**Poker Ruby**

**Design**

Poker Ruby has similar design in Poker Design doc

def best\_hands(hands)

create Poker object for each hand

then compare each to find the winner

class Poker

parse hand

check for flush, straight

create kinds

call score\_hand()

see Poker Design doc for more details

**Some technical highlights:**

\_ use '--23456789TJQKA'.index(e) to convert rank character to rank values

\_ use ternary operator, and check for existing key to create ransk\_hash,

ranks\_hsh.key?(e) ? ranks\_hsh[e] += 1 : ranks\_hsh[e] = 1

\_ use each\_cons to check for straight = [1, 2,3,4,5]

ranks\_keys.each\_cons(2).all? { |a,b| b == a + 1 }

\_ use case, when to compare score

\_ reverse values sort in hash => ranks\_values hash

ranks\_hsh.sort {|a1, a2| a2[1] <=> a1[1] }.to\_h

**Tesing:**

learn more about RSpec 3.8, RSpec is getting too big, it has many features, and there will be RSpec 4 soon,

<https://relishapp.com/rspec/rspec-core/v/3-8/docs/expectation-framework-integration/configure-expectation-framework>

I will learn more In future project.

For this project, I only need simple assert\_equal in :minitest for this project

It has 17 unit tests and they all passed.

17 tests compare different combination of hands: flush vs straight, square vs full house, two squares,….

rspec poker\_spec.rb

.................

Finished in 0.00696 seconds (files took 0.12145 seconds to load)

17 examples, 0 failures

=end