



Programming Assignment – 1

Topic: GUI and Event handling

Due: Feb 9, 2016

1. Black or White

There are 16 squares on a 4 x 4 sized game board, which are numbered from 0 – 15 as shown in the figure below. When the game starts, each square is randomly assigned either a black or a white background. The goal is to turn all 16 squares white or black by (clicking a set of 10 switches).

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

(a) Initial state

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

(b) Target state 1

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

(c) Target state 2

The only way to change the color of each square is to push pre-defined “switches”. Each switch is connected to 3 – 5 squares. Every time you push a switch, squares connected to that switch change their colors (black -> white, and white -> black). The following table contains the list of switches and connected squares.

Switch	Square
A	0, 1, 2
B	3, 7, 9, 11
C	4, 10, 14, 15
D	0, 4, 5, 6, 7
E	6, 7, 8, 10, 12
F	0, 2, 14, 15
G	3, 14, 15
H	4, 5, 7, 14, 15
I	1, 2, 3, 4, 5
J	3, 4, 5, 9, 13

For example, if switch G is pushed, three squares numbered 3, 14 and 15 toggle their color between black and white.

We want to know the minimum number of switches one has to push in order to make all squares black or white. We also want to know the sequence of switch presses. For example, if we have to push five switches A, B, C, D, and E, both ‘5’ and {A, B, C, D, E} have to be reported. For multiple solutions, print any valid solution. If the problem is unsolvable for a given input, print “No solution”.

Hint: You should not have to consider more than $^{10}C_0 + ^{10}C_1 + ^{10}C_2 + \dots + ^{10}C_{10}$ combinations.

2. Submission

You will have to do two things: 1) bring the App and your code, and demonstrate them to the TA or the course instructor in class (on the day of submission), and 2) upload your source code as a zip file (name the file as yourpid_lastname.zip) to Sakai (the deadline will be the same as the submission day).

3. Point Distribution

Tasks	COMP 590	COMP 790	Grading
1. Make 16 squares to form a 4 x 4 grid using 16 Android buttons or ImageViews. Alternative ways to draw the board is also okay. When your app launches, background of each square will be either black or white (randomly assigned).	20	20	
2. Add 10 buttons that represents the switches: A to J. Clicking them should flip the color of the corresponding squares according to the Table.	10	10	
3. A human player should be able to play the game by pushing the buttons/switches. Add two text views for 'Move Count' and 'Sequence'. As the player plays, update them appropriately. If the person wins, show a pop up message (use: Toast)	10	10	
4. Make two buttons below the grid and name them 'Restart' and 'Auto'. Clicking 'Restart' reinitializes the game, i.e. randomly assigns colors and clears the text views.	5	5	
5. Clicking 'Auto' computes the solution automatically from the current state and shows the minimum number of moves and the sequence. If your algorithm says there is no answer, just print 'No solution'.	15	15	
6. Extra Credit – When the 'Auto' button is pushed, animate the moves. For example, if the minimum number of moves is 3 and the 'Sequence' is (A, B, C), the color of squares should be changed in 3 steps: (1) Switch A is pushed -> squares 0,1,2 change colors, (2) switch B is pushed -> squares 3,7,9,11 change their colors, (3) then switch C is pushed -> squares 4,10,14,15 change their color. Use a delay between two steps to make it look like an animation. You should also highlight the switch that is pushed.	5	5	
Total:	60 + 5 (EC)	60 + 5 (EC)	