The Comprehensive Rules of Bloodless

Draft: 4.0.0

The Bloodless Team

May 20th, 2025

Semantic Versioning

This document is versioned using the following pattern:

```
<major version>.<minor version>.<patch>
```

Changes in the patch version will consist only of minor additions or changes to Bloodless, and aren't expected to have great ramifications in the majority of games. The order of rules is guaranteed not to change between patch versions.

Changes in the minor patch version will consist of additions that are expected to have ramifications in the game. The order of rules is guaranteed to not change between minor patch versions.

Changes in the major patch version are guaranteed to change the order of rules, and may change the meanings. Changes in a major version may be labeled "style patches," meaning that the content of the rules has not been changed, but their order has. It is *not* guaranteed that their meaning will not change, but this will be attempted.

The Rules

Definitions

This chapter defines all types of values that are used to process a game of Bloodless. The following chapter defines the processes used by the game.

1. Numbers
   - 1.1. Natural numbers are defined as the set of positive integers with zero appended.
   - 1.2. In Bloodless, a number is a natural number.
   - 1.3. Whenever an operation between numbers results in a number outside the natural integers, the result is zero.
   - 1.4. "Extended numbers" are natural numbers with positive infinity (∞) appended. Positive infinity is defined by the following two properties:
     - 1.4.1. It is greater than every natural number.
     - 1.4.2. It is equal to itself.
2. Names
   - 2.1. In Bloodless, Names are unique objects, which have representations in each language and medium. For example, "Ant Queen" is an English representation of a Name in Bloodless in a written medium.
   - 2.2. Every Name has an extended number associated with it, called its Limit. The default Limit for a number is 5.
3. Card Types
   - 3.1. There exist two basic types: Command and Creature.
   - 3.2. Blood Flask is a subtype of Creature.
   - 3.3. Extended Command is a subtype of Command.
   - 3.4. All Creature and Command types have Saga variants.
   - 3.5. All non-Vestige types have Vestige variants.
4. Events
   - 4.1. An event represents a change to the state of the game.
   - 4.2. When an event is applied, it creates an Outcome. The Outcome may be a Failure or a Success, containing information about all cards and players involved in the event.
5. Triggered Abilities (TAs)
   - 5.1. Event-Modifying Triggered Abilities (EMTAs)
     - 5.1.1. EMTAs have a trigger, which is an Event pattern.
     - 5.1.2. When applied to an event that triggers it, an EMTA will result in a list of new events, defined by the specific EMTA.
     - 5.1.3. When applied to an event that does not trigger it, an EMTA will just result in a list of only that same event.
   - 5.2. Event-Creating Triggered Abilities (ECTAs)
     - 5.2.1. ECTAs have a trigger, which is an Outcome pattern. They also have a Consequence, with is an ordered list of events.
     - 5.2.2. When triggered, they process their Consequence.
   - 5.3. For an ability to be triggered, its trigger has to match the event it is being applied to.
   - 5.4. Triggers:
     - 5.4.1. TODO! write a list of all triggers

6. Rules Text
    6.1. Rules Text is a list of Abilities or a list of Events.
7. Card
    7.1. A card has: Name, Type, and Cost (a number which may be a variable).
    7.2. A card may have a Flip Cost (a number which may be a variable).
    7.3. There exist certain rules that apply only to cards of a certain Type.
    7.4. While a card is in an owned zone, the owner of that zone is the card's "controller".
    7.5. A card's owner is the player who controls it when the game begins, or when it is created.
    7.6. Type Properties
        7.6.1. Creatures have Health, Defense and Power. They also have an Attack, which consists of a pattern of relative directions to spaces in the Board.
    7.7. Variables
        7.7.1. A number in a card may be variable. A variable is represented with a letter from the Latin or Greek alphabet. Variable numbers have two variants: Undefined and Defined. Their default state is Undefined.
        7.7.2. All numbers in a card represented with the same letter are the same number.
        7.7.3. The Defined variant of a variable number contains a natural integer. The Undefined variant contains no extra structure.
        7.7.4. When the value of a number that is variable is needed and that number is in a Defined variant, the number it is Defined with is used.
        7.7.5. When the value of a number that is variable is needed and that number is in its Undefined variant, one of the following can happen:
            7.7.5.1. If the same variable appears in the card's Cost, Health, Defense or Power, it is zero.
            7.7.5.2. Otherwise, the player chooses a value for it, following the restrictions specified on the card.
8. Places
    8.1. A Places is a place where a card may be. A place may have at most one card.
    8.2. All Places have an owner.
    8.3. Places are also linked by Zones. All spaces in a Zone have defined collective behavior with one another.
        8.3.1. Discard Pile: An unordered list of cards.
        8.3.2. Board
            8.3.2.1. The Board consists of 8 places, called spaces, in a 4x2 grid. Each player owns one row.
            8.3.2.2. Only cards of Creature type may be on the spaces of the Board.
        8.3.3. Timeline: An ordered list of cards.
            8.3.3.1. Only cards of Command type can be on the Timeline.
        8.3.4. Hand: An unordered list of cards.
        8.3.5. Deck: An ordered list of cards.
            8.3.5.1. The first element of the Deck is also known as "the top" of the Deck.
            8.3.5.2. The last element of the Deck is also known as "the bottom" of the Deck.
        8.3.6. Aside: An unordered list of cards.

