Artificial Intelligence Fundamentals
Project Report : Judy_Bot_T3

Simone Marzeddu - Università di Pisa

Master Degree in Computer Science A.Y. 2022/2023

0.1 Introduction

Released in 1987, NetHack [1] is still one of the hardest video games ever published, with the victory that is achieved for the first time by the most determined players only after years of experience. Over time NetHack has attracted the attention of researchers and Artificial Intelligence enthusiasts, embodying a property-rich environment relevant to bring scientific progress to the state-of-the-art in Artificial Intelligence.

Within the game, the player will have to face fifty procedurally generated levels plus five extra floors, characterized by an even more extreme difficulty then the previous ones. The numerous monsters, the hidden traps and the constant need to feed the protagonist will be just some of the elements that can lead the player to the "game over", causing the entire game to start over.

Starting from the analysis of the thesis project Judy_Bot_T3 [2], which aimed to create an open-source bot in Python for the videogame NetHack, the document will deal with the main characteristics of the developed agent and its evolutions in terms of what has been learned during the Artificial Intelligence Fundamentals course.

0.2 Related Works

Since its first release, the Judy_Bot_T3 project has been based on several previous studies, such as the implementations of the previous historical bots TAEB [3], BotHack [4] and AutoAscend [5], as well as on two main frameworks: NetHack Learning Environment (NLE) [6] and MiniHack [7]. NLE is a framework that aims to provide an interface to NetHack's terminal software. To succeed in this objective, NLE formalises the game commands (input) in Python methods with appropriate configuration attributes, making the game output usable by the higher-level implementation through special observation data structures in accordance with the canons of the Gym interface on which it is founded [8].

MiniHack, on the other hand, is a framework geared towards the meticulous study of specific features of the otherwise vast NetHack environment. Thanks to those defined as "tasks" within the framework's documentation [9], the developed agent can deal with restricted environments characterised by specific game elements in order to study a specific behaviour through limitations in the space of actions and observations available to the agent.

The same scientific personalities behind the development of these frameworks are also the creators of the NetHack NeurIPS Challenge [10], a competition that received a fair amount of attention during its 2021 edition, with the participation of more than four hundred and eighty developers and the submission of over forty implementations. Thanks to the results of the challenge, it was possible to compare Judy_Bot_T3's results with those of the other candidates, following the evaluation metric: ascension number - median of scores - average score. These statistics played a key role in establishing the degree of success of the agent, which with an average score of 744 and a median score of 645 3 surpassed, in its first version, the eighth competitor: "JustPaulsAI" [10].

With the aim of its presentation in the Artificial Intelligence Foundamentals exam, Judy_Bot_T3 underwent numerous refinements both from an aesthetic point of view (with the relocation of the component modules in separate files, the inclusion of comments describing each method and the writing of the README.md file) and from a functional point of view (with the refinement of existing modules according to the notions of the course and new elements learnt through further study of the game).

The capabilities of the agent after the updates as well as the technical details of its implementation will be analyzed in the rest of the report, after which a further comparison with the statistics of the NetHack NeurIPS Challenge 2021 [10] will be made for a final assessment.

0.3 Methodologies

The ability to move efficiently within the NetHack map represents one of the most important implementations for an agent developed in this environment. For this purpose, Judy_Bot_T3 offers the "DungeonWalker" class, aimed at encapsulating functions useful for exploring the game world, including also all the implementations necessary for path finding. In order to elaborate its navigation plans, Judy_Bot_T3 makes use of the "A*" [11] algorithm, widely used at the state of the art. However, an important observation concerns the heuristics associated with the algorithm, a determining characteristic to ensure maximum performance in the case study environment.

NetHack envisages that the playing character and other entities within the game can perform movements in the map grid in eight directions. Because of this characteristic of the environment, it proved essential to identify an appropriate admissible heuristic, with the design choice therefore falling on the "octile distance" heuristic after a careful research phase.

```python
# heuristic for the distance between two points
# environment with 8 directions of movement
def h_octile_distance(self, ay, ax, by, bx):
    x_d = abs(ax - bx)
    y_d = abs(ay - by)
    return (1.414 * min(x_d, y_d)) + abs(x_d - y_d)
```

Code 1: "octile distance" Python implementation

The "octile distance" heuristic evaluates a diagonal move as more costly than a move in the four directions, while giving it less weight than two moves in the same one, representing an admissible heuristic appropriate for problems such as NetHack. As an example, the following is a comparison of the identified "octile distance" heuristic with the famous "Manhattan distance" heuristic, known to be functional for searching in four-way environments.

