

# TRIS ONLINE

## IL PROGETTO

Questa applicazione vuole implementare il gioco del tris (o tic tac toe in inglese).

Vi sono due modalità di gioco:

- Un gioco locale che sfrutta un dispositivo android
- Un gioco multiplayer che sfrutta due device android connessi tra loro

## Gioco locale

Questa parte è molto semplice e si alterna tra tre schermate diverse tra loro.

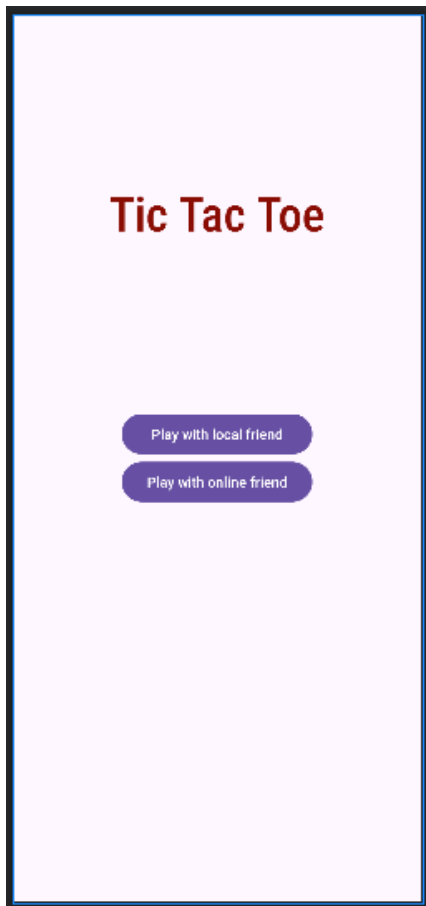
Innanzitutto si comincia da una schermata di menù in cui l'utente può scegliere quale tra le due modalità giocare.

Una volta scelta l'attività, in questo caso ci si concentra sulla parte locale, si passa alla schermata che viene chiamata di Login. Essa chiede la semplice e facoltativa immissione dei nomi dei player. Se nei due campi non viene immesso nulla allora l'applicazione ricaverà dei nomi di default per distinguere tra i due (*PLAYER1* e *PLAYER2*).

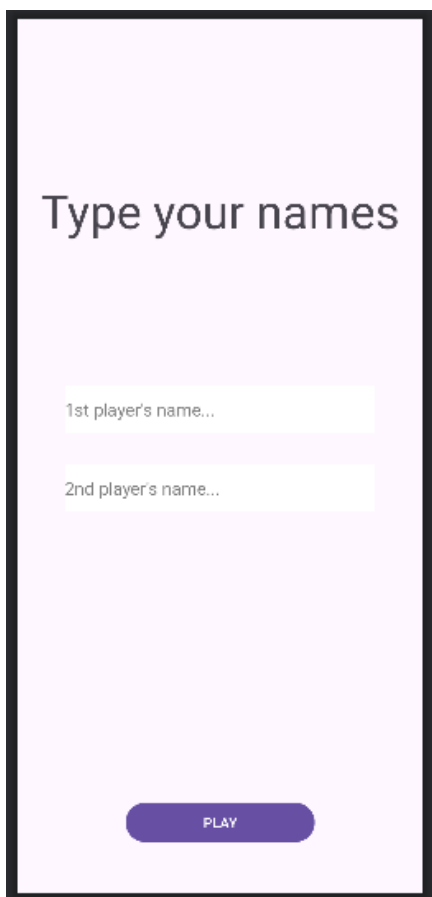
Premendo quindi sul tasto play si arriva quindi alla schermata di gioco. Qui sono stati inseriti innumerevoli argomenti che verranno spiegati in quanto si troveranno poi anche nella schermata di gioco nella modalità online.

