



Senior Project

BY: ETHAN POWELL

What's the problem?

In Dungeons and Dragons, everyone involved must keep track of statistics, modifiers, hp values, and more, erasing and re-writing values constantly. The creation of a Dungeons and Dragons utilities tool is necessary to ensure smooth gameplay, relieve stress, and prevent errors when calculating values.



Motivation

- I enjoy tabletop role-playing games and have played fifth edition Dungeons and Dragons for nearly 4 years now.
- Previous tools I have used never felt quite right, having problems such as:
 - Difficult interface
 - Felt Clunky
 - Lacked functionalities I thought obvious
 - Paid to use
- The desire to create something I felt was my own project
- Personal use

Research and Background

- Deciding a language
 - C++?
 - C#?
 - Python?
- How GUI's work in these languages
- Availability of libraries
 - C++ is built for more “low-to the ground” applications, offering control
 - Python is intended for high level projects; is more user friendly
- Documentation
 - Python seems more robust in its library documentation because of the userbase
 - C++ and C# felt lacking in usability without a strong understanding of many base tools and libraries

Hardware/Software Requirements

Hardware

- An intel I3 processor or above (or it's AMD equivalent)
- 1GB Ram
- Secondary Storage (size depends on amount of data wanting to be stored)
- Keyboard for input

Software

- A Linux based operating system
- Python3
- SQLite/SQLAlchemy
- Pandas
- Orator

Demo

Functional Requirements

- Allow the storing and loading of character data for later reference
- Allow for modification of character data
- Allow for the creation of new character data
- Allow for the importing of character data from a standardized PDF
- Allow for the viewing of character data

Database

- This displays the database functionality, showing stored characters from the program.

The screenshot displays a database application interface. The main window shows a table of characters with columns for name, level, proficiency, ac, initiative, current_hp, max_hp, str, dex, con, int, wis, cha, str_save, dex_save, con_save, int_save, wis_save, cha_save, death_success, death_failure, and acrobat. The table contains 5 rows of data. On the right, an 'Edit Database Cell' window is open, showing the text 'Tom' in a text field. Below the text field, it says 'Type of data currently in cell: Text / Numeric' and '3 character(s)'. There is an 'Apply' button. Below the text field, there is a 'Remote' section with a dropdown menu for 'Identity' set to 'Select an identity to connect'. Below that, there are tabs for 'DBHub.io', 'Local', and 'Current Database'. Below the tabs, there is a table with columns 'Name', 'Last modified', 'Size', and 'C'. The bottom of the main window shows a status bar with '1 - 4 of 5' and a 'Go to: 1' button.

	name	level	proficiency	ac	initiative	current_hp	max_hp	str	dex	con	int	wis	cha	str_save	dex_save	con_save	int_save	wis_save	cha_save	death_success	death_failure	acrobat
1	Tom	20	6	18	3	27	45	22	16	10	8	10	18	6	3	0	-1	0	4	0	0	0
2	Boblin the Goblin	2	2	12	2	10	10	8	14	12	6	13	4	-1	2	1	-2	1	-3	0	0	0
3	Gwendolynn	20	6	23	4	165	165	22	18	17	13	14	20	6	4	3	1	2	5	0	0	0
4	Caoimhe	10	4	16	4	73	73	12	18	13	13	16	14	1	4	1	1	3	2	0	0	0
5	Artyom	4	2	12	1	0	28	6	13	14	18	8	13	-2	1	2	4	-1	1	0	0	0

Character Data Creation

- This displays the process of character creation with error messages

```
----- Create Character -----
```

```
Saved Characters:
```

```
Tom
```

```
Boblin the Goblin
```

```
Gwendolynn
```

```
Caoimhe
```

```
Artyom
```

```
Please type your new character's name or press ENTER to go back: Kindred
```

```
Please type Kindred's level: 20
```

```
Please type Kindred's current HP: 999
```

```
Please type Kindred's max HP: a
```

```
Please input an integer value.
```

```
Please type Kindred's max HP: 998
```

```
Maximum HP cannot be smaller than current hp.
```

```
Please type Kindred's max HP: 999
```

```
Please type Kindred's armor class: 20
```

```
Please type Kindred's strength score: 20
```

```
Please type Kindred's dexterity score: 20
```

```
Please type Kindred's constitution score: 20
```

```
Please type Kindred's intelligence score: 20
```

```
Please type Kindred's wisdom score: 20
```

```
Please type Kindred's charisma score: 20
```

```
----- Kindred's Statistics -----
```

```
Level: 20
```

```
HP: 999/999
```

```
Armor Class: 20
```

```
STR: 20 (+5)
```

```
DEX: 20 (+5)
```

```
CON: 20 (+5)
```

```
INT: 20 (+5)
```

```
WIS: 20 (+5)
```

```
CHA: 20 (+5)
```

```
Does this look right (y/n)? y
```

High-Level Test Plan

- Ensure the product can run on the specified hardware without interruption
- Ensure each functional requirement is met
 - Data viewing
 - Data creation
 - Data storage
 - Data editing
- Ensure input validation to prevent improper data entry
- Ensure integrity of the database

Test Results

- Project runtime performance
 - While running with only 1 thread, no performance issues were detected
- Functional requirement testing
 - Data viewing – The data is displayed accurately and without error
 - Data creation – The data is created without improper input
 - Data editing – The data is able to be edited without improper input
 - Data saving/loading – The data is able to be saved and loaded from the database file
- Graphical user interface
 - No graphical user interface was implemented

Challenges Overcome

- Language decisions and implementation method
- Interacting with a PDF
- Understanding program to database interaction
- How to properly write documentation

Challenges Not Overcome

- Time constraints and full implementation
- Utilizing a fully graphical user interface
- Implementing a way to interpret Dungeons and Dragons spells and abilities

Future Enhancements

- Implement full functionality
- Create a graphical user interface for use with mouse pointer and buttons
- Web hosting/mobile app production
- Clean and optimize code for functionality and speed

Questions?