



Wydział Geodezji i Kartografii

POLITECHNIKA WARSZAWSKA

TRANSFORMACJE MIĘDZY RÓŻNYMI UKŁADAMI GEODEZYJNYMI

INFORMATYKA GEODEZYJNA
SEM. IV, ĆWICZENIA, ROK AKAD. 2022-2023

ADRIAN MAKSYMIOUK
GRUPA II, NUMERY INDEKSU: 319345
DAWID JUNDO
GRUPA II, NUMERY INDEKSU: 319328
01169882@pw.edu.pl lub 01169863@pw.edu.pl

WYDZIAŁ GEODEZJI I KARTOGRAFII
POLITECHNIKA WARSZAWSKA
Zakład Geodezji Wyższej i Astronomii
Warszawa, 11 czerwca 2023

Spis treści

| | | |
|----------|---|----------|
| 1 | Cel ćwiczenia | 2 |
| 2 | Wykorzystane narzędzia i materiały potrzebne do replikacji ćwiczenia | 2 |
| 2.1 | Wybrany język programowania i interpreter Spyder | 2 |
| 2.2 | System operacyjny | 2 |
| 2.3 | Potrzebne biblioteki i pliki | 2 |
| 3 | Przebieg ćwiczenia | 3 |
| 3.1 | Stworzenie wtyczki | 3 |
| 3.2 | Stworzenie - zliczanie obiektów | 3 |
| 3.3 | Stworzenie - Różnica wysokości | 3 |
| 3.4 | Stworzenie - Azymut | 3 |
| 3.5 | Stworzenie - Długość odcinka | 3 |
| 3.6 | Stworzenie - Wyświetlanie się współrzędnych | 3 |
| 3.7 | Stworzenie - Pole powierzchni | 4 |
| 3.8 | Stworzenie - Reset QLabel | 4 |
| 3.9 | Stworzenie - Wczytanie pliku | 4 |
| 3.10 | Stworzenie - Zapisanie pliku | 4 |
| 4 | Podsumowanie | 4 |
| 4.1 | Rezultaty | 4 |
| 4.2 | Umiejętności nabyte | 4 |
| 4.3 | Spostrzerzenia, problemy i ich rozwiązania: | 4 |

1 Cel ćwiczenia

Celem ćwiczenia jest stworzenie wtyczki do QGIS-a, której zadaniem jest wykonywanie obliczeń na współrzędnych punktów. Wtyczka powinna umożliwiać wykonanie takich operacji jak: zliczanie punktów, liczenie różnicy wysokości, liczenie azymutów, liczenie odległości, liczenie pola powierzchni oraz tworzenie poligonów.

■ Funkcje wtyczki w których potrzeba dwóch punktów

- Liczenie różnic wysokości
- Liczenie azymutów
- Liczenie odległości

■ Funkcje wtyczki w których potrzeba minimum trzech punktów

- Liczenie pól powierzchni
- Tworzenie poligonu

■ Funkcje wtyczki w których nie potrzeba punktów

- Zliczenie punktów
- Wyświetlanie współrzędnych

2 Wykorzystane narzędzia i materiały potrzebne do replikacji ćwiczenia

2.1 Wybrany język programowania i interpreter Spyder

- Python - język programowania, w którym napisany jest skrypt ćwiczenia.
- Spyder - jest to środowisko programistyczne dla języka Python, które zawiera edytor kodu, interpreter, konsolę i wiele innych funkcjonalności.
- Najlepiej pobrać Spydera za pośrednictwem anacondy, która ma domyślnie zainstalowane środowisko programistyczne Spyder www.anaconda.com/download (Wang i Oliphant, 2012)
- Qgis - program do który jest napisana nasza wtyczka. Zapośrednictwem tego programu zliczamy punkty itp.

2.2 System operacyjny

Ten skrypt został napisany w systemie operacyjnym Microsoft (Windows 10 oraz Windows 11).

