G1-TravelingSalesMan

Autor: Rahil Chopra

Projektmitglieder: Sebastian Goebel, Peter Pallnstorfer, Rahil Chopra

Projektverlauf:

Aufgrund dessen, dass einige Termine zusammenfielen (diverse Prüfungstermine und Abgaben) war es schwer für uns regelmäßig an unserem Projekt zu arbeiten. So hatten wir kurz vor der Deadline der Abgabe enormen Stress, konnten aber trotzdem noch gerade rechtzeitig alle geplanten Funktionen implementieren.

Zu Beginn haben wir uns damit beschäftigt GitHub zu verstehen und wie man es verwenden kann, um die Mitarbeit und die Kommunikation zwischen den Teammitgliedern zu ermöglichen. Weiters haben wir Git verwendet, um die Versionsverwaltung von Dateien zu ermöglichen, da dies der professionellste weg ist, ein Projekt zu realisieren. Zusätzlich haben wir verschiedenste Kommunikations-Plattformen, wie ,,WhatsApp'' und ,,Discord'', verwendet, um ein besonders transparentes Arbeiten zu ermöglichen und uns sofort mit aufkommenden Fehlern helfen konnten. Als nächstes haben wir uns angeschaut wie CMake funktioniert. Um das Projekt am best möglichsten realisieren zu können haben wir uns auch verschiedenste Codeeditor angeschaut und sind zu dem Entschluss gekommen, dass ,,Clion'' von ,,Jet Brains'' die beste Wahl ist. Dieser Editor ist ein intelligenter Editor für C-Programmierung und bietet nicht nur eine geeignete Darstellung der Ordnerstruktur, sondern auch einen integrierten terminal, mit dem die CMake-Commands ganz einfach in ,,Clion'' einzugeben waren. Als wir nun passende Tools hatten, um an unserem Projekt zu arbeiten, haben wir als erstes versucht unsere CSV-Datei einzulesen. Dabei kam es bereits zu den ersten Schwierigkeiten. Die teilweise unregelmäßige Formatierung der Datei hat uns beim Abspeichern in ein Array viel Zeit gekostet. Also haben wir zuerst eine eigene, kleinere CSV-Datei erstellt mit weniger Einträgen, um besser am Rest des Codes arbeiten zu können. Später sind wir darauf gekommen, dass sinnvoller ist ein dynamisches Array für unsere Daten anzulegen und worauf wir beim Einlesen der Daten achten müssen, damit dies ohne weitere Probleme funktioniert.

Um schlussendlich alle geplanten Funktionen zu implementieren haben wir uns diese aufgeteilt.

Rollenbeschreibungen:

- *Sebastian GOEBEL:*

  Der Kollege Goebel hat sich zu Beginn GitHub und CMake angeschaut und es uns anschließend beigebracht. Weiters hat er sich um das Einlesen aus der CSV-Datei in das Array gekümmert und die Funktion geschrieben, die die Größe und Zahl unserer Datensätze ausgibt.

- *Peter PALLNSTORFER:*

  Der Kollege Pallnstorfer hat sich um die ganzen Sortier-Algorithmen gekümmert, um die ganze CSV-Datei sortiert auszugeben bzw. nach einer bestimmten Stadt zu suchen.

- *Rahil CHOPRA:*

  Ich habe mich um die main() gekümmert und ein benutzerfreundliches Menu geschrieben, um die einzelnen Funktionen aufrufen zu können. Weiters habe ich dem Kollegen Goebel beim Einlesen der Daten geholfen, da wir, wie bereits erwähnt, Probleme haten ca. 15.000 Datensätze einzulesen.

Um das Hauptproblem: das Travel-Sales-Man-Problem, haben wir unser gemeinsam gekümmert, dass dies die herausforderndste Funktion des gesamten Projekts war.

Beschreibung des Programmablaufes:

Zu Beginn sieht der Benutzer den Namen des Programms und den unserer Gruppe. Es gibt ein Feedback wie viele Datensätze eingelesen wurden. Dann stehen vier Funktionen zur Auswahl: 1--Sortieren, 2--Suchen, 3--Travelling-Salesman, 0--Beenden.

1—Sortieren:

Wählt der Benutzer den Menü-Punkt 1 so steht ihm ein weiteres Menü zu Verfügung, wobei ausgesucht werden kann wonach sortiert werden soll. Wählt er einen davon so bekommt er eine sortierte Liste zurück.

2—Suchen:

Wählt der Benutzer Menü-Punkt 2 so wird er gefragt nach welcher Stadt er suchen möchte. Wenn er eine Stadt eingibt bekommt er die wichtigsten Eckdaten ausgegeben.

3—Travelling-Salesman:

Wählt Benutzer Menü-Punkt 3 so wird er gefragt wie viele Städte er einlesen möchte. Anschließend kann er die jeweiligen Städte eingeben und eine CSV Datei erstellen, in der die Daten gespeichert werden. Danach bekommt er die kurzmöglichste Strecke ausgegeben.

0—Beenden:

Mit 0 beendet der Benutzer das Programm.

Aufbau des Projektsordners:

Main.c →Project\common\Demo

Headers.h→Project\common\Headers

Functions.c→ Project\common\Sources

G1-SalesManProblem.exe →Project\build\bin