

Testing Procedures

When it comes to testing the DisPatron5e Discord bot, there are 4 basic commands that will need to be tested that then vary based on the information needed and the desired output, *roll, *add, *lookup, and *help. There will be various ways that it has been tested on and been improved to make sure that it doesn't break or crash the bot.

When it comes to testing *roll, 2 pieces of output will be required, what type of dice are being rolled, and how many are being rolled. Both inputs must be positive inputs since you can't roll a negative number of dice and in the game of Dungeons and Dragons, it is not possible to roll a negative number on a normal dice. To make sure that the output does not give an error, I will have if statements that verify that both inputs are of an integer type and are greater than 0. Else, you will be given the error, "Input for both values must be positive integers. Please try again."

When it comes to *add, the input must be 2 integers, but unlike *roll, the integers can be negative. So it will be similar to *roll in terms of testing where it will require the input to be an integer, however, it does not need it to be positive. It is possible to add a negative number in Dungeons and Dragons, like if a stat modifier is negative. So testing will be similar to *roll where I will have an if statement to verify that the input is an integer datatype, but it will not need to also make sure that the type is greater than 0. This should force the user to produce an integer type. If the user does give an invalid datatype, they will be given the error, "Input for both values must be integers. Please try again."

The `*lookup` command is the most unique compared to the others. This command will require that you be given a specific string so that it can print the valid table or information from the players handbook. If the user doesn't get a valid lookup, they will instead be given an error message stating "Invalid lookup." As well as a list of the information commands they can use in `*lookup`.

There will also be a `*help` command that will work in a similar way to `*lookup` but will give different information. The `*help` command will give a list of valid commands if given alone and will give syntax information if given a corresponding string of one of the above commands. Similar to the `*lookup`, if they give input that doesn't correspond to a specific command, they will be given a list of valid input that is accepted in the `*help` command.