inserire dei pulsanti a cui associamo i nomi di alimenti e sostituiamo lo sfondo con delle immagini associate al nome (ricordiamoci di impostare il colore del testo trasparente).

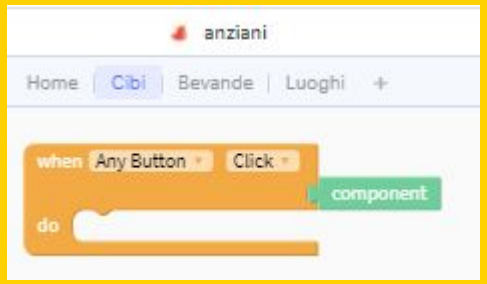


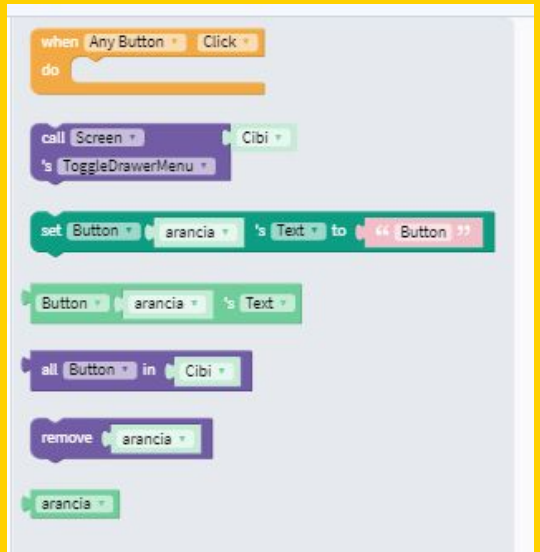


App per anziani

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





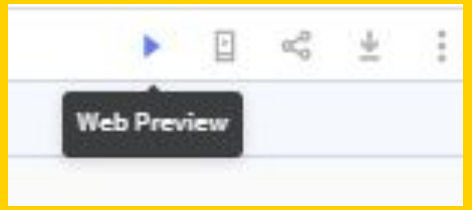


App per anziani

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.

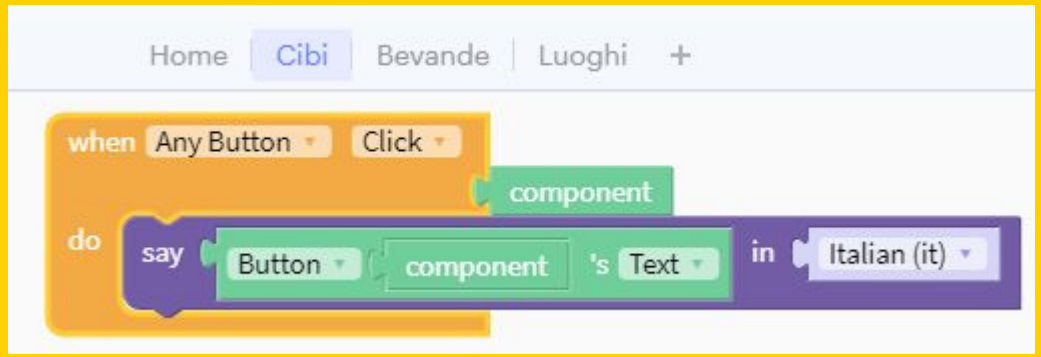


App per anziani

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

App per anziani

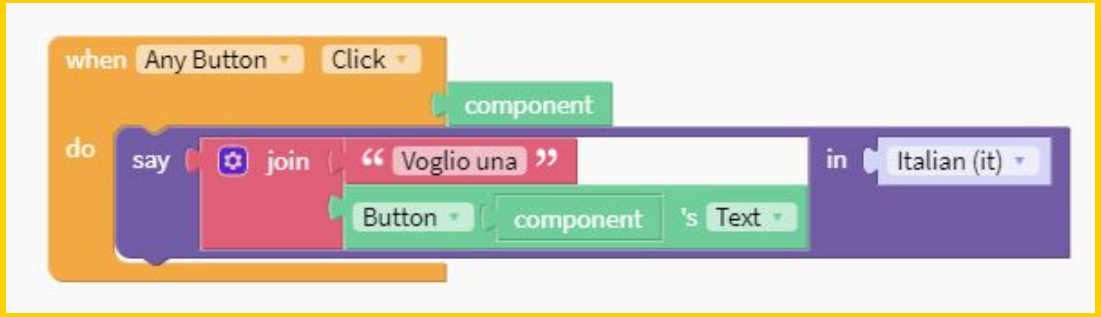
Selezioniamo il blocco “component” e trasciniamolo al posto del blocco “arancia”.



App per anziani

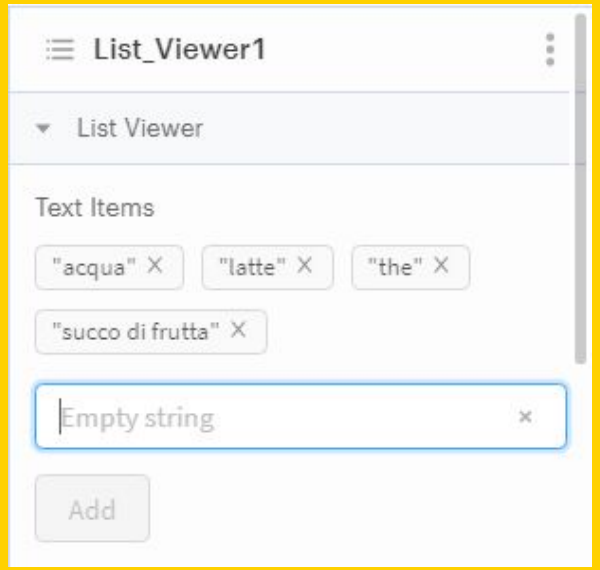
Se eseguiamo la app, adesso ciascun pulsante riproduce il testo associato. Utilizzando i blocchi con funzioni testuali possiamo migliorare la frase che viene letta; ad esempio, usando il blocco join, il sistema pronuncerà “voglio una arancia” anzichè dire solo “arancia”. Il testo statico impostato nella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

