Crafting a fairer Inferno with Forbidden Spreadsheet Lore

Simulating Eternal Damnation with Spreadsheets

Prologue

How to balance
a complex strategy game

Who am I?

- I'm **William Dyce** ☺
- I've worked in games for nearly 11 years now, including 7 as an **AI and systems designer** at *Amplitude Studios*.
  - In France I worked *Humankind, Endless Space 2, Endless Legend,* and a few titles you've not heard of.
- For about a year I've been working on ***Solium Infernum*** at *League of Geeks*.
  - *Solium* is a **turn-based diplomatic grand-strategy game** set in Hell.

Tuning

- Today's talk is about **strategy game *balancing***: the process of figuring out what these numbers should be →

- I prefer the word *"tuning"*, as *"balance"* suggests that it's all about fairness.

  We *do* want ***Fairness***, but also-

  - ***Pacing***,

  Variety,

  Resonance, … etc.

Rational tuning

- We'll be focusing on **how mathematical models can *empower* us** in the pursuit of our chosen goals.

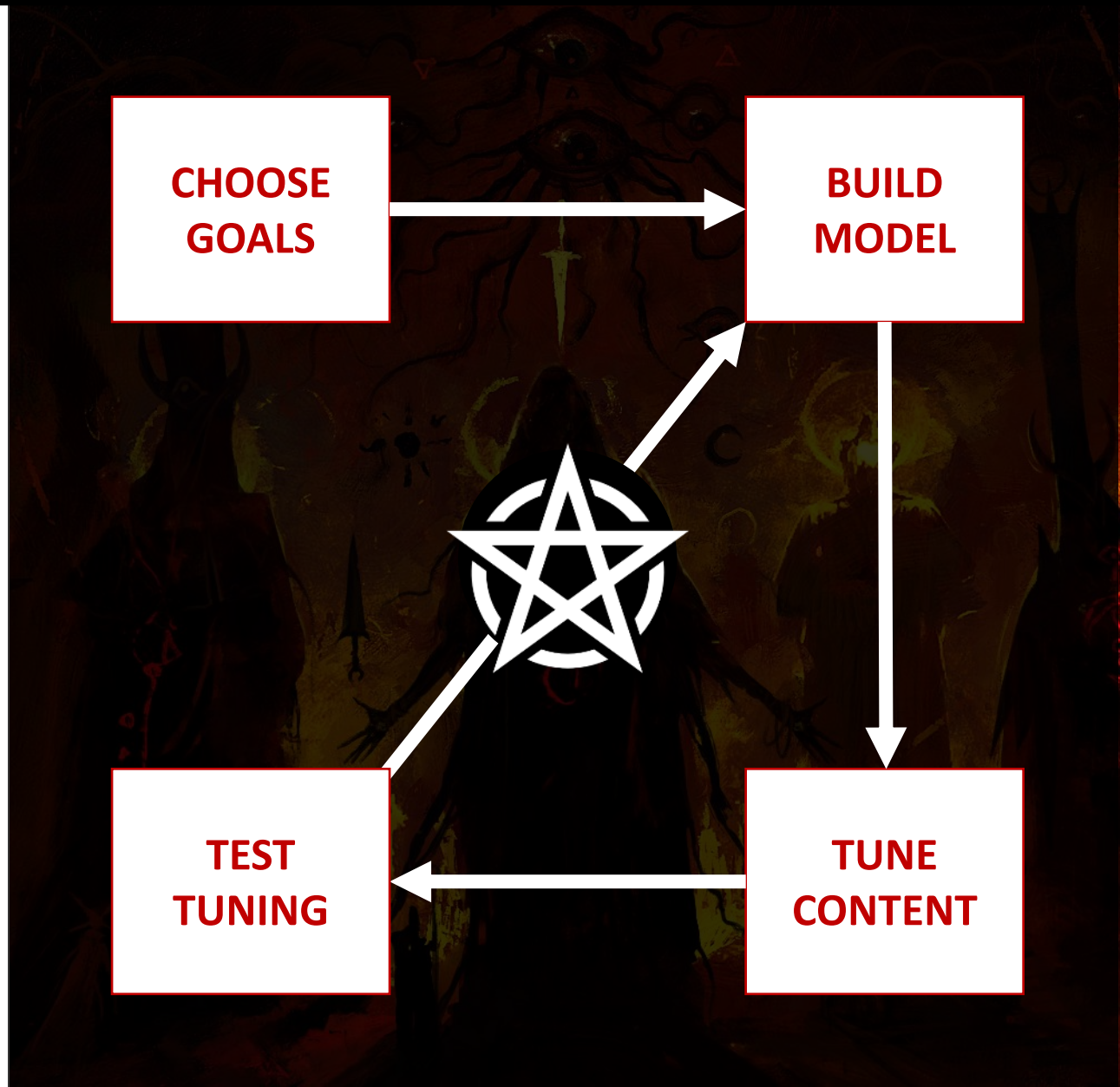
- I like to call this ***"rational"* tuning**, a nod to *Ubisoft's "rational level design".* RLD involves modelling **how difficult encounters are**, based the combination of game objects that are a part of them.

- A common example of RT would be modelling **the cost and the benefit of each piece of content** so that it can be placed on a **"power curve"**. This makes unbalanced content clearly visible →

Rational tuning process

For *Solium*, the baleful tuning ritual has tended to look something like this →

1. **Choose** goals for the tuning pass.
2. **Build** a model that tells us how well the set of numbers we've chosen accomplishes these goals.
3. **Tune** the content based on this model's evaluations.
4. **Test** the tuning in-game and revise the model if necessary.



Tool-assisted rational tuning

- A good model **outlives the specific tuning-pass** it was created to assist with.
- Such a model **grows and evolves with the game** as it advance through its development cycle.
- For this to be possible, the model must updated run whenever **our content's *"source of truth"* reference** is modified.
- This means that **automation is necessary**: we need *infernal, number-crunching machines* that run sanity-checks on each tuning pass.

