

Basic Rule: 2 sides, make decision in turn, when meet certain requirements, game ends.

End requirements: if time limits meet or round meets maximum rounds or one of the players’ pieces become 0, game ends.

Initial: 3 rows of pieces (0,1,2 on one side, the other:5,6,7) time:20 min each player, randonly assign color to each player, red goes first)

Decision:a piece can be moved to an area if new area and old area are in the same diag, and can be moved toward opponent’s direc only if it is not a king. Plus, abs(new col-col)<=2

Object: Player

Fields:

String name

Int numPieces: When game starts, it is 12, and it decreases when eaten by the other player.

Bool color: 0, red,move first; 1,black,move afterwards

Time leftTime: at first, it is 20 min, and it decreases every round by the time the player takes.

Time stepTime:at the beginning of every round, it is 1 min, and it decreases by the time the player takes.

Bool isWin: 0, still in game; 1 win  
bool isTurn:0, cannot do anything, 1, player’s round

Method:

pause(): when the player finishes his move, his clock pauses, the others’ begins to run.

Object: Piece

Fields:

Bool side: 0,belongs to red, 1,belongs to black

Bool king:0 king, 1 not a king

Bool eaten: 0 eaten ,1 on the board

pair<int,int>: location of the piece

bool choose: 0 not chosen, 1 chosen by player

methods:

bool move(int,int): parameter: new location, if it can move to that loc, do it and return true; otherwise, return false, remain where it is.

Void eat(): after moving to a new loc, if the piece of opponent is in the same loc, it is eaten.

Object: Game

Fields:

Int vs: 0 robot vs human,1 robot vs robot,2 human vs human

Int n: size of the board, n\*n;

Int maxPieces: the origin pieces each player has

Methods:

Void Initial():set the location of every piece, the clock of each player

bool CheckWin():check if one of the player wins the game(: if time limits meet or round meets maximum rounds or one of the players win, game ends;)

Void change(): when one player finishes his round, change to another player

Void showRes(): after game over, a new window jumps out, display sth.

Void refresh(): when player choose an area or clicks the destination, the game background changes accordingly, that is, choosing an area, if there is a piece of his , true, piece selected. If piece seleceted and move legi, refresh the loc.