2.3 Potrzebne biblioteki i pliki

Do wykonania ćwiczenia należy użyć następujących bibliotek:

1. os - to biblioteka standardowa w języku python, która zapewnia interfejs do operacji na systemie operacyjnym (np. dostęp do plików, zarządzanie procesami, zmiana katalogu roboczego)
2. PyQt5 - PyQt5 to zestaw narzędzi do tworzenia aplikacji graficznych w języku Python przy użyciu biblioteki Qt. Qt jest popularnym frameworkiem do tworzenia interfejsów użytkownika. Nie jest on wbudowany w pythona dlatego trzeba go zainstalować.
3. qgis.PyQt - moduł w bibliotece QGIS Python, który jest wykorzystywany do tworzenia interfejsów użytkownika w QGIS przy użyciu PyQt. Nie jest to samodzielna biblioteka PyQt5. Działa w połączeniu z biblioteką QGIS.
4. qgis.core - to moduł w bibliotece QGIS Python, który dostarcza podstawowe funkcje i klasy do pracy z danymi przestrzennymi w QGIS. Nie jest to samodzielna biblioteka PyQt5. Działa w połączeniu z biblioteką QGIS.

5. PyQt5.QtCore - to moduł w bibliotece PyQt5, który zawiera podstawowe klasy i funkcje rdzenne dla PyQt5. Moduł ten dostarcza elementy i funkcjonalności, które są częścią modułu QtCore w oryginalnej bibliotece Qt.
6. math - to biblioteka pythona, dzięki której można wykonywać podstawowe operacje matematyczne.

Należy również pobrać plik o nazwie "ttt.qgz", który znajduje się na zdalnym repozytorium GitHub pod linkiem: https://github.com/Sawoboh/Informatyka_Projekt_2.git da on możliwość testowania wtyczki w qgis na podanych punktach. Pliki "Przykładowe_wspolrzedne.txt" oraz "Przykładowe_wspolrzedne2.txt" zawierają przykładowe pliki tekstowe które można wczytać do wtyczki. Plikiy jest to zapis warstwy, która znajduje się w pliku "ttt.qgz".

3 Przebieg ćwiczenia

3.1 Stworzenie wtyczki

Za pomocą programu QGIS stworzono wtyczkę (nazwa itp.). Następnie została ona w ramach testu zmodyfikowana w Qt Designer. Kod został zamieniony na kod z rozszerzeniem ".py".

3.2 Stworzenie - zliczanie obiektów

Za pomocą Qt Designer dodano przycisk (QPushButton) o nazwie "Zlicz punkty". Następnie stworzono w Pythonie funkcję o nazwie "licz_elementy", która zlicza ilość punktów i pokazuje ją w QLabel o nazwie "pokaz_ilosc_punktow". Dodatkowo, w Pythonie ustawiono, że jeśli użytkownik kliknie przycisk, to wynik pojawi się w QLabel.

3.3 Stworzenie - Różnica wysokości

Za pomocą Qt Designer dodano przycisk (QPushButton) o nazwie "Różnica wysokości". Następnie stworzono w Pythonie funkcję o nazwie "roznica_wysokosci_funkcja", która liczy różnicę wysokości pomiędzy dwoma punktami i pokazuje ją w QLabel o nazwie "roznica_wysokosci_wynik". Dodatkowo, w Pythonie ustawiono, że jeśli użytkownik poda za dużą ilość punktów lub za małą, to wyświetli się napis "błąd" w QLabel oraz okienko z informacją jakiego rodzaju błąd wystąpił.

3.4 Stworzenie - Azymut

Za pomocą Qt Designer dodano przycisk (QPushButton) o nazwie "Azymut". Następnie stworzono w Pythonie funkcję o nazwie "azymut_funkcja", która liczy azymut pomiędzy dwoma punktami i pokazuje go w QLabel o nazwie "azymut_wynik". Funkcja ta liczy również azymut odwrotny, ponieważ użytkownik nie ma wpływu na wybór, który punkt jest pierwszy (może to sprawdzić za pomocą wyświetlania współrzędnych, gdzie jest podane id punktu). Dodano również QComboBox, w którym można wybrać jednostki, w jakich będzie wyświetlany wynik w QLabel. Pythonie ustawiono, że jeśli użytkownik poda za dużą ilość punktów lub za małą, to wyświetli się napis "błąd" w QLabel oraz okienko z informacją jakiego rodzaju błąd wystąpił.

3.5 Stworzenie - Długość odcinka

Za pomocą Qt Designer dodano przycisk (QPushButton) o nazwie "Długość odcinka". Następnie stworzono w Pythonie funkcję o nazwie "dlugosc_odcinka_funkcja", która liczy odległość pomiędzy dwoma punktami i pokazuje ją w QLabel o nazwie "dlugosc_odcinka_wynik". Dodatkowo, w Pythonie ustawiono, że jeśli użytkownik poda za dużą ilość punktów lub za małą, to wyświetli się napis "błąd" w QLabel oraz okienko z informacją jakiego rodzaju błąd wystąpił.