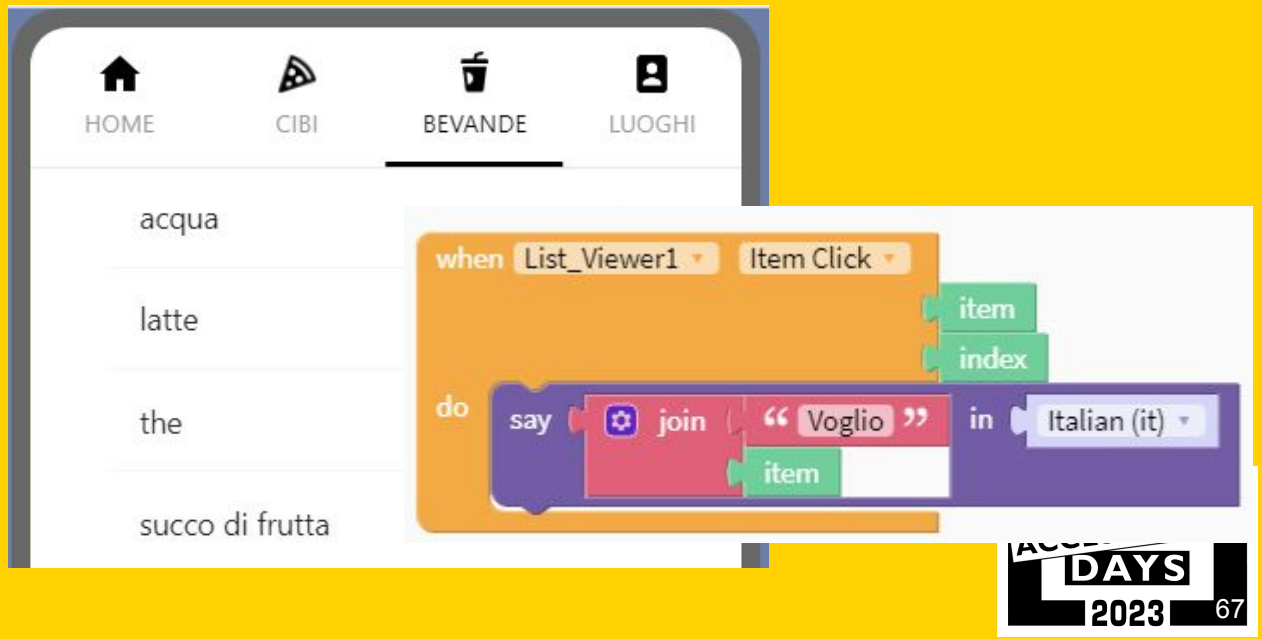
App per anziani



App per anziani

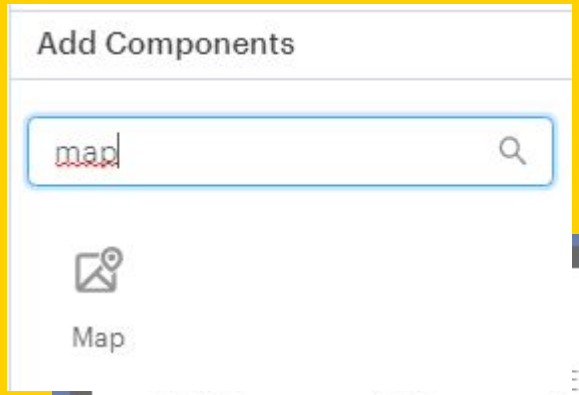
Possiamo fare altrettanto con la pagina delle bevande: anzichè usare i pulsanti con le immagini inseriamo un semplice elenco di elementi tra cui scegliere.