Given the existence of a point A (0,0) and a point B (4,4) 2, and considering the cost of a move in the four directions equal to 1, the "Manhattan distance" would predict a cost of moving from A to B equal to 8. Starting from the same premises, but looking at the case of the "octile distance" heuristic, which predicts $\sqrt{2}$ as the cost of a diagonal movement action, the evaluation heuristic would yield a result equal to 5.66.

It is therefore clear that the "Manhattan distance" applied to the environment under consideration would not lead to an approximation less than or equal to the true distance between the two points, resulting in a not admissible heuristic and consequently damaging the properties of "A*".

Following the modifications made to the code in preparation for the Artificial Intelligence Fundamentals exam, the agent demonstrated significant improvements in the tasks defined by the "Elbereth", "Eat" and "Fight" modules, in addition to the various slight enhancements of several minor sections of the code. Thanks to these changes, derived mainly from the in-depth study (facilitated by the MiniHack framework) of some specific game mechanics, the agent is now able to distinguish between healthy and unhealthy foods with an almost certain success rate, discarding suspect foods and thus avoiding potential risks. The agent is also now able to protect himself through "Elbereth" engraving, taking advantage of the game mechanics whereby this action prevents most enemies from harming him. Thanks to a more refined logic, the bot will now avoid engraving when faced with situations where it would not benefit from it.

Prayer is another powerful tool in the agent's hands, and thanks to the latest changes, the agent can now make more optimized use of it.

Finally, some updates to the combat module now allow Judy_Bot_T3 to more successfully deal with specific enemies deserving of dedicated strategies, such as "floating eyes" and "werefoos".

0.4 Assessment

The empirical evaluation of the agent was strongly guided by the comparison with the parameters and results of the NetHack NeurIPS Challenge 2021 [10]. Specifically, the code was run on a sample of 1000 games, all generated according to the parameters of the challenge and through the use of "NetHackChallenge-v0" environment, available on the latest version of NLE (0.8.1). This environment comes as close as possible to playing the real game of NetHack with a random character to start, and a full keyboard of actions to take.

Collecting the outcomes of the games, the agent was then evaluated according to the three metrics: number of ascensions (wins) achieved, median score between games and average score between games. Having thus calculated the same metrics used in the challenge for the evaluation of the competitors, it was then possible to assess the quality of Judy_Bot_T3 in comparison to the latters. As the agent never achieved an ascension, as well as the other bots competing in the 2021 edition of the challenge, the pivotal parameter for the evaluation was the median of the scores between the games [10].
Thanks to the changes in the last few updates , the evaluation phase of Judy_Bot_T3 ended with the encouraging results of 1046.96 for the average score and 817 as the median score, statistics that rank the agent with a solid sixth position 4 on the NetHack NeurIPS Challenge 2021 ranking list [10].

0.5 Conclusion

The results achieved by Judy_Bot_T3 are certainly encouraging, as is the relative speed of improvement compared to previous thesis project results (around fifty to sixty hours of work). Datas that certainly continue to demonstrate the very high potential of the NetHack Learning Environment and MiniHack frameworks in driving scientific research around the NetHack game.

However, it is clear that the way to build a bot capable of "solving" the problem of the NetHack game is still incredibly long. The challenge winners themselves have a huge difference in scores from what achieved by Judy_Bot_T3 (the winner, AutoAscend [5], boasts an unbeaten record of 5336.5 as a median score), and the same statistics of the podium agents are largely insufficient to consider the NetHack case of study as scientifically saturated 1.

0.6 Appendix

0.6.1 Team Contributions

Being the result of the evolution of the previous Thesis project, Judy_Bot_T3 has not seen the participation of a team for its realization and is presented for examination as an individual project.

0.6.2 GitHub Metrics

As an individual work, the distribution of GitHub commits per group member is ignored in the document. However, the entire project is open-source and available on GitHub at the link: https://github.com/SimoneMarzeddu/Judy_Bot_T3.

