

Giorgio Ambrosi
10567531

Sviluppo di Bot Telegram in Python

Progetto di Ingegneria Informatica

Tutor: Giovanni Agosta

0. Repository

Il codice del progetto può essere consultato a questo link:

<https://github.com/Ambro-Gio/13thAgeBot-progetto>

1. Descrizione del progetto

1.1 Obiettivi:

Questo progetto ha come obiettivo la realizzazione di un bot Telegram in grado di assistere una campagna di un gioco di ruolo open source, con funzioni di supporto utili ai giocatori.

Il gioco scelto nell'implementazione è 13th Age, basato su d20 System, con alcune interessanti modifiche alla classica impostazione di gioco, tra cui il sistema delle Icone e miglioramenti nella fluidità del sistema di combattimento

1.2 Tecnologia utilizzata:

Per la realizzazione del bot è stato scelto il linguaggio Python, e in particolare il modulo `python-telegram-bot` che fornisce una traduzione in Python di tutti i metodi dell'API messa a disposizione da Telegram, oltre ad offrire alcune classi estese (presenti nel sottomodulo `telegram.ext`) in grado di occuparsi di alcuni aspetti tecnici della realizzazione di un bot ad alto livello, come ad esempio il riconoscimento di comandi o la gestione di una conversazione, facilitando il compito dello sviluppatore.

1.3 Scelte implementative:

Il bot ha necessità di salvare alcune informazioni in maniera persistente, come quelle su personaggi e giocatori. Per realizzare questa funzionalità si è utilizzata una struttura dati a dizionario, che viene salvata su file json ad ogni modifica. A ciascun giocatore viene assegnato un file json, permettendo così una corretta concorrenza nel caso in cui due giocatori utilizzino il bot nello stesso momento. Ad ogni modifica il file json viene salvato, per evitare possibili errori in caso di interruzione forzata o imprevista del bot.

Sono stati utilizzati dei file json anche per salvare dizionari contenenti tutte le informazioni necessarie alla creazione del personaggio: ad esempio, razze e classi disponibili, armamentario da usare, poteri a disposizione, etc.

È stata usata la classe `telegram.ext.CommandHandler` per la gestione dei comandi. `CommandHandler` è in grado di invocare il metodo corretto corrispondente a un comando mandato dall'utente. Per le funzionalità del bot che richiedevano più passaggi è stata usata la classe `telegram.ext.ConversationHandler`, in grado di simulare una conversazione con l'utente, con un meccanismo simile a un automa a stati.

2. Funzioni di 13thAgeBot

2.1 Panoramica dei comandi disponibili

I comandi disponibili per 13th age bot permettono di:

- Creare un nuovo personaggio (ogni giocatore può avere più di un personaggio a disposizione)
- Consultare la scheda di un proprio personaggio
- Aggiungere e rimuovere oggetti dall'inventario di un proprio personaggio
- Aggiungere e rimuovere oggetti magici dall'inventario di un proprio personaggio
- Lanciare dadi tramite interfaccia grafica o testuale, offrendo la possibilità di lanci multipli complessi, e di aggiungere bonus e malus

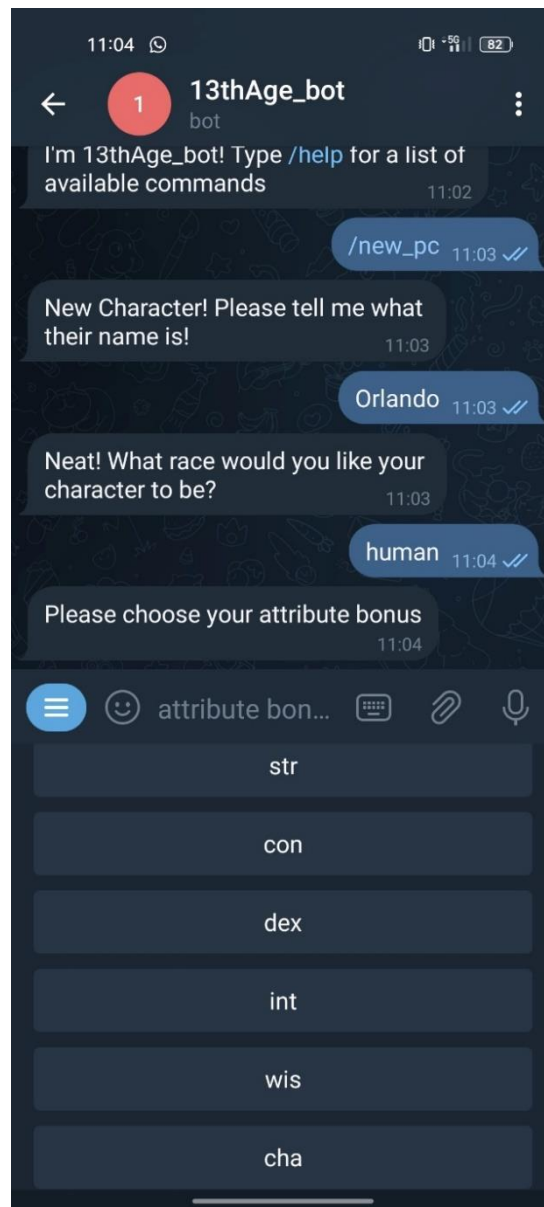
2.2 Creazione del personaggio:

La creazione del personaggio è sicuramente l'aspetto più complesso di gioco di cui il bot è in grado di occuparsi. L'utente è guidato in maniera lineare, attraverso una serie di messaggi, in tutti i passi richiesti alla creazione di un nuovo personaggio.

