| Game |
| --- |
| * String players[]; //玩家名稱列表(為顯示誰贏場遊戲) * Player playerA, playerB; //兩名玩家(目前為寫死，之後可以擴充，直接用Player[]存) * ChessBoard cb; //棋盤(目前大小為寫死的(10\*10)) * boolean who; //"誰"的回合 * int winner; //贏家的index, 用來取字串players[winner] |
| * void round(){} //顯示棋盤，並等待玩家放置棋子 * void rotate(){} //換另一位玩家回合 * int checkGameOver(){} //回傳數字，小於等於零表示未結束，大於則為贏家index * String whoIsWinner(){} //回傳贏家名稱 |

| player |
| --- |
| * int use; //玩家使用哪種棋子(1表示黑棋，2表示白旗)   # final static int BLACK = 1;  # final static int WHITE = 2; |
| * Player(int use){} //建構子，順便指定使用什麼棋子 * boolean setPieces(ChessBoard cb, piece position){} //先判斷位子合不合理，再放置旗子，諾放置成功則回傳true，否則false |

| ChessBoard |
| --- |
| * int width; * int height; * int stateBuf[][]; //棋盤狀態，存棋盤各個位子的狀態 * int dir[4][2]; //方向(右上、右、右下、下)，用來判斷連線時使用 |
| * ChessBoard(int n, int m){} //建構子，順便指定長寬 * boolean checkPositionPossible(Piece pos, int use){} //回傳位子是否合理，use=0表示位子未被放置，use=1或=2可在判斷是否連線時被使用 * void place(int use, Piece pos){} //放置棋子(前面先保護判斷過，所以這邊只做放置動作，未做多的判斷) * void show(){} //輸出棋盤狀態 * int haveALine(){} //回傳是否有連線，諾有則回傳玩家使用棋編號 |

| Piece |
| --- |
| * int x; * int y; |
| * Piece(int x, int y){} //建構子，給定座標去建構 * int getX(); //回傳x * int getY(); //回傳y |