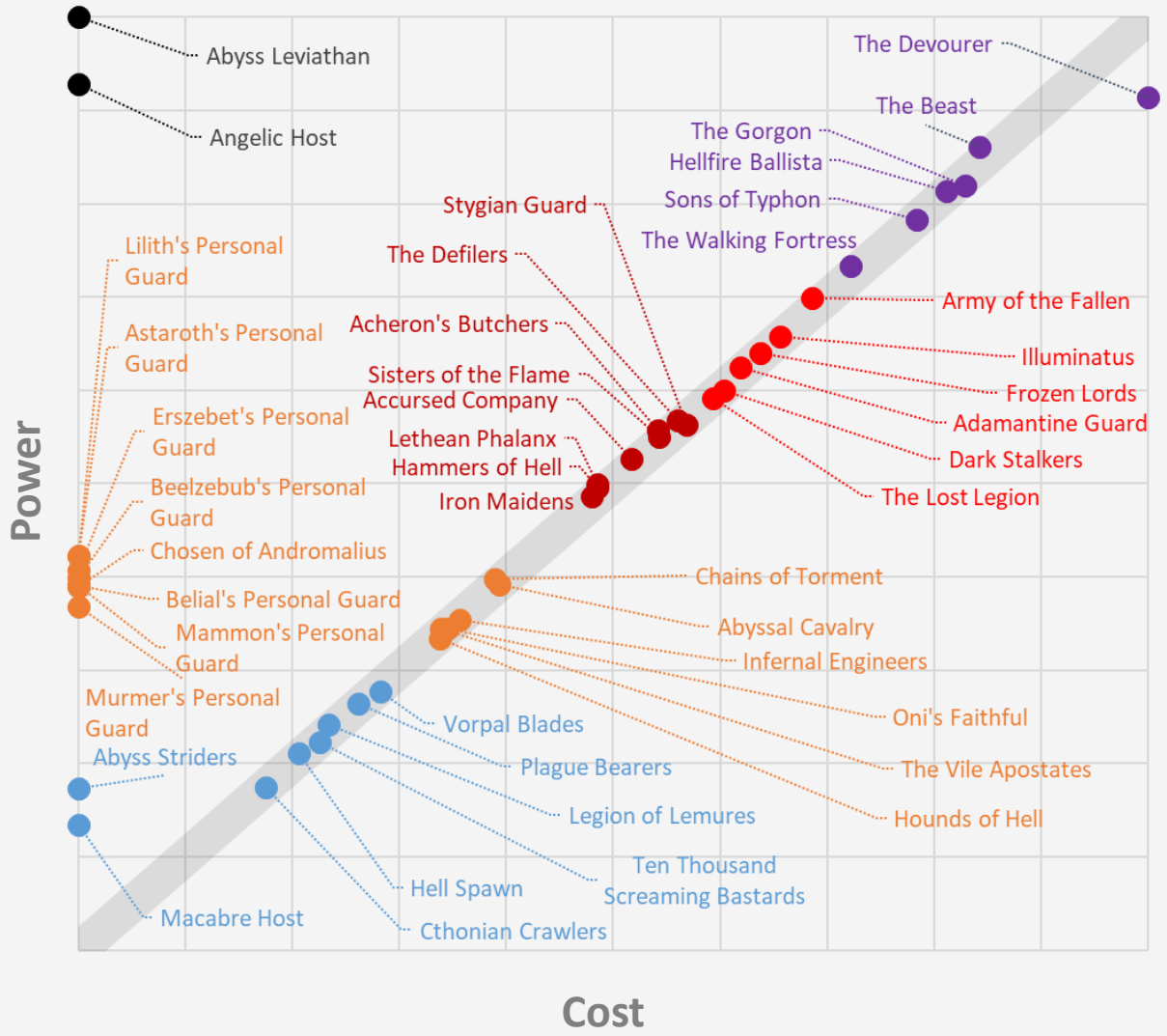
Tools for rational tuning

- My preferred tool for modelling is ***Microsoft Excel***: it's fast, you can copy-paste data in and out of it, and it has good tools for visualising your results →

- ***Google Sheets*** is much slower and is missing a few key feature that I'll discuss later in the talk. It is **"free"** though.

- *Both* support **real-time multi-user access and editing** online.

- Others to consider: ***Progressimo***, ***McDie***, and ***Machinations***…



Book I

Game power heuristics
and book-keeping in Excel

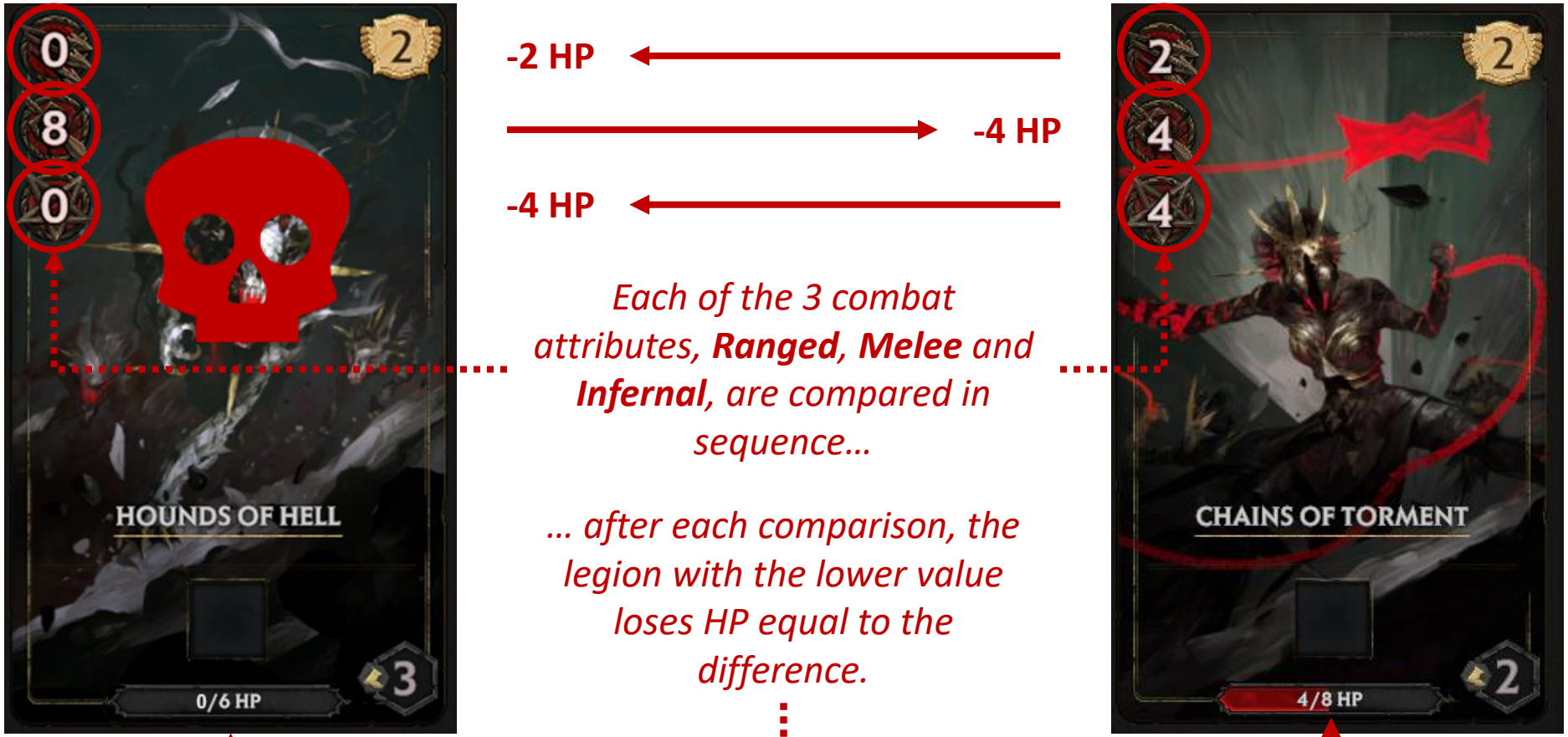
Combat in *Solium Infernum*

My first task on *Solium* was to give the game's military units their attributes. These ***"legions"* derive value mainly from how well they fare in combat**.

