The Journal of Open Source Software

GNOLL: Efficient Multi-Lingual Software for Real-World Dice Notation and Extensions

Ian Frederick Vigogne Goodbody Hunter [1]

1 Independent Researcher

DOI: 10.21105/joss.04816

Software

- Review
- Repository
- Archive

Editor: Gabriela Alessio Robles

Reviewers:

- @Smattr
- @steven-murray

Submitted: 09 September 2022
Published: 17 January 2023

Summary

Dice Notation is a system for describing how to roll collections of dice. It is often used to assist in understanding the rules of games - particularly tabletop roleplaying games (TTRPGs). Existing research software in this space has been primarily designed for other researchers and statisticians despite the fact that a large population of those actually playing these games are young (Natalt Design, 2020) or not involved in statistical research.

GNOLL is an open-source library for parsing commonly used dice notations in gaming system research and/or software development. GNOLL is performant, supports most popular and obscure notations, has permissive licensing, and is integratable into many other systems due to it being written in C. At present, the repository has working examples of integration into 13 different programming languages, such as Go, Java, Julia, Python, Perl and R.

Statement of Need

While there are several dice-rolling utilities on the market for research/commercial use, there is no current solution that:

- is open source / permissively licensed.
- can be easily integrated into other software despite language differences.
- provides a reference for other implementations.
- supports such a diverse set of dice notation.

While some solutions may offer one or two of these points, GNOLL addresses all of them.

Without a more extensive project for reference/integration, many software developers have created their own dice notation parsers (because a parser for a simple subset is not too difficult to develop). These are usually sufficient for their immediate needs but often create discrepancies in notation standards and do little to change this path for future developers.

Related Reading

Few publications specifically discuss dice notation. The most prominent papers are named ROLL (T. Mogensen, 2003) and TROLL (T. Æ. Mogensen, 2009). GNOLL is a recursive acronym expanded from "GNOLL's Not *OLL" to distinguish this research and still pay homage to the original work. The reason for its distinction is that GNOLL's notation is focused on real-world gaming usage of dice notation, whereas ROLL and TROLL are notations targeted at statistics research.

Example Notation

There are too many different operations and combinations of dice notation to describe within the space constraints of this paper but they are discussed at length in the project's documentation, including the rationale for each of the notation choices. We describe a sample of the base dice notation below.

The most basic dice roll in dice notation can be expressed as

$$x\mathbf{d}y,\ where\ x, y\ \in \mathbb{Z}^{+}.$$

x dice are rolled with values from 1 to y. Where x is not specified, it is assumed that its value is 1. the exclusion of y produces an error (It is ambiguous to have a dice with no sides).

Performance

GNOLL performs well against other dice-rolling libraries available online both in terms of performance and functional coverage. In the Figures section below, we show some simple benchmarking results against the TROLL system and a C++ parser and also test GNOLL's Python interface against popular Python packages, and find that GNOLL is generally more performant.

Figures

Figure 1 - GNOLL Performance (C)

Comparison of GNOLL's performance against other C/C++/SmallTalk dice notation parsers - TROLL and E Bailey's 'Dice Parser'.

