

2IF10 Componenten 1

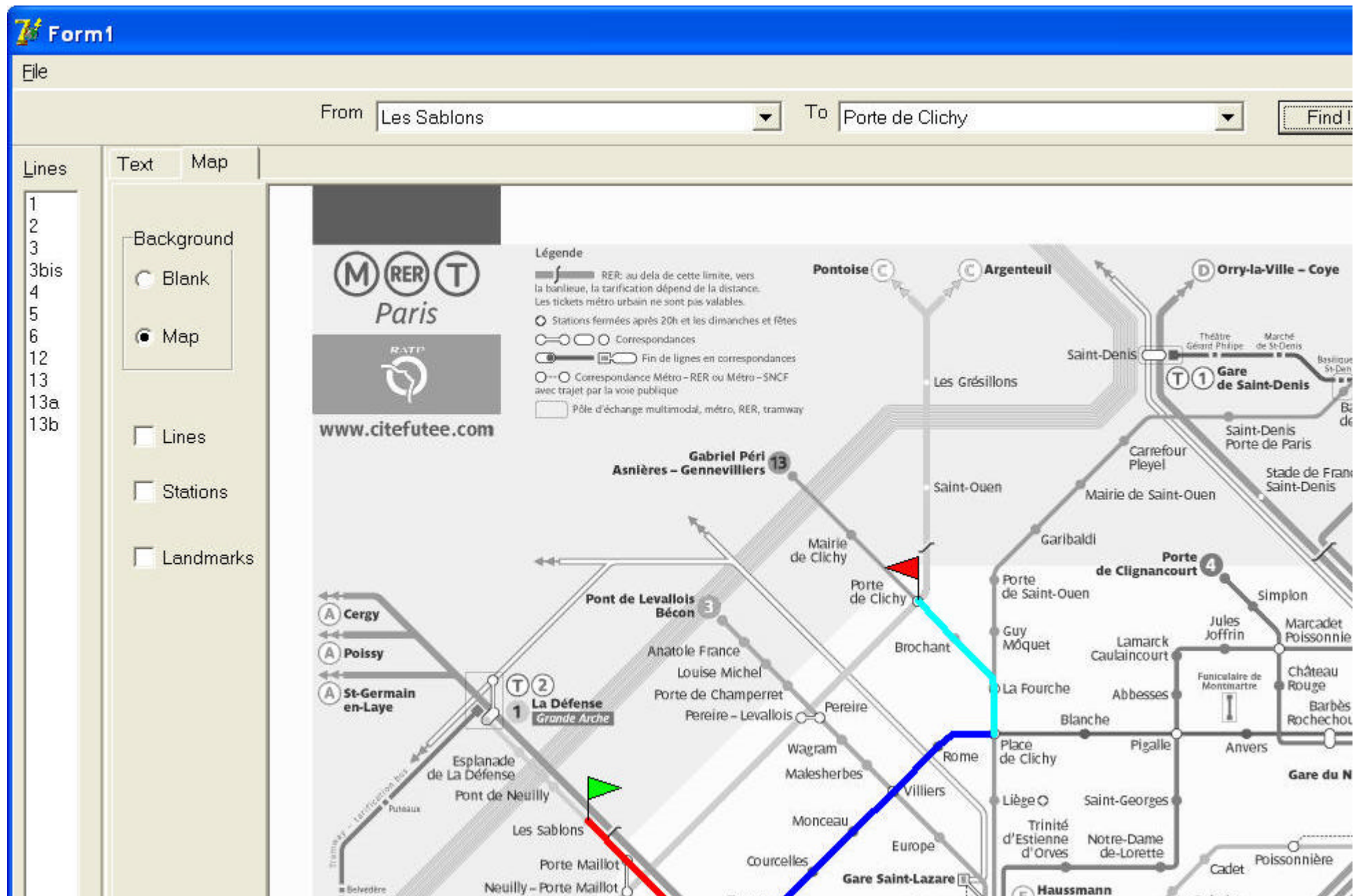
Practicum 5 (Metro Planner Case Study; Metro Planner Algorithms)

Opdracht 1

Deze opdracht heeft betrekking op de planner algoritmen die nodig zijn voor de Metro Planner applicatie. Het gaat om de algoritmen voor het vinden van een optimale route tussen twee stations in een metronetwerk.

De algoritmen daarvoor zijn behandeld op het college van 15 februari. Zie de website voor een samenvatting van de behandelde algoritmen. Het bijgaande Delphi-project is een vrijwel volledige applicatie. Alleen het algoritme dat een route met een minimaal stops moet opleveren is vervangen door een dummy-versie die een lege route oplevert. Doel van deze opgave is het vervangen van deze dummyversie door een implementatie van het op het college behandelde algoritme.

De bijgaande ZIP-file bevat in de directory Demo een executable die er bij uitvoering ongeveer als volgt uit ziet:



Na selectie van een start- en een stopstation (via de comboboxes of door rechtermuisklik op de kaart) kan door aanklikken van de Find-button een optimale route bepaald worden. Via de radiobuttons kan bepaald worden of het aantal Stops dan wel het aantal Transfers (overstappen) geminimaliseerd wordt. De gevonden route wordt op de kaart getekend. Op het tabblad Text wordt in de rechtertabel ook een tekstuele versie van de gevonden route weergegeven.

Verder kan de weergave in de kaart nog aangepast worden via de checkboxes en radiobuttons aan de linkerkant. Het is ook mogelijk in de lijst helemaal links een of meer lijnen te selecteren (klik of Ctrl-klik). De geselecteerde lijnen worden dan op de kaart weergegeven.

Het is ook mogelijk een ander netwerk te laden. Gebruik daarvoor het File|Open menu, navigeer eventueel naar de directory Networks en selecteer daar een van de .nwk files. Naast het (onvolledige) netwerk van Parijs is er een netwerk van Praag gegeven. Het netwerk Empty levert een leeg netwerk op en dient als default bij het ontbreken van andere netwerkfiles.

Gevraagd:

1. **Bewaar de oorspronkelijke project directory en voer onderstaande opdrachten uit op copieen daarvan.**
2. Experimenteer met het programma (executable in de map Demo) en bestudeer de code. Het gaat vooral om de volgende delen daarvan:
 - De event handler FindButtonClick in de unit Main;
 - De unit Planner.pas; deze definieert een paar hulpklassen voor het weergeven van een route en geeft tevens een abstracte class TPlanner, die de interface van plannercomponenten vastlegt;
 - De unit MinTransfersPlanner; deze dient voor het berekenen van een route met een minimaal aantal overstappen. Deze unit is volledig uitgewerkt, op basis van de op het college geschetste methode van Breadth First Search op *lijnen*;
 - De unit MinStopsPlanner; dit is de unit die aangepast moet worden. De gegeven versie van de method FindRoute levert slechts een lege route op. Deze methode moet vervangen worden door een implementatie van de Breadth First Search op *stations*. Zie voor een beschrijving hiervan de collegenotitie op de website.
3. Geef de code van de method FindRoute door systematische omzetting van het algoritme uit de collegenotitie naar Delphi.