1. Ranged

2. Melee

3. Infernal



*Each of the 3 combat attributes, **Ranged**, **Melee** and **Infernal**, are compared in sequence…*

… after each comparison, the legion with the lower value loses HP equal to the difference.

Implicit intersection operator @

If a legion is destroyed before the end of the battle, **the remaining phases are skipped** entirely. This means **the early-phase attributes are worth more** than the later-phase ones.

=4*[@Level] + 8*[@Hitpoints] + 12*[@Ranged]
+ 11*[@Melee] + 9*[@Infernal] + 30*[@Speed]

	A	B	C	D	E	F	G	H	I
1	Index	Name in English	Level	Hitpoints	Ranged	Melee	Infernal	Speed	Power heuristic
2	1	*Hounds of Hell*	2	6	0	8	0	3	[@Speed]
3	2	*Stygian Guard*	3	8	6	4	8	2	324
4	3	*Chains of Torment*	2	8	2	4	4	2	236
5	4	*Vile Apostates*	2	9	0	0	14	1	236
6	5	*Oni's Faithful*	2	6	5	4	0	2	220

Above is a first attempt to assign a power value to each legion. Thanks to *Excel*'s ***"implicit intersection"* operator, @**, we can read the formula without getting a nosebleed.

Many-to-one relationship

Legions can have a special ability that must be taken into account!

Many-to-one relationship

Legions can have a special ability that must be taken into account! Assuming we can put a value on an ability, how do we **incorporate it into the values of the legions that use it**?

XLOOKUP

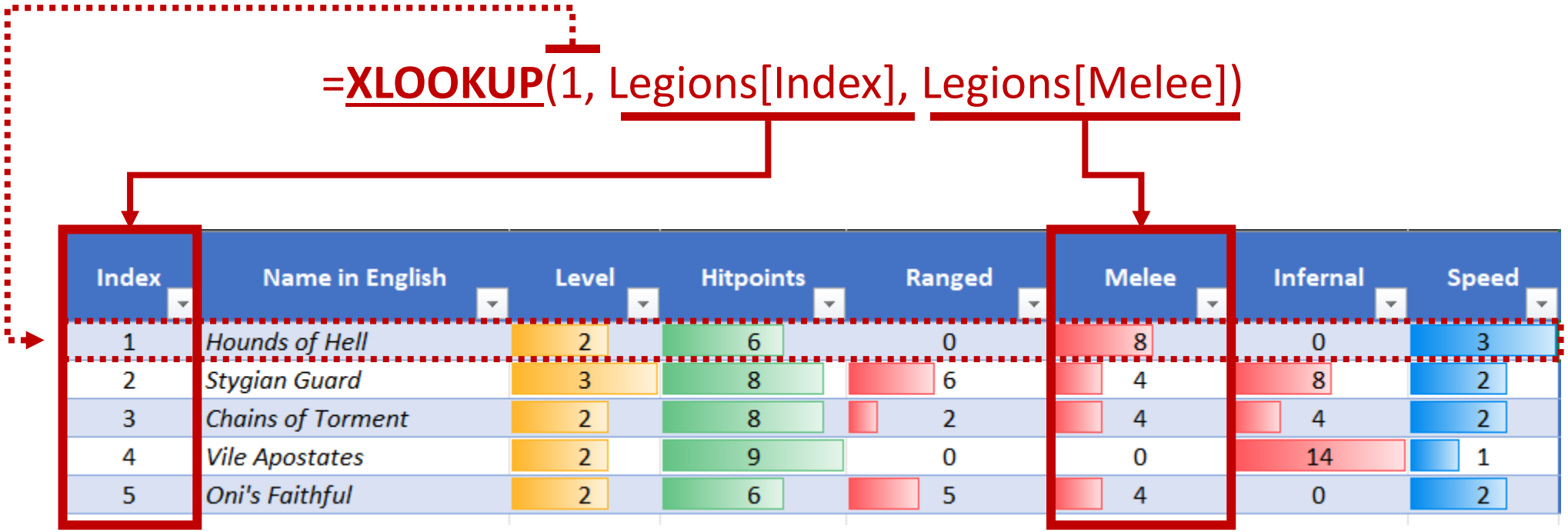
Index	Name in English	Power value
1	*Teleport*	50
2	*Arcane Bulwark*	5
3	*Lava Walker*	30
4	*True Sight*	10
5	*Ranged vulnerability*	-30

Index	Name in English	Power heuristic from attribute	Ability index
1	*Hounds of Hell*	234	
2	*Stygian Guard*	324	
3	*Chains of Torment*	236	
4	*Vile Apostates*	236	1
5	*Oni's Faithful*	220	

XLOOKUP and structured references

XLOOKUP to the rescue! It can be used to **find the value of given record's attribute.**

For instance *"find the melee value of Legion number 1"* looks like this in *Excel*:



Index	Name in English	Level	Hitpoints	Ranged	Melee	Infernal	Speed
1	*Hounds of Hell*	2	6	0	8	0	3
2	*Stygian Guard*	3	8	6	4	8	2
3	*Chains of Torment*	2	8	2	4	4	2
4	*Vile Apostates*	2	9	0	0	14	1
5	*Oni's Faithful*	2	6	5	4	0	2

The square braces denote a **structured reference**: **named ranges** that *Excel* creates based on the names of your tables and column headers. *Sheets* doesn't have them – boo!

XLOOKUP and structured references

So we can use **XLOOKUP** to **grab the corresponding ability's power value** and add it to the Legion's overall value.

=**IFERROR**(**XLOOKUP**(@[Ability index], Abilities[Index], Abilities[Power value]), 0)

Index	Name in English	Power valu
1	*Teleport*	50
2	*Arcane Bulwark*	5
3	*Lava Walker*	30
4	*True Sight*	10
5	*Ranged vulnerability*	-30

Index	Name in English	Power heuristic from attribute	Ability index	Power heuristic from ability
1	*Hounds of Hell*	234		0
2	*Stygian Guard*	324		0
3	*Chains of Torment*	236		0
4	*Vile Apostates*	236	1	50
5	*Oni's Faithful*	220		0

XLOOKUP will return an error if it can't find anything: we can **use IFERROR to provide a default value** when a problem is encountered.

Many-to-many relationship

I would **not** recommend creating multiple columns for handling multiple abilities, as doing so will **make all future formulas considerably harder to read and write**.

