SPA homework 03

The aim of this homework is to strengthen the student's knowledge of working with search algorithms.

Task 1 (max 1 point from outcome 4)

Attached is the application you need to create (Task1.exe), and it consists of the following: when starting the application, the user enters the row and column of the starting point A and the row (*redak*) and column (*stupac*) of the endpoint B (row number goes from 1 to 20, and the column number from 1 to 40). After that, the application needs to devise a path from point A to point B and every 100 milliseconds plot where we are at the moment. Point A and point B can be located in any allowed place (try how the attached app works with A = 1, 1 and B = 20, 40, and how it works with A = 20, 20 and B = 10, 10).

Task 2 (max 1 point from outcome 4)

The previous application is great in case there are no permanent obstacles. However, if walls appear on the playground, we will have to change the way we get from A to B so that we do not go through the walls, but bypass them. The second application (Task2.exe) demonstrates this mode of operation (try out how the attached application works with A = 1, 1 and B = 1, 40 - we will neatly bypass the wall and reach the goal. This principle applies to as many walls as we set up on the playground) .

Your task is to program the application so that the player on the way from A to B bypasses the walls and happily reaches the destination. There are many ways to do this, and the simplest is by applying Dijkstra's shortest path algorithm (https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm). Great implementation that **you can use** (you can use others, of course, and you can write your own, but this is the shortest way to the solution): http://rosettacode.org/wiki/Dijkstra%27s_algorithm.

Bonus

All solutions of the Task 2 made in SFML enter the 1, 2, and 3 dollars bonus competition. Bonuses will be divided according to the visual WAAAAUUUU effect caused by assistants, teachers and fellow students. Just set up SFML according to the project from the first or second homework (you can also use those projects).

Submission of the solution

When you're done, put the solution on your GitHub and send your assistant an email with a link.