```
public static void paintSquareLoc(Graphics g, int x, int y, GPiece piece, int rotation, int size){
    if(piece==null) return;
    g.setColor(Color.BLACK);
    g.fillRect(x, y, size, size);
    if(Character.isUpperCase(piece.getName().charAt(0))){
        g.setColor(Color.GREEN);
    }else {
        g.setColor(Color.YELLOW);
    }
    g.fillOval(x, y,  width: size-1,  height: size-1);

    g.setColor(Color.RED);
    if(rotation==1) {
        paintSymbol(g, x, y, piece.up().getSymbol(),  direction: 1, size);
        paintSymbol(g, x, y, piece.right().getSymbol(),  direction: 2, size);
        paintSymbol(g, x, y, piece.down().getSymbol(),  direction: 3, size);
        paintSymbol(g, x, y, piece.left().getSymbol(),  direction: 4, size);
    }else if(rotation==2){
        paintSymbol(g, x, y, piece.up().getSymbol(),  direction: 2, size);
        paintSymbol(g, x, y, piece.right().getSymbol(),  direction: 3, size);
        paintSymbol(g, x, y, piece.down().getSymbol(),  direction: 4, size);
        paintSymbol(g, x, y, piece.left().getSymbol(),  direction: 1, size);
    }else if(rotation==3){
        paintSymbol(g, x, y, piece.up().getSymbol(),  direction: 3, size);
        paintSymbol(g, x, y, piece.right().getSymbol(),  direction: 4, size);
        paintSymbol(g, x, y, piece.down().getSymbol(),  direction: 1, size);
        paintSymbol(g, x, y, piece.left().getSymbol(),  direction: 2, size);
    }else if(rotation==4){
        paintSymbol(g, x, y, piece.up().getSymbol(),  direction: 4, size);
        paintSymbol(g, x, y, piece.right().getSymbol(),  direction: 1, size);
        paintSymbol(g, x, y, piece.down().getSymbol(),  direction: 2, size);
        paintSymbol(g, x, y, piece.left().getSymbol(),  direction: 3, size);
    }else {
        System.out.println("rotation should be between 1-4 not "+rotation);
        paintSymbol(g, x, y, piece.up().getSymbol(),  direction: 1, size);
        paintSymbol(g, x, y, piece.right().getSymbol(),  direction: 2, size);
        paintSymbol(g, x, y, piece.down().getSymbol(),  direction: 3, size);
        paintSymbol(g, x, y, piece.left().getSymbol(),  direction: 4, size);
    }
    if(piece.changed){
        g.setColor(Color.BLUE);
        g.drawLine(x, y,  x2: x+(size-1),  y2: y+(size-1));
        g.drawLine( x1: x+(size-1), y, x,  y2: y+(size-1));
    }else {
        g.setColor(Color.BLACK);
    }
    g.drawRect(x, y,  width: size-1,  height: size-1);
}
```

This is the code used to print all of the pieces in the game. It first checks that it has been given a valid piece then prints a black background rectangle with the given size. Next the method checks which players piece it has been given and sets the appropriate colour, then an oval is draw with that colour. The oval has has a size of size - 1 to give a border around the piece. The colour is then set to red before drawing the pieces symbols. The location each symbol is drawn

at depends on the rotation int. This allows the piece to be drawn rotated without having to rotate it in the model, it is necessary to do this as some animations need to draw a rotated version but the model should not be affected by animations. I used static methods for drawing as it means I can be certain that the methods will only be effected by the variables passed into them, which is useful as I need the drawing to behave the same way every