

cirullino_bot

Un bot di telegram per giocare a cirulla con gli amici.

Questo bot è scritto in JavaScript ed utilizza il framework [Telegraf.js](#). Questo framework è molto capace ma ha una documentazione molto povera e mal strutturata. Avrei preferito scrivere il codice in TypeScript ma, anche in questo caso, il supporto lascia desiderare. Si presuppone però un refactoring generale del codice in TypeScript

Part I - descrizione

Ho scelto di realizzare un bot per evitare di progettare e realizzare elementi come autenticazione ee front-end.

Comportamento tipico di degli utenti A, B e C che vogliono giocare

1. A, B e c avviano il bot con **/start**
 - se A, B o C non possiedono uno username il bot gli chiede di rimandargli il comando **/start** quando lo avranno impostato.
2. A utilizza il comando **/sfida** per iniziare una nuova partita
3. Il bot chiede ad A con quante persone vuole giocare
4. A deve rispondere un numero tra 1 e 3
5. Il bot chiede ad A di inviare singolarmente gli username degli altri giocatori
6. Il bot controlla che tutti i giocatori abbiano avviato il bot
 - se così non fosse il bot avvisa A e gli dice da un link per invitare gli altri utenti ad attivare il bot
 - l'utente deve riprendere da capo
7. Il bot manda un messaggio a B e C avvisandoli dell'invito a giocare da parte di A
8. Se B e C rispondono entrambi **/enter** il gioco incomincia
 - B e/o C possono scegliere di rispondere **/rifiuta** per non entrare in gioco con A
 - se così fosse A e/o B e/o C vengono avvisati che B e/o C hanno rifiutato e il processo si conclude
9. il bot avvisa tutti i giocatori che il gioco è incominciato: comunicando chi da li carte e chi inizia. Nel nostro esempio A inizia e C da le carte
10. ogni giocatore riceve un messaggio con le carte in tavola, il numero delle proprie scope e il numero di carte nel proprio mazzetto.
11. Ogni giocatore riceve le carte della propria mano come pulsanti al posto della tastiera
 - se le carte in mano di un giocatore sono tali per bussare il giocatore avrà il pulsante 'bussa'. In caso lo premesse il bot controlla che effettivamente sia il suo turno e che possa bussare. Se fosse il caso il bot comunica agli altri giocatori lo stato del gioco e la mano dei giocatori che hanno bussato
12. il giocatore di turno deve giocare la propria carta mandandola al bot
 - se il giocatore gioca una carta che non ha in mano il bot lo informa di ciò e lo invita a inviarne un'altra
 - se il giocatore gioca una carta ma non è il suo turno il bot lo informa di ciò e lo invia ad aspettare
13. il giocatore visualizza le possibili prese come pulsanti al posto della tastiera e manda la preferita al bot

- se il giocatore manda una presa non valida il bot lo informa di ciò e lo invita a inviarne un'altra
- 14. il bot informa gli altri giocatori della mossa di chi ha giocato e manda il nuovo stato del gioco con le carte in tavola, le proprie scope e il proprio mazzetto
- 15. Il bot informa i giocatori di chi il turno al momento
- 16. Ogni volta che una mano termina il bot informa i giocatori delle mani restanti
- 17. Il loop 10-16 si ripete finché le carte nel mazzo non finiscono
- 18. Alla fine del gioco il bot informa i giocatori dei punti accumulati e annuncia il vincitore

Comandi

| comando | descrizione |
|------------------|-------------------------------------|
| /start | avvio del bot |
| /sfida | inizio di una nuova partita |
| /entra | partecipare ad un gioco su invito |
| /rifiuta | rifiutare l'invito a un gioco |
| /esci | uscire dal gioco |
| /status | visualizza le statistiche di gioco |
| /tutorial | come giocare a cirulla |
| /aiuto | info sui comandi del bot |
| /privacy | info sulla privacy dei dati del bot |
| /info | info sulla realizzazione del bot |
| /stop | arresto del bot |

Messaggi dell'utente

carta

Bussare

Part II - implementazione

Command Handlers

User Message Handlers

Scenes