3.6 Stworzenie - Wyświetlanie się współrzędnych

Za pomocą Qt Designer dodano przycisk (QPushButton) o nazwie "Wyświetl współrzędne". Następnie stworzono w Pythonie funkcję o nazwie "wspolrzedne_funkcja", która liczy współrzędne X i Y punktów i wyświetla je w QTextEdit o nazwie "wspolrzedne".

3.7 Stworzenie - Pole powierzchni

Za pomocą Qt Designer dodano przycisk (QPushButton) o nazwie "Pole powierzchni". Następnie stworzono w Pythonie funkcję o nazwie "pole_powierzchni_funkcja", która liczy pole powierzchni. Dodatkowo dodano QComboBox, w którym można wybrać jednostki wynikowe. W Pythonie ustawiono również, że jeśli użytkownik poda za małą ilość punktów, to wyświetli się napis "błąd" w QLabel oraz okienko z informacją jakiego rodzaju błąd wystąpił.

3.8 Stworzenie - Reset QLabel

Za pomocą Qt Designer dodano przyciski (QPushButton) o nazwie "Zresetuj wszystkie pola", "Wyczyść tablicę" oraz przycisk zamknięcia wtyczki. Następnie stworzono w Pythonie funkcje o nazwie "wyczyszczenie_tablicy_funkcja" oraz "wyczyszczenie_danych_funkcja", które czyszczą QLabel.

3.9 Stworzenie - Wczytanie pliku

Za pomocą Qt Designer dodano QgsFileWidget który umożliwia nam wskazanie pliku txt, który chcemy wczytać. Natępnie są na nim prowadzone operacje i w atrybutach zapisywane są XYh punktu.

3.10 Stworzenie - Zapisanie pliku

Za pomocą Qt Designer dodano QPushButton o nazwie zapisz plik wynikowy. Wtedy program wykonuje szereg operacji dzięki którym w pliku zapiszą się operacje wtyczki na wybranych punktach. Plik wynikowy zapisuje się na pulpicie i zawsze pod tą samą nazwą "Plik_wynikowy_wtyczki_AM_DJ.txt". Jeśli użytkownik chce mieć kilka plików powinien ręcznie zmieniać nazwy.

4 Podsumowanie

4.1 Rezultaty

Link do zdalnego repozytorium GitHub: https://github.com/Sawoboh/Informatyka_Projekt_2.git w gałęzi master

Znajduje się na nim pliki o nazwie:

- folder wtyczka_projekt_3 - folder z naszą wtyczką stworzoną w tym zadaniu.
- folder nasz_latex - folder ze sprawozdaniem
- ttt.qgz - plik w qgis na której można testować wtyczkę

4.2 Umiejętności nabyte

- Sprawne pisanie plików tekstowych w latex w celu nauki skorzystaliśmy z książki (Borkowski i Przybylski, 2015)
- Umiejętność tworzenia wtyczek w Qgis
- Tworzenie wtyczek w Qt Designer
- Praca zespołowa z wykorzystaniem platformy Github
- Poprawienie jakości i przyspieszenie pisania kodu w Pythonie

4.3 Spostrzerzenia, problemy i ich rozwiązania:

Spostrzerzenia:

- Wraz ze wzrostem ilości czasu poświęconego na projekt, zauważano coraz więcej luk w funkcjach, które usprawniono.
- Cały czas program nie jest kompletny w 100%. Zawsze się znajdzie nowy pomysł który można zaimplementować.
- Projekt dostarczył nam dużo wiedzy która przyda nam się w przyszłości.

| Problem | Rozwizanie |
|--|---|
| Brak uporządkowanego wybierania punktów przez Qgis (dziwne poligon i pola powierzchni) | Wykorzystanie algorytmu do uporządkowanie położenia punktów zgodnie z ruchem wskazówek zegara |
| Brak umiejętności korzystania z Radio Button | Wykorzystanie QComboBox |
| Współrzędne nie mają podanej wysokości | Stworzenie własnej warstwy za pomocą wczytania pliku z XYh |
| Zapis pliku do folderu z qgis | Postanowiono ze plik będzie zawsze zapisywał się na pulpicie użytkownika |

Literatura

Borkowski, M., i Przybylski, B. (2015). *Książka kucharska latex*. Springer Wien New York.
Wang, P., i Oliphant, T. (2012). *Anaconda*. <https://www.anaconda.com/download>.