Technical Digest

Hello, and thanks for coming! This is the technical digest, as the name would suggest, for people who want a deeper look into the systems at play in the game. This assumes that you've already read the abbreviated version of this directly above where you found it, so there won't be any duplicate information here! No, siree!

Let's get right into it!

Container Rebrand

First things first, the new class of containers. Monsters are now always containers, so that's a cool one. It's just because at some point or another, you're going to need to access an inventory from them for one thing or another, and it's just easier to roll the functionality together. It's a direct inheritance, so monsters are containers but containers are no longer monsters. This was mainly to reduce the operating weight of the game and to support in the setup of the loot tables, which are now fully implemented.

Loot Tables

I didn't touch on this in the devlog at all because it's entirely technical, but I finished setting up the main loot table tree for standard chests. The function to handle populating chests takes in a loot table, runs a weighted random based on that loot table, and looks up whatever item is needed. In TfQuest, a big change is being able to have multiple items in each chest, so watch out for that too.

Inheritance System

This is the big time sink for this week and also the most complex system. Let me give an example of some of the content that I've been working on.

Here's the default traits list for monsters:

```
"DefaultTraits": {
    "Height                  ": "set 165",
    "Mass                     ": "set 75",
    "Width                    ": "set 38",
    "Health                   ": "set 5",
    "Strength                 ": "set 5",
    "Dexterity                ": "set 5",
    "Intelligence             ": "set 5",
    "Luck                     ": "set 5",
    "Wisdom                   ": "set 5",
    "Charisma                 ": "set 5",
    "OutrageThreshold         ": "set 100"
    "Breast Volume            ": "set 650",
    "Penis Length             ": "set 13",
    "Ass Volume               ": "set 100",
    "Hip Width                ": "set 40"
},
```

You can see a list of defined variables and a modifier afterwards. In this case, the stat block will set the values of all of these variables to a specific number. These are starting points that are used as the foundation of the other modifier blocks. Now let me show you another version of the same part for an elf:

```
"DefaultTraits": {
    "Height                  ": "add 15",
    "Weight                   ": "add -10",
    "Width                    ": "add -6",
    "Health                   ": "mult 1.25",
    "Strength                 ": "add -1",
    "Dexterity                ": "add 2",
    "Charisma                 ": "add 4",
    "Wisdom                   ": "add 4",
    "OutrageThreshold         ": "mult 0.5",
}
```

This is a good showcase of most of the different modifiers you can use. In this case, you'd define an elf to be [Monster, Elf]. The script would start at monster, set all of the values, and then make the modifications listed in order to get the final result. Of course, this can get much more complicated, but it's best to keep examples simple.

This is also used for status effects as well as direct 1:1 substitutions like demonstrated previously. You can have a potion item that becomes a status effect and is applied to someone on use, for instance:

```
"Potion of Breast Growth": {
    "Breast Volume              ": "mult 1.5"
}
```

It's an incredibly powerful system.

For a quick rundown of the modifiers so far (I'm of course down to add more if anyone can think of useful ones):

- Set: Sets the value to a chose number
- Add: Adds a number to the current value
- Mult: Multiplies by a chosen value
- Final: Overrides all other inputs to return this value instead

In terms of content generation, I hope for TfQuest to be much more in-depth and detailed than TQ. The engine limitation harnesses are off, so I'm writing in a lot to try and add depth to the characters, transformations, and world overall. Some of the behaviours are still going to have to be written explicitly for each monster, but this will allow for a lot of 'easy' depth by adding layers.

The content I'm looking at first are all of the necessary layers to make the dungeon, starting with the monsters. I'm hoping to go in order by region and get all of the puzzle pieces I need one-by-one. Hopefully by getting the difficult things out of the way, I'll have a lot of foundation by the time that I move to higher regions!

As always, let me know if you have any questions about anything! I'm down to talk shop, whether for clarifications, new ideas, or anything else. Thanks for reading! I really appreciate your support.