

Dungeons & Dragons Fight Club: Using Monte Carlo Simulation to Determine Optimal Magic Item Selections for Player Characters

Project Scope Update

The overall goals of my project remain unchanged: I am developing a Monte Carlo combat simulator for Fifth Edition Dungeons & Dragons that evaluates win probability, expected rounds to defeat, and average damage per round using parsed character and magic-item data. The main adjustment since starting the project involves the source of rules data. I originally planned to retrieve monsters and items directly from D&D Beyond, but their authentication requirements and security protections prevent reliable automated access. Because of these limitations, I am shifting to the Open5E API, which provides publicly accessible and structured JSON data suitable for the simulator. The simulation design, data flow, and analytical objectives remain the same.

Data Sources

Character and magic-item information is being loaded from local xml and Excel files. These files contain the relevant combat statistics such as ability scores, attack bonuses, hit points, armor class, and item-based modifiers. My parsing code converts these values into Python objects that can be used during simulation runs.

For rules data, I am transitioning to Open5E, which offers a stable REST API for monster statistics. This information will be parsed so they align with the data structures used by my simulation engine. I am also considering caching API responses locally to reduce repeated calls during testing.

Issues / Difficulties

The primary issue has been the inability to automate data retrieval from D&D Beyond. Their site does not provide an open API and blocks scripted access through authentication and security controls, making it unsuitable as a programmatic data source for this project. Moving to Open5E resolves this access problem, but introduces some challenges, including differences in data structure, occasional missing fields, and the need to parse damage expressions such as “2d6+4” into numeric forms.

Potential Issues Expected

Upcoming challenges may include handling more complex D&D mechanics (advantage, resistances, conditional effects), ensuring accurate integration between character data and monster data, and managing runtime performance if simulations become computationally heavy. End-to-end validation will be an important next step as all components begin operating together.