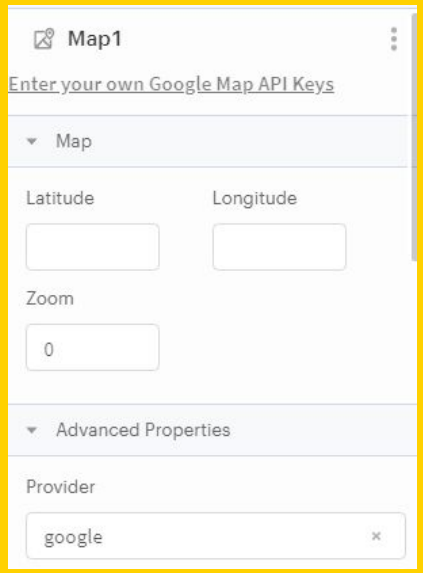


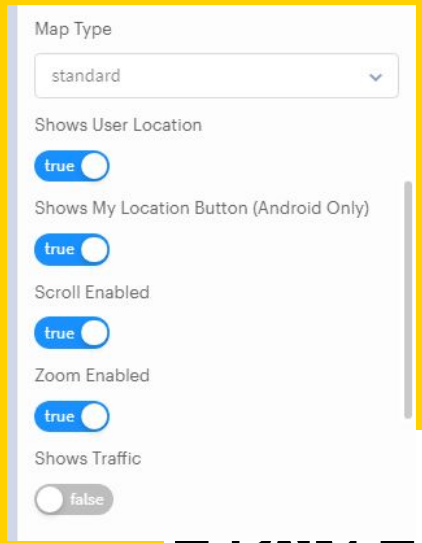


App per anziani - location

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

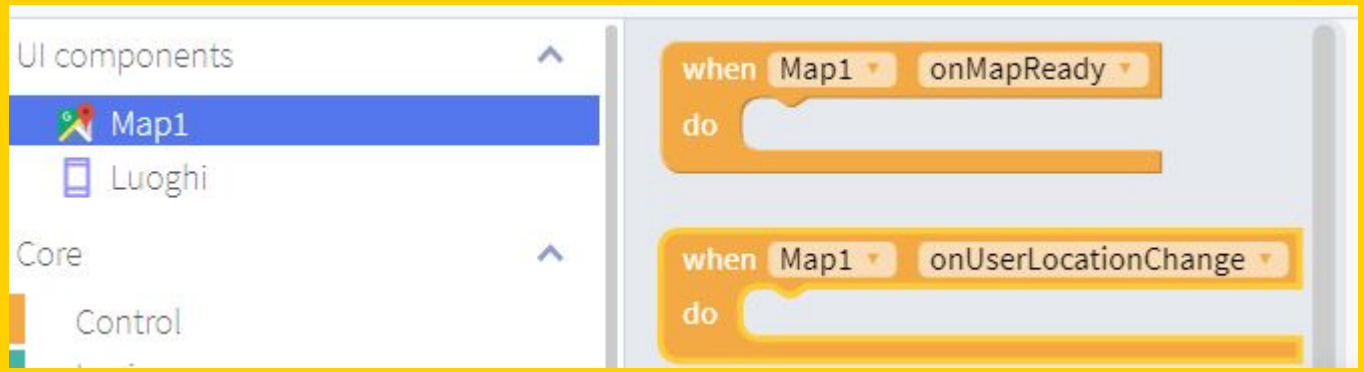






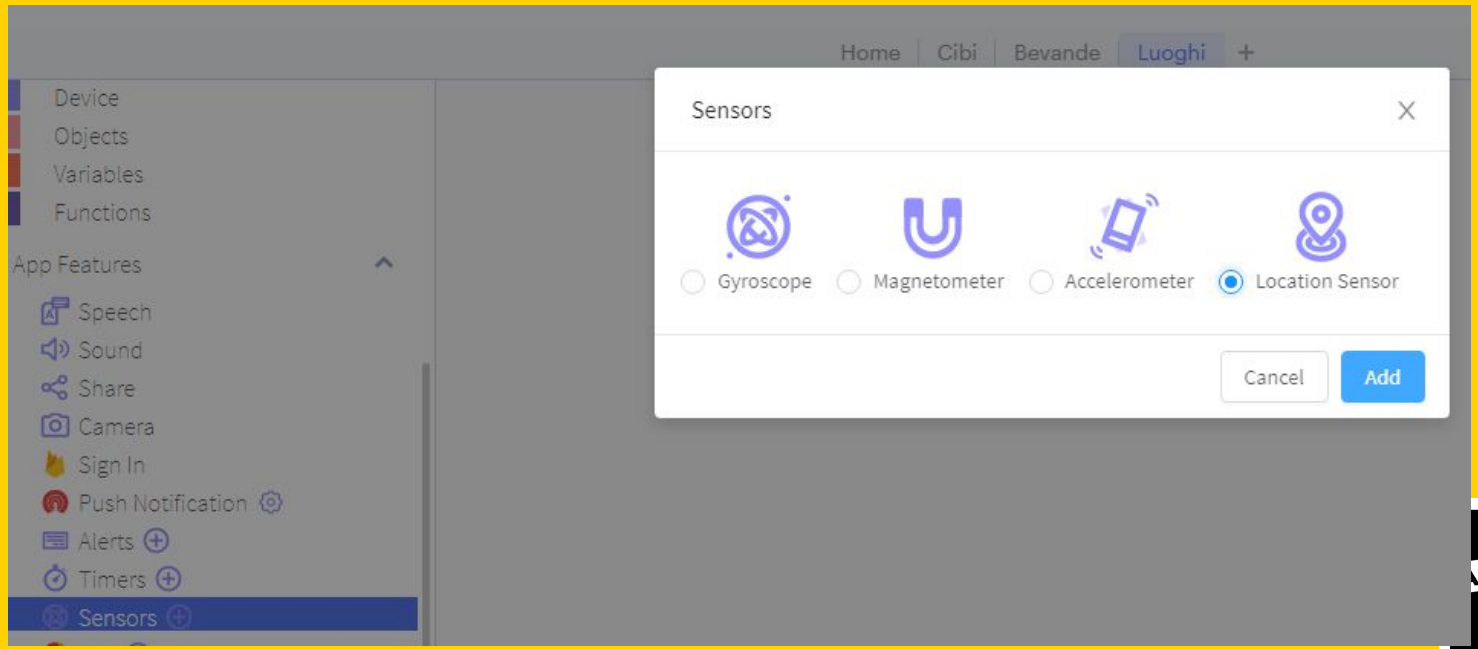
Blocco Mappa

Andiamo a trascinare nel desk un blocco evento associato alla mappa



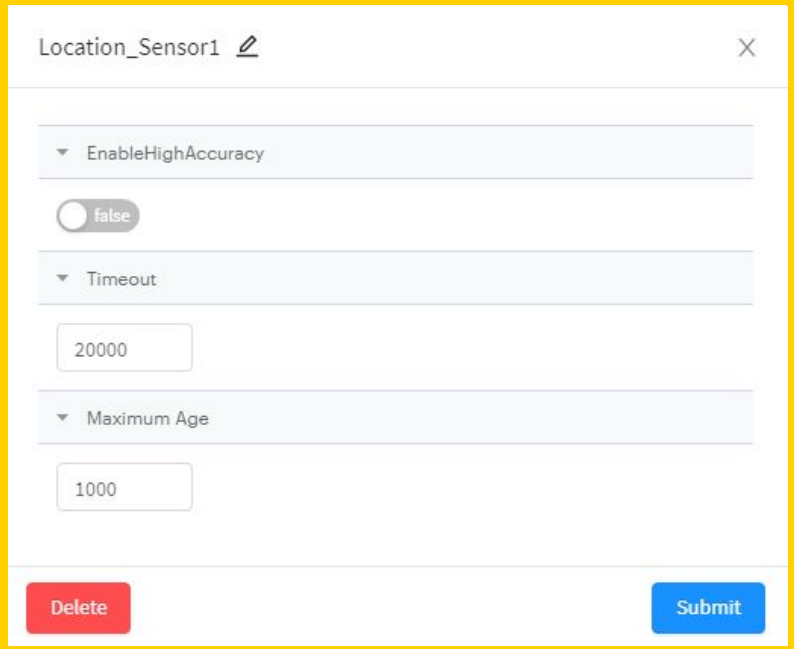
Sensore di posizione

Inizializziamo il location sensor (gps del dispositivo)



