

Grand Valley State University
ScholarWorks@GVSU

Technical Library

School of Computing and Information Systems

2018

Story Parsing and Adventure Generation with Python and Postgres

Ryan Solnik
Grand Valley State University

Follow this and additional works at: <https://scholarworks.gvsu.edu/cistechlib>

ScholarWorks Citation

Solnik, Ryan, "Story Parsing and Adventure Generation with Python and Postgres" (2018). *Technical Library*. 323.

<https://scholarworks.gvsu.edu/cistechlib/323>

This Project is brought to you for free and open access by the School of Computing and Information Systems at ScholarWorks@GVSU. It has been accepted for inclusion in Technical Library by an authorized administrator of ScholarWorks@GVSU. For more information, please contact scholarworks@gvsu.edu.

Story Parsing and Adventure Generation with Python and Postgres

By
Ryan J. Solnik
December 2018

Story Parsing and Adventure Generation with Python and Postgres

By
Ryan J. Solnik

A project submitted in partial fulfillment of the requirements for the degree of
Master of Science in
Computer Information Systems

at
Grand Valley State University
December, 2018

Table of Contents

Abstract.....	4
Introduction.....	4
Background and Related Work.....	5
Program Requirements	5
Implementation	6
Results, Evaluation, and Reflection.....	7
Conclusions and Future Work.....	8
Bibliography	9
Appendices.....	10

Abstract

Dungeons and Dragons is a tabletop roleplaying game that allows players to assume the roles of adventurers in medieval fantasy setting while one player is tasked as playing the role of the Dungeon Master (DM). This player facilitates the story and all other characters not played by the other players. Adventure Day is a toolset for Dungeons and Dragons 5th Edition that assists Dungeon Master in formatting their Story as well as gathering useful details for the challenges presented within that story. Adventure Day aims to accomplish this by associating relevant monster data from Postgres database while using the text input of a desired adventure parsed through a Python application to filter for highlighted terms and take actions based on a defined dictionary within the same text file. Adventure day is capable of building encounters with monsters specified by monster environment and/or abilities that they possess, provide inspiration for non-player characters that the DM will portray via personality traits, and finally help to scale a monster's difficulty depending on the power level of the player's characters. Adventure Day generates output (Markdown) for use in a third-party website that formats Markdown text into a format that mirrors that of published material.

Introduction

Dungeons and Dragons is at its core a fantasy storytelling game that pits players vs the perils of a fantasy world where they assume the role of legendary heroes while one player assumes the moniker of Dungeon Master. This player is responsible for all other elements of the game world including other characters the players may meet, challenging the players with deadly monsters, and crafting a story that will live on in the players memories for years to come.

Taking on the role of the Dungeon Master can be a daunting task for those new or inexperienced and can easily become overwhelming when faced with all the various hats that this player must wear. Roleplaying characters that the players meet, describing scenes to evoke themes, and crafting meaningful stories is no simple task. Every Dungeon Master fear that they either prepare too much content for their next game or by some cunning plan their players manage to evade all their carefully prepared plots and enemies. To combat this Dungeon Masters typically prepare a rough outline to have an idea of what the adventure is to contain and feel confident about the goal of the adventure while leaving enough room to improvise new scenes based on the players decisions or explorations. Adventure Day aims to assist a Dungeon Master in compiling his outline with pertinent data and formatting it in a legible output.

Adventure Day assists in compiling story scenes, characters that the players will encounter and the monsters that they'll confront by injecting needed statistics and generating content to inspire a Dungeon Master's stories. Adventure Day creates random encounters the players may happen across, compiles monsters for easy reference while in combat encounters, and suggests characteristics that pertinent characters that are portrayed may exhibit all while. The goal is to streamline the creation process of a Dungeons and Dragons session while preserving the creativity of the Dungeon Master.

Background and Related Work

There are many well-known tools utilized by the community. For generating combat encounters, **Kobold Fight Club** and **Goblinist**, are two of the best that are freely available web-based tools. They allow selection of monsters based on characteristics, random generation, as well as selection of what books are referenced for monster entries. They are both easy to use tools that can make creating a combat encounter for an adventure as simple as a few clicks. These tools create combat encounters composed of monsters however do not display statistics for these monsters or make them available. This type of information would be very helpful for actual gameplay but may be intentionally left out due to licensing reasons.

