game.py

Class ChineseChecker (object) :

def isEnd(self, state, iter)

# Decide if the game is over. State is the current state and iter is current step count

def actions(self, state)

# Return possible actions current player can take in a list [(old\_pos), (new\_pos)]

def player(self, state)

# Return current player

def succ(self, state, action)

# Update the board for current step.

board.py

Class Board (object) :

def isEmptyPosition(self, pos)

# Check if pos is empty (no pieces there)

def adjacentPositions(self, pos)

# Return a list of adjacent positions (6 at most) of current positions

def getPlayerPiecePositions(self, player)

# Return a list of positions that player's pieces occupy

def getOneDirectionHopPosition(self, pos, direction)

# Return possible target hop position in the direction designated by direction . One can hop as long as there's only one piece on the line between current position and target position and the piece hopped over is at the middle point.

def getOneHopPositions(self, pos)

# Return all positions can be reached from current position in one hop in all 6 directions

def getAllHopPositions(self, pos)

# Return all positions can be reached from current position in several hops

def ifPlayerWin(self, player, iter)

# Return if all player's pieces reach the target triangle and iter is current step count

def isEnd(self, iter)

# Return if current board is an ending board (True/False, winner) and iter is current step count

def printBoard(self) # Print current board