TSP Visualizer 1.1

This project aims at graphically representing a solution of the Travelling Salesman problem. The basic data of the problem are the number of cities and their distances as a matrix.

To have a better understanding we add the names of the cities and their positions on a map. You can also define the path that needs to be displayed or provide it as a command line argument.

There are three examples within this project in the *data* subdirectory. The solutions have been found using the **simulated annelaing** algorithm provided with this project:

- 13 cities in USA with a minimum distance of 7293 miles
- 32 cities in France with a minimum distance of 4753 kms
- 100 cities in France with a minimum distance of 7508 kms

1. Installation

This project is written in **python** and relies on **pygame** for the graphics part. You will need to install pygame 2.0 for the project to function. The current version I am using on my computer (Linux Ubuntu 20.04) is Anaconda with python 3.8.5 and pygame 2.0.0.dev6 (SDL 2.0.10).

1.1 Anaconda

If you are using Anaconda, please also install the following packages:

```
$ conda update -n base -c defaults conda
$ conda install -c conda-forge pyperclip
```

The **pyperclip** package is necessary in order to use the clipboard and copy a solution inside it using the F9 key.

1.2 LKH

LKH is an effective C implementation from **K. Helsgaun** of the Lin-Kernighan heuristic for solving the traveling salesman problem. You can download it here as version 2.0.9 of July 2018.
To install LKH, type the following:

```
tar xvfz LKH-2.0.9.tgz
```

2. Usage

2.1 Basic usage with input eztsp file

From the command line you have to provide a file of data in **eztsp format**, for example :

```
$ python tsp_visualizer.py -i data/france_32.eztsp
```

or

```
$ python tsp_visualizer.py --input=data/france_32.eztsp
```

2.2 Path

You can also specify a path to display, for example with the US file:

```
$ python tsp_visualizer.py --input=data/usa_13.eztsp -p "1
```

This will replace the path defined in the file “data/usa_13.eztsp”.

2.3 Resolution with LKH

To use LKH as a solver you have to provide the binary:

```
$ python tsp_visualizer.py --input=data/usa_13.eztsp -k /<p
```

or

```
$ python tsp_visualizer.py --input=data/usa_13.eztsp --lkh=
```

3. Commands of the interface

When you start the program you see the help menu. You can press Ctrl-C, F10 or Alt+F4 to quit the program.

- F1 for the help menu
- F2 to view path as a graph (graph mode)
- F3 to view path as a list of cities (cities mode)
- F4 for the about window
- F5 to change color model
- F6 to show / hide cities' names (in view mode F2)
- F7 to show / hide distances (in view mode F2)
- F8 to move the "Distance=" message
- F9 to print path on console and copy to clipboard (in view mode F2)
- F10, Ctrl-C to quit
- F11 to change the size of the window
- F12 to save the path view as an image (in graph mode F2)
- 'L' display list of cities
- 'I' Resolution of the problem based on an iterated local seach algorithm

4. EZ TSP File Format

The file that contains information about the cities and distances is called EZ TSP and has .eztsp extension. The comments are introduced by the hashtag ('#') symbol at the beginning of a line and the rest of the line is considered as a comment.

The file contains the following sections :

4.1 COUNT

The COUNT section is required and defines the number of cities:

```
COUNT [
    13
]
```

4.2 CITIES

The CITIES section is optional and helps define the names of the cities where there should be one city per line:

```
CITIES [
    New York, NY
    Los Angeles, CA
    Chicago, IL
```