Donjon is a website that provides a wide assortment of random generators for various tabletop roleplaying games. Its strong suit is in breadth of different content that it can generate for any game that a Dungeon Master may be planning. This website has been noted many times on various forums and subreddits as one of the best for generating random content. The downside is that it generates truly random content and doesn't accept input from a user to influence the "randomness" of the content generated.

The Homebrewery is a premier free web-application that generates "authentic looking Fifth-Edition" formatting for any text that a Dungeon Master may want to include. It seems to be the only tool of its kind that utilizes a markdown input with CSS to create a formatted document that looks like it could be included in a published product. There are drop down menus that allow creation of patterns for various text and table attributes as well as page formatting and separation for ease of reading. The only minor flaw that could be improved upon is the user friendliness of the website for first time users and potentially a REST API that could allow automatic content generation.

Program Requirements

Adventure Day is comprised of 3 main software modules:

Monster Vault contains a Postgres database that houses all the data that the Adventure Day program references and retrieves. This database contains important monster statistics that are used for gameplay as well as descriptive traits that are used when generating Non-Player Characters that the Dungeon Master will portray over the course of the adventure. Monster's have also had a environment associated with each of them that is used in generating random encounters based on environments that they would be potentially encountered by the players.

Adventure Parser is responsible for reading a text-based adventure outline and separating out pertinent elements that it identifies via regular expressions and can be used to create a Markdown formatted file. The parser identifies headings, scenes, and dictionary entries. These tokens make up a Dungeon Master's outline for an adventure and additional operations that will inform the data that is retrieved from the Monster Vault database and/or modify the entries. Once all entries have been retrieved Adventure Parser will format all relevant data into a markdown format.

Monster Works is a collection of different algorithms that make determinations on what type of records that are retrieved from the Monster Vault database, perform operations on the data retrieved and organize these records into combat encounters that Dungeon Master's can use in their adventures. Once this data is retrieved Monster works can modify a monster's difficulty level thereby altering the record's statistics based on the rules laid out in the Dungeons and Dragons rules. All the algorithms implemented in Monster Works are kicked off by key / value entries being present in dictionary entries that are parsed by Adventure Parser.

Implementation

The implementation of this application leverages both Python 3.7 and Postgres as a back-end database.

Monster Vault: This is a Postgres database whose schema is composed of an array of tables that contain a Monster's statistics. The main table is the "Monster" table which contains relevant combat statistics that are specific to individual monsters. Additionally eight other columns with Boolean values are appended to each monster record which details which environment these monsters could be encountered in. Tables have also been included in the schema to house special abilities, actions, and reactions and are named titled respectively. "Special Abilities," "Actions," and "Reactions" each house records that detail the rules and numeric values required for using the monster in the game. Intersect tables have been created between "Monsters" and each of the "Special Abilities," "Actions," and "Reactions" Tables associated to accommodate the many-to-many association between them. Finally, a "NPC" table is included to contain different Quirks, Traits, and Appearances that are randomly selected when generating a NPC in Monster Works.

Adventure Parser: The parser begins by accepting a text file that contains Markdown headings and textual scenes that describe a potential story scene (Appendix A). It will then process the file by pulling out the individual headings by leveraging regular expressions to find markdown headings, bolded terms, and a dictionary section that drives decision logic of what records to retrieve from the database what to generate and how to modify the records once returned. Once the text file is tokenized the parser will confirm if the bolded terms have an associated dictionary entry that would require additional operations. If no dictionary entry is found it is assumed that no additional operations are needed to be performed on the bolded term. If the term matches a monster's name, the monster's record will be queried from the database and returned. However, if a terrain type is specified, like "dungeon", four random encounters that vary in difficulty will be generated for the scene that the term was found in.

