



Wydział Geodezji i Kartografii

POLITECHNIKA WARSZAWSKA

INFORMATYKA GEODEZYJNA
SEM. IV, PROJEKT, ROK AKAD. 2022/2023

WTYCZKA QGIS

WIKTORIA SAMEK 319371

PATRYCJA PIETRYKOWSKA 319356

ADRIANNA ZAWIŚLAK 319406

WYDZIAŁ GEODEZJI I KARTOGRAFII, POLITECHNIKA WARSZAWSKA
Warszawa, 12 czerwca 2023

Spis treści

| | | |
|----------|--|----------|
| 1 | Cel Cwiczenia | 2 |
| 2 | Wykorzystane narzędzia i materiały: | 2 |
| 3 | Wstęp teoretyczny | 2 |
| 4 | Przebieg Ćwiczenia | 3 |
| 4.1 | Przygotowanie zmiennych środowiskowych | 3 |
| 4.2 | Pobranie wtyczek pomocniczych | 3 |
| 4.3 | Tworzenie wtyczki | 4 |
| 4.3.1 | Funckje wtyczki | 4 |
| 4.4 | Publikowanie wtyczki | 5 |
| 5 | Plik wejściowy | 5 |
| 5.1 | Uwagi | 5 |
| 5.2 | Błędy | 5 |
| 6 | Podsumowanie | 6 |

1 Cel Cwiczenia

Celem projektu było opracowanie wtyczki do oprogramowania QGIS, wykorzystującej narzędzie PyQGIS. Wtyczka miała umożliwiać przeprowadzanie różnych operacji geodezyjnych na danych geoprzestrzennych, w tym obliczanie różnicy wysokości między punktami oraz obliczanie pola powierzchni na podstawie współrzędnych zaznaczonych punktów metodą Gaussa. Dodatkowo, wtyczka miała posiadać funkcjonalności takie jak import plików, rysowanie poligonu na podstawie zaznaczonych punktów oraz opcję wyboru jednostek wyświetlania wyników obliczeń.

2 Wykorzystane narzędzia i materiały:

- zdalne repozytorium GitHub (LINK)
- Oprogramowanie QGIS: platforma, na której została opracowana wtyczka
- PyQGIS: biblioteka Python do programowania wtyczek QGIS
- Inne użyte biblioteki: QgsCoordinateReferenceSystem, QgsGeometry, QgsFeature, QgsField, QgsFields, QgsVectorLayer, QgsProject, QgsPoint, QgsWkbTypes, iface, QtWidgets, uic, math
- język programistyczny Python w wersjach 3.9.12 i 3.9.13
- zintegrowane środowisko programistyczne (IDE) PyCharm 2022.2.3
- zintegrowane środowisko programistyczne (IDE) Spyder 5.2.2
- Plik tekstowy .txt lub .csv: wykorzystany do importu danych geodezyjnych
- Qt Designer: narzędzie do wyświetlania danych tabelarycznych w interfejsie wtyczki
- zalecane wykorzystanie systemu operacyjnego Windows 10

3 Wstęp teoretyczny

Tworzenie wtyczek dla programu **QGIS** przy użyciu **biblioteki PyQt5** oraz **modułu "pyuic"** do konwersji **plików .ui** otwiera przed nami szerokie możliwości rozbudowy funkcjonalności QGIS. Wtyczki te pozwalają na dostosowanie QGIS do indywidualnych potrzeb oraz ułatwiają pracę z danymi geoprzestrzennymi.

Biblioteka PyQt5, będąca bindingiem dla **biblioteki Qt**, stanowi solidne fundamenty do tworzenia interfejsów użytkownika w programach desktopowych. Dzięki niej możemy projektować atrakcyjne i intuicyjne interfejsy, które poprawiają efektywność pracy użytkownika. W przypadku wtyczek dla QGIS, PyQt5 jest niezwykle przydatne ze względu na swoją wszechstronność i wydajność.

Proces tworzenia wtyczek QGIS z PyQt5 obejmuje wykorzystanie **narzędzia Qt Designer** do projektowania interfejsu użytkownika. Qt Designer to graficzne narzędzie, które umożliwia tworzenie interfejsów poprzez przeciąganie i upuszczanie elementów. W rezultacie otrzymujemy pliki .ui, które zawierają informacje o hierarchii elementów interfejsu oraz ich właściwości, takich jak rozmiar, położenie czy styl.