- I nomi dei due giocatori con al di sotto segnato il turno, come si vede in foto se ad es. *PLAYER1* ha già piazzato la sua X allora il turno è di *PLAYER2* e così via...
- Sotto ai turni si vede una barra colorata che si sposta a seconda del numero di vittorie. Se *PLAYER1* vince allora la barra si sposta di uno verso destra colorando più di blu che di rosso, viceversa se vince *PLAYER2* andrà verso sinistra.
- La tabella di gioco è composta da delle card colorate che, man mano che il gioco va avanti, si 'scoprono' lasciando il posto al simbolo lasciato dal giocatore. A seconda di chi è il turno il simbolo cambia da X a O o da O a X.
- Se i giocatori si stancano di giocare possono tornare alla home cliccando l'apposito tasto, lo scopo di questo gioco non era salvare dati e classifiche dei giocatori per cui una volta usciti dalla partita non ci sarà modo di ripristinarla.
- Se la partita finisce con nessuna vincita e la plancia di gioco tutta piena, quello che i giocatori dovranno fare se vogliono continuare è spingere l'apposito tasto di play again. Se invece la partita finisce con la vincita di uno dei due uscirà un pop-up che avverte chi ha vinto e chiede se si vuole giocare di nuovo.
- Spingendo il tasto back del telefono è comunque possibile tornare indietro in ogni stage del gioco tranne la schermata iniziale di menu.

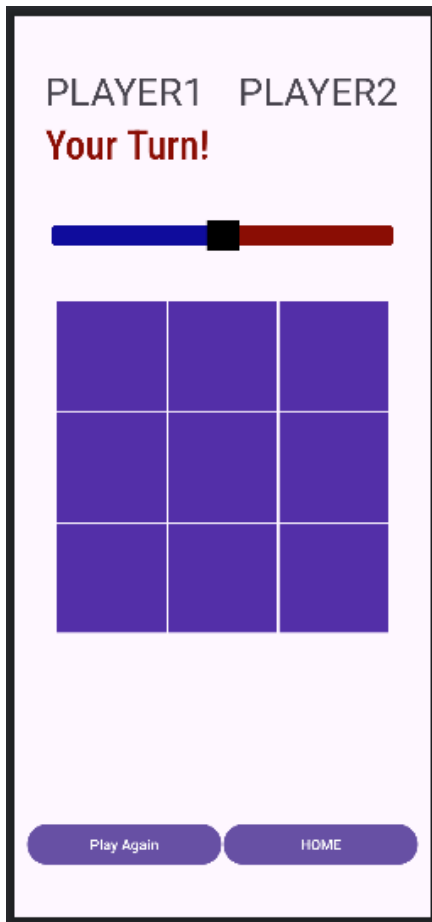
Vengono messe ora tutte le immagini di riferimento da un'attività esplorata all'altra:



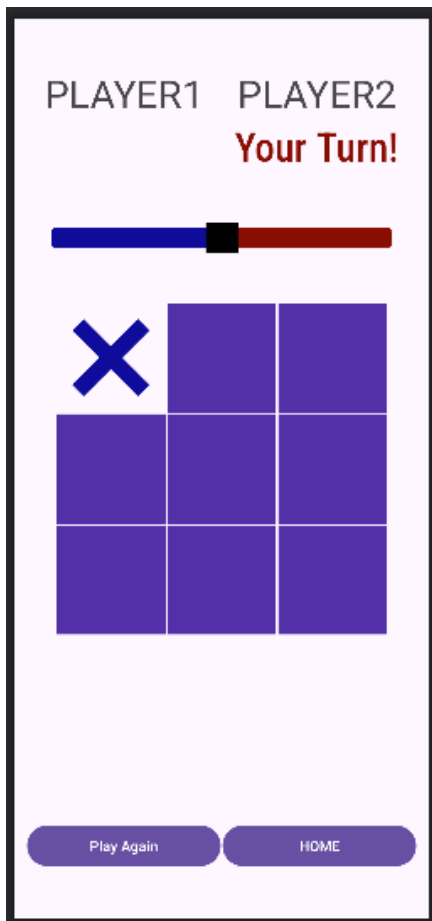
Schermata di menu in cui è possibile scegliere le due attività.



