I plan on putting all these functions in a class called MahJongBoard.

To store the tiles within the board I plan on using a Hash Map – Map<List<Int>, Tile>

This contains the coordinates of the tile (List<int>), and the tile (Tile) itself.

The coordinates will be given as such – List<int> coords = Arrays.asList(xCord, yCord, zCord)

The “deck” of tiles will be stored in a global Array List for ease of access/manipulation – List <Tile> deck

Once all tiles are instantiated they will be positioned on the board like this:

*Public position Tile(int x, int y, int z) {*

*for (Tile tile: deck) {*

*board.put(Arrays.asList(tile.xPos, tile.yPos, tile.zPos*

*}, tile);*

*}*

*}*

To check for an open tile, manipulate the tile’s coordinates and check the board to see if that coordinate returns null (empty tile). This can be done recursively,

Public boolean isTileOpen(Tile t) {

Int x = t.xPos;

Int y = t.yPos;

Int z = t.zPos; //Check left & right tiles

boolean isRightOpen = board.get(Arrays.asList(x + 1, y, z)) == null;

boolean isLeftOpen = board.get(Arrays.asList(x - 1, y, z)) == null;

boolean isTopOpen = board.get(Arrays.asList(x, y, z)) == null;

return (isTopOpen && isRightOpen && isLeftOpen);

}

This method is simple, using the map structure and the key (coordinates) we can easily return the tile.

Public Tile getTile(int x, int y, int z) {

return board.get(Arrays.asList(x, y, z);

}

To check if the tile can cast shadows, we perform a method similar to isTileOpen:

Public boolean canCastShadows(Tile t){

Int x = t.xPos;

Int y = t.yPos;

Int z = t.zPos;

If(!isTileOpen) return false;

//Check tiles above, repeat as necessary for z layers

boolean isTopRightOpen = board.get(Arrays.asList(x+1, y, z+1)) ==null;

boolean isTopLeftOpen = board.get(Arrays.asList(x-1, y, z+1)) ==null;

return (isTopRightOpen && isTopLeftOpen && isLeftOpen);

}

To get a tiles location, use something like this:

getLocation(Tile t){

//Repeat for other coords

t.xPos;

//or return the key stored with the given tile

List<int> cords= t.getKey();

Int xCord = Cords.get(0);

}

