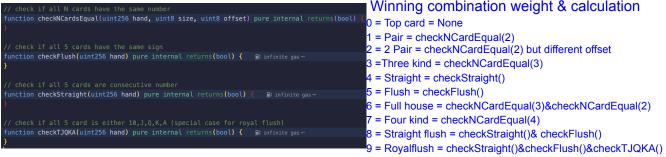# PokerChain Project Updates

## Game logic component:

We have started building a smart contract of core components of poker, defining how the poker hand will be represented, how to calculate winning hands and win checking logic.
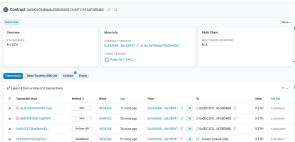
There are a total of 52 cards per deck. For the poker hand combination (consisting of 5 cards), each card will be assigned to number 0-51 (starting with id 0 for the 2 of clubs, id 1 for the 2 of diamonds, id 2 for the 2 of hearts, id 3 for the 2 of spades, and so on) which will take up 6 bits. So the total will be 30 bits for a hand of 5 cards, hence, we will use uint256 as a hand representation. With this format, the card number can be determined by (hand / i) / 4 and the card sign can be determined by (hand / i) % 4, where i is the card position (1-5).

For the winning combination, we have assigned the weight of the combination and also created a logic to check the winning hands according to the normal Poker rules. We have made a general card checking function as following:

```
// check if all N cards have the same number
function checkNCardsEqual(uint256 hand, uint8 size, uint8 offset) pure internal returns(bool) {
}

// check if all 5 cards have the same sign
function checkFlush(uint256 hand) pure internal returns(bool) {    infinite gas...
}

// check if all 5 cards are consecutive number
function checkStraight(uint256 hand) pure internal returns(bool) {    infinite gas...
}

// check if all 5 card is either 10,J,Q,K,A (special case for royal flush)
function checkTJQKA(uint256 hand) pure internal returns(bool) {    infinite gas...
}
```

Winning combination weight & calculation
0 = Top card = None
1 = Pair = checkNCardEqual(2)
2 = 2 Pair = checkNCardEqual(2) but different offset
3 =Three kind = checkNCardEqual(3)
4 = Straight = checkStraight()
5 = Flush = checkFlush()
6 = Full house = checkNCardEqual(3)&checkNCardEqual(2)
7 = Four kind = checkNCardEqual(4)
8 = Straight flush = checkStraight()& checkFlush()
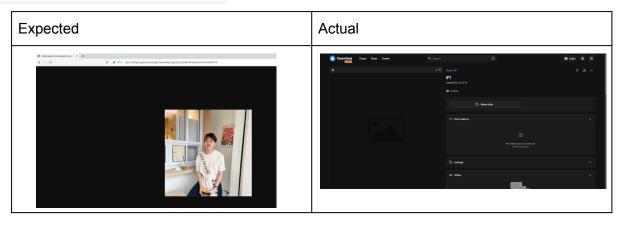9 = Royalflush = checkStraight()&checkFlush()&checkTJQKA()

Currently, we are in process of creating a game state component that will keep track of the tables (each player's hands, position such as small/big blind), assigning how the card will be dealt and the pot/prize on the poker table.

## Transaction component: NFT



For the transaction part, we will focus on NFT since it is one of our selling points. Currently, we've deployed the NFT contract for our project on the Sepolia Testnet. The NFT collection now shows up on Opensea, but we can't get it to show the content of the NFT deployed NFTStorage via ipfs at the moment.

| Expected | Actual |
|---|---|
|  |  |

After Investigation we found out that opensea API can't receive the NFT data from the testnet.

Our plans onwards is to solve this problem either by switching to another marketplace or try deploying another contract. AI continue our development on the frontend application.