Monster Works: If a dictionary entry is specified in the text file that contains an environment that also corresponds to one of the supported environment types the application will generate a series of encounters in varying difficulty. The encounters have four different difficulty levels which correspond to increasing thresholds of experience points. For example, a medium difficulty encounter for four 7th level players would be between 3,000 and 4,299 experience points where a Hard encounter is between 4,400 and

6,799 experience points. Monsters each have an allotment of Experience Points which are added together to determine the total experience the encounter is worth. As a rule of thumb for encounter design, the side of the conflict that has access to more actions has an advantage in winning the conflict. To account for this advantage a multiplying function is applied to the total experience points per encounter based on the number of monsters that are to be encountered. This algorithm will also attempt to prioritize multiple monsters of the same type after 4 different monsters have been selected from the Monster Vault to minimize the number of monsters a Dungeon Master would have to be familiar with. After all monsters are selected, the algorithm will check this vs the determined thresholds for encounter difficulties and ensure that they fall within the desired ranges.

Once all terms without associated dictionary entries have been queried the script will loop through all dictionary entries and perform operations on them based on specific tags. Dictionary elements are formatted in a JSON dictionary format for ease of accessing their keys and values. After a check to ensure that a highlighted term matches a dictionary entry's "name" key, operations are performed based on the content of the JSON object.

If the value of "type" is NPC a trio comprised of a quirk, appearance, and personality for a desired non-player character will be generated. This is based on a random selection from the table entries that are stored in the Postgres database. This is returned in a list of JSON elements that required some conditioning to be accessed correctly. After retrieving it is stored in a dictionary for later use when it needs to be formatted and output.

Secondly, if the JSON contains a key of either "ability" or "action" a query for a random monster will be executed and returned based on the value of the key. For example, if "ability": "Spellcasting" is specified as an element of the JSON object then a query a monster that has the special ability of "Spellcasting" will be returned. Subsequently, if an "action" of "Parry" is specified then a random monster record will be returned that contains an action named "Parry." This record will be stored in a dictionary of monsters to be formatted into markdown text for output as well.

Finally, if the JSON contains a key of "CR" with a value of an integer an algorithm that scales the difficulty of a desired monster is executed. The name of the monster is queried and if available the record is returned to have the scaling algorithm performed on it. This algorithm adjusts the challenge rating which in turn adjusts the total Hit Points, overall accuracy and probability of abilities affecting enemies, and the amount of damage that the monsters can inflict with the abilities that they have access to. The goal in mind is to allow Dungeon Master's to use any monster they desired without having to crunch numbers and tweak statistics of the monster in order to pose a threat to their players.

Once all operations are complete the parser will format each scene with title followed by scene summary and any relevant monster statistics, NPC traits or random encounters that have been generated in a Markdown format (Appendix B). By utilizing helper functions that format each of the stored dictionaries into markdown formats that are easily consumable by the [Homebrewery](#), to produce nicely documented and readable output that has styling very similar to printed Dungeons and Dragons publications (Appendix C).

Results, Evaluation, and Reflection

Overall this project was a success in proving that this concept can be implemented. It has proven to be a useful tool in my personal use and has helped to streamline and reduce some of the busy work that is normally associated with preparing adventures for Dungeons and Dragons. Its effective at reducing copying of statistics and with the generation of random encounters without having to aggregate data from multiple sources as well as providing inspiration. However, one fault it does have is its high barrier to use for someone unfamiliar with programming. The project mostly relies on having files in specific locations and having a locally installed database which could be considered unwieldy to say the least. The ideal implementation for this specific project would be a web-based application with a cloud instance running the postgres database.

From an efficiency perspective, the Python script is relatively speedy, and it minimizes the number of database queries that are required to compile the required data for output. The database being hosted in a docker instance makes it easy enough to spin up on multiple machines, but with this configuration persisting data is not an option. As mentioned above, persisting the database and hosting it on the web for access would make much more sense from an ease of use perspective.

Adventure Day is effective at what it does do, that being collecting monster data and displaying it in a fashion that is simple to reference when running a game of Dungeons and Dragons. Changing the difficulty of monsters and building new encounters is very useful for any Dungeon Master preparing an upcoming adventure. Currently the implementation for scaling monsters is limited to changing static values which leave room for improvements. Additionally, the random encounters are indeed random which could be viewed as a detriment. Finding a way to associate specific types of creatures together to build a cohesive encounter based on monster traits or similarities would improve the effectiveness of the random encounters significantly.

Conclusions and Future Work

This project was an experiment for a tool that I had long imagined creating. Despite the flaws of Adventure Day, the process of creating it has helped me to gain valuable experience in design and implementation that I had sought when considering the CIS Program. Overall Adventure Day is something I'm proud of and look forward to expanding its functionality and iterating on its design.