SUM

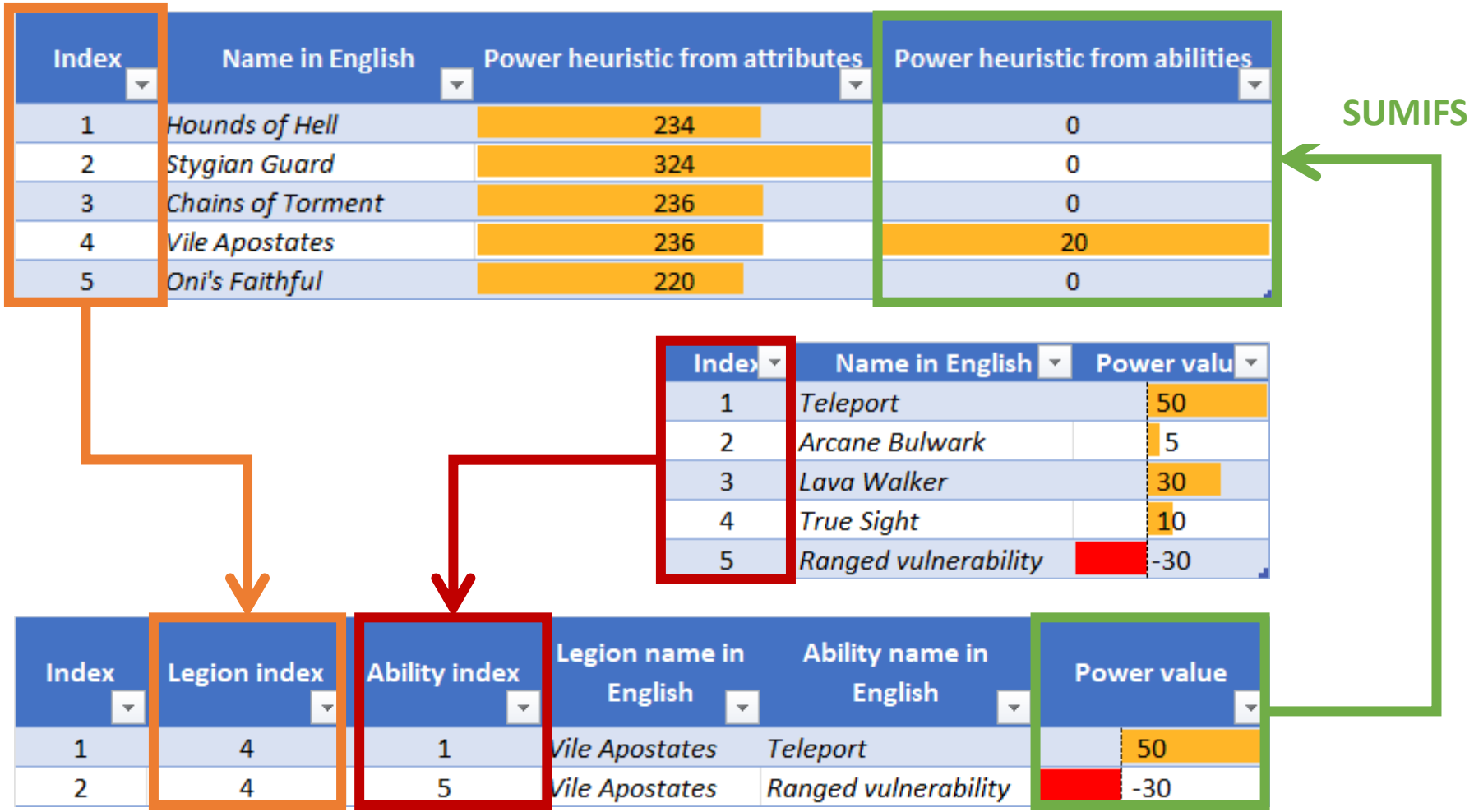
```
=IFERROR(XLOOKUP(@[Ability 1], Abilities[Index], Abilities[Power value]), 0) +
IFERROR(XLOOKUP(@[Ability 2], Abilities[Index], Abilities[Power value]), 0) +
IFERROR(XLOOKUP(@[Ability 3], Abilities[Index], Abilities[Power value]), 0) +
IFERROR(XLOOKUP(@[Ability 4], Abilities[Index], Abilities[Power value]), 0) +
IFERROR(XLOOKUP(@[Ability 5], Abilities[Index], Abilities[Power value]), 0)
```

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Index	Name in English	Level	Hitpoints	Ranged	Melee	Infernal	Speed	Ability 1	Ability 2	Ability 3	Ability 4	Ability 5	Power heuristic from abilitie
2	1	*Hounds of Hell*	2	6	0	8	0	3						value]), 0)
3	2	*Stygian Guard*	3	8	6	4	8	2						0
4	3	*Chains of Torment*	2	8	2	4	4	2						0
5	4	*Vile Apostates*	2	9	0	0	14	1	1	5				20
6	5	*Oni's Faithful*	2	6	5	4	0	2						0

The way we structure our data will have a huge impact on how easy it is to manipulate and to analyse: **rational tuning will be impossible without data hygiene**.

Many-to-many relationship

I would instead recommend **using an *"associative table"***: a third table that links legions to abilities. This allows us to use a single **SUMIFS** to get the total power from relevant abilities.



Index	Name in English	Power heuristic from attributes	Power heuristic from abilities
1	*Hounds of Hell*	234	0
2	*Stygian Guard*	324	0
3	*Chains of Torment*	236	0
4	*Vile Apostates*	236	20
5	*Oni's Faithful*	220	0

Index	Name in English	Power valu
1	*Teleport*	50
2	*Arcane Bulwark*	5
3	*Lava Walker*	30
4	*True Sight*	10
5	*Ranged vulnerability*	-30

Index	Legion index	Ability index	Legion name in English	Ability name in English	Power value
1	4	1	*Vile Apostates*	*Teleport*	50
2	4	5	*Vile Apostates*	*Ranged vulnerability*	-30

SUMIFS

SUMIFS adds together the attributes of records that meet a set of conditions. For instance, this is how we write *"the sum of values of this legion's abilities"*.

Index	Name in English	Power heuristic from attributes	Power heuristic from abilities
1	*Hounds of Hell*	234	0
2	*Stygian Guard*	324	0
3	*Chains of Torment*	236	0
4	*Vile Apostates*	236	20
5	*Oni's Faithful*	220	0

SUMIFS

=**SUMIFS**(LegionAbilities[Power value], LegionAbilities[Index], [@Index])

Index	Legion index	Ability index	Legion name in English	Ability name in English	Power value
1	4	1	*Vile Apostates*	*Teleport*	50
2	4	5	*Vile Apostates*	*Ranged vulnerability*	-30

TEXTJOIN

TEXTJOIN works a lot like SUMIFS, but **handles text instead of numbers**. Here's how we write *"this legion's abilities' names, separated by commas"*.

