Bagirishya Rwema Dominique

Canegie mellon University  bdominiq@andrew.cmu.edu

**data structures and algorithms for data engineers - Assignment 1**

Documentation

First and foremost, the problem tasked with involved developing a C++ program to address the processing of fixation points obtained from automatic eye trackers. These fixation points, which represented the user's visual attention, included identifiers, x and y coordinates, forming a scanpath. The primary goal was to create a unique identifier for each distinct fixation point based on its coordinates. According to what has been explained during recitation and in the document, the program had to handle multiple test cases, each terminated by the coordinates (-1,-1), and output a list of distinct fixation points with their corresponding unique identifiers. The uniqueness of a fixation point was determined solely by its x and y coordinates, and the identifiers started from 1, incrementing based on the order of encounter. I used sample input that was given in the question sheet, but it can be tasted with different one.

So, my code is intended to process fixation points from an input file (one that is located in the data directory of my zip file), identify distinct points, and write them to an output file. To test the functionality, a set of fixation points was provided in the "data/input.txt" file. The program read the input, identify distinct fixation points, and then write the results to the "data/output.txt" file. After executing the program, the output file contains the distinct fixation points for each test case. The effectiveness of the program can be assessed by examining the contents of the output file to ensure that distinct fixation points are correctly identified and written as it has been asked (you can also use different test cases).

When it comes to the functionality of the program, as it has been said, the program processes fixation points from an input file, identifying distinct points and writing them to an output file. The ‘processFixationPoints’ function reads fixation data, checks for distinctiveness, and stores unique points in a structure array. The main loop iterates through test cases, invoking this function to handle fixation points. Distinct points are then written to an output file by the ‘writeDistinctFixationPoints’ function. The program appends my Andrew ID to the output file (as it was asked) and separates each test case with a line of asterisks.

***Down here are my test cases:***

|  |  |
| --- | --- |
| Input | Output |
| 2  1 382 353  2 484 328  3 995 641  4 715 242  5 995 641  6 995 641  7 710 245  8 715 242  9 995 641  10 -1 -1  1 200 300  2 400 500  3 600 700  4 800 900  5 200 300  6 600 700  7 1000 1100  8 800 900  9 1200 1300  10 -1 -1 | bdominiq  1 382 353  2 484 328  3 995 641  4 715 242  5 710 245  \*\*\*\*\*\*\*\*\*\*  1 200 300  2 400 500  3 600 700  4 800 900  5 1000 1100  6 1200 1300  \*\*\*\*\*\*\*\*\*\* |