Language: PYTHON 3.6

* Class
  + [PARENT CLASS IF APPLICABLE]
  + function1()
  + function2()
* Casino
  + runGames(int number of games)
* Game
  + Int CurrentRoundBet
  + Int CurrentRountPot
  + Int MaxRounds
  + Int CurrentRound
  + Int RoundState
  + Deck currentDeck
  + Player[8] Players
  + Card[5] communityCards
  + LinkedList<> ActivePlayers //circularly linked list // currently **active** players
  + --- data above, functions below ---
  + initializeNewGame() //resets everything back to default state
    - Specific AI
    - # games
    - # rounds per game
    - start$ = 1000
    - Start the game
      * Start round
      * Resolveround()
        + initializeAllTheThings() // initialize data in game class, create new randomized deck and assign player and community cards
        + Deduct blind bets // state 0 and 1
        + 1) PollAllPlayersForActionandApply() //state 2
        + 2) IfAllBetsEqualAndAllPlayersActedNextState()
        + Loop 1 and 2 until state 9
        + ResolveWinnerAndAwardMoney()
    - Loop with a new round.
  + getPlayerAction()
  + applyPlayerAction()
  + resolveRound() // eliminate links until one player is left, give money, initNewG()
  + giveMoney(Player p, money x) //track change of money of a player
  + incrementPot(int x) // Increments pot by x
* Deck
  + cards[52]
  + randomize() //randomizes the deck
* Card
  + Suit
  + Value
* Player
  + Int amountOfMoney
  + Int Position // on table
  + handRanking int[]
  + getBestHand(gamestate x, playerCards) // get pairedness, suitedness, straightedness
  + Player’s cards
  + --- data above, functions below ---
  + changeMoney(int m) //accepts negatives
* AI
  + getAction(gamestate x)
  + evaluateStrength(gamestate x)
  + predictEV(gamestate x)