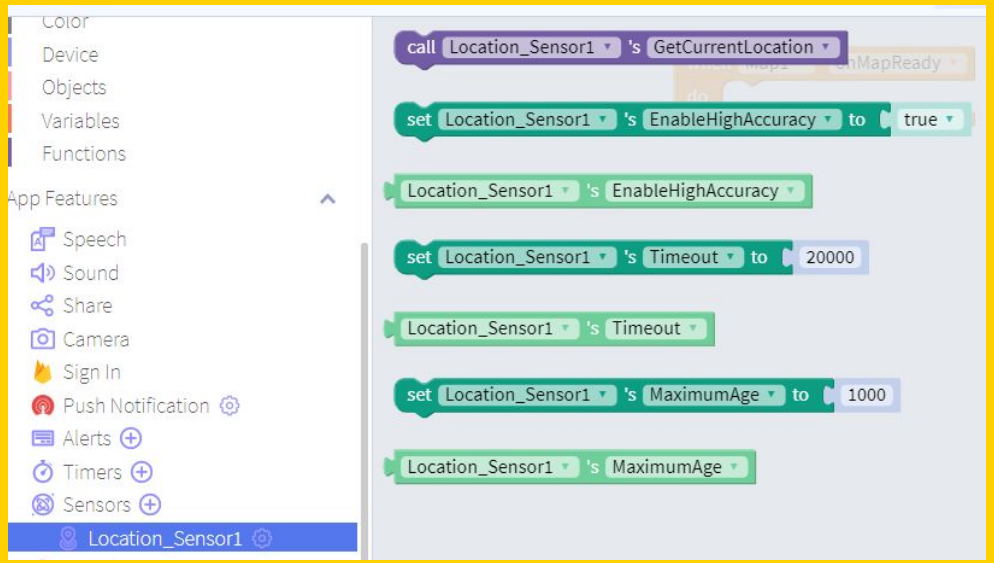
Sensore di posizione - parametri

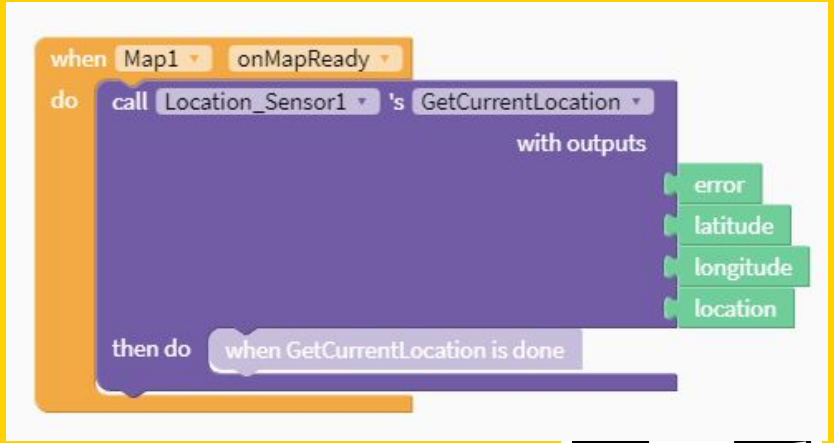
Impostiamolo con i parametri di default



Chiamata del sensore di posizione nel blocco mappa

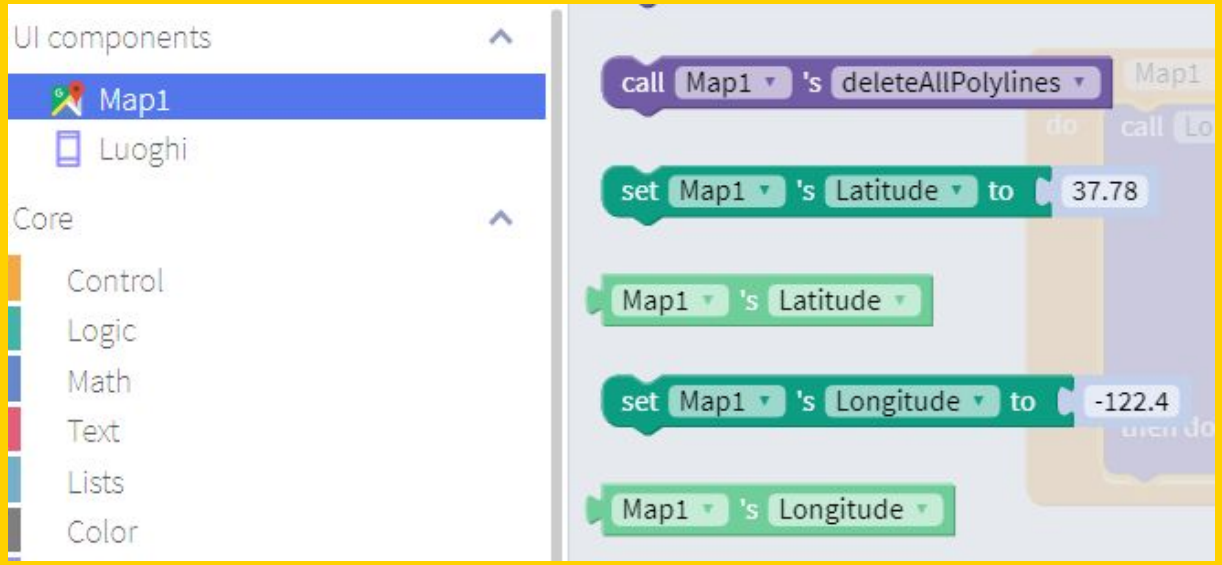
Ora trasciniamo la chiamata del sensore di posizione all'interno del blocco evento. Il sensore di posizione fornisce 4 output: errore, latitudine, longitudine e posizione.





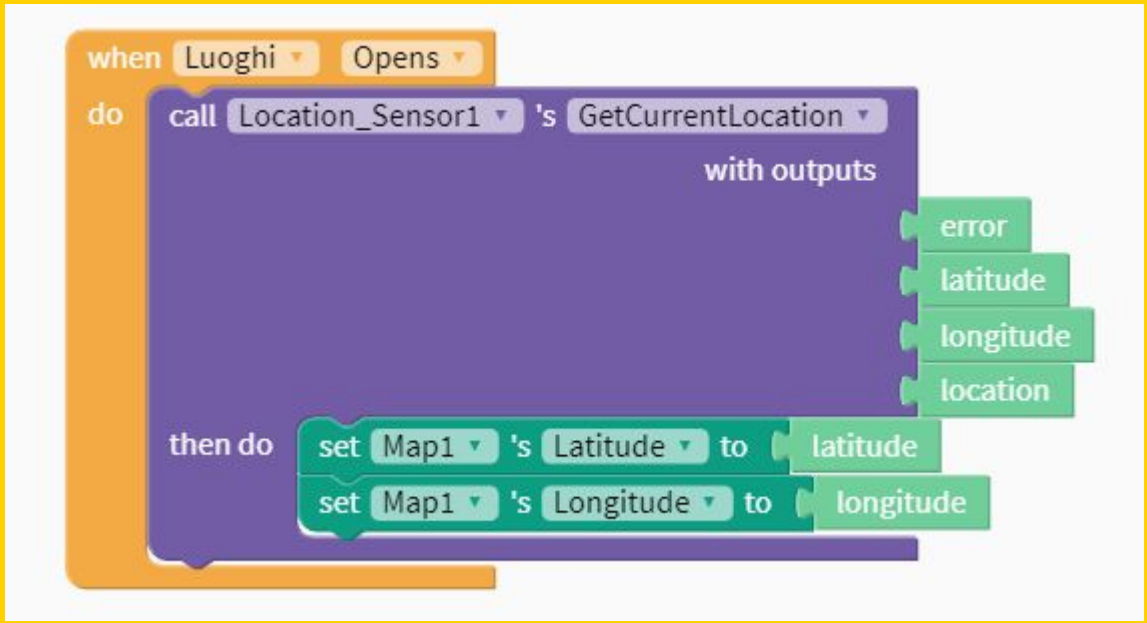
