Chasm of Chaos

**Description**

Chasm of Chaos is an endless, procedurally generated dungeon fighting game. In it, players assume the role of a dungeon explorer, trapped in the dungeon. They must navigate the labyrinth rooms of the many floors of the dungeon, fighting monsters, collecting loot, and finding 5 items of high value to escape. If they die, they must start from the first floor and work their way back down.

The player has access to their own inventory, where they can collect and manage items to use within the dungeon. These include potions that give them status effects which last for a certain number of rooms, or smoke bombs they can use to escape fights.

This game is inspired by other dungeon games such as *Enter the Gungeon*, *Hunt the Wumpus*, and *Dungeons & Dragons*. It was initially made in roughly 2 weeks for a Discord bot. As such, many of the gameplay elements were designed to be easy to implement and use with Discord interaction modals (buttons, select menus, etc.). This game takes the bot elements and turns them into interactive text using the specially developed fterm engine.

**Used Libraries**

* Python 3.8+ (Python 3.11 used in production)
* Curses
* fterm (builds off of Curses for dynamic text rendering in terminals)
* Original *Chasm of Chaos* for Discord bot

**Challenges**

One of the most prominent challenges (and arguably, the *biggest* challenge) is the complete UI redesign required. Initially, the game was built for use within Discord, and certain elements were designed for this. For example, each of the player’s actions (moving between rooms, accessing inventory items, etc.) are separate commands the player has to enter. By consolidating them into specially designed menus, less code has to be used to create the different views that would be necessary for Discord. However, one drawback this has is that certain data could be displayed at all times (such as enemy and player health in fights) within Discord embeds as separate fields. Within a terminal, each and every line is treated the same and has to be rendered individually. Lots of time needs to be spent finding a balance between showing enough information while not spending too much time showing *redundant* information.

Additionally, certain elements can be color-coded very easily within the terminal. With this, however, comes the need to keep everything visually consistent (such as all mentions of player HP being colored green, all enemy HP being colored red).

Other systems within the game, such as the planned shop system, may need certain elements reworked. For the Discord bot version, much of the UI was centered around using buttons and dropdown menus, and most of the gameplay was created with this UI in mind. For the shops, it would be simple to have a single dropdown menu with all the available items, and only add buttons later to purchase an item if the player was able to buy the item (if they had space in their inventory and enough money to do so). Since all the visual elements are removed by using a terminal, this system may need a bit of work to change. Additionally, the player currently has no way of earning money for these shops, as initially that was a completely separate system in the bot that the game didn’t need to handle. Without that system, an alternate system will need to be made and tested (perhaps giving a certain amount after defeating enemies, scaling up depending on how difficult the enemy is).

Finally, one challenge that will need to be addressed at some point in the process is maintaining player attention. Text elements and menus are significantly less appealing than buttons and visual menus, so some changes to the visual appeal of the game (or even the gameplay) will have to be made to ensure the player doesn’t get bored of the same gameplay within the first 20 minutes.