Schermata in cui devono essere inseriti i nomi e da cui poi partirà il gioco.



Schermata iniziale di gioco in cui non è successo ancora nulla.



Schermata in cui il primo giocatore (PLAYER1) ha fatto il suo turno posizionando la X nel primo quadrato in alto a sinistra. Si noti che la barra delle vincite non è variata in quanto hanno appena cominciato la partita.

## Gioco online

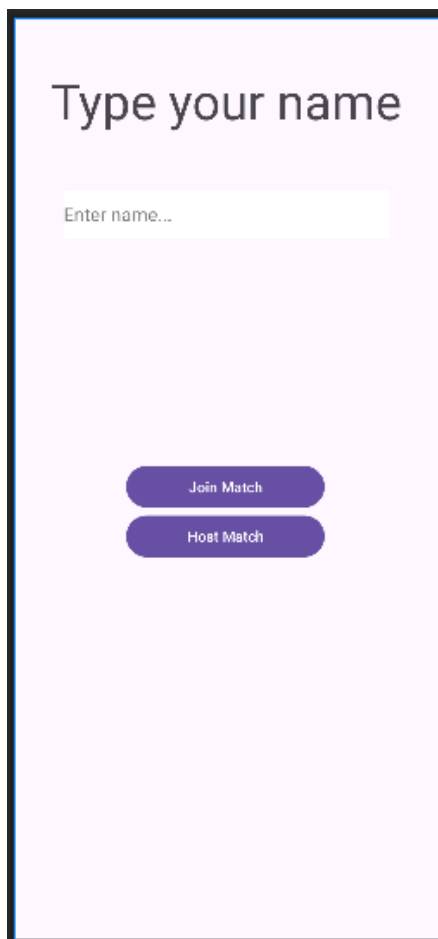
Nel gioco online le schermate sono visivamente le stesse, tranne il form di immissione del nome.

Quello che però succede per 'attivare' questa modalità v  spiegato. Scegliendo e quindi cliccando il tasto per giocare online viene informato l'utente che verranno usati Bluetooth e GPS. Per giocare, accettare ed abilitare questi due servizi   mandatorio, la connessione avviene tramite bluetooth e per poter scoprire device vicini   necessaria anche la posizione.

Dopo aver abilitato entrambi si pu  passare quindi alla schermata di inserimento del nome. Questa schermata funziona in modo simile a quella vista per il gioco locale ma ha bisogno di un singolo nome (device usato sta a un nome giocatore). Anche in questo caso l'utente pu  scegliere se metterne uno o meno, se non verr  inserito nulla ci sar  un nome di default.

Ci ritroviamo ora con due device che devono connettersi: per poterlo fare devono uno 'hostare' la partita e l'altro 'joinarla', In realt  i due termini vengono usati impropriamente ma si spiegher  cosa ci sta sotto a questa scelta nella parte di architettura.

Una volta che i dispositivi si trovano a vicenda la partita comincia con le stesse regole per il gioco locale. In realt  in questo caso alcune parti della logica sono state cambiate per il fatto che nel frattempo deve avvenire una comunicazione delle mosse tra i due device.



Schermata di immissione del nome.

# ARCHITETTURA

In questa parte viene discussa l'architettura e le funzioni usate nella parte di Gioco online. Verranno quindi coperte le azioni da quando i device spingono 'host' o 'join' in modo da concentrarsi sulla parte di comunicazione e connessione BluetoothSocket.

Viene creata una classe (BluetoothConnection) con al suo interno tutti i thread per il lato server, il lato client e la gestione comunicazione.