Sensore di posizione

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)



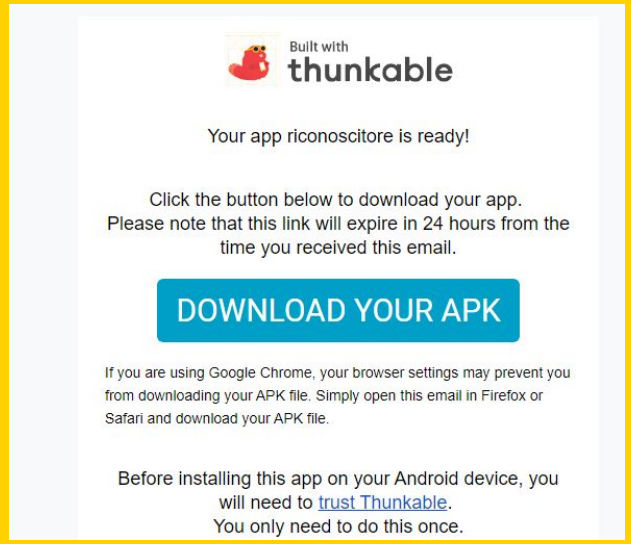
Mappa funzionante in app

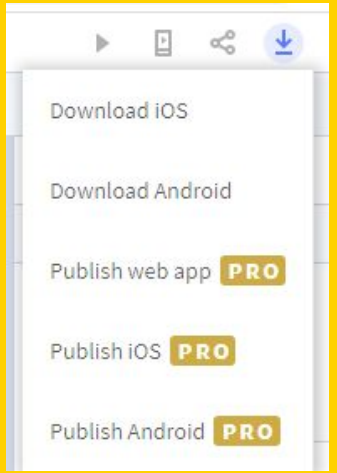
Proviamo la app e vediamo che viene mostrato un pallino blu sulla mappa in corrispondenza della nostra posizione (coordinate gps di latitudine e longitudine). Componendo altri blocchi è possibile creare delle regole o impostare su mappa un percorso o dei punti di interesse.