Currently the input to Adventure Day is a text file that requires special annotations. This is a rather rigid design that is error prone and inflexible to a user's input. Implementing a strategy of detecting keywords or matches close to a highlighted term could provide this much needed flexibility. Considering Python's Natural Language Tool kit is something that was initially explored but sacrificed for expediency of implementation, however given the time I think it could be a valuable addition for detecting meanings and interpreting inputs that a user could provide to the program.

The user interface for this project is relegated to the command line which leaves an opportunity to improve by moving this script to a web application. Providing a graphical user interface and exposing it to the web would make Adventure Day much more accessible for other users. The database would also be in cloud storage which would allow Monster Vault's data to be persisted.

Adventure Day as a whole, is a monolithic piece of software that could benefit from being refactored into an engine and various modules that could add additional functionality. Its current state makes it difficult to alter without potentially changing other sections of code but if refactored into an engine that accepts, parses and formats adventure text could open opportunities for including additional modules to increase and expand functionality. I envision Monster Works as a module that would provide scaling algorithms for monster difficulty, allow addition and modification of abilities for monsters and even allow creation of monsters to be added to the database.

Finally, Adventure Day's output is a Markdown formatted text file that is capable of being interpreted by another application that beautifies the output. Bringing this functionality into the application or providing a way of shipping this output to an endpoint via a REST API would remove the extra step to achieve a well-designed output file for use at the table.

Bibliography

RPG Tools - Goblinist. (n.d.). Retrieved from <http://tools.goblinist.com/>

Donjon. (n.d.). Retrieved from <http://donjon.bin.sh/>

The Homebrewery. (n.d.). Retrieved from <https://homebrewery.naturalcrit.com/>

Dungeons & Dragons (n.d.). Retrieved from <http://dnd.wizards.com/dungeons-and-dragons>

Appendices

Appendix A: Adventure Day Input

In Search of Adventure

A small part of the massive set of modules B1-B9 from TSR from 1987. This is an adventure designed for **7th** level adventurers and will take them to 10th level after the adventure is completed. They'll confront the great king and his army of undead that now ravage the land. With some swift wit and even more luck the party may be able to overcome the odds and bring peace to the land once again.

Threshold

Threshold is a logging community of about 5000, ably ruled by Sherlane Halaran, baron and Patriarch of the Church. In Threshold, the merchant, **Clifton Caldwell** has purchased the Old Antilles Castle, five miles west of Threshold. He has posted advertisements in local taverns for "Qualified Eviction Personnel".

Hamlet

A small hamlet only a day's ride to the east was overrun by a **Hobgoblin** and his band of mercenaries several days ago. Everyone was killed. The party learns of this on their way to Caldwell's Castle. There is also a **Adult Red Dragon** that lairs there that may reward the Adventurers for removing a thorn it its side.

Caldwell's Castle

A **dungeon** clearing adventure. The adventure begins with the party standing outside the double doors leading into the castle. The doors are unlocked, and swing open easily. The party must find and eliminate all monsters including two insane mages with **spellcasting**.

== Dictionary ==

```
{"name":"Clifton Caldwell", "type": "NPC", "level": 10, "description": "Clifton is a merchant"}
{"name":"Hobgoblin", "type": "monster", "CR": 5}
{"name":"Archmage", "type":"monster", "race": "human", "ability": "spellcasting" , "CR": -2}
{"name":"Evil Spelluser", "ability": "spellcasting" , "CR": -2}
```

