Comp2526 A#2c

Purpose: Testing the flexibility of your A2a code design with the addition of new features.

Description: You are to change the chess game to be a 3D chess game. Use the following guidelines:

3 boards displayed – either side by side or another view but all 3 boards must be visible at the same time

Movements – regular chess movements remain the same however pieces can move in their appropriate direction(s) up/down levels (boards) but limited to 3 positions (e.g. bishop can move diagonally up the 3 boards crossing 3 spaces, but cannot move straight up).

Code reuse is the key here. You will likely need to modify your class structure, extract common features and add general forms of your classes. E.g. I created an abstract Board class that had the same interface as the original ChessBoard (an interface could possibly be used instead).

```
public abstract class Board{

        public abstract void init();

        //etc

}

public ChessBoard extends Board{…….}

public ChessBoard3D extends Board{

        Board[] board = new ChessBoard[3];

        //etc

}
```

Then created a new ChessBoard3D from Board class. With a simple change to the instantiation of the Board from ChessBoard to ChessBoard3D no further code was modified. I then created an array of ChessBoard objects and used their methods as part of the ChessBoard3D methods

e.g.

```
public boolean isDiagonalPathClear(start, finish){
        if (onSameBoard) return board[0].isDiagonalPathClear(start,finish))
        else
                //check diagonalpath across boards is clear return answer
}
```

Marking Guide

Function	**Mark**
3D Board display	5
chess movements (queen, pawn)	20
Class hierarchy	5
Code reuse, OOP techniques	20

Assignment due March 17 midnight. Demo in lab that week.