

Modularbeit Geoinformatik

Lehrbrief 8

1. Vorgehensweise und Visualisierung des Punkt-Im-Polygon-Test

Vorgehensweise:

Ich habe mich in bei dieser Modularbeit dazu entschieden ein kleines Programm mit grafischer Oberfläche zu programmieren. Es sollte möglich sein, Koordinaten für einen Punkt und auch für das Polygon anzugeben. Damit man nichts extra herunter laden muss habe ich mich für die Python-Bibliothek „Tkinter“ entschieden. Diese kann einfach in den Code eingebunden und benutzt werden.

Für mein Programm habe ich eine Klasse geschrieben, die sich um die gesamte GUI kümmert. Dabei werden das Fenster, Buttons und Texte erstellt und in einem Grid angezeigt.

In die ersten beiden Input-Felder kann man Koordinaten eines Polygonpunktes eintragen. Mit einem Klick auf den Button drunter werden die Koordinaten in dem Polygonpunkt gespeichert und angezeigt. Die Input-Felder können danach weiter benutzt werden um die restlichen Polygonpunkte einzutragen.

Die anderen beiden Inputfelder sind für den Punkt, den man überprüfen möchte.

Hat man alle Polygonpunkte erzeugt und auch die Koordinaten des zu überprüfenden Punktes eingetragen kann man mit dem „Berechne“ Button die Überprüfung durchführen.

Das Ergebnis wird oben in grün(wenn der Punkt im Polygon liegt) oder in rot(wenn der Punkt nicht im Polygon liegt) angezeigt.

Mit dem upload-Button kann eine Liste von Polygonpunkten hochgeladen werden. Die einzelnen Punkte müssen jeweils in einer eigenen Zeile stehen. Als erstes wird der Punktname angegeben, dann die x und als letztes die y Koordinate. Die einzelnen Daten des Punktes werden mit Kommas separiert. Ein Strichpunkt kommt ans Ende der Zeile(bis auf die letzte Zeile). Das Koordinaten-File muss „punkte.txt“ heißen. Das File mit meinen Testdaten liegt mit in dem Odner „PolygonTest“.

Mit Reset können die Punkte gelöscht werden.

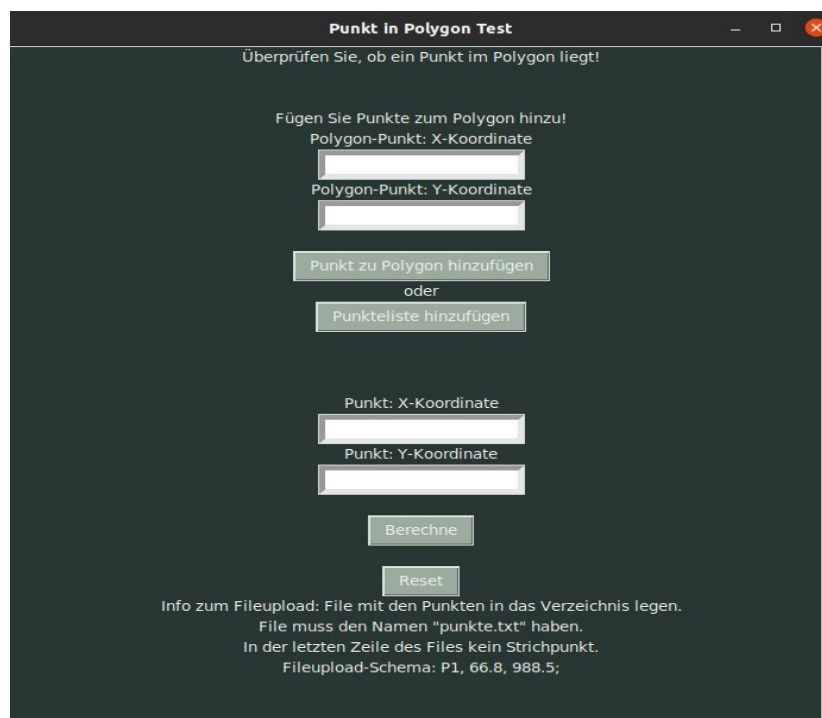


Abb.1: GUI: Punkt-im-Polygon-Test

Visualisierung eines realen Beispiels:

Mein Untersuchungsgebiet liegt in Mittenwald. Für das Punkt-in-Polygon-Test Programm habe ich mich für einen See ganz in der Nähe entschieden, den Ferchensee. Das Programm soll überprüfen, ob der Punkt (grob) innerhalb des Sees oder außerhalb liegt.

Für mein Beispiel habe ich eine Koordinatenliste erzeugt, die im Ordner „PolygonTest“ liegt. Diese punkte.txt beinhaltet alle 19 Polygonpunkte.

Die beiden Punkte, die getestet werden sollen sind (Mit UTM Koordinaten)

T1: 666873, 5256299 (innerhalb des Sees)

T2: 666944, 5256406 (außerhalb)

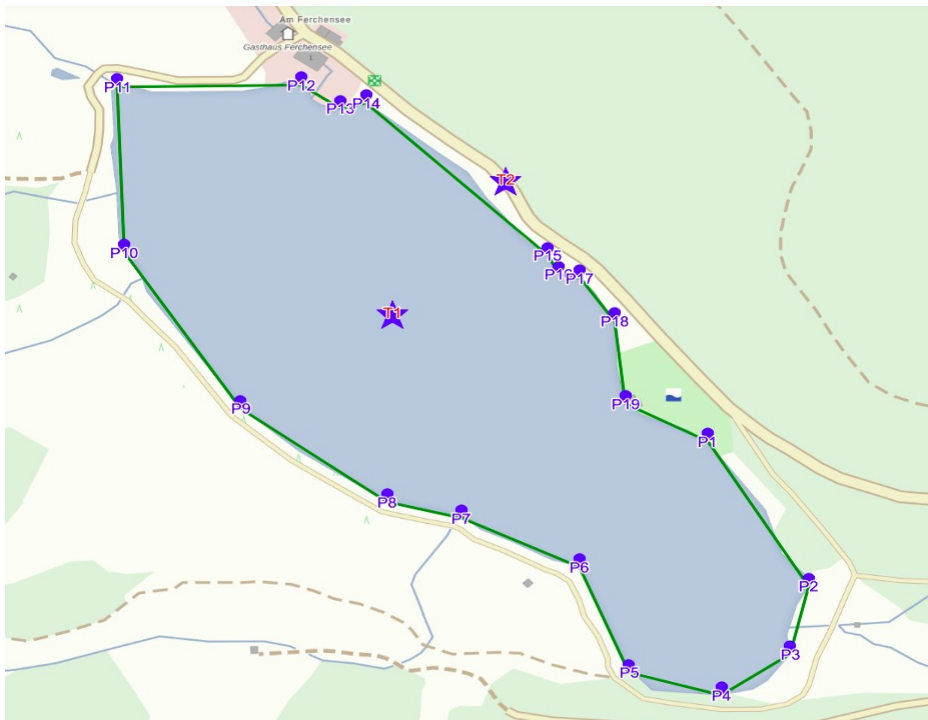


Abb. 2: Polygon um den See

Das Gebiet kann über folgenden Link abgerufen werden:

https://geoportal.bayern.de/bayernatlas/?lang=de&topic=ba&catalogNodes=11&bgLayer=atkis&E=666967.48&N=5256227.09&zoom=11.663266088078505&layers=KML%7C%7Chttps:%2F%2Fgeoportal.bayern.de%2Fba-backend%2Ffiles%2Ff_38716cc0-5f2c-11ec-8fd2-071ea897c93d_7f5c70be-198d-4611-bd88-a18897a1b0c8%7C%7Ctrue

Test der beiden Punkte:

Als erstes lade ich die Polygonpunkte hoch, indem ich den Button anklicke. Dann gebe ich den Punkt T1 ein und klicke auf „Berechne“. Das Ergebnis ist wie erwartet: Der Punkt liegt in dem See. (siehe Abbildung 3)

Dann teste ich den zweiten Punkt, T2. Auch hier kommt das erwartete Ergebnis raus: Der Punkt liegt nicht in dem See. (siehe Abbildung 4)

Punkt in Polygon Test

Überprüfen Sie, ob ein Punkt im Polygon liegt!

