

Passive Everything

An extension of passive modifications.

A property from similar card games that would be interesting to also have in Bloodless is the fact that cards have the ability to override rules of the game. Because Bloodless is defined as an algorithm, however, this is difficult to implement. While injecting behavior into parts of the game's processes is not necessarily difficult (the events system is just one example of such an injection system), we need ways to inject behavior into every structure the game defines as well.

The game is already considered to do this through the majorly undocumented mechanism of Passive Modifications. It is said that, when a number on a card is looked at, it is then put through these modifications. The actual value utilized is returned at the end of this pipeline. This naive description comes with certain problems (like would be infinite loops), but those will be addressed in this proposal.

Under this proposal, every part of the game would be capable of having its own layer of modifications before being truly observed. As an example, to obtain a space's adjacencies, you check if there are no modifications altering the base adjacencies.

Looping

The main problem with passive modifications is infinite loops. Consider the following two effects:

1. Creatures with 1 health gain +1 power.
2. Creatures with 1 power gain +1 health.

Let's consider a 1/0/0 creature.

- We want to know the creature's power.
- During checking the power, we find a modification that cares about the health, so we check that first.
- During checking the health, we find an modification that cares about power, so we check that first.
- During checking the power, we find a modification that cares about the health, so we check that first.
- This is an ability we've seen before, causing an infinite loop.

Our proposed solution is to make it so that recursive "look at" operations do not happen. Instead, we start with a base, and apply all the effects in order each time we look at a number. The process ends up looking this way:

Let's say that the abilities here are ordered such that lower numbers have higher precedence.

- We start with the base stats.
- The first event adds 1 power to it. It is now a 1/0/1
- The second event adds 1 health to it. It is now a 2/0/1