Download della app su smartphone Android o iOS

Potete ricevere la vostra app via email per condividerla con altri dispositivi.

La versione gratuita è limitata a 2 progetti scaricati al mese.





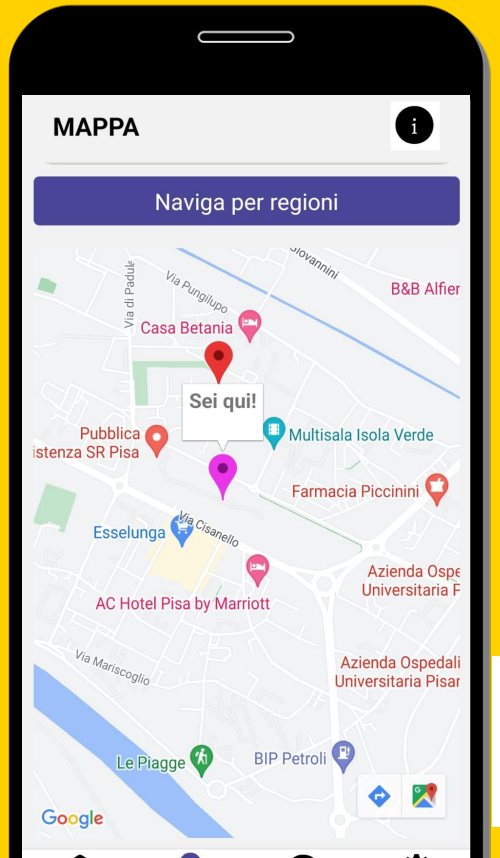
Mappa accessibile

1. Sensore di localizzazione
2. Menù accessibile
3. Aggiungere i marcatori (POI)
4. Feedback vocale (TTS)
5. Feedback tattile (vibrazione)

ACCESSIBILITY DAYS 2023

Sensore di localizzazione

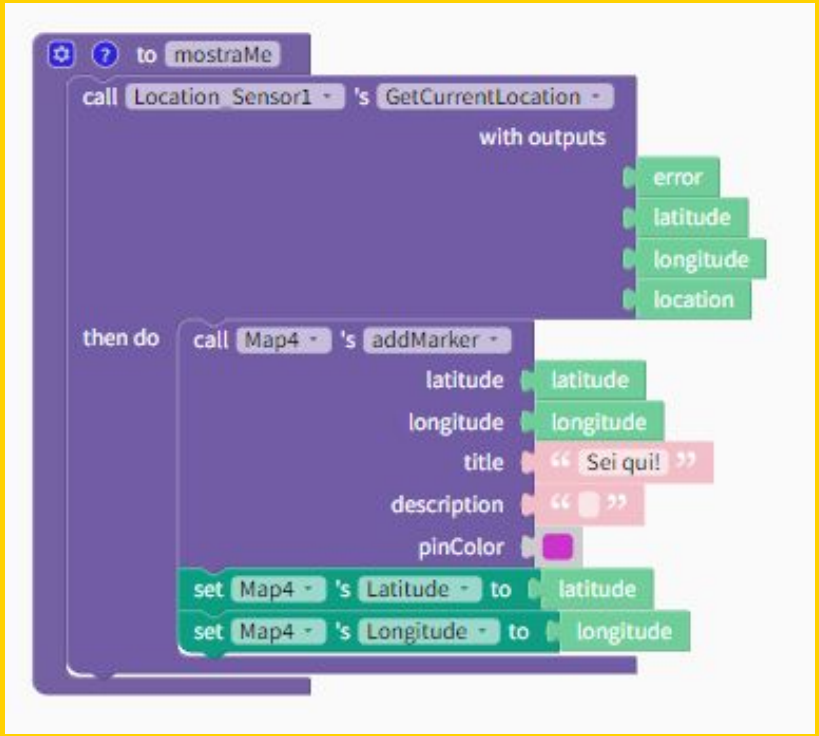
- Restituisce la posizione di un utente tramite l’aggiunta di un marcatore del colore diverso dagli altri POI.
- Text to speech “Sei qui”



ACCESSIBILITY DAYS 2023

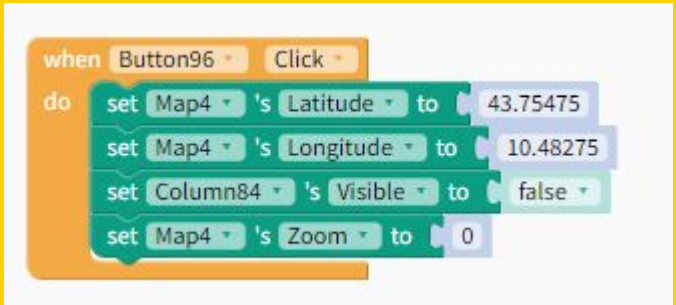
Sensore di localizzazione (2)

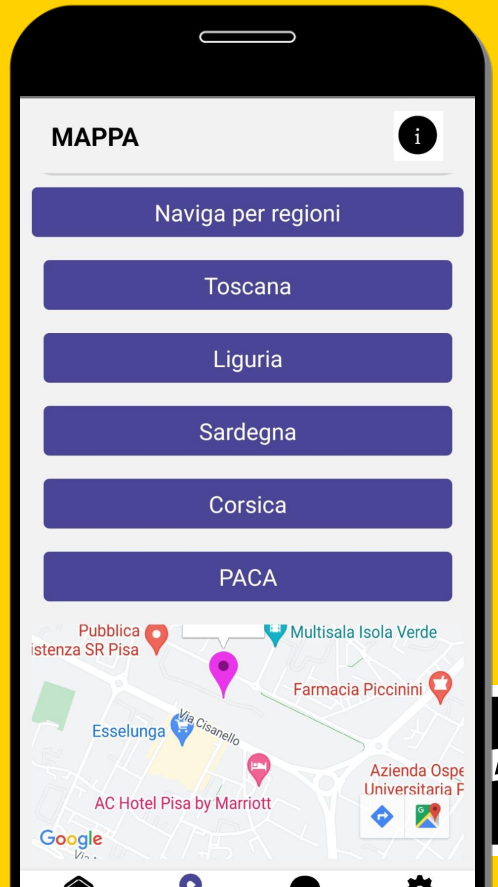
Trascinare *Locations Sensor* all'interno dell'evento, aggiungere *addMarker* per visualizzare il marcatore. Infine settare la longitudine e latitudine.