0.6.3 Relationship with the Course

The design phase of the agent exploited what is presented in AIMA Chapter 2 (second lesson of the course) to properly define the task environment and the agent performance measure.

The technical core of the entire agent is certainly the implementation of the "A*" algorithm supported by the "Octile Distance" heuristic, developed according to what was learned from the fourth lesson of the course (AIMA Chapter 3).

Δ	#	Participants	Median Score	Mean Score	Total Episodes Run	Tag
●	01	AutoAscend	5336.5000	3820.6989	4128.0000	scripted
▲	02	StudentsOfSto...	3203.5000	2498.6122	4128.0000	scripted
▼	03	Panic	2436.0000	3982.7529	4128.0000	scripted
▲	04	raph	1727.5000	1350.8106	4128.0000	neural
▲	05	KakaoBrain_N...	1456.0000	2062.7868	4128.0000	neural
▼	06	Chaotic-Dwar...	756.0000	956.7321	4128.0000	neural
▼	07	bwsyd__bosae...	681.5000	897.6080	4128.0000	scripted
▲	08	JustPaulsAI	475.0000	586.0673	4128.0000	neural

Figure 1: Official ranking NetHack NeurIPS Challenge 2021.

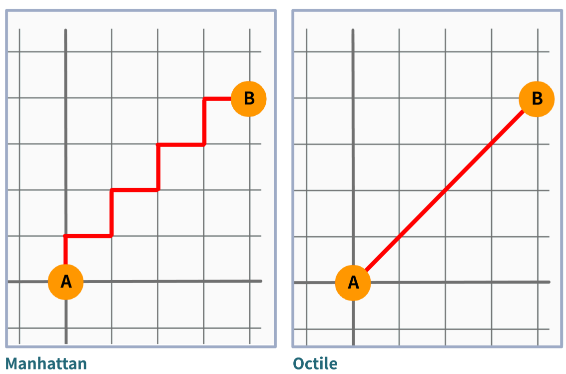

Figure 2: Comparison between Manhattan and Octile Distance heuristics.

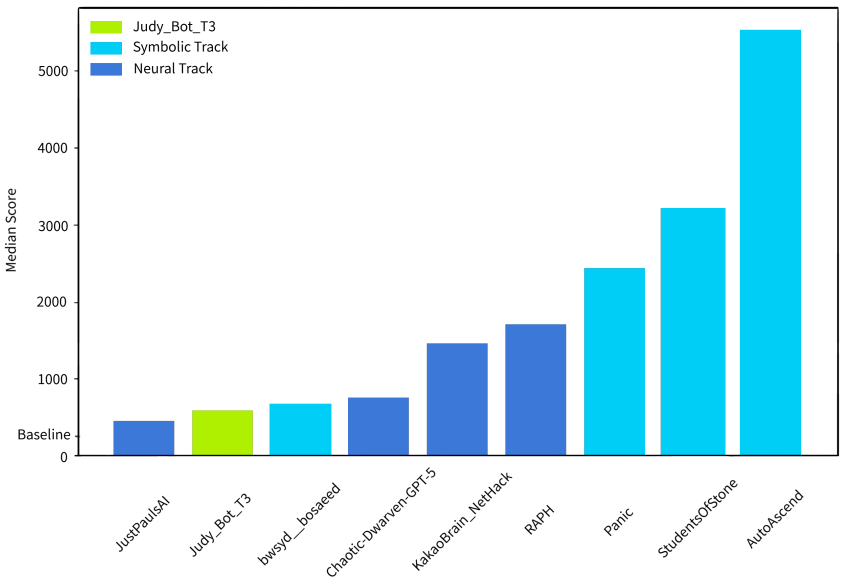

Figure 3: Comparison between Judy_Bot_T3 and the other competitors (before updates).

