Phase 1 Better - Emotion Formula Simplification

For this phase, I wanted to take a simpler, testable approach to how we calculate player emotions - specifically **Satisfaction** and **Frustration**, after each pack opening.
The goal is to start small, make sure our logic behaves predictably, and then scale it up once we understand how all the pieces interact.

Starting Point: Simplifying the Setup

We're narrowing everything down to just three pack types - **Bronze, Silver, and Gold** - and five rarities of cards: **Common, Uncommon, Rare, Epic, and Legendary**.

Each pack will contain exactly **three cards** for now.

That gives us an easy framework to reason about without too many variables. Here's how the drop probabilities look: Vrushali_Phase1

CARD PACKS AND CARD TYPES FOR PHASE 1

Pack Type	Common	Uncommon	Rare	Epic	Legendary
Bronze	75%	15%	10%	0%	0%
Silver	0%	65%	20%	10%	5%
Gold	0%	0%	60%	30%	10%

BEST AND WORST POSSIBLE OUTCOMES

Pack Type	Worst	Best		Total cards	
Bronze	3- common	2 uncommon, 1 rare		3 total	
Silver	3 uncommon	1 rare, 1 epic, 1 legendary		3 total	
Gold	3- rare	2 epic, 1 legendary		3 total	

Fig: To frame expectations, we also defined what *worst* and *best* possible outcomes mean for each pack

Step 1. Quantifying Rarity

We'll give each rarity a simple numeric value:

```
Common = 1
Uncommon = 2
Rare = 3
Epic = 4
Legendary = 5
```

When a player opens a pack, we add up the three card values to get a **raw score**.

Example: Rare (3) + Rare (3) + Epic (4) = 10.

Step 2. Establishing Each Pack's Range

From the "best" and "worst" cases we just defined, we can work out the lowest and highest possible raw scores per pack:

Pack	**Min (worst)**	**Max (best)**
Bronze	3 (1+1+1)	7 (2+2+3)
Silver	6 (2+2+2)	12 (3+4+5)
Gold	9 (3+3+3)	13 (4+4+5)

These ranges show how different the packs are - a Gold pack's *worst* result (9) is already higher than Bronze's *best* (7).
That difference is exactly why we can't just use raw numbers directly.

Step 3. Why We Need Normalization

Let's compare two actual **pulls**:

- A **Bronze** pack hits its *best* possible combo → 2 Uncommons + 1 Rare = 2 + 2 + 3 = **7**
- A **Gold** pack has a *worst* result → 3 Rares = 3 + 3 + 3 = **9**

Now what happens is:
Bronze's **best** case (7) is *lower* than Gold's **worst** case (9).
If we just used these raw scores, the system would assume the Gold pack was *better* - even though for Gold that's literally its *baseline*.
We'd end up rewarding the wrong emotional response.

To fix that, we "normalize" each result between its own minimum and maximum.
That simply means: *how far is this outcome between the worst and best that this pack could ever produce?*

$$quality01 \ = \ \frac{rawScore - minScore(pack)}{maxScore(pack) - minScore(pack)}$$

This gives a clean, universal value between 0 and 1:

- **0** → you hit the absolute worst case for that pack
- **1** → you hit the absolute best
- **0.5** → a middle-of-the-road result

Now every pack's result can be compared on the same emotional scale.

Step 4. Mapping Quality to Emotions

Once we have that normalized `quality01` value, we can translate it into changes in the player's emotional state.

We define two tunable constants:

`S_max = maximum Satisfaction gain per pack`

`F_max = maximum Frustration gain per pack`

Then we calculate:

$$Satisfaction = quality01 \times S\_max$$

$$Frustration = (1 - quality01) \times F\_max$$

So:

- A high-quality pull (close to 1) gives a big Satisfaction boost and almost no Frustration.
- A poor pull (close to 0) spikes Frustration but barely increases Satisfaction.
- Mid-range pulls sit in the middle.

For early testing we can set **S_max = F_max = 10** just to watch the numbers move and make sure the logic feels right.

Example

Let's walk through a full example with that same **Gold** pack to see how the math works.

Pack: Gold
Cards Pulled: 3 Rares (3 + 3 + 3 = 9)

1. **Raw Score** = 9
2. **Gold Range:** min = 9, max = 13
3. **Normalize:** (9 − 9) / (13 − 9) = 0 → this pull is the *lowest* possible for Gold
4. **Apply Emotions** (if S_max = F_max = 10):
   - Satisfaction = 0 × 10 = +**0**
   - Frustration = (1 − 0) × 10 = +**10**

Now if we compare it to the **Bronze** best-case pull (7):

Pack: Bronze
Cards Pulled: 2 Uncommons + 1Rare (2 + 2 + 3 = 7)

1. **Raw Score** = 7
2. **Bronze Range:** min = 3, max = 7
3. **Normalize:** (7 − 3) / (7 − 3) = 1.0 → this pull is the *best* possible for Bronze
4. **Apply Emotions** (if S_max = F_max = 10):
   - Satisfaction = 1 × 10 = +**10**
   - Frustration = (1 − 1) × 10 = +**0**

That clearly shows how normalization fixes the problem: even though the Gold raw score (9) looked higher, it translates to *zero satisfaction* because it was the *worst for Gold*.