

Anleitung

Stand: 22.3.2015

Inhaltsverzeichnis

| | |
|--------------------------------------------------|---|
| Programm Struktur..... | 2 |
| Code..... | 2 |
| Compilieren | 3 |
| Benutzung..... | 3 |
| Datenbank | 3 |
| Menus..... | 4 |
| Editiermenu | 4 |
| Navigationsmenue..... | 5 |
| Antworten zur Aufgabenstellung (Beispiele) | 6 |

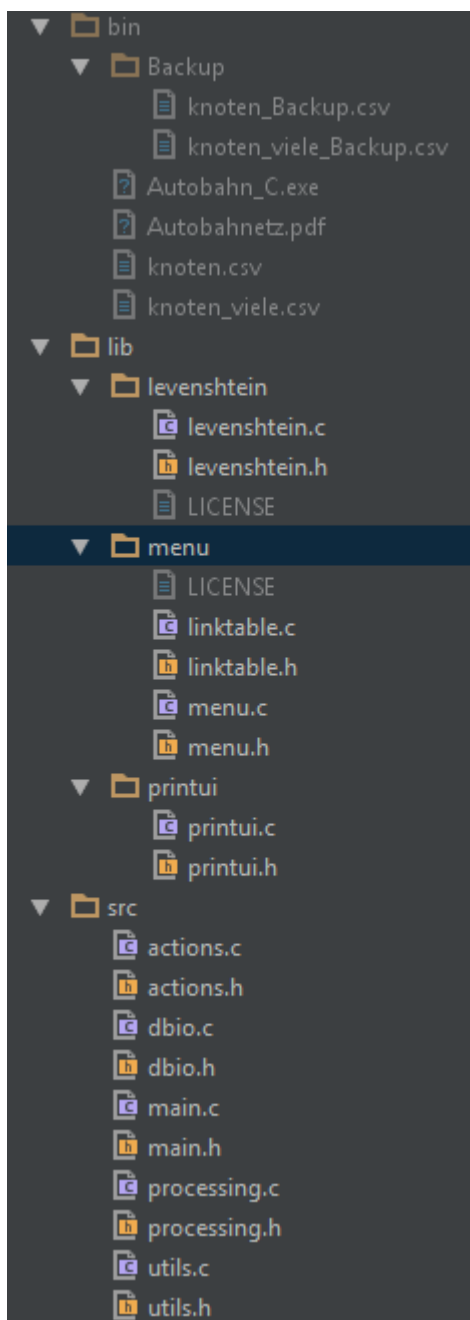
Programm Struktur

Code

Der Code ist in zwei Ordner aufgeteilt:

- „Src“ enthält den spezialisierten Quellcode auf das Projekt bezogen
- „Lib“ enthält allgemeinere Bibliotheken

Der gesamte Code ist selbstgeschrieben, mit Ausnahme von „lib/levenshtein“ und „lib/menu“ die Urheber sind in der LICENSE Datei in den Ordnern hinterlegt.



- „bin“ enthält die Executable, im gleichen Ordner wie die EXE-Datei muss auch die Datei „Knoten.csv“ liegen, diese wird dann geladen.
- Die Datei „Knoten.csv“ enthält den Standard-Datensatz, der durch die „Atobahn.pdf“ grafisch dargestellt wird.
- Die Datei „Knoten_viele.csv“ enthält 32.000 Datensätze und dient zum Testen von großen Datenmengen.
- Der Ordner „Backup“ existiert nur für den User falls er die Original-Datenbank wiederherstellen möchte.

- „levenshtein“ enthält den Damerau-Levenshtein Algorithmus

- „menu“ steuert die gesamte Eingabe (Tastatur) und stellt die Basis der Menuführung dar

- „printui“ – ist eine selbstgeschriebene Lib, die dafür sorgt, dass einfache Tabellen und „Fenster“ ausgegeben werden können.

- actions.c enthält die „backend“-Befehle von *new*, *edit*, *rm*, *ls*, *nav*
- dbio.c enthält alles was mit Datei Ein- und Ausgabe sowie Änderungen und Abfragen der Dateistruktur (Ram/Heap)
- main.c steuert den Ablauf, initialisiert die Menues und wertet Eingaben der Befehle aus (Parameter)
- processing.c enthält den Deijkstra-Algorithmus und führt die Berechnung und Ausgabe rund um das „PathFinding“ aus.
- utils.c sind Funktionen die man als „Helferlein“ braucht

Compilieren

Compile-Parameter :

- std=gnu99
- O3 funktioniert

Die Executable muss direkt im \bin Ordner landen!

Muss im selben Ordner wie „knoten.csv“ liegen.

Benutzung

Datenbank

Als „Datenbank“ wird immer die Datei „knoten.csv“ genommen, die im selben Ordner wie die ausführbare Datei (EXE) sein muss.

- Durch das Umbenennen von z.B. „knoten_viele.csv“ in „knoten.csv“ wird nun diese geladen (genauso jede andere Datenbank).
- Falls keine „knoten.csv“ existiert so wird das Programm ohne Daten geladen. Beim ordnungsgemäßen Exit (durch Eingabe „exit“ im Programm) kommt ein Dialog „möchten sie die Änderungen speichern“. Wenn man Änderungen hat und den Dialog mit „ja“ beantwortet, wird eine neue „knoten.csv“ Datei erstellt.