8.3.6.1. Cards in the Aside cannot be directly referenced by other cards, unless they had referenced them before they entered this zone.

8.4. In print, cards in the Board are called “creatures” and cards in the Timeline are called “commands”, whereas cards in other zones are called “cards.” Their type may be specified such as “creature card” or “command card.” The only exceptions to these rule are:

8.4.1. Spawning, summoning, moving to board, where “creature” includes (but is not exclusive to) “creature cards”.

8.4.2. Executing, moving to timeline, where “command” includes (but is not exclusive to) “command cards”.

8.4.3. Playing, where “command” and “creature” include (but are not exclusive to) “command card” and “creature card” respectively.

9. Passive Abilities

9.1. A Passive Ability modifies the state of a card and, importantly, undoes the modification when it’s finished.

9.2. In order to achieve this, a Passive Ability remembers the specific state change for everything that it modifies, if it has.

9.3. Passive Abilities have an opposite operation.

9.4. Passive Abilities may have a condition under which they apply.

Processes

This section defines all processes in Bloodless. When playing a game of Bloodless, the Bloodless process is initiated.

Every process has its own “current step”, separate from the process that initiated it. The process that initiated it can only move to the next step once all the processes it’s started have finished. However, child processes can change the current step of processes that are unique, such as the Bloodless process.

When a process moves to a step, that step is immediately executed, even if a child process has not finished.

1. Applying Events

1.1. The outcome of an event contains the same information as the event itself. However, if the event contains patterns, the outcome will contain definite information. (e.g. a specific card instead of a card pattern).

1.2. Move

1.2.1. Contains a card pattern and a zone pattern.

1.2.2. Removes a matching card from whatever zone it’s in and moves a matching zone.

1.2.3. Outcome additionally contains the zone the card moved from.

1.3. Play

1.3.1. Contains a card pattern to a command, a player pattern, an optional blood cost, an optional space on the board.

1.3.2. Player defines all the variable numbers present in the card’s Cost.

1.3.3. If there is no cost in the event, the cost in the event is the same as the card’s.

1.3.4. If the player has less blood than the paid blood cost, process ends here. Outcome is a failure.

        1.3.5. If card is a command and there is a space, process ends here. Outcome is a failure.
        1.3.6. If card is a creature and there is no space, space is chosen by player.
        1.3.7. Player loses that amount of blood (process event)
        1.3.8. Queue a move to the corresponding zone or space.
    1.4. Draw
        1.4.1. Contains a player pattern and a deck pattern.
        1.4.2. Processes moving a card from the top of a matching deck to a matching player's hand.
        1.4.3. Outcome additionally contains the car that was drawn.
    1.5. End of Turn
        1.5.1. Does not contain anything.
        1.5.2. Changes active player so that it is now their opponent.
        1.5.3. Clears event queue.
        1.5.4. Outcome contains the previous player.
    1.6. Search
        1.6.1. Contains the player who will do the search, the deck that will be searched, and the pattern to be matched.
        1.6.2. All cards in the deck become known to the player.
        1.6.3. The outcome contains the cards found to match. If the pattern cannot be matched, the player Fails To Find. This is *not* a failure outcome for this event (e.g. the next event in a "then" chain will work fine).
    1.7. Shuffle
        1.7.1. Contains which deck is shuffled.
        1.7.2. Randomizes order of cards in a deck.
2. Creature Death Checks
    2.1. The active player's spaces checked from left to right from their perspective. Then, the inactive player's, from left to right also from their perspective.
    2.2. If one of the spaces has a creature with 0 health, process a "death" event for that creature.
3. Game End Condition Checks
    3.1. If the health pool is zero, process a "finish game" event of with the player who last received player damage losing and their opponent.
4. Processing A Player's Turn Skips
    4.1. If the player has more than zero turn skips, do game end condition checks, Bloodless moves to step 8.4.
5. Applying Passive Abilities
    5.1. Every Passive Ability iterates through its targets.
    5.2. If the target passes the condition:
        5.2.1. If this Passive Ability is already modifying the target:
            5.2.1.1. If the modification to be done has changed, change the state accordingly.
        5.2.2. If this Passive Ability has not modified the target:
            5.2.2.1. Change the state accordingly
    5.3. If the target doesn't pass the condition:
        5.3.1. If this Passive Ability has not modified the target:
            5.3.1.1. Change the state accordingly

