

Übungsblatt 9

Abgabe bis Dienstag, den **5. Juli** um **12:00** Uhr

Die Aufgabe auf diesem Übungsblatt ist etwas freier gestaltet, als die Aufgaben auf den vorherigen Übungsblättern. Sie sollen sich natürlich an alles halten, was Sie bisher gelernt haben, dürfen aber jetzt alles benutzen, was die STL zu bieten hat. Achten Sie aber im Sinne der letzten Vorlesung darauf, die STL *sinnvoll* zu benutzen und so, dass Ihr Code möglichst *effizient* ist.

Aufgabe 1 (15 Punkte)

Implementieren Sie eine Klasse *HeatMap* mit den folgenden Methoden. Benutzen Sie dabei die auf dem Wiki und im SVN zur Verfügung gestellte Klasse *TerminalManager*, die eine leicht erweiterte Version der entsprechenden Klasse vom 5. Übungsblatt ist.

readPointsFromFile(const std::string& filename): Diese Methode soll eine Datei mit Koordinaten (Längengrad, Breitengrad) einlesen. Das genaue Format versteht man leicht, wenn man sich die *osm-...* Dateien auf dem Wiki anschaut, siehe Aufgabe 2. Extrahieren Sie mittels geeigneter Funktionen die Werte der Koordinaten jeweils als *float* und speichern Sie sie in einer geeigneten Membervariablen. Sie können dabei ein einzelnes Koordinatenpaar als *std::pair<float, float>* speichern oder in einer eigenen Klasse *Point*. Schreiben Sie für diese Methode einen einfachen Test mit einer geeigneten Testdatei (mit wenigen Punkten), die Sie mit in das SVN committen dürfen und sollen (solche Dateien sind integraler Bestandteil des Tests).

computeHeatMap(): Diese Methode soll jedes Koordinatenpaar durch eine einfache lineare Transformation auf einen Pixel des Bildschirms abbilden, wobei Pixel im Sinne der Klasse *TerminalManager* gemeint sind. Dabei sollte entweder die südlichste Koordinate auf die unterste Zeile und die nördlichste Koordinate auf die oberste Zeile *oder* die westlichste Koordinate auf die linkeste Spalte und die östlichste Koordinate auf die rechteste Spalte abgebildet werden. Denken Sie an ein Bild, das auf einem Bildschirm abgebildet werden soll und dessen Dimensionen nicht zu den Dimensionen des Bildschirms passen. Dann muss man entweder links und rechts einen Rand lassen oder oben und unten, aber nicht beides. Auch für diese Methode sollen Sie einen Test schreiben.

Versuchen Sie optional, das Bild zu zentrieren und das Bild so zu stauchen oder zu strecken, dass die Darstellung in Aufgabe 2 natürlich aussieht. Das ist keine Anforderung, um die volle Punktzahl zu bekommen. Es sieht aber deutlich schöner aus, wenn man es macht.

```
[#include <algorithm>; std::please_turn_over();]
```

Das Endergebnis der Methode sollte sein, dass in einer oder mehreren dazu geeigneten Membervariablen die Information steht, wie viele Punkte auf welchen Pixel abgebildet werden, siehe nächste Funktion.

drawHeatMap(TerminalManager)*: Malen Sie jeden Pixel mit einer Farbintensität, die der Anzahl Punkte entspricht, die in der vorherigen Funktion auf diesen Pixel abgebildet wurden. Der höchste Wert soll dabei mit der maximalen Intensität gemalt werden und der niedrigste Wert mit der minimalen Intensität. Beachten Sie, dass die zur Verfügung gestellte Klasse *TerminalManager* sowohl für die Methode *drawPixel* als auch für die Methode *drawString* ein Argument für eben diese Intensität hat. Die beiden Methoden erwarten dort einen *float* zwischen 0.0 (minimale Intensität) und 1.0 (maximale Intensität). Für die Methode *drawHeatMap* brauchen Sie keinen Test zu schreiben.

Aufgabe 2 (5 Punkte)

Schreiben Sie ein Programm *HeatMapMain*, das eine Datei von Punkten mit dem über Kommandozeilenargument gegebenen Namen einliest und diese Punkte als Heatmap gemäß Aufgabe 1 darstellt. Das Programm sollte nach dem Zeichnen der Heatmap so lange warten, bis die Benutzerin *q* drückt (auch dazu stellt die Klasse *TerminalManager* Funktionalität zur Verfügung).

Führen Sie Ihr Programm auf den auf dem Wiki verlinkten Dateien aus. Die Dateien enthalten die Koordinaten sämtlicher Punkte aus *OpenStreetMap* (*OSM*), die einen Namen haben. Es gibt je eine Datei für Deutschland, die Schweiz und China. Vergewissern Sie sich, dass die Karte so aussieht, wie Sie es erwarten würden und falls nicht, passen Sie Ihren Code aus Aufgabe 1 an oder Ihre Erwartungshaltung.

Die Dateien auf dem Wiki enthalten für jeden Punkt neben den beiden Koordinaten auch noch den Namen aus OSM. Sie brauchen die Namen für das Übungsblatt nicht. Aber wenn Ihnen irgendetwas einfällt, wie Sie damit Ihr Programm erweitern können, sehr gerne! Zum Beispiel könnten Sie nach jedem Tastendruck nur noch die Heatmap der Punkte anzeigen, deren Namen zu der zuletzt eingegebenen Buchstabenfolge passen (und nach ein paar Sekunden ohne Eingabe wieder zur vollen Ansicht zurückkehren).

Laden Sie wie gehabt alle Code-Dateien und das Makefile in unser SVN hoch, in einem neuen Unterverzeichnis *blatt-09*. Es gelten weiterhin die 10 Gebote und nehmen Sie dir Ratschläge Ihres Tutors ernst.

Laden Sie wie gehabt auch eine Datei *erfahrungen.txt* hoch (im Unterordner *blatt-09*), in der Sie kurz Ihre Erfahrungen mit dem Ü9 und der Vorlesung dazu beschreiben.

Wie viele Objekte gibt es auf OpenStreetMap und wer hat sie dort eingetragen?