Intanto va detto che si può parlare di connessione bluetooth solo quando i due device sono già accoppiati, quindi per poterli connettere si devono accoppiare. Inoltre va detto che per creare un lato server bisogna dare un nome univoco per poter essere trovati come server. Si è provato ad usare una password di gioco ma purtroppo l'UUID per usare il protocollo RFCOMM deve essere statico e random o creato a compile time non funziona. Si è quindi deciso di scegliere un nome adatto all'applicazione, cioè "TRISONLINE".

Si apre quindi, da entrambi i device, la parte server con la funzione creata appositamente startServer() che avvia a sua volta l'AcceptThread(). Sopra questo thread si fa girare una chiamata bloccante per aspettare che venga accettato un device (in questo caso solo uno in quanto abbiamo un'applicazione che ha bisogno di un solo altro giocatore/device). Il server è un oggetto BluetoothServerSocket che, come vedremo, servirà solo per questa parte e verrà poi chiuso.

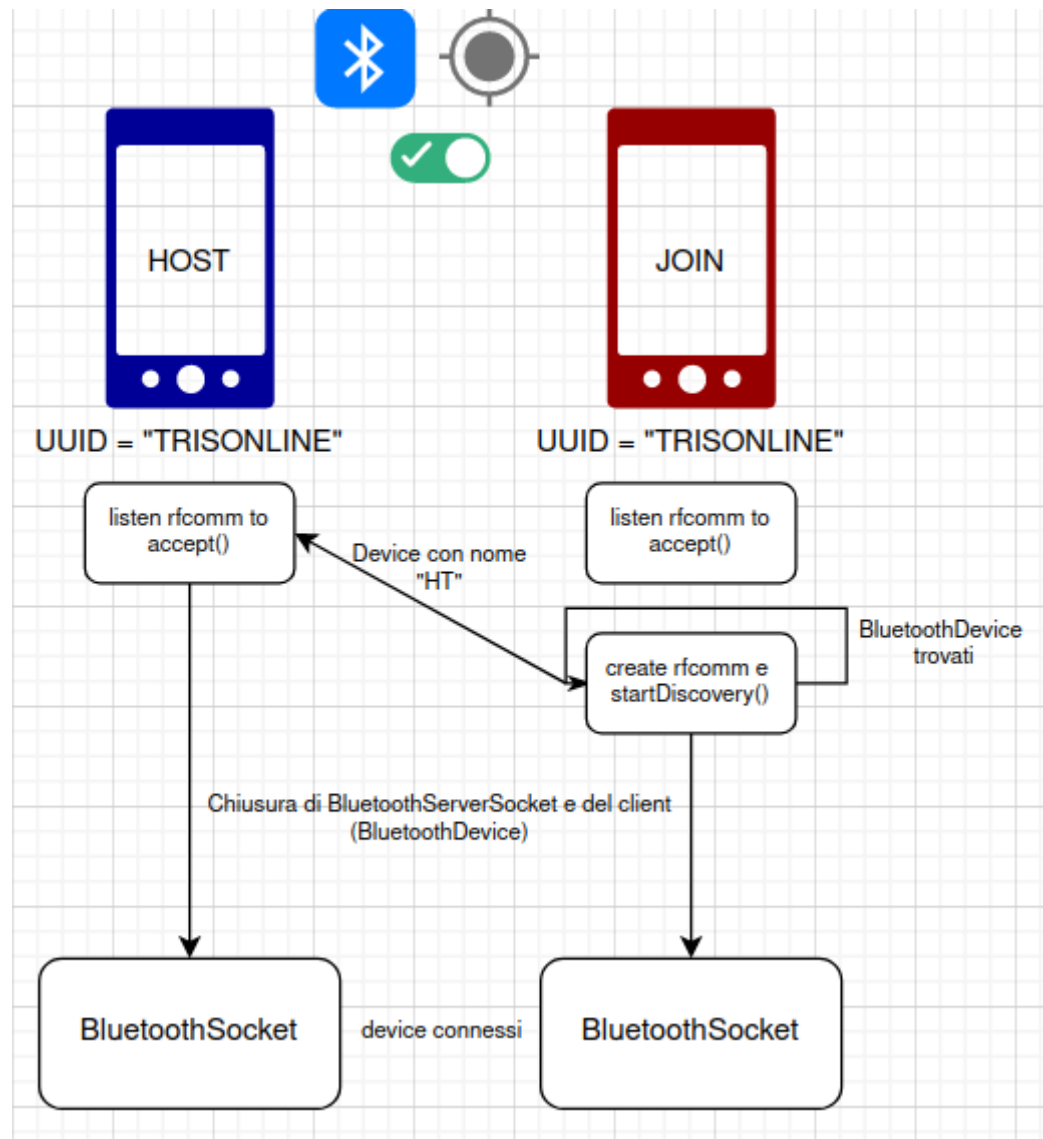
Dopo aver messo in ascolto i due dispositivi si chiama startClient() e startDiscovery() sul device che ha deciso di 'joinare'.