Aby zintegrować pliki .ui z wtyczką QGIS, korzystamy z **modułu "pyuic"**, który jest częścią biblioteki PyQt5. Dzięki temu narzędziu, możemy konwertować pliki .ui na pliki Pythona, które są łatwo integrowalne z kodem wtyczki. To umożliwia tworzenie instancji elementów interfejsu, dodawanie logiki, obsługę zdarzeń oraz dostosowywanie wyglądu wtyczki.

Wybór odpowiednich bibliotek w Pythonie zależy od specyfiki konkretnego zadania. Podczas tworzenia aplikacji lub skryptów w Pythonie, często napotykamy na różne problemy, które wymagają efektywnego rozwiązania. W takich sytuacjach warto poszukać odpowiednich bibliotek, które umożliwią nam szybkie i skuteczne rozwiązanie danego problemu. W przypadku tworzenia wtyczek dla QGIS, **biblioteka qgis.core** jest często wykorzystywana do manipulacji danymi geoprzestrzennymi oraz korzystania z **funkcji QGIS API**.

4 Przebieg Ćwiczenia

4.1 Przygotowanie zmiennych środowiskowych

Chcąc rozpocząć ćwiczenie, należało odpowiednio przygotować wszystkie zmienne środowiskowe, aby umożliwić skuteczne połączenie języka Python z programem QGIS. Istnieje kilka kluczowych kroków, które należy podjąć w celu zapewnienia prawidłowego środowiska pracy.

Po pierwsze, ważne jest upewnienie się, że QGIS został zainstalowany na komputerze. Można to zrobić pobierając najnowszą wersję QGIS z oficjalnej strony internetowej i postępując zgodnie z instrukcjami instalacji.

Po zainstalowaniu QGIS, kolejnym krokiem jest skonfigurowanie zmiennych środowiskowych w systemie operacyjnym. W przypadku systemu Windows, można to zrobić, otwierając menu Start i wyszukując "Zaawansowane ustawienia systemu". Następnie należy kliknąć "Zmienne środowiskowe" i dodać nową zmienną o nazwie "PYTHONPATH", której wartością jest ścieżka do folderu zawierającego biblioteki Pythona dla QGIS. Ta ścieżka może różnić się w zależności od lokalizacji instalacji QGIS, ale zwykle można znaleźć ją w folderze "apps" w katalogu QGIS.

Kolejnym krokiem jest dodanie ścieżki do folderu "bin" QGIS do zmiennej środowiskowej "Path". To umożliwi systemowi operacyjnemu odnalezienie niezbędnych plików wykonywalnych QGIS podczas wywoływania poleceń z Pythona.

Po skonfigurowaniu zmiennych środowiskowych, można rozpocząć pisanie skryptów Pythona, które będą korzystać z funkcji i bibliotek dostępnych w QGIS. W tym celu należy importować odpowiednie moduły QGIS, takie jak "qgis.core" i "qgis.gui". Następnie można tworzyć obiekty QGIS, takie jak warstwy wektorowe, obiekty projektu i mapy, oraz wykonywać różne operacje na danych geoprzestrzennych.

Ważne jest również pamiętanie o uruchomieniu interpretera Pythona w środowisku, które ma dostęp do zainstalowanego QGIS. Może to być zwykły interpreter Pythona, tak jak ten dostępny w terminalu systemu operacyjnego, lub środowisko programistyczne, takie jak PyCharm, które można skonfigurować do korzystania z interpretera Pythona z QGIS.

Wniosek jest taki, że aby skutecznie połączyć Pythona z programem QGIS, należy odpowiednio skonfigurować zmienne środowiskowe, zainstalować niezbędne moduły i biblioteki, oraz zapewnić prawidłowe środowisko pracy dla interpretera Pythona. Dzięki temu będziemy mogli pisać skrypty Pythona.

4.2 Pobranie wtyczek pomocniczych

Po skonfigurowaniu odpowiednich zmiennych środowiskowych i przygotowaniu środowiska pracy, możemy przejść do pobrania dwóch użytecznych wtyczek w programie QGIS: Plugin Builder i Plugin Reloader.

Plugin Builder to narzędzie, które ułatwia tworzenie nowych wtyczek dla QGIS. Umożliwia generowanie podstawowej struktury plików, szablonów kodu i interfejsu użytkownika dla nowego pluginu. Dzięki temu narzędziu możemy szybko rozpocząć pracę nad własnymi wtyczkami, dostosować ich funkcjonalność i interfejs do naszych potrzeb.

Plugin Reloader to wtyczka, która umożliwia dynamiczne przeładowanie innych wtyczek bez konieczności ponownego uruchamiania QGIS. Jest to bardzo przydatne narzędzie podczas procesu rozwoju wtyczek, ponieważ pozwala na bieżąco testować zmiany w kodzie i interfejsie użytkownika, bez potrzeby zamykania i ponownego uruchamiania QGIS.

