**Code Inspection**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Module** | **Severity** | **Code Snippet** | **Description** | **Suggestion for Fix** |
| MainGUI | 1 | Board board = new Board();  mineApplet = new MineApplet(board, 25);  board.setMyapplet(mineApplet);  mainPanel.add(mineApplet);    add(mainPanel,BorderLayout.CENTER); | Does not actually keep the board centered. | add the board to the panel with scaling, to allow resizing of the frame. |
| MineApplet | None | None | All methods seem to prevent errors and inefficiency | None |
| Board | 1 | private int[][] getAdjacent(int i, int j) {    int adj[][] = new int[8][2];  adj[0][0] = j-1;  adj[0][1] = i-1;  adj[1][0] = j;  adj[1][1] = i-1;  adj[2][0] = j+1;  adj[2][1] = i-1;  adj[3][0] = j-1;  adj[3][1] = i;  adj[4][0] = j+1;  adj[4][1] = i;  adj[5][0] = j-1;  adj[5][1] = i+1;  adj[6][0] = j;  adj[6][1] = i+1;  adj[7][0] = j+1;  adj[7][1] = i+1;    return adj;  } | Must be a more efficient way to compute the adjacent spaces. | have each space be connected to an array of adjacent spaces during initialization of board |
|  | 2 | **if** (!foundAdjacentMine) {  **for** (**int** i = 0; i < 8; i++) {  **try** {  hidden[adj[i][0]][adj[i][1]] = **false**;  q.add(adj[i]);  **catch**(IndexOutOfBoundsException e) { /\*ignore \*/ }  }  } | Loops sometimes due to spaces being added to the queue multiple times. | Maintain a list of spaces that have been checked, and don’t add the space to the queue if it’s already gone through. |
| GameClient | None | None | GameClient is temporary at this time, TODO: add API for connecting game to server | None |
| GameServer | 2 | ServerSocket listener = new ServerSocket(9898); | Works 99% of the time, but since it’s run on a shared network it’s possible to have port conflicts that this code does not handle | Create an optional port or port range to use if port is in use. |
| GameServer | 1 | public boolean checkForDups(String[] nameList){  duplicates=false;  for (j=0;j<nameList.length;j++)  for (k=j+1;k<nameList.length;k++)  if (k!=j && nameList[k] == nameList[j])  duplicates=true;  return duplicates;  } | Not the most efficient implementation for checking duplicates of names in the server | Find a different implementation, possibly using a hash set |
| GameServer | 3 | public String[] nameList = new String[256]; int ind = 0; ....  ....  nameListy[ind] = input;  ind++; | There is no safety measure to stop array out of bounds errors | Create a check to make sure names do not cause an array out of bounds error |