**Muhammed Emin Güre - 2014400171**

**Problem Description**

We will create a game named P\*G. There will be a board which consists of 4 rows that are consisting of 4 letters but 1 ‘\*’ character only in a row. The user will decide the initial board configuration or she can use default board configuration. After that she will decide how many times she wants to move the ‘\*’ character. Lastly, she will pick a side as left, right, up or down to swap the ‘\*’ character. If the swapped character is ‘P’ or ‘G’, program will print ‘I’ instead of ‘P’ or ‘G’. And for ‘P’ changes to ‘I’, score will be increased by 1 or if ‘G’ changes to ‘I’, score will be increased by 5. After every move program will print the new board configuration and the total score.

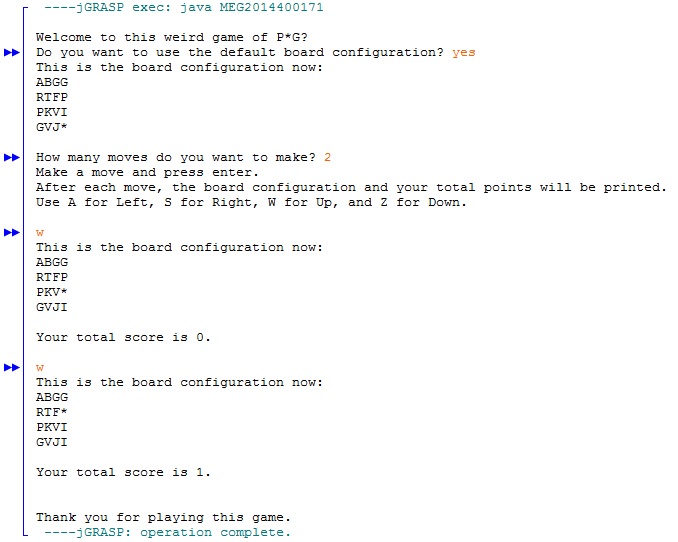
**Problem Solution**

The program has 3 method except main method. “initialBoard” method returns a string whose length is 16 as the board of the game. If the user says no to this method, then she needs to enter 4 rows to configure the initial board. Otherwise, program will assign a default board. Taking input from user as row1,row2 etc. is being made by a for loop. “printBoard” method returns a string like prior method and also prints the board as a table whatever the string which is called as a parameter into the method. “move” method is actually the method that makes the important part of the game. It takes a number from user and iterate the rest of the method. A for loop makes this iteration. After that according to user’s input string ‘str’ changes every time. And also score changes if its requirements are provided. At the end of every iteration, with calling “printBoard” method program prints the last board configuration. Lastly, the number of for loops that is used is 2.

