

Exposé

Gruppe: Julian Schumacher und Gregor Gottschewski

Gruppennummer: 8

Aufgabenstellung

Es soll ein Commandline-Routenplanungstool in C mit erweiterten Funktionalitäten umgesetzt werden. Als Algorithmus soll der Dijkstra- oder der optimierte A*-Algorithmus verwendet werden. Dieser berechnet den kürzesten Pfad von einem Start- (über Wegpunkte) zu einem Zielpunkt.

Ein- und Ausgabe

- **Eingabe:** Der Nutzer gibt über die Kommandozeile einen Startpunkt, Endpunkt und den Pfad zu einer Datei, die alle Wegpunkte beinhaltet, ein.
- **Ausgabe:** Nach der Berechnung wird der kürzeste Pfad auf der Kommandozeile ausgegeben (z.B. `(Start) Stuttgart->München->Nürnberg->Erfurt (Ende)`)

Nutzungsszenarien:

- *Basisszenario:* Der Anwender gibt den Start- und Zielort ein und das Tool berechnet den kürzesten Weg.
- *Wegpunkte:* Der Anwender gibt Startort, Zielort und Wegpunkte an, die zusätzlich erreicht werden sollen. Das Tool berechnet den Pfad unter Berücksichtigung der Wegpunkte.
- *Debug-Modus:* Bei aktiviertem Debug-Modus zeigt das Tool detaillierte Schritte der Berechnung an (z.B. welche Knoten untersucht wurden und welche Verbindungen gewählt wurden).
- *Fehlererkennung (optional):* Bei Hindernissen (gesperrte Straßen oder Wegpunkte) wird der Anwender darauf hingewiesen, und das Tool berechnet den optimalen Pfad unter Berücksichtigung dieser Einschränkungen.

Funktionsumfang

- **Wegpunkte laden:** Das Tool liest Wegpunkte aus einer Datei ein
 - Die Wegpunkte werden gespeichert mit einem Namen und Position
- **Pfadberechnung:** Der Algorithmus berechnet den schnellsten Pfad unter Verwendung des Dijkstra- oder (optional) optimierten A*-Algorithmus.
- **Ausgabe:** Der ermittelte Pfad wird sowohl in der Konsole angezeigt als auch optional in eine Datei geschrieben.
- **Hinderniserkennung (optional):** Gesperrte Straßen oder Wegpunkte werden in die Berechnung einbezogen, um unbrauchbare Routen zu vermeiden.
- **Kraftstoffverbrauch (optional):** Das Tool berechnet basierend auf dem durchschnittlichen Kraftstoffverbrauch eines PKWs den voraussichtlichen Kraftstoffbedarf für die geplante Strecke.

Angestrebte Lösung aus technischer Sicht

- Plattform: Die Anwendung wird für Unix-Plattformen mit einem kompatiblen `gcc`-Compiler (11.4) oder macOS (macOS Clang 16.0.0) entwickelt
- Bibliotheken:
 - Standardbibliotheken von C für die Grundfunktionen wie Dateizugriff und Konsolenausgabe.
 - Datenstrukturen: Nutzung von verketteten Listen oder Heaps für die effiziente Verwaltung der Knoten im Dijkstra- und A*-Algorithmus.
- Algorithmen:
 - Dijkstra-Algorithmus (o.g.)
 - A-Algorithmus* (optional): Eine optimierte Version des Dijkstra-Algorithmus mit einer Heuristik zur schnelleren Berechnung des Pfads
- Datenformat für Wegpunkte: Wegpunkte werden in einer Textdatei im CSV-Format gespeichert.

Beispiel

Eingabe:

```
./routenplaner --start Stuttgart --ziel Berlin --waypoints Hamburg
```

Ausgabe:

```
Berechnung des kürzesten Pfades...  
Start -> Stuttgart -> Nürnberg -> Erfurt -> München (Ziel)
```