Appendix B : Adventure Day Output

```
1 # In Search of Adventure
2
3 A small part of the massive set of modules B1-B9 from TSR from 1987. This is an adventure designed for 7th level adventurers and will
4 take them to 10th level after the adventure is completed. They'll confront the great king and his army of undead that now ravage the land.
5 With some swift wit and even more luck the party may be able to overcome the odds and bring peace to the land once again.
6
7 Threshold
8
9 Threshold is a logging community of about 5000, ably ruled by Sherlane Halaran, baron and Patriarch of the Church. In Threshold, the
10 merchant, Clifton Caldwell has purchased the Old Antilles Castle, five miles west of Threshold. He has posted advertisements in local
11 taverns for "Qualified Eviction Personnel".
12
13 <div class='descriptive'>
14 Clifton Caldwell
15 - appearance: small/scrawny
16 - personality: envious
17 - quirks: artistic / dreamer / delusional
18 </div>
19 Hamlet
20
21 A small hamlet only a day's ride to the east was overrun by a Hobgoblin and his band of mercenaries several days ago. Everyone was
22 killed. The party learns of this on their way to Caldwell's Castle. There is also a Adult Red Dragon that lairs there that may reward
23 the Adventurers for removing a thorn it its side.
24
25 Hobgoblin
26 > humanoid, Lawful evil
27 >
28 > - Armor Class 18
29 > - Hit Points 61 (2d8)
30 > - Speed 30 ft.
31 >
32 > STR|DEX|CON|INT|WIS|CHA|
33 > :---:|:---:|:---:|:---:|:---:|:---:|
34 > 13 (13) |12 (12) |12 (12) |10 (10) |10 (10) |9 (9) |
35 >
36 > - Damage Vulnerabilities None
37 > - Damage Resistances None
38 > - Damage Immunities None
39 > - Condition Immunities None
40 > - Senses darkvision 60 ft., passive Perception 10
41 > - Languages Common, Goblin
42 > - Challenge Rating 5
43 >
44 > Hobgoblin - Martial Advantage:
45 > Once per turn, the hobgoblin can deal an extra 7 (2d6) damage to a creature it hits with a weapon attack if that creature is within 5 ft. of
46 an ally of the hobgoblin that isn't incapacitated.
47 >
48 > Actions
49 >
50 > Hobgoblin - Longsword:
51 > Melee Weapon Attack: +3 to hit, reach 5 ft., one target. Hit: 5 (1d8 + 1) slashing damage, or 6 (1d10 + 1) slashing damage if used with two
52 hands.
53 >
54 > Hobgoblin - Longbow:
55 > Ranged Weapon Attack: +3 to hit, range 150/600 ft., one target. Hit: 5 (1d8 + 1) piercing damage.
56 >
57 > page
```

Appendix C : Homebrew Format

IN SEARCH OF ADVENTURE

A small part of the massive set of modules B1-B9 from TSR from 1987. This is an adventure designed for 7th level adventurers and will take them to 10th level after the adventure is completed. They'll confront the great king and his army of undead that now ravage the land. With some swift wit and even more luck the party may be able to overcome the odds and bring peace to the land once again.

THRESHOLD

Threshold is a logging community of about 5000, ably ruled by Sherlane Halaran, baron and Patriarch of the Church. In Threshold, the merchant, **Clifton Caldwell** has purchased the Old Antilles Castle, five miles west of Threshold. He has posted advertisements in local taverns for "Qualified Eviction Personnel".

NPC: CLIFTON CALDWELL

Clifton is a merchant

- **appearance:** small/scrawny
- **personality:** envious
- **quirks:** artistic / dreamer / delusional

HAMLET

A small hamlet only a day's ride to the east was overrun by a **Hobgoblin** and his band of mercenaries several days ago. Everyone was killed. The party learns of this on their way to Caldwell's Castle. There is also a **Adult Red Dragon** that lairs there that may reward the Adventurers for removing a thorn it its side.

HOBGOBLIN

humanoid, lawful evil

Armor Class 18

Hit Points 61 (2d8)

Speed 30 ft.

STR	DEX	CON	INT	WIS	CHA
13 (13)	12 (12)	12 (12)	10 (10)	10 (10)	9 (9)

Damage Vulnerabilities None

Damage Resistances None

Damage Immunities None

Condition Immunities None

Senses darkvision 60 ft., passive Perception 10

Languages Common, Goblin

Challenge Rating 5

Hobgoblin - Martial Advantage: Once per turn, the hobgoblin can deal an extra 7 (2d6) damage to a creature it hits with a weapon attack if that creature is within 5 ft. of an ally of the hobgoblin that isn't incapacitated.

Actions

Hobgoblin - Longsword: Melee Weapon Attack: +3 to hit, reach 5 ft., one target. Hit: 5 (1d8 + 1) slashing damage, or 6 (1d10 + 1) slashing damage if used with two hands.

Hobgoblin - Longbow: Ranged Weapon Attack: +3 to hit, range 150/600 ft., one target. Hit: 5 (1d8 + 1) piercing damage.