**Implementation**

1 import java.util.\*;  
 2   
 3 public class MEG2014400171{  
 4   
 5 public static void main(String[] args){  
 6   
 7 Scanner input = new Scanner(System.in);  
 8 System.out.println("Welcome to this weird game of P\*G?");  
 9 String board=initialBoard(input);   
 10 printBoard(board); // prints the initial board of the game  
 11 System.out.print("How many moves do you want to make? ");  
 12 int re=input.nextInt(); // re: how many moves the user wants to make  
 13 System.out.println("Make a move and press enter. \nAfter each move, the board configuration and your total points will be printed. \nUse A for Left, S for Right, W for Up, and Z for Down.\n");  
 14 move(board,re,input); // calls method to do the important part of the game  
 15 System.out.print("\nThank you for playing this game.");  
 16   
 17 }  
 18   
 19 /\*This method returns a string as the board of the game according to user's decision\*/  
 20 public static String initialBoard(Scanner in){  
 21   
 22 String str=""; // str is assigned for reassigning it for every situation  
 23 System.out.print("Do you want to use the default board configuration? ");  
 24 String yn=in.next(); // yn: yes or no  
 25 if(yn.equalsIgnoreCase("yes")){  
 26 str="abggrtfppkvıgvj\*"; //default board configuration  
 27 }  
 28 else if(yn.equalsIgnoreCase("no")){  
 29 for(int i=1;i<=4;i++){// this for loop takes inputs to assign a string as the board  
 30 System.out.print("Enter row "+i+" of the board: ");  
 31 str+=in.next(); // it adds every row one after another  
 32 }  
 33 }  
 34 str=str.toUpperCase(); // makes the board upper case so the program do not complain about asciis  
 35 return str;  
 36 }  
 37   
 38 /\* This method takes the iteration number and board string and prints the board after every move of user by calling printBoard method\*/  
 39 public static void move(String str,int re,Scanner in){  
 40   
 41 int sum=0; // sum: total score of user according to her moves  
 42 for(int i=1;i<=re;i++){ //this for loop makes the move how many times the user wants  
 43 String move=in.next(); // takes input from user as to play game  
 44 int a=str.indexOf("\*"); // finds the index number of \* character  
 45 if(move.equalsIgnoreCase("S")){   
 46 if((a+1)%4!=0){ // at the 4th column of the board \* character cannot be swapped to right  
 47 if(str.charAt(a+1)=='P' || str.charAt(a+1)=='G'){ // check for whether the swapped character is P or G   
 48 if(str.charAt(a+1)=='P'){  
 49 sum++; // adds 1 to total score for P  
 50 }  
 51 else{  
 52 sum+=5; // adds 5 to total score for G  
 53 }  
 54 str=str.substring(0,a)+"I\*"+str.substring(a+2); // reassign the string with changes in case of P&G rule of the game  
 55 }  
 56 else{  
 57 str=str.substring(0,a)+str.charAt(a+1)+"\*"+str.substring(a+2); // reassign the string with changes   
 58 }  
 59 }  
 60 }  
 61 else if(move.equalsIgnoreCase("A")){  
 62   
 63 if((a+1)%4!=1){ // at the 1st column of the board \* character cannot be swapped to left  
 64 if(str.charAt(a-1)=='P' || str.charAt(a-1)=='G'){  
 65 if(str.charAt(a-1)=='P'){  
 66 sum++;  
 67 }  
 68 else{  
 69 sum+=5;  
 70 }  
 71 str=str.substring(0,a-1)+"\*"+"I"+str.substring(a+1);  
 72 }  
 73 else{  
 74 str=str.substring(0,a-1)+"\*"+str.charAt(a-1)+str.substring(a+1);  
 75 }  
 76 }  
 77 }  
 78 else if(move.equalsIgnoreCase("W")){   
 79 if(a>3){ // at row 1 of the board \* character cannot be swapped to upward  
 80 if(str.charAt(a-4)=='P' || str.charAt(a-4)=='G'){  
 81 if(str.charAt(a-4)=='P'){  
 82 sum++;  
 83 }  
 84 else{  
 85 sum+=5;  
 86 }  
 87 str=str.substring(0,a-4)+"\*"+str.substring(a-3,a)+"I"+str.substring(a+1);  
 88 }  
 89 else{  
 90 str=str.substring(0,a-4)+"\*"+str.substring(a-3,a)+str.charAt(a-4)+str.substring(a+1);  
 91 }  
 92 }  
 93 }  
 94 else if(move.equalsIgnoreCase("Z")){   
 95 if(a<12){ // at row 4 of the board \* character cannot be swapped to downward  
 96 if(str.charAt(a+4)=='P' || str.charAt(a+4)=='G'){  
 97 if(str.charAt(a+4)=='P'){  
 98 sum++;  
 99 }  
100 else{  
101 sum+=5;  
102 }  
103 str=str.substring(0,a)+"I"+str.substring(a+1,a+4)+"\*"+str.substring(a+5);  
104 }  
105 else{  
106 str=str.substring(0,a)+str.charAt(a+4)+str.substring(a+1,a+4)+"\*"+str.substring(a+5);  
107 }  
108 }  
109 }  
110 printBoard(str); // by calling this method, program prints the last board configuration after the last move  
111 System.out.println("Your total score is "+sum+".\n"); // prints the total score after every move  
112 }  
113 }  
114   
115 /\* This method takes a string of board and prints as a table\*/  
116 public static String printBoard(String str){  
117   
118 System.out.println("This is the board configuration now:");  
119 System.out.println(str.substring(0,4)+"\n"+str.substring(4,8)+"\n"+str.substring(8,12)+"\n"+str.substring(12)+"\n"); // prints the board every row as a line  
120 return str;   
121 }  
122   
123 }

**Output of the program**



**Conclusion**

This version is actually my 3rd try of coding this game. In every time program worked smoothly. But this version is the most understandable and stable. It works fine without any error at this time also.