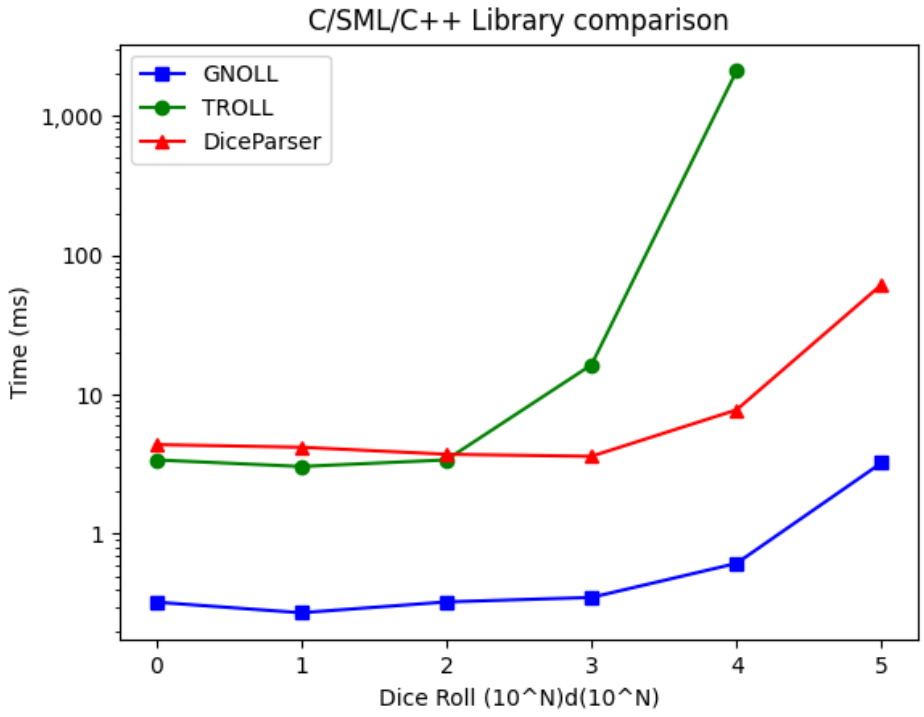

Figure 1: A graph showing GNOLL's performance on different sizes of dice rolls. It is faster than TROLL and DiceParser on each tick

Figure 2 - GNOLL Performance (Python)

Comparison of `GNOLL`'s performance against other Python dice notation parsers. (Chosen from popular dice rollers on PyPi (Jones, 2002) - "Dice", "RPG Dice", "Python Dice", "d20"). Unplotted points either were non-functional or exceeded a set timeout.

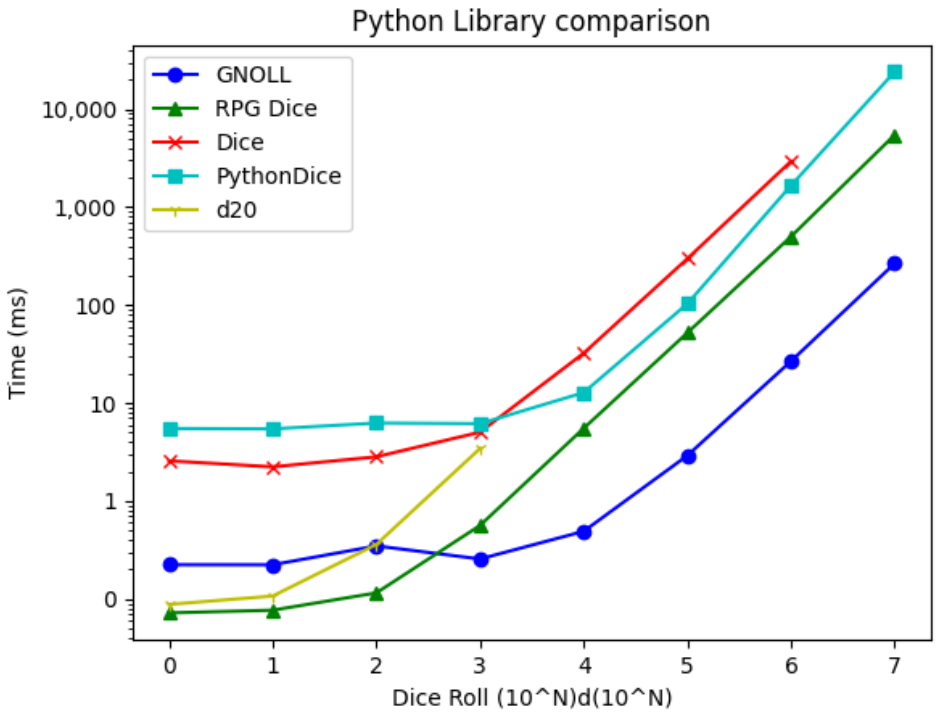

Figure 2: A graph showing GNOLL's performance (via Python binding) on different sizes of dice rolls. It slightly underperforms for small sizes, but is faster than other Python libraries for large sizes.

Acknowledgments

Thanks to my dog for the desk-side companionship and the demand for healthy stick-fetching breaks. Thank you also to the various TTRPG publishers and online communities for enabling people all over the world to play make-believe together.

References

Jones, R. (2002). *Package index and metadata for distutils* (PEP No. 301). https://www.python.org/dev/peps/pep-0301/

Mogensen, T. (2003). Roll: A language for specifying die-rolls. *Practical Aspects of Declarative Languages*, *2562*, 145–159. https://doi.org/10.1007/3-540-36388-2_11

Mogensen, T. Æ. (2009). Troll, a language for specifying dice-rolls. *Proceedings of the 2009 ACM Symposium on Applied Computing*, 1910–1915. https://doi.org/10.1145/1529282.1529708

Natalt Design. (2020). *A history check on 2019 | D&D's best year ever_2019*. Wizards of The Coast.