The Task

Describe, in a document, your design strategy for the game. What are your Actors going to be? How are you going to resolve the race aspects of the game, such as whose hand slaps the deck first?

In the early stages of planning, a few configurations for actors were considered and tested. The first iteration involved delegating each pile as well as the discard pile their own actor where the idea was that they could run independently, and check for valid discards concurrently with no chance for race conditions. The isolation would also ensure that states would not be tampered with by any external control. Practically, this became difficult to combine and coordinate card operations across each set of 5 piles to progress the rules of the game.

This lead to a simpler, three actor system, in closer simulation to the real world. One actor would be designated as players while the other would simulate a dealer. Due to inexperience in the Akka model, there was some time taken to get to grips with the certain limitations and rules in regards to managing mutable states, the affect of sharing states -- directly or indirectly -- and in some instances, the illusion of synchronisation.

Akka's ask pattern was employed at first in handling the validation of discards with each move. The dealer would initiate a round by sending tell requests to players who would after calculation of valid moves, send an ask message to the dealer waiting for a response or confirmation of whether it was valid or not. This resulted in frequent race conditions as it was found that threads would calculate changes in mutable state asynchronously if enclosing these calculations within futures, rather than blocking the actor as previously thought.

To reduce this, all mutable states were freezed to immutable values as soon as they are received and before they are used for calculation. In most cases, this required moving the majority of these tasks off the actor's receive method as well as replacing almost all ask patterns into simple tell messages.

Although this greatly improved the system, some problems still are present. The biggest unsolved challenge was to determine how to handle stuck cases. If a player had no valid choice for a card, they would report their "stuck" state to the dealer, who would store it to then initiate a next round if the other player reported the stuck state. However, it was not yet found how to have an actor restart themselves when new discard choices arose.

Currently, messages are configured to be sent to the other player when discards are approved. This aided an actor to remain active although again presented race conditions when these messages are sent while an actor is presenting a card to discard. Although much of the rest of the functionality of the program is relatively complete, it is evident that more experience and understanding is required to solve this concurrency issue.