5.3.2. If this Passive Ability is modifying the target:
5.3.2.1. Modify the state oppositely

6. Resolving an Event
6.1. An ordered list of only the processed event is made. This is the current events list.
6.2. A new, empty ordered list of events is made. This is the future events list.
6.3. EMTAs are sorted by precedence.
6.4. Each EMTA is processed in the following way:
6.4.1. For each Event in the current events list:
6.4.1.1. If the Event does not match the EMTA's trigger, add it, unmodified, to the future events list.
6.4.1.2. Otherwise, apply the modification to it, and append the result to the future events list.
6.4.2. Then, make the current events list equal to the future events list.
6.4.3. Empty the future events list.
6.5. ECTAs are sorted by precedence.
6.6. Then, for each event in the current events list,
6.6.1. Apply the event to the current board state.
6.6.2. Apply each ECTA to the Outcome.
6.6.3. If the Outcome is of a Play-related event, process its next step.
6.7. Apply passive abilities
6.8. Process game end condition checks. (see 3)
6.9. Process creature death checks. (see 2)
6.10. Process turn skips. (see 4)
6.11. Then, players can make any instant speed decisions they are allowed to, if they have any.

7. Main Phase choices
7.1. The Active Player can choose to do one of the following things:
7.1.1. Playing a card from their hand.
7.1.2. If they have additional card draws, they can draw a card, then reduce the additional card draws by one.
7.1.3. Bloodless goes to step 8.3.5.

8. Bloodless
8.1. Players draw 5 cards from their main deck, then one from their blood deck.
8.2. If a player chooses to do so, they may Mulligan. They may only do this once. Continue with the next step once both players have decided to or not to mulligan.
8.3. Turn loop:
8.3.1. Process "start of turn" event for the active player.
8.3.2. If it is not the Active Player's first turn, their additional card draws are set to 1.
8.3.3. Process "start of main phase" event for the active player.
8.3.4. Main Phase loop:
8.3.4.1. The Active Player makes their Main Phase choices
8.3.5. Process "end of main phase" event for the active player.
8.3.6. Process "start of attack phase" event for the active player.
8.3.7. Attack Phase loop:
8.3.7.1. Currently attacking space is leftmost space

8.3.7.2. If currently attacking space has a creature, it attacks.
8.3.7.3. If there is a space right of the currently attacking space, the currently attacking space is now that space, and return to step 8.3.7.
8.3.7.4. Otherwise, continue to step 8.3.8.
8.3.8. Process “end of attack phase” event for active player.
8.4. Process “end of turn” event.
8.5. Continue with step 8.3.

Notes For This Draft

This draft is, like, violently incomplete. You wouldn't believe it.

It's missing:

- A list of triggers
- A list of events
- What, like, any pattern even is
- How ECTAs *actually* create their consequence event using event outcomes
- Literally anything about flip costs??? oh my god
- When you play a card, you have to set all its undefined variables that are part of one of its stats
- Play card events self trigger
- Also technically events trigger other events trigger other events. Like when you play a command it goes play -> execute -> move to timeline
- Maaaaaaaaybe card parsing? That could probably be part of a separate chapter specific to English rules, since every language will have its own parsing scheme

Despite all of this... I'm actually quite happy with this structure? It feels like the first version of Bloodless' rules that has a chance to be complete. I think it's up to Yellow now, whenever he has the time and energy for this stuff. He's good at breaking everything I make (/pos <3)