Vision

Our vision for Rekop poker is to have a full-fledged multi-player experience where users can modify their match settings to their liking, be free of in-app products / upgrades and enjoy a casual, sociable poker experience. For example, customization options could include reduced time to make their decision, resulting in haste gameplay, custom themes for the table such as different backgrounds, card faces, etc or simply changing the minimum bet on each round.
We also wish to include a server program so that players are able to host their own servers to perform the aforementioned customization option. This server should be able to run on multiple platforms, including Windows, Android and Linux.

Scope

Major features

We need to be able to produce the following pieces and aspects of software for the game to become functional:

- An Android app that:
  - Has the ability to evaluate hand power in Texas Hold'em
  - Has the ability to create an account or play as a guest
  - Has the ability to run on devices using Android version KitKat or above.
    - This may be changed during development, as some essential functions may not be available
- A server application that:
  - Has the ability to run on multiple platforms, most critical are:
    - Windows
    - Android
    - Linux
    - Since Android uses the Linux kernel, it is imagined that creating a server should not be difficult. However, we may need root access on the device and thus may have to resort to a UPnP system; this is more error prone however it is an option.
- Testing documents / utilities, including:
  - Test suites that are able to evaluate:
    - Hand power in Texas Hold'em
    - Fuzz-testing on all user input
      - This can be in the form of using non-English letters in fields, extreme random numbers where applicable, etc.
    - Server connection / disconnection issues
    - Server input data
  - Black-box testing
    -

Risks

One of our main risks is that evaluating hand power in poker involves a lot of hands and this can make testing and development very time consuming. Though they can be split into four attributes - flush, straight, pairs and high cards (in the absence of the past 3) - many of these hands contain one or more attributes and more than one of them can be found in a single hand.
For example, the following is [SUIT, VALUE] where:
D = Diamonds, C = Clubs, H = Hearts and S = Spades

Say we have the end hand of:
C9, C8, D7, D6, D5, D4, D3

It becomes very difficult to evaluate this efficiently, i.e. evaluate that it has a straight and a flush in the same area. Because naturally, the straight function should return the highest power straight - which would be C9 - D4 as it does not care about the suits. This means that we could have a harder time testing if we attempted to make the algorithm very modular, as some hands may slip through the cracks.
Thus, it is most likely we will suffer degraded performance and code complexity by evaluating each and every type of hand - straight, straight flush for example. This will however save us time, allowing more areas of the project to be developed.