Menus

Unterschiedliche Menus werden benötigt, damit nicht bei jedem editieren die Daten neu verarbeitet werden müssen, sondern nur wenn man vom Edit- in den Navigationsmodus wechselt. Dies dient der eindeutigen Trennung diese zwei Funktionen.

Allgemein sind folgende Befehle in jedem Menu vorhanden.

- „help“ zeigt verfügbare Befehle für das derzeitige Menu an
- „clear“ leert die Konsole
- „exit“ beendet das Programm ordnungsgemäß und ermöglicht das Speichern von Änderungen

Editiermenu

Im Editiermenu können Daten angezeigt, hinzugefügt, gelöscht und verändert werden.

Befehle

- „ls [Ausfahrt/Kreuz/Autobahn]“ gibt Daten aus
 - Parameter „-a“ (ls -a) zeigt alle Autobahnen an, die es gibt.
 - Parameter „-t“ führt zur Ausgabe in Tabellenform
 - Parameter „-n“ funktioniert nur zusammen mit „-t“ und sortiert die Tabelle nach Namen
- „new [AusfahrtName] [AutobahnName] [AutobahnKm]“
 - erstellt eine Ausfahrt
 - erstellt auch neue Autobahn falls der [AutobahnName] noch nicht Existiert.
- „new [AusfahrtName] [AutobahnName1] [AutobahnKm1] [AutobahnName2] [AutobahnKm2]“
 - erstellt ein Autobahnkreuz
 - erstellt auch neue Autobahn falls ein [AutobahnName] noch nicht existiert.
- „edit [NameAlt] [NameNeu]“
 - benennt eine Ausfahrt/Kreuz/Autobahn um.
- „edit [AusfahrtName] [NeuerAutobahnName] [NeueAutobahnKm]“
 - bearbeitet eine Ausfahrt
- „edit [AusfahrtName] [AutobahnName1] [AutobahnKm1] [AutobahnName2] [AutobahnKm2]“
 - bearbeitet ein Autobahnkreuz
- „rm [Name]“
 - löscht Ausfahrt/Kreuz/Autobahn

Navigationsmenue

Im Navigationsmenü können Daten angezeigt und ein Weg von A nach B berechnet werden.

Befehle

- „ls [Ausfahrt/Kreuz/Autobahn]“
 - gibt Daten aus
 - Parameter „-a“ (ls -a) zeigt alle Autobahnen die es gibt
 - Parameter „-t“ führt zur Ausgabe in Tabellenform
 - Parameter „-n“ funktioniert nur zusammen mit „-t“ und sortiert die Tabelle nach Namen
- „nav [Von] [Nach]“
 - navigiert von Punkt A nach B

Antworten zur Aufgabenstellung (Beispiele)

- (i) Bearbeitung der Datei
 - a. „new Neckarwestheim A81 80.32“
 - i. Eine neue Ausfahrt wird erstellt, falls A81 noch nicht existiert wird diese auch erstellt
 - b. „new Weinsberg A81 99,32 A6 291.3“
 - i. Ein neues Autobahnkreuz (A81 <-> A6) wird erstellt, falls A81 oder A6 noch nicht existieren werden diese auch erstellt
 - c. „edit Mosbach MOS“
 - i. Die Ausfahrt Mosbach wird in MOS umbenannt
 - d. „edit A1 A11“
 - i. Die A1 ist eine Autobahn, A1 wird in nun in A11 umbenannt, alle Autobahnkreuze und Ausfahrten auf A1 liegen danach auf A11
 - e. „edit Mosbach A2 993.12“
 - i. Mosbach wird auf die A2 auf km 993.12 verlegt
 - f. „edit K1_2 A1 34 A3 33“
 - i. Das Autobahnkreuz K1_2 wird nun auf A1 auf km 34 und auf A3 Km 33 sein (A1 <-> A3)
 - g. „rm Mosbach“
 - i. Mosbach wird gelöscht
 - h. „rm A1“
 - i. Die komplette A1 wird gelöscht, mit allen Ausfahrten und Autobahnkreuzen. Die Autobahnkreuze verschwinden auch von der mit A1 verknüpften Autobahnen
 - i. Die Datei speichern falls Änderungen gemacht wurden:
 - i. „exit“ eingeben und dem Dialog folgen
 - ii. bei „Änderungen speichern“ die Frage mit „y“ beantwortet. Nur so werden Änderungen zurück in die Datei gespeichert
- (ii) Auflistung von Abfahrten
 - a. „ls Mosbach“
 - i. gibt die Autobahn an der Mosbach liegt aus und markiert Mosbach
 - b. „ls A1“
 - i. gibt die Autobahn A1 aus
 - c. „ls A1 -t“
 - i. gibt die Autobahn A1 in einer Tabelle aus
 - d. „ls A1 -t -n“
 - i. gibt die Autobahn A1 in einer Tabelle nach Namen sortiert aus
 - e. „ls -a“
 - i. gibt alle Autobahnen aus die es gibt
- (iii) Routenberechnung
 - a. „nav Heilbronn Stuttgart“
 - i. findet den kürzesten Weg von Heilbronn nach Stuttgart
 - b. „nav K1_2 Kiel“
 - i. findet den kürzesten Weg vom Kreuz K1_2 zu Ausfahrt Kiel