Der Punkt liegt in dem Polygon

Fügen Sie Punkte zum Polygon hinzu!

Polygon-Punkt: X-Koordinate

Polygon-Punkt: Y-Koordinate

oder

[667086.0,5256207.0][667158.0,5256097.0][667147.0,5256043.0][667103.0,5256009.0][667041.0,5256025.0][667005.0,5256106.0][666925.0,5256141.0][666875.0,5256151.0][666775.0,5256220.0][666693.0,5256339.0][666683.0,5256467.0][666806.0,5256474.0][666832.0,5256456.0][666849.0,5256461.0][666975.0,5256347.0][666982.0,5256333.0][666997.0,5256330.0][667021.0,5256298.0][667030.0,5256234.0]

Punkt: X-Koordinate

Punkt: Y-Koordinate

Info zum Fileupload: File mit den Punkten in das Verzeichnis legen.
File muss den Namen "punkte.txt" haben.
In der letzten Zeile des Files kein Strichpunkt.
Fileupload-Schema: P1, 66.8, 988.5;

Abb.3: Ergebnis Abfrage T1

Punkt in Polygon Test

Überprüfen Sie, ob ein Punkt im Polygon liegt!

Der Punkt liegt nicht in dem Polygon

Fügen Sie Punkte zum Polygon hinzu!

Polygon-Punkt: X-Koordinate

Polygon-Punkt: Y-Koordinate

oder

[667086.0,5256207.0][667158.0,5256097.0][667147.0,5256043.0][667103.0,5256009.0][667041.0,5256025.0][667005.0,5256106.0][666925.0,5256141.0][666875.0,5256151.0][666775.0,5256220.0][666693.0,5256339.0][666683.0,5256467.0][666806.0,5256474.0][666832.0,5256456.0][666849.0,5256461.0][666975.0,5256347.0][666982.0,5256333.0][666997.0,5256330.0][667021.0,5256298.0][667030.0,5256234.0]

Punkt: X-Koordinate

Punkt: Y-Koordinate

Info zum Fileupload: File mit den Punkten in das Verzeichnis legen.
File muss den Namen "punkte.txt" haben.
In der letzten Zeile des Files kein Strichpunkt.
Fileupload-Schema: P1, 66.8, 988.5;

Abb.4: Ergebnis Abfrage T2

2. Vorgehensweise und Visualisierung des Dijkstra Algorithmus

Vorgehensweise:

Ähnlich wie bei dem Punkt-in-Polygon-Test habe ich für den Dijkstra-Algorithmus auch ein kleines Programm geschrieben.

Zunächst habe ich das Bild des Gebiets eingefügt, damit der Benutzer die Punkte und Strecken sehen kann. Die Knoten und Kanten habe ich entsprechend meines Gebiets abgeändert. Dabei habe ich die Strecken auf ganze Meter gerundet.

Die Koordinaten der einzelnen Punkte sind ebenfalls sichtbar in dem Programm.

Um eine Strecke zu berechnen kann der Benutzer zwei Punkte in die beiden Eingabefelder eingeben. Diese müssen P1-P14 sein. Falls der angegebene Punkt nicht in den Knoten ist wird dies angezeigt.

Gibt der Benutzer zwei existierende Punkte ein und klickt auf „Berechne“, wird der kürzeste Weg berechnet und angezeigt.