4.3 Tworzenie wtyczki

1. Opracowanie wtyczki

- Stworzenie struktury wtyczki w oparciu o PyQGIS
- Utworzenie interfejsu użytkownika wtyczki
- Implementacja funkcji umożliwiających wybór punktów z warstwy oraz obliczanie różnicy wysokości i pola powierzchni.
- Dodanie obsługi komunikatów wynikowych na pasku informacyjnym interfejsu QGIS.
- Dodanie opcji importu pliku tekstowego i wyświetlania danych w tabeli
- Implementacja funkcji rysowania poligonu na podstawie zaznaczonych punktów

2. Testowanie wtyczki

- Wykonanie testów funkcjonalności wtyczki.
- Sprawdzenie poprawności obliczeń różnicy wysokości oraz pola powierzchni.
- Weryfikacja importu danych z pliku tekstowego.

3. Poprawki i optymalizacja:

- Dokonanie ewentualnych poprawek w kodzie wtyczki.
- Optymalizacja wydajności i interfejsu użytkownika.

4.3.1 Funkcje wtyczki

- **Liczenie liczby zaznaczonych obiektów**

Funkcja `count objects()` pozwala użytkownikowi na zliczenie liczby zaznaczonych obiektów na warstwie. Po kliknięciu przycisku "Count Objects", funkcja pobiera zaznaczone obiekty na aktualnej warstwie i wyświetla liczbę tych obiektów w etykiecie.

- **Generowanie współrzędnych wybranych punktów**

Funkcja `enter data on marked object()` umożliwia użytkownikowi generowanie współrzędnych wybranych punktów na aktywnej warstwie. Po kliknięciu przycisku "Show Coordinates", funkcja pobiera zaznaczone obiekty na aktywnej warstwie i wyświetla informacje o tych punktach w panelu wyświetlającym dane.

- **Obliczanie różnicy wysokości między punktami**

Funkcja `calculate height difference()` pozwala użytkownikowi na obliczanie różnicy wysokości między dwoma zaznaczonymi punktami na warstwie. Po kliknięciu przycisku "Calculate Height Difference", funkcja pobiera zaznaczone punkty na aktualnej warstwie, odczytuje ich współrzędne i wysokości, a następnie oblicza różnicę wysokości między nimi. Wynik jest wyświetlany w etykiecie.

- **Obliczanie pola powierzchni**

Funkcja `area()` umożliwia użytkownikowi obliczanie pola powierzchni dla zaznaczonych punktów na warstwie. Po kliknięciu przycisku "Calculate Area", funkcja pobiera zaznaczone punkty na aktualnej warstwie, sortuje je w kolejności przeciwnie do ruchu wskazówek zegara i oblicza pole powierzchni używając wzoru Gaussa. Wynik jest wyświetlany w etykiecie w wybranej jednostce powierzchni.

- **Czyszczenie danych**

Funkcja `clear info()` służy do czyszczenia wszystkich wyświetlanych danych. Po kliknięciu przycisku "Clear", funkcja czyści etykiety, listy i inne komponenty interfejsu użytkownika.

- **Generowanie raportu**

Funkcja `generate report()` umożliwia użytkownikowi generowanie raportu z wynikami obliczeń. Po kliknięciu przycisku "Generate Report", funkcja pobiera wyniki wszystkich przeprowadzonych

4.4 Publikowanie wtyczki

Kolejnym krokiem po stworzeniu wtyczki QGIS jest jej publikacja, aby inni użytkownicy mogli z niej skorzystać. Istnieje wiele sposobów publikacji wtyczek, ale jednym z najpopularniejszych i najbardziej wygodnych jest udostępnienie jej na platformie **GitHub**.

W celu rozpoczęcia pracy zespołowej nad wtyczką, stworzyliśmy **zdalne repozytorium na GitHubie**. Repozytorium to służy jako centralne miejsce, gdzie można przechowywać, zarządzać i udostępniać kod wtyczki. Dzięki temu każda osoba zaangażowana w projekt może łatwo zaktualizować kod, dodawać nowe funkcje czy poprawiać błędy, a wszyscy członkowie zespołu mogą mieć dostęp do najnowszej wersji wtyczki.

Praca zespołowa umożliwia efektywną współpracę nad rozwojem wtyczki. Każda osoba może przejść do wykonywania swojego zadania, wprowadzać zmiany do kodu i tworzyć nowe funkcje, które są następnie weryfikowane i integrowane w głównym repozytorium. Dzięki temu procesowi, wtyczka rozwija się w sposób spójny i efektywny, a wszyscy członkowie zespołu mają możliwość wniesienia swojego wkładu.

