Lab 1: Summer Orienteering

In this lab, optimal paths for orienteering will be generated based on the terrains. A computer-friendly inputs are given so that an algorithm can be used to determine the best path.

States: All the pixels within an area of 395x500 pixels with each pixel being equivalent to 10.29 m in longitude (X) and 7.55 m in latitude (Y).

Initial State: Starting co-ordinate location(pixel) provided by the path file.

Actions: Move to the neighboring pixel in one of the 4 cardinal directions.

Goal Test: Target co-ordinate location(pixel) provided by the path file.

lab1 Class

To run the program, lab1 class is run.

It takes 4 arguments, in order: terrain-image, elevation-file, path-file, output-image-filename.

On passing the correct number of arguments, the program initializes all the terrains with different speeds.

The program then reads the terrain image, converts the elevations file into an array, reads the pixels to be travelled from, through and to, from the path file, and calls the 'aStarSearch` class to find the path.

The program then outputs an image of the input map with the optimal path drawn on top of it and outputs the total path length in meters the terminal.

aStarSearch Class

The A * Search is used to find the optimal path between the start point and target point. At each step it picks the pixel according to the f_score which is a parameter equal to the sum of g_score and h_score. At each step it picks the pixel having the lowest f_score, and processes that pixel.
g_score is the movement cost to move from the current pixel to the next pixel, following the path generated to get there.
h_score is the estimated movement cost to move from the current pixel to the target pixel. This is often referred to as heuristic.

The cost function generates the g_score by calculating the Euclidean distance, dist, between the current and next pixel and then uses ((dist/2)/speed of current pixel) + ((dist/2)/speed of next pixel) to take the average time based of the 2 pixels. The cost function is admissible, as it calculates the cost of path between neighboring pixels and Is added to the total cost of the path.

The heuristic function generates the h_score by returning the Euclidean distance between the current pixel and the target pixel. The heuristic function assumes that the path passes through the best terrain.

A priority queue is used to return the pixel with the lowest f_score first.

Terrain Class

The terrain class represents the different terrains provided. It stores the name of the terrain, the color representation on the map and the speed you will be able to proceed at in the given terrain. Based on the images, these normalized speeds were used:

Open land	1.0
Rough meadow	0.4
Easy movement forest	0.75
Slow run forest	0.6
Walk forest	0.5
Impassible vegetation	0.1
Lake/Swamp/Marsh	0.01
Paved road	0.95
Footpath	0.8
Out of bounds	0.00001

Pixel Class

The pixel class represents an individual pixel. It stores location using Point provided by Java, the g_score of the pixel, the h_score of the pixel, the f_score of the pixel and the parent pixel to keep track of the path. It has function to compare pixels to be used by priority queue while searching the path by A * search algorithm.

Run the program

1. Compile the files present in src directory using *javac *.java*
2. Run the command using java lab1 terrain.png mpp.txt path.txt output_image_file_name.png

Note: In 2nd step, make sure it is lab1, not lab1.java

Output

The program outputs an image of the input map with the optimal path drawn on top of it and also the total path length in meters to the terminal.