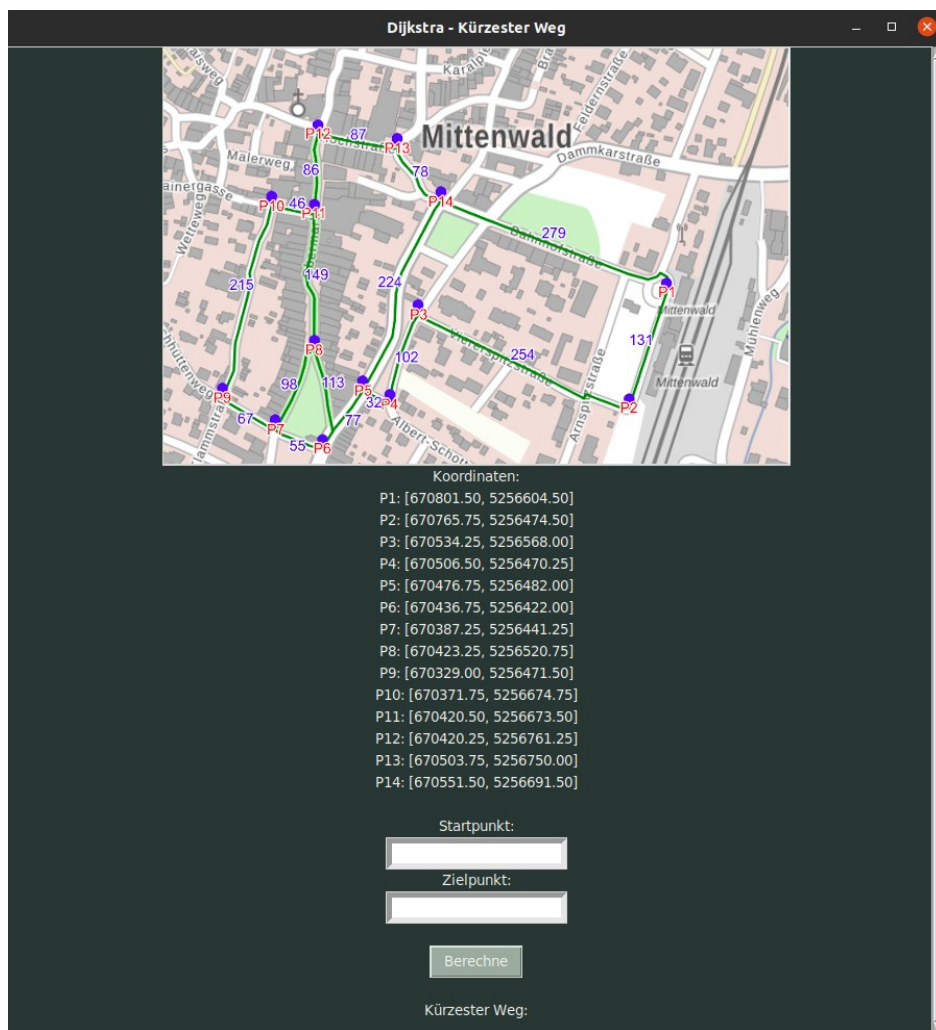


Abb. 5: GUI: Dijkstra – Kürzester Weg

Marina Rug | 02174416

Visualisierung des Gebiets:

