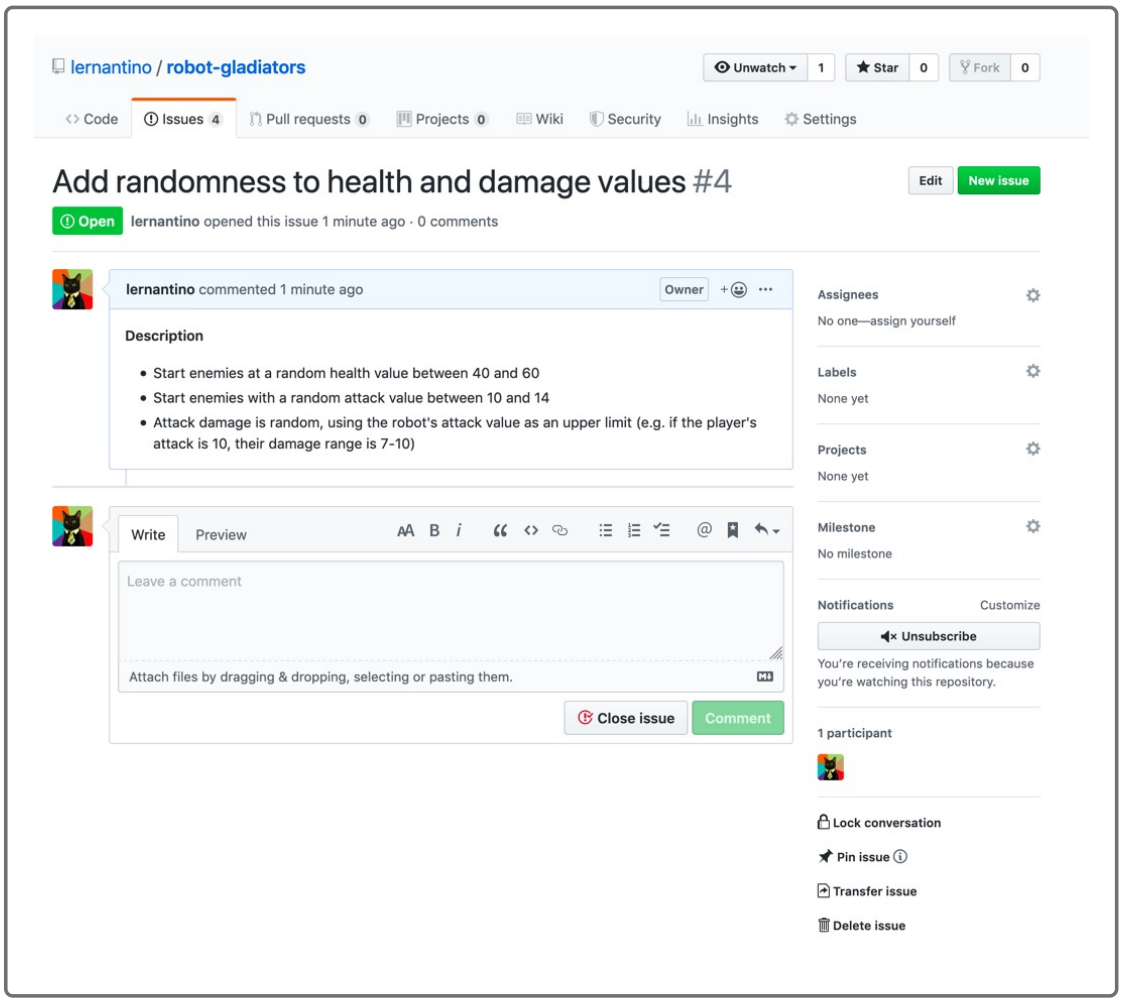
3.4.1 Introduction

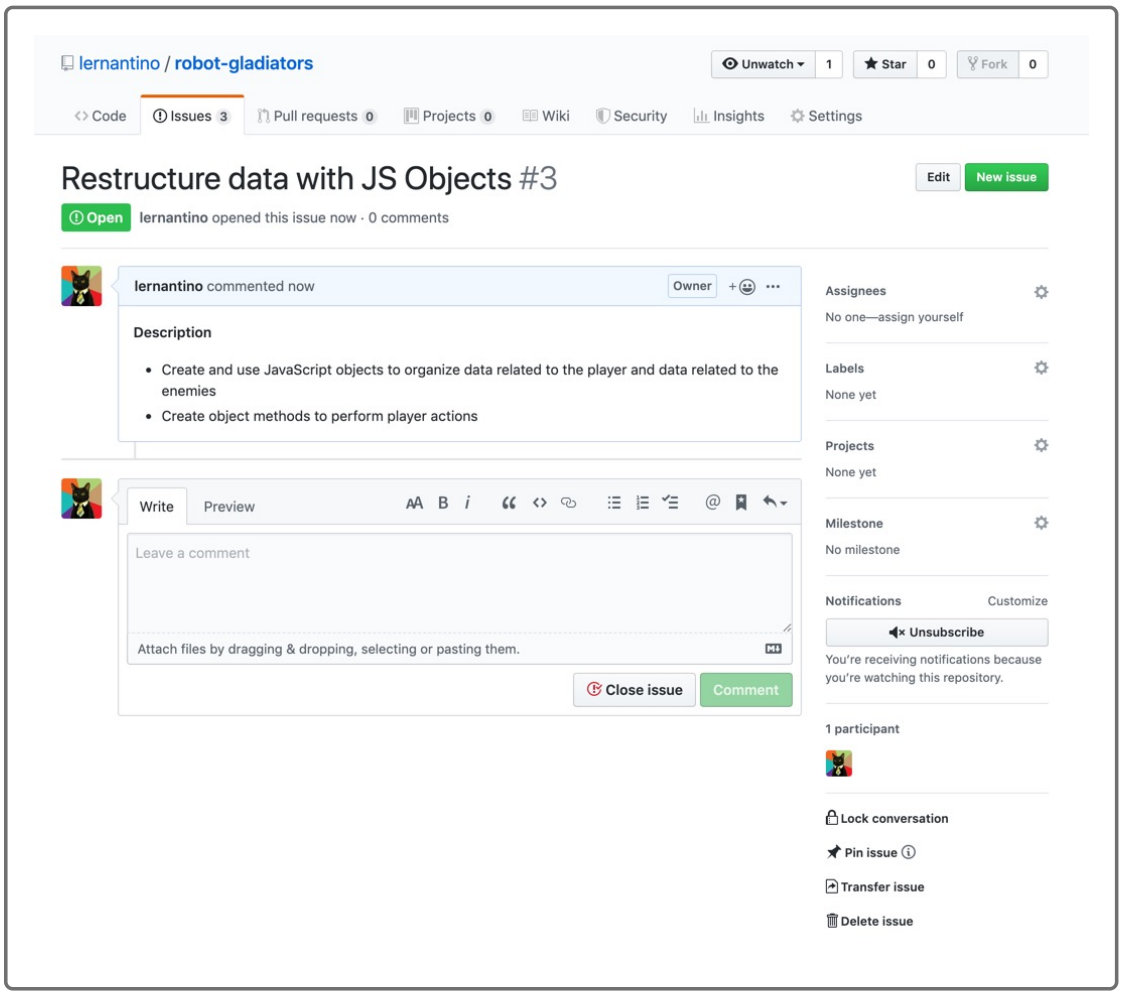
Now that our MVP goal has been met, we could technically submit the game as is. However, it's still not that interesting to play multiple times, because the health and damage values are always the same. We need to update our code to generate these values randomly.

Our next GitHub issue addresses this:



We're still making improvements to the game, but we should also take time to clean up the codebase. This will give us a better shot at that "code quality" prize in the game jam. More importantly, it will leave our project in a state where we can easily expand on it afterwards.

The other GitHub issue we'll tackle in this lesson addresses this type of code optimization:



JavaScript objects will help us accomplish both tasks: randomness and restructuring. We'll use built-in objects that, like the `window` object, come with useful methods for generating random numbers. We can even make our own objects to tie data together, which is perfect for consolidating player and enemy stats.

Having a more **object-oriented** structure would definitely appeal to the game jam judges. In larger games and apps, building around objects can help immensely in keeping track of what data you have and how it can be accessed. Similar to our practice with functions, becoming familiar with objects now will pay huge dividends later!

In this lesson, the learning goals will be:

- Create objects and their associated methods and properties to organize the data in the application.

- Understand the utility of the built-in objects like the `Math` object, how to use it, when it can be used, and what methods and properties are available.
- Return values back to the function call by using the keyword `return`.