Index	Name in English	Power heuristic from attributes	Power heuristic from abilities	Ability names in English
1	*Hounds of Hell*	234	0	
2	*Stygian Guard*	324	0	
3	*Chains of Torment*	236	0	
4	*Vile Apostates*	236	20	Teleport, Ranged vulnerability
5	*Oni's Faithful*	220	0	

=**TEXTJOIN**(", ", TRUE, IF(LegionAbilities[Legion index]=[@Index], LegionAbilities[Ability name in English], ""))

Index	Legion index	Ability index	Legion name in English	Ability name in English
1	4	1	*Vile Apostates*	*Teleport*
2	4	5	*Vile Apostates*	*Ranged vulnerability*

TEXTJOIN, SUMIFS and COUNTIFS for sanity

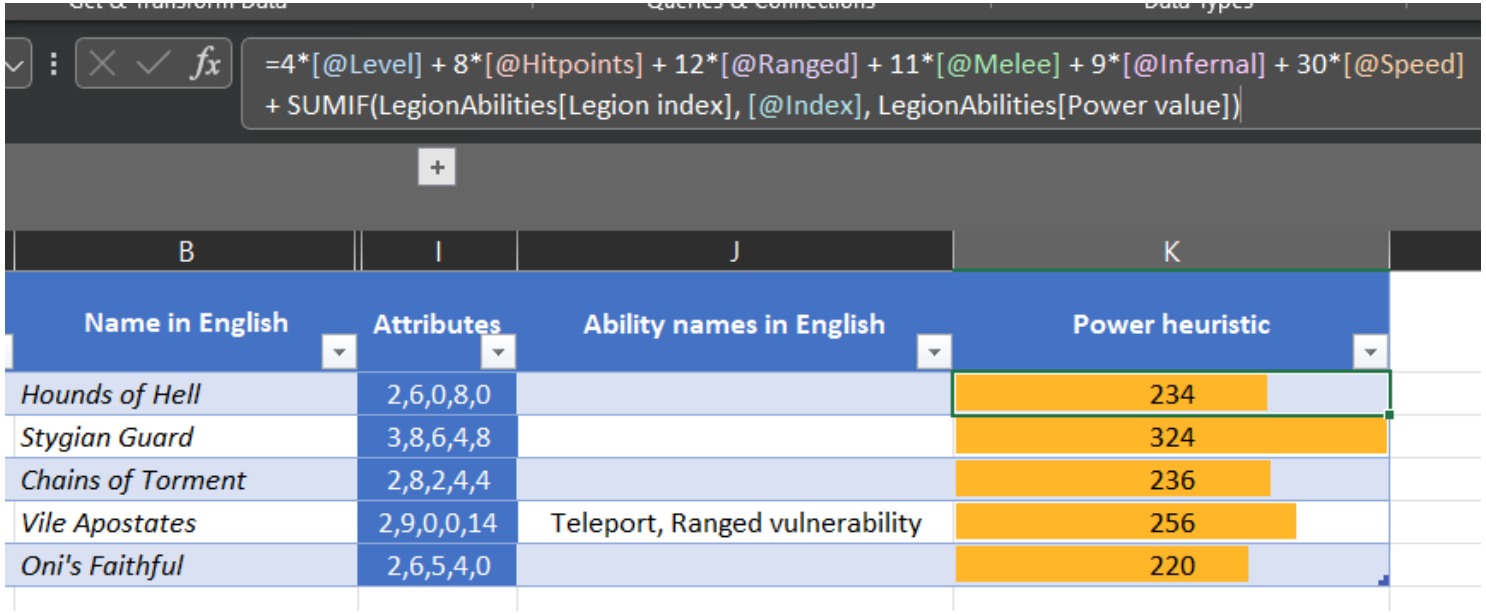
There are myriad applications for formulas that find and count, sum, or concatenate matching values: they are excellent for **making automated sanity checks**.

		COUNTIFS	SUMIFS	TEXTJOIN	TEXTJOIN	TEXTJOIN	TEXTJOIN
A	B						
Index	Name	*Total providers count*	*Sum of provider levels*	*Artifact providers*	*Place Of Power providers*	*Power providers*	*Relic providers*
6	Unlock Anointed of Ash	0	0				
7	Charisma Power	**4**	5	Key of Solomon Faust's Contract	The Garden of Infernal Delights		
8	Command Rating	**6**	8	Bottle of Whispers	Beelzebub's Stronghold	Wrath_Level_2 Wrath_Level_5 Wrath_Level_6	Ring of Command
9	Administrator	**1**	1				Crown of the Administrator
10	Potentate	0	0				
11	Kingmaker	**1**	0				Crown of the Kingmaker

Example: the engineers and UI designers may want to know how many legions a player might field in the worst-case scenario, if the player min-maxes their legion count.

Success?

We now have **a power heuristic for legions that takes into account both its attributes and its abilities**. We're all good to create our power curve now… right?



B	I	J	K
Name in English	Attributes	Ability names in English	Power heuristic
Hounds of Hell	2,6,0,8,0		234
Stygian Guard	3,8,6,4,8		324
Chains of Torment	2,8,2,4,4		236
Vile Apostates	2,9,0,0,14	Teleport, Ranged vulnerability	256
Oni's Faithful	2,6,5,4,0		220

Book II

Why poor system resilience is great,
but also terrible

“Anchors” for tuning

- **Assigning abstract values by hand** to *Solium’s* many *game-breaking* special abilities is more an art than a science… and more a folly than an art.

- The received wisdom: find **a common currency to convert everything** into. *Damage-per-second* is an example of this kind of *“value anchor”.*

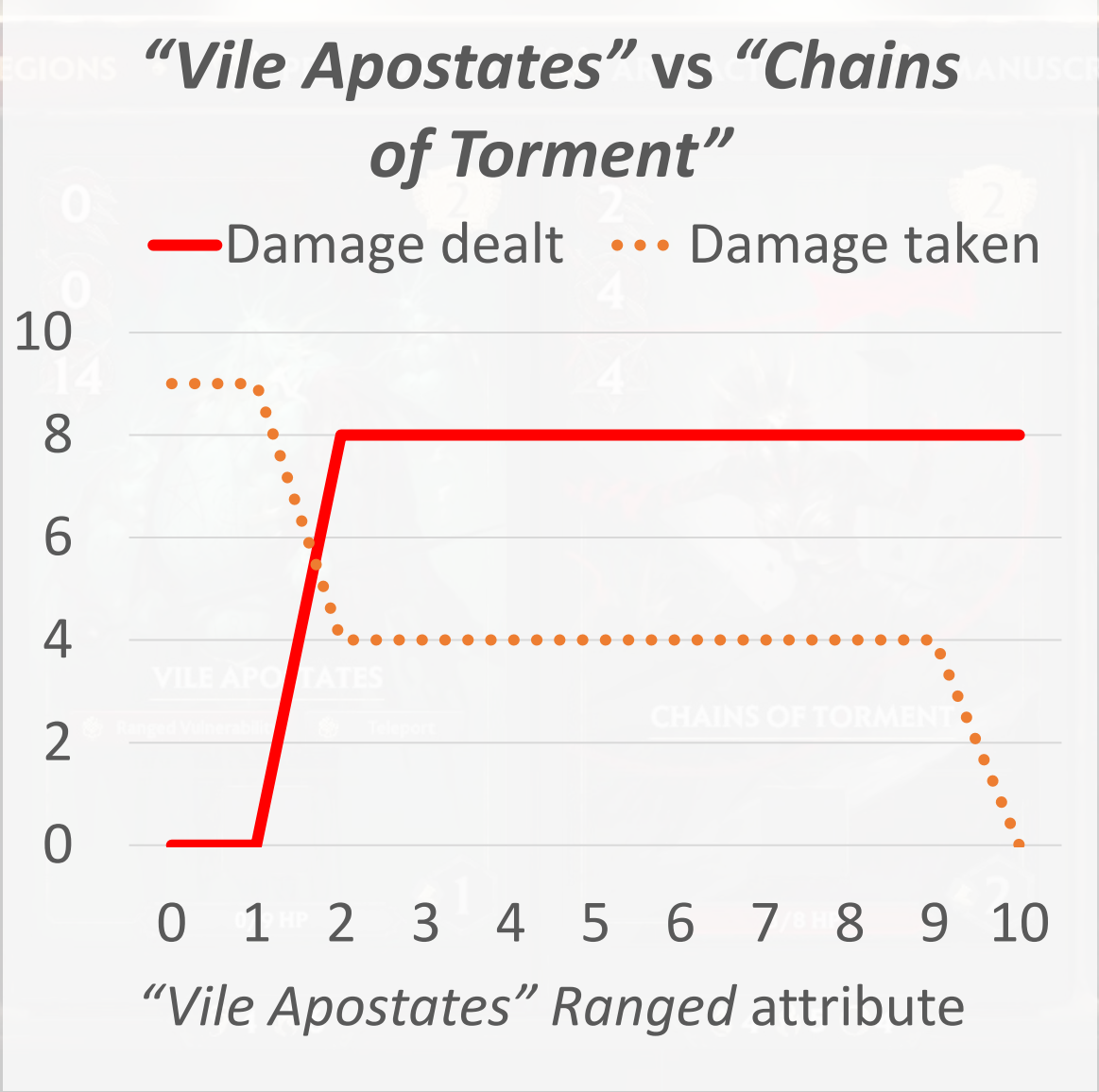
- It’s certainly a lot easier to see that something is *“off”* when your model is **evaluating content using a *concrete* unit of measurement** rather than a purely *abstract* one.

