README file for OOP - Project :

**מגישים:**

יבגני בורקו: 309543247

אלכסיי בוטוב: 317552610

================================================================================

General Description of program:

The Main purpose of the exercis is to build a fully functional PC Game "Lights" using the "SFML" Graphic Library.

The program defines the following 5 classes: Controller, Point, Level1,Level2, Level3.

and starts the game, the purpose of the game is to lit all lights by fliping the connectons between them until all lights are connected, when all lights are on the game loads to the next level until all levels are completed and than it finishes.

================================================================================

Design

The UML is very small, there are not so many objects in the game, mostly copies of points and edges.

to find the neighbours of the central light sourse i used a recursive function "traverse" to traverse between neighbours untill all of them are lit.

================================================================================

List of files

1) Controller.cpp - runs the menu window and the main function, runs alongside the other classes as well, and also it runs all nesting classes such as Levels 1-3.

2) Point.cpp – defines the data structure of each point in the game.

3) Level1.cpp– saved level1 file in the code as a vector of vectors of int.

4) Level2.cpp – saved level1 file in the code as a vector of vectors of int.

5) Level3.cpp - saved level1 file in the code as a vector of vectors of int.================================================================================

Data structures:

1) Controller: holds hpp functions using SFML, array of objects, insertion to files.

2) Point stores the data structure of each point in the game.

3) Level1 holds a matrix of the stored file.

4) Level2 holds a matrix of the stored file.

5) Level3 holds a matrix of the stored file.

================================================================================

Worth to mention algorithms:

Recursion

================================================================================

Known Bugs: ================================================================================

Other notes:

The levels are planned beforehead and stored in the game code itself

if you want to add more levels you'll need to extand the if conditions in Run function inside controller.cpp also to build a vector of points using the same build function as in other level clases but with more conditions, to build the vector of neighbours from it as well using a for loop by m\_points size, also to store a level variable in controller.h and finally to increase levelcounter to 4.

\*The code is non generic as you might see, i was working alone and even late for the submition date of this excersise but it was a very educational experience for me.