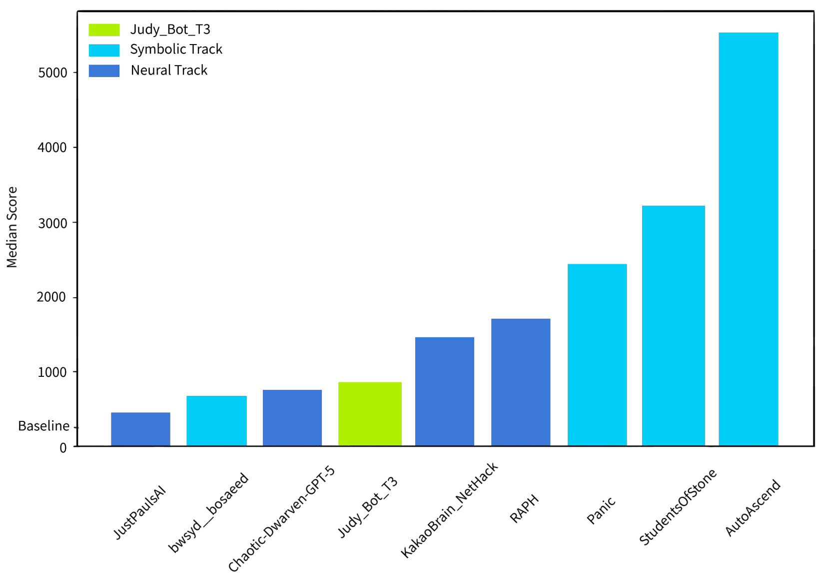

Figure 4: Comparison between Judy_Bot_T3 and the other competitors (after updates).

Bibliography

[1] nethack.org - nethack official site. //https://nethack.org. [Online; in data 29-ottobre-2022].

[2] Simone Marzeddu. Judy_bot_t3 1.0. https://github.com/SimoneMarzeddu/Judy_Bot_T3, 2022.

[3] Stefan O'Rear (sorear) Sebastian P. (Sebbe) Sean Kelly (arcanehl) Alex Smith (ais523) (sawtooth) Shawn M. Moore (sartak), Jesse Luehrs (doy). Taeb - blog. http://taeb-nethack.blogspot.com/. [Online; in data 01-novembre-2022].

[4] Bothack - github. https://github.com/krajj7/BotHack, 2016. [Online; in data 01-novembre-2022].

[5] Autoascend - github. https://github.com/maciej-sypetkowski/autoascend, 2021. [Online; in data 01-novembre-2022].

[6] Alexander H. Miller Roberta Raileanu Marco Selvatici Edward Grefenstette Tim Rocktäschel Heinrich Küttler, Nantas Nardelli. Nle github. https://github.com/facebookresearch/nle, 2020.

[7] Vitaly Kurin Jack Parker-Holder Minqi Jiang Eric Hambro Fabio Petroni Heinrich Küttler Edward Grefenstette Tim Rocktäschel Mikayel Samvelyan, Robert Kirk. Minihack github. https://github.com/facebookresearch/minihack, 2020.

[8] Alexander H. Miller Roberta Raileanu Marco Selvatici Edward Grefenstette Tim Rocktäschel Heinrich Küttler, Nantas Nardelli. The nethack learning environment. https://arxiv.org/abs/2006.13760v2, 2020.

[9] Mikayel Samvelyan, Robert Kirk, Vitaly Kurin, Jack Parker-Holder, Minqi Jiang, Eric Hambro, Fabio Petroni, Heinrich Küttler, Edward Grefenstette, and Tim Rocktäschel. Minihack the planet: A sandbox for open-ended reinforcement learning research. In Joaquin Vanschoren and Sai-Kit Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021.

[10] Eric Hambro, Sharada P. Mohanty, Dmitrii Babaev, Minwoo Byeon, Dipam Chakraborty, Edward Grefenstette, Minqi Jiang, DaeJin Jo, Anssi Kanervisto, Jongmin Kim, Sungwoong Kim, Robert Kirk, Vitaly Kurin, Heinrich Küttler, Taehwon Kwon, Donghoon Lee, Vegard Mella,

Nantas Nardelli, Ivan Nazarov, Nikita Ovsov, Jack Holder, Roberta Raileanu, Karolis Ramanauskas, Tim Rocktäschel, Danielle Rothermel, Mikayel Samvelyan, Dmitry Sorokin, Maciej Sypetkowski, and Michal Sypetkowski. Insights from the neurips 2021 nethack challenge. In Douwe Kiela, Marco Ciccone, and Barbara Caputo, editors, *NeurIPS 2021 Competitions and Demonstrations Track, 6-14 December 2021, Online*, volume 176 of *Proceedings of Machine Learning Research*, pages 41–52. PMLR, 2021.

[11] Peter Hart, Nils Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.