

Praca domowa 2 – Generowanie modelu siatkowego LOD 1

Opis ogólny

Ćwiczenie polega na stworzeniu algorytmów pozwalających na półautomatyczne wygenerowanie danych geometrycznych w standardzie zbliżonym do CityGML LOD-1.

Dane wejściowe

Jako dane wejściowe do zadania należy wykorzystać warstwę wektorową przekazaną w załączniku do zadania (*hextiles.fgb*). Warstwa zawiera sześciokątną siatkę kafelków. Każdej osobie przypisany jest jeden kafelek. Indeks kafelek z warstwy wektorowej jest numer z załączonego do ćwiczenia wykazu.

Elementy zadania

W ramach zadania (w wersji podstawowej) należy stworzyć 3 skrypty:

Skrypt 1 – Pobieranie danych z GUGIK

- Jako dane wejściowe do skryptu traktowana jest ścieżka do pliku *hextiles.fgb*, indeks kafelek oraz folder do pobierania danych.
- Z warstwy wektorowej pobierana jest geometria kafelek o zadanym indeksie.
- Dla wybranego kafelek kompletowana jest lista koniecznych do pobrania sekcji NMT, NMPT oraz paczki BDOT. Sekcje modeli wysokościowych oraz paczki BDOT wybierane są na podstawie zapytań do WFS GUGIK testowanych na zajęciach.
 - Jako geometrię wystarczy wykorzystać jedynie BBOX kafelek.
- Skrypt samodzielnie pobiera pliki do wskazanego w danych wejściowych folderu.
- (Dla chętnych) W przypadku danych BDOT pobierana jest paczka ZIP. Można spróbować zaimplementować pobieranie w taki sposób, żeby od razu z paczki ZIP wyciągać jedynie dane z klasy obiektów BUBD_A. Alternatywnie dane z paczek ZIP należy wypakować ręcznie i zostawić jedynie dane BUBD_A
- (Dla chętnych) Każde źródło danych może składać się z więcej niż jednego „kawałka” (w przypadku BDOT – więcej niż jednego powiatu, w przypadku NMT i NMPT z więcej niż jednej sekcji). Można spróbować zaimplementować samemu łącznie poszczególnych typów danych (np. jeżeli pobrano 4 sekcje NMPT, to zostaną one połączone w jeden raster). Alternatywnie dane należy połączyć ręcznie (np. z wykorzystaniem QGIS).
 - <https://gis.stackexchange.com/questions/361213/merging-rasters-with-gdal-merge-py>
 - <https://write.agrevolution.in/merge-multiple-rasters-into-one-28df5d82ef3f>
 - https://geopandas.org/en/stable/docs/user_guide/mergingdata.html#appending

Skrypt 2 – Ograniczenie zakresu przestrzennego danych

- Jako dane wejściowe do skryptu traktowana jest ścieżka do pliku *hextiles.fgb*, indeks kafelek oraz 3 ścieżki do plików – raster NMT, raster NMPT, warstwa wektorowa z BUBD_A. Zakładamy, że na wejściu są połączone dane będące wynikiem działania skryptu 1 (po ewentualnym ręcznym łączeniu).
- Z warstwy wektorowej pobierana jest geometria kafelek o zadanym indeksie.
- Dla każdego typu danych wejściowych przycinamy dane do geometrii kafelek.
- Przycięte dane zapisujemy do wskazanego folderu.

Skrypt 3 – wygenerowanie pełnego modelu mesh

- Jako dane wejściowe do skryptu traktowane są ścieżki do przyciętych danych z plików – raster NMT, raster NMPT, warstwa wektorowa z BUDB_A. Zakładamy, że na wejściu są dane będące wynikiem działania skryptu 2.
- Dane NMT są konwertowane na model siatkowy mesh stanowiący podstawę finalnego LOD 1.
 - (Dla chętnych) W danych rastrowych można zdefiniować wartość NoData dla pikseli. Można spróbować dodać na etapie konwersji mechanizm usuwania pikseli NoData z modelu mesh. Wtedy w mesh nie będzie ogromnych „uskoków” na granicach obszaru kafla.
- Każdy budynek z warstwy wektorowej budynków zamieniany jest na prostopadłościan (bazujemy na skrypcie z zajęć). Wysokość podstawy budynku należy pobrać z NMT (może to być środek podstawy, może to być minimalna wysokość dla każdego z wierzchołków, można dla pewności odjąć jakąś wartość żeby budynek nie odrywał się od ziemi – generalnie bez różnicy), natomiast dach należy wypiętrzyć na wysokość pobraną z NMPT (tak jak poprzednio – wariant wyznaczania wysokości nie ma większego znaczenia; może to być wysokość dla środka poligonu, mediana wysokości...).
- Mesh z NMT i bryły budynków łączone są w jeden model i zapisywane w pliku PLY pod wskazaną przez użytkownika ścieżką.

Inne informacje

- Podział na 3 skrypty traktujemy jako w miarę sensowne minimum. Jako opcję dla ambitnych bardzo polecam próbę zamknięcia całego zadania w formie pojedynczego skryptu.
- Za zadanie można uzyskać bazowo 4 punkty (dobrze działające skrypty, bez punktów „dla chętnych”)
- Za wykonanie punktów „dla chętnych” i/lub próbę zmniejszenia liczby skryptów z 3 można zdobyć maksymalnie dodatkowy 1 punkt 😊

Przekazanie zadania

Jako wynik przekazujemy paczkę ZIP zawierającą:

1. Skrypty Python realizujące pracę domową
2. Plik PLY przetworzonego kafelka