Anchor limitations

- Unfortunately, there are **contexts in which a high attribute or powerful ability is literally worthless** in *Solium*.
- **Example:** *"Vile Apostates"* has an extremely high *Infernal* value…
- … but will be killed by *"Chains of Torment"* without dealing any damage…
- … unless their *Ranged* increased by 2!

Anchor limitations

- Unfortunately, there are **contexts in which a high attribute or powerful ability is literally worthless** in *Solium*.
- **Example:** *"Vile Apostates"* has an extremely high *Infernal* value…
- … but will be killed by *"Chains of Torment"* without dealing any damage…
- … unless their *Ranged* increased by 2!
- We can't really use an *anchor*, **because you won't find a linear function** that can accurately model these *"tipping points"*.



Tipping points

- A *“tipping point”* is a critical threshold of value which, when crossed, causes a system to **shift from one state to another** – ARPGs are rife with examples.

- Exploiting such a threshold creates a **visceral, primordial joy** – *it may be why your cat knocked your favourite mug off your desk...*

- These thresholds **turn a difference in quantity into a difference in kind** by changing how input maps to output.

- This makes them **very difficult to tune**…

Tipping points in Solium Infernum

- Tipping points are particularly common in *Solium Infernum* – **its systems have limited *"resilience"***.
- This is intentional: high stakes and few actions nudge players towards **rumination, paranoia and betrayal**.
- They help **make the world feel dark and hostile**, and the player feel clever.
- They are why we can call *Solium* ***"the Dark Souls of strategy games"***… but they make it a *Hell* of a game to balance! What, then, to do?

Book III

Welcome to Fight Club,
aka simulate all the things

Every possible fight

There are 70 combatants in *Solium Infernum*, so $70^2 = 4900$ possible match-offs. What if we created a table of ***every* possible fight** that might occur?

C2 =MOD([@Index]-1, Constants[Contender count])+1

	A	B	C	D	E	F
1	Index	Attacking contender index	Defending contender index	GUID	Attacker name	Defender name
2	1	1	1	Fight_1V1	*Legion of Lemures*	*Legion of Lemures*
3	2	1	2	Fight_1V2	*Legion of Lemures*	*Hell Spawn*
4	3	1	3	Fight_1V3	*Legion of Lemures*	*Cthonian Crawlers*
5	4	1	4	Fight_1V4	*Legion of Lemures*	*Plague Bearers*
6	5	1	5	Fight_1V5	*Legion of Lemures*	*Vorpal Blades*
7	6	1	6	Fight_1V6	*Legion of Lemures*	*Ten Thousand Screaming Bastards*
8	7	1	7	Fight_1V7	*Legion of Lemures*	*Hounds of Hell*
9	8	1	8	Fight_1V8	*Legion of Lemures*	*Abyssal Cavalry*
10	9	1	9	Fight_1V9	*Legion of Lemures*	*Oni's Faithful*
11	10	1	10	Fight_1V10	*Legion of Lemures*	*The Vile Apostates*
12	11	1	11	Fight_1V11	*Legion of Lemures*	*Infernal Engineers*
13	12	1	12	Fight_1V12	*Legion of Lemures*	*Chains of Torment*
14	13	1	13	Fight_1V13	*Legion of Lemures*	*Iron Maidens*
15	14	1	14	Fight_1V14	*Legion of Lemures*	*Sisters of the Flame*

Every possible sequence override

Certain abilities can override the combat sequence and, say, cause the *Ranged* phase to repeat. Another table is used to look up **which attribute we'd need to check** in each round.

Name	Round	Attribute
Reverse	1	Infernal
Reverse	2	Melee
Reverse	3	Ranged
Reverse	4	None
Reverse	5	None
Reverse	6	None
Ranged Twice	1	Ranged
Ranged Twice	2	Ranged
Ranged Twice	3	Melee
Ranged Twice	4	Infernal
Ranged Twice	5	None
Ranged Twice	6	None

Every possible sequence override conflict

There can even be **conflicts between sequence overrides**: in such cases we use another lookup table to decide which to use, based on the contenders' respective levels.

GUID	Resolution
Ranged Last=Melee Last	Melee Last
Ranged Last=Ranged Twice	Ranged Last Twice
Ranged Last>Melee First	Ranged Last
Melee Last=Ranged Last	Ranged Last
Melee Last=Ranged Twice	Ranged Twice Melee Last
Melee Last>Melee First	Melee Last
Ranged Twice=Ranged Last	Ranged Last Twice
Ranged Twice=Melee Last	Ranged Twice Melee Last
Ranged Twice>Melee First	Ranged Twice Melee First
Melee First<Ranged Last	Ranged Last
Melee First<Melee Last	Melee Last
Melee First<Ranged Twice	Ranged Twice Melee First
Ranged Last>Melee Last	Ranged Last
Ranged Last>Ranged Twice	Ranged Last Twice

Every possible round

This allows us to determine what the 0-to-6 rounds of combat will involve for any given match-off. All that remains is to create **a table of 4900 x 6 = 29,400 combat rounds**.