Menù accessibile

- Lista regioni
- Zoom sulla regione scelta

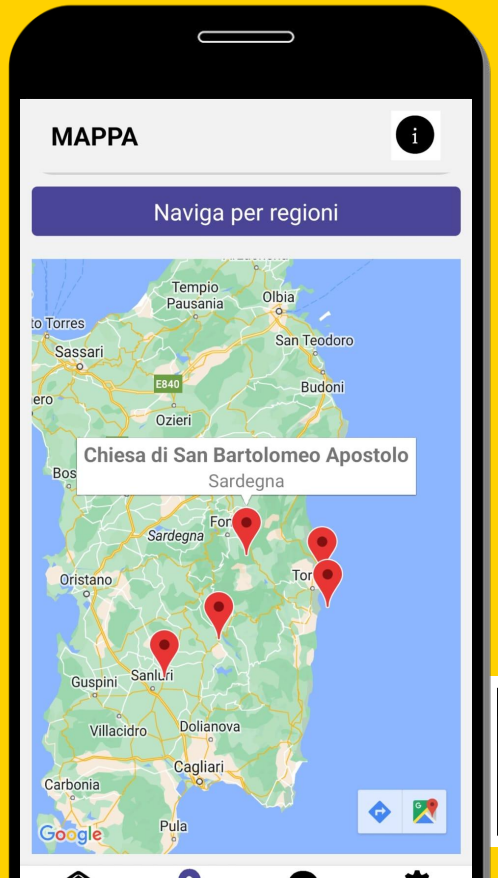




ACCESSIBILITY DAYS 2023

Marcatori

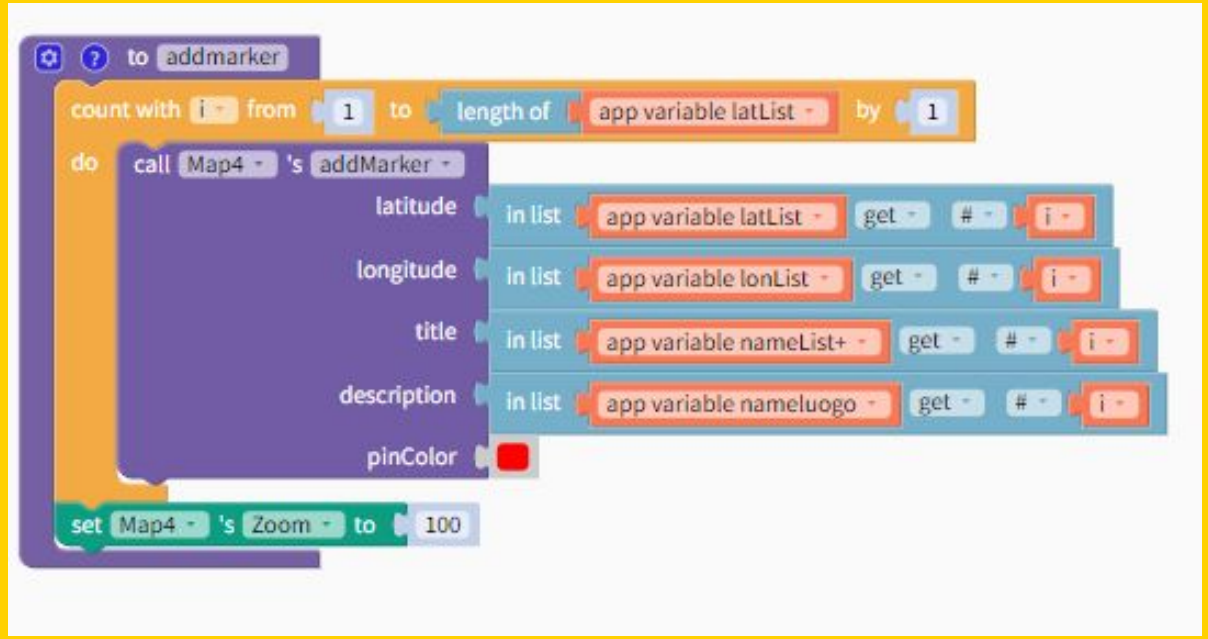
Selezionando, per esempio, Sardegna dal menù, la mappa si sposta sulle coordinate impostate nel codice mostrando i POI presenti in quel luogo.