startClient crea una comunicazione RFCOMM con UUID sempre "TRISONLINE" e aspetta di essere accettato. Per poter però essere creato ha bisogno di un BluetoothDevice. Esso viene trovato e scoperto con startDiscovery() e un listener di eventi (BroadcastReceiver) che ci avvisa quando viene trovato un nuovo dispositivo. Se questo dispositivo si chiama "HT" allora chiederemo al server di essere accettati.

Una volta che i device si saranno trovati, si accetteranno e avverrà prima l'accoppiamento (nel caso in cui non fossero già paired) e successivamente la connessione.

Dopo essersi connessi entrambi i device chiudono le parti server e client e ottengono un BluetoothSocket da usare per comunicare.

Di seguito si riporta uno schema indicativo di come sono i passaggi tra i due device.



Dopo aver instaurato la connessione si passa quindi alla parte di comunicazione e di come i due device riescono ad interagire tra loro.

Per entrambi i device si è quindi deciso di avere un numero finito di messaggi:

- "start"; questa stringa permette ai due device di capire quando mandare il proprio nome all'altro. Viene mandato quando i due device sono connessi e stanno per entrare nel loop di lettura dei dati provenienti dall'altro device.
- "^(\d)\$"; questa espressione regolare ci indica che possiamo ricevere un numero. Esso serve per comunicare all'altro device che l'altro giocatore ha toccato una casella di gioco e quindi ora sta a lui. È il modo in cui si rende sincrono il gioco. In questo modo viene anche segnalata la casella su cui fare l'update.
- "disconnect"; questa stringa dice ad uno dei due device che si sta per disconnettere e che quindi dovrà farlo anche l'altro.
- "again"; questa stringa dice all'altro giocatore che si vuole rigiocare, sia che uno dei due abbia vinto o che semplicemente si voglia resettare la tabella di gioco.
- tutto il resto delle combinazioni viene inteso come il nome del giocatore da cambiare nella TextView contenuta nella schermata. Nel caso in cui lo scambio dei nomi non avvenga correttamente il gioco continua a funzionare ma il giocatore vedrà il nome pre-impostato su quello dell'altro.

Si noti come i messaggi iniziali di spazio e nome giocatore sono praticamente in contemporanea (nella foto c'è solo lo scambio di una coppia di questi) mentre per il resto dei messaggi non sappiamo quando arriveranno nel tempo perché prima deve avvenire la mossa.

Di seguito si riporta quindi l'immagine schematica di come i due BluetoothSocket si scambiano i messaggi. In questa ipotetica partita vengono scambiati i nomi (come detto prima nella foto viene visionato solo uno dei due che manda il nome), dopodiché al tempo  $t_i$  il giocatore passa la sua prima mossa che è di mettere il segno X nella prima casella in alto a sinistra. Dopo la mossa, che arriva subito dopo a PLAYER1, esso decide di disconnettersi per cui viene spedito il messaggio di disconnessione anche all'altro giocatore.