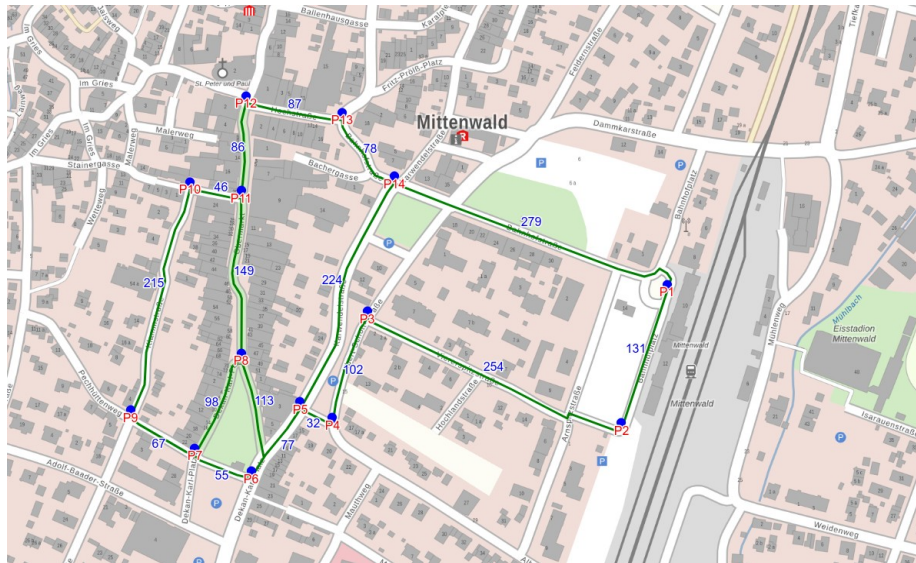


Abb. 6: Knoten und Kanten in Mittenwald

Das Gebiet kann über folgenden Link abgerufen werden:

https://geoportal.bayern.de/bayernatlas/?lang=de&topic=ba&catalogNodes=11&bgLayer=atkis&E=670581.43&N=5256596.17&zoom=12.552938511776194&layers=KML%7C%7Chttps:%2F%2Fgeoportal.bayern.de%2Fba-backend%2Ffiles%2Ff_38716cc0-5f2c-11ec-8fd2-071ea897c93d_7f5c70be-198d-4611-bd88-a18897a1b0c8

Test:

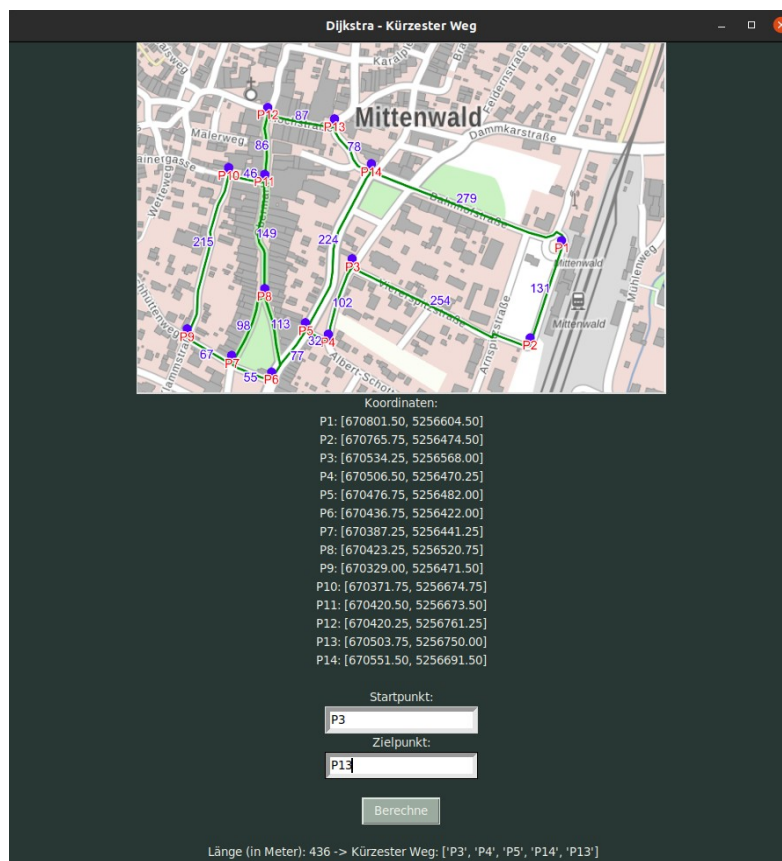


Abb. 7: Testergebnis