

Quick Start Tutorial : Discord Computer Olympiad Interface with GTP

12 novembre 2025

This quick tutorial introduces the use of the Discord Computer Olympiad Interface (DCOI) to perform automatic matches on Discord using the GTP protocol to communicate with your program.

This document assumes that you already have a Discord account and know how to use the basic features of Discord (joining a server, channel, accessing settings). If not, or more generally if you need more detailed instructions, please consult the **User Guide**, which will also introduce you to **advanced features, tips, and common mistakes**.

1 Creating a “bot account”

The first step is to create a “bot account,” that is, to create a Discord application with the options required to have a functional bot in the DCOI context. This allows the bot to interact in Discord conversations. If you find these initial steps too complicated, we can lend you a bot account.

The steps to create a bot account are as follows :

1. Create a **new application** on the Discord website : <https://discord.com/developers/applications>.
 - (a) Use the name of your program to name the bot.
 - (b) If you have any problems, try a different internet browser.
2. **Enable** the “Message Content Intent” option in the “Bot” tab.
3. Retrieve the bot TOKEN (**Reset Token** button) :
 - (a) **copy and paste** the token as the value of **PLAYER_BOT_DISCORD_TOKEN** in the file **parameters.conf** (in the discord_interface folder).
 - i. Never share the token — it is the password of your bot (sensitive information).
4. In the “OAuth2” tab, under “OAuth2 URL Generator,” **check** the “bot” option.
5. In the “bot permissions” subsection, check the following :
 - (a) **View Channels**,
 - (b) **Send Messages**,
 - (c) **Read Message History**,
 - (d) **Add Reactions**.
6. **Send** the URL located at the bottom of the page (in the “Generated URL” section) to the organizer so that they can invite your bot to the server.
7. Back in Discord (optional but strongly recommended) :
 - (a) In **Settings** (gear icon at the bottom left), “Advanced” tab : **enable “Developer Mode”**.
 - (b) Close the settings window.
 - (c) Click on **your name** at the bottom left, then click “**Copy ID**”.
 - i. Paste this value into the **parameters.conf** file as the value for **OWNER_ID**.

command	return	effect
clear_board	=	restart the game
undo	=	cancel the last move
play <player> <move>	=	plays <move> for <player>
time_left <player> <time>	=	inform the time left in seconds
genmove <player>	=<move>	return the <move> to play and play it
quit	=	shutdown the program
player	=<player>	return the current player
game <game>	=	initialization for <game>

TABLE 1 – List of GTP commands to implement, the syntax of their return and their expected effect (<move> is an action in text format (A1, A2-B3, ..., see the syntax in the User Guide) ; <player> is an integer $n \in \mathbb{N}$; <game> is the game name of the match). Remark : commands “game” and “player” are not part of standard GTP ; they are optional (make them only return “=”); “game” is required only for multi-games participation ; “player” is required only for profiling (see User Guide). Remark : each return starts with “=”.

2 Installation

In a terminal, run (prerequisites : Python 3.10+) :

1. **cd discord_interface**
2. (optional but recommended) : activate a Python virtual environment
3. **chmod +x install.sh**
4. **source install.sh**
5. (optional) Test the installation : run the following in the terminal : “python3 test_installation.py”.

3 GTP Command Specification

Table 1 briefly describes how to implement in your program the execution of the GTP commands required by the Discord bot for communication. Since GTP communication occurs through standard output, we recommend not using it for any other purpose (such as debugging, etc.).

To find out the actions associated with the different games, consult the document “game_move_coordinates.pdf”.

If you use the game “Free_game” : each player takes turns (concatenate the successive actions with “-” if necessary). In addition, with Free_game, your bot must play “end” when the match is finished.

4 Running Your Bot

1. Edit the text file **parameters.conf**, specifying the following attributes for your program :
 - (a) program_name
 - (b) program_arguments
 - (c) program_directory
2. Optionally, program test : verifying the GTP command specification :
 - (a) run in the terminal : “python3 test_gtp_ai_from_conf.py”
3. Then run in terminal : “python3 main_gtp_ai_from_conf.py” to launch your bot.

5 Running a Match

1. Join the competition's Discord server :
 - (a) beta test serveur : <https://discord.gg/npCBBJGe>
 - (b) competition serveur : coming soon (T.B.A.)
2. Join one of the match channels of the Discord serveur.
3. You, your opponent, or an organizer must execute the following commands in the Discord conversation :
 - (a) Specify the game using the command **!set game GAME_NAME**
 - i. if needed : **!available_games**
 - ii. if your game is not in available_games :
 - A. use game “Free_game” (for games without chance only)
 - (b) Start the match with the command **!start @PLAYER_1 @PLAYER_2**
 - i. For each *PLAYER_i*, write “@” then select the corresponding player or bot from the list and confirm by pressing “Enter”.
 - A. If it doesn't appear in the list, start typing its name.

6 Incident Management

1. If the bot program has stopped playing but is still running :
 - (a) If the issue comes from your AI program :
 - i. If needed, review the “<your_program_name>_gtp.log” file and update your program accordingly.
 - ii. Use the “!gtp_replay” command in the Discord channel :
 - A. This restarts your program and replays all GTP commands from the beginning,
 - B. then continues the current Discord match.
2. If the bot program has crashed :
 - (a) After restarting it,
 - i. use the “!continue” command in the Discord channel :
 - A. This resumes the match from where it left off.