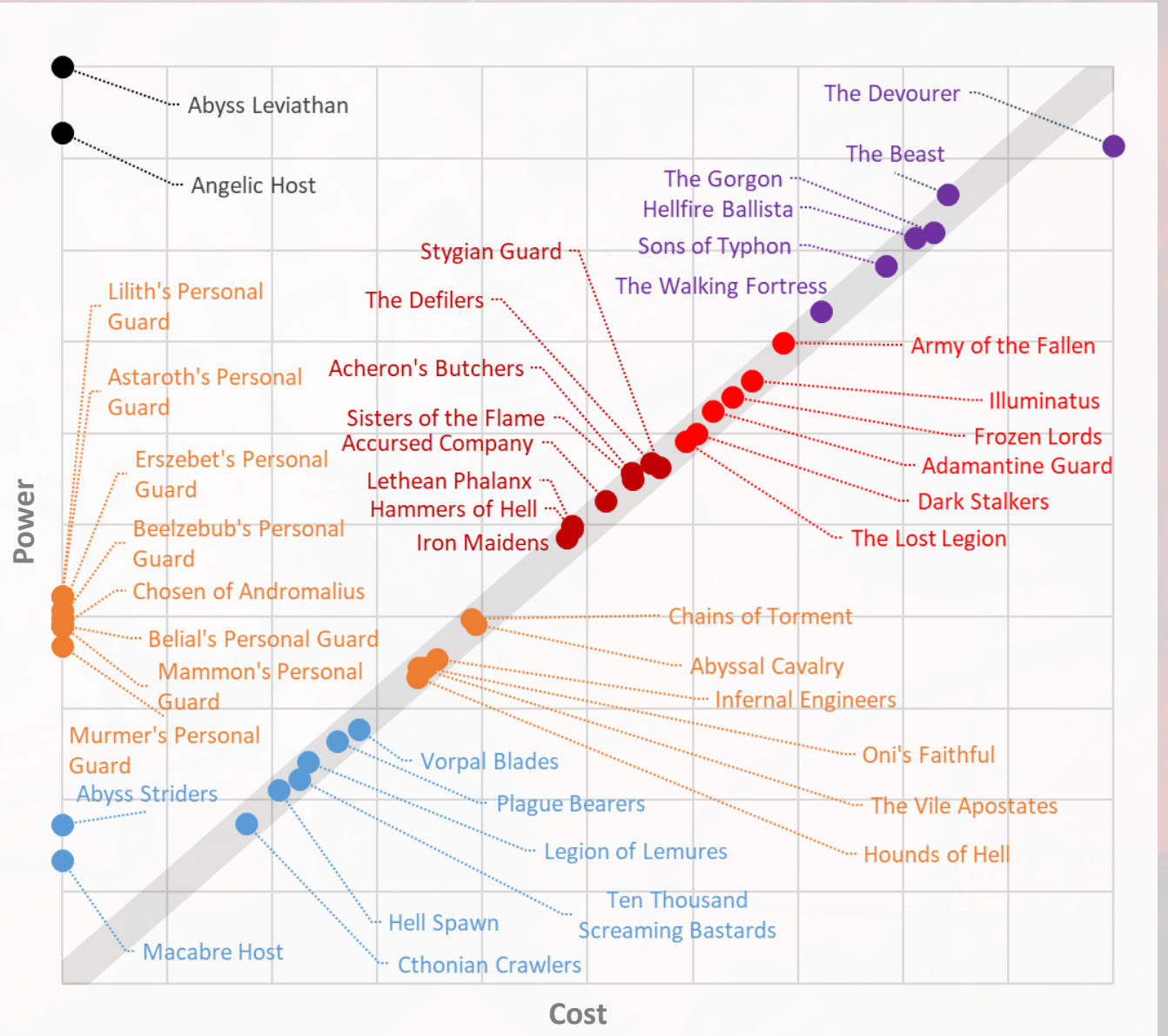
I15 =XLOOKUP([@[Attacking contender index]],Contenders[Index], Contenders[Hitpoints]) - SUMIFS([Attacker damage taken], [Fight index], [@[Fight index]], [Round], "<"&[@Round])

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Index	Fight index	Round	GUID	Attribute	Attacking contender index	Defending contender index	Is fight possible?	Attacker hitpoints at round sta	Defender hitpoints at round sta	Attacker strength	Defender strength	Attacker resistance	Defender resistance	Attacker damage taken
11	10	2	4	Fight_1V2_Round_4	None	1	2	TRUE	6	2	0	0	0	0	0
12	11	2	5	Fight_1V2_Round_5	None	1	2	TRUE	6	2	0	0	0	0	0
13	12	2	6	Fight_1V2_Round_6	None	1	2	TRUE	6	2	0	0	0	0	0
14	13	3	1	Fight_1V3_Round_1	Ranged	1	3	TRUE	6	4	2	3	2	3	1
15	14	3	2	Fight_1V3_Round_2	Melee	1	3	TRUE	5	4	4	3	4	3	0
16	15	3	3	Fight_1V3_Round_3	Infernal	1	3	TRUE	5	0	0	4	0	4	0
17	16	3	4	Fight_1V3_Round_4	None	1	3	TRUE	5	0	0	0	0	0	0
18	17	3	5	Fight_1V3_Round_5	None	1	3	TRUE	5	0	0	0	0	0	0
19	18	3	6	Fight_1V3_Round_6	None	1	3	TRUE	5	0	0	0	0	0	0
20	19	4	1	Fight_1V4_Round_1	Ranged	1	4	TRUE	6	7	2	0	2	0	0
21	20	4	2	Fight_1V4_Round_2	Melee	1	4	TRUE	6	5	4	5	4	5	1
22	21	4	3	Fight_1V4_Round_3	Infernal	1	4	TRUE	5	5	0	0	0	0	0
23	22	4	4	Fight_1V4_Round_4	None	1	4	TRUE	5	5	0	0	0	0	0

Here we **XLOOKUP the result of the previous round** in order to determine the starting state for the next round (and whether it will even play out).

