

A Summary of Our Project: GenCardGame

Our project is an API for making concurrent card games. We wrote code that starts a server and connects clients, then runs a series of concurrent and non-concurrent events that make up a game. In order to make our API applicable to most games, we allow the user to define all of the actual game logic. This means that the user writes functions for what to do on a turn, how to check win condition, how to determine the next player, etc.

Because we felt like this could be a burden for the user, we wrote a partial library of classes to handle common game functionality. This library includes a Player class, a Hand class, a Deck class, etc. It was our goal that in using these classes, it would be easy for a user to write any relatively simple game logic. Using our API we created the game Bullshit as a test case. Bullshit has both non-concurrent (playing a card) and concurrent (calling “Bullshit”) actions, making it a good test case.

Below, on the left is a diagram representing how the internal state might look while a non-concurrent event is happening, and on the right is a diagram representing how the internal state might look while a concurrent event is happening.