Per realizzare questa sequenza lineare, è stato scelto di utilizzare un ConversationHandler: ad ogni scelta dell'utente corrisponde uno stato della conversazione, e un metodo che si occupa di mostrare la corretta tastiera virtuale all'utente, e di salvare la sua scelta nel dizionario che rappresenta la scheda del personaggio.

Il dizionario è salvato nel file .json dell'utente a creazione finalizzata. Il file .json è univoco per ogni giocatore, e contiene tutti i personaggi che ha creato. Altri file .json sono utilizzati per salvare tutte le informazioni sul funzionamento delle razze e delle classi disponibili al giocatore (ad esempio, un elenco delle spell di primo livello che possono essere apprese dalla classe mago).

La funzione è chiamata semplicemente con il comando `"/new_pc"`



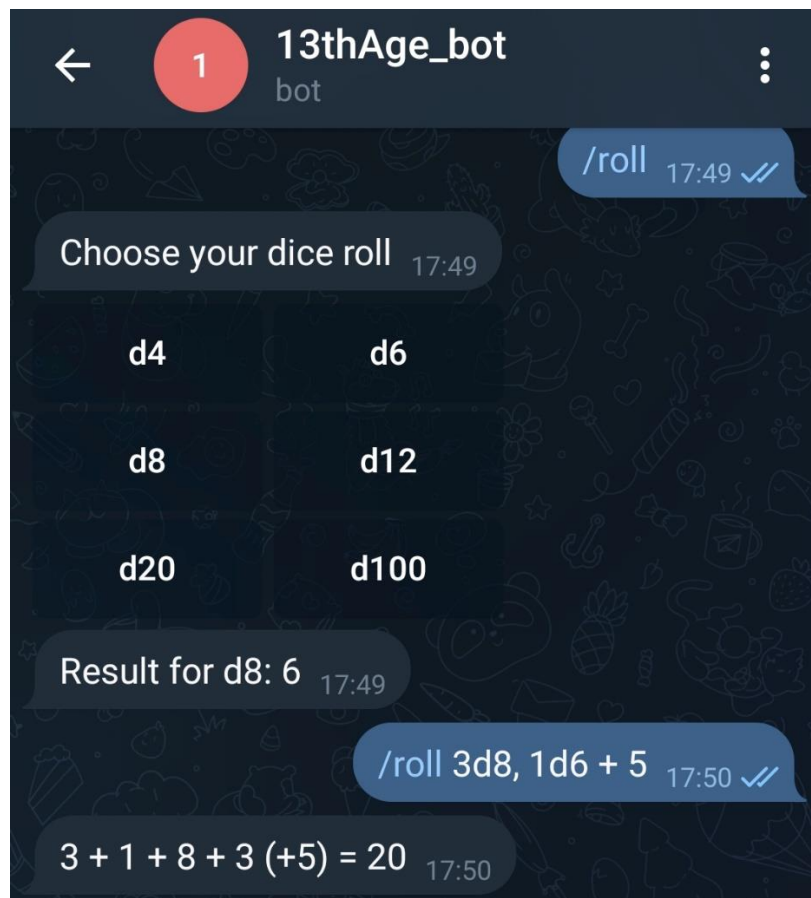
Un esempio di funzionamento della creazione personaggio

2.3 Lancio dei dadi

La funzione di lancio dei dadi è disponibile sia in versione testuale che grafica, tramite l'utilizzo di una tastiera virtuale inline.

La versione testuale è in grado di accettare il lancio di multipli gruppi di dadi, oltre a un bonus o malus. Il comando viene invocato tramite la sintassi `"/roll <input>`". Ad esempio, un comando accettato è `"/roll 3d8, 1d6, 2d4 + 5"`. Per riconoscere la validità della richiesta dell'utente viene utilizzata un'espressione regolare. In seguito, una fase di parsing dell'input si occupa di riconoscere i diversi gruppi di lanci da effettuare. Il risultato è presentato all'utente facendo vedere i lanci parziali e il totale.

La versione grafica è invocata semplicemente dal comando `"/roll"`. L'utente riceve in risposta una tastiera virtuale persistente, che permette di scegliere tra tutti i dadi disponibili del d20 system.



esempio di entrambe le versioni di lancio dei dadi

2.4 visione della scheda personaggio:

Il comando viene invocato con la sintassi `"/get_sheet <character name>"`. Le informazioni della scheda del personaggio corrispondente vengono recuperate dal file .json dell'utente e mostrate con un messaggio di risposta.

2.5 modifiche all'inventario

I comandi di modifica dell'inventario hanno un funzionamento molto simile tra loro. Permettono di aggiungere o rimuovere oggetti, oggetti magici, e valuta.

3. Possibili funzioni aggiuntive per 13thAgeBot

13thAgeBot copre una parte degli strumenti utili durante una vera campagna di gioco. Alcuni aspetti assenti nell'implementazione, che potrebbero essere utili a rendere l'esperienza di gioco più completa e fluida possono essere:

- Level up: un comando `"/level_up"`, dal funzionamento simile alla creazione personaggio, che gestisca il momento in cui un personaggio sale di livello. Analogamente a `"/new_pc"`, la funzionalità verrebbe implementata mediante ConversationHandler, e con l'utilizzo di un file json dove sono salvati tutti i cambi di stato per ogni livello (ad esempio, le nuove magie a disposizione per un mago a lv3)
- Gestione del turno di combattimento: un comando a utilizzo del Master, che si occupa di lanciare i dadi iniziativa per ogni giocatore e stilare una lista per il turno.