

Jamie Kasulis and Teiheim Edwards

Repository: williams-team-omondiericqianwenzheng

29 November 2019

## Code Quality Report

Over all, the code base looks great. We find the code to be highly readable, well-formatted in most places, and the logic of the functions to be no more complicated than necessary. It was very painless to figure out how things were working currently in the code base. Making modifications ended up being a modest task rather than a gigantic one, which indicates the extensibility of the system.

One area where the system was not so readily extendable was with the game event functions. These encompass the functionality for `:on-minion-damage`, `:on-end-of-turn`, `:on-divine-shield-removal`, etc. In the old system, arguments for these were passed as a series of additional function arguments, as shown below:

```
(do-game-event-functions state :on-end-of-turn :player-id "p1")
```

The original implementation of this function accepted additional arguments `:player-id` and `:target-id`.

While this worked very well initially, implementing the “Acolyte of Pain” card shows some ways in which this was inflexible. For instance, Acolyte of Pain not only has to check for the id of the minion with the `:on-end-of-turn` function, but the minion whose being damaged triggered Acolyte’s `:on-minion-damage`. Moving forward, there could theoretically be any number of additional arguments needed. This means that you would need to keep adding these to the function argument, which would add a lot of maintenance overhead over

time. Additionally, there was no mechanism in place to throw an error if an invalid argument (i.e. `:silly-id`) was passed, making the enumeration of explicit arguments seem to be without much benefit.

The approach we took to modify this was to edit the card definitions and the function `do-game-event-functions` so that the arguments taken would be `[state this-minion-id & [kvs]]`. This allows us to throw all of our additional arguments into `kvs` as-needed. Now, each new card will only have to pull the key-value pairs it needs from `kvs`. You will never have to think about what the particular key-value function arguments are again. No more changes to `do-game-event-functions` arguments should be needed moving forward.

Making this new change was relatively painless and quick. This reflects the good organization and readability of the original code.

Some additional constructive remarks:

- `create-secret` is an unused function which seems like it can be deleted
- There are two definitions for `create-secret` in `construct.cljc` which seems like it could easily lead to errors if you call the wrong one
- Some formatting inconsistencies, as well as closing parentheses and brackets alone on some lines.
- Some one-line documentation can help with functions whose names don't clearly indicate their effects, such as `get-random-minions-distinct` or `valid-minion-effect-target`

Some additional positive remarks:

- The use of `cond` instead of multiple nested `ifs` greatly helps readability!
- Some tests try calling functions consecutively, which is a great added layer of durability