ACCESSIBILITY DAYS 2023

Marcatori (2)

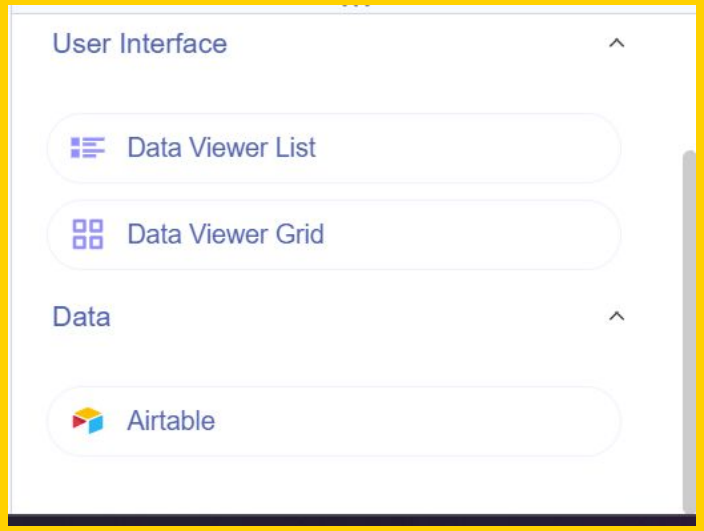
Dal blocco *Control* selezionare l'evento *count with* che ha l'obiettivo di ripetere un'azione per un certo numero di volte, con l'indice di incremento i. In questo caso, con l'aggiunta di *addMarker,* si ripete l'aggiunta dei marcatori in base alla regione scelta.



ACCESSIBILITY DAYS 2023

Airtable

E’ un foglio di calcolo che permette di creare tabelle di dati in vari formati.



	Luogo	Lat	Lon
1	Toscana	43.75475	10.48275
2	Toscana	43.69620	10.42868
3	Toscana	43.72744	10.51904
4	Toscana	43.71201	10.43160
5	Toscana	43.74928	10.50178
6	Liguria	44.10772	9.86808
7	Liguria	44.07912	9.92301
8	Liguria	44.11117	9.82756

ACCESSIBILITY DAYS 2023

Airtable (2)

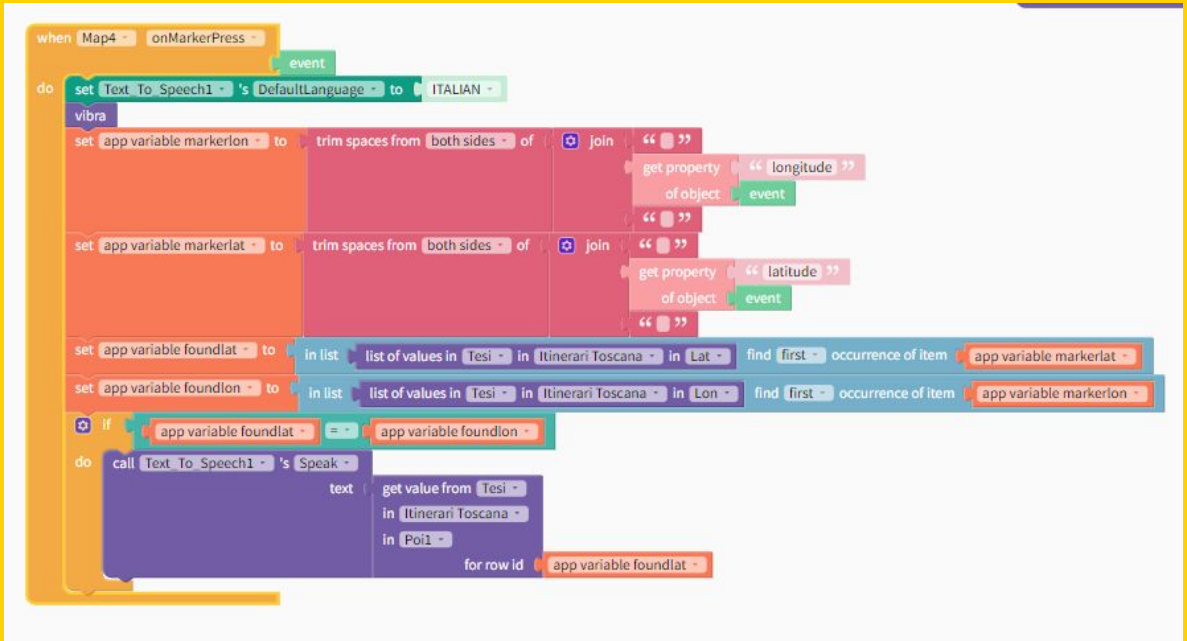
Inizializzare le variabili.

Settare ogni variabile nella lista delle informazioni prese dal database.

ACCESSIBILITY DAYS 2023

Text to speech

Per ogni POI è stato inserito il titolo che viene letto con il Text to speech ad ogni click. Questo procedimento è stato possibile grazie alla creazione di quattro variabili all'interno del codice. Le variabili sono due per trovare le coordinate all'interno del database (latitudine e longitudine) e due che richiamano le coordinate per ogni marcatore.

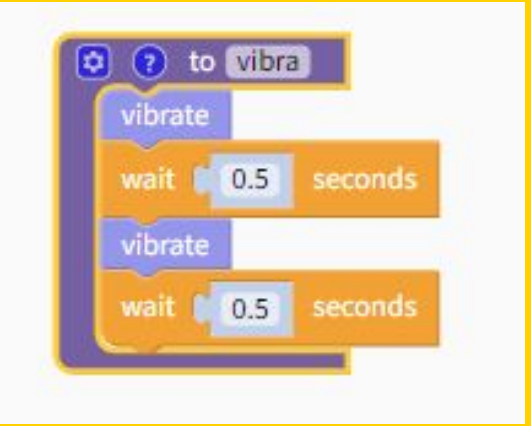


ACCESSIBILITY DAYS 2023

Vibrazione

Quando un marcatore viene selezionato, avviene una doppia vibrazione con 0.5 secondi di pausa. Un modo per far capire ai non vedenti che si sta selezionando un POI e non un bottone o altro.

Questo viene anche spiegato nella descrizione dell'utilizzo della mappa.



ACCESSIBILITY DAYS 2023

Video

ACCESSIBILITY DAYS 2023

Grazie per l’attenzione!