ADA University
School of Information Technologies and Engineering
CSCI 2304: Data Structures & Algorithms
Spring 2020

Data Structures & Algorithms Project

Prepared by
Sadig Akhund
for CSCI-2304 at
the ADA University
May 2020

Easy Metro

1. Introduction

Project Description

This is a project proposal for partial fulfillment of the requirements of the *CSCI 2304: Data Structures & Algorithms* course in the School of Information Technologies and Engineering at ADA University, Baku, Azerbaijan.

A Desktop app designed to guide pedestrians in the metro to their desired destinations, in the shortest, the fastest or with the minimum amount of transfers possible.

Program's Functions

Application will have following functions:

- Choose a city - Welcome screen of application which will introduce you the map of the city you will choose the departure station and destination station. Additionally, you can choose to stop at the intermediary (Via) station, which is optional.
- Add/edit cities - You can create your own city and define own map of the Metro.
- Choose start and destination points - You can pin your location and destination.
- Choose different routes - You can select whether you want to travel using the shortest, the fastest or with the minimum path.
- Refresh – Returns application to its initial state (e.g. clears every text, returns left navigation bar to the initial position if it ever was moved).
- Expand – By pressing this button application will choose full detailed version of the calculated path, with all the intermediary stations.
- Save – From the File menu you can save the current results.
- City – From the Edit menu you can select from available cities.
- Scale Options – From this menu you have different options which will apply different algorithms for scaling (zooming) the map. Each algorithm may vary by its speed and quality.
- Look & Feel – From this menu you can choose application's look and feel from available options. Usually, default is "Metal".
- Show Map Details – Located at the Settings menu, this option allows you to display full information about the Metro network (e.g. lists all the stations, shows the possible routes from each station).
- Drag – Located at the Settings menu, this option when pressed enables the function of dragging the left navigation bar.

2. Requirements

Functional Requirements

1. The added city folder should contain following:
    - map.png image of the city's metro map
    - fare.txt which will contain the cost of transfer between lines and the currency
    - line-color.txt which will define color of the output according to the line color
    - lines folder which will have the list of lines and each line contains sequence of stations.
2. The user input is handled by a method which is used to check input by user. Malformed input cases:
    - Departure station was not specified.
    - Incorrect Departure station was given.
    - Incorrect via station was given; but only when it is not empty
    - Destination station was not specified.
    - The Destination station was given but not correctly.
    - When user tries to input the same station.
    - When there is no possible route, or the lines are not connected.
3. Data Requirements:
   The name of the city, the network of the Metro infrastructure: all the names of the Metro lines and corresponding stations. For now, the approximated or an average value for time and distance between stations are given. For example, the first line of each 'line'.txt defines the average time and distance between subsequent station:

```
10 2

Kirazli Bagcilar

Bagcilar Meydan

Ucyuzlu

Menderes

…
```

   Here it is implied that it takes 10 minutes to travel and it 2 km between "Kirazli Bagcilar" and "Bagcilar Meydan" or any other adjacent station. This approach might not have the best accuracy, however due to lack of data, it is an only option. The algorithm can easily be modified by adding an extra field after each line (which shows the name of a station) representing the amount of time and distance it takes to get from that station to the next one.

Performance Requirements

The program does not necessarily require a beefy computer; however, some actions can require complex computations. In addition, it was originally developed on 1366 x 768 resolution, but other options are considered as well, so that is the recommended resolution.

Usability

There can be terminals at stations, whenever inexperienced Metro user does not know which train to take it comes in handy. It is easy to use – does not require a

help of a Metro personnel to use it; The program can be used by people with no training, and possibly no understanding of English.

Personalization

I have made so that user could personalize some general features of the program, or start-up settings by editing "settings.cfg" in project folder. Here is the file itself:

```
[Display]

    COMPONENT_MARGIN=10

    DISPLAY_PANE_LINE_LENGTH=85

    DEFAULT_FONT=Consolas

[Variables]

    RATIO=0.86

    VSCROLL_RATE=16

    HSCROLL_RATE=16

    ZOOM_SCALE=0.1

    CITY=Istanbul

[Colours]

    TEXT_BRIGHT_COLOR=255, 255, 255

    TEXT_DARK_COLOR=110, 159, 255
```

- COMPONENT_MARGIN – defines the margin between components
- DISPLAY_PANE_LINE_LENGTH – defines the length of the vertical text that could be fit in one line of the display pane
- DEFAULT_FONT – the font of text-fields and display pane
- RATIO – the portion of screen program will have the size of
- VSCROLL_RATE – the vertical scroll rate of a map
- HSCROLL_RATE – the horizontal scroll rate of a map
- ZOOM_SCALE – the ratio of zooming
- CITY – the start-up city
- TEXT_BRIGHT_COLOR – brighter color for the text
- TEXT_DARK_COLOR – darker color for the text

In addition, user could add own city if all the necessary data provided.