```
57      public boolean move(int x, int y, String direction){
58          if(board[x][y]==null) return true;
59          int x2 = -1;
60          int y2 = -1;
61
62          switch (direction) {
63              case "up":
64                  x2 = x;
65                  y2 = y-1;
66                  break;
67              case "right":
68                  x2 = x+1;
69                  y2 = y;
70                  break;
71              case "down":
72                  x2 = x;
73                  y2 = y+1;
74                  break;
75              case "left":
76                  x2 = x-1;
77                  y2 = y;
78                  break;
79          }
80
81          if(offBoard(x2,y2)){
82              return false;
83          }else if(!move(x2, y2, direction)){
84              return false;
85          }else {
86              board[x2][y2] = board[x][y];
87              board[x][y] = null;
88              return true;
89          }
90      }
```

This is the move method in my Board class. The move method is called any time a GPiece needs to be move from one location to another. The direction String specifies which way the contents of board[x][y] should be moved. The move method returns true if the location is empty or movement is successful, if a GPiece would be moved off the board then move returns false. Because the board returns false when a GPiece would bove off the board the SSGame is able to recored the removal of that piece so that it can be undone when the move is undone. When moving the method recursively calls itself on the next square to move any GPiece there before it moves into it, this way there is no chance of a GPiece being overwritten. I feel like this is good code because it succinctly completes a specific task without any unnecessary complexity.