Platforma GitHub oferuje również narzędzia do monitorowania problemów (issues) oraz zarządzania wersjami (branching, merging), co ułatwia organizację pracy zespołowej i śledzenie postępów projektu.

Publikacja wtyczki na GitHubie umożliwia również innym użytkownikom QGIS dostęp do naszego kodu. Mogą oni pobrać wtyczkę, przetestować ją, zgłaszać błędy czy sugestie, a także wnieść swoje własne poprawki. To tworzy dynamiczną społeczność wokół wtyczki i przyczynia się do jej dalszego rozwoju.

5 Plik wejściowy

INPUT .TXT or .CSV FILE [Point_number; X[m]; Y[m]; H[m]]:

```
nr;X;Y;H
1;5653340,992;7532377,133;198,34
2;5653338,278;7532477,123;197,49
3;5653335,563;7532577,113;199,61
4;5653443,697;7532279,857;185,96
5;5653440,982;7532379,847;186,27
```

5.1 Uwagi

Jeśli plik zawiera punkty w różnych układach współrzędnych, użytkownik musi przed wybraniem punktów zaznaczyć warstwę po lewej stronie okna Qgis zawierającą interesujące go punkty, aby móc przejść do zaznaczania i dalszych obliczeń.

5.2 Błędy

Jeśli użytkownik wybierze zbyt małą ilość punktów do obliczenia przewyższenia lub pola, program wyświetli:

```
Not enough points selected
```

Jeśli użytkownik wybierze zbyt dużą ilość punktów do obliczenia przewyższenia program wyświetli:

```
Too many points were selected
```

Jeśli użytkownik najpierw obliczy pole powierzchni, a następnie przejdzie do generowania współrzędnych wybranych punktów, to nie zostanie wyświetlony żaden wynik, ponieważ operacje te odnoszą się do różnych warstw. Jeśli użytkownik chce uzyskać wynik generowania współrzędnych wybranych punktów, powinien najpierw przejść na odpowiednią warstwę, a następnie wybrać punkty i wywołać odpowiednią funkcję.

6 Podsumowanie

Link do zdalnego repozytorium GitHub → **repo**

Lub: <https://github.com/WiktoriaSamek/InformaticsPlugin.git>

Podczas tworzenia całego repozytorium i pisania skryptu implementującego dane transformacje zdecydowaliśmy się korzystać z języka angielskiego. Wynika to z faktu, że jest to język powszechnie używany w programowaniu i może ułatwić porozumienie z innymi programistami z różnych krajów. Ponadto wiele bibliotek i dokumentacji jest napisanych w języku angielskim, co ułatwia korzystanie z nich.

W trakcie trwania tego projektu nabyliśmy takie umiejętności jak:

- **Tworzenie wtyczek QGIS:** Zrozumienie procesu tworzenia wtyczek dla QGIS. Zapoznanie z modułami PyQGIS i wiedza, jak tworzyć interfejs użytkownika, obsługiwać zdarzenia, a także manipulować danymi przestrzennymi.
- **Praca z danymi przestrzennymi:** umiejętności manipulacji danymi przestrzennymi za pomocą biblioteki PyQGIS. Znajomość wczytywania warstw wektorowych, pobierania informacji o cechach, wykonywania operacji przestrzennych, takich jak obliczanie różnicy wysokości czy sortowanie punktów. Wykonywanie operacji na geometrii, takich jak sprawdzanie typu geometrii czy konwersja między typami.
- **Projektowanie interfejsu użytkownika:** W kodzie użyto pliku .ui wygenerowanego przez Qt Designer. Nabyliśmy wiedzę, jak korzystać z takiego pliku, wczytywać go i korzystać z elementów interfejsu, takich jak przyciski czy etykiety.

Literatura

(n.d.).

(n.d.).

https://docs.qgis.org/testing/en/docs/pyqgis_developer_cookbook/, title = *PyQGIS*

Developer Cookbook, year = 2023, (n.d.)

[https://docs.qgis.org/testing/en/docs/developers_guide/codingstandards.html?highlight=](https://docs.qgis.org/testing/en/docs/developers_guide/codingstandards.html?highlight=qt%2$0designer#qt-designer)

[qt%2\\$0designer#qt-designer](https://docs.qgis.org/testing/en/docs/developers_guide/codingstandards.html?highlight=qt%2$0designer#qt-designer), title = 1. *QGIS Coding Standards*, year = 2023, (n.d.)