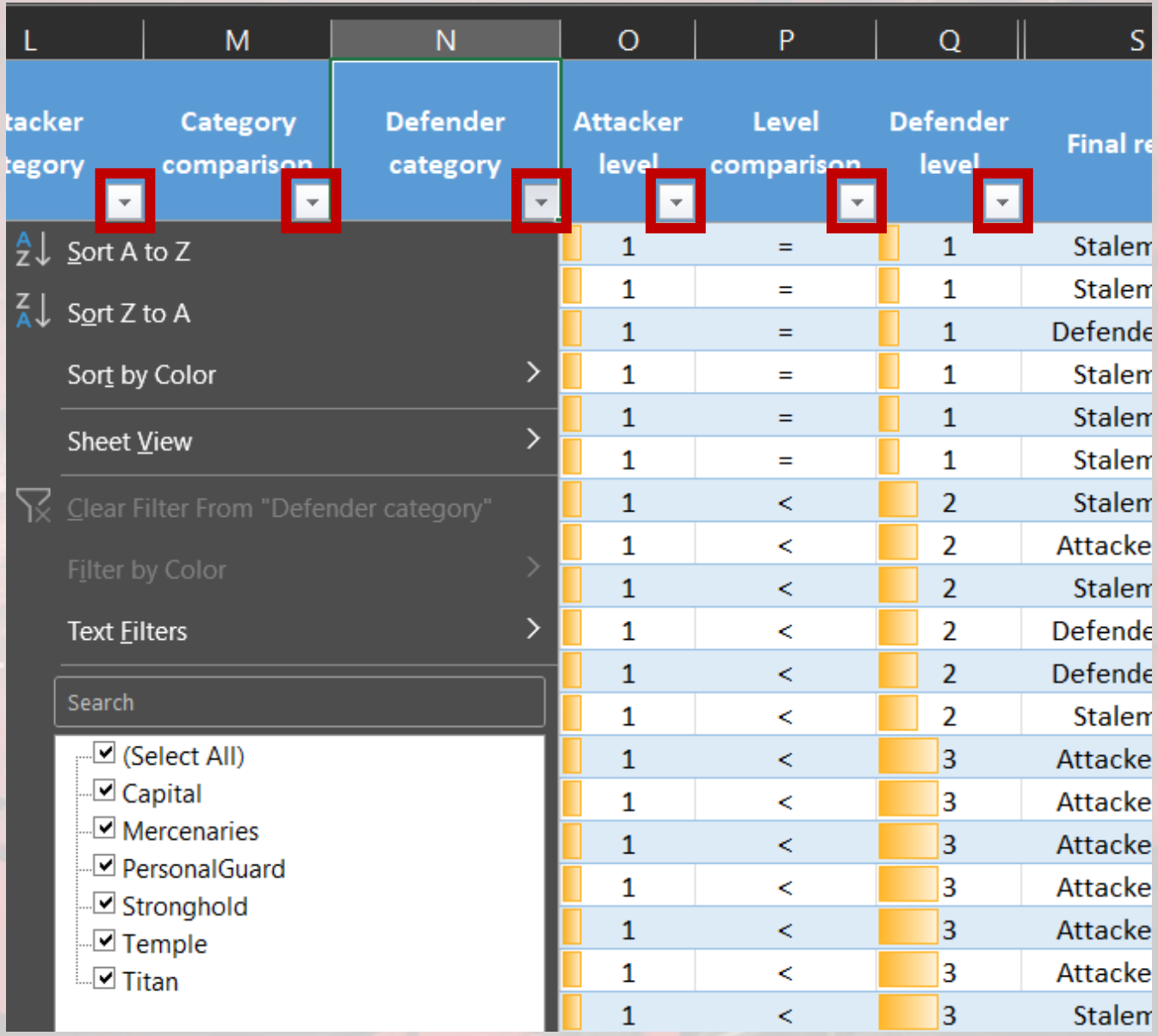
A reliable power metric at last

- The result: **how much damage each legion would deal, and take**, if it fought ever other one in turn.
- The strength metric that we used for **our power curve is based on the results** of this battle royale →
- We **didn't need to add power values by hand** for abilities that override sequences or provide simple combat bonuses: we just simulated them.
- *Excel* can run all of this in **under 5 seconds**. Quicker than opening Unity.



Filtering and sorting data

- Most importantly, we can **filter the columns to understand exactly *why*** a given legion is doing so well, or so poorly →
- Some examples would be examining the rounds of specific battles that…
  - … end in a stalemate…
  - … involve starting legions…
  - … last for at least 5 rounds…
  - … all of the above at once… etc.



Content tuning auto-tests

Last but not least, a high level dashboard checks whether **we're breaking any of the rules we've set ourselves**. E.g. no *Personal Guards should defeat a Place of Power unassisted.*

K274

=IF([@[Number of fights]]=0, 0, COUNTIFS(Fights[Defending contender index], [@[Defending contender index]], Fights[Is fight possible?], TRUE, Fights[Attacker category], [@[Attacker category]], Fights[Final result], "DefenderDied")/[@[Number of fights]])

	A	B	C	D	E	G	H	I	J	K	L	M
1	Index	Defending contender index	Defendering contender category	Defender contender name	Attacker category	Any fights to analyse?	Defender kills attacker percentage	Both kill eachother percentage	Stalemate percentage	Attacker kills defender percentage	Min defender percent hitpoints left	Max defender percent hitpoints left
274	273	46	Temple	*The City of Dis*	PersonalGuard	TRUE	100%	0%	0%	0%	100%	100%
280	279	47	Temple	*The Palace of Gluttony*	PersonalGuard	TRUE	100%	0%	0%	0%	62%	100%
286	285	48	Temple	*The Mines of Gehenna*	PersonalGuard	TRUE	100%	0%	0%	0%	58%	100%
292	291	49	Temple	*The Altar of Abominatic*	PersonalGuard	TRUE	50%	0%	50%	0%	50%	100%
298	297	50	Temple	*The Mouth of Abbadon*	PersonalGuard	TRUE	75%	0%	25%	0%	42%	100%
304	303	51	Temple	*The Garden of Infernal*	PersonalGuard	TRUE	38%	0%	63%	0%	45%	100%
310	309	52	Temple	*The Temple of Lust*	PersonalGuard	TRUE	88%	0%	13%	0%	56%	100%
316	315	53	Temple	*The Wood of the Suicide*	PersonalGuard	TRUE	25%	0%	75%	0%	46%	77%
322	321	54	Temple	*The Face in the Sand*	PersonalGuard	TRUE	38%	0%	63%	0%	9%	73%
328	327	55	Temple	*The Vats of Torment*	PersonalGuard	TRUE	13%	0%	88%	0%	50%	100%
334	333	56	Temple	*The Theater of Sloth*	PersonalGuard	TRUE	0%	0%	100%	0%	31%	85%
340	339	57	Temple	*The Unholy Fountain*	PersonalGuard	TRUE	0%	0%	100%	0%	11%	78%
346	345	58	Temple	*The Tower of Pride*	PersonalGuard	TRUE	0%	0%	100%	0%	8%	62%
352	351	59	Temple	*The Great Wheel of Pai*	PersonalGuard	TRUE	0%	0%	100%	0%	18%	73%
358	357	60	Temple	*The Tree of Woe*	PersonalGuard	TRUE	0%	0%	100%	0%	30%	60%
412	411	69	Temple	*Citadel of Wrath*	PersonalGuard	TRUE	88%	0%	13%	0%	100%	100%
418	417	70	Temple	*Black Altar of Despair*	PersonalGuard	TRUE	0%	0%	100%	0%	33%	83%

Epilogue

How this can empower you,
if you don’t stoop to idolatry

It worked… at least for us

- **Good balance tends to be invisible**, and *Solium* has now gone through a number closed beta tests now with **few, if any, complaints** about the legion balancing.

- I've no doubt our **tool-assisted tuning** is to thank for this: *"Fight Club"* allowed us to **catch silly tuning "bugs" early**, and thus to avoid wasting our play-testers' time catching them.

- Using spreadsheets meant **seamless integration with our *"source of truth"*** and **good visualisation tools** – it also avoided our tying up a programmer.

But beware!

- **Breaking a pattern you've established** in the player's mind can cause *delight, comedy*, *intrigue, …* all the good stuff!
- But while need to *have* a pattern to break one, **following it too slavishly** will result in a really boring game!
- Moreover, working on models **can be a distraction** from working on the game itself: they are **a *means*, not an *end***!
- Rational tuning is ***just* a *"lens"***, as *Schell* would put it: one more tool to consider, but ***not* a dogma** to follow blindly!

Thank you for your time!

Questions?

mastodon.social/@wilbefast

linkedin.com/in/williamdyce