

クラス

Player()

hand = [] #手札

state = 1 #1:同じ数字のみを出せる状態, 2:一つ大きい数字のみを出せる状態 3:2種類出せる状態, 4:何も出せない状態, 5:終了状態, 6:上がった状態

black_set = 0#set単位のblack

white_set = 0#set単位のwhite

black_game = 0#game単位のblack

white_game = 0#game 単位の white

関数

eqplusChange(field)

field を元に要素1つ目を eq, 要素2つ目を plus にした配列を返す

stateJudge(pl, eq, plus)

pl (Player クラス) の state を更新する

stateUpdate(pl1, pl2, eq, plus)

pl1, pl2(両方とも Player クラス) の state を更新する

exchange(black, white)

black, white の値を両替して要素1つ目を black, 2つ目を white にした配列にして返す

pointManage(pl)

pl (Player クラス) の hand に対する black, white の値を, 要素1つ目を black, 2つ目を white にした配列にして返す

setpointUpdate

pl (Player クラス) の現時点での持っている点数と hand の点数を足した値を black_set, white_set に格納する

gamepointUpdate(pl)

セット終了時の手札に対するポイントとゲームのポイントを加算して black_game, white_game に格納する

`initialize(pl1, pl2, lama_deck, field)`

`pl1, pl2` (Player クラス) を初期化して手札を `hand` に格納する。一つ目の要素を手札を配った後のデッキ、要素 2 つ目を `field` にした配列で返す

`pointMinus(pl)`

`pl` (Player クラス) の点数を黒があれば黒を、白しかなければ白の枚数を一枚減らす。何も持っていないければ何もしない。

`fold(pl)`

`pl` (Player クラス) の `state` を 5 (降りた状態) に更新する

`isFinish(pl1, pl2)`

`pl1, pl2` (Player クラス) が両方降りるもしくはどちらかが上がっていれば `False` を返す。それ以外の時は `True` を返す。

`gameFinish(pl1, pl2)`

`pl1, pl2` (Player クラス) のどちらかが 20 点を超えているときは `False`、それ以外は `True` を返す

`playCard(pl, lama_deck, field, eq, plus)`

基本的に各自に作ってもらう必要のある部分。行動選択をする。ここで行動に対する `pl` (Player クラス) の `hand` や `state` を更新する必要がある。一つ目の要素を行動選択後のデッキ、要素 2 つ目を行動選択後の `field` にした配列で返す